MULTISENSOR SIGNAL PROCESSING: THEORY AND ALGORITHMS FOR IMAGE-BASED RENDERING AND MULTICHANNEL SAMPLING

BY

HA THAI NGUYEN

D.Ing., Ecole Polytehnique, 2001
D.Ing., Ecole Nationale Supérieure des Télécommunications, 2003
D.E.A., Université de Nice Sophia Antipolis, 2005

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2007

Urbana, Illinois

ABSTRACT

Multisensor applications have recently inspired important research projects to utilize existing infrastructure and exploit spatiotemporal information. This dissertation focuses on two multisensor applications: image-based rendering and multichannel sampling.

Although many *image-based rendering* (IBR) algorithms have been proposed, few of them possess rigorous interpolation processes. We propose a conceptual framework, called the *Propagation Algorithm*, that generalizes many existing IBR algorithms, using calibrated or uncalibrated images, and focuses on rigorous interpolation. We propose novel techniques to remove occlusions, for both calibrated and uncalibrated cases, and to interpolate the virtual image using both intensity and depth.

Besides algorithms, quantitative analysis is important to effectively control the quality and cost of IBR systems. We analyze the rendering quality of IBR algorithms using per-pixel depth. Working on the spatial domain, we consider the IBR problem as a nonuniform interpolation problem of the virtual image or the surface texture. The rendering errors can be quantified using the sample errors and jitters. We approximate the actual samples, in the virtual image plane or object surfaces, as a generalized Poisson process, and bound the jitters caused by noisy depth estimates. We derive bounds for the mean absolute error (MAE) for two classes of IBR algorithms: image-space interpolation and object-space interpolation. The bounds highlight the effects of depth and intensity estimate errors, the scene geometry and texture, the number of actual cameras, their positions and resolution. We find that, in smooth regions, MAE decays as $\mathcal{O}(\lambda^{-2})$ for 2D scenes and as $\mathcal{O}(\lambda^{-1})$ for 3D scenes, where λ is the local sample density.

Finally, motivated by *multichannel sampling* applications, we consider hybrid filter banks consisting of fractional delay operators, analog analysis filters, slow A/D converters, digital expanders, and digital synthesis filters to approximate a fast A/D converter. The synthesis filters are to be designed to minimize the maximum gain of an induced error system. We show the equivalence of this system to a digital system, used to design the synthesis filters using control theory tools, including model-matching and linear matrix inequality. The designed system is robust against delay estimate errors.

Kính tặng bố mẹ!

ACKNOWLEDGMENTS

First, I would like to sincerely thank Minh Do for being more than an excellent adviser during my graduate study. His enthusiasm, support, and confidence make me a better researcher. I thank Narendra Ahuja, David Forsyth, Thomas Huang, Yoshihisa Shinagawa, and Yi Ma for serving on my Ph.D. committee and providing me with valuable suggestions on my research. I am also grateful to Bruce Hajek, Daniel Liberzon, Geir Dullerud, and Trac Tran for enlightening discussions.

My colleagues and friends are great sources of ideas and support. I would like to thank Arthur da Cunha, Chinh La, Yue Lu, Mathieu Maitre, Robert Morrison, Hien Nguyen, Linh Vu, Chun Zhang, and Jianping Zhou. I will remember good moments in Urbana-Champaign with great friends "Robert" Hung Duong, "Danny" Quoc Le, Giang Nguyen, the Vietnamese soccer team, and other friends.

I thank Pascal Dorster, Zoran Nikolić, Thanh Tran, and Brooke Williams at Texas Instruments, Patrick Rault and Pankaj Topiwala at FastVDO, for wonderful hand-on experiences that opened me to a range of industrial applications.

My graduate study would have been impossible without the help I received even before I started. I am indebted to professors Truong Nguyen Tran, Tran Thanh Van, Nguyen Van Mau, and Nguyen Thien Thuat, among others, for being influential on my career and beyond. I also thank Quoc Anh, Duc Duy, Vu Hieu, Hong Son, and Dac Tuan for sharing enjoyable as well as difficult moments during my undergraduate study in France.

I thank my wife, Hạnh, *the* major discovery of my nonscientific career, for her sacrifice and patience, and my son, Duy, for being a source of inspiration.

Last but not least, I would like to thank my parents for their limitless support and guidance. This dessertation is dedicated to them.

TABLE OF CONTENTS

LIST OF FIGURES . . . . . **ix**

LIST OF TABLES . . . . . **xiii**

CHAPTER 1 INTRODUCTION **1**
1.1 Motivations and Challenges 1
1.1.1 Image-based rendering 1
1.1.2 Multichannel sampling 3
1.2 Related Work 5
1.2.1 Image-based rendering 5
1.2.2 Multichannel sampling 7
1.3 Problem Statement 7
1.4 Thesis Outline 8

CHAPTER 2 UNIFIED FRAMEWORK FOR CALIBRATED AND UNCALIBRATED IMAGE-BASED RENDERING . **10**
2.1 Introduction 10
2.2 Background 11
2.2.1 Problem setup 11
2.2.2 The Propagation Algorithm 12
2.2.3 Existing IBR algorithms 12
2.3 Calibrated IBR with Full Depth Information 13
2.3.1 Information propagation 13
2.3.2 Occlusion removal 13
2.3.3 Intensity interpolation 15
2.3.4 Experimental results 15
2.4 Calibrated IBR with Partial Depth 16
2.4.1 Motivations and approach 16
2.4.2 Segmentwise depth interpolation 18
2.4.3 Experimental results 19
2.5 Uncalibrated IBR Using Projective Depth 19
2.5.1 Motivations and approach 20
2.5.2 Occlusion removal in projective reconstructions 20
2.5.3 Triangulation-based depth approximation 22
2.5.4 Experimental results 22
2.6 Conclusion and Discussion 22

CHAPTER 3 QUANTITATIVE ANALYSIS FOR IMAGE-BASED RENDERING: 2D UNOCCLUDED SCENES . . . **26**
3.1 Introduction 26
3.2 Problem Setting 28
3.2.1 The scene model 28

3.2.2 The camera model 29
3.2.3 IBR algorithms and problem statement 30
3.3 Methodology 30
3.4 Analysis for an IBR Algorithm Using Image-Space Interpolation 32
3.4.1 Rendering using the Propagation Algorithm 32
3.4.2 Properties of sample intervals 33
3.4.3 Bound for sample jitters 35
3.4.4 Error analysis 36
3.4.5 Discussion 37
3.5 Analysis for an IBR Algorithm Using Object-Space Interpolation 38
3.5.1 A basic algorithm 38
3.5.2 Properties of sample intervals 39
3.5.3 Bound for sample jitters 40
3.5.4 Error analysis 41
3.5.5 Discussion 43
3.6 Validations 43
3.6.1 Synthetic scene 44
3.6.2 Actual scene 46
3.7 Discussion and Implications 47
3.7.1 Actual cameras with different resolutions 48
3.7.2 Where to put the actual cameras? 48
3.7.3 Budget allocation 49
3.7.4 Bit allocation 49
3.7.5 Limitations 50
3.8 Conclusion 50

CHAPTER 4 QUANTITATIVE ANALYSIS FOR IMAGE-BASED RENDERING: 2D OCCLUDED SCENES AND 3D SCENES 52
4.1 Introduction 52
4.2 Problem Setup 53
4.2.1 The scene model 53
4.2.2 The camera model 53
4.2.3 Problem statement 55
4.3 Analysis for 2D Scenes 55
4.3.1 Methodology 56
4.3.2 Part I revisited – analysis for 2D scenes without occlusions 57
4.3.3 Analysis for 2D occluded scenes 58
4.4 Analysis for 3D Scenes 62
4.4.1 Methodology 62
4.4.2 Properties of Poisson Delaunay triangles 64
4.4.3 Bound for sample jitters 65
4.4.4 Analysis for 3D unoccluded scenes 65
4.4.5 Numerical experiments 67
4.5 Conclusion 68

CHAPTER 5 MINIMAX DESIGN OF HYBRID MULTIRATE FILTER BANKS WITH FRACTIONAL DELAYS 70
5.1 Introduction 70
5.2 Equivalence of $\mathcal{K}$ to a Model-Matching Problem 74
5.2.1 Equivalence of $\mathcal{K}$ to a digital system 74
5.2.2 Equivalence of $\mathcal{K}$ to a finite-dimensional digital system . . 77
5.2.3 Equivalence of $\mathcal{K}$ to a linear time-invariant system 78

5.3 Design of IIR Filters 80
5.3.1 Conversion to the standard $\mathcal{H}_\infty$ control problem 80
5.3.2 Design procedure 81
5.4 Design of FIR Filters 81
5.4.1 Conversion to a linear matrix inequality problem 81
5.4.2 Design procedure 83
5.5 Robustness against Delay Uncertainties 83
5.6 Experimental Results 86
5.6.1 Example of IIR filter design 86
5.6.2 Example of FIR filter design 86
5.6.3 Comparison to existing methods 91
5.7 Conclusion and Discussion 92

CHAPTER 6 CONCLUSION AND FUTURE WORK 94
6.1 Conclusion 94
6.2 Future Work 95

APPENDIX A SUPPORTING MATERIAL 97
A.1 Geometrical Interpretation of $H_\Pi(u)$ 97
A.2 Proof of Proposition 4.1 98
A.3 Geometrical Interpretation of $\boldsymbol{H}_\Pi(u,v)$ 100
A.4 Review of State-Space Methods 101
A.5 Computation of the Norm of BB^* 102

REFERENCES 104

AUTHOR'S BIOGRAPHY 113

LIST OF FIGURES

Figure Page

2.1: Illustration of the Occlusion Removal step. Point $\boldsymbol{C}$ is considered occluded by point $\boldsymbol{A}$ and therefore is removed. Point $\boldsymbol{B}$ is not removed by looking at point $\boldsymbol{A}$ alone. 14

2.2: Pseudocode of the Occlusion Removal step. All points in whose neighborhood there exist other points with sufficiently smaller depth are removed. Parameters ε and σ are used to determine neighborhood and to differentiate surfaces. 14

2.3: Inputs of the Propagation Algorithm using full depth. (a) Image at $u_0 = 2$, (b) image at $u_1 = 6$, (c) depth at $u_0 = 2$, (d) image at $u_1 = 6$. 16

2.4: Rendered image of the Full Depth Propagation Algorithm. (a) The ground truth image taken at $u_v = 4$, and (b) the rendered image at $u_v = 4$. 17

2.5: The scene at $u_0 = 2$ and depth-available pixels on a regular grid of dimension 6×6 (highlighted dots). 17

2.6: Segmentwise interpolation of depth. Dots are intensity pixels, and circles are depth-available pixels. We interpolate the depth for each unit square of depth-available pixels. Bilinear interpolation is used for square falling inside the same depth segment. Nearest neighbor interpolation is used for segment-crossing unit squares. 18

2.7: The reconstructed depth at actual camera $u_1 = 6$ using segmentwise depth interpolation technique. The depth at $u_0 = 2$ is obtained similarly. 19

2.8: Virtual image at $u_v = 4$ rendered using depth at about 3% of pixels. To be compared with the case of full depth in Fig. 2.4. . . 19

2.9: Chirality parameter χ preserves the scene structure better than depth under projective transformations. All x-axis is image plane. (a) Surface points seen from a camera; y-axis is depth. (b) The scene after being applied a projective transformation; y-axis is projective depth. (c) The projective scene with y-axis is the chirality parameter $\chi = -1/d$. 21

2.10: Inputs at actual camera Π_2, Π_4. For each camera we have as inputs the intensity image and the set of feature points (circles). The Delaunay triangulation of feature points is also plotted. We will interpolate the depth in each of these triangles. 23

2.11: The rendered *Model House* image at Π_3. (a) The ground truth image at Π_3. (b) The rendered image at Π_3. 24

3.1: The 2D calibrated scene-camera model. The scene surface is modeled as a parameterized curve $\boldsymbol{S}(u)$ for $u \in [a, b] \subset \mathbb{R}$. The texture map $T(u)$ is "painted" on the surface. We assume pinhole camera model with calibrated projection matrix $\Pi = [\mathsf{R}, \boldsymbol{T}] \in \mathbb{R}^{2\times 3}$. The camera resolution is characterized by the pixel interval Δ_x on the image plane. 28
3.2: Linear interpolation. The interpolation error can be bounded using sample errors $\varepsilon_1, \varepsilon_2$, sample positions x_1, x_2, and their jitters μ_1, μ_2. 31
3.3: Sample intervals and jitters at virtual camera Π_v. Samples $\{y_{i,n}\}$ in virtual image plane are propagated from actual pixels $\{x_{i,n}\}$. The jitter $\mu = \widehat{y}_{i,n} - y_{i,n}$ is caused by a noisy estimate $\boldsymbol{S}_e$ of surface point $\boldsymbol{S}(u_n)$. 33
3.4: The reconstructed scene $\widehat{\boldsymbol{S}}(u)$ using piecewise linear interpolation. The intensity at virtual pixel y is the interpolated intensity at an approximated surface point $\widehat{\boldsymbol{S}}(\widehat{u})$ instead of the actual surface point $\boldsymbol{S}(u)$. 39
3.5: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the number of actual pixels in the loglog axis. Both the MAE and the theoretical bound decay with slope $s = -2$, consistent with the result of Theorem 3.1. 44
3.6: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the texture estimate error bound E_T. . 45
3.7: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D. . . 45
3.8: The scene's ground truth at the virtual camera $C_v = 4$. 46
3.9: The mean absolute error (solid) of the virtual image rendered using the Propagation Algorithm compared to the estimated error bound of Theorem 3.1 (dashed) for each scanline of the scene shown in Fig. 3.8. 48

4.1: The 3D calibrated scene-camera model. The scene surface is modeled as a 3D parameterized surface $\boldsymbol{S}(u, v)$ for $(u, v) \in \Omega \subset \mathbb{R}^2$. The texture $T(u, v)$ is "painted" on the surface. We assume pinhole camera model with calibrated positional matrix $\Pi \in \mathbb{R}^{3\times 4}$. The camera resolution is characterized by the pixel intervals Δ_x, Δ_y in horizontal and vertical direction on the image plane. 54
4.2: Linear interpolation of a discontinuous function. 56
4.3: A 2D occluded scene. We differentiate two kinds of discontinuities: those due to occlusions (such as $x_{d,n}$ with parameters $u^+_{d,n}$ and $u^-_{d,n}$) and those due to the texture $T(u)$ (such as $x_{t,n}$ with parameter $u_{t,m}$). 59
4.4: The observed interval $[y_{m_n}, y_{m_n+1}]$ around a discontinuity x_n of the virtual image $f_v(x)$. Note that the sample density function $\lambda_x(x)$ may or may not, depending on whether $x_n \in \mathcal{X}_d$ or $x_n \in \mathcal{X}_t$, be discontinuous at x_n. 60
4.5: Triangulation-based linear interpolation is often used with the Delaunay triangulation. For each triangle, the interpolation error can be bounded using the circumcircle radius R, the sample errors ε, and the sample jitters $\boldsymbol{\mu}$ (see Proposition 4.4). 63

4.6: The rendering errors plotted against the total number of actual pixels. We note the errors indeed have decay $\mathcal{O}(\lambda^{-1})$, where λ is the local sample density, as stated in Theorem 4.3. 67
4.7: The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the intensity estimate error bound E_T. 68
4.8: The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D. . . 69

5.1: (a) The desired high-rate system, (b) the low-rate system. The fast-sampled signal $y_0[n]$ can be approximated using slow-sampled signals $\{x_i[n]\}_{i=1}^N$. 71
5.2: The hybrid induced error system $\mathcal{K}$ with analog input $f(t)$ and digital output $e[n]$. We want to design synthesis filters $\{F_i(z)\}_{i=1}^N$ based on the transfer function $\{\Phi_i(s)\}_{i=0}^N$, the fractional delays $\{D_i\}_{i=1}^N$, the system delay tolerance m_0, the sampling interval h, and the super-resolution rate M to minimize the $\mathcal{H}_\infty$ norm of the induced error system $\mathcal{K}$. 71
5.3: The hybrid (analog input digital output) subsystem $\mathcal{G}$ of $\mathcal{K}$. Note that the sampling interval of all channels is h. 75
5.4: The $\mathcal{H}_\infty$ norm equivalent digital system K_d of $\mathcal{K}$ (see Proposition 5.3). Here $\{\mathsf{H}_i(z)\}_{i=0}^N$ are rational transfer functions defined in (5.16). Note that the input $\boldsymbol{u}[n]$ is of n_u dimension. 78
5.5: The equivalent LTI error system $\mathsf{K}(z)$ (see Theorem 5.1). Note that the system $\mathsf{K}(z)$ is Mn_u input M output, the transfer matrices $\mathsf{W}(z), \mathsf{H}(z)$ are of dimension $M \times Mn_u$, and $\mathsf{F}(z)$ is of dimension $M \times M$. 79
5.6: The induced error system $\mathsf{K}(z)$ of the form of the standard problem in $\mathcal{H}_\infty$ control theory with input $\boldsymbol{u}_p[n]$ output $\boldsymbol{e}_p[n]$. We want to design synthesis system $\mathsf{F}(z)$ to minimize $\|\mathsf{K}\|_\infty$. 80
5.7: The hybrid system $\mathcal{K}$ and the uncertainty operator Δ caused by delay estimate errors. 84
5.8: Example of IIR filter design. The magnitude and phase response of the transfer function $\Phi(s)$ modeling the measurement device. We use $\Phi_i(s) = \Phi(s)$ for $i = 0, 1, 2$. 87
5.9: Example of IIR filter design. The magnitude and phase response of synthesized IIR filters $F_1(z)$ (dashed), and $F_2(z)$ (solid) designed using the proposed method. The order of $F_1(z)$ and of $F_2(z)$ are 28. 87
5.10: Example of IIR filter design. The error $e[n]$ (solid) plotted against the desired output $y_0[n]$ (dashed). The induced error is small compared to the desired signal. The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4.68\%$. 88
5.11: The magnitude and phase response of the transfer function $\Phi(s)$ modeling the measurement devices. We use $\Phi_i(s) = \Phi(s)$ for $i = 0, 1, 2$. 89
5.12: The equivalent analysis filters $\mathsf{H}_0(z)$ of the first channel. Since $\mathsf{H}_0(z)$ takes multiple inputs, in this case $n_u = 4$ inputs, the i-th input is passed through filter $\mathsf{H}_{0i}(z)$ for $1 \le i \le 4$. 89
5.13: The magnitude and phase response of synthesis FIR filters $F_1(z)$ (dashed), and $F_2(z)$ (solid) designed using the proposed method. 90

5.14: The error $e[n]$ (solid) plotted against the desired output $y_0[n]$ (dashed). The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4\%$. 90
5.15: The norm $\|\mathcal{K}\|_\infty$ of the induced error system plotted against jitters δ_1 (solid) and δ_2 (dashed). 91
5.16: Performance comparison of the error of the proposed method (solid) and of the Sinc method truncated to 23 taps (dotted). . . 92
5.17: Error comparison between the proposed method (solid) and the Separation method (dotted). 93

A.1: The derivative $H'_{\Pi}(u)$ is proportional to the area S_{QSC} of the triangle $\triangle\mathsf{QSC}$, and inversely proportional to the square of the depth. 98
A.2: Linear interpolation error. 100

LIST OF TABLES

Table Page

3.1: Comparison of moments $E[(y_{m+1} - y_m)^k]$, for $N = 10$ actual cameras, with moments of the approximated Poisson process. . . 50

4.1: Experimental values of $E[R^2], E[S]$, and $E[S^2]$ in the case where $N = 10$ actual cameras are used, compared to theoretical values of Poisson Delaunay triangles. 67

5.1: Performance comparison using different inputs. Columns RMSE_1 and Max_1: step function input as in (5.42). Columns RMSE_2 and Max_2: input $f(t) = \sin(0.3t) + \sin(0.8t)$. 92

CHAPTER 1

INTRODUCTION

1.1 Motivations and Challenges

Hardware technologies have shown tremendous advancements in recent years. These advancements significantly decrease the cost of measurement devices, such as digital cameras, analog-to-digital converters, and sensors. As a result, in many applications, one can use more and more devices in the measurement process. Furthermore, pushing the limit of hardware technologies is often hard and expensive for a given application. An alternative is to design algorithms and systems, called *multisensor systems*, to fuse the data measured and processed from many inexpensive devices.

Collecting data from different measurement devices has additional rationales. In many cases, systems built from few but very high performance devices can be less robust than systems that use a large number of inexpensive devices and appropriate algorithms. Moreover, in some applications, such as sensor networks [1, 2], image-based rendering [3, 4, 5], and sound-field reconstruction [6], using multiple sensors can also provide users with crucial spatiotemporal information to exploit that one high performance measurement device alone cannot produce.

In this context, multisensor algorithms and systems need to be developed to efficiently exploit a large amount of data collected using multiple sensors. Moreover, faithful analysis of these multisensor algorithms and systems is also necessary to control the quality and cost of multisensor systems.

In this thesis, we particularly focus our interest in two types of multisensor systems: image-based rendering and multichannel sampling. In the following, we present motivations and challenges for both types of applications.

1.1.1 Image-based rendering

The goal of *photorealism* is a Holy Grail for the field of computer graphics. For many years, scientists have been endeavoring to produce high quality images that are indistinguishable from images taken from natural scenes. To date, the advancement is encouraging. Beautiful images of human beings, animals, man-made objects, and landscape are successfully rendered. Commercial advertise-

ments produced by computer graphics technologies are successfully introduced to the demanding public. Interactive computer and video games attract the attention of millions of players around the world. Computer-aid design (CAD) applications facilitate the jobs of a wide range of professionals.

The state of the art of computer graphics has been pushed forward, very far certainly, but not without a cost. In the search for photorealism, more and more details are added to model the geometry of the scene, reflectance properties of the object surfaces, and lighting conditions [7]. The modeling process becomes a very complex job, depending on the scene's complexity, and hence requires well-trained experts. With so many parameters taken into account, the rendering process becomes very time-consuming, sometimes taking hours or days to render a single image. Although some dedicated hardware is designed to speed up the rendering process, the high cost required certainly will inhibit the spread of the applications to all their potential users. Despite these collective efforts, synthesis images are still differentiable from natural images.

While the scene geometry, its physical properties, and the lighting are hard to model, a certain level of these properties can be learned from images. Characteristics of the scene–such as the surface texture and lighting–although very hard to model, can be easily "imitated" from real images. Moreover, the maturity of computer vision algorithms [8, 9] in the mid-1990s allowed understanding of the 3D scene geometry from multiple-view images.

In this context, *image-based rendering* (IBR) is developed as an alternative to model-based rendering techniques of computer graphics. IBR applications synthesize novel (or virtual) images, as taken by virtual cameras at arbitrary viewpoints, using a set of acquired images. Potential applications of IBR include virtual reality [10, 11], telepresence [12], augmented reality [13, 14], and 3D television [15, 16].

Virtual reality applications use computers to simulate environments, in many cases visual, that offer participants some desired experiences. Telepresence applications, such as video conferencing, produce experiences in which people feel present at some remote location different from one's physical location. In augmented reality, virtual objects are introduced into natural images of a real scene in order to create some particular effects on visual perceptions. Finally, 3D television applications create illusions that different objects have different depths. The effects are enabled by the screen sending a customized image to different user viewpoints.

With IBR technologies promising many potential applications, IBR challenges attract the efforts of many scientists from computer graphics, computer vision, and signal processing. Main IBR challenges concern IBR representations and algorithms, compression, and sampling.

A major challenge of the problem of IBR is to find representations and algorithms to effectively exploit the hidden information of the scene from actual images. If 3D models of the scene geometry, assumed in computer graphics

methods as prior knowledge, are no longer appropriate, what form of geometrical information is optimal for the rendering of the virtual images while keeping the amount of IBR data and computation acceptable? Existing IBR algorithms seem to choose the depth information, explicitly as depth maps or implicitly as feature correspondences [3, 4]. Furthermore, as many IBR algorithms are ultimately intended for real-time applications, it is important that IBR algorithms are fast and reliable.

Another importance of representations of IBR data is for compression. Effective data representations also enable compact compression, which is highly necessary in IBR applications since the size of IBR data are typically very large compared to images and videos. IBR compression techniques should possess at least the basic requirements of video compression. However, IBR data are expected to contain more redundancies, both in time and space. In some IBR applications, we also desire the ability of random access down to the pixel level to facilitate the rendering process of IBR algorithms.

Finally, the problem of IBR can be considered as an instance of the sampling and reconstruction framework [17, 18, 19, 20, 21]. Hence, fundamental questions of the classical sampling and reconstruction framework need to be addressed. These questions include how many samples (in this case the number of cameras and/or resolution) and the optimal sample locations (i.e., camera locations) to ensure the rendering quality of IBR systems. Also, what level of the scene's characteristics, such as the scene geometry and texture, is necessary to ensure a predefined rendering quality? Work toward answering these questions is very important; we cannot control the quality and cost of IBR systems without faithful analysis of these fundamental questions. In fact, many existing IBR algorithms have to rely on oversampling to limit the effects of aliasing on the rendered images [22, 23].

These IBR challenges pose various difficulties. IBR data are non-uniform, even if the cameras are uniformly placed in the scene, due to projective mapping from the scene surfaces to the camera image planes. Moreover, IBR data belong to vector spaces of high dimensions; in particular, the plenoptic function [24] is a function of seven variables. Finally, IBR algorithms are highly nonlinear, in particular because of the occlusions of surface points and because the scene surfaces exhibit non-Lambertian properties.

1.1.2 Multichannel sampling

With the explosion of the digital era, many analog data such as image, audio, speech, and text become available in digital formats. Key technical issues arising from this context include data conversion from continuous-domain to discrete-domain, called the *sampling process*, and from discrete-domain to continuous-domain, called the *reconstruction process*. As a result, the problem of sampling and reconstruction recently has become a very active research area [25].

A classic result on sampling and reconstruction dates back to the Shannon reconstruction formula [26, 27]

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT) \cdot \mathrm{sinc}(t/T - n), \tag{1.1}$$

where $f(nT)$ are equidistant samples of a function $f(t)$ whose bandwidth is bounded by the Nyquist frequency $f_N = 1/(2T)$. Equation (1.1) is fundamental in the design of analog-to-digital (A/D) converters. If the bandwidth limit condition of $f(t)$ is satisfied, we can sample $f(t)$ for *lossless* storage and transmission using digital communications channels and devices.

Equation (1.1) can be considered in another way. Given an analog function $f(t)$, a faithful sampling of function $f(t)$ requires the use of A/D converters with sample interval $T < 1/(2f_{\max})$. Hence, if function $f(t)$ has energies at high frequencies, the sample interval T needs to be small enough to capture these high frequency energies of $f(t)$, hence avoiding aliasing. However, in some applications, decreasing the sample interval T is not preferable, if not impossible, because of limits of hardware technologies. A novel sampling framework using multiple sensors, or *multichannel sampling*, arises from this context as a necessity. Potential applications of multichannel sampling include superresolution and data fusion.

Superresolution applications [28, 29, 30] enhance the resolution of imaging systems using signal processing techniques. In these applications, the sample interval T in (1.1) can be understood as the size of pixels on the chip. Decreasing T will cause unexpected levels of noise, especially shot noise, in the acquired images [28]. Hardware technologies to reduce the pixel size are replaced by signal processing alternatives to reduce system cost and to utilize existing low-resolution imaging systems.

Another application of multichannel sampling is to fuse low resolution samples of analog signals, such as speech or audio signals, to obtain high resolution samples [31, 32]. Multiple *slow* A/D converters, with large sample interval T (measured in time or space), are used to sample analog signals. These samples can be fused to synthesize high resolution signals, as if it is sampled using a *fast* A/D converter. Because of its low cost, this approach is preferred to using fast A/D converters directly.

In order to build multichannel sampling systems, several problems need to be addressed. First, we need to align low-resolution digital signals with the precision of a fraction of the sample interval. This problem is very challenging because we only know the signal's value at discrete positions; the intersample behaviors are not apparent. Moreover, we need to design efficient algorithms and analyze their performances faithfully. These problems are difficult because the system is inherently hybrid and a lot of information of the signal is lost in the sampling process.

1.2 Related Work

We discuss related work for image-based rendering applications in Section 1.2.1 and for multichannel sampling applications in Section 1.2.2.

1.2.1 Image-based rendering

Adelson and Bergen, although not being aware of image-based rendering (IBR) applications, introduced in 1991 the *plenoptic function* [24] that afterward helped to define mathematically the problem of image-based rendering as a reconstruction problem. The appearance of the scene is considered as the dense array of light rays. For each pinhole camera, the light rays passing the camera center determine the image. The plenoptic function is a 7D variable function of the form $P = P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$ characterizing the intensity of the light ray passing through location (V_x, V_y, V_z) with direction (θ, ϕ), for every wavelength λ, and at every time t.

Another step toward the emergence of image-based rendering was the maturity of computer vision algorithms. The Eight Point algorithm [33, 34] was proposed in 1987 to reconstruct the scene's 3D structure (the essential matrix) using as few as eight nondegenerate correspondences at calibrated actual images. (In fact, the 3D structure of the scene can be determined using as few as five correspondences using the nonlinear Kruppa equation [9].) In the case where only uncalibrated images are available, we can reconstruct the scene's 3D structure in the form of the fundamental matrix [35]. The fundamental matrix helps to recover the scene geometry up to an unknown projective transformation [8, 9, 36]. In summary, information of the scene's 3D structure can be exploited from actual images of the scene.

Among IBR pioneers, Chen and Williams introduced view interpolation [37, 38] that rendered in-between images by interpolating the image flows. Laveau and Faugeras [39] took correspondences to predict virtual images using some projective basis. Their method also allowed one to resolve occlusions using the knowledge of vanishing points without the reconstruction of the 3D scene geometry.

The problem of IBR was not well-defined mathematically until 1995, when McMillan and Bishop introduced [40] the term "image-based rendering" and recognized the connection of the IBR problem with the reconstruction of the plenoptic function. They proposed that the IBR problem is nothing but to reconstruct the plenoptic function (virtual images) using a set of discrete samples (that is actual images). McMillan also proposed [41] the warping equation to transfer actual pixels to the virtual image plane, and a technique to compute the visibility of samples at the virtual camera, using a painterlike algorithm.

Without any knowledge of the scene geometry, how much can a purely image-based rendering system offer in terms of the rendering quality? In 1996, Gortler

et al. [22] introduced Lumigraph while Levoy and Hanrahan [23] described light field rendering. Both systems interpolated the virtual images in the ray-domain. However, they relied on a large number of images to compensate for the lack of geometry.

Debevec et al. [42, 43] proposed a mixture between the model-based approach of computer graphics [7] and the image-based approach. Their system starts with simple modeling primitives of the scene and uses images to tune the model to be more faithful to the 3D scene geometry. The virtual images are rendered using 3D warping technique [41] and view-dependent texture mapping, a technique that conceptually interpolates samples in the 3D space, having fast image-space implementations using perspective correction [44].

Many IBR algorithms have been proposed to date. A prominent classification of IBR algorithms, proposed by Shum et al. [3, 4], is based on the level of geometrical information used as a priori knowledge. In this continuum, IBR algorithms using explicit depth maps include Unstructured Lumigraph Rendering [45], and Layered Depth Images [46], while IBR algorithms using feature correspondences as implicit geometrical information include View Morphing [47], and Joint View Triangulation [48].

While many IBR methods have been proposed, little research has addressed the fundamental question of the sampling and reconstruction of the plenoptic function and/or the rendering quality of IBR algorithms. In [49], Chai et al. analyzed the minimum number of images necessary to reconstruct the light field (a special case of the plenoptic function). They also provided a minimum sampling curve that determines the tradeoff between the number of images and the scene geometry. In [50], Zhang and Chen proposed a surface plenoptic function to analyze non-Lambertian and occluded scenes. In [51], Chan and Shum considered the problem of plenoptic sampling as a multidimensional sampling problem to estimate the spectral support of the light field given an approximation of the depth function.

In the analysis of IBR data, most existing literature addresses the Fourier domain because the IBR data exhibit fan-type structure in the frequency domain. However, Do et al. [52] showed that in general the plenoptic is not bandlimited unless the surface of the scene is flat. A similar conclusion was also reached by Zhang and Chen [50].

The problem of IBR data compression is still understudied. In [53, 54], the depth maps are compressed using the JPEG2000 encoder [55]. In [56], Taubin discussed the problem of geometry compression. Light fields [23] and Lumigraph [22] proposed their own compression techniques. In [57], Duan and Li addressed the problem of compression layered depth images. In [58], Fehn presented a compression and transmission technique for 3D television applications. Many of the current techniques are adapted from image and video compression techniques; the redundancies typical of IBR data are not fully exploited. Hence, there is certainly room for improvements in IBR data compression.

1.2.2 Multichannel sampling

The problem of sampling and reconstruction was formulated almost 60 years ago in a seminal paper of Shannon [26, 27] by means of the reconstruction formula (1.1). In the mathematical literature, the result is known [25] as the cardinal series expansion that can be traced further back to Whittaker [59].

The modern approach formulates the Shannon reconstruction formula (1.1) as an orthogonal projection of analog signals to a Hilbert space formed by a kernel function, such as the sinc function or B-splines, and its shifted and scaled versions [17, 25].

The first attempt to generalize the sampling and reconstruction problem to multichannel sampling was proposed by Papoulis in 1977 [32]. Papoulis showed that a band-limited signal $f(t)$ could be perfectly reconstructed from the equidistant samples of the responses of m linear shift-invariant systems with input $f(t)$, sampled at $1/m$ the Nyquist rate.

From a more general viewpoint, the sampling operator is usually part of systems with digital implementations (hybrid systems). In this situation, sampled-data control techniques [60] can be used to take into account intersample behaviors of the signals. Moreover, systems with multichannel sampling as a component are inherently multirate [31]. Literature on multirate systems can be found in an excellent book of Vaidyanathan [61].

1.3 Problem Statement

This thesis is concerned with two different multisensor applications, namely, image-based rendering (in Chapter 2, 3, and 4) and multichannel sampling (in Chapter 5).

For the problem of image-based rendering (IBR), our goal is to bring rigorous ampling and reconstruction techniques to IBR algorithms and analysis. Specifically, we would like to:

IBR algorithm. Develop IBR algorithms to generate valid views of 3D scenes using acquired calibrated or uncalibrated images and full or partial depth maps. We focus on the interpolation process using rigorous sampling and reconstruction frameworks.

IBR analysis. Analyze the performance of IBR algorithms based on the characteristics of the scenes, such as the scene geometry and texture, and of the camera configuration, such as the number of actual cameras and their positions and resolutions.

For the multichannel sampling problem, our objective is to exploit the intersample behaviors of the signals to improve the performance over existing techniques. Specifically, we would like to:

Filter design. Design IIR and FIR filters to complete a hybrid system that approximates the output of a fast A/D converter using outputs of multiple slow A/D converters. The goal is to optimize the gain of an induced error system.

1.4 Thesis Outline

We dedicate the next three chapters (Chapters 2, 3, and 4) to the problem of image-based rendering (IBR). The subsequent chapter (Chapter 5) is concerned with the multichannel sampling problem. The specific outlines of the chapters are given in the following.

In Chapter 2, we propose a conceptual framework, called the *Propagation Algorithm*, that generalizes many existing IBR algorithms, using calibrated or uncalibrated images, and focuses on rigorous interpolation techniques. The framework consists of three steps: information collection to the virtual image plane, occlusion removal, and interpolation of the virtual image. We apply the framework to three different IBR scenarios–namely, calibrated IBR with full or partial depth, and uncalibrated IBR using sparse correspondences–by proposing innovative techniques. These techniques include occlusion removal for both calibrated and uncalibrated cases, interpolation using depth and intensity, and segmentwise and image-consistent depth interpolation. Experiments show that the proposed Propagation Algorithm obtains excellent rendering quality.

Next, we provide a quantitative analysis of the rendering quality of image-based rendering (IBR) algorithms per-pixel depth in Chapter 3. Assuming the ideal pinhole camera model and 2D unoccluded scene, we show that IBR errors can be quantified using sample intervals, sample errors, and jitters. We derive bounds for the mean absolute error (MAE) of two classes of IBR algorithms: image-space interpolation and object-space interpolation. The proposed error bounds highlight the effects of various factors including depth and intensity estimate errors, the scene geometry and texture, the number of actual cameras, their positions and resolution. Experiments with synthetic and actual scenes show that the theoretical bounds accurately characterize the rendering errors. We discuss implications of the proposed analysis on camera placement, budget allocation, and bit allocation.

In Chapter 4, we extend the analysis of IBR algorithms to 2D occluded scenes and 3D unoccluded scenes. For 2D occluded scenes, we measure the effects of jumps in sample intervals around the discontinuities of the virtual image, resulting in additional terms. For 3D scenes, we derive an error bound for triangulation-based linear interpolation and exploit properties of Poisson Delaunay triangles. We show that the mean absolute errors (MAE) of the virtual images can be bounded based on the characteristics of the scene and the camera configuration. An intriguing finding is that triangulation-based linear

interpolation for 3D scenes results in a decay order $\mathcal{O}(\lambda^{-1})$ of the MAE in smooth regions, where λ is the local density of actual samples, compared to $\mathcal{O}(\lambda^{-2})$ for 2D scenes.

Motivated by multichannel sampling applications, we consider in Chapter 5 hybrid multirate filter banks consisting of a set of fractional delay operators, analog analysis filters, slow A/D converters, digital expanders, and digital synthesis filters (to be designed). The synthesis filters are designed to minimize the maximum gain of a hybrid induced error system. We show that the induced error system is equivalent to a digital system. This digital system enables the design of stable synthesis filters using existing control theory tools such as model-matching and linear matrix inequalities (LMI). Moreover, the induced error is robust against delay estimate errors. Numerical experiments show the proposed approach yields better performance than existing techniques.

Finally, in Chapter 6, we present the conclusions and future work.

Each chapter is self-contained so that readers can start directly in the chapter of interest.

