

A PHYSICAL SAMPLING METRIC FOR
IMAGE-BASED COMPUTER GRAPHICS

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Ryan McCloud Ismert

January 2003

© 2003 Ryan McCloud Ismert

ALL RIGHTS RESERVED

ABSTRACT

Computer models of the real world often use images of the environment to capture realistic visual complexity. Image-based modeling techniques permit the creation of geometric models with a high level of visual detail from photographs. These models are textured by resampling these images of the scene; we call this process *image-based texturing*. The problem with traditional image-based texturing is the poor quality of the extracted textures, which are often blurred or stretched due to sampling problems. Furthermore, the extent of this degradation varies across the scene, due to differences in the pose and position of the camera relative to each object in each image.

This thesis makes two contributions to image-based computer graphics. First, it introduces a physically-based metric of sampling quality, based on the Jacobian matrix of the imaging transform, which captures the interaction of the imaging system with the imaged environment. This metric provides a direct, physical measure of the quality of resampled textures, and suggests a physical interpretation of the multi-resolution image representations widely used in texture synthesis. The second contribution, which builds on this insight, is a novel use of the metric for extending current texture synthesis methods to image-based texturing processes. Use of the sampling metric enables *detail synthesis* – the insertion of high spatial frequency detail into regions of an image-based model’s textures where the imaging process captures only low frequency texture data. Given a small set of input images and a geometric model of the scene, this technique allows the creation of uniform, high-resolution textures. Our synthesis approach relieves the user of the burden of

collecting large numbers of images and increases the quality of user-driven image-based modeling systems. The research described in this thesis allows both the quantification of sampling effects in image-based computer graphics systems, as well as the correction of degradation in image-based textures.

The sampling metric introduced in this thesis has usefulness far outside the image-based texturing application demonstrated here. Such a metric will have a potential impact in the fields of vision-based geometric reconstruction, material measurement, image-based rendering, and geometric level-of-detail management. The goal of this thesis is merely to introduce the metric and validate its usefulness for one critical application.

BIOGRAPHICAL SKETCH

Ryan Ismert was born and raised in Eugene, Oregon. He became interested in computers at an early age, attending an after school programming class (using BASIC on PET computers) during elementary school. His family bought a IBM XT and Ryan quickly became fascinated with computer hardware and programming. In 1992, he graduated from North Eugene High School, and entered Cornell University that same year. In January 1993, Ryan took a leave of absence from Cornell, during which time he served for four years as an Intelligence Specialist in the US Navy.

In 1997, Ryan returned to Cornell and completed his BA in 2000, majoring in Computer Science with concentrations in Asian History, Business, and Digital Studies. After working as a Software Development Manager for Silver Oak Partners in Boston and London for a year, Ryan returned to the Cornell University Program of Computer Graphics, in pursuit of a Master's degree in Computer Graphics.

To my parents, Roseann and Larry.

ACKNOWLEDGEMENTS

I must first express my gratitude to my advisor Professor Donald Greenberg, for providing me with the opportunity to study under him at the Program of Computer Graphics. The depth of Professor Greenberg's knowledge, his insight into all manner of business and technical issues, and his way of asking challenging questions – not for answers, but to illuminate the fact that there *aren't* clear answers – will be an example to me the rest of my life. I must also express heartfelt thanks to Moreno Piccolotto and Professor Kavita Bala, for their guidance – academic, professional, and personal – over the past two years. The research and technical staff (including Steve Westin, Hurf Sheldon, and others) has provided me with hundreds of hours of their time, enriching my mind, asking hard questions, and bringing my hardware back to life at inconvenient times. Last, without a doubt, the greatest asset of the Program of Computer Graphics is its students – thank you to all of my fellow students, but especially to Suanne Fu and Steve Berman, for their company, camaraderie, and many hours in Stella's helping me to 'focus' and listening to my griping.

The work in this thesis was supported in part by the National Science Foundation Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219) and the generous support of the Intel Corporation.

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Background | 1 |
| 1.2 | Our work | 5 |
| 2 | Related Work | 7 |
| 2.1 | Texture synthesis | 7 |
| 2.2 | Image-based texturing | 8 |
| 2.2.1 | Techniques for small- and medium-scale objects | 9 |
| 2.2.2 | Techniques for large-scale objects | 10 |
| 2.2.3 | Texture quality metrics | 10 |
| 3 | Digital Image System Calibration | 12 |
| 3.1 | Digital imaging systems as sampling systems | 12 |
| 3.1.1 | Sampling basics | 13 |
| 3.1.2 | Environmental considerations | 15 |
| 3.1.3 | Imaging sensor considerations | 17 |
| 3.2 | Properties of digital imaging systems | 18 |
| 3.2.1 | Sensor resolution | 18 |
| 3.2.2 | The camera model | 19 |
| 3.2.3 | Lens distortion | 20 |
| 3.2.4 | Opto-electronic Conversion Function | 22 |
| 3.2.5 | Demosaicing | 24 |
| 3.3 | Calibration assumptions | 25 |
| 4 | Sampling Quality | 26 |
| 4.1 | Image formation | 28 |
| 4.2 | Texture reconstruction | 33 |
| 4.3 | Formal description of sampling | 34 |
| 4.4 | Physically-based sampling metric | 39 |
| 4.4.1 | Sampling in world units | 40 |
| 4.4.2 | Sampling in texture units | 42 |
| 4.4.3 | An example using the Jacobian metric | 44 |
| 4.5 | Comparison of metrics | 47 |
| 5 | Detail Synthesis | 51 |
| 5.1 | Texture Synthesis | 53 |
| 5.2 | Detail Synthesis | 56 |
| 5.2.1 | Overview | 56 |
| 5.2.2 | Determining the source texture scaling | 58 |
| 5.2.3 | Normalizing the target texture | 59 |
| 5.2.4 | Creating the Laplacian pyramids | 60 |
| 5.2.5 | Creating the validity pyramid | 62 |

| | | |
|----------|--|-----------|
| 5.2.6 | Synthesis implementation | 64 |
| 5.3 | Results | 66 |
| 5.3.1 | Image acquisition and modeling | 66 |
| 5.3.2 | Rendered bricks | 67 |
| 5.3.3 | Building columns | 69 |
| 5.3.4 | Pavement | 69 |
| 5.3.5 | Moroccan Doors | 72 |
| 6 | Future Work and Conclusions | 75 |
| | Bibliography | 77 |

LIST OF FIGURES

| | | |
|-----|--|----|
| 1.1 | Several approaches to image-based modeling. A user-driven system, Facade, is shown in (a). A fully automated, computer vision approach is shown in (b). Only a single image (in this case a painting) is used to create the model shown in (c). | 2 |
| 1.2 | Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the changing sampling pattern over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures. | 4 |
| 3.1 | (a) shows the original analog signal. The signal is sampled and the digitized values are shown in (b). When reconstructed with the sinc filter shown in (c), the reconstructed signal (d) is identical to the original signal. | 14 |
| 3.2 | (a) shows the original analog signal. The signal is sampled and the digitized values are shown in (b). When reconstructed with the sinc filter shown in (c), the higher frequency portion of the signal has aliased to a lower frequency, and the reconstruction in (d) is significantly different from the original signal. | 16 |
| 3.3 | (a) shows the original calibration target. The distorted target is shown in (b). Note the strong curvature to previously orthogonal rows and columns at the edges of the target. | 21 |
| 3.4 | The profile in (a) shows an approximately linear OECF. Digital values are proportional to intensity values. The profile in (b) shows the OECF for a gamma-corrected image. In this case, digital values are approximately logarithmic with respect to intensity. | 23 |
| 4.1 | Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the sampling patterns over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures. | 27 |
| 4.2 | An illustration of spatially varying Nyquist frequency, f_N , due to perspective projection. In this example the sample distance A is half the length of the sample distance B. Thus, the $f_{N_A} = 2f_{N_B}$. . . | 30 |
| 4.3 | An illustration of the camera sampling the environment on a non-uniform sampling grid. | 32 |
| 4.4 | An illustration of a texture sampling an image on a nonuniform sampling grid. | 35 |
| 4.5 | Graphical depiction of the relationships between \mathbf{M}_{Img} , \mathbf{M}_{Proj} , and \mathbf{M}_{Tex} | 39 |
| 4.6 | A simple example illustrating the use of the elements of J to judge sample rate. In (a) the texture has more resolution than is necessary to capture the detail in the image. In (b) the texture has too little resolution and cannot capture all of the detail present in the image. | 43 |

| | | |
|-----|---|----|
| 4.7 | A simple example illustrating sampling metrics over an environment. The first row shows the Jacobian for each texel on the indicated faces. The bottom two rows show the determinant of the Jacobian matrix and the cosine of the angle between the view ray and the surface normal, respectively. | 45 |
| 4.8 | A rescaled visualization of the values of the Jacobian matrix for the front face shown in Figure 4.7. Although it is not immediately apparent in the original image, the front face is not parallel to the image plane – it is actually tilted slightly back and to the left. Notice the sensitivity of the metric to slight variations in the sampling rate induced by this very small rotation. | 46 |
| 4.9 | A comparison of the Jacobian metric to the determinant of the Jacobian for several sample projections. All of these patterns evaluate to the same determinant value, while the full Jacobian matrix preserves the actual sampling behavior. | 49 |
| 5.1 | Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the sampling patterns over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures. | 52 |
| 5.2 | Traditional texture synthesis at work. (a) shows the source texture with several neighborhoods indicated. Row (b) illustrates the progress of the synthesis process. (c) shows the result of the synthesis. | 55 |
| 5.3 | An illustration of a Laplacian pyramid with three levels. Level 0 contains the frequencies from f_N , to $\frac{f_N}{2}$. Level 1 contains the frequencies from $\frac{f_N}{2}$ to $\frac{f_N}{4}$. Level 2 contains the frequencies from $\frac{f_N}{4}$ to 0. Note that illustration does not show pyramid data in order to preserve clarity. | 61 |
| 5.4 | Laplacian pyramids for a good source texture and a degraded target texture. | 63 |
| 5.5 | The target Laplacian pyramid is shown in (a) for the texture in (c). The validity pyramid (b) was calculated from the Jacobian values for the surface displayed in (d). The Jacobian values range from approximately 1.0 to 5.2, as shown by the color bar to the right of (d). | 65 |
| 5.6 | Detail synthesis results for the synthetic brick scene. The middle row shows the original textures resampled from the image at top. The bottom row shows the results after detail synthesis. The texture for A was used as the source texture, and thus was not changed. The textures for B and C had significant degradation which was corrected by our technique. | 68 |

| | | |
|-----|---|----|
| 5.7 | Results for columns along the face of Olin library. The front facing side of the nearest column (A) was the source. Both the third column (B) and the fourth column (C) have been synthesized. Although improvement is subtle, the result textures have a full extra octave of data. | 70 |
| 5.8 | Results for a paved pedestrian walkway. The image in a) was used to resample the texture for the target pavement section (outlined in red). A portion of the original texture for this pavement section is shown in b). The results after synthesis are shown in c). The texture portion shown in d) shows the source texture (resampled from another image). | 71 |
| 5.9 | Results for a Moroccan door. An image of the set of doors is shown in a). A detail image is shown in b). A cropped image of the door panel is shown in c), with a blow up showing the lack of detail. The results of detail synthesis are shown in d). Here, the synthesized door panel reveals the intricacies of the inlaid wood pattern. | 73 |

Chapter 1

Introduction

Image-based modeling and texturing are compelling methods for creating detailed and accurate representations of real world structures. Using image-based computer graphics techniques, it is possible to quickly capture or reproduce a much higher level of detail and realism than would be possible using traditional geometric modeling approaches. Highly detailed renderings created in such a way are valuable for cultural heritage, tourism, urban planning, and entertainment purposes.