CHAPTER 2

UNIFIED FRAMEWORK FOR CALIBRATED AND UNCALIBRATED IMAGE-BASED RENDERING

2.1 Introduction

Image-based rendering (IBR) applications synthesize novel (or virtual) images, as taken by virtual cameras at arbitrary viewpoints, using a set of acquired images. With advantages of photorealism and low complexity over traditional model-based techniques, IBR has many potential applications, hence attracting many researchers in the field [3, 5].

Most IBR algorithms replace the scene's physical models with geometrical information such as explicit depth maps for calibrated images [41, 46, 64] and feature correspondences for uncalibrated images [38, 39, 47, 48]. In this context, we believe that the depth information offers a good tradeoff between the approaches of purely-images-alone [23] and model-based techniques [7]. Moreover, while many IBR techniques have been proposed by computer graphics and computer vision researchers, little work has concentrated on interpolation using rigorous signal processing frameworks.

In this chapter, we suggest that many IBR algorithms, using depth maps for calibrated images or correspondences for uncalibrated images, can be analyzed using a unified framework, called the *Propagation Algorithm*. The Propagation Algorithm possesses well-separated steps of information collection, occlusion removal, and intensity interpolation, hence opening areas for improvement in the interpolation process. Moreover, the Propagation Algorithm also facilitates quantitative analysis, as shown in Chapters 3 and 4.

The main contributions of this chapter include techniques proposed for the Propagation Algorithm framework. While the Propagation Algorithm is simple, it serves as a conceptual framework for different IBR configurations: calibrated images with full or partial depth information and uncalibrated images with feature correspondences. For calibrated images with full depth information, we

[0]This chapter includes research conducted jointly with Prof. Minh Do [62, 63].

propose a simple and adaptive technique to remove occlusions, and interpolate the virtual image using both the intensity and depth of visible samples, following the linear inverse framework. In the case of calibrated IBR with partial depth information, we propose to segmentwise interpolate the depth for all pixels before applying the Propagation Algorithm framework. Finally, for uncalibrated IBR, we propose to weakly calibrate the corresponding features and interpolate the projective depth using the Delaunay triangulation [65]. We also propose to resolve occlusions directly in the projective domain, using the chirality parameter [66].

The remainder of the chapter is organized as follows. In Section 2.2, we present the problem definition, the Propagation Algorithm framework, and literature review. In Section 2.3, we propose an algorithm for the case of calibrated IBR with full depth information. We consider calibrated IBR with partial depth information in Section 2.4. Uncalibrated IBR is treated in Section 2.5. Finally, we give concluding remarks in Section 2.6.

2.2 Background

We first present the problem definition in Section 2.2.1. Next, in Section 2.2.2, we describe the Propagation Algorithm framework. We discuss existing IBR texture mapping algorithms in light of the Propagation Algorithm in Section 2.2.3.

2.2.1 Problem setup

We assume that the scene surfaces are Lambertian [67], that is, images of a surface point at different cameras have the same intensity. In addition, the pixel intensity is assumed to be scalar valued. In the case where pixel intensity is vector valued, for example RGB, the algorithm simply performs each coordinate independently.

A 3D pinhole camera is characterized by a matrix $\Pi \in \mathbb{R}^{3\times 4}$. We denote $\widetilde{\boldsymbol{x}} = [x, y, 1]^T$ and $\widetilde{\boldsymbol{P}} = [X, Y, Z, 1]^T$ the *homogeneous coordinate* of $\boldsymbol{x} = [x, y]^T$ and $\boldsymbol{P} = [X, Y, Z]^T$, respectively. For a 3D surface point $\boldsymbol{P}$, its image position is determined by the projection equation

$$d \cdot [x, y, 1]^T = \Pi \cdot [X, Y, Z, 1]^T. \tag{2.1}$$

Problem definition. The inputs of our algorithm are the actual cameras' projection matrices $\{\Pi_i\}_{i=1}^N$, the intensity images $\{I_i(\boldsymbol{x})\}_{i=1}^N$, the depth maps $\{d_i(\boldsymbol{x})\}_{i=1}^N$, and the virtual projection matrix Π_v. The output is the virtual image $I_v(\boldsymbol{x})$.

2.2.2 The Propagation Algorithm

The key idea of the Propagation Algorithm [62] is to focus on signal processing techniques of IBR algorithms. At the virtual camera, we collect all available information provided by actual cameras and resolve occlusions to turn the IBR problem into a traditional nonuniform 2D interpolation problem. The Propagation Algorithm has the three following steps:

- **Information Propagation.** Using depth, surface points, whose images are actual pixels, are reconstructed and reprojected onto the virtual camera. The actual pixels are said to be *propagated* to the virtual image plane.
- **Occlusion Removal.** Remove all the points that are occluded at the virtual camera.
- **Intensity Interpolation.** Interpolate the virtual image using the depth and intensity of the visible points.

The approach of the Propagation Algorithm enables systematic investigation of the interpolation process using traditional signal processing techniques [21, 25, 68]. The framework also allows quantitative analysis of the rendering quality, as demonstrated in the next Chapters 3 and 4.

2.2.3 Existing IBR algorithms

In this section, we suggest that the interpolation process of many existing IBR algorithms can be analyzed using the Propagation Algorithm framework.

View-dependent texture mapping [43, 45]. In these IBR algorithms, actual pixels are propagated to the virtual image plane (projective mapping). The virtual image is interpolated as the weighted average of nearby samples. This interpolation scheme can be considered as being derived from some kernel function on the virtual image plane.

3D warping [41, 46, 64]. These IBR algorithms propagate patches, such as an elliptical weighted average filter [44], around actual pixels, instead of pixels themselves, to the virtual image plane. This process is equivalent to interpolate the virtual image from propagated samples (not patches) using warped versions of the kernel functions. The interpolation process hence can be analyzed using signal processing frameworks [21, 68, 69].

IBR using correspondences [37, 39, 47, 48]. These techniques usually transfer corresponding features to the virtual image plane using projective basis or epipolar constraints. We note a classical result [8, 36] in computer vision that 3D scene reconstruction can be reconstructed up to an unknown projective transformation (weak calibration) without affecting image-space constraints. Hence,

these IBR techniques can be conceptually thought of as texture mapping algorithms using projective depths.

In summary, the interpolation process of many existing IBR algorithms can be analyzed using the Propagation Algorithm framework. In the next three sections, applying the Propagation Algorithm framework, we propose an algorithm for three progressive challenging situations that have practical significance.

2.3 Calibrated IBR with Full Depth Information

In Sections 2.3.1–2.3.3, we present three steps of the Propagation Algorithm for IBR with calibrated images where the depth information is available at all actual pixels. Finally, experiments are given in Section 2.3.4.

2.3.1 Information propagation

The main idea of this step is to collect all the available information to the virtual camera before doing any further processing. At actual pixels, the depth information allows us to reconstruct the 3D surface points before reprojecting them to the virtual image plane. (Since we work on the continuous domain, the reprojection may not be at pixel positions.) We say that the intensity and depth information are *propagated* from actual pixels to the virtual image plane.

Let $\boldsymbol{e}_4 = [0, 0, 0, 1]^T$. Using the projection equation, a 3D point $\boldsymbol{X}$ can be recovered from its image $\boldsymbol{x}$ at a camera Π using the depth as follows:

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \Pi \\ \boldsymbol{e}_4^T \end{bmatrix}^{-1} \cdot \begin{bmatrix} d_\Pi \widetilde{\boldsymbol{x}} \\ 1 \end{bmatrix}. \tag{2.2}$$

Inversely, the depth of a 3D point $\boldsymbol{X}$ relative to camera $\Pi = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3]^T$ can be computed as the last coordinate of $\Pi\widetilde{\boldsymbol{X}}$, i.e.,

$$d_\Pi(\boldsymbol{x}) = \boldsymbol{\pi}_3 \widetilde{\boldsymbol{X}}. \tag{2.3}$$

2.3.2 Occlusion removal

In the Information Propagation step, actual pixels are propagated to the virtual camera Π_v without considering their visibility at Π_v. We present a technique to remove occlusions that is adaptive and simple to implement.

The Occlusion Removal step removes all points in whose neighborhood (parameterized by $\varepsilon \in \mathbb{R}$) there exist other points with sufficiently smaller depth (parameterized by $\sigma \in \mathbb{R}$). As illustrated in Fig. 2.1, in considering point $\boldsymbol{A}$, we create a removal zone (the shaded zone) for which all other points falling in

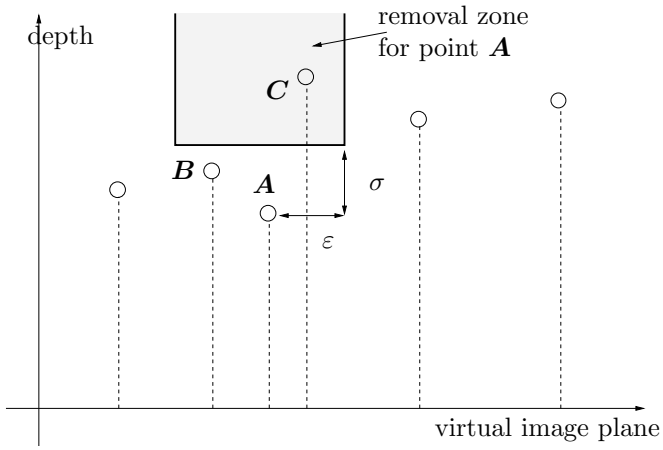


Figure 2.1: Illustration of the Occlusion Removal step. Point $\boldsymbol{C}$ is considered occluded by point $\boldsymbol{A}$ and therefore is removed. Point $\boldsymbol{B}$ is not removed by looking at point $\boldsymbol{A}$ alone.

```
% Remove occluded points from P to get Q
Q = P;
For each point x ∈ P
    If ∃ y ∈ P such that ||x − y||_2 ≤ ε
                         and d_v(x) − d_v(y) ≥ σ
        Q = Q − {x};
    Endif
Endfor
```

Figure 2.2: Pseudocode of the Occlusion Removal step. All points in whose neighborhood there exist other points with sufficiently smaller depth are removed. Parameters ε and σ are used to determine neighborhood and to differentiate surfaces.

this zone will be removed. Hence, point $\boldsymbol{C}$ is considered occluded by point $\boldsymbol{A}$ and therefore is removed. Point $\boldsymbol{B}$ is not removed by looking at point $\boldsymbol{A}$ alone; it may be visible or occluded by another point. The pseudocode of this step is shown in Fig. 2.2.

Determining σ and ε. The parameters σ and ε can be tuned according to the characteristics of the scene and/or applications. If a large σ is chosen, we risk keeping background points, whereas if σ is small we may remove foreground points of inclining surfaces. As for ε, a large value of ε removes more points, keeping only visible points with high confidence. However, it also removes background points around the boundaries, hence reducing the image quality. For small value of ε, again we risk keeping background points because no foreground point may fall in the neighborhood.

2.3.3 Intensity interpolation

Finally, when all relevant information is at our disposal, we interpolate the virtual image using the intensity information of remaining points $\mathcal{Q}$. The virtual image is then simply the value of this function at the pixel positions.

A major challenge at this step is to avoid interpolating samples of different surfaces around the edges. We propose to follow the linear inverse framework [70] with a regularization using the depth to allow fast transitions around edges. In the following, we limit ourselves to the 1D interpolation case. The case arises in configurations where all cameras are located in the same plane, or if we rectify actual images before the rendering process. The 2D case is left for future research.

Let $I_v(x)$ and $d_v(x)$ be the intensity and the depth at remaining samples $x \in \mathcal{Q} \subset \mathbb{R}$. We assume that the virtual image $I_v(x)$ belongs to a shift-invariant space formed by some kernel function $\varphi(x - m\Delta)$ [17], such as B-splines [21] or $\mathrm{sinc}(x)$. In other words, we solve for $I_v(x)$ of the form

$$J(x) = \sum_{m=1}^{M} c_m \varphi(x - m\Delta_x), \tag{2.4}$$

where M is number of virtual pixels. Our goal is to find coefficients $\boldsymbol{c} = \{c_m\}_{m=1}^{M}$ to minimize the following cost function

$$f(\boldsymbol{c}) = \sum_{x \in \mathcal{Q}} (I_v(x) - J(x))^2 + \lambda V_x (J'(x))^2, \tag{2.5}$$

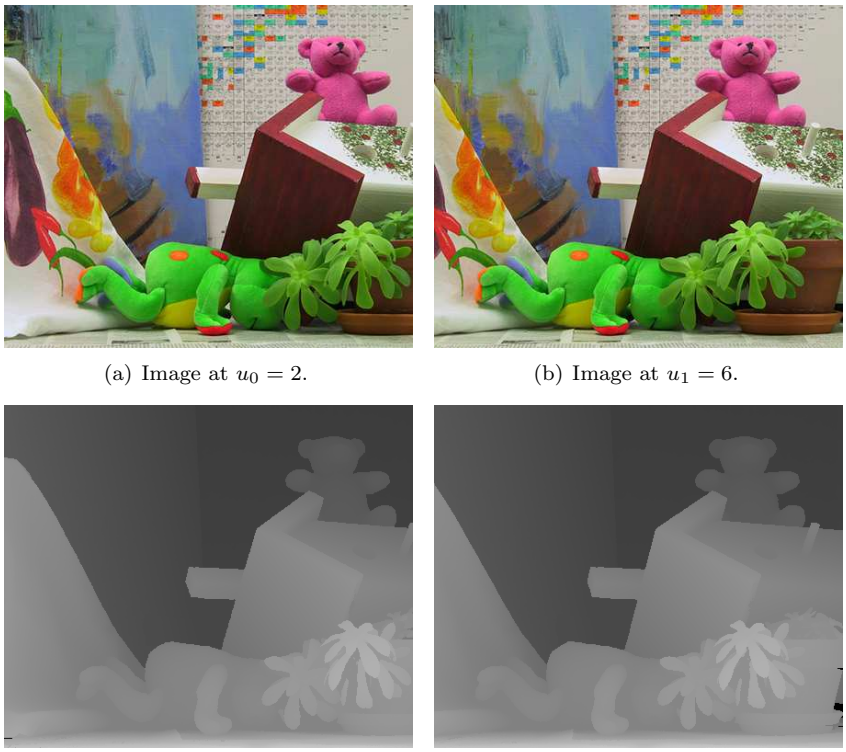
where V_x denotes the inverse of the local depth variation around point $\boldsymbol{x} \in \mathcal{Q}$. Again, similar to Section 2.3.2, two samples are local if their distance is smaller than a parameter ε.

The first term of $f(\boldsymbol{c})$ is to obtain a faithful approximation at sample points $\mathcal{Q}$. The idea of the second term (the regularization term) allows large derivative, hence sharp increase or decrease, around edges. The solution for the above minimization is standard [70] and can be solved using matrix inversions or gradient descent techniques.

2.3.4 Experimental results

Stereo data of a translational configuration, provided by Scharstein et al. [71],[1] are used in our numerical experiments. All the cameras placed in the line $y = 0$ and focused to the same direction of the y-axis. Two actual cameras, at $u_0 = 2$ and $u_1 = 6$, are used to render the virtual image at $u_v = 4$. Because the images are color, we render the virtual image for each color channel (RGB) separately. Using the *Teddy* images as inputs (Fig. 2.3), we plot the rendered image in

[1] The data is available at `http://cat.middlebury.edu/stereo/newdata.html`.



(a) Image at $u_0 = 2$.

(b) Image at $u_1 = 6$.

(c) Depth at $u_0 = 2$.

(d) Depth at $u_1 = 6$.

Figure 2.3: Inputs of the Propagation Algorithm using full depth. (a) Image at $u_0 = 2$, (b) image at $u_1 = 6$, (c) depth at $u_0 = 2$, (d) image at $u_1 = 6$.

Fig. 2.4. Parameters $\varepsilon = 0.6$ and $\sigma = 0.05 d_v(\boldsymbol{y})$ are used in the Occlusion Removal step.

2.4 Calibrated IBR with Partial Depth

We start with the motivations and approach in Section 2.4.1. In Section 2.4.2, we present the depth interpolation technique. Finally, we show numerical experiments in Section 2.4.3.

2.4.1 Motivations and approach

The algorithm proposed in Section 2.3 relies on the availability of depth at all actual pixels. This section is inspired by the assumption that the depth provided by range finders [72, 71] is of lower resolution than that of intensity images. In this section, we consider that the depth maps are available at a downsampled version of the full depth maps used in Section 2.3. In Fig. 2.5, we highlight the depth-available pixels (one depth every 6×6, or about 3%, of intensity pixels).

There are two approaches to utilize the Propagation Algorithm framework: preprocessing (interpolate the depth for other actual pixels first) and postpro-

(a) Ground truth image at $u = 4$. (b) Rendered image at $u = 4$.

Figure 2.4: Rendered image of the Full Depth Propagation Algorithm. (a) The ground truth image taken at $u_v = 4$, and (b) the rendered image at $u_v = 4$.

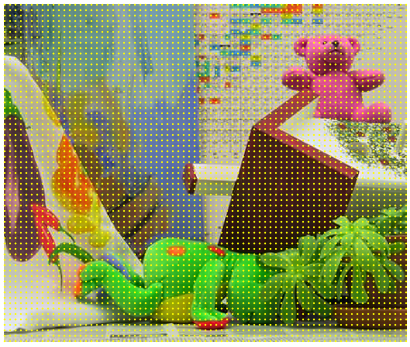


Figure 2.5: The scene at $u_0 = 2$ and depth-available pixels on a regular grid of dimension 6×6 (highlighted dots).

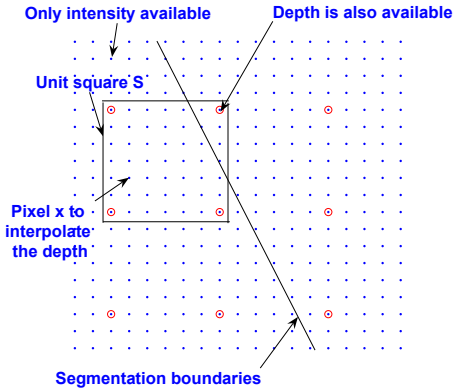


Figure 2.6: Segmentwise interpolation of depth. Dots are intensity pixels, and circles are depth-available pixels. We interpolate the depth for each unit square of depth-available pixels. Bilinear interpolation is used for square falling inside the same depth segment. Nearest neighbor interpolation is used for segment-crossing unit squares.

cessing (propagate depth-available pixels first). We adopt the preprocessing approach for two reasons. First, by doing this we do not discard any available information in our processing. Second, the depth is smoother and has fewer discontinuities to process than the intensity.

2.4.2 Segmentwise depth interpolation

A simple technique is to bilinearly interpolate [62] the depth maps to obtain depth for all actual pixels. Bilinear interpolation leads to blurred object boundaries. We propose a new technique, called *segmentwise depth interpolation*, to incorporate both intensity and depth information to interpolate the depth.

We start with a segmentation of the intensity images and of the low-resolution depth maps. We interpolate the depth for each unit square of depth-available pixels as illustrated in Fig. 2.6. For a unit square S, if all four vertices of S belong to the same depth segment, bilinear interpolation is used to interpolate the depth at pixels inside S. Otherwise, S lies on boundaries of different depth segments, and the depth of a pixel $\boldsymbol{x}$ is assigned by the depth of the closest vertex of S among those that share the same intensity segment with $\boldsymbol{x}$.

In practice, small intensity segments, usually in texture regions, may not contain any depth-available pixel. The pixels in these segments are marked and later filled using the depth of neighboring pixels, using morphological techniques such as dilation and erosion [73].

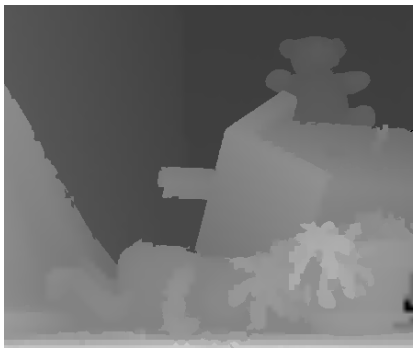


Figure 2.7: The reconstructed depth at actual camera $u_1 = 6$ using segmentwise depth interpolation technique. The depth at $u_0 = 2$ is obtained similarly.

Figure 2.8: Virtual image at $u_v = 4$ rendered using depth at about 3% of pixels. To be compared with the case of full depth in Fig. 2.4.

2.4.3 Experimental results

We use the same translational configuration as in Section 2.3.4, and the depth images are downsampled versions of the intensity images with rate $k = 6$ for both horizontal and vertical directions, as in Fig. 2.5. Moreover, the Mean Shift algorithm, proposed by Comeniciu et al. [74],[2] is used to segment images.

In Fig. 2.7, we show the depth at actual camera $u_1 = 6$ reconstructed using the proposed technique. The depth at actual camera $u_0 = 2$ is obtained similarly. These depth images are used as inputs of the Propagation Algorithm proposed in Section 2.3 to render the virtual image. The virtual image at $u_v = 4$ is rendered using the proposed technique in this section as shown in Fig. 2.8.

2.5 Uncalibrated IBR Using Projective Depth

We present in this section an algorithm to render the virtual image using feature correspondences of uncalibrated images. We present the motivations and approach in Section 2.5.1. In Section 2.5.2, we resolve occlusions directly in

[2]The software is available at http://www.caip.rutgers.edu/riul/research/code/EDISON/.

projective reconstructions. In Section 2.5.3, we approximate projective depths using a triangulation of feature points. We present numerical experiments in Section 2.5.4.

2.5.1 Motivations and approach

Motivated by the fact that for uncalibrated images, correspondences are usually detected at "good features" such as image corners [75], we consider to render virtual images using correspondences of uncalibrated images. We show that the same Propagation Algorithm framework is applicable for IBR using uncalibrated images.

In our algorithm, feature correspondences are used to reconstruct a projective reconstruction. The projective depth is interpolated at remaining actual pixels using an image-consistent triangulation [76]. The Propagation Algorithm framework can be used at this point, using the chirality parameter, to resolve the visibility directly in projective reconstructions.

2.5.2 Occlusion removal in projective reconstructions

It is a classical result [8, 36] in computer vision that from correspondences of uncalibrated images, we can reconstruct the 3D points up to a projective transformation, characterized by an *unknown* 4×4 matrix H. Under the transformation H, a 3D point $\widetilde{\boldsymbol{X}}$ and the projection matrix $\boldsymbol{\Pi}$ becomes

$$\widetilde{\boldsymbol{X}}_p = \boldsymbol{H}\widetilde{\boldsymbol{X}}, \quad \boldsymbol{\Pi}_p = \boldsymbol{\Pi}\boldsymbol{H}^{-1}. \tag{2.6}$$

Interestingly, in this projective reconstruction, the image $\boldsymbol{x}_p$ of point $\widetilde{\boldsymbol{X}}_p$ on camera $\boldsymbol{\Pi}_p$ coincides with $\boldsymbol{x}$:

$$\widetilde{\boldsymbol{x}}_p = \boldsymbol{\Pi}_p\widetilde{\boldsymbol{X}}_p = \boldsymbol{\Pi}\boldsymbol{H}^{-1}\boldsymbol{H}\boldsymbol{X} = \boldsymbol{\Pi}\boldsymbol{X} = \widetilde{\boldsymbol{x}}. \tag{2.7}$$

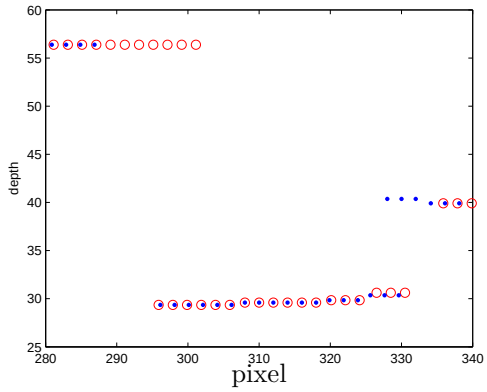
However, the projective depth d_p of $\widetilde{\boldsymbol{X}}_p$ with respect to the camera $\boldsymbol{\Pi}_p$ is different from d. In fact

$$d_p = d/t_p, \tag{2.8}$$

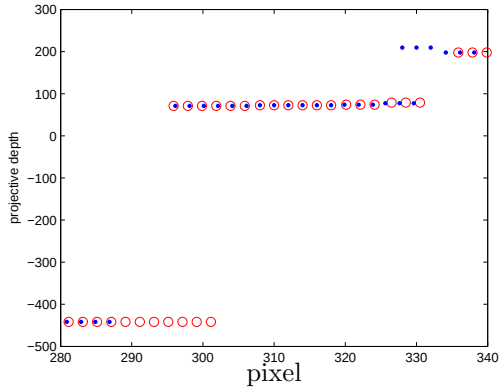
where t_p is the last coordinate of $\widetilde{\boldsymbol{X}}_p = \boldsymbol{H}\boldsymbol{X}$.

In Fig. 2.9(b), we illustrate how a projective transformation can deform the scene compared to the original scene in Fig. 2.9(a), causing difficulties in resolving occlusions. In Fig. 2.9(c), we show that a projective reconstruction still has the essential structure of the scene if we view it in the chirality domain. The chirality parameter χ is defined as the negative inverse of the projective depth:

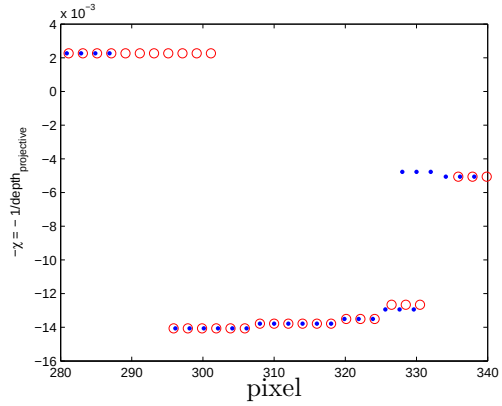
$$\chi = -\frac{1}{d_p}. \tag{2.9}$$



(a) Actual depth.



(b) Projective depth.



(c) Projective depth.

Figure 2.9: Chirality parameter χ preserves the scene structure better than depth under projective transformations. All x-axis is image plane. (a) Surface points seen from a camera; y-axis is depth. (b) The scene after being applied a projective transformation; y-axis is projective depth. (c) The projective scene with y-axis is the chirality parameter $\chi = -1/d$.

It is known [66] that χ can be used to resolve occlusions.

The use of the chirality parameter offers an intuitive way to resolve occlusions directly in projective reconstructions, compared to other existing occlusion removal techniques [39, 41].

2.5.3 Triangulation-based depth approximation

Having shown that occlusions can be resolved directly in projective reconstructions, in this section we interpolate the projective depth for other actual pixels before applying the Propagation Algorithm framework. To this end, we linearly interpolate the depth using some triangulation of image features. Existing triangulations, such as the Delaunay triangulation [65] and the *image-consistent triangulation* [76], can be used. In this section, the Delaunay triangulation is used for its simplicity and available implementations.

2.5.4 Experimental results

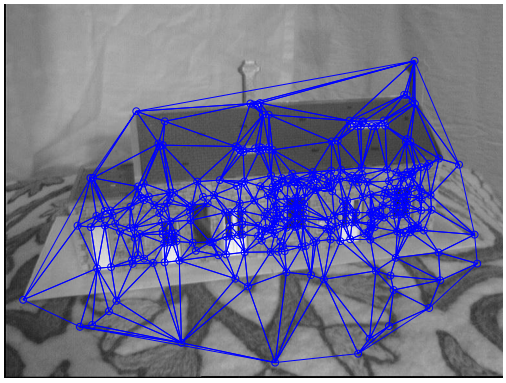
We use the multiple-view data set *Model House*.[3] Images at the actual cameras Π_2 and Π_4 are used to render the image at the virtual camera Π_3. In Fig. 2.10, we show the inputs of our algorithm: two actual images at Π_2, Π_4 and the feature correspondences (circle). We also plot the Delaunay triangulation of the feature points. The projective depth of other actual pixels will be interpolated in each triangle of the triangulation.

In Fig. 2.11(b), we show the rendered image at the virtual camera Π_3 compared to the ground truth (Fig. 2.11(a)). The virtual image is observed to have good rendering quality.

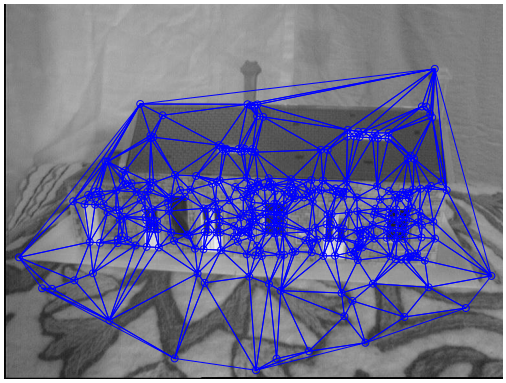
2.6 Conclusion and Discussion

We suggested that many existing IBR algorithms, using calibrated or uncalibrated images, can be analyzed using a unified framework, called the Propagation Algorithm. The framework is useful for improvements in the interpolation process using rigorous signal processing techniques. We applied the Propagation Algorithm to three IBR scenarios. For calibrated IBR with depth maps, we proposed an adaptive occlusion removal and interpolation of the virtual image using the intensity and depth information, following the linear inverse framework. In the case of calibrated IBR with partial depth, we segmentwise interpolated the depth for all actual pixels. For uncalibrated IBR using feature correspondences, we weakly calibrated the scene, interpolated projective depths using the Delaunay triangulation, and removed occlusions directly in projective reconstructions using the chirality parameter.

[3] The data set is available at http://www.robots.ox.ac.uk/~vgg/data1.html.



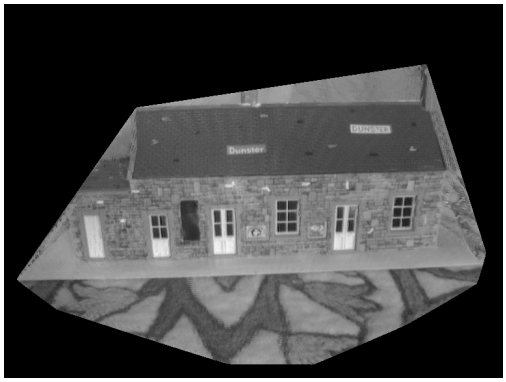
(a) Inputs at actual camera Π_2.



(b) Inputs at actual camera Π_4.

Figure 2.10: Inputs at actual camera Π_2, Π_4. For each camera we have as inputs the intensity image and the set of feature points (circles). The Delaunay triangulation of feature points is also plotted. We will interpolate the depth in each of these triangles.

(a) The ground truth image at camera Π_3.



(b) The rendered image at camera Π_3.

Figure 2.11: The rendered *Model House* image at Π_3. (a) The ground truth image at Π_3. (b) The rendered image at Π_3.

The approach proposed in this chapter also allows rigorous analysis of IBR algorithms using depth maps, as shown in our companion papers [77, 78]. As future work, we would like to address the implementation issues of the Propagation Algorithm.

CHAPTER 3

QUANTITATIVE ANALYSIS FOR IMAGE-BASED RENDERING: 2D UNOCCLUDED SCENES

3.1 Introduction

Image-based rendering (IBR) applications synthesize novel (or virtual) images, as taken by virtual cameras at arbitrary viewpoints, using a set of acquired images. With a range of applications, many algorithms have been proposed for IBR [4, 5, 80]. However, little research has addressed the fundamental issue of analyzing the effects of different factors on the rendering quality. These factors include the number of actual cameras and their characteristics (position and resolution), and the geometry and texture of the scene. The answer to this fundamental question is crucial for both theoretical and practical purposes; we cannot effectively control the rendering quality and the cost of IBR systems without accurate quantitative analysis of the rendering quality. In fact, many IBR systems have to rely on oversampling to counter undesirable aliasing effects.

In an early approach, McMillan and Bishop [40] formalized the IBR problem as a sampling and interpolation problem of the plenoptic function. The plenoptic function was defined by Adelson and Bergen [24] to characterize the radiant energy of the scene at all positions and directions. Chai et al. [49] analyzed the spectral support of the plenoptic function for layered depth scenes and found that, under some assumptions, it was bounded only by the minimum and maximum of the depths, not by the number of layers in the scene. Another approach, proposed by Zhang and Chen [50], is to analyze the IBR representations using the surface light field.

In most existing analysis, the plenoptic function and the surface light field are assumed to be bandlimited. However, in general, the plenoptic function and surface light field are *not* bandlimited [50, 52]. In addition, frequency-based analysis often implies uniform sampling and sinc interpolation, a strict

[0]This chapter includes research conducted jointly with Prof. Minh Do [77, 79]. We thank Professors Narendra Ahuja, David Forsyth, Bruce Hajek, and Yi Ma (University of Illinois at Urbana-Champaign, USA) for valuable hints and criticisms.

assumption in IBR context. Furthermore, uniform sampling causes aliasing noise, resulting in objectionable visual artifacts on the rendered images [81].

In this chapter, we propose a new approach to quantitatively analyze the rendering quality for IBR algorithms using per-pixel depth. For simplicity of exposition, 2D unoccluded scenes are considered in this chapter; generalizations to 2D occluded scenes and 3D scenes are possible as shown in the next chapter. The case of 2D scenes is useful in itself when all the cameras lie in the same plane, or when actual images are rectified before the rendering process. In addition, we assume the ideal pinhole camera model. Although this assumption is somewhat strict, it allows us to derive concrete results. Finally, the analysis is proposed in the spatial domain; thus it can quantify the rendering quality in a local portion of interest of the scene.

The main contribution of the chapter is a new methodology that enables quantitative analysis of the rendering quality of several IBR algorithms using per-pixel depth. Whether the virtual image is interpolated in image-space or object-space, the rendering error can be bounded based on the sample values, sample positions and their errors (i.e., the sample errors and jitters). We name the proposed methodology the *error aggregation framework*, as it successfully aggregates different sources of error in the same framework. The proposed framework consists of the following innovative techniques:

1. *Nonuniform interpolation framework.* In Proposition 3.1, we show that interpolation using splines [18], commonly used in practice, has errors that can be bounded based on sample intervals, sample errors, and jitters.

2. *Properties of sample intervals.* We show in Proposition 3.2 and 3.4 that the set of available sample positions (provided by actual cameras) can be approximated as a generalized Poisson process. This approximation allows closed form formulae to compute the sums of powers of sample intervals, used to derive the error bounds.

3. *Bounds of sample jitters.* We derive in Proposition 3.3 a bound for sample jitters based on the virtual camera, the scene geometry, and the error of depth estimates.

We apply the error aggregation framework to derive *bounds* for the mean absolute errors (MAE) of two classes of IBR algorithms: image-space interpolation (Theorem 3.1) and object-space interpolation (Theorem 3.2). The derived error bounds highlight the effects on the rendering quality of various factors including depth and intensity estimate errors, the scene geometry and texture, the number of cameras, their positions and resolution. Experiments for synthetic and actual scenes show that the theoretical bounds accurately characterize the rendering errors. Based on the proposed analysis, we discuss implications for IBR-related problems such as camera placement, budget allocation, and bit allocation.

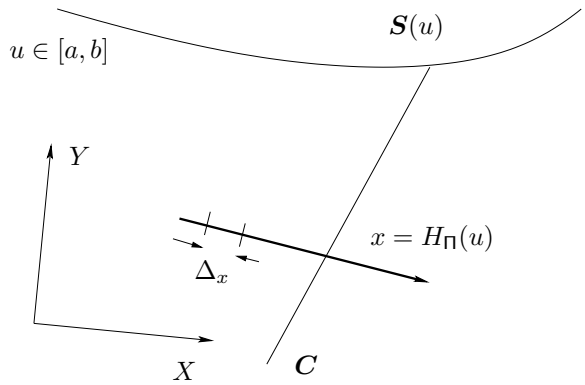


Figure 3.1: The 2D calibrated scene-camera model. The scene surface is modeled as a parameterized curve $\boldsymbol{S}(u)$ for $u \in [a, b] \subset \mathbb{R}$. The texture map $T(u)$ is "painted" on the surface. We assume pinhole camera model with calibrated projection matrix $\Pi = [\mathsf{R}, \boldsymbol{T}] \in \mathbb{R}^{2\times 3}$. The camera resolution is characterized by the pixel interval Δ_x on the image plane.

The remainder of the chapter is organized as follows. We present the problem setting in Section 3.2 and the methodology in Section 3.3. Then, we derive error bounds for an IBR algorithm using image-space interpolation (in Section 3.4) and an IBR algorithm using object-space interpolation (in Section 3.5). Validations of the bounds are shown in Section 3.6. In Section 3.7, limitations and implications of the analysis are discussed. Finally, conclusion is given in Section 3.8.

3.2 Problem Setting

We describe the scene model and the camera model in Sections 3.2.1 and 3.2.2, respectively. In Section 3.2.3, we briefly categorize IBR algorithms, in particular ones using per-pixel depth–the featured algorithm of this chapter. Finally, we formally introduce the problem definition.

3.2.1 The scene model

Consider the 2D unoccluded scene as in Fig. 3.1. Its surface is modeled as a 2D parameterized curve $\boldsymbol{S}(u) : [a, b] \to \mathbb{R}^2$, for some interval $[a, b] \subset \mathbb{R}$. Each $u \in [a, b]$ corresponds to a surface point $\boldsymbol{S}(u) = [X(u), Y(u)]^T$.

We denote $T(u) : [a, b] \to \mathbb{R}$ as the texture map "painted" on the surface $\boldsymbol{S}(u)$. Given a parametrization of the scene surface $\boldsymbol{S}(u)$, the texture map $T(u)$ is independent of the cameras and the scene geometry $\boldsymbol{S}(u)$.

In this chapter, we assume that both $\boldsymbol{S}(u)$ and $T(u)$ are twice continuously differentiable. We assume further that the surface is Lambertian [67], that is the images of the same surface point at different cameras have the same intensity.

3.2.2 The camera model

The scene-to-image mapping. A 2D *calibrated* pinhole camera is characterized by the projection matrix $\Pi \in \mathbb{R}^{2\times 3}$. For each surface point $\boldsymbol{S} = [X, Y]^T$ in the scene, let $\widetilde{\boldsymbol{S}} = [X, Y, 1]^T$ be the homogeneous coordinate [67] of $\boldsymbol{S}$. The projection matrix $\Pi = [\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T]^T$ maps surface points $\boldsymbol{S} = [X, Y]^T$ into image point x using the projection equation

$$d \cdot [x, 1]^T = \Pi \cdot [X, Y, 1]^T, \tag{3.1}$$

where $d = \boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}$ is the depth of a surface point $\widetilde{\boldsymbol{S}}$ relative to the camera Π. In this chapter, we use Π to refer to the camera.

Equation (3.1) implies a mapping from a surface point $\boldsymbol{S}(u)$, $u \in [a, b]$, to its image point x on the camera Π as

$$x = \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}(u)}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}(u)} \stackrel{\text{def}}{=} H_\Pi(u). \tag{3.2}$$

We name $H_\Pi(u)$ the *scene-to-image mapping*. For unoccluded scenes, the mapping $H_\Pi(u)$ is monotonic in $[a, b]$. Other properties of $H_\Pi(u)$ are shown in Appendix A.1.

Image formation process. On the image plane of a camera Π, the *image light field* $f_\Pi(x)$ at image point x characterizes the brightness $T(u)$ of the surface point $\boldsymbol{S}(u)$ having image at $\boldsymbol{x}$. In other words, function $f_\Pi(x)$ is a perspectively corrected version of the texture map $T(u)$:

$$f_\Pi(x) = T(H_\Pi^{-1}(x)). \tag{3.3}$$

Let Δ_x be the pixel interval in the image plane, or the *resolution*, of a camera Π. The intensity $I_\Pi[n]$ at n-th pixel is the value of the convolution between $f_\Pi(x)$ and a sampling kernel $\varphi(x)$ evaluated at the image position $x_n = n\Delta_x$:

$$I_\Pi[n] = (f_\Pi * \varphi)(x_n) = \int_{H_\Pi(a)}^{H_\Pi(b)} f_\Pi(x)\varphi(x_n - x)dx. \tag{3.4}$$

In this chapter, we assume the Dirac delta function as the sampling kernel, i.e., $\varphi(x) = \delta(x)$. In other words:

$$I_\Pi[n] = f_\Pi(n\Delta_x). \tag{3.5}$$

Intensity and depth estimate error. In practice, the depth can be obtained using the range finders [72, 71] or using structure-from-motion techniques [8, 82]. If Π is known, it is easy to convert from x and d to $[X, Y]^T$, and vice-versa, in the registration process using Equation (3.1). Hence we assume that a set of surface points $[X, Y]^T$ are available at the actual cameras. Due to depth estimation errors, the surface points are registered as $[X_e, Y_e]^T$ instead of

their actual value $[X, Y]^T$. The magnitude of the error $\boldsymbol{\varepsilon}_D = [X_e - X, Y_e - Y]^T$ is supposedly bounded by some bound $E_D > 0$.

The texture is also subject to errors. We assume that in the formation of actual pixels, noisy estimates $T_e(u)$ of $T(u)$ are registered. Again, the error $\varepsilon_T = T_e(u) - T(u)$ is supposedly bounded by some bound $E_T > 0$. In summary:

$$\|\boldsymbol{\varepsilon}_D\|_2 \leq E_D, \qquad |\varepsilon_T| \leq E_T. \tag{3.6}$$

3.2.3 IBR algorithms and problem statement

IBR algorithms. Many IBR algorithms have been proposed [4, 5, 80]. Shum et al. [4] categorized IBR algorithms in a continuum based on the levels of prior knowledge of the scene geometry. This chapter focuses on IBR algorithms using per-pixel depth. Other IBR algorithms, using implicit or no geometry, are not addressed in this chapter.

Among IBR algorithms using per-pixel depth, we focus further on two main interpolation methods: image-space interpolation [41, 46, 62, 64] and object-space interpolation [42, 43, 45]. We remark that this difference is only conceptual. In practice, the later methods also have efficient implementations in image-space using perspective correct methods [44].

Problem statement. Consider IBR texture mapping algorithms using explicit depth maps to render the image at virtual camera Π_v from images of N actual cameras $\{\Pi_n\}_{n=1}^N$. We want to quantify the effects on the rendering quality of the matrices $\{\Pi_n\}_{n=1}^N$ and Π_v, the resolution Δ_x, the depth and intensity error bounds E_D and E_T, the texture map $T(u)$, and the surface geometry $\boldsymbol{S}(u)$.

3.3 Methodology

We assume that piecewise linear interpolation, widely used in practice thanks to its ease of use and decent interpolation quality, is used in the rendering of the virtual image. Presenting the results for linear interpolation also helps IBR practitioners find this chapter directly useful. Similar analysis applies for interpolation techniques using higher order splines [18, 25].

The linear interpolation $\widehat{f}(x)$ of $f(x)$ in the interval $[x_1, x_2]$, see Fig. 3.2, is defined as

$$\widehat{f}(x) \stackrel{\text{def}}{=} \frac{x_2 - x}{x_2 - x_1} \cdot [f(x_1 + \mu_1) + \varepsilon_1] + \frac{x - x_1}{x_2 - x_1} \cdot [f(x_2 + \mu_2) + \varepsilon_2], \tag{3.7}$$

where μ_1, μ_2 are sample jitters and $\varepsilon_1, \varepsilon_2$ are sample errors. The $\mathcal{L}_\infty$ norm of a function $g(x)$ is defined as

$$\|g\|_\infty = \sup_x \{g(x)\}. \tag{3.8}$$

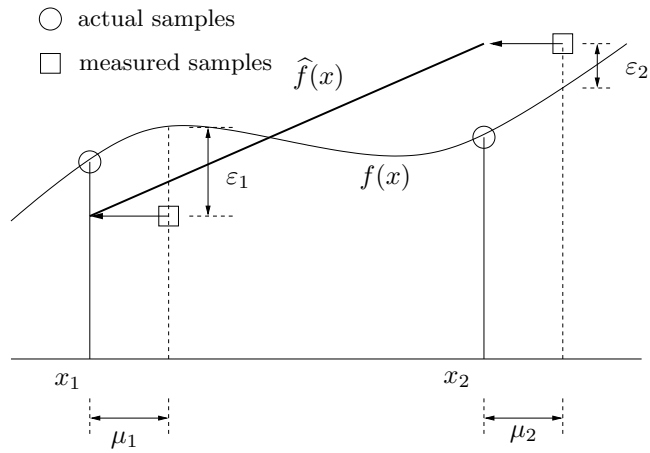


Figure 3.2: Linear interpolation. The interpolation error can be bounded using sample errors $\varepsilon_1, \varepsilon_2$, sample positions x_1, x_2, and their jitters μ_1, μ_2.

Proposition 3.1 *Consider a function $f(x)$ that is twice continuously differentiable. The linear interpolation $\widehat{f}(x)$ given in* (3.7) *has the error bounded by*

$$\begin{aligned} |\widehat{f}(x) - f(x)| \quad \leq \quad & \frac{1}{8}(x_2 - x_1)^2 \cdot \|f''\|_\infty + \max\{|\varepsilon_1|, |\varepsilon_2|\} \\ & + \max\{|\mu_1|, |\mu_2|\} \cdot \|f'\|_\infty. \end{aligned} \tag{3.9}$$

Proof 3.1 *Using the Taylor expansion, there exists $\xi_i \in [x_i, x_i + \mu_i]$ such that $f(x_i + \mu_i) - f(x_i) = \mu_i f'(\xi_i)$, for $i = 1, 2$. Hence we can bound the error caused by the sample errors and jitters, for $i = 1, 2$, as*

$$|f(x_i + \mu_i) + \varepsilon_i - f(x_i)| \leq |\varepsilon_i| + |\mu_i| \cdot \|f'\|_\infty. \tag{3.10}$$

In addition, let the linear interpolation using true sample values at x_1, x_2 be

$$\widetilde{f}(x) = \frac{x_2 - x}{x_2 - x_1} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2).$$

It can be shown [79] that

$$|f(x) - \widetilde{f}(x)| \quad \leq \quad \frac{1}{2}(x - x_1)(x_2 - x) \cdot \|f''\|_\infty \tag{3.11}$$

$$\leq \quad \frac{1}{8}(x_2 - x_1)^2 \cdot \|f''\|_\infty. \tag{3.12}$$

From (3.10) *and* (3.12) *we indeed verify* (3.9).

Remark 3.1 *Proposition 3.1 can be considered as a local error analysis, providing a bound for the interpolation error in individual intervals. The error bound in* (3.9) *is a summation of three terms. Apart from intrinsic properties of the function $f(x)$, the first term depends on the sample intervals $(x_2 - x_1)$, the*

second term depends on the sample errors $\varepsilon_1, \varepsilon_2$, and the third term is related to the jitters μ_2, μ_2. Note that the bound is tight in the sense that equality does happen, for example, for linear functions $f(x)$.

Remark 3.2 *Generalization of Proposition 3.1 is possible for interpolation using splines of higher orders [18]. In these cases, only the first term in* (3.9) *will change to the product of higher powers of the sample interval $|x_2 - x_1|$, the L_∞ norm of higher order derivative of $f(x)$, and a constant.*

In the next two sections, we analyze the sample intervals and jitters in the context of IBR. The analysis is then used to derive error bounds of the virtual images. Note that the analysis is different for both cases since the interpolation process is conducted in different spaces.

3.4 Analysis for an IBR Algorithm Using Image-Space Interpolation

We derive the error bound for the Propagation Algorithm [62] as an IBR algorithm using image-space interpolation. The analysis is applicable to other IBR algorithms [41, 46, 64] under some simplistic assumptions.

We start by giving a brief description of the Propagation Algorithm in Section 3.4.1. In Section 3.4.2, properties of sample intervals at the virtual image plane are derived. We analyze the jitters caused by depth estimate errors in Section 3.4.3. In Section 3.4.4, we derive a bound for the rendering error. Finally, discussions will be given in Section 3.4.5.

3.4.1 Rendering using the Propagation Algorithm

The Propagation Algorithm consists of three main steps as follows:

1. Information Propagation. All the intensity information available at the actual image planes is propagated to the virtual image plane. This step is feasible since the depth information is available.

Consider N actual cameras $\{\Pi_i\}_{i=1}^N$ and a virtual camera Π_v. We denote $\{x_{i,n}\}$ the set of actual pixels of Π_i and $\{u_{i,n}\}$ are such that $x_{i,n}$ is the image of 2D surface point $\boldsymbol{S}(u_{i,n})$ (see Fig. 3.3). Let $y_{i,n}$ be the image of $\boldsymbol{S}(u_{i,n})$ at the virtual camera Π_v. These points $\{y_{i,n}\}$ will serve as samples in the interpolation process performed at virtual image plane.

2. Occlusion Removal. All the points in whose neighborhood there is another point with sufficiently smaller depth are removed; these points are likely occluded at the virtual camera. This step is crucial when we consider occluded scenes. However, this step is irrelevant in this chapter because the scene is supposed to be free of occlusions.

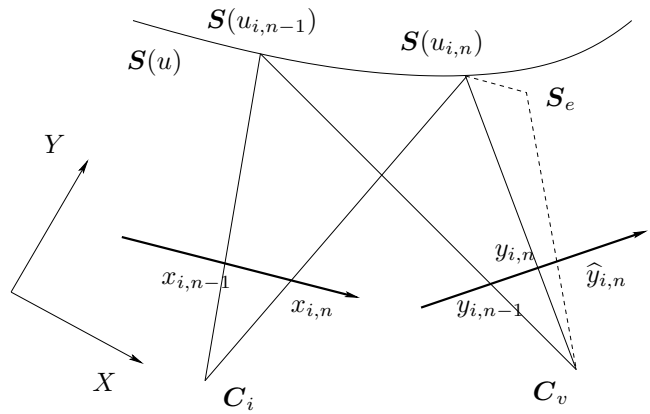


Figure 3.3: Sample intervals and jitters at virtual camera Π_v. Samples $\{y_{i,n}\}$ in virtual image plane are propagated from actual pixels $\{x_{i,n}\}$. The jitter $\mu = \widehat{y}_{i,n} - y_{i,n}$ is caused by a noisy estimate $\boldsymbol{S}_e$ of surface point $\boldsymbol{S}(u_n)$.

3. Intensity Interpolation. The virtual image is interpolated using the remaining samples. We suppose that piecewise linear interpolation is used.

For each actual camera Π_i, for $i = 1, \ldots, N$, let $\mathcal{Y}_i = \{y_{i,n}\}$ be the set of points on the virtual image plane propagated from $\{x_{i,n}\}$. Let the union of $\{\mathcal{Y}_i\}_{i=1}^N$ be

$$\mathcal{Y} = \bigcup_{i=1}^{N} \mathcal{Y}_i = \{y_m\}_{m=1}^{N_{\mathcal{Y}}}, \tag{3.13}$$

ordered so that $y_m \leq y_{m+1}$. Hence, $\mathcal{Y}$ contains all the *actual samples* propagated from the actual cameras.

The sources of rendering errors come from the intensity errors and jitters at $\{y_m\}$, in addition to the interpolation technique in use. We address these issues in the following sections.

3.4.2 Properties of sample intervals

In this section, we want to investigate the properties of the sample intervals $(y_{m+1} - y_m)$. As we see later, we are most interested in the summation $\sum (y_{m+1} - y_m)^k$ for $k \in \mathbb{N}$.

We analyze the sample intervals $(y_{m+1} - y_m)$ by considering each individual set $\mathcal{Y}_i$ as a point process. The set $\mathcal{Y}$ can be regarded as the union of point processes $\{\mathcal{Y}_i\}_{i=1}^N$. It is known that if the component processes $\{\mathcal{Y}_i\}_{i=1}^N$ have *identically distributed* intervals, their superposition can be approximated as a Poisson process in the distribution sense [83, 84, 85, 86].

Lemma 3.1 *In each infinitesimal* $[x, x + dx]$*, the point process* $\mathcal{Y}$*, defined as in* (3.13)*, can be approximated as a Poisson process with density*

$$\lambda_x(x) = \frac{1}{\Delta_x H_v'(u)} \cdot \sum_{i=1}^{N} H_i'(u), \tag{3.14}$$

where $u = H_v^{-1}(x)$ is such that point x is the image of surface point $\boldsymbol{S}(u)$ at the virtual camera Π_v.

Proof 3.2 *Consider an infinitesimal interval $[x, x + dx]$ on the virtual image plane. In this interval, we can assume that $H_i'(u)$ is constant; thus, sample intervals $(y_{i,n+1} - y_{i,n})$ can be considered identically distributed. Hence, as demonstrated in [84], locally, the set of sample points $\mathcal{Y}$ can be approximated as a Poisson process.*

Let $[u, u+du]$ be the portion of the scene whose image is the interval $[x, x+dx]$ at the virtual image plane. Hence $u = H_v^{-1}(x)$ and

$$dx = H_v(u + du) - H_v(u) \approx H_v'(u)du.$$

For each actual camera Π_i, for $i = 1, \ldots, N$, the average number of pixels that are images of $\boldsymbol{S}(\tau)$, for $\tau \in [u, u+du]$, is $H_i'(u)du/\Delta_x$. The average number of points in $\mathcal{Y}$ hence can be computed as

$$E[N_p] = \frac{du}{\Delta_x} \sum_{i=1}^{N} H_i'(u).$$

The density $\lambda_v(x)$ hence can be obtained as

$$\lambda_v(x) = \frac{E[N_p]}{dx} = \frac{1}{\Delta_x H_v'(u)} \cdot \sum_{i=1}^{N} H_i'(u).$$

If the density $\lambda_x(x)$ is a constant over the whole interval $[a, b]$, the set of points $\mathcal{Y} = \{y_m\}_{m=1}^{N_\mathcal{Y}}$ can be approximated as a Poisson process. However, since $\lambda_x(x)$ changes over $[a, b]$, $\mathcal{Y}$ is approximated as a *generalized*, or *inhomogeneous*, Poisson process [87, 88]. We use this key result to derive properties of sample intervals.

Proposition 3.2 *The point process $\mathcal{Y}$ can be approximated as a generalized Poisson process with density function $\lambda_x(x)$ satisfying* (3.14) *for $x \in [H_v(a), H_v(b)]$. The sum of powers of the sample intervals can be computed as*

$$\sum_{n=1}^{N_\mathcal{Y}-1} (y_{m+1} - y_m)^k \approx k! Y_k \Delta_x^{k-1}, \tag{3.15}$$

where

$$Y_k = \int_a^b \left(\sum_{i=1}^{N} H_i'(u) \right)^{1-k} (H_v'(u))^k \, du. \tag{3.16}$$

Proof 3.3 *Using the result of Lemma 3.1, the point process $\mathcal{Y}$ can be approximated as a Poisson process of density $\lambda_x(x)$ in each infinitesimal interval $[x, x + dx]$. As a consequence, the average number of points $y_m \in \mathcal{Y}$ falling*

into $[x, x+dx]$ *is* $\lambda_x(x)dx$*, and the expectation of intervals* $E[(y_{m+1}-y_m)^k]$ *inside* $[x, x+dx]$ *is equal to* $k!/\lambda_x(x)^k$*. Hence:*

$$\sum_{n=1}^{N_{\mathcal{Y}}-1} (y_{m+1}-y_m)^k \approx \int_{H_v(a)}^{H_v(b)} \frac{k!}{\lambda_x(x)^k} \lambda_x(x) dx.$$

By changing the variable under the integral from x *to* u*, using* $dx = H_v'(u)du$*, we indeed obtain* (3.15)*.*

Note that Y_k, called the image-space multiple-view term of order k, for $k \in \mathbb{N}$, depends only on the relative positions of the (actual and virtual) cameras and the scene. In unoccluded scenes, the derivative $H_i'(u)$ has positive infimum and finite supremum. As a consequence:

$$0 \; < \; \lim_{N\to\infty} Y_k N^{k-1} \; < \; \infty. \tag{3.17}$$

We denote $Y_k = \mathcal{O}(N^{1-k})$ in the remainder of the chapter. In particular, $Y_1 = H_v(b) - H_v(a)$ is a constant equal to the length of the image of $\boldsymbol{S}(u)$, for $u \in [a, b]$, in the virtual image plane. In practice, computing Y_k requires the knowledge of $H_v'(u)$ and $\{H_i'(u)\}_{i=1}^N$. The derivatives $H_\Pi'(u)$ have a simple geometrical interpretation as given in Appendix A.1.

3.4.3 Bound for sample jitters

Another source of IBR error is the jitters caused by noisy depth estimates. Let $\mathsf{S} = [X, Y]^T$ be a surface point, and y be the image of S at the virtual camera Π_v. We denote $\mathsf{S}_e = [X_e, Y_e]^T$ a noisy estimate of S with reconstruction error $\boldsymbol{\varepsilon}_D = \mathsf{S}_e - \mathsf{S}$, and $\widehat{y}$ to be the image of S_e at Π_v (see Fig. 3.3). In this section, we derive a bound for the sample jitters $\mu = \widehat{y} - y$.

Proposition 3.3 *The jitter* $\mu = \widehat{y} - y$ *at virtual camera* Π_v *caused by the depth estimate error* $\boldsymbol{\varepsilon}_D$ *is bounded by*

$$|\mu| \leq E_D B_v. \tag{3.18}$$

In the above inequality, B_v *is determined as follows using the center* $\boldsymbol{C}_v$ *of the virtual camera* Π_v*:*

$$B_v \quad = \quad \sup_{u\in[a,b]} \quad \left\{ \frac{\|\boldsymbol{C}_v - \boldsymbol{S}(u)\|_2}{d(u)^2} \right\}. \tag{3.19}$$

Proof 3.4 *Let* $\boldsymbol{\varepsilon} = [\varepsilon_X, \varepsilon_Y, 0]^T$ *and* $\boldsymbol{p} = [p_X, p_Y, 0]^T = \widetilde{\boldsymbol{S}}(u) - \widetilde{\boldsymbol{C}}_v$*. We can*

easily verify

$$\begin{aligned}\widetilde{\boldsymbol{S}}_e &= \widetilde{\boldsymbol{C}}_v + \boldsymbol{p} + \boldsymbol{\varepsilon},\\ \widetilde{\boldsymbol{S}} &= \widetilde{\boldsymbol{C}}_v + \boldsymbol{p},\\ \boldsymbol{\pi}_i^T \widetilde{\boldsymbol{C}}_v &= 0, \quad i = 1, 2.\end{aligned}$$

Denote $\Pi_v = [\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T]^T$. *Using the above equalities and simple manipulations we get*

$$\mu = \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}_e}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}_e} - \frac{\boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}}}{\boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}}} = \frac{\boldsymbol{p}^T(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)\boldsymbol{\varepsilon}}{(\pi_2^T \widetilde{\boldsymbol{S}}_e)\cdot(\pi_2^T \widetilde{\boldsymbol{S}})}.$$

Note that both $\boldsymbol{\varepsilon}, \boldsymbol{p}$ *have the third coordinate equal to* 0. *Hence, only the upper-left* 2×2 *block of matrix* $(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)$ *needs to be investigated. If we let* R_v *be the rotation matrix of* Π_v, *it can be verified that the maximum singular value of the upper-left* 2×2 *block of matrix* $(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)$ *is in fact* $\det(\mathsf{R}_v) = 1$. *Hence*

$$|\mu| \approx \frac{|\boldsymbol{p}^T(\boldsymbol{\pi}_2\boldsymbol{\pi}_1^T - \boldsymbol{\pi}_1\boldsymbol{\pi}_2^T)\boldsymbol{\varepsilon}|}{d(u)^2} \leq \frac{\|\boldsymbol{p}\|_2 \cdot \|\boldsymbol{\varepsilon}\|_2}{d(u)^2} \leq B_v E_D.$$

The bound $E_D B_v$ depends on the depth estimate error E_D and the relative position between the virtual camera and the scene defined by B_v.

3.4.4 Error analysis

Combining the results of Proposition 3.2 and 3.3, we derive in this section an error bound for the *mean absolute error* (MAE) of the virtual image. Let $e(x) = \widehat{f}_v(x) - f_v(x)$ be the interpolation error and N_{pixel} be the number of virtual pixels being images of surface points $\boldsymbol{S}(u)$ for $u \in [a, b]$. The mean absolute error MAE_{IM} is defined as

$$\text{MAE}_{\text{IM}} = \frac{1}{N_{\text{pixel}}} \sum_{n=1}^{N_{\text{pixel}}} |e(n\Delta_x)|. \tag{3.20}$$

Theorem 3.1 *The mean absolute error* MAE_{IM} *of the virtual image is bounded by*

$$\text{MAE}_{\text{IM}} \leq \frac{3Y_3}{4Y_1}\Delta_x^2 \|f_v''\|_\infty + E_T + E_D B_v \|f_v'\|_\infty, \tag{3.21}$$

where Y_1, Y_3 *are defined as in* (3.16), B_v *is as in* (3.19), *and* E_D, E_T *are as in* (3.6).

Proof 3.5 *Note that the* n*-th virtual pixel has position* $x_n = n\Delta_x$ *in the virtual*

image plane, hence $\mathrm{MAE}_{\mathrm{TM}}$ *can be approximated as*

$$\mathrm{MAE}_{\mathrm{IM}} \approx \frac{1}{H_v(b) - H_v(a)} \int_{H_v(a)}^{H_v(b)} |e(x)| dx. \tag{3.22}$$

We break down the integral above into intervals $[y_m, y_{m+1})$ *and apply Proposition 3.1 to each interval to get*

$$\begin{aligned}
\int_{H_v(a)}^{H_v(b)} |e(x)| dx &= \sum_{n=1}^{N_{\mathcal{Y}}-1} \int_{y_m}^{y_{m+1}} |e(x)| dx \\
&\leq \sum_{n=1}^{N_{\mathcal{Y}}-1} \Big[\frac{1}{8} (y_{m+1} - y_m)^3 \|f_v''\|_\infty \\
&\qquad + (y_{m+1} - y_m) \big(E_T + E_D B_v \|f_v'\|_\infty \big) \Big] \\
&= \frac{3}{4} Y_3 \Delta_x^2 \|f_v''\|_\infty + Y_1 \left(E_T + E_D B_v \|f_v'\|_\infty \right).
\end{aligned}$$

In the last inequality, $\sum (y_{m+1} - y_m)^k$ *are replaced by* $k! Y_k \Delta_x^{k-1}$ *using* (3.16)*. Substituting the above bound of the integral* $\int |e(x)| dx$ *into* (3.22)*, we indeed obtain* (3.21)*.*

The result of Theorem 3.1 can be extended to other error measures (e.g., mean square error) and other interpolation techniques (e.g., using higher order splines).

3.4.5 Discussion

The error bound in (3.21) consists of three terms. In the first term, $\|f_v''\|_\infty$ and Y_1 depend only on the virtual camera position, where as Y_3 depends on the camera configuration and the scene. We can consider Y_3 as the spatial information contributed by the actual cameras. Overall, the first term characterizes the gain of using multiple actual cameras.

To decrease the first term in an IBR setting, we can either use actual cameras of finer resolution Δ_x or increase the number of actual cameras N. Theorem 3.1 indicates that both methods yield comparable effects on the rendering quality. Moreover, the error bound decays as $\mathcal{O}(\lambda^{-2})$, where

$$\lambda = N / \Delta_x \tag{3.23}$$

can be interpreted as the local density of actual samples.

The second term, E_T, characterizes the noise level at actual cameras. This can be considered as the limit of the rendering quality imposed by the quality of actual images.

The third term contains the precision E_D of range finders and two factors $\|f_v'\|_\infty, B_v$ that depend on the relative position between the virtual camera and

the scene. Only E_D, among three factors, can be reduced by using better range finders. The remaining two factors, $\|f_v'\|_\infty$ and B_v, are fixed once the virtual camera is specified.

3.5 Analysis for an IBR Algorithm Using Object-Space Interpolation

We analyze the rendering quality of a basic IBR algorithm using object-space interpolation. After a brief description of the algorithm in Section 3.5.1, we investigate the sample intervals and jitters in Sections 3.5.2 and 3.5.3, respectively. In Section 3.5.4, we derive an error bound for the virtual image. Finally, a discussion is given in Section 3.5.5.

3.5.1 A basic algorithm

Interestingly, IBR algorithms using object-space interpolation can be implemented in image-space using perspective correct interpolation [44], although the texture is conceptually interpolated in object-space. However, to follow the proposed methodology in Section 3.3, we analyze the rendering quality in object-space. The analysis is shown for a basic IBR algorithm consisting of three main steps: surface reconstruction, texture interpolation, and ray-scene intersection.

1. Surface reconstruction. The scene geometry is reconstructed using the depths available from actual cameras. We consider linear interpolation in this step.

Consider an actual camera Π_i and let $\{x_{i,n}\}$ be the positions of actual pixels on the actual image plane of Π_i. Suppose that $x_{i,n}$ is the image of 2D point $\boldsymbol{S}(u_{i,n})$ in the scene. Let $\mathcal{U}_i = \{u_{i,n}\}$ and let

$$\mathcal{U} = \bigcup_{i=1}^{N} \mathcal{U}_i = \{u_m\}, \tag{3.24}$$

ordered so that $u_m \leq u_{m+1}$, be the union of visible points on the surface of the scene. In this step, the surface geometry $\boldsymbol{S}(u)$ is reconstructed using samples $\boldsymbol{S}(u_m)$, for $u_m \in \mathcal{U}$.

2. Texture interpolation. The texture map $T(u)$ is linearly interpolated on the reconstructed surface using the intensity of actual pixels.

Let $\widehat{\boldsymbol{S}}(u)$, for $u \in [a, b]$, be the linear approximation of the scene $\boldsymbol{S}(u)$ at the surface reconstruction process. The set of samples are visible points

$$\{\widehat{\boldsymbol{S}}(u_m) = \boldsymbol{S}(u_m) + \boldsymbol{\varepsilon}_m, \quad u_m \in \mathcal{U}\}.$$

3. Ray-scene intersection. For each virtual pixel y, draw a ray connecting

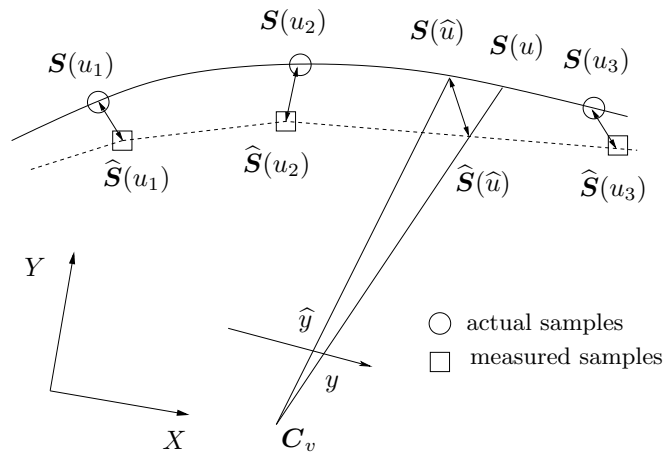


Figure 3.4: The reconstructed scene $\widehat{\boldsymbol{S}}(u)$ using piecewise linear interpolation. The intensity at virtual pixel y is the interpolated intensity at an approximated surface point $\widehat{\boldsymbol{S}}(\widehat{u})$ instead of the actual surface point $\boldsymbol{S}(u)$.

y and the camera center $\boldsymbol{C}_v$. Determine where this line intersects with the surface of the scene. The intensity of y will be the brightness of the intersection point.

In Fig. 3.4, for each point y in the image plane of virtual camera Π_v, let $\boldsymbol{S}(u)$ be the surface point whose image is y. Hence, $\boldsymbol{S}(u)$ is supposed to be the intersection of the scene surface and the ray connecting the virtual camera center $\boldsymbol{C}$ and the pixel position y. However, since only the approximated scene $\widehat{\boldsymbol{S}}(u)$ is available, the intersection is at $\widehat{\boldsymbol{S}}(\widehat{u})$ instead. Note that the parameter u is also jittered to $\widehat{u}$.

The primary sources of errors come from the interpolated texture $\widehat{T}(u)$ and the jitters $(u - \widehat{u})$ caused by the interpolated geometry $\widehat{\boldsymbol{S}}(u)$. Our approach in this section defers from the derivation in Section 3.4 due to two aspects. First, the surface reconstruction and texture interpolation are independent of the virtual camera. Second, the ray-scene intersection step results in jittered samples of the interpolated texture as the rendered image, whereas in Section 3.4, the virtual image is rendered using jittered samples of the texture function.

3.5.2 Properties of sample intervals

We first derive the properties of the sample intervals $(u_{m+1} - u_m)$ on the scene surface. Similarly to our derivation in Section 3.4.2, we assume that the intervals $(u_{i,n+1} - u_{i,n})$ are identically distributed in each infinitesimal interval. Hence, the union $\mathcal{U}$ can be approximated as a generalized Poisson process of some density function $\lambda_u(u)$. Similarly to Proposition 3.2, we can derive the following result.

Proposition 3.4 *The union $\mathcal{U}$ defined as in* (3.24) *can be approximated as a*

generalized Poisson process with density

$$\lambda_u(u) = \frac{1}{\Delta_x} \cdot \sum_{i=1}^{N} H_i'(u). \tag{3.25}$$

As a consequence, we have

$$\sum_{n=1}^{N_{\mathcal{U}}-1} (u_{m+1} - u_m)^k \approx k! U_k \Delta_x^{k-1}, \tag{3.26}$$

where

$$U_k = \int_a^b \left(\sum_{i=1}^{N} H_i'(u) \right)^{1-k} du. \tag{3.27}$$

Note that U_k, called object-space multiple-view terms of order $k \in \mathbb{N}$, depends only on the relative positions of the actual cameras and the scene. Unlike Y_k, terms U_k are independent of the virtual camera, since the interpolation uses only information provided by the actual cameras. Furthermore, it can be verified that U_k decays as $\mathcal{O}(N^{1-k})$ when N is large.

3.5.3 Bound for sample jitters

As shown in Fig. 3.4, the surface $\boldsymbol{S}(u)$ is approximated by $\widehat{\boldsymbol{S}}(u)$. Hence, the ray-scene intersection step results in the brightness at surface point $\widehat{\boldsymbol{S}}(\widehat{u})$ instead of $\boldsymbol{S}(u)$. Let y and $\widehat{y}$ be the images of $\boldsymbol{S}(u)$ and $\widehat{\boldsymbol{S}}(u)$. We first bound the depth interpolation errors $\|\boldsymbol{S}(u) - \widehat{\boldsymbol{S}}(u)\|_2$, and use this result to bound the jitters $(y - \widehat{y})$, similar to Section 3.4.3.

Lemma 3.2 *The Euclidean norm of the depth interpolated errors in an interval $[u_m, u_{m+1}]$ can be bounded as*

$$\|\boldsymbol{S}(\widehat{u}) - \widehat{\boldsymbol{S}}(u)\|_2 \leq \frac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} + E_D, \tag{3.28}$$

where

$$K_{\boldsymbol{S}} = \left(\|X''\|_\infty^2 + \|Y''\|_\infty^2 \right)^{1/2}. \tag{3.29}$$

Proof 3.6 *Let the depth estimate errors at surface points $\boldsymbol{S}(u_m)$ and $\boldsymbol{S}(u_{m+1})$ be $\boldsymbol{\varepsilon}_m = [\varepsilon_{X,m}, \varepsilon_{Y,m}]^T$ and $\boldsymbol{\varepsilon}_{m+1} = [\varepsilon_{X,m+1}, \varepsilon_{Y,m+1}]^T$. Denote $\gamma = (u - u_m)/(u_{m+1} - u_m) \in [0, 1]$. The interpolated depth $\widehat{\boldsymbol{S}}(u) = [\widehat{X}(u), \widehat{Y}(u)]^T$ is*

$$\widehat{\boldsymbol{S}}(u) = \gamma \cdot [\boldsymbol{S}(u_{m+1}) + \boldsymbol{\varepsilon}_{m+1}] + (1 - \gamma) \cdot [\boldsymbol{S}(u_m) + \boldsymbol{\varepsilon}_m].$$

Using techniques similar to those of Proposition 3.1, we can derive

$$\begin{aligned}|\widehat{X}(u) - X(u)| &\leq \frac{1}{8}(u_{m+1} - u_m)^2 \|X''\|_\infty \\ &\quad + \gamma \cdot |\varepsilon_{X,m+1}| + (1-\gamma) \cdot |\varepsilon_{X,m}|.\end{aligned}$$

Similar inequality can also be derived for $|\widehat{Y}(u) - Y(u)|$. *Using the generalized triangular inequality*[1] *we obtain*

$$\begin{aligned}\|\widehat{\boldsymbol{S}}(u) - \boldsymbol{S}(u)\|_2 &= \sqrt{|\widehat{X}(u) - X(u)|^2 + |\widehat{Y}(u) - Y(u)|^2} \\ &\leq \frac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} + \gamma \|\boldsymbol{\varepsilon}_{m+1}\|_2 + (1-\gamma)\|\boldsymbol{\varepsilon}_{m+1}\|_2 \\ &\leq \frac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} + E_D.\end{aligned}$$

In (3.29), $K_{\boldsymbol{S}}$ can be interpreted as the geometrical complexity of the scene. In particular, if the scene is piecewise linear, $K_{\boldsymbol{S}} = 0$. We can use the result of Lemma 3.2 to derive the following proposition.

Proposition 3.5 *Let* y *and* $\widehat{y}$ *be the image of* $\boldsymbol{S}(u)$ *and* $\widehat{\boldsymbol{S}}(u)$, *for* $\widehat{u} \in [u_m, u_{m+1}]$, *at the virtual camera* Π_v. *The jitter* $(y - \widehat{y})$ *is bounded by*

$$|y - \widehat{y}| \leq \frac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} B_v + E_D B_v, \tag{3.30}$$

where B_v *is as in* (3.19) *and* $K_{\boldsymbol{S}}$ *is defined as in* (3.29).

Proof 3.7 *Similar to* (3.19), *the jitter* $(y - \widehat{y})$ *can be bounded based on* $\|\boldsymbol{S}(u) - \widehat{\boldsymbol{S}}(u)\|_2$ *as*

$$\begin{aligned}|y - \widehat{y}| &\leq \|\boldsymbol{S}(\widehat{u}) - \widehat{\boldsymbol{S}}(u)\|_2 \cdot B_v \\ &\leq \frac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} B_v + E_D B_v.\end{aligned}$$

3.5.4 Error analysis

In practice, the texture map $T(u)$ is linearly interpolated in each interval $[u_m, u_{m+1})$ to get an approximation $\widehat{T}(u)$ for $u \in [a, b]$. We first derive the pointwise intensity interpolation error $\left(\widehat{T}(\widehat{u}) - T(u)\right)$ before combining them, in Theorem 3.2, to analyze the overall rendering quality.