1.1 Background

A wide variety of approaches to modeling the imaged environment have been developed, especially over the past 20 years. These techniques include approaches using active sensors, such as laser scanners, which generate a three-dimensional model of the geometry of the scene. Other methods create a representation of the environment (including surface appearance) by analyzing only sets of images (or video) from passive sensors, such as digital cameras. We refer to all approaches in the latter group collectively as *image-based modeling*.

Advances in computer- or user-driven image-based modeling have been especially impressive in recent years. Figure 1.1 shows several image-based modeling approaches to geometric reconstruction. Figure 1.1 (a) shows a screen-shot of Facade [DTM96], a user-driven image-based modeling system. The images in (b) show an uncalibrated, fully automated, computer vision approach [PKVG98] at work. Figure 1.1 (c) shows a model created from a single image (in this case the painting ‘La Trinita’, by Masaccio) [CRZ99]. In addition to recent improvements

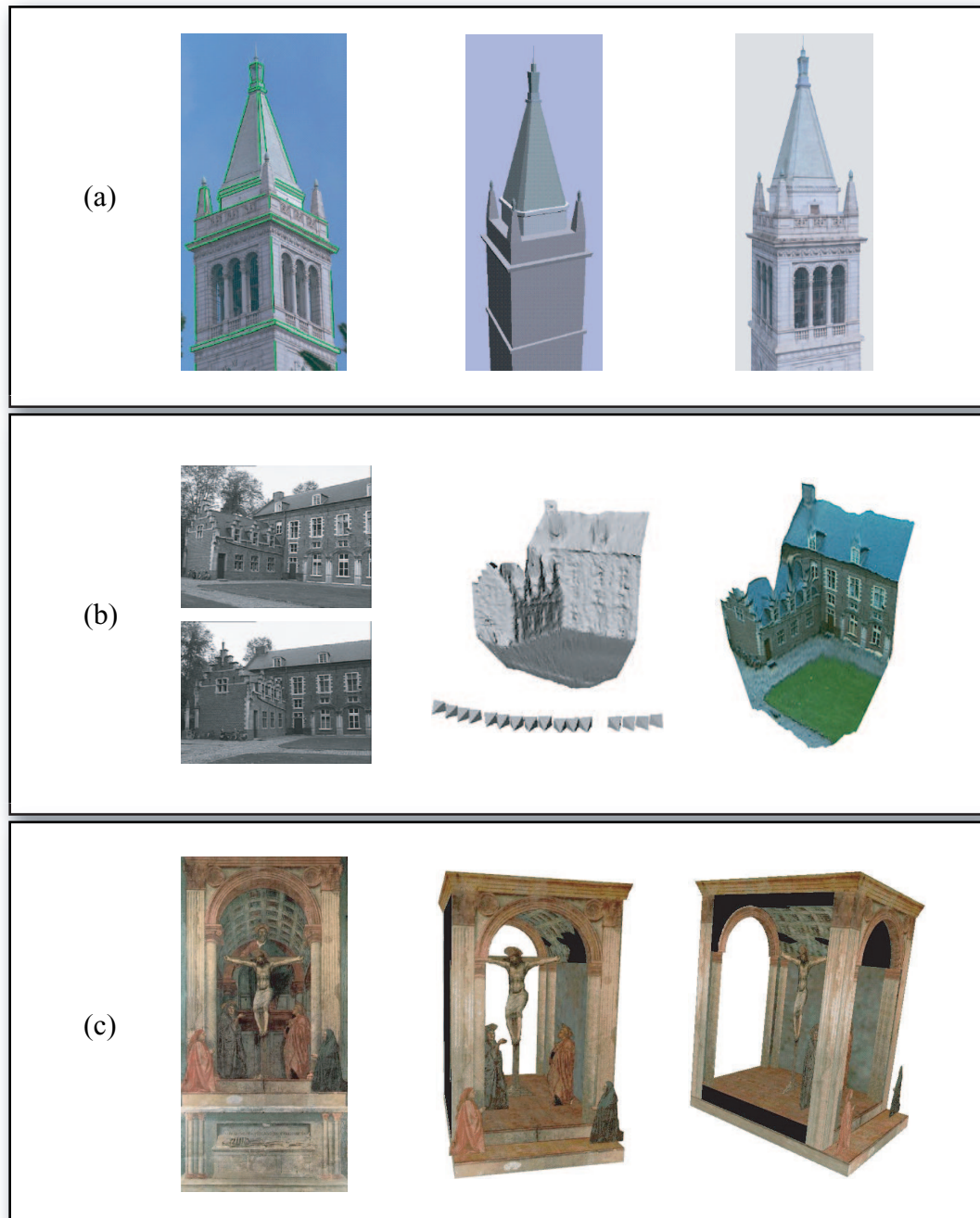


Figure 1.1: Several approaches to image-based modeling. A user-driven system, Facade, is shown in (a). A fully automated, computer vision approach is shown in (b). Only a single image (in this case a painting) is used to create the model shown in (c).

in the quality of reconstructed geometry, shortcomings in geometric accuracy can often be masked by detailed textures[Hec86] derived from the imaged environment.

In spite of the promising progress in image-based modeling, however, these models, and especially the textures applied to their surfaces, suffer from several severe sampling-related problems. In general an image from a single camera position yields an uneven sampling of the texture across the surface of an object, even though the sampling rate is constant across the image plane. This effect is due to the camera pose and position with respect to each surface in the scene, as well as the effects of projection and lens distortion, and is demonstrated in Figure 1.2. As the distance between the location of samples in the world gets larger, the sampling rate gets smaller (see Figure 1.2, top row). The result is that the minimum resolvable feature size becomes larger. The visual effect of this is progressive blurring, as shown in the bottom row of Figure 1.2. This degradation is obviously undesirable in image-based textures.

Current approaches attempt to improve the appearance of image-based models either by merging textures extracted from several different images of the scene from different viewpoints, or by constructing an alternative surface appearance model such as a View-dependent Texture Map [DTM96, DYB98] or a Surface Light Field [WAA⁺00]. These approaches often require many more images than are required to model the scene geometry. Additionally, the requirement for a well-distributed and often dense sampling of the desired rendering viewpoints imposes a heavy and often impossible burden on the image capture process. For example, these techniques to accurately represent a tall building would require images of the top of the building facade. However, as stated previously, using fewer images leaves the surface texture sampled in a highly non-uniform manner, resulting in some

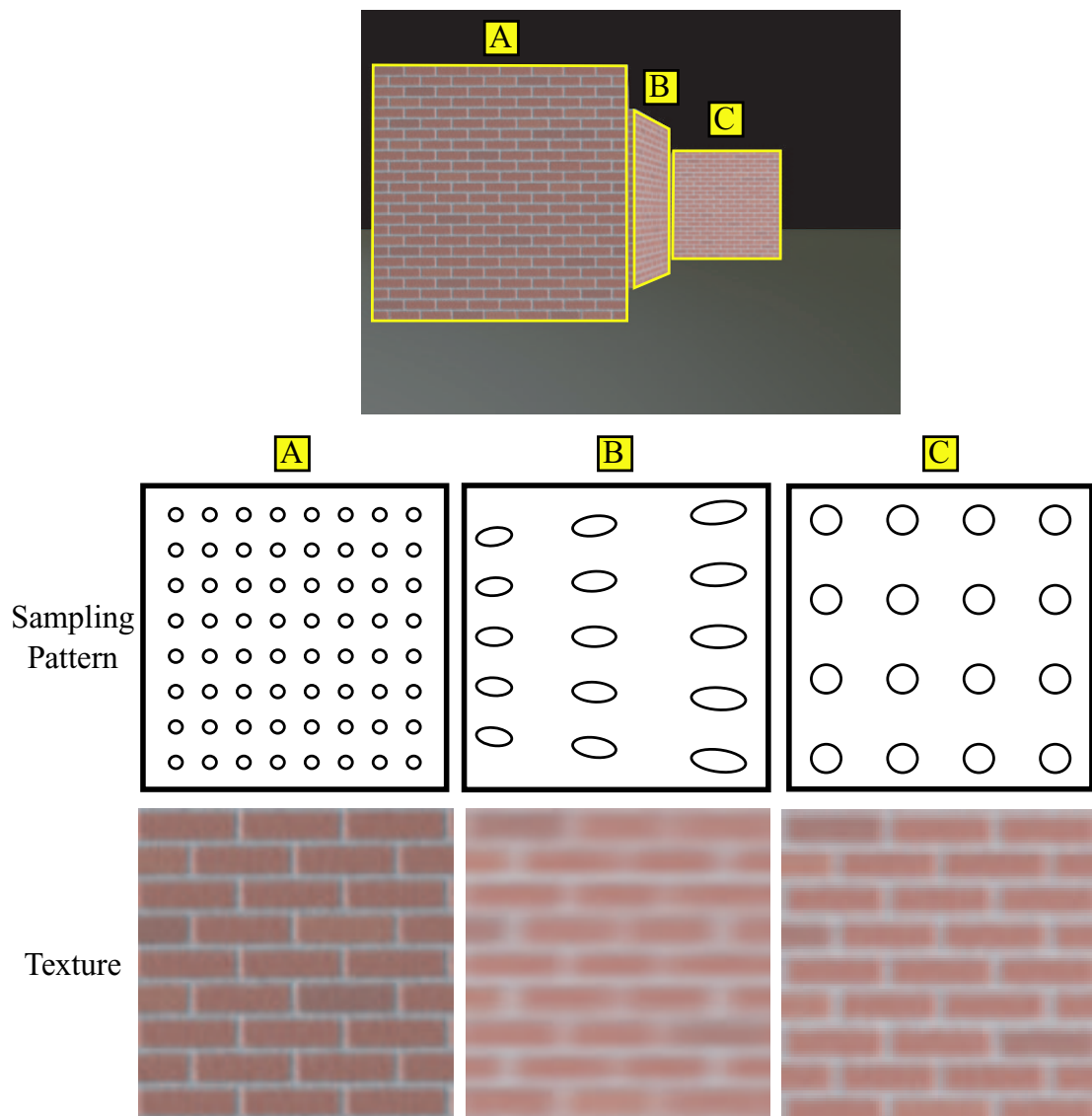


Figure 1.2: Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the changing sampling pattern over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures.

areas with high detail, but leaving most areas with obvious degradation and visual defects.

1.2 Our work

This thesis addresses the problem of poor sampling in two ways; first we identify a physically-based metric of sampling quality. Then we show how to extend existing texture synthesis algorithms to generate the missing detail, injecting statistically correct high frequency information into areas with poor detail, yet preserving any detail present.