Lemma 3.3 *The interpolation error* $\left(\widehat{T}(\widehat{u}) - T(u)\right)$ *at each point* $u \in [u_m, u_{m+1}]$ *is bounded by*

$$|\widehat{T}(\widehat{u}) - T(u)| \leq \frac{1}{8} K_1 (u_{m+1} - u_m)^2 + K_2, \tag{3.31}$$

[1] The generalized triangular inequality states that for any real number $\{a_i, b_i\}_{i=1,2,3}$, the following inequality holds:

$$\left((\sum_{i=1}^{3} a_i)^2 + (\sum_{i=1}^{3} b_i)^2 \right)^{1/2} \leq \sum_{i=1}^{3} \left(a_i^2 + b_i^2\right)^{1/2}.$$

where

$$K_1 = \|T''\|_\infty + B_v K_{\boldsymbol{S}} \|f_v'\|_\infty \tag{3.32}$$

$$K_2 = E_T + E_D B_v \|f_v'\|_\infty. \tag{3.33}$$

Proof 3.8 *Using* (3.9) *for the interval* $[u_m, u_{m+1})$*, the intensity error is bounded by*

$$|\widehat{T}(\widehat{u}) - T(\widehat{u})| \leq \frac{1}{8}(u_{m+1} - u_m)^2 \|T''\|_\infty + E_T. \tag{3.34}$$

Using the mean value theorem, there exists $\theta \in [y, \widehat{y}]$ *such that* $f_v(\widehat{y}) - f_v(y) = (\widehat{y} - y) f_v'(\theta)$*. Hence*

$$\begin{aligned} |T(\widehat{u}) - T(u)| &= |f_v(\widehat{y}) - f_v(y)| \leq |\widehat{y} - y| \cdot \|f_v'\|_\infty \\ &\leq \tfrac{1}{8}(u_{m+1} - u_m)^2 K_{\boldsymbol{S}} B_v \|f_v\|_\infty + E_D B_v \|f_v'\|_\infty. \end{aligned} \tag{3.35}$$

The last inequality used the result of Proposition 3.5. Finally, using (3.34) *and* (3.35) *we indeed obtain*

$$\begin{aligned} |\widehat{T}(\widehat{u}) - T(u)| &\leq |\widehat{T}(\widehat{u}) - T(\widehat{u})| + |T(\widehat{u}) - T(u)| \\ &\leq \frac{1}{8} K_1 (u_{m+1} - u_m)^2 + K_2. \end{aligned}$$

At this point, we are ready to bound the rendering error. Let $e(x) = \widehat{T}(\widehat{u}) - T(u)$ be the interpolation error and N_{pixel} be the number of virtual pixels being images of the scene $\boldsymbol{S}(u)$. The *mean absolute error* MAE_{OBJ} is defined as

$$\text{MAE}_{\text{OBJ}} = \frac{1}{N_{\text{pixel}}} \sum_{n=1}^{N_{\text{pixel}}} |e(n\Delta)|. \tag{3.36}$$

Theorem 3.2 *The mean absolute error* MAE_{OBJ} *of the virtual image is bounded by*

$$\text{MAE}_{\text{OBJ}} \leq \frac{3}{4} \cdot \frac{M_v K_1 U_3}{H_v(b) - H_v(a)} \Delta_x^2 + E_T + E_D B_v \|f_v'\|_\infty, \tag{3.37}$$

where U_3 *and* K_1 *are as in* (3.27), (3.32)*, and* M_v *is such that*

$$M_v = \max_{u \in [a,b]} \{H_v'(u)\}. \tag{3.38}$$

Proof 3.9 *Since* K_2*, defined as in* (3.33)*, is a constant, we can approximate* MAE_{OBJ} *as*

$$\text{MAE}_{\text{OBJ}} \approx K_2 + \tfrac{1}{\Delta_x N_{\text{pixel}}} \int_{H_v(a)}^{H_v(b)} (|e(x)| - K_2)\, dx \tag{3.39}$$

$$\leq K_2 + \tfrac{M_v}{\Delta_x N_{\text{pixel}}} \int_a^b \left(|\widehat{T}(\widehat{u}) - T(u)| - K_2 \right) du. \tag{3.40}$$

In the last inequality, we use the fact that $dx \leq M_v du$. The integral can be broken down into integrals in intervals $[u_m, u_{m+1})$. Using (3.31) *of Lemma 3.3 we get*

$$\begin{aligned}
&\int_a^b \left(|\widehat{T}(\widehat{u}) - T(u)| - K_2\right) du \\
&\qquad \leq \sum_{n=1}^{N_{\mathcal{U}}-1} \int_{u_m}^{u_{m+1}} \frac{1}{8} K_1 (u_{m+1} - u_m)^2 du \\
&\qquad \leq \frac{1}{8} K_1 \sum_{n=1}^{N_{\mathcal{U}}-1} (u_{m+1} - u_m)^3 \\
&\qquad \approx \frac{3}{4} K_1 U_3 \Delta_x^2.
\end{aligned}$$

Substituting the last inequality into (3.40)*, and replacing $\Delta_x N_{\text{pixel}} \approx H_v(b) - H_v(a)$ and K_1, K_2 as in* (3.32)*,* (3.33)*, we indeed obtain* (3.37)*.*

Again, we note that this result can be extended to other error measures (e.g., mean square error) and other interpolation techniques (e.g., using higher order splines).

3.5.5 Discussion

The error bound in (3.37) shares the last two terms with the bound in (3.21); their interpretation can be found in Section 3.4.5. In the first term of the bound, $K_1 M_v/(H_v(b) - H_v(a))$ is a constant depending only on the scene and the virtual camera. The factor $U_3 \Delta_x^2$, depending on the relative position of the scene and the actual cameras, decays as $\mathcal{O}(\lambda^{-2})$, where $\lambda = N/\Delta_x$ is the local density of actual samples.

The major difference of the bound in Theorem 3.2 compared to its counterpart of Theorem 3.1 resides in the multiple-view terms U_3 and Y_3. In fact, U_3 depends only on the positions of the actual cameras, whereas Y_3 also incorporates the virtual camera position. For planar sloping surfaces, comparison of the first terms in Theorems 3.1 and 3.2 may explain why interpolation using perspective correction [44] is necessary.

3.6 Validations

We show numerical experiments to validate the error bound of Theorem 3.1; validations for Theorem 3.2 are similar. The experiments use a synthetic scene in Section 3.6.1 and an actual scene in Section 3.6.2. Section 3.6.2 also serves as an example on estimating the bound in practice.

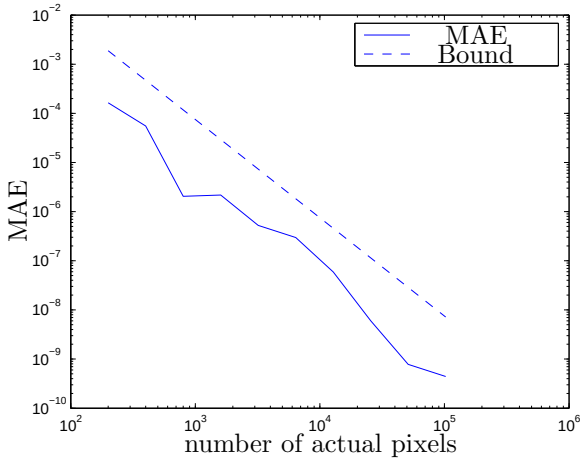


Figure 3.5: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the number of actual pixels in the loglog axis. Both the MAE and the theoretical bound decay with slope $s = -2$, consistent with the result of Theorem 3.1.

3.6.1 Synthetic scene

We adopt a simple translational camera configuration in our experiments. All the actual and virtual cameras are located in the X-axis, looking to the direction of the Y-axis. The 2D scene consists of a flat surface with distance $d = 10$ to the cameras and the texture $T(u) = \sin(u)$ painted on the surface. We use camera resolution $\Delta_x = 0.01$.

To validate the first term in (3.21), we set $E_T = E_D = 0$, and vary the number of actual cameras N. In Fig. 3.5, we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the number of actual pixels in the loglog axis. We observe that both the MAE and the theoretical bound decay with slope $s = -2$, consistent with the result of Theorem 3.1.

To validate the second term, we use $N = 10$ actual cameras and $E_D = 0$, and vary E_T. For each actual pixel, the intensity estimate error ε_T is randomly chosen in the interval $[-E_T, E_T]$ using the uniform distribution. In Fig. 3.6, we plot the mean absolute error MAE (solid) and the theoretical bound (dashed) against E_T. Note that the error bound is about two times the MAE, since the error bound is derived for the worst case, whereas the actual errors tend to follow the average case.

Finally, we validate the third term of (3.21) by using $N = 10$ actual cameras and setting $E_T = 0$. The depth estimate errors $\boldsymbol{\varepsilon}_D$ are randomly chosen in interval $[-E_D, E_D]$ using the uniform distribution. In Fig. 3.7, we plot the mean absolute error MAE and the theoretical bound against the depth estimate error bound E_D. We observe that the MAE indeed appears below the error bound and approximately linear to E_D.

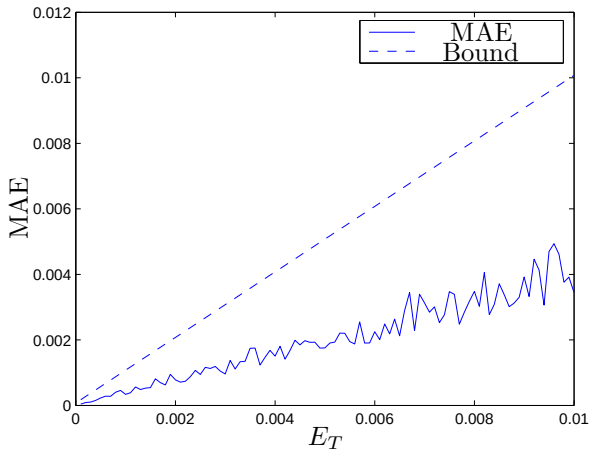


Figure 3.6: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the texture estimate error bound E_T.

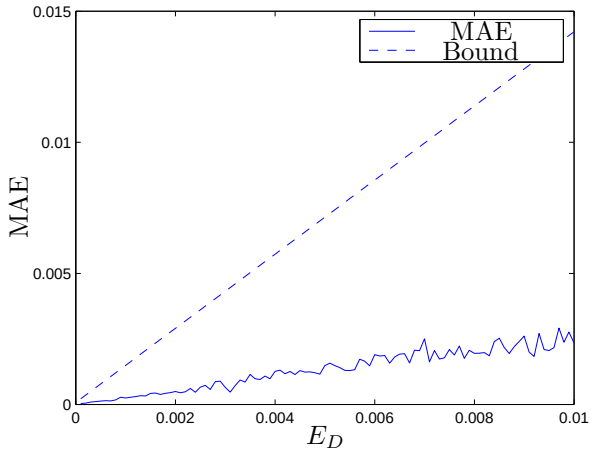


Figure 3.7: The mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D.

Figure 3.8: The scene's ground truth at the virtual camera $C_v = 4$.

3.6.2 Actual scene

A data set of a real scene is used to validate the error bound of Theorem 3.1.[2] This section can also be considered as an example of the computation of various factors in the error bound in practice.

All the actual and virtual cameras are located in the X-axis looking to the direction of the Y-axis. The virtual image at $C_v = 4$ is rendered, scanline by scanline, using the images and depth maps from actual cameras $C_1 = 2$ and $C_2 = 6$. The mean absolute error is computed using the available ground truth (see Fig. 3.8). The error bound of Theorem 3.1 is approximately computed, based on the data set, as follows.

Intensity estimate error bound E_T. Since the intensities are integers in the interval $[1, 255]$, we adopt

$$E_T = 1/2. \tag{3.41}$$

Bound of jitters $E_D B_v$. The data set provide the disparities, instead of depth, between two actual cameras $C_1 = 2$ and $C_2 = 6$. Hence, the bound of jitters $E_D B_v$ is directly computed instead of the depth estimate error bound E_D. Since the disparities between $C_1 = 2$ and $C_2 = 6$ are rounded to quarters of pixels, the disparities between C_1, C_2 and $C_v = 4$ are rounded to one eighth of a pixel. Hence, we adopt

$$E_D B_v = 1/16. \tag{3.42}$$

The resolution Δ_x. The images in have 450 columns assumedly spread over the image line of length 2. Hence

$$\Delta_x = 1/225. \tag{3.43}$$

[2]The data set is available at http://cat.middlebury.edu/stereo/newdata.html.

In practice, it turns out that the choice of the image line's length, and hence the resolution Δ_x, is not crucial. Its effect will be neutralized by the computation of $\|f_v'\|_\infty$ and $\|f_v''\|_\infty$.

Multiple-view terms Y_k. For a camera located at $X = X_0$ looking to the direction of the Y-axis, its projection matrix is

$$\Pi = \begin{bmatrix} 1 & 0 & -X_0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.44}$$

Suppose that the scene surface is a parameterized curve $[X(u), Y(u)]^T$, for $u \in [a, b]$. The corresponding scene-to-image mapping is

$$H_\Pi(u) = \frac{X(u) - X_0}{Y(u)}. \tag{3.45}$$

In this camera setting with two actual cameras at $C_1 = 2$ and $C_2 = 6$ and the virtual camera at $C_v = 4$, it can be verified that

$$H_1'(u) + H_2'(u) = 2H_v'(u), \quad u \in [a, b]. \tag{3.46}$$

As a consequence,

$$Y_3/Y_1 = 1/4. \tag{3.47}$$

Note that the length of the image line does not affect the ratio Y_3/Y_1, although it does change Y_k individually.

L_∞ norms $\|f_v'\|_\infty$ and $\|f_v''\|_\infty$. Since there exist noises and discontinuities in the virtual image f_v, a preprocessing step is necessary to estimate $\|f_v^{(k)}\|_\infty$. To limit the effect of noise, using a similar idea to edge detection techniques [89], the virtual image is first convolved with the derivative of order k of a Gaussian kernel

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(\frac{x^2}{2\sigma^2}\right). \tag{3.48}$$

In our experiment, we use the filter of length 10 pixels and $\sigma = 1$. Next, since the virtual image is discontinuous, we use the 95%-point value (instead of the maximum or 100%-point value) of the convolution as the L_∞ norm.

For each scanline, the error bounds of Equation (3.21) are computed using the procedure described above. In Fig. 3.9 we show the mean absolute error (solid) of the virtual image rendered using the Propagation Algorithm [62] compared to the estimated error bounds (dashed) for each scanline. Observe that the bound is tighter for scanlines with smoother intensity function $f_v(x)$.

3.7 Discussion and Implications

We discuss the case where actual cameras have different resolutions in Section 3.7.1. In Sections 3.7.2–3.7.4, implications of the proposed analysis on

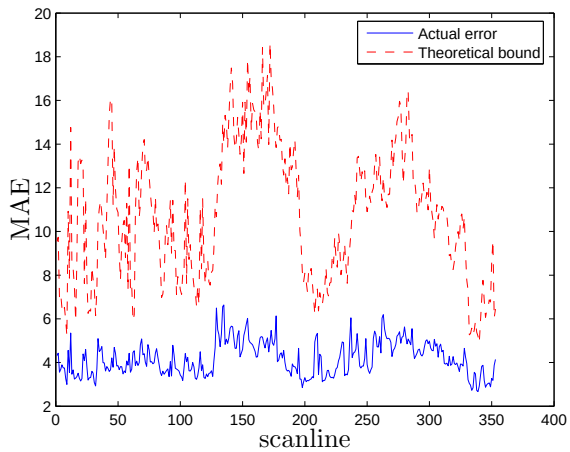


Figure 3.9: The mean absolute error (solid) of the virtual image rendered using the Propagation Algorithm compared to the estimated error bound of Theorem 3.1 (dashed) for each scanline of the scene shown in Fig. 3.8.

three IBR-related problems–namely, camera placement, budget allocation, and bit allocation–are briefly considered. The discussion focuses on the result of Theorem 3.1; similar implications can be drawn from Theorem 3.2. Finally, we present limitations of the proposed analysis in Section 3.7.5.

3.7.1 Actual cameras with different resolutions

The proposed analysis can be generalized to the case where actual cameras $\{\Pi_i\}_{i=1}^N$ have different resolutions $\{\Delta_i\}_{i=1}^N$. In this case, we need to modify the computation of the density function $\lambda_x(x)$ in Lemma 3.1. As a result, Equation (3.15) of Proposition 3.2 will also be changed to

$$\sum_{n=1}^{N_y-1} (y_{m+1} - y_m)^k \approx \int_a^b \left(\sum_{i=1}^{N} \frac{H_i'(u)}{\Delta_i} \right)^{1-k} (H_v'(u))^k \, du. \tag{3.49}$$

This equation suggests that different actual cameras contribute different amounts of information to the rendering process, depending on the relative position of the cameras to the scene and the resolution Δ_i (via the ratio $H_i'(u)/\Delta_i$). In particular, the larger $H_i'(u)$, the more information is contributed by the actual camera Π_i. Intuitively, as suggested in Appendix A.1, the derivative $H_i'(u)$ is larger if the camera is pointed toward the scene.

3.7.2 Where to put the actual cameras?

Theorem 3.1 suggests a potential application for camera placement. Suppose that we render a virtual image at camera Π_v given a number of N actual cameras

with given depth and texture estimate errors E_D, E_T. We want to find the optimal camera positions, that is, optimal matrices $\{\Pi\}_{i=1}^N$.

Given that E_D, E_T, and N are fixed, the last two terms of the error bound in Theorem 3.1 are also fixed. To decrease the error bound, the only way is to minimize Y_3 (see (3.16) for $k = 3$):

$$Y_3 = \int_a^b \left(\sum_{i=1}^N H_i'(u) \right)^{-2} (H_v'(u))^3 \, du.$$

In case Y_3 cannot be analytically minimized, numerical methods can be used to approximate the optimal configuration.

3.7.3 Budget allocation

Suppose that a monetary budget c is available to buy range finders and cameras of cost c_D, c_T, respectively. The question is to how to allocate the budget c into range finders and cameras to best render the virtual image at Π_v.

We assume that, due to the registration process, the depth estimate error E_D is a function of the number of range finders N_D. The texture estimate error E_T and resolution Δ_x are similar for all cameras. Hence, the error bound in (3.21) depends only on Y_3 and E_D. The optimal budget allocation is to use N_D^* range finders and N_T^* cameras, where N_D^*, N_T^* are the solution of the following optimization:

$$\min_{c_D N_D + c_T N_T \leq c} \left\{ \frac{3\Delta_x^2 \|f_v''\|_\infty}{4Y_1} Y_3(N_T) + B_v \|f_v'\|_\infty E_D(N_D) \right\}. \tag{3.50}$$

3.7.4 Bit allocation

Suppose that depth maps and images are recorded at the encoder and need to be transmitted over some communications channel to IBR decoders. The virtual image is rendered at the decoder upon receiving the depth maps and images. The question is how to distribute the channel capacity R into R_D for depth maps and R_T for images to optimize the rendering quality of the virtual images.

Let $E_T = E_T(R_T)$ and $E_D = E_D(R_D)$ be distortions of intensity and depth images corresponding to the transmission rate R_T, R_D. Since the first term of Theorem 3.1 does not depend on E_D, E_T, our optimal distribution of channel capacity $\{R_D^*, R_T^*\}$ is the solution of the following optimization:

$$\min_{R_D + R_T \leq R} \left\{ E_T(R_T) + B_v \|f_v''\|_\infty E_D(R_D) \right\}. \tag{3.51}$$

Table 3.1: Comparison of moments $E[(y_{m+1}-y_m)^k]$, for $N = 10$ actual cameras, with moments of the approximated Poisson process.

	$k=1$	$k=2$	$k=3$
Experiments	0.1	0.184	0.0046
Theory using Poisson	0.1	0.2	0.0060
Relative error	0%	8%	23%

3.7.5 Limitations

Limitations of the proposed analysis in Theorems 3.1 and 3.2 are due to two approximations, namely, the Poisson approximations and the integral approximations.

In Propositions 3.2 and 3.4, actual samples $\mathcal{Y}$ are approximated as a generalized Poisson process. It is known [83, 90] that the superposition of i.i.d. renewal processes converges to a Poisson process with convergence rate of order N^{-1}. To have an idea, note that the convergence rate of the Central Limit Theorem is of order $N^{-1/2}$. Table 3.1 shows a comparison of moments $E[(y_{m+1} - y_m)^k]$ to values calculated using Proposition 3.2.

In (3.22) and (3.39), the mean absolute errors are approximated as an integral using the *trapezoid rule*. This approximation's error can be bounded by [91, Chapter V]

$$E_{\text{trapezoid}} \leq \frac{H_v(b) - H_v(a)}{12} \Delta_x^2 \|f_v''\|_\infty. \tag{3.52}$$

Hence, a more conservative error bound can be used by adding the term $E_{\text{trapezoid}}$ in the right-hand side of (3.21).

Moreover, the second approximation also suggests why the resolution of the virtual camera does not appear in Theorems 3.1 and 3.2. We expect that the resolution Δ_x of the virtual camera does appear for nonideal sampling kernels $\varphi(x)$.

3.8 Conclusion

We proposed a new framework, the *error aggregation framework*, to quantitatively analyze the rendering quality of IBR algorithms using per-pixel depth. We showed that IBR errors can be bounded based on sample intervals, sample errors, and jitters. We approximated actual samples as a generalized Poisson process and bounded sample jitters. We derived, in Theorems 3.1 and 3.2, theoretical bounds for the mean absolute errors (MAEs). The bounds successfully captured, as validated by synthetic and actual scenes, the effects of various factors such as depth and intensity estimate errors, the scene geometry and texture, the number of cameras and their characteristics. We also discussed implications of our analysis for camera placement, budget allocation, and bit allocation.

For future research, we would like to further analyze the relationship between

Π and $H'_{\Pi}(u)$ and characterize the mean and variance of the rendering errors. This chapter's results may be extended to weakly calibrated scene-camera configurations. Generalizations of the results to 2D occluded scenes and 3D scenes will be presented in the next chapter.

CHAPTER 4

QUANTITATIVE ANALYSIS FOR IMAGE-BASED RENDERING: 2D OCCLUDED SCENES AND 3D SCENES

4.1 Introduction

Although many algorithms and systems have been proposed for *image-based rendering* (IBR) applications [3, 5], little research has been addressing the fundamental issue of analyzing the effects of the scene and the camera setting on the rendering quality. Understanding these effects is crucial to controlling the rendering quality and the cost of IBR systems. Many IBR algorithms in practice have to rely on oversampling to counter undesirable aliasing effects.

In the previous chapter, we quantitatively analyzed, for 2D unoccluded scenes, the quality of IBR texture mapping algorithms using explicit depth maps. We proposed an *error aggregation framework* to bound rendering errors based on the sample values, sample positions, and their errors, whether the virtual image is interpolated in image-space or object-space. The union of sample positions is approximated as a generalized Poisson process, while the sample jitters are bounded based on the relative position between the virtual camera and the scene. We derived error bounds for several IBR algorithms using per-pixel depth. The derived error bounds show the effects on the rendering quality of various factors including depth and intensity estimate errors, the scene geometry and texture, the number of actual cameras, their positions and resolutions. Implications of the proposed analysis include camera placement, budget allocation, and bit allocation.

In this chapter, we extend the analysis in [77] to 2D occluded scenes and 3D unoccluded scenes. The main contribution of the chapter is a methodology armed with a set of techniques to analyze the rendering quality of IBR algorithms, assuming per-pixel depth as inputs, using image-space interpolation. To analyze 2D occluded scenes, we measure, in Proposition 4.1, the effects of jumps in sample intervals around the discontinuities of the virtual image, re-

[0]This chapter includes research conducted jointly with Prof. Minh Do [78].

sulting in additional terms in the error bound. We extend the analysis to 3D unoccluded scenes by proposing novel machineries, including an error bound for triangulation-based linear interpolation and the use of Poisson Delaunay triangles' properties–classical results from stochastic geometry [92]. We find that, in smooth regions, triangulation-based linear interpolation for 3D scenes results in a decay order $\mathcal{O}(\lambda^{-1})$ of the mean absolute error (MAE), where λ is the local density of actual samples, compared to $\mathcal{O}(\lambda^{-2})$ for 2D scenes. This intriguing finding implies that for 3D scenes, building IBR systems that can simplify to 2D, such as adopting image rectifications and planar camera configurations, besides decreasing the complexity, also increases the decay order of the rendering errors in smooth regions.

This chapter is organized as follows. The problem setup is presented in Section 4.2. We present analysis of 2D occluded scenes in Section 4.3. Generalization to 3D is given in Section 4.4. Finally, we offer concluding remarks in Section 4.5.

4.2 Problem Setup

We start with a description of the scene model in Section 4.2.1. The camera model is presented in Section 4.2.2. We describe our models for the 3D case that are parallel with 2D models considered in the previous chapter. This description also introduces the notation used in the chapter. Finally, we state the problem in Section 4.2.3.

4.2.1 The scene model

The surface of the scene is modeled as a 3D parameterized surface $\boldsymbol{S}(u,v) : \Omega \to \mathbb{R}^3$, for some region $\Omega \subset \mathbb{R}^2$. The texture map $T(u,v) : \Omega \to \mathbb{R}$ is an intensity function "painted" on the surface $\boldsymbol{S}(u,v)$. We assume that the surface is Lambertian [67], that is, images of the same surface point at different cameras have the same intensity. Furthermore, we assume that the surface $\boldsymbol{S}(u,v)$ and the texture $T(u,v)$ have derivative of second order at all points and all directions, except at the discontinuities.

4.2.2 The camera model

A 3D pinhole camera (Fig. 4.1) is characterized by the positional matrix $\Pi = [\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3]^T \in \mathbb{R}^{3\times 4}$. For each surface point $\boldsymbol{S} = [X, Y, Z]^T$ in the scene, let its homogeneous coordinate [67] be $\widetilde{\boldsymbol{S}} = [X, Y, Z, 1]^T$. The projection equation is

$$d \cdot [x, y, 1]^T = \Pi \cdot [X, Y, Z, 1]^T \,, \tag{4.1}$$

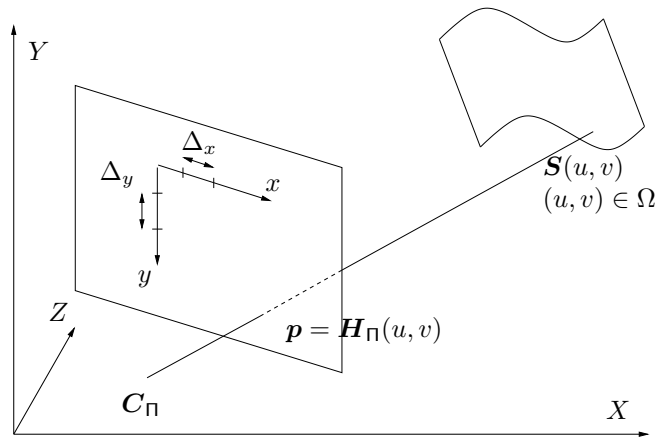


Figure 4.1: The 3D calibrated scene-camera model. The scene surface is modeled as a 3D parameterized surface $\boldsymbol{S}(u,v)$ for $(u,v) \in \Omega \subset \mathbb{R}^2$. The texture $T(u,v)$ is "painted" on the surface. We assume pinhole camera model with calibrated positional matrix $\Pi \in \mathbb{R}^{3\times 4}$. The camera resolution is characterized by the pixel intervals Δ_x, Δ_y in horizontal and vertical direction on the image plane.

where $d = \boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}$ is the depth of $\boldsymbol{S}$ relative to Π. We derive a *scene-to-image mapping* $\boldsymbol{H}_\Pi(u,v)$ from surface points $\boldsymbol{S}(u,v)$, for $(u,v) \in \Omega$, to their image points (x,y) as

$$\begin{bmatrix} x \\ y \end{bmatrix} \stackrel{\text{def}}{=} \boldsymbol{H}_\Pi(u,v) = \begin{bmatrix} H_x(u,v) \\ H_y(u,v) \end{bmatrix}, \tag{4.2}$$

where

$$H_x(u,v) = \frac{\boldsymbol{\pi}_1^T \cdot \widetilde{\boldsymbol{S}}(u,v)}{\boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}(u,v)}, \quad H_y(u,v) = \frac{\boldsymbol{\pi}_2^T \cdot \widetilde{\boldsymbol{S}}(u,v)}{\boldsymbol{\pi}_3^T \cdot \widetilde{\boldsymbol{S}}(u,v)}. \tag{4.3}$$

The Jacobian matrix of $\boldsymbol{H}_\Pi$ is

$$\frac{\partial \boldsymbol{H}_\Pi(u,v)}{\partial(u,v)} = \begin{bmatrix} \partial H_x/\partial u & \partial H_x/\partial v \\ \partial H_y/\partial u & \partial H_y/\partial v \end{bmatrix}. \tag{4.4}$$

At the image plane of a camera Π, the *image light field* $f_\Pi(x,y)$ at image point (x,y) characterizes the "brightness" $T(u,v)$ of surface point $\boldsymbol{S}(u,v)$ having image at (x,y). In other words, the image light field $f_\Pi(x,y)$ is perspectively corrected from the texture map $T(u,v)$ as

$$f_\Pi(x,y) = T\big(\boldsymbol{H}_\Pi^{-1}(x,y)\big). \tag{4.5}$$

Let Δ_x, Δ_y be the sample intervals in horizontal and vertical directions of the discrete grid on which the actual images are sampled from the image light field. We refer the product $\Delta_x\Delta_y$ to the resolution of the camera. If $\varphi(x,y)$ is the sampling kernel of the camera Π, the pixel intensity $I_\Pi[m,n]$ is the value of the convolution of $f_\Pi(x,y)$ and $\varphi(x,y)$, evaluated at $(x_m, y_n) = (m\Delta_x, n\Delta_y)$,

as follows:

$$I_{\Pi}[m,n] = (f_{\Pi} * \varphi)(x_m, y_n) \tag{4.6}$$

$$= \int_{\boldsymbol{H}_{\Pi}(\Omega)} f_{\Pi}(x,y) \cdot \varphi(x_m - x, y_n - y) dx dy. \tag{4.7}$$

In this chapter, we assume the ideal pinhole camera model with the Dirac delta function as the sampling kernel $\varphi(x,y)$, i.e., $\varphi(x,y) = \delta(x,y)$. In other words,

$$I_{\Pi}[m,n] = f_{\Pi}(m\Delta_x, n\Delta_y). \tag{4.8}$$

Depth and intensity estimate error. In practice, the depth and the intensity at actual pixels are subjected to errors $\boldsymbol{\varepsilon}_D = [X_e - X, Y_e - Y, Z_e - Z]^T$ and $\varepsilon_T = T_e(u,v) - T(u,v)$, respectively. We suppose that $\boldsymbol{\varepsilon}_D$ and ε_T are bounded by E_D and E_T, that is,

$$\|\boldsymbol{\varepsilon}_D\|_2 \leq E_D, \qquad |\varepsilon_T| \leq E_T. \tag{4.9}$$

4.2.3 Problem statement

IBR algorithms. Many IBR algorithms have been proposed [3, 5, 80]. This chapter is concerned with IBR algorithms using per-pixel depth and image-space interpolation [41, 46, 62, 64]. We present our analysis for the Propagation Algorithm [62], although the proposed techniques are applicable to other algorithms.

We assume that piecewise linear interpolation is used for the 2D case and Delaunay triangulation-based linear interpolation is used for the 3D case. Both methods are widely used in practice thanks to their simplicity and decent interpolation qualities. Furthermore, we hope to help IBR practitioners find the chapter directly useful. We note that the proposed analysis also applies for interpolation techniques using higher order splines [18, 93].

Problem statement. Suppose the virtual image at virtual camera Π_v is rendered using images and depth maps of N actual cameras $\{\Pi_i\}_{i=1}^N$. We want to quantify the effects on the rendering quality of projection matrices $\{\Pi_i\}_{i=1}^N$ and Π_v, the resolution $\Delta_x\Delta_y$, the depth and intensity estimate error bound E_D, E_T, the texture map $T(u,v)$, and the surface geometry $\boldsymbol{S}(u,v)$.

4.3 Analysis for 2D Scenes

In this section, we extend the analysis proposed in the previous chapter to 2D occluded scenes. We present the new methodology in Section 4.3.1 and revisit relevant results of [77] in Section 4.3.2. In Section 4.3.3, we analyze the rendering quality for 2D occluded scenes.

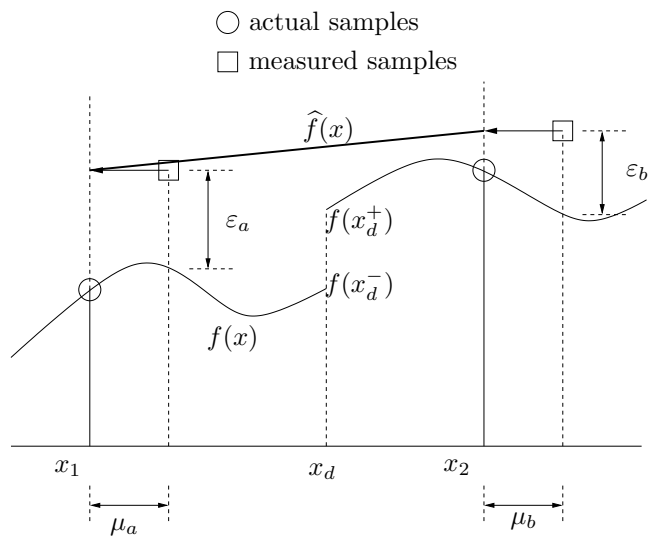


Figure 4.2: Linear interpolation of a discontinuous function.

4.3.1 Methodology

We extend the methodology proposed in [77] to consider discontinuous functions. The linear interpolation $\widehat{f}(x)$ of $f(x)$ in an interval $[x_1, x_2]$ is defined as

$$\widehat{f}(x) = \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1) + \frac{x - x_1}{x_2 - x_1} \cdot f(x_2). \tag{4.10}$$

In the presence of sample errors $\varepsilon_1, \varepsilon_2$ and sample jitters μ_1, μ_2 (see Fig. 4.2), the sample values $f(x_1), f(x_2)$ in (4.10) are replaced by $f(x_1 + \mu_1) + \varepsilon_1, f(x_2 + \mu_2) + \varepsilon_2$, respectively. The $\mathcal{L}_\infty$ norm of a function $g(x)$ is defined as

$$\|g\|_\infty = \sup_x \{g(x)\}. \tag{4.11}$$

In the following, we bound the linear interpolation error for functions with only one discontinuity. Note that general analysis is possible, although less elegant, and provides similar findings. First, for simplicity we introduced notations:

$$\Delta_1 = x_d - x_1, \quad \Delta_2 = x_2 - x_d, \quad \Delta = x_2 - x_1. \tag{4.12}$$

Proposition 4.1 *Consider a function $f(x)$ that is twice continuously differentiable except at* the *discontinuity x_d. The aggregated error over $[x_1, x_2]$ of the linear interpolation given in* (4.10)*, defined below, can be bounded by*

$$\begin{aligned}\int_{x_1}^{x_2} |\widehat{f}(x) - f(x)| \quad \leq \quad & \frac{1}{8}\Delta^3 \cdot \|f''\|_\infty + \frac{1}{2}\Delta_1\Delta_2 \cdot |J_1| + \frac{3}{2}\Delta \cdot |J_0| \\ & + \Delta\Big(\max_{i=1,2}\{|\varepsilon_i|\} + \max_{i=1,2}\{|\mu_i|\} \cdot \|f'\|_\infty\Big),\end{aligned} \tag{4.13}$$

where J_0, J_1 are the jumps of $f(x)$ and its derivative at the discontinuity x_d:

$$J_0 = f(x_d^+) - f(x_d^-), \quad J_1 = f'(x_d^+) - f'(x_d^-). \tag{4.14}$$

Proof 4.1 *See Appendix A.2.*

Remark 4.1 *The bound in Proposition 4.1 is proposed for the aggregated error over the interval $[x_1, x_2]$. This is different from the pointwise bound given in [77, Proposition 1]. Proposition 4.1 can be considered as a local analysis, providing a bound for the interpolation error in individual intervals. Because of the discontinuity at x_d, the aggregated error increases by an amount of*

$$\frac{1}{2}\Delta_1\Delta_2 \cdot |J_1| + \frac{3}{2}\Delta \cdot |J_0|.$$

If $J_0 = J_1 = 0$, the bound of Proposition 4.1 simplifies to the case where $f(x)$ is twice continuously differentiable in $[x_1, x_2]$ (see [77, Proposition 1]). The bound is characterized by sample intervals, sample errors, and jitters, in addition to intrinsic properties of $f(x)$. Similar remarks can be drawn for interpolation using splines of higher orders [18].

The bound in (4.13) suggests that we need to investigate the sample intervals, especially observed sample intervals around the discontinuities, and sample jitters in the context of IBR.

4.3.2 Part I revisited – analysis for 2D scenes without occlusions

We state in this section key results of the previous chapter for 2D unoccluded scenes. The presentation helps to understand previous results and the development of this chapter. In Proposition 4.2, we present the property of sample intervals. We give a bound for sample jitters in Proposition 4.3. We provide an error bound, in Theorem 4.1, for the virtual images rendered using the Propagation Algorithm [62].

Properties of sample intervals. On the image plane of the virtual camera Π_v, let $\mathcal{Y}$ be the set of points *propagated* from actual pixels [62].

Proposition 4.2 *The point process $\mathcal{Y}$ can be approximated as a generalized Poisson process with density function*

$$\lambda_x(x) = \frac{1}{\Delta_x H'_v(u)} \cdot \sum_{i=1}^{N} H'_i(u), \tag{4.15}$$

where $u = H_v^{-1}(x)$. The sum of powers of the sample intervals can be computed as

$$\sum_{n=1}^{N_{\mathcal{Y}}-1} (y_{m+1} - y_m)^k \approx k!\, Y_k\, \Delta_x^{k-1}, \tag{4.16}$$

where

$$Y_k = \int_a^b \left(\sum_{i=1}^{N} H_i'(u) \right)^{1-k} \left(H_v'(u)\right)^k du. \tag{4.17}$$

Bounds for sample jitters. Let $\boldsymbol{S}$ be a surface point and $\boldsymbol{S}_e$ be an erroneous estimate of $\boldsymbol{S}$. We suppose that y and $\widehat{y}$ are images of $\boldsymbol{S}$ and $\boldsymbol{S}_e$ at the virtual camera Π_v.

Proposition 4.3 *The jitter $\mu = \widehat{y} - y$, at virtual camera Π_v with camera center $\boldsymbol{C}_v$, caused by depth estimate errors is bounded by*

$$|\mu| \leq E_D B_v. \tag{4.18}$$

In the above inequality, B_v is determined as

$$B_v \quad = \quad \sup_{u \in [a,b]} \quad \left\{ \frac{\|\boldsymbol{C}_v - \boldsymbol{S}(u)\|_2}{d(u)^2} \right\}. \tag{4.19}$$

Bound for rendering errors. Apply the methodology of Proposition 4.1, using the results of Proposition 4.2 and 4.3, we can derive an error bound for the rendered image of the Propagation Algorithm.