The sampling metric is based on and captures the interaction of imaging systems with the imaged environment, and thus quantifies the degradation present in textures resampled from images. Our metric uses the Jacobian matrix of the imaging transform – the transform which maps the surface of an object into the image from which a texture will be extracted. The bottom row of Figure 1.2 shows portions of textures extracted from an image. Note that the quality of the extracted textures varies widely. As discussed previously, these quality variations are due to the way in which the surfaces were sampled. The Jacobian-based metric enables us to quantify the sampling behavior across texture space and leads directly to a physical interpretation of the multi-resolution image representations widely used in texture synthesis (a relationship we discuss in Chapter 5). Using this metric, we quantify which parts of a texture are poorly sampled. We then adapt conventional texture synthesis techniques to populate empty high spatial frequency bands with detail, while preserving any existing low frequency texture data captured by the imaging process. Our technique allows the creation of models with uniform, high-resolution textures from small input image sets, and perhaps more significantly,

from sets which are poorly distributed in the space of desired viewing positions. Generating textures with these qualities constitutes a major improvement in the ability to generate highly detailed, reality-preserving models with reasonable input constraints for users.

The remainder of this thesis is organized as follows. Chapter 2 is a discussion of previous related work. Chapter 3 examines the sampling behavior of image sensors and discusses sensor calibration, laying the groundwork for the derivation of the metric presented and demonstrated in the following chapters. Chapter 4 describes the use of the Jacobian as a sampling quality metric. Chapter 5 describes how we apply the metric to the task of detail synthesis and presents the results of that technique. Chapter 6 concludes with a discussion of future work.

Chapter 2

Related Work

This chapter will review work with immediate relevance to our research focus. The research presented in this thesis is related to previous research in both texture synthesis and image-based texturing (IBT). We discuss work in these two areas below.

2.1 Texture synthesis

Texture synthesis based on Markov Random Fields (MRF) has been studied extensively from the standpoint of generating an arbitrarily sized texture patch from a small example patch. Under Markov Random Field theory, spatial properties of image pixels can be modelled through contextual constraints. This is achieved by characterizing mutual relationships among such entities, i.e. pixel neighborhoods, using conditional MRF distributions. This technique relies on the assumption that the process which generated the texture is both stationary and local (that is, the pixel is determined solely by its neighboring pixels, not by other factors such as proximity to an image edge, etc). In order to capture both small and large scale phenomena, texture synthesis algorithms based on MRFs generally use an image pyramid, in which a pixel's neighborhoods at higher levels represent larger features than similarly sized pixel neighborhoods at lower levels.

Heeger and Bergen [HB95] and DeBonet [DeB97] presented the basic approach using steerable pyramids¹ [KS96]. Efros and Leung [EL99] demonstrated synthesis

¹A steerable pyramid is a linear, multi-scale, multi-orientation image decomposition. The basis functions for a steerable pyramid are directional derivative operators at different sizes and orientations.

with Gaussian pyramids and Wei and Levoy [WL00] introduced a Tree-structured Vector Quantization accelerated search method. These approaches have not been directly applied to image-based texturing.

Freeman et al.[FJP02] describe a synthesis approach for sharpening images. They demonstrate synthesis of the only highest single octave² of data, and their technique is restricted to image space. A texture metric similar to the one presented in this paper would be complementary to their approach and provide a degradation model suitable for extending their training based approach to 3D surfaces.

Zalesny and Van Gool[ZG01] demonstrate a method of synthesizing oriented textures for surfaces, but their approach does not model texture quality, and thus replaces the low resolution texture, rather than using it to guide the synthesis.

The detail synthesis approach introduced in Chapter 5 builds on these algorithms (primarily Wei and Levoy [WL00]) to allow higher quality image-based texturing on 3D surfaces using real camera information. Our algorithm synthesizes only the detail missing from the original images, not the entire texture, thus preserving any data from the images that was correctly captured.

2.2 Image-based texturing

It is often useful to discuss image-based texturing research with respect to the scale of the objects of interest: small- to medium- scale objects (including items suitable for inspection on a turntable such as toys [LHS00], as well as larger items such as statuary [BMR01]); and large-scale objects such as buildings [Deb96]. Although we are interested in applying the technique presented in this paper exclusively

²An octave is an interval between two frequencies which have a ratio of two to one (2:1).

to the last category, we briefly review relevant work in the former category as applicable. Finally, we discuss the various texture quality metrics currently used by image-based texturing techniques.

2.2.1 Techniques for small- and medium-scale objects

Several approaches have been proposed recently for extracting textures from multiple images [BMR01, LHS00, NK99, RCM99]. These techniques do not perform detail synthesis, but instead merge (blend) the contributions from several images using quality metrics (discussed below). These methods require complete and dense image coverage to ensure that for every texture patch there exists at least one image that captures the desired level of detail. To date these approaches have primarily been applied to small- and medium-scale objects.

Surface Light Fields[WAA⁺00], and Bi-directional Texture Functions[LYS01] (BTFs) are methods for representing the view- and illumination-dependent aspects of the appearance of the surface of an object. Both methods construct a higher order model of surface appearance that is continuous with respect to both the view direction and the illumination angle, based on a discrete number of samples in each domain. These methods can be used to create and render attractive surface appearance, but generally require many more images than are needed to model the geometry. Surface Light Fields already require very large, dense data sets, making their extension to large scale objects and environments uncertain. Although BTFs have been demonstrated using smaller numbers of input images for small images, their robustness under this constraint for large-scale objects has yet to be tested.

2.2.2 Techniques for large-scale objects

For large-scale objects, View-dependent Texture Maps[DTM96, DYB98] (VDTMs) have been proposed as a way to improve the appearance of these often complex surfaces. A VDTM is a texture which is generated for a viewpoint on an as-needed basis, blending contributions from different images based on viewing angle. VDTMs tend to capture the view-dependent aspects of surfaces, such as specularly, quite well – so long as the illumination direction constant. But VDTMs inherently require a large number of images, much like Surface Light Fields and BTFs. VDTMs often require twice as many images to texture as to model[Deb96]. Also, like Surface Light Fields and BTFs, VDTMs require that the space of possible rendering viewpoints be fairly evenly and densely sampled. This can be an unreasonable burden on the modeling process, as it may be impossible to acquire enough images to create a robust surface appearance representation for all of the geometry. For example, the top of a facade of a tall building or for the sides of many structures in constricted urban settings would be exceedingly difficult to capture.

2.2.3 Texture quality metrics

Texture quality metrics are often used by image-based texturing techniques that need to select or weight contributions to a texture from pixels in several images. A good quality metric should account for all of the geometric properties of the scene and camera which affect the sampling behavior of the scene. That is, the metric should account for the relative difference in pose and position of the camera, with respect to each object in the environment, including depth, viewing angle, and camera properties such as field-of-view and lens distortion (if present). Note that

the following common metrics only approximate the dominant (or easily computed) properties affecting texture quality.

Image-based texturing methods such as VDTMs, Surface Light Fields, and BTFs apply a texture quality metric involving the viewing angle. As noted by Debevec et al.[DTM96], the viewing angle metric is invariant with respect to scene depth, and thus will not differentiate between two images from the same camera angle but different depths. Several of the approaches which simply merge images[LHS00, RCM99] also use the same metric.

Bernardini et al. [BMR01] use the viewing angle divided by depth as a metric. This approach can be used to evaluate two texture patches with either viewing angle or depth held constant, but may not perform correctly where both vary. It is also difficult to formulate a direct physical interpretation for this metric in terms of sampling behavior.

Ofek et al.[OSRW97] and Neugebauer and Klein[NK99] use projected pixel area to evaluate texture quality. This metric has an obvious physical interpretation, but it does not contain any information regarding the anisotropic behavior of the sampling without making assumptions about the shape of the projected area. We find that this information is required for texture quality analysis.

In contrast to the above approaches, the sampling quality metric introduced in this thesis accounts for all of the geometric factors which affect the spatially-varying sampling rate of the image sensor, while also preserving information on the orientation and eccentricity of the samples.

Chapter 3

Digital Image System Calibration

The purpose of this chapter is twofold. First, this chapter explains the need for a metric describing the performance of imaging sensors commonly used for image-based computer graphics systems, based on sampling theory. Second, this chapter provides an analysis of the factors ultimately affecting sensor performance. These factors must be properly calibrated in order to formulate the sampling behavior metric which is presented in Chapter 4. The characterization of imaging systems provided in the current chapter is specific to consumer or professional grade digital camera systems. These cameras are inexpensive and readily available, making them ideal for image-based computer graphics. We begin by describing digital imaging systems as sampling systems, then discuss their physical properties as they relate to image formation. Last, we last state explicitly the camera model assumptions in place for the remainder of this thesis.

3.1 Digital imaging systems as sampling systems

Considering digital imaging systems as sampling systems is *the* critical perspective used in this thesis. Working within this paradigm determines which problems in the domain are interesting; likewise it directs our search for solutions to these problems. We begin with a basic introduction to sampling theory, and then use the sampling framework to examine the problems inherent in the process of imaging the environment.

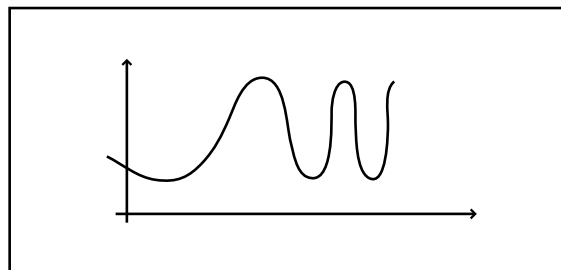
3.1.1 Sampling basics

This section provides a very brief introduction to sampling theory. The reader is directed to excellent works by Holst[Hol98, Hol00] and Smith[Smi97] for a more comprehensive treatment of the concepts covered in this section.

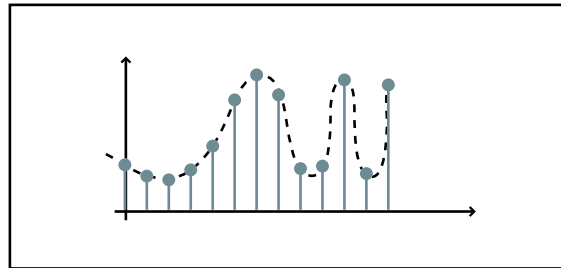
Sampling is the process of digitizing an analog signal. This can be performed for a variety of reasons, including but not limited to ease of storage, compression, reproduction, and processing. These applications are referred to collectively as *digital signal processing*. All digital signal processing applications are concerned with the central question of how many digital values are required to reproduce the analog signal with a minimum loss of information. This question is answered by Shannon's Sampling Theorem[Sha49].

Shannon's Sampling Theorem (often called the Shannon-Nyquist Theorem, the Whittaker-Shannon Theorem, or simply the sampling theorem) states that a time- or space-varying signal is completely determined by samples taken $\frac{1}{2f_{MAX}}$ units apart, if and only if the signal contains no frequencies higher than f_{MAX} . The signal can then be reconstructed from these samples by using an ideal low-pass filter as a reconstruction filter. The frequency f_{MAX} is also known as the *Nyquist frequency*, designated f_N , and it is the highest frequency that can be reconstructed without aliasing from a signal sampled at a sampling rate of $2f_N$. Observe that for the sampling theorem to be applicable, the signal must satisfy two related conditions: it must be band-limited to f_{MAX} , and it must be sampled at $\geq 2f_{MAX}$.

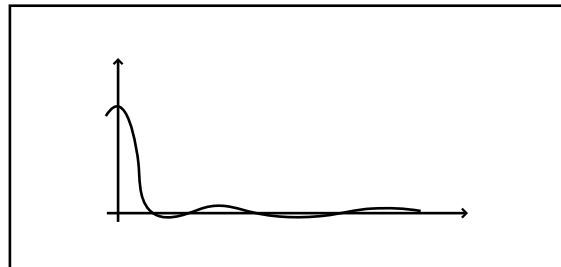
Consider for a moment the condition illustrated in Figure 3.1. The original analog signal is shown in Figure 3.1 (a). The sampled signal is shown in part (b). When reconstructed with the ideal low pass filter (the sinc function) shown in (c), the reconstructed signal (d) is identical to the original signal. Note that



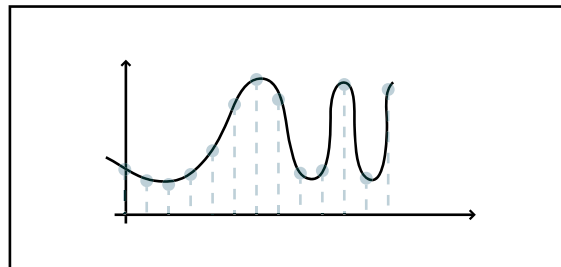
(a)



(b)



(c)



(d)

Figure 3.1: (a) shows the original analog signal. The signal is sampled and the digitized values are shown in (b). When reconstructed with the sinc filter shown in (c), the reconstructed signal (d) is identical to the original signal.

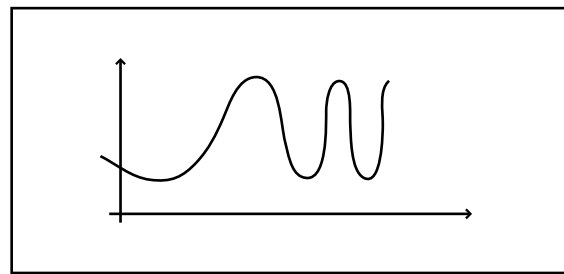
even the higher frequencies toward the right-hand side of the original signal are reconstructed correctly.

In contrast, Figure 3.2 illustrates the reconstruction of the same signal using a sampling rate that is too low. The original signal – identical to that in Figure 3.1 – is shown in (a). The sampled signal is shown in (b). Observe that the samples are approximately twice as far apart as in Figure 3.1; that is, the sampling rate in Figure 3.2 is approximately half that shown in the previous figure. When reconstructed with the low pass filter shown in (c), the reconstruction is *not* identical to the original signal. Specifically, the higher frequencies in the right hand portion of the original signal have been aliased to lower frequencies.

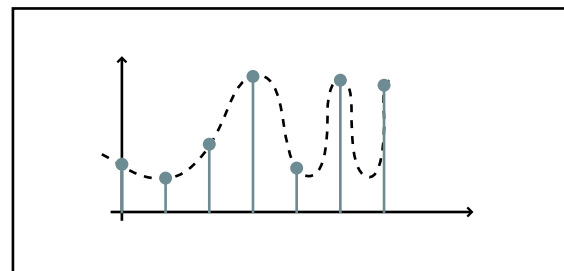
The limitations on sampling-based reconstruction described by Shannon’s Theorem point to our primary concern with the process of image formation by digital image sensors – namely, how does the information content of the reconstructed signal (the image) differ from the information content of the original signal (the environment)? This question is of especially pressing relevance for image-based modeling and texturing techniques, the strength of which is their proposed ability to capture environmental complexity that would otherwise be difficult or impossible to model. Characterizing the extent to which these techniques are capable of representing this complexity is one of the primary goals of this thesis. Next, we examine in greater depth the sampling theorem’s constraints on the frequency content of the signal and on the rate of sampling, in the context of image formation.

3.1.2 Environmental considerations

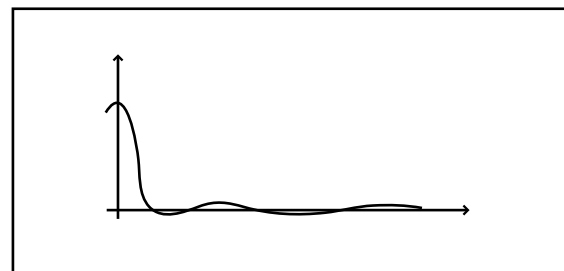
The sampling theorem requires that the signal being sampled be band-limited in order to be reconstructable. However – for imaged environments – this condition



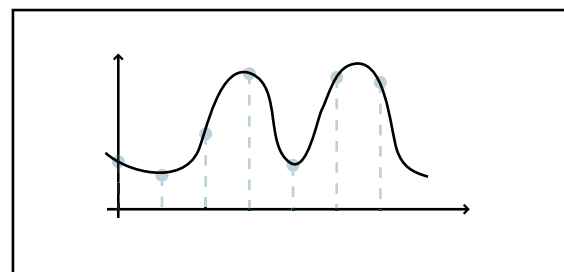
(a)



(b)



(c)



(d)

Figure 3.2: (a) shows the original analog signal. The signal is sampled and the digitized values are shown in (b). When reconstructed with the sinc filter shown in (c), the higher frequency portion of the signal has aliased to a lower frequency, and the reconstruction in (d) is significantly different from the original signal.

simply does not exist: the imaged environment has infinite bandwidth.

This characteristic is particularly relevant to the built environment, where square waves (which require an infinite number of sinusoids to reconstruct) are prevalent. Most scenes contain regions of increasing fine detail visible under magnification. The implication is that *no* digitization will ever yield a perfect reconstruction of even the visible environment. No amount or combination of laser scanning, high resolution imagery, radar range data, or other sensor data will ever permit this perfect reconstruction.

In practice, this is often not perceived to be a limiting factor. Although there is some lower limit to the level of detail that can be captured, there is also often a lower limit to the size of details of interest. As long as the sensor can properly sample objects or features of this size, then the digitization of the environment is sufficient for the given application. The question remains whether the physical properties of the image sensor permit sufficient performance to capture the requisite level of detail.

3.1.3 Imaging sensor considerations

In order to determine what quality of reconstruction it is possible to achieve, we must understand the bandwidth limitations of digital image sensors. The next section will examine the physical properties of sensor systems that affect these bandwidth limitations. These properties have an impact on both the Nyquist frequency (the stop-band) of the sensor, as well as the way that the data that *is* captured is modified. In the latter case, the modified data must either be corrected through proper calibration or accounted for explicitly in the system assumptions. Factors which impact the sensor stop-band must be calibrated because they figure

prominently into the sampling metric presented in the next chapter to measure the sensor performance over the environment. The remainder of this chapter examines the calibration of the relevant physical properties of image sensors.

3.2 Properties of digital imaging systems

Several complex and often interrelated factors affect the sampling behavior of the image sensor. These factors need to be calibrated in order to properly account for data modification and to initialize the sampling behavior metric described in Chapter 4. In addition to the camera's sensor resolution, the camera's intrinsic parameters, such as the focal length, optical center, and pixel shape, must be calibrated in order to characterize its projective behavior. Closely related to these parameters is the geometric lens distortion to which images are subject; this behavior will also be analyzed, along with the linearity of the sensor's intensity response, and any discontinuous spatial response such as demosaicing.

3.2.1 Sensor resolution

The resolution of the image sensor has an obvious impact on frequencies that the sensor can record. As a practical note, not all of the sensor elements on a chip may be in the camera's imageable area. If a particular chip has an imageable area of 1600×1200 sensor elements, with 19nm center-to-center element spacing, then the highest frequency that can be detected is $\frac{1}{38}$ cycles per nm. Observe that this frequency measurement is on the sensor surface; that is, $\frac{1}{38}$ cycles per nm is the limit of observable frequencies *after all projective properties of the system have been taken into account*. This value on its own is of limited use, since we want the analysis to be with respect to a three-dimensional environment, not a

two-dimensional sensor surface. The projective properties, as expressed in the calibrated camera model, are described next.

3.2.2 The camera model

We model the intrinsic parameters of the camera using a special case of the camera model utilized by the Camera Calibration Toolkit for Matlab [Cal]. The general camera model has the following parameters:

| | |
|----------|--|
| fc_x | Focal length in units of horizontal pixels |
| fc_y | Focal length in units of vertical pixels |
| α | Angle between the x and y sensor axes |
| p_x | The x coordinate of the center of projection |
| p_y | The y coordinate of the center of projection |

We can make several assumptions to simplify the camera model significantly. All modern image sensor fabrication processes guarantee sensor elements that are equal dimensions in the horizontal and vertical directions and have zero skew. Thus, $fc_x = fc_y$ and $\alpha = 0$. We can express this simplified camera model as a projection matrix of the form:

$$\mathbf{M} = \begin{bmatrix} fc & 0 & p_x & 0 \\ 0 & fc & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.1)$$

Note that these intrinsic parameters are based on this one specific pinhole camera model. Several alternate pinhole camera models have been proposed [Tsa87, Zha99, HS97]. These models are all equivalent to each other, and to the general Toolkit camera model, under various sets of simplifying assumptions.

Camera models other than the pinhole model have been proposed, such as the thin lens model [BW93] and Willson’s automated zoom lens model [Wil94]. In general, these non-pinhole models are much more difficult to calibrate and provide little benefit for image-based computer graphics applications. Projectively, the difference between an actual digital camera with a compound lens assembly and the pinhole model is negligible, and is a difference which most common lenses are manufactured to minimize.

3.2.3 Lens distortion

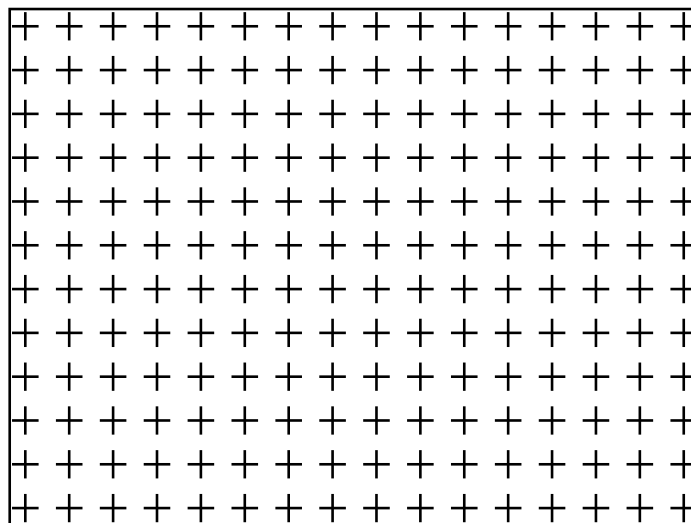
Geometric distortion of the image is caused by the curvature of optical elements in a compound lens assembly. Most modern lenses exhibit little or no geometric lens distortion, except at very short focal lengths. However, because digital cameras often have an effective focal length multiplier¹, these short focal length lenses may be used more frequently for digital photography. This is especially true for imaging the built environment, where a wide field of view may be necessary in order to photograph a complete structure without obstruction.