Theorem 4.1 *The* mean absolute error $\mathrm{MAE}_{\mathrm{PA}}$ *of the virtual image using the Propagation Algorithm is bounded by*

$$\mathrm{MAE}_{\mathrm{PA}} \leq \frac{3Y_3}{4Y_1} \Delta_x^2 \|f_v''\|_\infty + E_T + E_D B_v \|f_v'\|_\infty, \tag{4.20}$$

where $f_v(x)$ is the virtual image, Y_k is defined as in (4.17)*, B_v is as in* (4.19)*, and E_D, E_T are as in* (4.9)*.*

Remark 4.2 *In the first term of* (4.20)*, $Y_1 = H_v(b) - H_v(a)$ is independent of the number of actual cameras N. The value of Y_3, called image-space multiple-view term of third order, encodes the geometrical position between the actual cameras and the scene. Note that the first term has decay order $\mathcal{O}(\lambda^{-2})$, where λ is the local density of actual samples. The second term is the intensity estimate error bound E_T of the actual cameras. The third term relates to the depth estimate error bound E_D and the geometrical position between the scene and the virtual camera (via B_v).*

4.3.3 Analysis for 2D occluded scenes

In this section, we consider 2D occluded scenes by introducing two adjustments compared to the analysis in [77]. First, the presence of occlusions requires modification of the sample density $\lambda_x(x)$. Second, intervals containing discontinuities

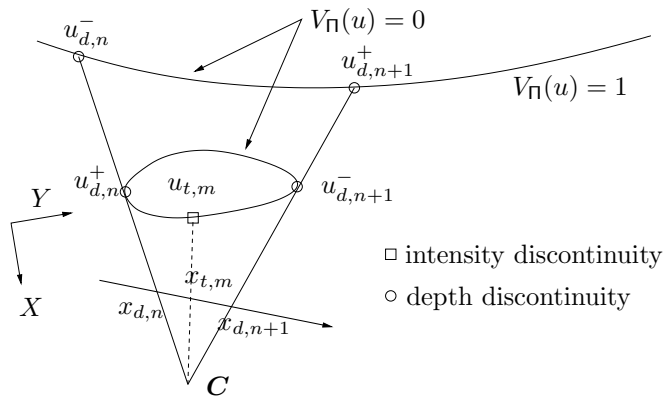


Figure 4.3: A 2D occluded scene. We differentiate two kinds of discontinuities: those due to occlusions (such as $x_{d,n}$ with parameters $u^+_{d,n}$ and $u^-_{d,n}$) and those due to the texture $T(u)$ (such as $x_{t,n}$ with parameter $u_{t,m}$).

of the virtual image, either caused by the intensity or depth discontinuities, need to be analyzed using the new methodology of Proposition 4.1. For simplicity, we assume that all the occluded samples are successfully removed, and the set of remaining samples are dense enough so that there exists at most one discontinuity in each sample interval. General analysis is possible, though less elegant, and produces similar findings.

Modification of the sample density. Consider a 2D occluded scene (see Fig. 4.3). For a camera Π, we define the visibility function

$$V_\Pi(u) = \begin{cases} 1, & \text{if } \boldsymbol{S}(u) \textit{ is } \text{visible at } \Pi \\ 0, & \text{if } \boldsymbol{S}(u) \textit{ is not } \text{visible at } \Pi. \end{cases} \tag{4.21}$$

Proposition 4.2 is modified as

$$\lambda_x(x) = \frac{1}{\Delta_x H'_v(u)} \cdot \sum_{i=1}^{N} V_i(u) H'_i(u), \tag{4.22}$$

where u is the parameter of the surface point $\boldsymbol{S}(u)$ having image at x:

$$u = \arg\min_u \{d(u) : H_\Pi(u) = x\}. \tag{4.23}$$

For this modification, Y_k will also be changed to

$$Y_k = \int_a^b \left(\sum_{i=1}^{N} V_i(u) H'_i(u) \right)^{1-k} \left(V_v(u) H'_v(u) \right)^k du. \tag{4.24}$$

Intuitively, the modification in (4.24) signifies that, if a surface point $\boldsymbol{S}(u)$ is occluded at an actual camera Π_i, or equivalently $V_i(u) = 0$, this camera Π_i contributes no information to the rendering of virtual pixel $x = H_v(u)$. Similarly, if $\boldsymbol{S}(u)$ is occluded at the virtual camera Π_v, or equivalently $V_v(u) = 0$,

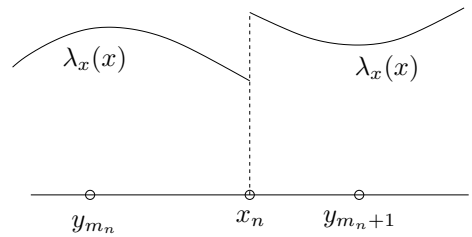


Figure 4.4: The observed interval $[y_{m_n}, y_{m_n+1}]$ around a discontinuity x_n of the virtual image $f_v(x)$. Note that the sample density function $\lambda_x(x)$ may or may not, depending on whether $x_n \in \mathcal{X}_d$ or $x_n \in \mathcal{X}_t$, be discontinuous at x_n.

no information from actual cameras is necessary.

Incorporation of jumps. We differentiate two categories of discontinuities at the virtual image $f_v(x)$, namely, the depth discontinuities and the texture discontinuities (see Fig. 4.3). The *depth* discontinuities are at image object boundaries (backgrounds and foregrounds). Let $\mathcal{X}_d$ be the set of depth discontinuities. For each point $x_{d,n} \in \mathcal{X}_d$, denote

$$u_{d,n}^{+} = \lim_{x \to x_{d,n}^{+}} H_v^{-1}(x), \tag{4.25}$$

$$u_{d,n}^{-} = \lim_{x \to x_{d,n}^{-}} H_v^{-1}(x). \tag{4.26}$$

The above equations are well defined since $H_v^{-1}(x)$ is a one-to-one mapping everywhere except at discontinuities of $f_v(x)$. Intuitively, $u_{d,n}^{+}$ is the parameter on the background and $u_{d,n}^{-}$ is the parameter on the foreground, or vice-versa.

The *texture* discontinuities are discontinuities of the texture $T(u)$. We denote the set of texture discontinuities $\mathcal{X}_t$. For consistency, we also use notation $u_{t,n}^{+}$ and $u_{t,n}^{-}$, as in (4.25) and (4.26), for $x_{t,n} \in \mathcal{X}_t$, though they are in fact equal.

For each discontinuity

$$x_n \in \mathcal{X} = \mathcal{X}_t \bigcup \mathcal{X}_d, \tag{4.27}$$

the interval $[y_{m_n}, y_{m_n+1}]$ containing x_n is called an *observed interval* (or sampled interval–see Fig. 4.4). The following lemma is a classical result of Poisson processes.

Lemma 4.1 *[88, 87] Let $(y_{m_n+1} - y_{m_n})$ be the observed interval around each discontinuity x_n. The length of intervals $\Delta_{2,n} = y_{m_n+1} - x_n$ and $\Delta_{1,n} = x_n - y_{m_n}$ are independent and follow exponential distributions of parameter $\lambda(H_v(u_n^{+}))$ and $\lambda(H_v(u_n^{-}))$, respectively.*

Corollary 4.1 *The following equations hold:*

$$E[y_{m_n+1} - y_{m_n}] = \frac{1}{\lambda(H_v(u_n^+))} + \frac{1}{\lambda(H_v(u_n^-))} \tag{4.28}$$

$$E[\Delta_{1,n}\Delta_{2,n}] = \frac{1}{\lambda(H_v(u_n^+))} \cdot \frac{1}{\lambda(H_v(u_n^-))}. \tag{4.29}$$

We define operators $\mathcal{J}_0(f)$ and $\mathcal{J}_1(f)$ as

$$\mathcal{J}_0(f) = \sup_x \left\{ \left| \lim_{y\to x^+} f(y) - \lim_{y\to x^-} f(y) \right| \right\} \tag{4.30}$$

$$\mathcal{J}_1(f) = \sup_x \left\{ \left| \lim_{y\to x^+} f'(y) - \lim_{y\to x^-} f'(y) \right| \right\}. \tag{4.31}$$

Theorem 4.2 *The mean absolute error* $\mathrm{MAE}_{\mathrm{PA}}$ *of the virtual image using the Propagation Algorithm is bounded by*

$$\begin{aligned} \mathrm{MAE}_{\mathrm{PA}} \leq\ & \frac{3Y_3}{4Y_1}\Delta_x^2 \|f_v''\|_\infty + E_T + E_D B_v \|f_v'\|_\infty \\ & + \frac{3}{2} D_0 \Delta_x \mathcal{J}_0(f_v) + \frac{1}{2} D_1 \Delta_x^2 \mathcal{J}_1(f_v), \end{aligned} \tag{4.32}$$

where $f_v(x)$ is the virtual image, B_v is as in (4.19)*, Y_k is defined in* (4.24)*, $\mathcal{J}_0(f)$ and $\mathcal{J}_1(f)$ are defined in* (4.30) *and* (4.31)*, and D_0, D_1 are*

$$D_0 = \sum_{x_d \in \mathcal{X}} \frac{1}{\lambda_x(x_d^+)} + \frac{1}{\lambda_x(x_d^-)} \tag{4.33}$$

$$D_1 = \sum_{x_d \in \mathcal{X}} \frac{1}{\lambda_x(x_d^+)} \cdot \frac{1}{\lambda_x(x_d^-)}. \tag{4.34}$$

Proof 4.2 *The proof is similar to the proof of [77, Theorem 1]; we need to consider in addition the aggregated error in observed intervals $[y_{m_n}, y_{m_n+1}]$ around jumps $\{x_n \in \mathcal{X}\}$. Hence, the error bound needs to increase by an amount*

$$\frac{3}{2}|y_{m_n+1} - y_{m_n}| \cdot \mathcal{J}_0(f) + \frac{1}{2}\Delta_{1,n}\Delta_{2,n} \cdot \mathcal{J}_1(f). \tag{4.35}$$

The summation these terms, for all x_n, in fact results in the additional fourth and fifth terms.

Remark 4.3 *Compared to Theorem 4.1, the bound in* (4.32) *has additional fourth and fifth terms to incorporate the discontinuities of the virtual image $f_v(x)$. Overall, the fourth term decays as $\mathcal{O}(\lambda^{-1})$ and the fifth term decays as $\mathcal{O}(\lambda^{-2})$, where λ is the local density of actual samples.*

4.4 Analysis for 3D Scenes

In this section, we extend the analysis into the 3D case. A natural generalization of piecewise linear interpolation into 2D is the Delaunay triangulation-based linear interpolation. We present the 3D methodology for individual triangles in Section 4.4.1. Then, we show properties of Poisson Delaunay triangles in Section 4.4.2 and a bound for sample jitters in Section 4.4.3. Finally, an error analysis for 3D scenes without occlusions is given in Section 4.4.4.

4.4.1 Methodology

In this section, we investigate the interpolation error for an individual triangle. We define the $\mathcal{L}_\infty$ norm of the gradient $\nabla f(x,y)$ and the Hessian matrix $\nabla^2 f(x,y)$ as follows:

$$\|\nabla f(x,y)\|_\infty = \sup_{(x,y)} \{\|\nabla f(x,y)\|_2\} \tag{4.36}$$

$$\|\nabla^2 f(x,y)\|_\infty = \sup_{(x,y)} \left\{\sigma_{\max}\left[\nabla^2 f(x,y)\right]\right\}, \tag{4.37}$$

where $\sigma_{\max}[\mathsf{M}]$ denotes the maximum singular value [94] of a matrix M. The linearly interpolated value at a 2D point $\boldsymbol{X}$ inside a triangle $\boldsymbol{ABC}$ is defined as

$$\widehat{f}(\boldsymbol{X}) = \frac{S_{\boldsymbol{A}}}{S} f(\boldsymbol{A}) + \frac{S_{\boldsymbol{B}}}{S} f(\boldsymbol{B}) + \frac{S_{\boldsymbol{C}}}{S} f(\boldsymbol{C}), \tag{4.38}$$

where $S_{\boldsymbol{A}}, S_{\boldsymbol{B}}, S_{\boldsymbol{C}}$, and S denote the area of triangles $\boldsymbol{XBC}, \boldsymbol{AXC}, \boldsymbol{ABX}$, and $\boldsymbol{ABC}$, respectively. In other words, $\widehat{f}(\boldsymbol{X})$ is a bivariate linear function that is equal to $f(\boldsymbol{X})$ at locations $\boldsymbol{A}, \boldsymbol{B}$, and $\boldsymbol{C}$ (see Fig. 4.5). In the presence of sample errors and jitters, sample values $f(\boldsymbol{A}), f(\boldsymbol{B})$, and $f(\boldsymbol{C})$ in (4.38) are replaced by $f(\boldsymbol{A}+\boldsymbol{\mu}_{\boldsymbol{A}})+\varepsilon_{\boldsymbol{A}}, f(\boldsymbol{B}+\boldsymbol{\mu}_{\boldsymbol{B}})+\varepsilon_{\boldsymbol{B}}$, and $f(\boldsymbol{C}+\boldsymbol{\mu}_{\boldsymbol{C}})+\varepsilon_{\boldsymbol{C}}$, respectively.

Proposition 4.4 *We consider a function $f(x,y)$ that is twice continuously differentiable. The linear interpolation on a triangle given in* (4.38) *has the error bounded by*

$$\begin{aligned} |\widehat{f}(x,y) - f(x,y)| \leq\ & \frac{1}{2}R^2 \cdot \|\nabla^2 f\|_\infty + \max\{|\varepsilon|\} \\ & + \max\{\|\boldsymbol{\mu}\|_2\} \cdot \|\nabla f\|_\infty, \end{aligned} \tag{4.39}$$

where R is the circumcircle radius of the triangle $\boldsymbol{ABC}$.

Proof 4.3 *We show the proof for $\varepsilon = 0$ and $\boldsymbol{\mu} = \boldsymbol{0}$. In this case, the error bound in the right-hand side of* (4.39) *reduces into the first term. The techniques to incorporate the sample error (second term) and jitter (third term) are similar to the proof of [77, Proposition 1].*

Let $\boldsymbol{O}$ be the center of the circumcircle of triangle $\boldsymbol{ABC}$. Using vector

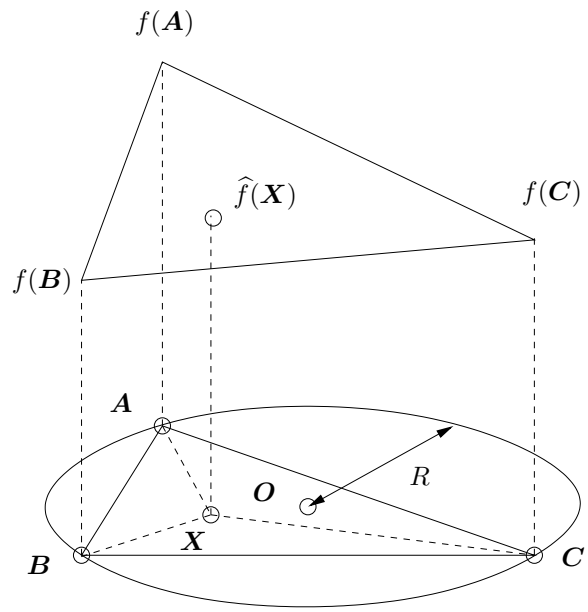


Figure 4.5: Triangulation-based linear interpolation is often used with the Delaunay triangulation. For each triangle, the interpolation error can be bounded using the circumcircle radius R, the sample errors ε, and the sample jitters $\boldsymbol{\mu}$ (see Proposition 4.4).

manipulations, it can be shown that

$$R^2 - \|\boldsymbol{X} - \boldsymbol{O}\|_2^2 = \frac{S_{\boldsymbol{A}}}{S}\|\Delta_{\boldsymbol{A}}\|_2^2 + \frac{S_{\boldsymbol{B}}}{S}\|\Delta_{\boldsymbol{B}}\|_2^2 + \frac{S_{\boldsymbol{C}}}{S}\|\Delta_{\boldsymbol{C}}\|_2^2, \tag{4.40}$$

where $\Delta_{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{X}, \Delta_{\boldsymbol{B}} = \boldsymbol{B} - \boldsymbol{X}$, *and* $\Delta_{\boldsymbol{C}} = \boldsymbol{C} - \boldsymbol{X}$. *Using the 2D Taylor expansion we can obtain*

$$f(\boldsymbol{A}) = f(\boldsymbol{X}) + \nabla f(\boldsymbol{X})^T \cdot \Delta_{\boldsymbol{A}} + \frac{1}{2}\Delta_{\boldsymbol{A}}^T \cdot \nabla^2 f(\boldsymbol{X}_a) \cdot \Delta_{\boldsymbol{A}}$$

for some point $\boldsymbol{X}_a$. *Similar equations can be obtained for* $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *as well. Hence,*

$$\begin{aligned}
|\widehat{f}(\boldsymbol{X}) - f(\boldsymbol{X})| &= \frac{1}{2} \cdot \Big| \frac{S_{\boldsymbol{A}}}{S}\Delta_{\boldsymbol{A}}^T \cdot \nabla^2 f(\boldsymbol{X}_a) \cdot \Delta_{\boldsymbol{A}} + \\
&\quad \frac{S_{\boldsymbol{B}}}{S}\Delta_{\boldsymbol{B}}^T \cdot \nabla^2 f(\boldsymbol{X}_b) \cdot \Delta_{\boldsymbol{B}} + \frac{S_{\boldsymbol{C}}}{S}\Delta_{\boldsymbol{C}}^T \cdot \nabla^2 f(\boldsymbol{X}_c) \cdot \Delta_{\boldsymbol{C}} \Big| \\
&\leq \frac{1}{2}\|\nabla^2 f\|_\infty \Big(\frac{S_{\boldsymbol{A}}}{S}\|\Delta_{\boldsymbol{A}}\|_2^2 + \frac{S_{\boldsymbol{B}}}{S}\|\Delta_{\boldsymbol{B}}\|_2^2 + \frac{S_{\boldsymbol{C}}}{S}\|\Delta_{\boldsymbol{C}}\|_2^2 \Big) \\
&\leq \frac{1}{2}\|\nabla^2 f\|_\infty R^2.
\end{aligned}$$

The bound in (4.39) suggests that we need to investigate the properties of Delaunay triangles and the sample jitters. The next two sections will present these properties.

4.4.2 Properties of Poisson Delaunay triangles

We assume in this section that the scene is unoccluded. We start by proposing an equivalence of [77, Lemma 1] for the 2D case. A 2D process p is called *identically distributed scattering* [92] if the density of points of p over an arbitrary region ω follows a fixed probability mass distribution independent of ω. Intuitively, there is a profound similarity between the 1D and 2D cases, since they are both related to the probability mass function (pmf) of the number of points falling inside an arbitrary region. Hence, in the following, we assume that the Hypothesis 4.1 below holds.

Hypothesis 4.1 *The superposition of 2D point processes with identically distributed scattering property can be approximated as a 2D Poisson process.*

On the image plane of the virtual camera Π_v, let $\mathcal{Y}$ be the set of points *propagated* from actual pixels [62].

Proposition 4.5 *The point process $\mathcal{Y}$ can be approximated as a 2D generalized Poisson process with density function*

$$\lambda(x,y) = \frac{1}{\Delta_x \Delta_y} \sum_{i=1}^{N} \frac{\det\Big(\partial \boldsymbol{H}_i / \partial(u,v)\Big)}{\det\Big(\partial \boldsymbol{H}_v / \partial(u,v)\Big)}, \tag{4.41}$$

where $(u,v) = \boldsymbol{H}_v^{-1}(x,y)$.

Proof 4.4 *Since we assume that Hypothesis 4.1 holds, in each infinitesimal region, the point process $\mathcal{Y}$ can be considered as a 2D Poisson process. Hence, overall, $\mathcal{Y}$ can be considered as a generalized Poisson process. The density $\lambda(x,y)$ can be computed, similarly to [77, Section III.B], as the average number of points falling on an unit area. This indeed results in* (4.41).

Once we approximate the set of propagated points $\mathcal{Y}$ as a 2D Poisson process, the next step is to investigate properties of Poisson Delaunay triangles. In the following, we exploit results from stochastic geometry.

Lemma 4.2 *[92, Chapter 5] The circumradius R and the area S of Delaunay triangles of a 2D Poisson process of density λ are* independent. *The circumradius R has the probability density function (pdf)*

$$2(\pi\lambda)^2 r^3 e^{-\pi\lambda r^2}, \quad r > 0. \tag{4.42}$$

The moments $E[S^k]$ can be computed using explicit formula. In particular

$$E[R^2] = \frac{2}{\pi\lambda}, \quad E[S] = \frac{1}{2\lambda}, \quad E[S^2] = \frac{35}{8\pi^2\lambda^2}. \tag{4.43}$$

4.4.3 Bound for sample jitters

Let $\boldsymbol{S} = [X, Y, Z]^T$ be a surface point, and $\boldsymbol{p}$ be the image of $\boldsymbol{S}$ at the virtual camera Π_v. We denote $\boldsymbol{S}_e = [X_e, Y_e, Z_e]^T$ a noisy estimate of $\boldsymbol{S}$ with reconstruction error $\boldsymbol{\varepsilon}_D = \boldsymbol{S}_e - \boldsymbol{S}$, and $\widehat{\boldsymbol{p}}$ the image of $\boldsymbol{S}_e$ at Π_v.

Proposition 4.6 *The jitter $\boldsymbol{\mu} = \widehat{\boldsymbol{p}} - \boldsymbol{p}$, at virtual camera Π_v with camera center $\boldsymbol{C}_v$, caused by depth estimate errors can be bounded by*

$$\|\boldsymbol{\mu}\|_2 \leq \sqrt{2} E_D B_v. \tag{4.44}$$

In the above inequality, B_v is computed as

$$B_v = \sup_{(u,v)\in\Omega} \left\{ \frac{\|\boldsymbol{C}_v - \boldsymbol{S}(u,v)\|_2}{d(u,v)^2} \right\}. \tag{4.45}$$

Proof 4.5 *The jitter $\boldsymbol{\mu}$ is a two-coordinate vector $\boldsymbol{\mu} = [\mu_x, \mu_y]^T$. It can be shown [77, Section III.C] that the norm of both μ_x and μ_y is bounded by $E_D B_v$. Hence*

$$\|\boldsymbol{\mu}\|_2 = \sqrt{\mu_x^2 + \mu_y^2} \leq \sqrt{2} E_D B_v.$$

The bound $E_D B_v$ depends on the depth estimate error E_D and the relative position between the virtual camera and the scene defined by B_v.

4.4.4 Analysis for 3D unoccluded scenes

Consider the intensity function $f_v(x, y) = T\big(\boldsymbol{H}_v^{-1}(x, y)\big)$ at virtual camera Π_v. Let $e(x, y) = \widehat{f}_v(x, y) - f_v(x, y)$ be the interpolation error and $\mathcal{N}_\Omega$ be the set of virtual pixels (m, n) being images of surface points $\boldsymbol{S}(u, v)$ for $(u, v) \in \Omega$. Denote $\#\mathcal{N}_\Omega$ the number of pixels in $\mathcal{N}_\Omega$. The *mean absolute error* $\mathrm{MAE}_{\mathrm{PA}}$ is defined as

$$\mathrm{MAE}_{\mathrm{PA}} = \frac{1}{\#\mathcal{N}_\Omega} \sum_{(m,n)\in\mathcal{N}_\Omega} |e(m\Delta_x, n\Delta_y)|. \tag{4.46}$$

Theorem 4.3 *The* mean absolute error $\mathrm{MAE}_{\mathrm{PA}}$ *of the virtual image using the Propagation Algorithm is bounded by*

$$\mathrm{MAE}_{\mathrm{PA}} \leq \frac{X_2}{\pi X_1} \Delta_x \Delta_y \|\nabla^2 f_v\|_\infty + E_T + \sqrt{2} E_D B_v \|\nabla f_v\|_\infty, \tag{4.47}$$

where f_v is the the virtual image, B_v is as in (4.45)*, E_D, E_T are as in* (4.9)*, and*

$$X_k = \int_\Omega \left(\sum_{i=1}^{N} \det\left(\frac{\partial \boldsymbol{H}_i(u,v)}{\partial(u,v)} \right) \right)^{1-k} \left(\det\left(\frac{\partial \boldsymbol{H}_v(u,v)}{\partial(u,v)} \right) \right)^k dudv. \tag{4.48}$$

Proof 4.6 *Let $\mathcal{D}$ be the set of Delaunay triangles, and $\Omega_{xy} = \boldsymbol{H}_v(\Omega)$ be the image region of the surface $\boldsymbol{S}(u,v)$ at the virtual camera. The* $\mathrm{MAE}_{\mathrm{PA}}$ *can be approximated as*

$$\begin{aligned}
\mathrm{MAE}_{\mathrm{PA}} &\approx \frac{1}{S_{\Omega_{xy}}} \int_{\Omega_{xy}} |e(x,y)| dxdy && (4.49)\\
&= \frac{1}{S_{\Omega_{xy}}} \sum_{\Delta_i \in \mathcal{D}} \int_{\Delta_i} |e(x,y)| dxdy \\
&\leq \frac{1}{S_{\Omega_{xy}}} \sum_{\Delta_i \in \mathcal{D}} S_{\Delta_i} \Big(\frac{1}{2} R_i^2 \|\nabla^2 f_v\|_\infty + \max\{|\varepsilon|\} \\
&\qquad + \max\{\|\boldsymbol{\mu}\|_2\} \cdot \|\nabla f_v\|_\infty \Big). && (4.50)
\end{aligned}$$

In each infinitesimal patch $d\omega$ around $(x,y) \in \Omega_{xy}$, we can approximate $R^2 \approx 2/(\pi\lambda(x,y))$ (see Lemma 4.2). Hence

$$\sum_{\Delta_i \in \mathcal{D}} S_{\Delta_i} R_i^2 \approx \int_{\Omega_{xy}} \frac{2}{\pi\lambda(x,y)} d\omega = \frac{2X_2}{\pi} \cdot \Delta_x \Delta_y. \tag{4.51}$$

By changing the variables from (x,y) to (u,v), and substituting (4.51) *into inequality* (4.50)*, we indeed get* (4.47)*.*

Remark 4.4 *The first term of* (4.47)*, $X_1 = S_{\Omega_{xy}}$, is the area of the scene's image on the virtual image plane and does not depend on the actual camera configuration. The value of X_2, called 3D multiple-view term of second order, encodes the geometrical information of the actual cameras and the scene. We note that X_2 decays with order N^{-1} when N tends to infinity. The first term also depends on the resolution $\Delta_x \Delta_y$. Overall, in smooth regions, the first term has decay order $\mathcal{O}(\lambda^{-1})$, where λ is the local density of actual samples. The second term is the intensity estimate error bound E_T. The third term relates to the depth estimate error bound E_D (linearly) and the geometrical position between the scene and the virtual camera (via B_v).*

Remark 4.5 *A notable difference between the 3D case and the 2D case resides in the decay order of the first term. In* (4.20)*, the first term has decay order $\mathcal{O}(\lambda^{-2})$, while in* (4.47) *the decay order is $\mathcal{O}(\lambda^{-1})$. To see this difference, note that the first term in inequality* (4.39) *contains R^2 having the same dimension with the sample density λ, whereas in* (4.13)*, the first term contains $(x_2 - x_1)^2$ of the same dimension with λ^2. This intriguing finding supports a common practice of conducting image rectifications to simplify the rendering process into 2D. Rectifying images using bilinear interpolation offers a decay of $\mathcal{O}(\lambda^{-2})$ in smooth regions, and $\mathcal{O}(\lambda^{-1})$ around the discontinuities. Hence, the image rectification not only reduces the complexity, but also increases the decay rate in smooth regions from $\mathcal{O}(\lambda^{-1})$ to $\mathcal{O}(\lambda^{-2})$. Obviously, one needs to take into account that image rectifications cause additional errors elsewhere.*

Table 4.1: Experimental values of $E[R^2], E[S]$, and $E[S^2]$ in the case where $N = 10$ actual cameras are used, compared to theoretical values of Poisson Delaunay triangles.

	$E[R^2]$	$E[S]$	$E[S^2]$
Experiments	0.0568	0.05	0.003845
Poisson Delaunay triangles	0.0637	0.05	0.004432
Relative error	11%	0%	13%

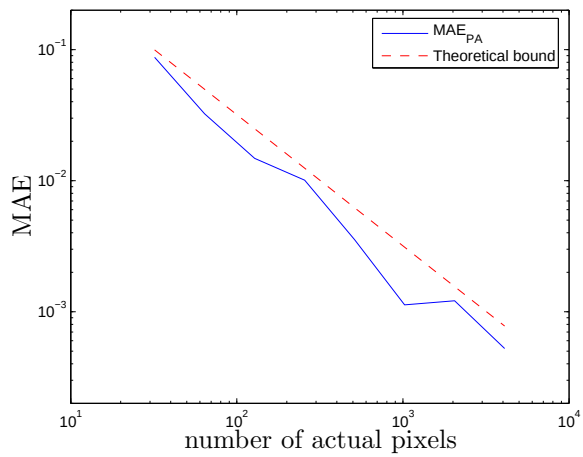


Figure 4.6: The rendering errors plotted against the total number of actual pixels. We note the errors indeed have decay $\mathcal{O}(\lambda^{-1})$, where λ is the local sample density, as stated in Theorem 4.3.

4.4.5 Numerical experiments

Support for a positive answer of Hypothesis 4.1 is shown in Table 4.1. Experimental values for R^2, S, and S^2 of Delaunay triangles, where $N = 10$ actual cameras are used, are computed and compared to the theoretical values of Poisson Delaunay triangles proposed in Lemma 4.2. Observe that the approximations are relatively accurate.

Next, we validate the error bound (4.47) of Theorem 4.3 for a 3D synthetic scene consisting of a flat surface with constant depth $z = 10$ and the texture map $T(u, v) = \sin(u) + \sin(v)$. The Propagation Algorithm [62] is used for a planar camera configuration. All the actual and virtual cameras are placed in the xy-plane and focus to the direction of the z-axis. Specifically, $N = 10$ actual cameras are randomly placed in the square of dimensions 2×2 centered around the virtual camera position at $[5, 5, 0]^T$.

To validate the first term, we set $E_D = E_T = 0$ and plot in Fig. 4.6 the mean absolute errors MAE (solid) and the error bound (dashed) against the total number of actual pixels (equivalent to the local density of actual samples λ). The variation of λ is obtained by changing the resolution $\Delta_x \Delta_y$. Observe that the MAE indeed decays as $\mathcal{O}(\lambda^{-1})$, conforming to Theorem 4.3.

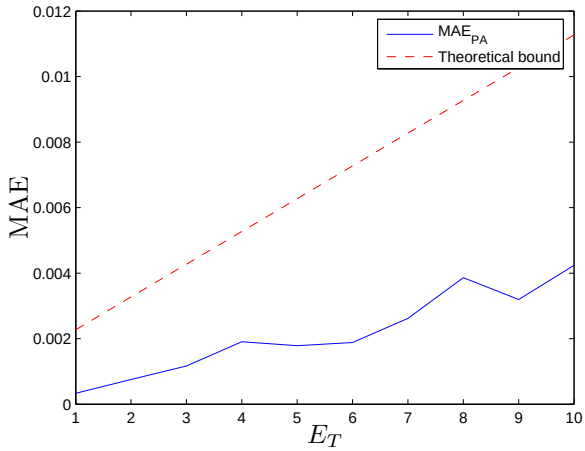


Figure 4.7: The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the intensity estimate error bound E_T.

To validate the second term, we fix $\Delta_x = \Delta_y = 0.02$ and $E_D = 0$, and vary E_T. For each value of E_T, the intensity estimate errors are chosen randomly in the interval $[-E_T, E_T]$ following the uniform distribution. In Fig. 4.7, we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the intensity estimate error bound E_T. Observe that the actual MAE fluctuates around one half of the error bound. The reason is that the error bound of Theorem 4.3 is derived for the worst case, whereas the actual MAE tends to follow the average errors.

Finally, we validate the last term of (4.47) by fixing $\Delta_x = \Delta_y = 0.02, E_T = 0$, and varying E_D. For each value of E_D, the depth estimate errors are chosen randomly in the interval $[-E_D, E_D]$ following the uniform distribution. In Fig. 4.8, we show the mean absolute error MAE (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D. Observe that the MAE indeed appears below the error bound and approximately linear to E_D.

4.5 Conclusion

We presented a quantitative analysis for IBR algorithms to 2D occluded scenes and 3D unoccluded scenes, extending the error aggregation framework proposed in the previous chapter. To analyze 2D occluded scenes, we modified the sample density function and measured the effects of jumps in observed sample intervals around the discontinuities. For 3D unoccluded scenes, we proposed an error bound for the technique of triangulation-based linear interpolation and exploited properties of Poisson Delaunay triangles. We derived an error bound for the mean absolute error (MAE) of the virtual images. The error bound successfully

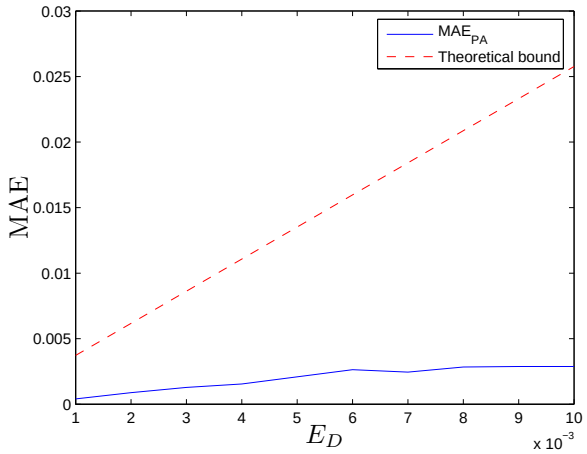


Figure 4.8: The mean absolute error (MAE) (solid) and the theoretical bound (dashed) plotted against the depth estimate error bound E_D.

captures the effects of the scene and the camera configuration to the rendering quality, as validated by numerical experiments. In particular, the proposed analysis suggests that the decay order of the MAE is $\mathcal{O}(\lambda^{-1})$ for 3D scenes and $\mathcal{O}(\lambda^{-2})$ for 2D scenes. An implication is that building IBR systems that can simplify to 2D, besides reducing the complexity, also increases the decay rate of the rendering errors from $\mathcal{O}(\lambda^{-1})$ to $\mathcal{O}(\lambda^{-2})$ in smooth regions.

Limitations. The proposed analysis approximates summations as integrals in Equations (4.49) and (4.51), and assembles actual samples as a generalized Poisson process. These approximations can be further analyzed, though it might not lead to further understanding in the context of IBR.

Future work. We would like to prove Hypothesis 4.1, extend the analysis to 3D occluded scenes, and analyze the mean and variance of the rendering errors.

CHAPTER 5

MINIMAX DESIGN OF HYBRID MULTIRATE FILTER BANKS WITH FRACTIONAL DELAYS

5.1 Introduction

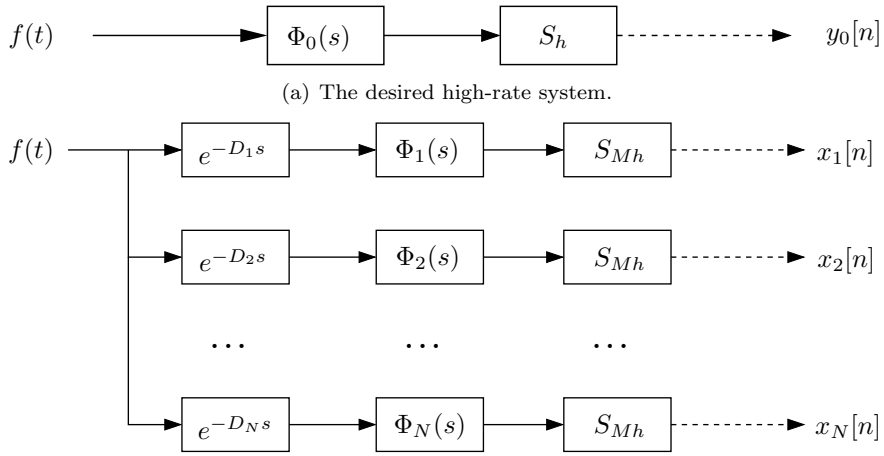
This chapter is motivated by multichannel sampling applications. Figure 5.1(a) shows the model of a fast analog-to-digital (A/D) converter used to obtain a *desired* high-resolution signal. An analog input signal $f(t)$ is convolved with an antialiasing filter $\phi_0(t)$ (also known as the sampling kernel function) whose Laplace transform is $\Phi_0(s)$. The output of the convolution is then sampled at small sampling interval h. The desired high-resolution signal is denoted by $y_0[n] = (f * \phi_0)(nh)$ for $n \in \mathbb{Z}$.

Figure 5.1(b) depicts *actual* low-resolution signals $\{x_i[n]\}_{i=1}^N$, sampled using slow A/D converters. The same analog input $f(t)$ is sampled in parallel using N slow A/D converters. In the i-th channel, for $1 \leq i \leq N$, the input $f(t)$ is first convolved with a function $\phi_i(t)$ (with Laplace transform $\Phi_i(s)$) before being delayed by $D_i > 0$ (to compensate for different time arrivals). The low-rate signals $x_i[n] = (f * \phi_i)(nMh - D_i)$, for $n \in \mathbb{Z}$, can be used to synthesize the high-resolution signal $y_0[n]$ of Fig. 5.1(a).

The goal of this chapter is to design the digital synthesis filter banks $\{F_i(z)\}_{i=1}^N$ to minimize the errors, defined using a criterion below, of a hybrid induced error system $\mathcal{K}$ shown in Fig. 5.2. Once the digital synthesis filters are designed, an approximate of the high-rate signal $y_0[n]$ can be computed, with a delay of m_0 samples, as the summation of N channels after the filtering process.