Geometric distortion can be modelled in many ways. The work in this thesis uses the “Plumb Bob” (radial polynomial + ”thin prism”) distortion model introduced by Brown [Bro66]. This model consists of both radial and tangential components and is defined by:

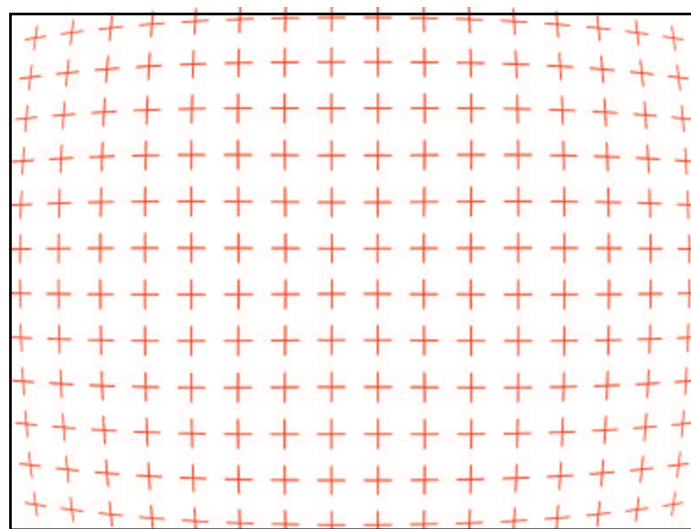
$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = (1 + R_2r^2 + R_4r^4 + R_6r^6) \begin{pmatrix} x \\ y \end{pmatrix} + dx \quad (3.2)$$

Where $r^2 = \sqrt{x^2 + y^2}$, and dx is the tangential distortion vector:

¹The focal length multiplier is due to the fact that most image sensors do not fill the entire 35mm film frame.



(a)



(b)

Figure 3.3: (a) shows the original calibration target. The distorted target is shown in (b). Note the strong curvature to previously orthogonal rows and columns at the edges of the target.

$$dx = \begin{pmatrix} T_x(r^2 + 2x^2) + 2T_yxy \\ 2T_xxy + T_y(r^2 + 2y^2) \end{pmatrix} \quad (3.3)$$

The tangential distortion is due to "decentering", or imperfect centering of the lens components and other manufacturing defects in a compound lens. The five parameters to the model are:

| | |
|-------|---|
| T_x | Tangential distortion 'horizontal' coefficient |
| T_y | Tangential distortion 'vertical' coefficient |
| R_2 | 2 nd order radial distortion coefficient |
| R_4 | 4 th order radial distortion coefficient |
| R_6 | 6 th order radial distortion coefficient |

Figure 3.3 shows a simple calibration target distorted using this distortion model with $T_x = T_y = 0$, $R_2 = -.7$, $R_4 = .05$, and $R_6 = 0.0$. Figure 3.3 (a) shows the calibration target before distortion, and (b) shows the same calibration target after distortion. Notice that the effect grows with the distance from the optical center, and that at the edges the curvature is quite noticeable.

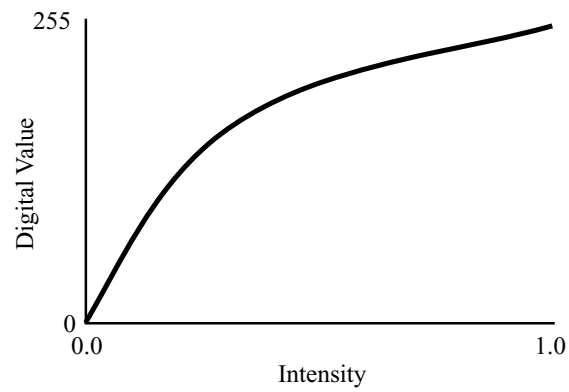
In practice, calibration of the first four parameters is sufficient to capture the behavior of the distortion. It is generally unnecessary to calibrate the 6th order coefficient.

3.2.4 Opto-electronic Conversion Function

The opto-electronic conversion function (OECF) defines how intensity values from the environment are mapped by the sensor to digital values in the image. Figure 3.4 shows two example OECFs for a hypothetical, 8-bit, single channel image



(a)



(b)

Figure 3.4: The profile in (a) shows an approximately linear OECF. Digital values are proportional to intensity values. The profile in (b) shows the OECF for a gamma-corrected image. In this case, digital values are approximately logarithmic with respect to intensity.

sensor. In Figure 3.4 (a), the OECF is linear, that is, the digital values are proportional to the scene intensities. Figure 3.4 (b) shows the OECF for a typical, gamma corrected image. This OECF profile shows that the digital values are approximately logarithmic with respect to intensity. Early image sensors and some modern scientific sensors exhibit strongly non-linear OECFs, usually with approximately logarithmic behavior. This behavior can be beneficial for applications that require images with a large dynamic range.

Almost all image sensors used in modern consumer or professional digital cameras are now approximately linear on the chip. However, calibration of the OECF is still necessary for several reasons. First, there is still some slight variation from true linearity, which may prove significant for some applications. Second, not all digital cameras provide access to the raw sensor values. Default image processing on most digital cameras will typically gamma-correct the image intensities, in order to make the final image appear correctly displayed on a computer monitor. Calibration of the OECF permits quick and easy linearization of an image.

3.2.5 Demosaicing

Demosaicing is the process by which the sensor interpolates between red, green, and blue sensor sites on a Bayer array to arrive at an RGB value for each pixel. Modern demosaicing approaches are difficult or impossible to calibrate: they tend to be highly non-linear, proprietary algorithms, and are generally kept as trade secrets by camera manufacturers. To the best of our knowledge, no current calibration work attempts to take the effects of the demosaicing algorithm into account.

Any concern about the effects of demosaicing may be short-lived, given the state-of-the-art in image sensor design. At least one company, Foveon, has suc-

ceeded in producing a CMOS sensor chip that captures RGB at each sensor site, and which thus uses no demosaicing algorithm. Because use of the Foveon sensors is not yet widespread, analyses of image sensors generally assumes that the demosaicing algorithm ‘does the right thing’: that is, demosaicing does not modify the information content of the scene, and thus the RGB values at a pixel are true samples of the environment.

3.3 Calibration assumptions

Calibrated parameters for sensor resolution, projective behavior (the intrinsic parameters), and lens distortion will be used to formulate the metric in the next chapter, under the assumption that a pinhole camera model with Plumb Bob distortion is a sufficient projective model. These are the factors that directly affect the upper limit of the frequencies that the sensor is able to sample in the environment. Furthermore, we assume that the OECF is linear, or the image has been linearized using the calibrated OECF. We use the standard assumption that demosaicing neither creates nor destroys information.

The sensor characteristics discussed in this chapter can be calibrated using a number of different methods. This thesis uses the Camera Calibration Toolkit for Matlab [Cal] to calibrate the intrinsic parameters, and HDRShop [HDR] to verify sensor linearity. Using the calibration information and the above assumptions the next chapter provides a physical metric to answer the question of what information the image sensor can capture from the three-dimensional environment.

Chapter 4

Sampling Quality

This chapter describes the problem of characterizing the sampling behavior of an imaging sensor during the image formation process. It then explains how the values of the Jacobian matrix of the imaging transform can be interpreted as sample distances within the environment, and further how the Jacobian values can serve as a measure of image-based texture quality.

Recall from Section 3.1 that every imaging system modifies and removes some – arguably most – of the information in the environment during the imaging process. The key to analyzing how an imaging system modifies the information content of a given environment is to determine how the limit of the system passband – the set of observable spatial frequencies – varies across the environment. The passband limit (the Nyquist frequency) is the primary performance parameter that varies across the environmental geometry¹.

The behavior of the imaging system with respect to the observable spatial frequencies in the environment is of broad concern in computer graphics. This behavior is of theoretical import to a variety of research areas, such as level-of-detail management, material measurement, geometric reconstruction, and image-based modeling and rendering – all of which deal in some way with the quality of data (usually visual detail) observed in some environment by some sensor.

This thesis presents the task of reconstructing surface textures by resampling an image as the motivating example, demonstrating the importance of a physically-

¹As noted in Section 3.2.5, the work in this thesis will assume that the demosaicing algorithm, when present, does not modify the information content of the environment.

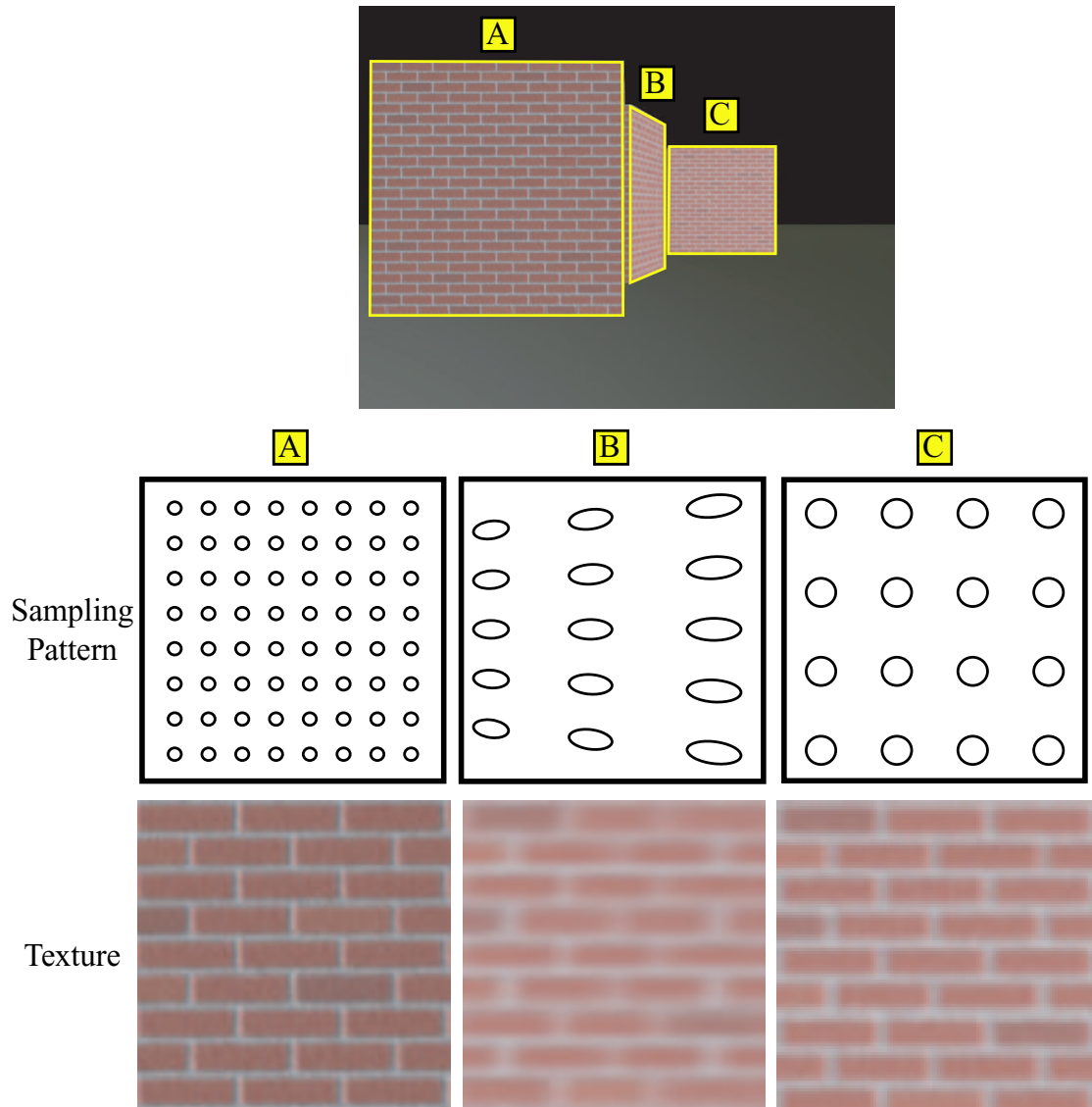


Figure 4.1: Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the sampling patterns over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures.

based sampling metric. The sampling metric described in this chapter provides a way to quantify the physical effects which lead to the blurring or stretching in poor sampled textures. Figure 4.1 shows these sampling effects. The top image in Figure 4.1 shows the scene. The middle row illustrates the sampling patterns over the three faces of interest. The characteristic sampling patterns on the three faces are easily observed, as are the resulting visual effects in the reconstructed textures, shown in the bottom row.

Application of the proposed metric makes it possible to correct the degradation in the reconstructed textures, as demonstrated in the following chapter. This metric embodies a new approach to quantifying the sampling behavior of imaging systems in computer graphics. To the best of our knowledge, it is the only attempt which uses a physically-derived, analytical approach to measuring sampling quality. The benefits of this approach are demonstrated in Section 4.5.

The following discussion is based on the observation that both image formation and image-based texturing are sampling problems – in fact in some sense they are inverse problems. The image formation process samples the environment; image-based texturing resamples the image to reconstruct the texture for the surface of an object in the environment. Both processes are discussed in more depth below.

4.1 Image formation

During the image formation process, an imaging sensor (typically a digital camera) samples a scene to produce a digital *image* – a two-dimensional array of data containing intensity values on a regular grid. These images may be either gray-scale or color, so each sample may contain, respectively, a single value or an RGB tuple. We refer to the individual samples in the image as *sensor elements* or

sensels, in order to emphasize that they correspond to physical sites on the sensor chip. We describe sensel positions by using coordinates in *image-space*, a discrete two-dimensional space parameterized in (u, v) , and we assume that the origin of image-space is at the top left of the image. We further assume that image-space has horizontal and vertical extents equal to the horizontal and vertical resolution of the digital image, i.e. u and v range from 0 to $\langle image_width \rangle$ and 0 to $\langle image_height \rangle$, respectively.

Recall that the problem which is being addressed is that of characterizing the spatial frequency response of an imaging sensor. We wish to determine how the limit of the observable spatial frequencies (the passband) varies across the environment². By focusing on the limit of the passband, we risk neglecting the behavior of the MTF within this band. However, the MTF can be quickly determined by either computing the projective warping of the sensor PSF or by estimating of the MTF using the sensor MTF and the spatially-varying Nyquist frequency. This chapter focuses exclusively on determining this latter quantity.

In order to determine which spatial frequencies in the environment can be captured physically by the image sensor, the centers of adjacent sensels are projected into the environment. The distance between the projected pixel centers in the environment is measured in world units. This quantity is the *sample distance*. More frequently we refer to the sampling behavior in terms of the *sampling rate*, which is the quantity $\frac{1}{sample\ distance}$ cycles per world unit. Recall that basic sampling theory (see Section 3.1.1) tells us that the highest spatial frequency that can

²This is independent from the actual spatial frequency composition of the environment. For example, a ‘boring’ part of the environment may contain only spatial frequencies far lower than the camera’s physical upper limit. On the other hand, no portion of the image can capture spatial frequencies higher than those dictated by this upper limit, even if such frequencies are present in the environment.

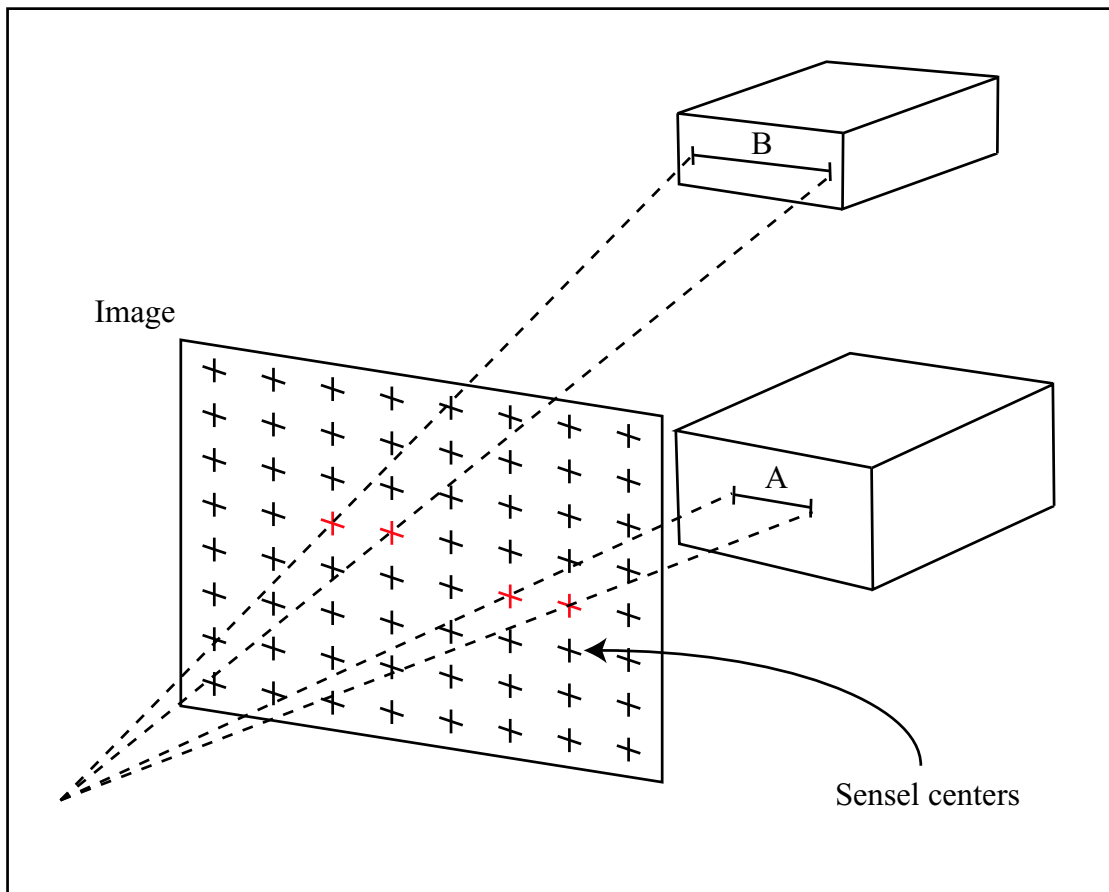
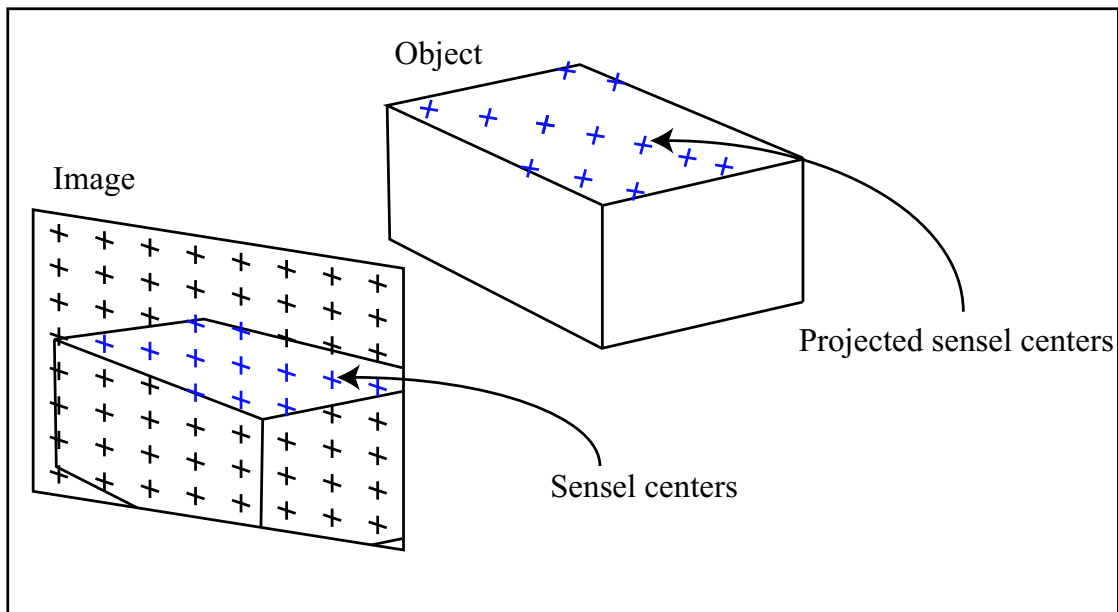


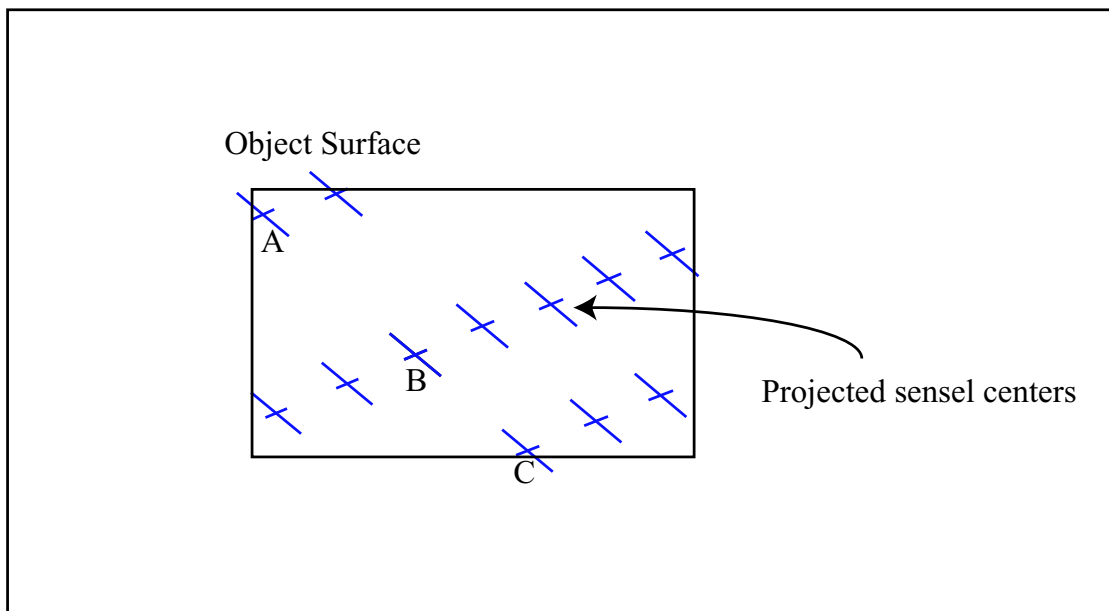
Figure 4.2: An illustration of spatially varying Nyquist frequency, f_N , due to perspective projection. In this example the sample distance A is half the length of the sample distance B . Thus, the $f_{N_A} = 2f_{N_B}$.

be captured – the Nyquist frequency, f_N – at a point in the world is $\frac{\text{sampling rate}}{2}$. As the sampling rate varies across the objects in the scene, so does the Nyquist frequency. A simple illustration of this variation is shown in Figure 4.2. Two adjacent sensel centers are projected onto a nearby object, where they are separated by the distance labeled ‘A’. Two adjacent sensel centers from a different part of the image are projected onto a distant object, where they are separated by the distance labeled ‘B’. If $A = .5B$ due to perspective projection, then $f_{N_A} = 2f_{N_B}$.