We assume that, through construction and calibration, information about the sampling kernel functions $\{\Phi_i(s)\}_{i=0}^N$ and delays $\{D_i\}_{i=1}^N$ are available. In such case, we want to design a corresponding optimal synthesis filter bank $\{F_i(z)\}_{i=1}^N$ so that the resulting system depicted in Fig. 5.2 can be subse-

[0]This chapter includes research conducted jointly with Prof. Minh Do [95, 96]. We thank Dr. Masaaki Nagahara (Kyoto University, Japan) for sharing the code of the paper [97], and Dr. Trac Tran (Johns Hopkins University, USA) and Dr. Geir Dullerud (University of Illinois at Urbana-Champaign) for insightful discussions.



(a) The desired high-rate system.

(b) The low-rate system.

Figure 5.1: (a) The desired high-rate system, (b) the low-rate system. The fast-sampled signal $y_0[n]$ can be approximated using slow-sampled signals $\{x_i[n]\}_{i=1}^N$.

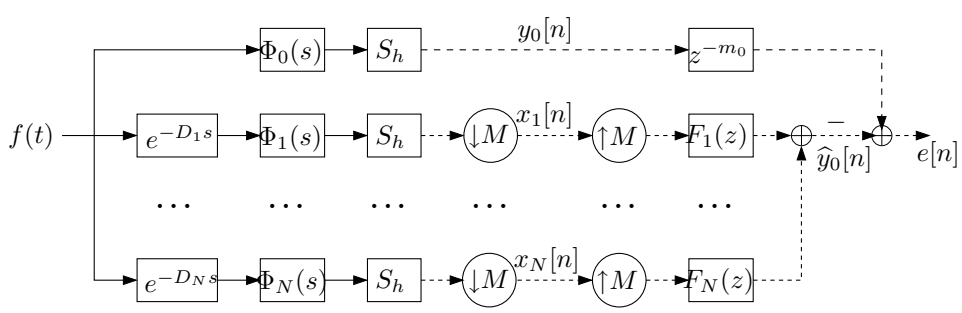


Figure 5.2: The hybrid induced error system $\mathcal{K}$ with analog input $f(t)$ and digital output $e[n]$. We want to design synthesis filters $\{F_i(z)\}_{i=1}^N$ based on the transfer function $\{\Phi_i(s)\}_{i=0}^N$, the fractional delays $\{D_i\}_{i=1}^N$, the system delay tolerance m_0, the sampling interval h, and the super-resolution rate M to minimize the $\mathcal{H}_\infty$ norm of the induced error system $\mathcal{K}$.

quently put in operation for arbitrary input signals $f(t)$. A special case of this multichannel sampling setup is called *time-interleaved A/D converters* where $\Phi_i(s) = \Phi_0(s)$ and $D_i = ih$ for $i = 1, 2, \ldots, N$. Then the synthesis filter bank can simply interleave samples, i.e. $F_i(z) = z^i$. Multichannel sampling extends time-interleaved A/D converters by allowing mismatch in sampling kernels before slow A/D converters [31]. Moreover, in many cases, the time delays $\{D_i\}_{i=1}^N$, although they can be measured [98, 99, 100, 101], cannot be controlled precisely. Under these conditions, the multichannel sampling setup studied in this chapter can be ideally applied.

We note that, in Fig. 5.2, system $\mathcal{K}$ is a hybrid system with analog input $f(t)$ and digital output $e[n]$. Among components of $\mathcal{K}$, the transfer functions $\{\Phi_i(s)\}_{i=0}^N$ characterize antialiasing filters, and $\{D_i\}_{i=1}^N$ model system setup such as arrival times or sampling positions. Many practical systems, such as electrical, mechanical, and electromechanical systems, can be modeled by differential equations [102, Chapter 1]. Their Laplace transforms are thus rational functions of form $A(s)/B(s)$ for some polynomials $A(s)$ and $B(s)$, while $e^{-D_i s}$ is not rational when D_i is fractional (i.e., noninteger). Working with delay operators $e^{-D_i s}$ is necessary, though nontrivial, to keep intersample behaviors of the input signals.

In the design of the synthesis filter banks $\{F_i(z)\}_{i=1}^N$, the system performances are evaluated using the $\mathcal{H}_\infty$ approach [103, 104, 105]. In the digital, we work on the Hardy space $\mathcal{H}_\infty$ that consists of all complex-value transfer matrices $\mathsf{G}(z)$ which are analytic and bounded outside of the unit circle $|z| > 1$. Hence $\mathcal{H}_\infty$ is the space of transfer matrices that are stable in the bounded-input bounded-output sense. The $\mathcal{H}_\infty$ norm of $\mathsf{G}(z)$ is defined as the maximum gain of the corresponding system. If a system G, analog or digital, has input $\boldsymbol{u}$ and output $\boldsymbol{y}$, the $\mathcal{H}_\infty$ norm of G is [103]

$$\|\mathsf{G}\|_\infty = \sup \left\{ \|\boldsymbol{y}\|_2 : \boldsymbol{y} = \mathsf{G}\boldsymbol{u}, \|\boldsymbol{u}\|_2 = 1 \right\}, \tag{5.1}$$

where the norms are regular Euclidean norm $\|\cdot\|$; that is,

$$\|\boldsymbol{x}\|_2 = \left(\sum_{n=-\infty}^{\infty} \|\boldsymbol{x}[n]\|^2 \right)^{1/2}$$

for digital signals $\boldsymbol{x}[n]$, and

$$\|\boldsymbol{x}\|_2 = \left(\int_{-\infty}^{\infty} \|\boldsymbol{x}(t)\|^2 dt \right)^{1/2}$$

for analog signals $\boldsymbol{x}(t)$.

The use of $\mathcal{H}_\infty$ optimization framework, originally proposed by Shenoy et al. [106] for filter bank designs, offers powerful tools for signal processing problems. In our case, using the $\mathcal{H}_\infty$ optimization framework, the induced error is

uniformly small over all finite energy inputs $f(t) \in \mathcal{L}_2(\mathbb{R})$ (i.e., $\|f(t)\|_2 < \infty$). Furthermore, no assumptions of $f(t)$, such as band-limitedness, are necessary. We minimize the worst induced error over all finite energy inputs $f(t)$. This is important since many practical signals are not bandlimited [107]. Finally, since $\mathcal{H}_\infty$ optimization is performed in the Hardy space, the designed filters are guaranteed to be stable.

The chapter's main contributions are twofold. First, we use sampled-data control techniques to convert the design problem for $\mathcal{K}$ into a $\mathcal{H}_\infty$ norm equivalent finite-dimensional model-matching problem. The conversion enables the design synthesis filters, IIR or FIR, to minimize the $\mathcal{H}_\infty$ norm of $\mathcal{K}$. The norm equivalence property reduces the induced errors compared to methods that approximate the fractional delays by IIR or FIR filters [108, 109, 110, 111]. IIR synthesis filters are designed using available solutions to the model-matching problem [103, 104]. To design FIR filters, we use linear matrix inequality (LMI) methods [112, 113]. Although FIR filter designs using LMI methods have been proposed for other problems [97, 114, 115], to our knowledge, only IIR filter designs are proposed for related problems [95, 99, 105, 116]. The second main contribution, shown in Section 5.5, is the robustness of the designed induced error system $\mathcal{K}$ against delay estimate errors.

Related work. Herley and Wong addressed the problem of the sampling and reconstruction of an analog signal from a periodic nonuniform set of samples assuming that the input signals have fixed frequency support [117]. Marziliano and Vetterli also addressed the problem of reconstructing a digital signal from a periodic nonuniform set of samples using Fourier transform [118]. However, in both cases, the authors only considered a restricted set of input signals that are bandlimited. Moreover, they only considered *rational* delays, that is, the set of samples is the set left after discarding a uniform set of samples in a periodic fashion (the ratio between the delays and the sample intervals is a rational number, hence the name rational delays). Jahromi and Aarabi considered the problem of estimating the delays $\{D_i\}_{i=1}^N$ and of designing analysis and synthesis filters to minimize the $\mathcal{H}_\infty$ norm of an induced error system [99]. However, the authors only considered integer delays or *approximation* of fractional delays by IIR or FIR filters. Shu et al. addressed the problem of designing the synthesis filters for a filter bank to minimize the $\mathcal{H}_\infty$ norm of an induced system [105]. Their problem was similar to the problem considered in this chapter, except that it did not consider the fractional delays but a rational transfer function instead. Nagahara et al. synthesized IIR and FIR filters to approximate fractional delays using $\mathcal{H}_\infty$ optimization [114, 115]. Although, strictly speaking, their problem is not SR, the result therein can be considered as a special case of our problem when $M = N = 1$.

Problem formulation. We consider the hybrid system $\mathcal{K}$ illustrated in Fig. 5.2.

The $\mathcal{H}_\infty$ norm of the system $\mathcal{K}$ is defined as

$$\|\mathcal{K}\|_\infty := \sup \left\{ \frac{\|e\|_2}{\|f\|_2} \right\}, \tag{5.2}$$

where $\|e\|_2$ is the l_2 norm of $e[n]$ and $\|f\|_2$ is the L_2 norm of $f(t)$.

We want to design (IIR or FIR) synthesis filters $\{F_i(z)\}_{i=1}^N$ to minimize $\|\mathcal{K}\|_\infty$. The inputs of our algorithms consist of the strictly proper transfer functions $\{\Phi_i(s)\}_{i=0}^N$, the positive fractional delays $\{D_i\}_{i=1}^N$, the system delay tolerance $m_0 \geq 0$, the sampling interval $h > 0$, and the upsampling-rate $M \geq 2$.

Throughout this chapter, we adopt the following conventions. A single-input single-output transfer function G is written in regular font, a multi-input and/or multi-output G is written in bold, and a hybrid system $\mathcal{G}$ is written in calligraphic font. We write scalars in regular font as x, and vectors in bold as $\boldsymbol{x}$. In our figures, solid lines illustrate analog signals, and dashed lines are intended for digital ones.

The remainder of this chapter is organized as follows. In Section 5.2, we show that the design problem is equivalent to a model-matching problem. Design procedures for IIR and FIR synthesis filters are presented in Section 5.3 and 5.4, respectively. Robustness of the designed system against delay estimates is presented in Section 5.5. We show experimental results in Section 5.6. Finally, we give conclusion and discussion in Section 5.7.

5.2 Equivalence of $\mathcal{K}$ to a Model-Matching Problem

In this section, we show that there exists a finite-dimensional digital linear time-invariant system K having the same $\mathcal{H}_\infty$ norm with $\mathcal{K}$. We demonstrate this in three steps. In Section 5.2.1, we convert $\mathcal{K}$ into an infinite-dimensional digital system. Next, in Section 5.2.2, we convert the system further into a finite-dimensional system K_d. Finally, in Section 5.2.3, we convert K_d into a linear time-invariant system.

5.2.1 Equivalence of $\mathcal{K}$ to a digital system

The idea is to show that the hybrid subsystem $\mathcal{G}$ (see Fig. 5.3) of $\mathcal{K}$ is $\mathcal{H}_\infty$ norm equivalent to a digital system. In Fig. 5.3, we denote $\{d_i\}_{i=1}^N$ the fractional parts of $\{D_i\}_{i=1}^N$. In other words, we have $0 \leq d_i < h$ and $m_i \in \mathbb{Z}$ such that

$$D_i = m_i h + d_i \quad (1 \leq i \leq N). \tag{5.3}$$

Note that by working with system $\mathcal{G}$, we need to compensate for the difference between $e^{-D_i s}$ and $e^{-d_i s}$. These differences are analog delay operators $e^{-m_i h s}$

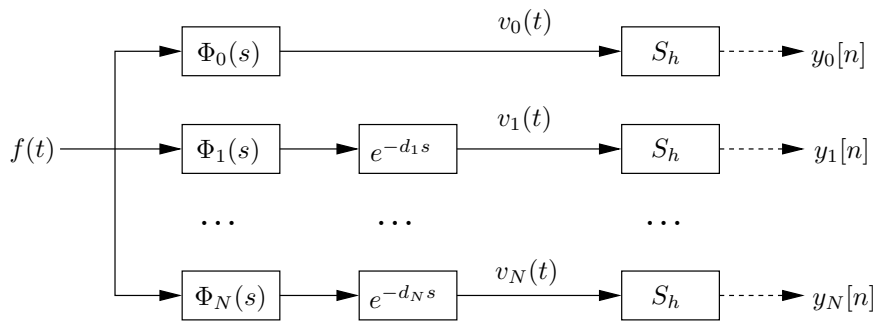


Figure 5.3: The hybrid (analog input digital output) subsystem $\mathcal{G}$ of $\mathcal{K}$. Note that the sampling interval of all channels is h.

that can be *interchanged* with the sampling operators S_h to produce digital integer delay operators z^{-m_i}.

To find a $\mathcal{H}_\infty$ norm equivalent digital system for $\mathcal{G}$, we adopt a divide-and-conquer approach: each channel of $\mathcal{G}$ will be shown to be $\mathcal{H}_\infty$ norm equivalent to a digital system. Since $\Phi_0(s)$ is strictly proper, there exist state-space matrices $\{A_0, B_0, C_0, 0\}$ and state function $\boldsymbol{x}_0(t)$ such that

$$\begin{cases} \dot{\boldsymbol{x}}_0(t) &= A_0\boldsymbol{x}_0(t) + B_0 f(t) \\ v_0(t) &= C_0\boldsymbol{x}_0(t). \end{cases}$$

For $0 \le t_1 < t_2 < \infty$, we can compute the future state value $\boldsymbol{x}_0(t_2)$ from a previous state value $\boldsymbol{x}_0(t_1)$ as follows:

$$\boldsymbol{x}_0(t_2) = e^{(t_2-t_1)A_0}\boldsymbol{x}_0(t_1) + \int_{t_1}^{t_2} e^{(t_2-\tau)A_0} B_0 f(\tau) d\tau. \tag{5.4}$$

Define linear operator Q_0 taking inputs $u(t) \in \mathcal{L}_2[0,h)$ as

$$\mathsf{Q}_0 u = \int_0^h e^{(h-\tau)A_0} B_0 u(\tau) d\tau.$$

Applying (5.4) using $t_1 = nh$ and $t_2 = (n+1)h$ we get

$$\boldsymbol{x}_0((n+1)h) = e^{hA_0}\boldsymbol{x}_0(nh) + \mathsf{Q}_0\widetilde{f}[n], \tag{5.5}$$

where $\widetilde{f}[n]$ denotes the portion of $f(t)$ on the interval $[nh, nh+h)$ translated to $[0,h)$. In other words, we consider the analog signal $f(t)$ as a sequence $\{\widetilde{f}[n]\}_{n\in\mathbb{Z}}$ with $\widetilde{f}[n] \in \mathcal{L}_2[0,h)$. The mapping from $f(t)$ into $\{\widetilde{f}[n]\}_{n\in\mathbb{Z}}$ is called the *lifting* operator [60, Section 10.1]. Clearly, the lifting operator preserves the energy of the signals, that is,

$$\|f(t)\|_2 = \|\widetilde{f}\|_2 = \left(\sum_{n=-\infty}^{\infty} \|\widetilde{f}[n]\|_2^2\right)^{1/2},$$

where $\|\widetilde{f}[n]\|_2^2 := \int_{nh}^{(n+1)h} |f(t)|^2 dt$.

Let $\mathcal{G}_0$ be the hybrid subsystem of $\mathcal{G}$ with input $f(t)$ and output $y_0[n]$ (see Fig. 5.3). An implication of (5.5) is that $y_0[n] = v_0(nh)$ can be considered as the output of a digital system with input $\widetilde{f}[n]$ and state $\boldsymbol{x}_{d0}[n] = \boldsymbol{x}_0(nh)$ as follows:

$$\begin{cases} \boldsymbol{x}_{d0}[n+1] &= e^{hA_0}\boldsymbol{x}_{d0}[n] + \mathsf{Q}_0\widetilde{f}[n] \\ y_0[n] &= C_0\boldsymbol{x}_{d0}[n]. \end{cases}$$

Since the lifting operator preserves the norm, system $\mathcal{G}_0$ is $\mathcal{H}_\infty$ norm equivalent to the system $\mathsf{G}_0 = \{e^{hA_0}, \mathsf{Q}_0, C_0, 0\}$.

The same technique can be used for the remaining channels. Let $\mathcal{G}_i$, for $1 \le i \le N$, be the hybrid subsystem of $\mathcal{G}$ with input $f(t)$ and output $y_i[n]$ (see Fig. 5.3). Suppose that $\{A_i, B_i, C_i, 0\}$ is a state-space realization of $\Phi_i(s)$ with state function $\boldsymbol{x}_i(t)$. We define linear operators Q_i and R_i taking inputs $u(t) \in \mathcal{L}_2[0,h)$ as

$$\mathsf{Q}_i u = \int_0^h e^{(h-\tau)A_i} B_i u(\tau) d\tau, \tag{5.6}$$

and

$$\mathsf{R}_i u = C_i \int_0^{h-d_i} e^{(h-d_i-\tau)A_i} B_i u(\tau) d\tau. \tag{5.7}$$

Similar to (5.5), we can obtain

$$\boldsymbol{x}_i((n+1)h) = e^{hA_i}\boldsymbol{x}_i(nh) + \mathsf{Q}_i\widetilde{f}[n]. \tag{5.8}$$

Applying (5.4) again with $t_1 = nh$ and $t_2 = (n+1)h - d_i$ we get

$$\begin{aligned} \boldsymbol{x}_i((n+1)h - d_i) &= e^{(h-d_i)A_i}\boldsymbol{x}_i(nh) + \\ &+ \int_{nh}^{(n+1)h-d_i} e^{((n+1)h-d_i-\tau)A_i} B_i f(\tau) d\tau. \end{aligned}$$

Since $v_i(t) = C_i\boldsymbol{x}_i(t-d_i)$ for all t, using $t = (n+1)h$ we obtain

$$v_i((n+1)h) = C_i e^{(h-d_i)A_i}\boldsymbol{x}_i(nh) + \mathsf{R}_i\widetilde{f}[n]. \tag{5.9}$$

From (5.8) and (5.9) we see that $y_i[n] = v_i(nh)$ can be considered as the output of a digital system with input $\widetilde{f}[n]$ and state $\boldsymbol{x}_{di}[n] = \begin{bmatrix} \boldsymbol{x}_i(nh) \\ v_i(nh) \end{bmatrix}$ as follows:

$$\begin{cases} \boldsymbol{x}_{di}[n+1] &= \underbrace{\begin{bmatrix} e^{hA_i} & 0 \\ C_i e^{(h-d_i)A_i} & 0 \end{bmatrix}}_{A_{di}} \boldsymbol{x}_{di}[n] + \underbrace{\begin{bmatrix} \mathsf{Q}_i \\ \mathsf{R}_i \end{bmatrix}}_{\mathsf{B}_i} \widetilde{f}[n] \\ y_i[n] &= \underbrace{[\mathbf{0}, 1]}_{C_{di}} \boldsymbol{x}_{di}[n]. \end{cases} \tag{5.10}$$

Since the lifting operator preserves the norm, system $\mathcal{G}_i$ is $\mathcal{H}_\infty$ norm equivalent

to the system $\mathsf{G}_i = \{A_{di}, \mathsf{B}_i, C_{di}, 0\}$.

Finally, we note that the system $\mathcal{G}$ is the vertical concatenation of subsystems $\{\mathcal{G}_i\}_{i=0}^N$. Since each subsystem $\mathcal{G}_i$ is $\mathcal{H}_\infty$ norm equivalent to the system G_i with the same input $\widetilde{f}[n]$, for $0 \leq i \leq N$, the system $\mathcal{G}$ is also $\mathcal{H}_\infty$ norm equivalent to the vertical concatenation system G of subsystems $\{\mathsf{G}_i\}_{i=0}^N$. We summarize the result of this Section in Proposition 5.1.

Proposition 5.1 *The system $\mathcal{G}$ is $\mathcal{H}_\infty$ norm equivalent to the infinite-dimensional digital system*

$$\mathsf{G} \Leftrightarrow \left[\begin{array}{c|c} A_d & \mathsf{B} \\ \hline C_d & 0 \end{array}\right], \tag{5.11}$$

where A_d, B, C_d are determined as

$$\left\{\begin{array}{rcl} A_d & = & \mathrm{diag}_{N+1}\left(e^{hA_0}, A_{d1}, \ldots, A_{dN}\right) \\ \mathsf{B} & = & [\mathsf{Q}_0^T \quad \mathsf{B}_1^T \quad \ldots \quad \mathsf{B}_N^T]^T \\ C_d & = & \mathrm{diag}_{N+1}(C_0, [\mathbf{0}, 1], \ldots, [\mathbf{0}, 1]). \end{array}\right. \tag{5.12}$$

In the above equations, and in the remainder of the chapter, we denote $\mathrm{diag}_k(\alpha_1, \alpha_2, \ldots, \alpha_k)$ the matrix with α_i in the diagonal, for $1 \leq i \leq k$, and 0 elsewhere, where $\{\alpha_i\}_{i=1}^k$ can be scalars, vectors, or matrices.

5.2.2 Equivalence of $\mathcal{K}$ to a finite-dimensional digital system

Proposition 5.1 shows that $\mathcal{G}$ is $\mathcal{H}_\infty$ norm equivalent to an infinite-dimensional digital system G. Next, we convert G further into some finite-dimensional digital system G_d.

Proposition 5.2 *Let B^* be the adjoint operator of B and B_d be a square matrix such that*

$$B_d B_d^T = \mathsf{B}\mathsf{B}^*. \tag{5.13}$$

The finite-dimensional digital system $\mathsf{G}_d(z) \Leftrightarrow \{A_d, B_d, C_d, 0\}$ has the same $\mathcal{H}_\infty$ norm with $\mathcal{G}$:

$$\|\mathsf{G}_d\|_\infty = \|\mathcal{G}\|_\infty. \tag{5.14}$$

Proof 5.1 *The product $\mathsf{B}\mathsf{B}^*$ is a linear operator characterized by a square matrix of finite dimension (the computation of $\mathsf{B}\mathsf{B}^*$ is given in Appendix). Hence $\mathsf{G}_d(z) \Leftrightarrow \{A_d, B_d, C_d, 0\}$ is a finite-dimensional digital system. The proof of* (5.14) *can be found in [60, Section 10.5].*

Proposition 5.2 claims that, for all analog signals $f(t)$, there exists a digital signal $\boldsymbol{u}[n]$ having the same energy as $f(t)$ such that $[y_0, \ldots, y_N]^T = \mathsf{G}_d \boldsymbol{u}$. The

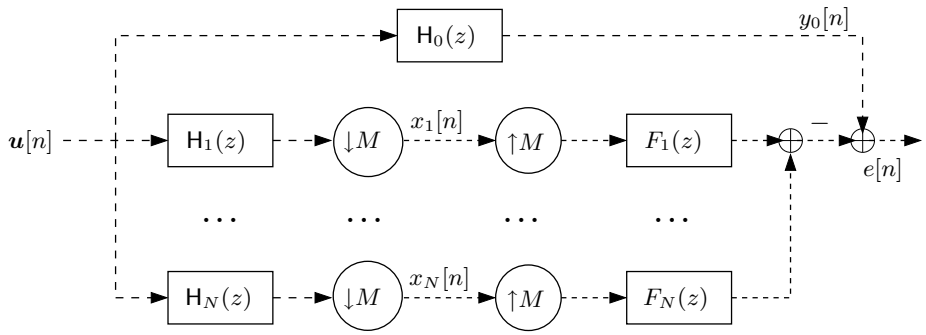


Figure 5.4: The $\mathcal{H}_\infty$ norm equivalent digital system K_d of $\mathcal{K}$ (see Proposition 5.3). Here $\{\mathsf{H}_i(z)\}_{i=0}^N$ are rational transfer functions defined in (5.16). Note that the input $\boldsymbol{u}[n]$ is of n_u dimension.

dimension n_u of $\boldsymbol{u}[n]$ is equal to the number of rows of A_d (see Proposition 5.1), i.e.,

$$n_u = n_0 + n_1 + \ldots + n_N + N, \tag{5.15}$$

where n_i is the number of rows of A_i, for $0 \leq i \leq N$. Since we want to minimize the worst induced error over all inputs $f(t)$ of finite energy, we also need to minimize $\|\mathsf{G}_d\|_\infty$ for all inputs $\boldsymbol{u}[n]$ (having the same energy with $f(t)$).

At this point, we take into account the integer delay operators $\{z^{-m_i}\}_{i=0}^N$ to obtain a digital system K_d that has the same $\mathcal{H}_\infty$ norm as $\mathcal{K}$.

Proposition 5.3 *Let C_{di} be the i-th row of the $(N+1)$-row matrix C_d (see Proposition 5.1), and $\mathsf{H}_i(z)$ be the multi-input single-output rational function that outputs $y_i[n]$ from input $\boldsymbol{u}[n]$, for $0 \leq i \leq N$. The system $\mathsf{H}_i(z)$ can be computed as*

$$\mathsf{H}_i(z) \Leftrightarrow z^{-m_i} \left[\begin{array}{c|c} A_d & B_d \\ \hline C_{di} & 0 \end{array}\right] \quad (0 \leq i \leq N). \tag{5.16}$$

As a result, system $\mathcal{K}$ is equivalent to the multiple-input one-output digital system $\mathsf{K}_d(z)$ illustrated in Fig. 5.4.

5.2.3 Equivalence of $\mathcal{K}$ to a linear time-invariant system

The finite-dimensional digital system K_d is not linear time-invariant (LTI) because of the presence of upsampling and downsampling operators $(\uparrow M), (\downarrow M)$. We apply polyphase techniques [61, 119] to make K_d an LTI system.

Let $\mathsf{H}_{0,j}(z)$, for $0 \leq j \leq M-1$, be the polyphase components of filter $\mathsf{H}_0(z)$. In other words,

$$\mathsf{H}_0(z) = \sum_{j=0}^{M-1} z^j \mathsf{H}_{0,j}(z^M). \tag{5.17}$$

We also denote $\boldsymbol{u}_p[n]$ and $\boldsymbol{e}_p[n]$ the polyphase versions of $\boldsymbol{u}[n]$ and $e[n]$. Note that $\|\boldsymbol{u}_p\|_2 = \|\boldsymbol{u}\|_2$ and $\|\boldsymbol{e}_p\|_2 = \|e\|_2$. Hence, by working in the polyphase

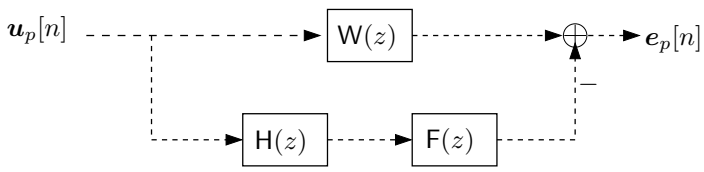


Figure 5.5: The equivalent LTI error system $\mathsf{K}(z)$ (see Theorem 5.1). Note that the system $\mathsf{K}(z)$ is Mn_u input M output, the transfer matrices $\mathsf{W}(z), \mathsf{H}(z)$ are of dimension $M \times Mn_u$, and $\mathsf{F}(z)$ is of dimension $M \times M$.

domain, $\mathsf{K}_d(z)$ is converted into an LTI system with the same $\mathcal{H}_\infty$ norm.

Proposition 5.4 *The digital error system $\mathsf{K}_d(z)$ is $\mathcal{H}_\infty$ norm equivalent to the LTI system*

$$\mathsf{K}(z) = \mathsf{W}(z) - \mathsf{H}(z)\mathsf{F}(z) \tag{5.18}$$

with input $\boldsymbol{u}_p[n]$ and output $\boldsymbol{e}_p[n]$. In (5.18), $\mathsf{H}(z)$ and $\mathsf{F}(z)$ are standard polyphase matrices of $\{\boldsymbol{H}_i(z)\}_{i=1}^N$ and $\{\boldsymbol{F}_i(z)\}_{i=1}^N$, and

$$(\mathsf{W}(z))_{i,j} = \begin{cases} \mathsf{H}_{0,j-i}(z) & \text{if } 1 \le i \le j \le M \\ z\mathsf{H}_{0,M+j-i}(z) & \text{if } 1 \le j < i \le M. \end{cases} \tag{5.19}$$

Proof 5.2 *The proof uses standard polyphase techniques [61, 119], hence omitted here.*

Figure 5.5 shows the equivalent digital, LTI error system $\mathsf{K}(z)$. The transfer function matrix $\mathsf{F}(z)$ is to be designed. State-space realizations of $\mathsf{H}(z)$ and $\mathsf{W}(z)$ are given in Theorem 5.1 using state-space realizations $\{A_{\mathsf{H}_i}, B_{\mathsf{H}_i}, C_{\mathsf{H}_i}, 0\}$ of $\{\mathsf{H}_i(z)\}_{i=0}^N$ (it can be easily verified that the D-matrix of $\mathsf{H}_i(z)$ is a zero-matrix).

Theorem 5.1 *The original induced error system $\mathcal{K}$ has an $\mathcal{H}_\infty$ norm equivalent digital, LTI system $\mathsf{K}(z) = \mathsf{W}(z) - \mathsf{F}(z)\mathsf{H}(z)$ (see Fig. 5.5); that is,*

$$\|\mathcal{K}\|_\infty = \|\mathsf{W}(z) - \mathsf{F}(z)\mathsf{H}(z)\|_\infty, \tag{5.20}$$

where $\mathsf{F}(z)$ is the polyphase matrix of $\{F_i(z)\}_{i=1}^N$ to be designed. State-space realizations of $\mathsf{W}(z)$ and $\mathsf{H}(z)$ can be computed as follows:

$$\begin{aligned} A_W &= A_{\boldsymbol{H}_0}^M \\ B_W &= [A_{\boldsymbol{H}_0}^{M-1} B_{\boldsymbol{H}_0},\ A_{\boldsymbol{H}_0}^{M-2} B_{\boldsymbol{H}_0}, \ldots,\ B_{\boldsymbol{H}_0}] \\ C_W &= [(C_{\boldsymbol{H}_0})^T,\ (C_{\boldsymbol{H}_0} A_{\boldsymbol{H}_0})^T, \ldots,\ (C_{\boldsymbol{H}_0} A_{\boldsymbol{H}_0}^{M-1})^T]^T \\ (D_W)_{ij} &= \begin{cases} C_{\boldsymbol{H}_0} A_{\boldsymbol{H}_0}^{i-j-1} B_{\boldsymbol{H}_0} & \text{if } 1 \le j < i \le M, \\ 0 & \text{else} \end{cases} . \end{aligned} \tag{5.21}$$

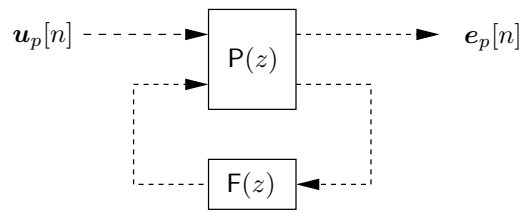


Figure 5.6: The induced error system $\mathsf{K}(z)$ of the form of the standard problem in $\mathcal{H}_\infty$ control theory with input $\boldsymbol{u}_p[n]$ output $\boldsymbol{e}_p[n]$. We want to design synthesis system $\mathsf{F}(z)$ to minimize $\|\mathsf{K}\|_\infty$.

and

$$\begin{aligned} A_H &= \mathrm{diag}_N(A^M_{\boldsymbol{H}_1}, \ldots, A^M_{\boldsymbol{H}_1}) \\ (B_H)_{ij} &= A^{M-j}_{\boldsymbol{H}_i} B_{\boldsymbol{H}_i}, \quad \text{for } 1 \le i \le N, 1 \le j \le M \\ C_H &= \mathrm{diag}_N(C_{\boldsymbol{H}_1}, \ldots, C_{\boldsymbol{H}_N}) \\ D_H &= 0. \end{aligned} \tag{5.22}$$

Proof 5.3 *We give here the proof for* (5.21)*. The proof for Eq.* (5.22) *can be derived similarly. Consider the transfer function* $\mathsf{H}_{00}(z)$ *in the block* $(1,1)$ *of* $\mathsf{W}(z)$ *(see Proposition 5.4):*

$$\begin{aligned} \mathsf{H}_{00}(z) &= \sum_{i=1}^{\infty} C_{\mathsf{H}_0} A^{iM-1}_{\mathsf{H}_0} B_{\mathsf{H}_0} z^{-i} \\ &= \sum_{i=1}^{\infty} C_{\mathsf{H}_0} \left(A^M_{\mathsf{H}_0}\right)^{i-1} \left(A^{M-1}_{\mathsf{H}_0} B_{\mathsf{H}_0}\right) z^{-i} \\ &\Leftrightarrow \left[\begin{array}{c|c} A^M_{\mathsf{H}_0} & A^{M-1}_{\mathsf{H}_0} B_{\mathsf{H}_0} \\ \hline C_{\mathsf{H}_0} & 0 \end{array}\right]. \end{aligned}$$

The state-space representation of the block $(1,1)$ *of* $\mathsf{W}(z)$ *is in agreement with* (5.21)*. The same technique can be applied for the remaining blocks.*

5.3 Design of IIR Filters

5.3.1 Conversion to the standard $\mathcal{H}_\infty$ control problem

The problem of designing $\mathsf{F}(z)$ to minimize $\|\mathsf{K}\|_\infty$ (see Fig. 5.5) has a similar form to the model-matching form which is a special case of the *standard problem* in $\mathcal{H}_\infty$ control theory [103, 104]. Figure 5.6 shows the system $\mathsf{K}(z)$ in the standard form. The system $\mathsf{P}(z)$ of Fig. 5.6 has a state-space realization derived from

ones of $\mathsf{W}(z)$ and $\mathsf{H}(z)$ as

$$\left[\begin{array}{c|c} A_P & B_P \\ \hline C_P & D_P \end{array}\right] = \left[\begin{array}{cc|cc} A_W & 0 & B_W & 0 \\ 0 & A_H & B_H & 0 \\ \hline C_W & 0 & D_W & -I \\ 0 & C_H & D_H & 0 \end{array}\right]. \tag{5.23}$$

Solutions to the standard problem have existing software, such as MATLAB's Robust Control Toolbox [120], to facilitate the optimization procedures.

5.3.2 Design procedure

- **Inputs:** Rational transfer functions $\{\Phi_i(s)\}_{i=0}^N$ (strictly proper), positive fractional delays $\{D_i\}_{i=1}^N$, the system tolerance delay $m_0 \geq 0$, the sampling interval $h > 0$, the superresolution rate $M \geq 2$.
- **Outputs:** Synthesis IIR filters $\{F_i(z)\}_{i=1}^N$.

1. Let $D_i = m_i h + d_i$ for $1 \leq i \leq N$ as in (5.3).
2. Compute a state-space realization $\{A_i, B_i, C_i, 0\}$ of $\Phi_i(s)$ for $0 \leq i \leq N$.
3. Compute the system $\mathsf{G}_d = \{A_d, B_d, C_d, 0\}$ as in Proposition 5.1 and 5.2.
4. Compute a state-space realization of $\mathsf{H}_i(z)$, for $0 \leq i \leq N$, as in Proposition 5.3.
5. Compute the state-space realization of $\mathsf{W}(z)$ and $\mathsf{H}(z)$ as in (5.21) and in (5.22) of Theorem 5.1.
6. Compute the state-space realization of $\mathsf{P}(z)$ from $\mathsf{H}(z)$ and $\mathsf{W}(z)$ as in (5.23).
7. Design a synthesis system $\mathsf{F}(z)$ using existing $\mathcal{H}_\infty$ optimization tools.
8. Obtain $\{F_i(z)\}_{i=1}^N$ from $\mathsf{F}(z)$ by

$$[F_1(z) \quad F_2(z) \ldots F_N(z)] = [1 \quad z^{-1} \ldots z^{-M+1}]\mathsf{F}(z^M).$$

5.4 Design of FIR Filters

5.4.1 Conversion to a linear matrix inequality problem

In this section, we present a design procedure to synthesize FIR filters $\{F_i(z)\}_{i=1}^N$. For some practical applications, FIR filters are preferred to IIR filters for their robustness to noise and computational advantages.

We first derive a state-space realization $\{A_F, B_F, C_F, D_F\}$ of the polyphase matrix $\mathsf{F}(z)$ of $\{F_i(z)\}_{i=1}^N$ based on the coefficients of $\{F_i(z)\}_{i=1}^N$. Assuming

that the synthesis FIR filters $\{F_i(z)\}_{i=1}^N$ are of maximum length $nM > 0$, for $1 \le i \le N$, we denote

$$F_i(z) = d_{i0} + d_{i1}z^{-1} + d_{i2}z^{-2} + \ldots + d_{i,nM-1}z^{-nM+1},$$

and

$$C_{ij} = [d_{i,j+M} \quad d_{i,j+2M} \quad \ldots \quad d_{i,j+(n-1)M}].$$

The polyphase system $\mathsf{F}(z)$ of $\{F_i(z)\}_{i=1}^N$ has a state-space realization $\{A_F, B_F, C_F, D_F\}$ as

$$\begin{cases} A_F & = & \mathrm{diag}_M(A_n, \ldots, A_n) \\ B_F & = & \mathrm{diag}_M(B_n, \ldots, B_n) \\ (C_F)_{ij} & = & C_{ji} \qquad (0 \le i \le M-1 \text{ and } 1 \le j \le N) \\ (D_F)_{ij} & = & d_{ji} \qquad (0 \le i \le M-1 \text{ and } 1 \le j \le N), \end{cases} \tag{5.24}$$

where matrix $A_n \in \mathbb{R}^{n\times n}$ and vector $B_n \in \mathbb{R}^n$ are

$$A_n = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \quad B_n = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Note that, given the number n, the matrices A_F, B_F do not depend on $\{F_i(z)\}_{i=1}^N$. Hence, designing $\{F_i(z)\}_{i=1}^N$ is equivalent to finding the matrices C_F, D_F to minimize $\mathsf{K}(z)$. The system $\mathsf{K}(z)$ has a state-space realization $\{A_K, B_K, C_K, D_K\}$ as follows:

$$\mathsf{K} \Leftrightarrow \left[\begin{array}{ccc|c} A_W & 0 & 0 & B_W \\ 0 & A_H & 0 & B_H \\ 0 & B_F C_H & A_F & B_F D_H \\ \hline C_W & -D_F C_H & -C_F & D_W - D_F D_H \end{array}\right]. \tag{5.25}$$

We observe that the state-space matrices of $\mathsf{K}(z)$ depend on C_F, D_F in a linear fashion. Hence we can use the linear matrix inequalities (LMI) [113, 121] techniques to solve for the matrices C_F, D_F.