It is easy to observe that although the sample values recorded by the camera are uniformly distributed in image-space, it is generally not the case that the locations that were sampled by the camera are uniformly distributed in the scene. The non-uniform nature of the sampling is due to the pose and position of the camera relative to the objects in the scene, as well as the effects of perspective and lens distortion. This effect is illustrated in greater detail in Figure 4.3. Figure 4.3 (a) depicts the imaging process, with the sensel centers marked with pluses. The projection of the sensel centers onto an object in the environment is shown. Sensel centers in blue sample the top surface of the object. The projected sensels show that the environment is sampled on a grid that is strongly non-uniform in the world coordinate frame. This is shown more clearly in Figure 4.3 (b), which is an orthographic view of the top surface of the object that is being imaged in Figure 4.3 (a). The non-uniformity of the sampling grid is even more apparent in this view. Observe that the sensels with projections marked A, B, and C all lie adjacent to one another (in the same column) in the image. In the environment, however, the distance from A to B is not equal to the distance from B to C, thus the sample distance and sampling rate vary from A to C.



(a)



(b)

Figure 4.3: An illustration of the camera sampling the environment on a non-uniform sampling grid.

4.2 Texture reconstruction

The previous section described the process by which a camera samples a given environment to form an image. The current section describes the process through which surface textures of a model are reconstructed, given a geometric description and an image of the object. We refer to this process as *image-based texturing*.

We store the appearance of the surface of an object in a *texture*, which is a discrete, two-dimensional array of data parameterized in (s, t) . Data elements in the texture are referred to as *texels*. In order to reconstruct a texture, given an image and a geometric description for the surface of an object, each texel is projected into the image. The values in the image are sampled at the projection locations. This process is illustrated in Figure 4.4 (a). Although the texels are arranged on a regular grid in the surface of the object (or, more precisely, in the space parameterized in (s, t)), the projection of the texels into the image is highly nonregular. This is shown graphically in Figure 4.4 (b), which shows the two-dimensional image-space.

Image formation and image-based texturing are closely related processes. In the most general sense, image formation relies on the mapping $\mathcal{R}_{Img}^2 \rightarrow \mathcal{R}_{Env}^3$, the mapping that projects sensors in the image into the three-dimensional environment. In practice, we are often concerned only with the properties of the projection over the surfaces of objects, not over the empty space in the environment, so we limit the mapping to $\mathcal{R}_{Img}^2 \rightarrow \mathcal{R}_{Surf}^2$, without loss of information. Image-based texturing is concerned with the inverse problem – projecting texels on the surface of an object into an image. Therefore, image-based texturing relies on the mapping $\mathcal{R}_{Tex}^2 \rightarrow \mathcal{R}_{Img}^2$. If we restrict the relationship between surfaces and textures to be one of known uniform scaling, then these are inverse problems, and we have

the liberty of choosing the representation of the problem which is most convenient for the given task at hand. More importantly, this isomorphism reveals why the sampling behavior of the imaging system is the key factor in measuring texture quality.

To illustrate the relationship between sampling behavior and texture quality, consider an image-based texture reconstructed from the imaging configuration in Figure 4.4. Such a texture would exhibit several sampling effects worth noting. At the sampling locations in the region labeled ‘A’ in Figure 4.4 (b), the texture supersamples the image. That is, the texture contains more samples than are necessary to represent the information in the image. At the far right of the image, in the region marked ‘B’, the texture subsamples the image in the horizontal direction. This will result in a directional blur in the reconstructed texture. For further examples of this blurring effect, refer to Figure 4.1.

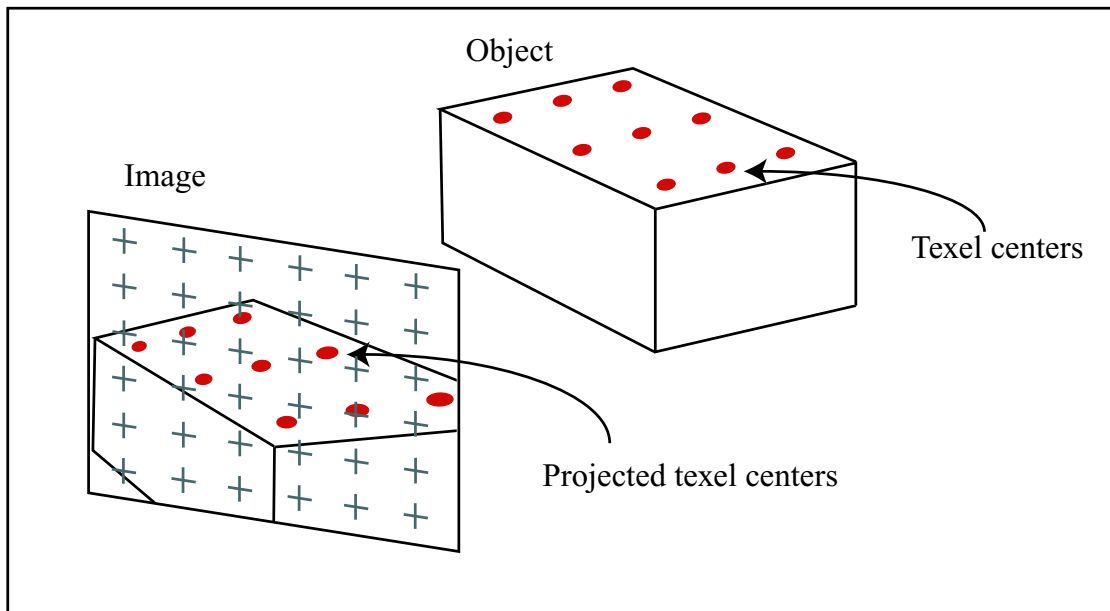
4.3 Formal description of sampling

The mapping $\mathfrak{R}_{img}^2 \rightarrow \mathfrak{R}_{surf}^2$, which projects sensels in the image onto the surfaces of objects in the environment, is now described more formally. In the most general sense, this mapping can be expressed as some invertible function:

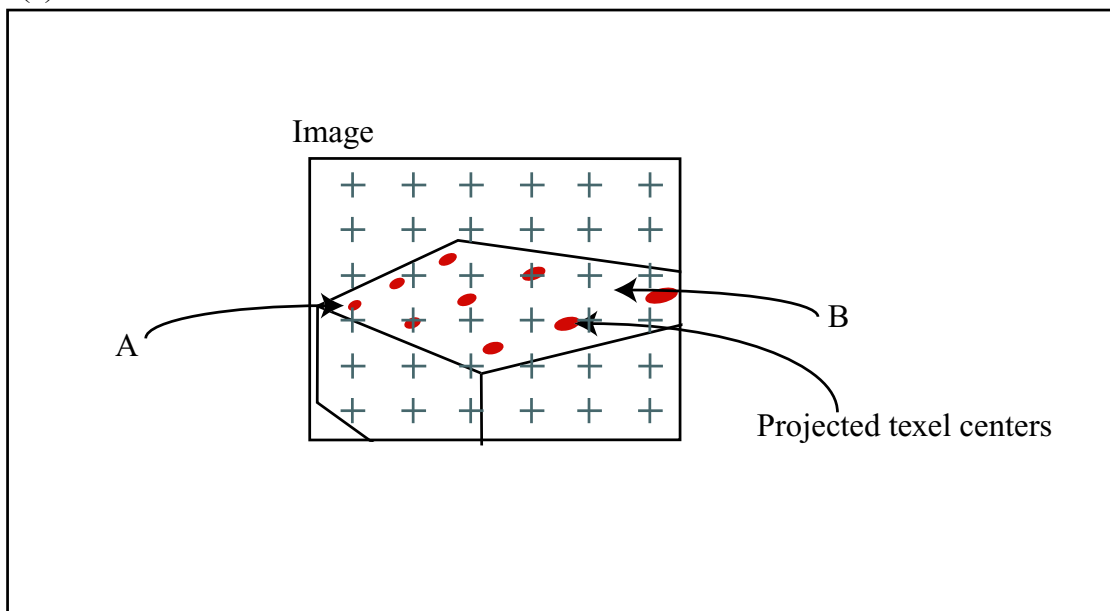
$$\alpha = f(\beta) \tag{4.1}$$

Where α is a vector in (s, t) (on the surface) and β is a vector in (u, v) (in image-space). It is often useful to consider f as composed of several component functions:

$$f = surf \cdot obj \cdot ext \cdot int \cdot dist \tag{4.2}$$



(a)



(b)

Figure 4.4: An illustration of a texture sampling an image on a nonuniform sampling grid.

In this decomposition, *surf* maps three-dimensional points on the surface of the object into the two-dimensional parameterization of object surface. The function *obj* models the metric transform which aligns an object's local coordinate frame to the world coordinate frame. The function *ext* models the metric transform induced by the camera's extrinsic parameters. The function *int* models the perspective transformation induced by the camera's intrinsic parameters. The function *dist* is a function modeling the lens distortion.

If we further consider:

$$f' = tex \cdot f \tag{4.3}$$

Where the function *tex* maps points on the surface of the object into a texture, then we have a decomposition for the mapping $\mathfrak{R}_{Img}^2 \rightarrow \mathfrak{R}_{Tex}^2$. This mapping and its inverse will become very useful shortly.

Consider an alternate formulation for Equation 4.1. The sampling of the scene by the camera (i.e., the image formation process) is often modelled in computer graphics as series of matrix transforms which map textures onto objects, then map objects into the image. Note that this is the actually the inverse of *f'*, and maps $\mathfrak{R}_{Tex}^2 \rightarrow \mathfrak{R}_{Img}^2$. This formulation is convenient given its rough mapping to the stages of the computer graphics hardware rendering pipeline. The series of matrix transforms is called the *imaging transform*, denoted \mathbf{M}_{Img} :

$$\mathbf{M}_{Img} = \mathbf{M}_{Proj} \cdot \mathbf{M}_{Tex} \tag{4.4}$$

Observe that \mathbf{M}_{Img} is composed of two transforms: \mathbf{M}_{Proj} , which projects the surface of an object into the image plane, and \mathbf{M}_{Tex} , which projects the surface texture onto the object surface. This relationship is depicted in Figure 4.5.