Proposition 5.5 *[115, 121] For a given $\gamma > 0$, the system $\mathsf{K}(z)$ satisfies $\|\mathsf{K}\|_\infty < \gamma$ if and only if there exists a positive definite matrix $P > 0$ such that*

$$\begin{bmatrix} A_K^T P A_K - P & A_K^T P B_K & C_K^T \\ B_K^T P A_K & B_K^T P B_K - \gamma I & D_K^T \\ C_K & D_K & -\gamma I \end{bmatrix} < 0. \tag{5.26}$$

For any $\gamma > 0$, Proposition 5.5 provides us with a tool to test if $\|\mathsf{K}\|_\infty < \gamma$. Hence, we can iteratively decrease γ until we get close to the optimal perfor-

mance (within a predefined performance tolerance). Available implementations such as MATLAB's *LMI Control Toolbox* [122] can facilitate the design procedure.

5.4.2 Design procedure

- **Inputs:** Rational transfer functions $\{\Phi_i(s)\}_{i=0}^N$ (strictly proper), positive fractional delays $\{D_i\}_{i=1}^N$, the system tolerance delay $m_0 \geq 0$, the sampling interval $h > 0$, the superresolution rate $M \geq 2$.
- **Outputs:** Synthesis FIR filters $\{F_i(z)\}_{i=1}^N$.

1. Let $D_i = m_i h + d_i$, for $1 \leq i \leq N$, as in (5.3).
2. Compute a state-space realization $\{A_i, B_i, C_i, 0\}$ of $\Phi_i(s)$ for $0 \leq i \leq N$.
3. Compute the system $\mathsf{G}_d = \{A_d, B_d, C_d, 0\}$ as in Proposition 5.1 and 5.2.
4. Compute a state-space realization of $\mathsf{H}_i(z)$, for $0 \leq i \leq N$, as in Proposition 5.3.
5. Compute the state-space realization of $\mathsf{W}(z)$ and $\mathsf{H}(z)$ as in (5.21) and in (5.22) of Theorem 5.1.
6. Design synthesis filter $\{F_i(z)\}_{i=1}^N$ using Proposition 5.5.
7. Obtain $\{F_i(z)\}_{i=1}^N$ from $\mathsf{F}(z)$ by

$$[F_1(z) \quad F_2(z) \ldots F_N(z)] = [1 \quad z^{-1} \ldots z^{-M+1}]\mathsf{F}(z^M).$$

5.5 Robustness against Delay Uncertainties

The proposed design procedures for synthesis filters assume perfect knowledge of the delays $\{D_i\}_{i=1}^N$. In this section, we show that the induced error system $\mathcal{K}$ obtains nearly optimal performance if the synthesis filters are designed using estimates $\{\widehat{D}_i\}_{i=1}^N$ sufficiently close to the actual delays $\{D_i\}_{i=1}^N$.

We denote $\{\delta_i\}_{i=1}^N$ the delay jitters

$$\delta_i = D_i - \widehat{D}_i, \quad i = 1, 2, \ldots, N, \tag{5.27}$$

and $\overline{\delta}$ be the maximum jitter

$$\overline{\delta} = \max_{i=1}^{N}\{|\delta_i|\}. \tag{5.28}$$

For convenience, we also define operators

$$\Delta(s) = \mathrm{diag}_N(e^{-\delta_1 s}, \ldots, e^{-\delta_N s}), \tag{5.29}$$

$$\Phi(s) = \mathrm{diag}_N(\Phi_1(s), \ldots, \Phi_N(s)). \tag{5.30}$$

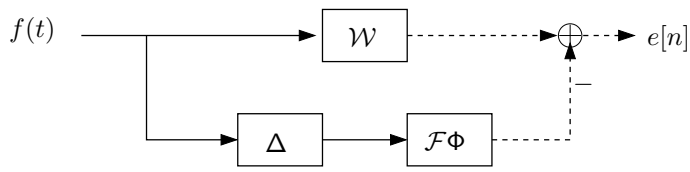


Figure 5.7: The hybrid system $\mathcal{K}$ and the uncertainty operator Δ caused by delay estimate errors.

The induced error system $\mathcal{K}$, see Fig. 5.2, can be rewritten as in Fig. 5.7, where $\mathcal{W}$ represents the high-resolution channel of $\mathcal{K}$, and $\mathcal{F}$ signifies the hybrid MIMO system composed of the delay operators $\{e^{-\widehat{D}_i s}\}_{i=1}^N$, the sampling operators S_{Mh}, the synthesis filters $\{F_i(z)\}_{i=1}^N$, and the summation of all the low resolution channels. The uncertainty operator Δ only affects the low-resolution channels:

$$\mathcal{K} = \mathcal{W} - \mathcal{F}\Phi\Delta. \tag{5.31}$$

It is easy to see that all these operators have bounded $\mathcal{H}_\infty$ norm. Let $\overline{\omega} \in \mathbb{R}^+$ be an arbitrary, but fixed, positive number. The following Lemma gives a bound for the singular values of $\mathsf{I} - \Delta(j\omega)$ and $\Phi(j\omega)$ for each frequency ω.

Lemma 5.1 *The maximum singular value of* $\mathsf{I} - \Delta(j\omega)$ *and* $\Phi(j\omega)$ *can be bounded as*

$$\sigma_{\max}[\mathsf{I} - \Delta(j\omega)] \leq \sqrt{2\overline{\delta}|\omega|}, \tag{5.32}$$

$$\begin{cases} \sigma_{\max}[\Phi(j\omega)] & \leq & C_\Phi/\sqrt{|\omega|} & \text{if } |\omega| > \overline{\omega} \\ \sigma_{\max}[\Phi(j\omega)] & \leq & C_\Phi & \text{if } |\omega| \leq \overline{\omega}, \end{cases} \tag{5.33}$$

where C_Φ *is a constant depending on* $\overline{\omega}$ *and* $\{\Phi_i\}_{i=1}^N$.

Proof 5.4 *To show* (5.32)*, observe that the operator*

$$\big(\mathsf{I} - \Delta(j\omega)\big) \cdot \big(\mathsf{I} - \Delta^*(j\omega)\big) \tag{5.34}$$

is a matrix with $2 - 2\cos(\delta_i\omega)$*, for* $i = 1, 2, \ldots, N$*, in the diagonal and zeros elsewhere. Using*

$$1 - \cos(x) \leq |x|, \quad x \in \mathbb{R}, \tag{5.35}$$

that can be easily verified, we indeed prove (5.32)*.*

To show (5.33)*, it is sufficient to note that* $\Phi(j\omega)$ *is a diagonal operator with strictly proper rational functions in the diagonal. Its maximum singular values hence decay at least as fast as* $\mathcal{O}(|\omega|^{-1})$ *when* $|\omega| > \overline{\omega}$*, and are bounded when* $|\omega| \leq \overline{\omega}$*, which in fact implies* (5.33)*.*

We use the result of Lemma 5.1 to derive the bound for the composite operator $\Phi - \Phi\Delta$ based on $\overline{\delta}$.

Proposition 5.6 *The following inequality holds:*

$$\|\Phi - \Phi\Delta\|_\infty \leq \overline{C}\sqrt{\overline{\delta}}, \tag{5.36}$$

for some $\overline{C} > 0$.

Proof 5.5 *We denote* $\boldsymbol{u}(t)$ *the output of* Φ *for the input* $f(t)$*, denote* $\boldsymbol{g}(t)$ *the output of* $(\mathsf{I} - \Delta)$ *for the input* $\boldsymbol{u}(t)$*. Hence:*

$$\begin{aligned} \|\boldsymbol{G}(j\omega)\|_2 &= \|\big(\mathsf{I} - \Delta(j\omega)\big)\Phi(j\omega)F(j\omega)\|_2 \\ &\leq \sigma_{\max}[\mathsf{I} - \Delta(j\omega)] \cdot \sigma_{\max}[\Phi(j\omega)] \cdot \|F(j\omega)\|_2. \end{aligned}$$

Using the result of Lemma 5.1 for $\omega > \overline{\omega}$ *we derive*

$$\begin{aligned} \int_{|\omega|>\overline{\omega}} \|\boldsymbol{G}(j\omega)\|_2^2 d\omega &\leq \int_{|\omega|>\overline{\omega}} 2\overline{\delta}|\omega| \cdot \frac{C_\Phi^2}{|\omega|} \cdot \|F(j\omega)\|_2^2 d\omega \\ &\leq 2\overline{\delta}C_\Phi^2 \cdot \|f\|_2^2. \end{aligned} \tag{5.37}$$

Similarly, for $\omega \leq \overline{\omega}$*, we can obtain*

$$\begin{aligned} \int_{|\omega|\leq\overline{\omega}} \|\boldsymbol{G}(j\omega)\|_2^2 d\omega &\leq \int_{|\omega|\leq\overline{\omega}} 2\overline{\delta}|\omega| \cdot C_\Phi^2 \cdot \|F(j\omega)\|_2^2 d\omega \\ &\leq 2\overline{\delta}C_\Phi^2\overline{\omega} \cdot \|f\|_2^2. \end{aligned} \tag{5.38}$$

From (5.37) *and* (5.38) *we can easily obtain*

$$\|\boldsymbol{g}\|_2 \leq \overline{C} \cdot \|f\|_2, \tag{5.39}$$

for

$$\overline{C} = C_\Phi\sqrt{2(\overline{\omega}+1)}. \tag{5.40}$$

Equation (5.39) *indeed implies* (5.36).

The following theorem shows the robustness of the induced error system $\mathcal{K}$ against the delay jitters $\{\delta_i\}_{i=1}^N$.

Theorem 5.2 *In the presence of delay estimate errors, the induced error system* $\mathcal{K}$ *is robust in the sense that its* $\mathcal{H}_\infty$ *norm is bounded as*

$$\|\mathcal{K}\|_\infty \leq \|\mathcal{W} - \mathcal{F}\Phi\|_\infty + \sqrt{\overline{\delta}} \cdot \overline{C} \cdot \|\mathcal{F}\|_\infty, \tag{5.41}$$

where $\overline{\delta}$ *is the maximum jitters and* C_Φ *is defined as in* (5.33).

Proof 5.6 *Indeed:*

$$\begin{aligned} \|\mathcal{K}\|_\infty &= \|\mathcal{W} - \mathcal{F}\Phi\Delta\|_\infty \\ &\leq \|\mathcal{W} - \mathcal{F}\Phi\|_\infty + \|\mathcal{F}\Phi - \mathcal{F}\Phi\Delta\|_\infty \\ &\leq \|\mathcal{W} - \mathcal{F}\Phi\|_\infty + \sqrt{\overline{\delta}} \cdot \overline{C} \cdot \|\mathcal{F}\|_\infty. \end{aligned}$$

Hence, the induced error system $\mathcal{K}$ is robust against the delay estimate errors $\{\delta_i\}_{i=1}^N$. In fact, its performance is degraded from the design performance $\|\mathcal{W} - \mathcal{F}\Phi\|_\infty$, in the worst case, by an amount of order $\mathcal{O}(\sqrt{\overline{\delta}})$.

5.6 Experimental Results

We present in Section 5.6.1 and 5.6.2 examples of IIR and FIR filter design. In Section 5.6.3, we compare the performance of proposed method to existing methods.

5.6.1 Example of IIR filter design

We design IIR synthesis filters for the following setting:

- We use two channels to double the resolution, that is, $M = N = 2$.
- All transfer functions $\Phi_i(s) = \Phi(s)$, for $0 \leq i \leq 2$, where $\Phi(s)$ is the Chebyshev type-2 filter of order 6 with stopband attenuation 20 dB and with stopband edge frequency of 300 Hz (for the data sampled at 1000 Hz). The MATLAB command to design $\Phi(s)$ is `cheby2(6,30,300/500)`. (We normalize $\Phi(s)$ so that it has unit gain.) The Bode diagram of the transfer function $\Phi(s)$ is plotted in Fig. 5.8.
- The input $f(t)$ is a step function having energy at all frequencies:

$$f(t) = \begin{cases} 0, & \text{if } t < \tau \\ 1, & \text{if } t \geq \tau. \end{cases}$$

- $m = 10, h = 1, D_1 = 1.2, D_2 = 0.6$.

In Fig. 5.9, we show the magnitude and phase response of synthesized filters $F_1(z)$ (dashed) and $F_2(z)$ (solid). The orders of $F_1(z), F_2(z)$ are 28 in this case. It is interesting to note that the synthesized filters are nearly linear phase.

In Fig. 5.10, we plot the error $e[n]$ (solid) against the desired output $y_0[n]$ (dashed). We can see that the approximation error is very small compared to the desired signal. The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4.68\%$. Note that the system is designed without any assumption on the input signals.

5.6.2 Example of FIR filter design

The experimental setting is as follows:

- We use two channels to double the resolution; that is, $M = N = 2$.

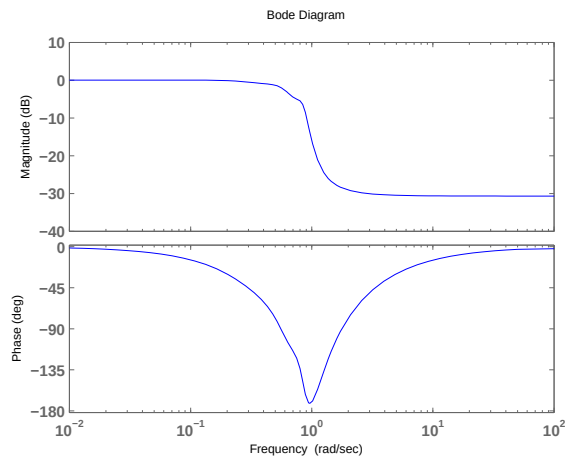


Figure 5.8: Example of IIR filter design. The magnitude and phase response of the transfer function $\Phi(s)$ modeling the measurement device. We use $\Phi_i(s) = \Phi(s)$ for $i = 0, 1, 2$.

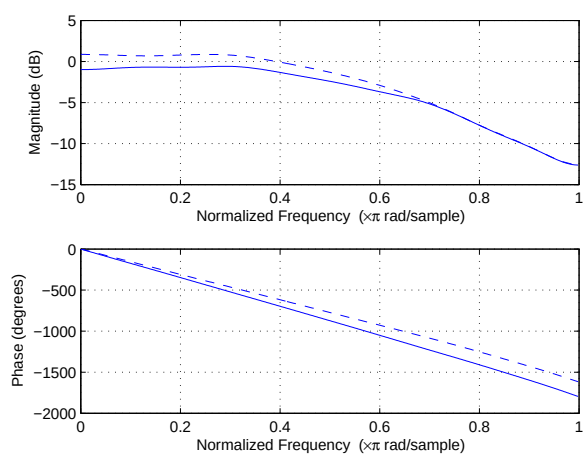


Figure 5.9: Example of IIR filter design. The magnitude and phase response of synthesized IIR filters $F_1(z)$ (dashed), and $F_2(z)$ (solid) designed using the proposed method. The order of $F_1(z)$ and of $F_2(z)$ are 28.

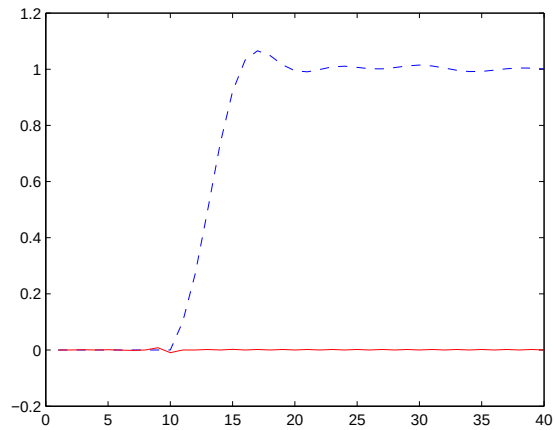


Figure 5.10: Example of IIR filter design. The error $e[n]$ (solid) plotted against the desired output $y_0[n]$ (dashed). The induced error is small compared to the desired signal. The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4.68\%$.

- All functions $\Phi_i(s) = \omega_c^2/(s+\omega_c)^2$ for $\omega_c = 0.5$ and $i = 0, 1, 2$. Fig. 5.11 plots the Bode diagram of the transfer function $\Phi_i(s)$.
- Input signal is a step function:

$$f(t) = \begin{cases} 0 & t < 0.3 \\ 1 & t \geq 0.3. \end{cases} \tag{5.42}$$

- $m = 10, h = 1, D_1 = 1.2, D_2 = 0.6$.
- Maximum filter length is $nM = 22$.

Figure 5.12 shows the equivalent filters $\mathsf{H}_0(z)$ of the first channel. Note that $\mathsf{H}_i(z)$, for $i = 0, 1, 2$, take multiple inputs (in this case $n_u = 4$ inputs, hence 4 filters for each $\mathsf{H}_i(z)$ are required). The magnitude and phase response of the designed filters $F_1(z), F_2(z)$ are shown in Fig. 5.13. In Fig. 5.14, we show the error $e[n]$ of the induced system (solid) and the desired output $y_0[n]$ (dashed). The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4\%$. Observe that the induced error $e[n]$ is small compared to the desired signal $y_0[n]$.

We also test the robustness of $\mathcal{K}$ against jitters $\{\delta_i\}_{i=1,2}$. The synthesis filters are designed for $\widehat{D}_1 = 1.2h$ and $\widehat{D}_2 = 0.6h$, but the system uses inputs produced with jittered time delays D_1, D_2. Figure 5.15 shows the $\mathcal{H}_\infty$ norm of the induced errors plotted against jitters in δ_1 (solid) and δ_2 (dashed). The errors are observed to be robust against delay estimate errors.

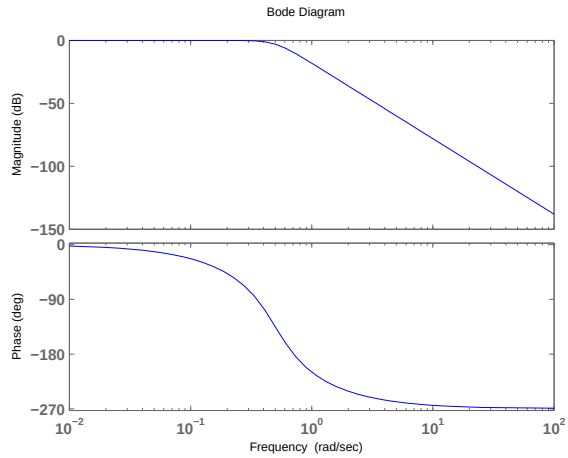


Figure 5.11: The magnitude and phase response of the transfer function $\Phi(s)$ modeling the measurement devices. We use $\Phi_i(s) = \Phi(s)$ for $i = 0, 1, 2$.

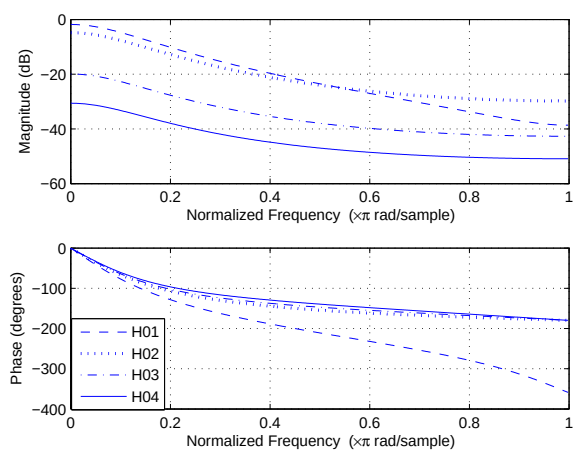


Figure 5.12: The equivalent analysis filters $\mathsf{H}_0(z)$ of the first channel. Since $\mathsf{H}_0(z)$ takes multiple inputs, in this case $n_u = 4$ inputs, the i-th input is passed through filter $\mathsf{H}_{0i}(z)$ for $1 \leq i \leq 4$.

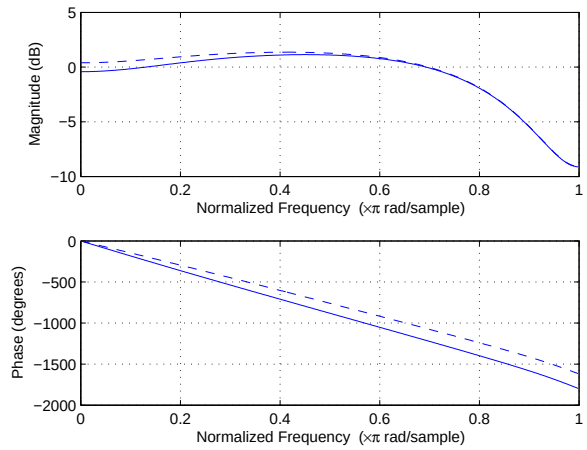


Figure 5.13: The magnitude and phase response of synthesis FIR filters $F_1(z)$ (dashed), and $F_2(z)$ (solid) designed using the proposed method.

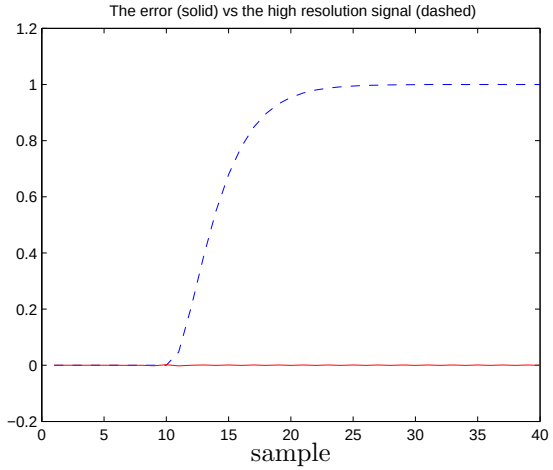


Figure 5.14: The error $e[n]$ (solid) plotted against the desired output $y_0[n]$ (dashed). The $\mathcal{H}_\infty$ norm of the system is $\|\mathcal{K}\|_\infty \approx 4\%$.

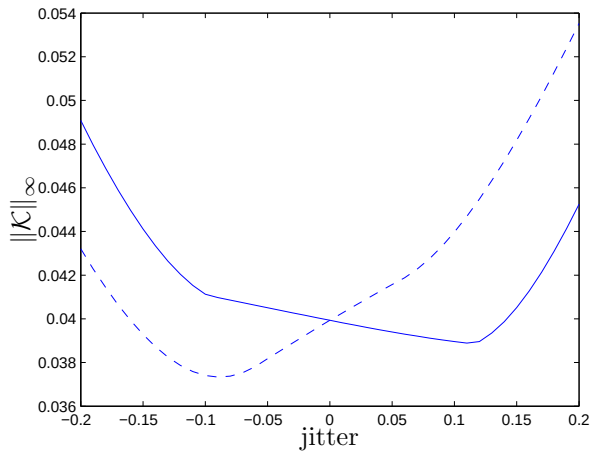


Figure 5.15: The norm $\|\mathcal{K}\|_\infty$ of the induced error system plotted against jitters δ_1 (solid) and δ_2 (dashed).

5.6.3 Comparison to existing methods

We compare the proposed method to an existing method, called the Sinc method. The Sinc method approximates the fractional delay operator e^{-Ds} by an FIR filter using the function $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$:

$$F_D^{(sinc)}[n] = \mathrm{sinc}\Big(n - \frac{D}{2h}\Big),$$

with $|n| \leq N_{cutoff} = 11$. Hence filters of the Sinc method are of 23 taps. Note that, in the formula above, the sampling interval is $2h$.

The Sinc method filters the low resolution signal $x_1[n]$ by the approximated FIR filter $F_{D_1}^{(sinc)}$ to get the *even* samples of $y_0[n]$, and filters the second low resolution signal $x_2[n]$ by the approximated FIR filter $F_{D_2+h}^{(sinc)}$ to get the *odd* samples of $y_0[n]$. In other words, the high resolution signal is obtained by interleaving individually filtered low resolution channels.

Figure 5.16 compares the error of the proposed method to the error of the Sinc method. Both sets of synthesis filters have similar length (length 23 for the Sinc method and length 22 for the proposed method). We observe that the proposed method shows a better performance, especially around the discontinuity.

The improved performance of the proposed technique in Fig. 5.16 is due to two reasons. First, replacing fractional delays $\{e^{-D_i s}\}_{i=1}^N$ by equivalent analysis filters $\{\mathsf{H}_i(z)\}_{i=1}^N$ enhances the results. Second, the use of $\mathcal{H}_\infty$ optimization allows the system to perform even for inputs that are not necessarily bandlimited.

We also compare the proposed method to a second method, called the Separation method. This method, similar to the Sinc method above, obtains the high resolution signal by interleaving individually processed low resolution channels. What distinguishes the Separation method from the Sinc method is that the Separation method approximates the fractional delay operator e^{-Ds} by an

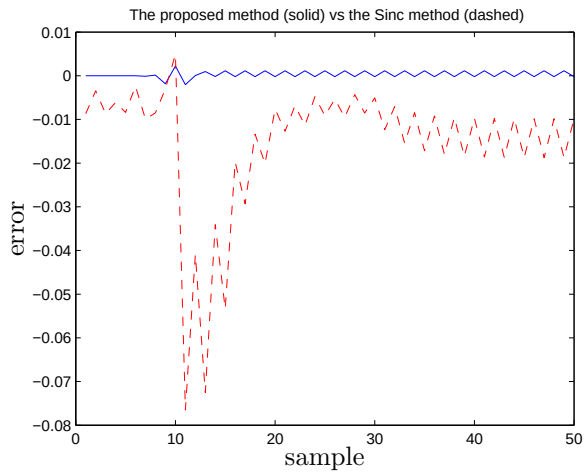


Figure 5.16: Performance comparison of the error of the proposed method (solid) and of the Sinc method truncated to 23 taps (dotted).

Table 5.1: Performance comparison using different inputs. Columns RMSE_1 and Max_1: step function input as in (5.42). Columns RMSE_2 and Max_2: input $f(t) = \sin(0.3t) + \sin(0.8t)$.

	RMSE_1	Max_1	RMSE_2	Max_2
Sinc method	0.0171	0.0765	0.0677	0.1782
Separation method	0.0029	0.0293	0.0084	0.0180
Proposed method	0.0008	0.0023	0.0018	0.0048

IIR operator designed to minimize the $\mathcal{H}_\infty$ norm of an induced error system corresponding to that channel [114].

Figure 5.17 compares the error of the proposed method and the Separation method. Again, the proposed method hence yields a better performance. This is expected as the synthesis filters are designed together, allowing effective exploitation of all low resolution signals.

Table 5.1 shows the comparison of the three methods in terms of the root mean square error (RMSE) and the maximum value (Max). We use two inputs of different characteristics: a step function as in (5.42), and a bandlimited function $f(t) = \sin(0.3t) + \sin(0.8t)$. Observe that the proposed method outperforms existing methods in both norms and both inputs.

5.7 Conclusion and Discussion

In this chapter, we designed digital synthesis filters for a hybrid multirate filter banks with fractional delays, with potential applications in multichannel sampling. We showed that this hybrid system is $\mathcal{H}_\infty$-norm equivalent to a digital system. The equivalent digital system then can be used to design stable synthesis filters, using model-matching or linear matrix inequality methods. We

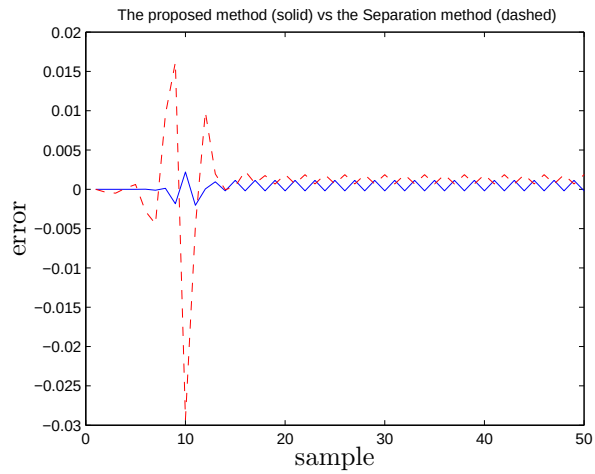


Figure 5.17: Error comparison between the proposed method (solid) and the Separation method (dotted).

also showed the robustness of the induced error system in the presence of delay estimate errors. Experimental results confirmed the superior performance of the proposed method compared to existing methods.

A limitation of the proposed method is the lack of an explicit solution for the synthesis filter $\{F_i(z)\}_{i=1}^N$. However, the design is performed only once for all input signals. Moreover, this drawback can be compensated with the wide availability of design implementations.

It is interesting to note that in our setup, prefiltering is not necessary because the strictly proper system $\Phi_0(s)$ presents in the high-resolution channel. In previous work on hybrid filter design [105, 114, 115], a low-pass filter is usually used to select frequencies of interest of the input signal. Without this prefiltering process, the $\mathcal{H}_\infty$ norm of the induced error system can become infinite [105, 123].

For future work, we would like to investigate the relationship between the upsampling rate M and the number of low resolution channels N to guarantee a predefined performance. Another direction is to design synthesis filters taking into account the uncertainties in the first place, using traditional robust control techniques [124].

CHAPTER 6

CONCLUSION AND FUTURE WORK

Systems using multiple sensors have inspired many important research topics recently for their ability to utilize existing infrastructure and to exploit spatiotemporal information of signals. In this thesis, we propose novel theory and algorithms for two multisensor applications: image-based rendering and multichannel sampling. In this chapter, we recap the main contributions of the thesis and discuss directions for future research.

6.1 Conclusion

This thesis contributes theory, analysis, and algorithms to the applications of image-based rendering and of multichannel sampling.

For image-based rendering (IBR), many existing IBR algorithms use heuristic interpolation of the virtual images as weighted sum of surrounding samples. In Chapter 2, we propose a rigorous approach for IBR algorithms. Specifically, we propose

- A conceptual framework that generalizes many existing IBR algorithms, using calibrated or uncalibrated images, and focuses on rigorous interpolation techniques.
- A technique for IBR to determine what samples are visible at the virtual cameras. The technique, applicable for both calibrated and uncalibrated cases, is adaptive and allows simple implementation.
- A technique to interpolate the virtual images using both the intensity and depth of actual samples.

Little research has addressed the sampling problem for IBR, in particular for the analysis of IBR algorithms. As a consequence, many IBR systems have to rely on oversampling to encounter aliasing at the virtual cameras. The most important contribution of the thesis is the analysis of IBR texture mapping algorithms using depth maps, presented in Chapters 3 and 4. The contributions in these chapters include novel techniques to analyze the rendering quality of

IBR algorithms, and bounds for the mean absolute errors derived using these novel techniques. Specifically, we propose

- A methodology to analyze the rendering quality of IBR texture mapping algorithms using explicit depth maps.

In order to apply the above methodology, we also propose two novel techniques:

- An approximation of available samples (derived from the actual pixels to the virtual image plane or the scene surface) as a generalized Poisson process.
- Bounds for sample jitters (caused by wrong depth estimates) based on the relative position between the virtual camera and the scene.

Using the proposed methodology, we derive:

- Bounds for the mean absolute errors (MAE) of IBR texture mapping algorithms using depth maps. In particular, the bounds successfully capture the decay ($\mathcal{O}(\lambda^{-2})$ for 2D scenes and $\mathcal{O}(\lambda^{-1})$ for 3D scenes) of the MAE with respect to the local density of actual samples λ.

Finally, in Chapter 5, we design synthesis filters for a hybrid system to approximate the output of a fast A/D converter using outputs of multiple slow A/D converters. Specifically, we

- Show the equivalence of a hybrid system to a discrete-time linear time-invariant system.
- Use the equivalent system to design the synthesis filters using standard $\mathcal{H}_\infty$ optimization tools such as model-matching and linear matrix inequalities (LMI).
- Show that the system using the designed synthesis filters are stable against the uncertainties of the delay estimates.

6.2 Future Work

As for future work, we intend to investigate the following problems.

Framing IBR data. In Chapters 3 and 4, we analyze the rendering quality of IBR algorithms, assuming the ideal pinhole camera model. We briefly discussed that, in practice, the intensity at a pixel is the convolution of the image light field with a point spread function. Hence, the actual pixels can be considered as samples of the surface texture using scaled and shifted versions of the point spread function as sampling functions. Because these

sampling functions are linearly dependent, they form not a basis but rather a frame. We intend to analyze the sampling and reconstruction of IBR data using techniques of the frame theory [125, 126, 127, 128].

Algorithm and analysis of IBR with non-Lambertian surfaces. An extension of for the IBR problem considered in this thesis is for non-Lambertian surfaces. Since images of non-Lambertian surfaces change according to viewpoints, another dimension must be added to incorporate the angle between the viewpoint and the surface normal. If the surface normals are available, rendering the virtual images can still be considered as a problem of nonuniform interpolation, though in a higher dimensional space. Moreover, since the change in the angular dimension of the surface light field is usually slower than its change in the spatial dimension [129], the quality of the virtual image may be suboptimal if we interpolate using Delaunay triangulation or tessellation. We intend to propose novel algorithms and analysis for IBR with non-Lambertian surfaces.

IBR distributed coding. Results derived in Chapters 3 and 4 suggest that the rendering quality depends strongly on the local density of the actual samples. Since the sample density of the overall IBR system is the summation of the sample density at the actual cameras, we expect that the local "innovative information" provided by the actual cameras is linearly additive. An implication is that an independent coding scheme of depth image-based rendering data at the actual cameras can achieve comparable performance to joint coding schemes. We intend to investigate this issue using information theoretical frameworks [130, 131, 132, 133, 134].

2D multichannel sampling. We intend to extend the result into 2D, focusing on the design of 2D FIR synthesis filters for practical purposes. The designed systems have potential applications in image superresolution. In Chapter 5, we design synthesis filters to approximate fast A/D converters using slow A/D converters in the presence of fractional delays. The main building blocks of the design procedure for the 1D case are a hybrid-to-digital system conversion technique, multirate system theory, and the bounded-real lemma to convert the $\mathcal{H}_\infty$ optimization problem into an LMI problem. All these building blocks have corresponding literature in 2D [135, 136, 137, 138, 139, 140].

APPENDIX A

SUPPORTING MATERIAL

A.1 Geometrical Interpretation of $H_{\Pi}(u)$

The derivative $H'_{\Pi}(u)$ of the scene-to-image mapping, defined in (3.2), is necessary to estimate Y_k and U_k, important factors in the error bounds of Theorem 3.1 and 3.2. In this appendix, we present a geometrical interpretation of $H'_{\Pi}(u)$. Let $\Pi = [\mathsf{R}, \boldsymbol{T}]$ and $\boldsymbol{Q}(u) = \boldsymbol{S}(u) + \boldsymbol{S}'(u)$ (see Fig. A.1).

Lemma A.1 *The derivative $H'_{\Pi}(u)$ of the scene-to-image mapping can be computed as*

$$H'_{\Pi}(u) = \frac{\det(\mathsf{A})}{d(u)^2}, \tag{A.1}$$

where

$$\mathsf{A} = \Pi \cdot \left[\widetilde{\boldsymbol{Q}}(u), \widetilde{\boldsymbol{S}}(u)\right]. \tag{A.2}$$

Let $\boldsymbol{e}_3 = [0, 0, 1]^T$ and $S_{\boldsymbol{QSC}}$ be the area of the triangle $\boldsymbol{QSC}$. Taking the determinant of the equality

$$\begin{bmatrix} \boldsymbol{\pi}_1^T \\ \boldsymbol{\pi}_2^T \\ \boldsymbol{e}_3^T \end{bmatrix} \cdot \left[\widetilde{\boldsymbol{Q}}, \widetilde{\boldsymbol{S}}, \widetilde{\boldsymbol{C}}\right] = \begin{bmatrix} \boldsymbol{\pi}_1^T \widetilde{\boldsymbol{Q}} & \boldsymbol{\pi}_1^T \widetilde{\boldsymbol{S}} & 0 \\ \boldsymbol{\pi}_2^T \widetilde{\boldsymbol{Q}} & \boldsymbol{\pi}_2^T \widetilde{\boldsymbol{S}} & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

we obtain

$$2S_{\boldsymbol{QSC}} = \det(\mathsf{A}). \tag{A.3}$$

From (A.2) and (A.3) we obtain the following proposition:

Proposition A.1 *The derivative $H'_{\Pi}(u)$ of the scene-to-image mapping can be computed as*

$$H'_{\Pi}(u) = \frac{2S_{\boldsymbol{QSC}}}{d(u)^2}. \tag{A.4}$$

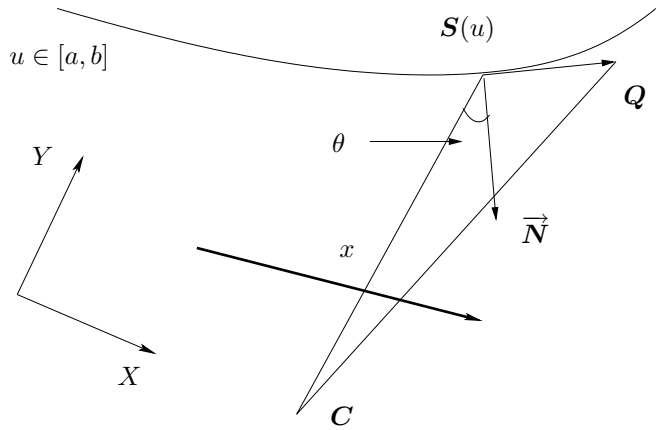


Figure A.1: The derivative $H'_{\Pi}(u)$ is proportional to the area S_{QSC} of the triangle $\triangle\mathsf{QSC}$, and inversely proportional to the square of the depth.

Using (A.4), the derivative $H'_v(u)$ corresponding to the virtual camera can be computed as

$$H'_v(u) = \frac{\|\boldsymbol{C}_v - \boldsymbol{S}(u)\|_2}{d(u)^2} \cdot \|\boldsymbol{S}'(u)\|_2 \cdot \cos(\theta),$$

where θ is the angle between vector $\overrightarrow{\boldsymbol{SC}_v}$ and the normal vector $\overrightarrow{\boldsymbol{N}}$ of the scene at $\boldsymbol{S}$ (see Fig. A.1). We note the connection of $H'_v(u)$ to B_v defined in (3.19). Moreover, $H'_{\Pi}(u)$ becomes larger if the angle θ is reduced. In other words, $H'_{\Pi}(u)$ is larger if the camera is placed toward the scene surface. Finally, we note that the value of $H'_{\Pi}(u)$ is related to the notion of *foreshortening* in computer vision [67].

A.2 Proof of Proposition 4.1

Denote the following functions as linear interpolations of corresponding samples:

$$\begin{aligned}
\widehat{f}_{12}(x) &= \frac{x - x_1}{x_2 - x_1} f(x_2) + \frac{x_2 - x}{x_2 - x_1} f(x_1) \\
\widehat{f}_{1d}(x) &= \frac{x - x_1}{x_d - x_1} f(x_d^-) + \frac{x_d - x}{x_d - x_1} f(x_1) \\
\widehat{f}_{d2}(x) &= \frac{x - x_d}{x_2 - x_d} f(x_2) + \frac{x_2 - x}{x_2 - x_d} f(x_d^+).
\end{aligned}$$

Let $e_{12}(x), e_{1d}(x)$, and $e_{d2}(x)$ be the corresponding interpolation errors. The aggregated interpolation error is defined as

$$E_{12} = \int_{x_1}^{x_2} |e_{12}(x)| dx \tag{A.5}$$

for $\widehat{f}_{12}(x)$ over the interval $[x_1, x_2]$. The aggregated errors E_{1d} and E_{d2} are

defined similarly.

Lemma A.2 *The equality*

$$\widehat{f}_{12}(x_d) = \frac{\Delta_1}{\Delta} f(x_d^+) + \frac{\Delta_2}{\Delta} f(x_d^-) + \frac{\Delta_1 \Delta_2}{\Delta} J_1 + B \tag{A.6}$$

holds for some B such that

$$|B| \leq \frac{1}{2} \Delta_1 \Delta_2 \cdot \|f''\|_\infty. \tag{A.7}$$

Proof A.1 *Using the Taylor expansion we write*

$$f(x_1) = f(x_d^-) - \Delta_1 f'(x_d^-) + \frac{1}{2} \Delta_1^2 f''(\xi_1) \tag{A.8}$$

for some $\xi_1 \in [x_1, x_d]$. A similar equation can be also derived for x_2. Hence (A.6) *holds for*

$$B = \frac{\Delta_1^2 \Delta_2}{2\Delta} f''(\xi_1) + \frac{\Delta_1 \Delta_2^2}{2\Delta} f''(\xi_2). \tag{A.9}$$

For B defined above, it is easy verify (A.6).

Next, we propose a bound, for the case $\mu_i = \varepsilon_i = 0$ for $i = 1, 2$, that is in fact tighter than the one proposed in Proposition 4.1.

Lemma A.3 *The aggregated error E_{12}, when there are no sample errors and jitters, can be bounded by*

$$E_{12} \leq \frac{1}{12} \Delta^3 \cdot \|f''\|_\infty + \frac{\Delta_1^2 + \Delta_2^2}{2\Delta} \cdot |J_0| + \frac{1}{2} \Delta_1 \Delta_2 \cdot |J_1|. \tag{A.10}$$

Proof A.2 *We can bound E_{12} by the summation of E_{1d}, E_{d2}, and the area of the quadrangle formed by $[x_1, f(x_1)]^T$, $[x_2, f(x_2)]^T$, $[x_d, f(x_d^+)]^T$, and $[x_d, f(x_d^-)]^T$ (the shaded region in Fig. A.2). Hence*

$$E_{12} \leq E_{1d} + E_{d2} + \frac{\Delta_1}{2} |\widehat{f}_{12}(x_d) - f(x_d^-)| + \frac{\Delta_2}{2} |\widehat{f}_{12}(x_d) - f(x_d^+)|. \tag{A.11}$$

Next, inequalities similar to [77, Equation (11)] can be derived for E_{1d}, E_{d2}. Integrating both sides of these inequalities we obtain

$$E_{1d} \leq \frac{1}{12} \Delta_1^3 \cdot \|f''\|_\infty, \qquad E_{d2} \leq \frac{1}{12} \Delta_2^3 \cdot \|f''\|_\infty. \tag{A.12}$$

Substituting $\widehat{f}_{12}(x_d)$ as in (A.6) *into* (A.11)*, together with inequalities* (A.12)*, we will indeed prove* (A.10).

Finally, to extend Lemma A.3 in the presence of sample errors and jitters, it is sufficient to prove the following lemma.

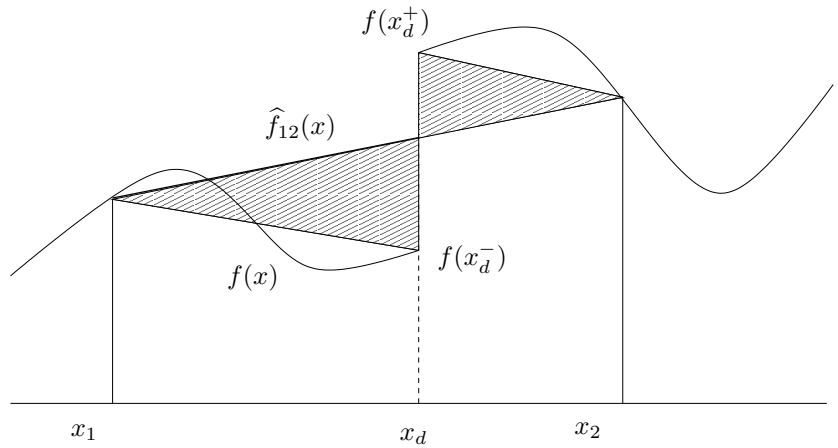


Figure A.2: Linear interpolation error.

Lemma A.4 *The following inequality holds for $i = 1, 2$:*

$$|f(x_i + \mu_i) + \varepsilon_i - f(x_i)| \leq |\varepsilon_i| + |\mu_i| \cdot \|f'\|_\infty + |J_0|. \tag{A.13}$$

Proof A.3 *For arbitrary $x, y \in [x_1, x_2]$, with $x \leq x_d \leq y$:*

$$\begin{aligned} |f(y) - f(x)| &\leq |f(y) - f(x_d^+)| + |J_0| + |f(x_d^-) - f(x)| \\ &\leq |y - x_d| \cdot |f'(\theta_1)| + |x_d - x| \cdot |f'(\theta_2)| + |J_0| \\ &\leq |y - x| \cdot \|f'\|_\infty + |J_0|. \end{aligned}$$

The last inequality easily implies (A.13).

A.3 Geometrical Interpretation of $\boldsymbol{H}_\Pi(u, v)$

We present a property of the scene-to-image mapping $\boldsymbol{H}_\Pi(u, v)$ in this appendix– a generalization of the 2D case shown in [77]. In the following, we use $\boldsymbol{S}$ instead of $\boldsymbol{S}(u, v)$. We denote

$$\begin{aligned} \boldsymbol{S}_u(u, v) &= \boldsymbol{S}(u, v) + \frac{\partial \boldsymbol{S}(u, v)}{\partial u}, \\ \boldsymbol{S}_v(u, v) &= \boldsymbol{S}(u, v) + \frac{\partial \boldsymbol{S}(u, v)}{\partial v}. \end{aligned}$$

Lemma A.5 *The Jacobian $\partial \boldsymbol{H}_\Pi(u, v)/\partial(u, v)$ of the scene-to-image mapping has the determinant*

$$\det\left(\frac{\partial \boldsymbol{H}_\Pi(u, v)}{\partial(u, v)}\right) = \frac{\det(\mathsf{A})}{d(u, v)^3}, \tag{A.14}$$

where

$$\mathsf{A} = \Pi \cdot \left[\widetilde{\boldsymbol{S}}_u, \widetilde{\boldsymbol{S}}_v, \widetilde{\boldsymbol{S}}\right]. \tag{A.15}$$

Let $\boldsymbol{e}_4 = [0,0,0,1]^T \in \mathbb{R}^4$. Taking the determinant of the following equality

$$\begin{bmatrix} \Pi \\ \boldsymbol{e}_4^T \end{bmatrix} \cdot \left[\widetilde{\boldsymbol{S}}_u, \widetilde{\boldsymbol{S}}_v, \widetilde{\boldsymbol{S}}, \widetilde{\boldsymbol{C}}\right] = \begin{bmatrix} \mathsf{A} & \mathbf{0} \\ \mathbf{1} & 1 \end{bmatrix},$$

we obtain:

$$\det(\mathsf{A}) = 6V_{\boldsymbol{S}_u\boldsymbol{S}_v\boldsymbol{S}\boldsymbol{C}}, \tag{A.16}$$

where $V_{\boldsymbol{S}_u\boldsymbol{S}_v\boldsymbol{S}\boldsymbol{C}}$ is the volume of the tetrahedron $\boldsymbol{S}_u\boldsymbol{S}_v\boldsymbol{S}\boldsymbol{C}$. We summarize the result in Proposition A.2.

Proposition A.2 *The Jacobian $\partial \boldsymbol{H}_\Pi(u,v)/\partial(u,v)$ of the scene-to-image mapping the has determinant*

$$\det\left(\frac{\partial \boldsymbol{H}_\Pi(u,v)}{\partial(u,v)}\right) = \frac{6V_{\boldsymbol{S}_u\boldsymbol{S}_v\boldsymbol{S}\boldsymbol{C}}}{d(u,v)^3}. \tag{A.17}$$

A.4 Review of State-Space Methods

This appendix reviews basic notions of state-space methods. For more details, readers are referred to [141]. We consider a finite-dimensional, linear time-invariant, causal system G whose transfer function $\mathsf{G}(s)$ is proper. Let $\boldsymbol{u}(t) \in \mathbb{R}^m$ be the input, $\boldsymbol{y}(t) \in \mathbb{R}^p$ be the output, and $\boldsymbol{x}(t) \in \mathbb{R}^n$ be a set of states of G. Then G has a state-space representation of form

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) + D\boldsymbol{u}(t), \end{cases} \tag{A.18}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}, D \in \mathbb{R}^{p\times m}$ are constant matrices, and $\dot{\boldsymbol{x}}(t)$ denotes the time derivative of $\boldsymbol{x}(t)$.

Let $\mathsf{U}(s), \mathsf{X}(s)$, and $\mathsf{Y}(s)$ be the Laplace transforms of $\boldsymbol{u}(t), \boldsymbol{x}(t)$, and $\boldsymbol{y}(t)$, respectively. Then (A.18) implies

$$\begin{cases} s\mathsf{X}(s) &= A\mathsf{X}(s) + B\mathsf{U}(s) \\ \mathsf{Y}(s) &= C\mathsf{X}(s) + D\mathsf{U}(s). \end{cases}$$

Thus, the transfer function from $\boldsymbol{u}(t)$ to $\boldsymbol{y}(t)$ is the $p \times m$ rational matrix $\mathsf{G}(s)$:

$$\mathsf{G}(s) = D + C(sI - A)^{-1}B. \tag{A.19}$$

Inversely, any proper rational transfer function $\mathsf{G}(s)$ has a state-space realization satisfying (A.19). If $\mathsf{G}(s)$ is strictly proper, the D-matrix of $\mathsf{G}(s)$ is a

zero matrix. We also use package notation $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.

The state-space method in digital is similar to analog. A finite-dimensional, linear-time-invariant system, causal system with input $\boldsymbol{u}[n] \in \mathbb{R}^m$, output $\boldsymbol{y}[n] \in \mathbb{R}^p$, has a state-space model of the form

$$\left\{ \begin{array}{rcl} \boldsymbol{x}[n+1] & = & A\boldsymbol{x}[n] + B\boldsymbol{u}[n] \\ \boldsymbol{y}[n] & = & C\boldsymbol{x}[n] + D\boldsymbol{u}[n], \end{array} \right.$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$ are constant matrices. Note that in this case, $\boldsymbol{x}[n+1]$ denotes the time advance of $\boldsymbol{x}[n]$ instead of $\dot{\boldsymbol{x}}(t)$ as in (A.18). The transfer function from $\boldsymbol{u}[n]$ to $\boldsymbol{y}[n]$ is

$$\begin{array}{rcl} \mathsf{G}(z) & = & D + C(zI - A)^{-1}B \\ & = & D + \sum_{n=1}^{\infty} CA^{n-1}Bz^{-n}. \end{array}$$

A.5 Computation of the Norm of BB^*

This appendix presents how to compute the norm of the product BB^*. The adjoint operators of $\{\mathsf{Q}_i\}_{i=0}^N$ and $\{\mathsf{R}_i\}_{i=1}^N$ are

$$\begin{array}{rcl} (\mathsf{Q}_i^* x)(t) & = & B_i^T e^{(h-t)A_i^T} x \\ (\mathsf{R}_i^* x)(t) & = & \mathbf{1}_{[0,h-d_i)} B_i^T e^{(h-d_i-t)A_i^T} C_i^T x. \end{array}$$

Hence, the adjoint operator of B_i is $\mathsf{B}_i^* = [\mathsf{Q}_i^*, \mathsf{R}_i^*]$ and the adjoint operator of B is $\mathsf{B}^* = [\mathsf{Q}_0^*, \mathsf{B}_1^*, \ldots, \mathsf{B}_N^*]$. Lemma A.6 provides a formula to compute the product BB^*.

Lemma A.6 *The operator BB^* is a linear operator characterized by a symmetric matrix $\Delta = (\Delta_{ij})_{i,j=0}^N$ with*

$$\Delta_{ij} = \left\{ \begin{array}{lcll} \mathsf{Q}_0\mathsf{Q}_0^*, & & & \text{if } i = j = 0 \\ \mathsf{Q}_0\mathsf{B}_j^* & = & \begin{bmatrix} \mathsf{Q}_0\mathsf{Q}_j^* & \mathsf{Q}_0\mathsf{R}_j^* \end{bmatrix}, & \text{if } 0 = i < j \\ \mathsf{B}_i\mathsf{B}_j^* & = & \begin{bmatrix} \mathsf{Q}_i\mathsf{Q}_j^* & \mathsf{Q}_i\mathsf{R}_j^* \\ \mathsf{R}_i\mathsf{Q}_j^* & \mathsf{R}_i\mathsf{R}_j^* \end{bmatrix}, & \text{if } 0 < i \le j \\ \Delta_{ji}^T, & & & \text{if } i > j. \end{array} \right.$$

Each block Δ_{ij} is composed by components of forms $\mathsf{Q}_i\mathsf{Q}_j^*, \mathsf{Q}_i\mathsf{R}_j^*$ and $\mathsf{R}_i\mathsf{R}_j^*$ that can be computed as

$$\mathsf{Q}_i\mathsf{Q}_j^* = \mathsf{M}_{ij}(h) \tag{A.20}$$

$$\mathsf{Q}_i\mathsf{R}_j^* = e^{d_jA_j}\mathsf{M}_{ij}(h-d_j)C_j^T \tag{A.21}$$

$$\mathsf{R}_i\mathsf{R}_j^* = \left\{ \begin{array}{ll} C_i e^{(d_j-d_i)A_i}\mathsf{M}_{ij}(h-d_j)C_j^T, & \text{if } d_i < d_j \\ C_i\mathsf{M}_{ij}(h-d_i)e^{(d_i-d_j)A_i^T}C_j^T, & \text{if } d_i \ge d_j, \end{array} \right. \tag{A.22}$$

where

$$\mathsf{M}_{ij}(t) := \int_0^t e^{\tau A_i} B_i B_j^T e^{\tau A_j^T} d\tau.$$

Proof A.4 *We show here the proof of* (A.22)*. The proofs of* (A.20) *and* (A.21) *are similar. Consider the case $d_i < d_j$. For any x of appropriate dimension we have*

$$\begin{aligned} \left(\mathsf{R}_i \mathsf{R}_j^*\right) x &= C_i \int_0^{h-d_i} e^{(h-d_i-\tau)A_i} B_i (\mathsf{R}_j^* x)(\tau) d\tau \\ &= \left(C_i e^{(d_j-d_i)A_i} \mathsf{M}_{ij}(h-d_j) C_j^T\right) x. \end{aligned}$$

Hence if $d_i < d_j$ we indeed verify

$$\mathsf{R}_i \mathsf{R}_j^* = C_i e^{(d_j-d_i)A_i} \mathsf{M}_{ij}(h-d_j) C_j^T.$$

The proof is similar for the case where $d_i \geq d_j$.

Finally, note that $\mathsf{M}_{ij}(t)$ can be efficiently computed as [142]

$$\mathsf{M}_{ij}(t) = e^{A_i t} \pi_{12}(t),$$

where $\pi_{12}(t)$ is the block $(1,2)$ of the matrix

$$\begin{bmatrix} \pi_{11}(t) & \pi_{12}(t) \\ 0 & \pi_{22}(t) \end{bmatrix} = \exp\left(\begin{bmatrix} -A_i & B_i B_j^T \\ 0 & A_j^T \end{bmatrix} t \right).$$

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[3] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. New York, NY: Springer, 2007.

[4] H. Y. Shum, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 13, pp. 1020–1037, November 2003.

[5] C. Zhang and T. Chen, "A survey on image-based rendering - representation, sampling and compression," *EURASIP Signal Processing: Image Communication*, pp. 1–28, January 2004.

[6] T. Ajdler, L. Sbaiz, and M. Vetterli, "The plenacoustic function and its sampling," *IEEE Trans. Signal Proc.*, vol. 54, no. 10, pp. 3790–3804, October 2006.

[7] J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics*. Boston, MA: Addison-Wesley, 1990.

[8] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY: Cambridge University Press, 2004.

[9] Y. Ma, J. Kosecka, S. Soatto, and S. Sastry, *An Invitation to 3-D Vision*. New York, NY: Springer, 2003.

[10] R. Kalawsky, *The Science of Virtual Reality and Virtual Environments*. Boston, MA: Addison-Wesley Longman, 1993.

[11] N. R. Council, *Virtual Reality: Scientific and Technological Challenges*, N. I. Durlach and A. S. Mavor, Eds. Washington, DC: National Academy Press, 1994.

[12] T. B. Sheridan, "Musings on telepresence and virtual presence," *Presence: Teleoperators and Virtual Environments*, vol. 1, no. 1, pp. 120–126, January 1992.

[13] R. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, August 1997.

[14] R. Azuma, Y. Baillot, R. Behringer, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–47, November/December 2001.

[15] W. Matusik and H. Pfister, "3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes," in *Proc. of SIGGRAPH*, 2004, pp. 814–824.

[16] C. Fehn, "A 3D-TV approach using depth-image-based rendering (DIBR)," in *Proc. of Visualization, Imaging, and Image Processing*, Spain, September 2003, pp. 482–487.

[17] A. Aldroubi and K. Gröchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *SIAM Review*, vol. 43, no. 4, pp. 585–620, April 2001.

[18] C. de Boor, *A Practical Guide to Splines*. New York, NY: Springer-Verlag, 1978.

[19] C. de Boor, *A Practical Guide to Splines, Revised Edition*. New York, NY: Springer-Verlag, 2001.

[20] F. Marvasti, *Nonuniform Sampling: Theory and Practice*. New York, NY: Kluwer Academic/Plenum Publishers, 2001.

[21] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp. 22–38, November 1999.

[22] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *Proc. of SIGGRAPH*, 1996, pp. 43–54.

[23] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. of SIGGRAPH*, 1996, pp. 31–40.

[24] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 3–20.

[25] M. Unser, "Sampling - 50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.

[26] C. Shannon, "Classic paper: Communication in the presence of noise," *Proc. IEEE*, vol. 86, no. 2, pp. 447–457, 1998.

[27] C. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10–21, January 1949.

[28] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Proc. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

[29] H. Shekarforoush, M. Berthod, and J. Zerubia, "3D super-resolution using generalised sampling expansion," in *Proc. IEEE Int. Conf. on Image Proc.*, vol. 2, October 1995, pp. 300–303.

[30] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graph. Models Image Process.*, vol. 54, no. 2, pp. 181–186, 1992.

[31] J. Franca, A. Petraglia, and S. K. Mitra, "Multirate analog-digital systems for signal processing and conversion," in *Proc. IEEE*, vol. 35, no. 2, February 1997, pp. 242–262.

[32] A. Papoulis, "Generalized sampling expansion," *IEEE Trans. Circ. and Syst.*, vol. 24, no. 11, pp. 652–654, November 1977.

[33] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Patt. Recog. and Mach. Intell.*, vol. 19, no. 6, pp. 580–593, June 1997.

[34] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, no. 293, pp. 133–135, September 1981.

[35] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *Int. J. Comput. Vision*, vol. 17, pp. 43–75, January 1996.

[36] O. D. Faugeras, *Three-dimensional computer vision: A geometric viewpoint*. Cambridge, MA: MIT Press, 1993.

[37] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. of SIGGRAPH*, 1995, pp. 29–38.

[38] S. E. Chen, "Quicktime VR - an image-based approach to virtual environment navigation," in *Proc. of SIGGRAPH*, 1995, pp. 29–38.

[39] S. Laveau and O. Faugeras, "3-D scene representation as a collection of images," in *Int. Conf. Patt. Recog.*, vol. 1, 1994, pp. 689–691.

[40] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. of SIGGRAPH*, 1995, pp. 39–46.

[41] L. McMillan, "An image-based approach to three-dimensional computer graphics," Ph.D. dissertation, University of North Carolina at Chapel Hill, 1997.

[42] P. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. SIGGRAPH*, 1996, pp. 11–20.

[43] P. E. Debevec, G. Borshukov, and Y. Yu, "Efficient view-dependent image-based rendering with projective texture-mapping," in *In 9th Eurographics Rendering Workshop*, Vienna, Austria, June 1998.

[44] P. S. Heckbert and H. P. Moreton, "Interpolation for polygon texture mapping and shading," in *State of the Art in Computer Graphics: Visualization and Modeling*, D. F. Rogers and R. A. Earnshaw, Eds. New York, NY: Springer-Verlag, 1991, pp. 101–111.

[45] C. Bueler, M. Bosse, L. McMillan, G. Bishop, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. of SIGGRAPH*, 2001, pp. 425–432.

[46] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Proc. of SIGGRAPH*, 1998, pp. 231–242.

[47] S. M. Seitz and C. M. Dyer, "View morphing," in *Proc. of SIGGRAPH*, 1996, pp. 21–30.

[48] M. Lhuillier and L. Quan, "Image-based rendering by joint view triangulation," *IEEE Trans. Circ. and Syst.*, vol. 13, no. 11, pp. 1051–1062, November 2003.

[49] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Proc. of SIGGRAPH*, 2000, pp. 307–318.

[50] C. Zhang and T. Chen, "Spectral analysis for sampling image-based rendering data," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 13, no. 11, pp. 1038–1050, November 2003.

[51] S. C. Chan and H. Y. Shum, "A spectral analysis for light field rendering," in *Proc. IEEE Int. Conf. on Image Proc.*, September 2000, pp. 10–13.

[52] M. N. Do, D. Marchand-Maillet, and M. Vetterli, "On the bandlimitedness of the plenoptic function," in *Proc. IEEE Int. Conf. on Image Proc.*, vol. 3, September 2005, pp. 17–20.

[53] B. Chai, S. Sethuraman, H. Sawhney, and P. Hatrack, "Depth map compression for real-time view-based rendering," *Pattern Recognition Letters*, vol. 25, no. 7, pp. 755–766, May 2004.

[54] R. Krishnamurthy, B. B. Chai, H. Tao, and S. Sethuraman, "Compression and transmission of depth maps for image-based rendering," in *Proc. IEEE Int. Conf. on Image Proc.*, 2001, pp. 828–831.

[55] Joint Photographic Experts Group (JPEG), "JPEG community homepage," http://www.jpeg.org.

[56] G. Taubin, "3D geometry compression and progressive transmission," in *Eurographics – State of the Art Report*, September 1999, tutorial.

[57] J. Duan and J. Li, "Compression of the layered depth image," *IEEE Trans. Image Proc.*, vol. 12, pp. 365–372, March 2003.

[58] C. Fehn, "Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV," in *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, San Jose, CA, January 2004, pp. 93–104.

[59] E. T. Whittaker, "On the functions which are represented by the expansions of the interpolation theory," *Proc. Royal Soc. Edinburgh*, vol. 35, pp. 181–194, 1915.

[60] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems*. London, U.K.: Springer, 1995.

[61] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. New York, NY: Prentice Hall, 1993.

[62] H. T. Nguyen and M. N. Do, "Image-based rendering with depth information using the propagation algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 2, Philadelphia, March 2005, pp. 589–592.

[63] H. T. Nguyen and M. N. Do, "A unified framework for calibrated and uncalibrated image-based rendering," *IEEE Trans. Image Proc.*, submitted for publication.

[64] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Proc. I3D Graphics Symp.*, 1997, pp. 7–16.

[65] B. Delaunay, "Sur la sphere vide," *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, vol. 7, pp. 793–800, 1934.

[66] R. Hartley, "Cheirality," *Int. J. Comput. Vision*, vol. 26, no. 1, pp. 41–61, 1998.

[67] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. New York, NY: Prentice-Hall, 2002.

[68] M. Unser and J. Zerubia, "A generalized sampling theory without bandlimiting constraints," *IEEE Trans. Circ. and Syst. II: Analog and Digital Signal Proc.*, vol. 45, pp. 959–969, May 1998.

[69] G. Wolberg, *Digital Image Warping*. Los Alamitos, CA: IEEE Computer Society Press, 1994.

[70] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. New York, NY: Springer Verlag, 2003.

[71] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 195–202, June 2003.

[72] M. Hebert, "Active and passive range sensing for robotics," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 102–110.

[73] P. Soille, *Morphological Image Analysis: Principles and Applications*. New York, NY: Springer-Verlag, 1999.

[74] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Patt. Recog. and Mach. Intell.*, pp. 603–619, May 2002.

[75] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.

[76] D. Morris and T. Kanade, "Image-consistent surface triangulation," in *IEEE Proc. 19th Conf. Computer Vision and Pattern Recognition*, vol. 1, June 2000, pp. 332–338.

[77] H. T. Nguyen and M. N. Do, "Quantitative analysis for image-based rendering with depth information: 2D unoccluded scenes," *IEEE Trans. Image Proc.*, submitted for publication.

[78] H. T. Nguyen and M. N. Do, "Quantitative analysis for image-based rendering with depth information: Part II–2D occluded scenes and 3D scenes," *IEEE Trans. Image Proc.*, submitted for publication.

[79] H. T. Nguyen and M. N. Do, "Error analysis for image-based rendering with depth information," in *Proc. IEEE Int. Conf. on Image Proc.*, October 2006, pp. 381–384.

[80] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum, *Foundations and Trends in Computer Graphics and Vision: Image-based Rendering*, B. Curless, L. V. Gool, and R. Szeliski, Eds. Hanover, MA: Now Publishers, 2006, vol. 2, no. 3.

[81] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Trans. Graph.*, vol. 5, no. 1, pp. 51–72, 1986.

[82] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IEEE Workshop on Stereo and Multi-Baseline Vision*, pp. 131–140, December 2001.

[83] S. L. Albin, "On Poisson approximations for superposition arrival processes in queues," *Management Science*, vol. 28, no. 2, pp. 126–137, 1982.

[84] D. R. Cox and W. L. Smith, "On the superposition of renewal processes," *Biometrika*, vol. 41, pp. 91–99, June 1954.

[85] E. Cinlar, "Superposition of point processes," *Stochastic Point Processes: Statistical Analysis, Theory, and Applications*, pp. 546–606, 1972.

[86] M. T. Hoopen and H. A. Reuver, "The superposition of random sequences of events," *Biometrika*, vol. 53, no. 3–4, pp. 383–389, 1966.

[87] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 2nd ed. New York, NY: McGraw-Hill, 1984.

[88] R. G. Gallager, *Discrete Stochastic Processes.* Norwell, MA: Kluwer Academic Publishers, 1996.

[89] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, November 1986.

[90] P. Franken, "A refinement of the limit theorem for the superposition of independent renewal processes," *Theory of Probability and its Applications*, vol. 8, no. 3, pp. 320–328, 1963.

[91] D. Prasad, *Introduction to Numerical Analysis*, 2nd ed. Middlesex, U.K.: Alpha Science International, 2005.

[92] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. New York, NY: Wiley, 2000.

[93] P. M. Prenter, *Splines and Variational Methods.* New York, NY: Wiley, 1975.

[94] G. H. Golub and C. F. V. Loan, *Matrix Computations (3rd Ed.).* Baltimore, MD: Johns Hopkins University Press, 1996.

[95] H. T. Nguyen and M. N. Do, "Signal reconstruction from a periodic nonuniform set of samples using $\mathcal{H}_\infty$ optimization," in *Proc. of SPIE*, vol. 6498, San Jose, February 2007.

[96] H. T. Nguyen and M. N. Do, "Minimax design of hybrid multirate filter bank with fractional delays," *IEEE Trans. Signal Proc.*, submitted for publication.

[97] Y. Yamamoto, B. D. O. Anderson, M. Nagahara, and Y. Koyanagi, "Optimal FIR approximation for discrete-time IIR filters," *IEEE Signal Proc. Letters*, vol. 10, no. 9, pp. 273–276, September 2003.

[98] J. Benesty, J. Chen, and Y. Huang, "Time-delay estimation via linear interpoation and cross correlation," *IEEE Trans. Speech Audio Proc.*, vol. 12, no. 5, pp. 509–519, September 2004.

[99] O. S. Jahromi and P. Aarabi, "Theory and design of multirate sensor arrays," *IEEE Trans. Signal Proc.*, vol. 53, no. 5, May 2005.

[100] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-24, no. 4, pp. 320–327, August 1976.

[101] F. Viola and W. F. Walker, "A spline-based algorithm for continuous time-delay estimation using sampled data," *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 1, pp. 80–93, January 2005.

[102] B. P. Lathi, *Linear Systems and Signals*, 2nd ed. New York, NY: Oxford University Press, 1992.

[103] B. Francis, *A Course in $\mathcal{H}_\infty$ Control Theory.* Heidelberg, Germany: Springer-Verlag, 1987.

[104] M. Green and D. J. N. Limebeer, *Linear Robust Control.* Upper Saddle River, NJ: Prentice-Hall, Inc., 1995.

[105] H. Shu, T. Chen, and B. Francis, "Minimax design of hybrid multirate filter banks," *IEEE Trans. Circ. and Syst.*, vol. 44, no. 2, February 1997.

[106] R. G. Shenoy, D. Burnside, and T. W. Parks, "Linear periodic systems and multirate filter design," *IEEE Trans. Signal Proc.*, vol. 42, no. 9, pp. 2242–2256, September 1994.

[107] D. Slepian, "On bandwidth," in *Proc. IEEE*, vol. 64, no. 3, 1976, pp. 292–300.

[108] J. Lam, "Model reduction of delay systems using Pade approximation," *International Journal of Control*, vol. 57, no. 2, pp. 377–391, February 1993.

[109] L. D. Philipp, A. Mahmood, and B. L. Philipp, "An improved refinable rational approximation to the ideal time delay," *IEEE Trans. Circ. and Syst.*, vol. 46, no. 5, pp. 637–640, May 1999.

[110] M. G. Yoon and B. H. Lee, "A new approximation method for time-delay systems," *IEEE Trans. Autom. Control*, vol. 42, no. 7, pp. 1008–1012, July 1997.

[111] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay - tools for fractional delay filter design," *IEEE Signal Proc. Mag.*, vol. 13, no. 1, pp. 30–60, 1996.

[112] V. Balakrishnan and L. Vandenberghe, "Linear matrix inequalities for signal processing: An overview," in *Proceedings of the 32nd Annual Conference on Information Sciences and Systems*, Princeton, NJ, March 1998.

[113] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory.* Philadelphia, PA: SIAM Journ. of Math. Anal., 1994.

[114] M. Nagahara and Y. Yamamoto, "Optimal design of fractional delay filters," *IEEE Conference on Decision and Control*, vol. 6, pp. 6539–6544, December 2003.

[115] M. Nagahara and Y. Yamamoto, "Optimal design of fractional delay FIR filters without band-limiting assumption," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, vol. 4, pp. 221–224, March 2005.

[116] T. Chen and B. Francis, "Design of multirate filter banks by $\mathcal{H}_\infty$ optimization," *IEEE Trans. Signal Proc.*, vol. 43, no. 12, pp. 2822–2830, December 1995.

[117] C. Herley and P. W. Wong, "Minimum rate sampling and reconstruction of signals with arbitrary frequency support," *IEEE Trans. Info. Theory*, vol. 45, no. 5, pp. 1555–1564, July 1999.

[118] P. Marziliano and M. Vetterli, "Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations," *IEEE Trans. Signal Proc.*, vol. 48, no. 12, pp. 3462–3471, December 2000.

[119] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* New York, NY: Prentice-Hall, 1995.

[120] R. Y. Chiang and M. G. Safonov, "MATLAB - robust control toolbox," http://www.mathworks.com, 2005.

[121] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to $\mathcal{H}_\infty$ control," *International Journal of Robust and Nonlinear Control*, vol. 4, pp. 421–448, 1994.

[122] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, "LMI control toolbox," http://www.mathworks.com, 1995.

[123] T. Chen and B. A. Francis, "Input-output stability of sampled-data systems," *IEEE Trans. Autom. Control*, vol. 36, no. 1, pp. 50–58, January 1991.

[124] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach.* New York, NY: Springer-Verlag, 2000.

[125] R. Duffin and S. Schaeffer, "A class of nonharmonic Fourier series," *Trans. Amer. Math. Soc.*, vol. 72, pp. 341–366, 1952.

[126] J. Benedetto, "Irregular sampling and frames," in *Wavelets–A Tutorial in Theory and Applications*, C. Chui, Ed. Boca Raton, FL: CRC Press, 1992, pp. 445–507.

[127] A. Teolis and J. J. Benedetto, "Local frames and noise reduction," *Signal Proc.*, vol. 45, no. 3, pp. 369–387, 1995.

[128] S.-C. Pei and M.-H. Yeh, "An introduction to discrete finite frames," *IEEE Signal Proc. Mag.*, vol. 14, no. 6, pp. 84–96, November 1997.

[129] R. Ramamoorthi, D. Mahajan, and P. Belhumeur, "A first-order analysis of lighting, shading, and shadows," *ACM Trans. Graph.*, vol. 26, no. 1, p. 2, 2007.

[130] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," in *Proc. IEEE*, vol. 93, no. 1, January 2005, pp. 71–83.

[131] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Fusion-based multiview distributed video coding," in *Proc. ACM Int. Workshop on Video Surveillance and Sensor Networks*, 2006, pp. 139–144.

[132] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Info. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[133] A. Wyner, "The rate-distortion function for source coding with side information at the decoder-II: General sources," *Information and Control*, vol. 38, no. 1, pp. 60–80, July 1978.

[134] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Info. Theory*, vol. 22, no. 1, pp. 1–10, January 1976.

[135] C. W. Chen, J. S. H. Tsai, and L. S. Shieh, "Two-dimensional discrete-continuous model conversion," *Circuits, Systems, and Signal Processing*, vol. 18, no. 6, pp. 565–585, 1999.

[136] T. Chen and P. P. Vaidyanathan, "Recent developments in multidimensional multirate systems," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 3, no. 2, pp. 116–137, April 1993.

[137] C. Du and L. Xie, $\mathcal{H}_\infty$ *Control and Filtering of Two-Dimensional Systems.* Heidelberg, Germany: Springer-Verlag, 2002.

[138] T. Kaczorek, *Lecture Notes in Control and Information Sciences 68: Two-Dimensional Linear Systems.* Heidelberg, Germany: Springer-Verlag, 1985.

[139] G. Karlsson and M. Vetterli, "Theory of two dimensional multirate filter banks," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. ASSP-38, no. 6, pp. 925–937, June 1990.

[140] S. Xu, J. Lam, Y. Zou, Z. Lin, and W. Paszke, "Robust $\mathcal{H}_\infty$ filtering for uncertain 2-D continuous systems," *IEEE Trans. Signal Proc.*, vol. 53, no. 5, pp. 1731–1738, May 2005.

[141] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. New York, NY: Oxford University Press, 1999.

[142] C. F. V. Loan, "Computing integrals involving the matrix exponential," *IEEE Trans. Autom. Control*, vol. 23, no. 3, pp. 395–404, June 1978.

AUTHOR'S BIOGRAPHY

Ha Thai Nguyen was born in Phú Thọ, Vietnam, on June 26, 1978. He received the Diplôme d'Ingénieur from the Ecole Polytechnique and Ecole Nationale Supérieure des Télécommunications, and Diplôme d'Etudes Approfondies from Université de Nice Sophia Antipolis, France. Since Spring 2004, he has been a Ph.D. student in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

Ha Thai Nguyen received a Gold Medal at the 37th International Mathematical Olympiad (Bombay, India 1996). He was a coauthor (with Professor Minh Do) of a Best Student Paper in the 2005 IEEE International Conference on Audio, Speech, and Signal Processing (ICASSP), Philadelphia, PA, USA. His principal research interests include computer vision, wavelets, sampling and interpolation, image and signal processing, and speech processing.