\mathbf{M}_{Proj} can be further decomposed as:

$$\mathbf{M}_{Proj} = \mathbf{M}_{Dist}\mathbf{M}_{Int}\mathbf{M}_{Ext}\mathbf{M}_{Obj}\mathbf{M}_{Surf} \quad (4.5)$$

Observe that \mathbf{M}_{Proj} corresponds to f^{-1} , just as \mathbf{M}_{Img} corresponds to f'^{-1} . \mathbf{M}_{Dist} is the lens distortion model matrix. Note that the matrix form of \mathbf{M}_{Dist} limits the distortion model to a linear function. In practice this limitation can be overcome in two ways. First, the functional form for the lens distortion can easily be substituted, since both \mathbf{M}_{Dist} and the function $dist$ map $\mathfrak{R}^2 \rightarrow \mathfrak{R}^2$. Substitution of the functional form allows an arbitrary order distortion model to be used. As an alternative, images can be preprocessed with an arbitrary order image-warping operation derived from calibration data, in order to undistort the images. This is the approach utilized to obtain the results presented in Chapter 5. In this case, \mathbf{M}_{Dist} is simply the identity matrix.

\mathbf{M}_{Int} is the transform associated with the intrinsic camera parameters. The form of \mathbf{M}_{Int} , described in Section 3.2.2, is given again here for convenience:

$$\mathbf{M}_{Int} = \begin{bmatrix} fc & 0 & p_u & 0 \\ 0 & fc & p_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.6)$$

Where fc is the focal length, p_u is the u (horizontal) coordinate of the principal point, and p_v is the v (vertical coordinate of the principal point. The values of fc , p_u , and p_v are obtained during camera calibration.

\mathbf{M}_{Ext} is the transform associated with the the extrinsic camera parameters (the camera pose and position). \mathbf{M}_{Ext} is a constrained metric transform, and takes the form:

$$\mathbf{M}_{Ext} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \quad (4.7)$$

Where R is a 3×3 matrix for the rotation of the camera, and C is a 1×3 vector containing the position of camera center in world coordinate frame. Collectively, \mathbf{M}_{Dist} , \mathbf{M}_{Dist} , and \mathbf{M}_{Dist} model the camera and map the three-dimensional world coordinate frame into two-dimensional image-space.

\mathbf{M}_{Obj} is the transform that maps the object into world-space, for scenes where individual objects are defined in coordinate frames not necessarily coincident with the world coordinate frame. \mathbf{M}_{Obj} is a general metric transform and has the same form as \mathbf{M}_{Ext} , as shown in Equation 4.7.

\mathbf{M}_{Surf} is the transform that maps coordinates on the two-dimensional surface of object to three-dimensional points in the local coordinate system of the object. Note that, in a similar manner to the lens distortion transform, \mathbf{M}_{Dist} , the matrix form of \mathbf{M}_{Surf} constrains the surface parameterization to have a linear form. Again, in practical situations this is easy to overcome by using the functional form *surf*. This allows the use of parametric curved and high order surfaces, instead of piecewise planar regions (such as triangle meshes).

\mathbf{M}_{Tex} models the *texture transform*, which maps a texture onto an object's surface, and which also defines a sampling of the surface with a constant sampling rate. The sampling is constant because we assume that the surface parameterization matches the texture parameterization except for a scaling factor, so \mathbf{M}_{Tex} is simply a uniform scaling (translation and rotation are represented in \mathbf{M}_{Obj} or \mathbf{M}_{Ext} , as appropriate). Adjusting \mathbf{M}_{Tex} adjusts the sample rate of the mapping $\mathfrak{R}_{Surf}^2 \rightarrow \mathfrak{R}_{Img}^2$.

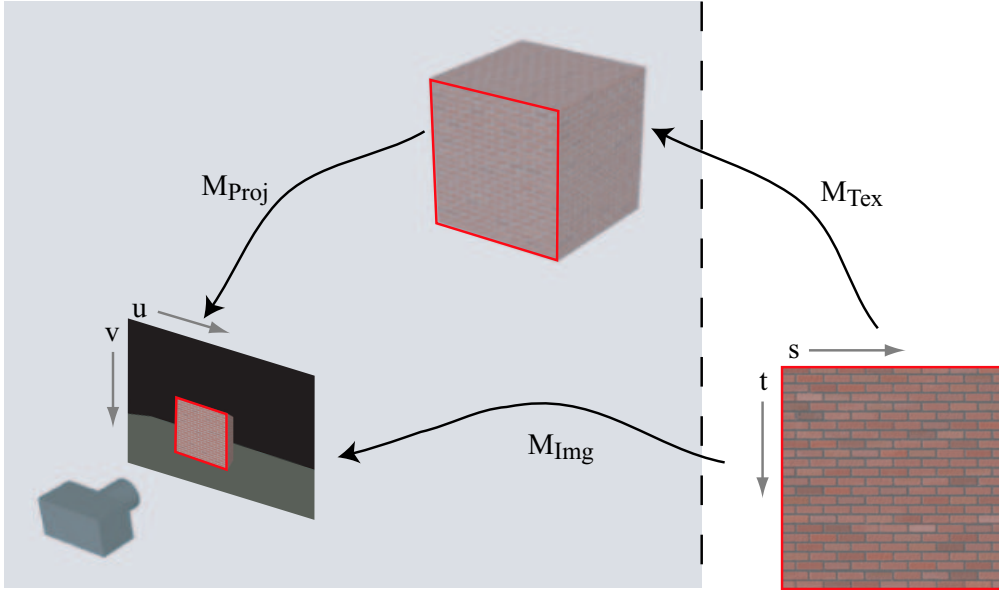


Figure 4.5: Graphical depiction of the relationships between \mathbf{M}_{Proj} , \mathbf{M}_{Img} , and \mathbf{M}_{Tex} .

4.4 Physically-based sampling metric

Optimally, the surfaces of objects in the environment should be sampled at a constant, high rate by the imaging sensor. Under these conditions, high quality image-based textures can be resampled from the image for the visible surfaces in the environment. As has been demonstrated, however, imaging systems do not generally sample the environment in this manner. Thus, in order to assess the degradation in the image-based textures, and to have any hope of reconstructing textures with constant sampling rates, we require a *physical, sampling-based texture quality metric*. The remainder of this chapter explores such a metric.

4.4.1 Sampling in world units

Consider first, only the transform \mathbf{M}_{Proj} . This transform maps some object surface into the image plane. We assume without loss of generality that the surface is parameterized in two-dimensional coordinates (s, t) , since this is required for the texture mapping operation in any case. Points on the surface are described in (homogeneous) coordinates in world units. Thus we have:

$$\beta = \mathbf{M}_{Proj} \cdot \alpha \quad (4.8)$$

where β is now a homogeneous vector in (u, v) and α is a homogeneous vector in (s, t) . This transform is illustrated in Figure 4.5. As noted previously, this is a standard way to express the “imaging” of the environment in computer graphics. If \mathbf{M}_{Proj} projects an object surface into the image, then we are interested in characterizing the inverse process: how does the inverse of \mathbf{M}_{Proj} project the image data onto an object surface? In keeping with this perspective, from this point forward we deal primarily with \mathbf{M}_{Proj}^{-1} .

The key insight which allows the quantification of the sampling behavior is that the Jacobian matrix [Kap84] of \mathbf{M}_{Proj}^{-1} , denoted $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$, characterizes the sampling induced by the imaging transform.

$$\mathbf{J}(\mathbf{M}_{Proj}^{-1}) = \begin{bmatrix} \frac{\partial s}{\partial u} & \frac{\partial s}{\partial v} \\ \frac{\partial t}{\partial u} & \frac{\partial t}{\partial v} \end{bmatrix} \quad (4.9)$$

The values in the Jacobian matrix, as partial derivatives, describe the change in sample distances in the directions indicated. Thus, the elements of this matrix describe the change in sampling behavior induced by the transformation \mathbf{M}_{Proj}^{-1} . The Jacobian matrix gives the distance of a step on the surface, given a unit step

in the image. *This is exactly the sample distance in world units.* Note that because it is derived directly from \mathbf{M}_{Proj}^{-1} , $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$ accounts for all of the factors affecting the sampling rate – projective effects, camera pose and position, lens distortion, etc. Since the sample distance determines the Nyquist frequency, quantitative claims may be made regarding the sampling behavior under \mathbf{M}_{Proj}^{-1} based on the values in the matrix $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$.

If the entries in \mathbf{M}_{Proj}^{-1} are labeled:

$$\mathbf{M}_{Proj}^{-1} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (4.10)$$

Then $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$ is simply:

$$\mathbf{J}(\mathbf{M}_{Proj}^{-1}) = \begin{bmatrix} \frac{(ah-bg)v+(ai-cg)}{(gu+hv+i)^2} & \frac{(bg-ah)u+(bi-hc)}{(gu+hv+i)^2} \\ \frac{(dh-eg)v+(di-gf)}{(gu+hv+i)^2} & \frac{(eg-dh)u+(ei-hf)}{(gu+hv+i)^2} \end{bmatrix} \quad (4.11)$$

Although the formulation in Equation 4.11 is most useful for the current work, note that the Jacobian can be calculated for the functional form of the mapping as well. This is useful, for example, when dealing with higher order lens distortion models than may be modelled with a matrix.

As an example of the interpretation of the values in the Jacobian matrix, assume some \mathbf{M}_{Proj}^{-1} such that, at some point on the surface:

$$\mathbf{J}(\mathbf{M}_{Proj}^{-1}) = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad (4.12)$$

If world coordinates are given in units of centimeters, the Jacobian matrix in Equation 4.12 indicates that a sensel projected onto the object at this point would

cover a region 5 cm high by 5 cm wide. Recall that the Nyquist frequency is equal to $\frac{\text{sampling rate}}{2}$, or in this case $\frac{1}{5 \cdot 2} \frac{\text{cycles}}{\text{cm}} = \frac{1}{10} \frac{\text{cycles}}{\text{cm}}$, in both the horizontal and vertical directions. Thus the smallest feature³ that can be detected without aliasing is 10cm in height or width.

4.4.2 Sampling in texture units

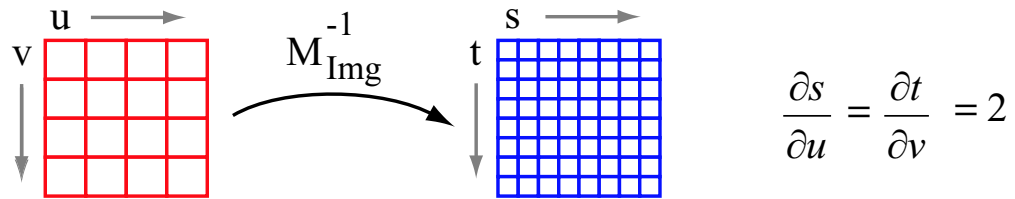
Although the transform \mathbf{M}_{Proj} describes the sampling behavior in the environment with respect to world units, it neglects \mathbf{M}_{Tex} , the mapping of the texture representation to the surface. Note that the surface parameterization is assumed to have a constant scaling factor relationship to the parameterization of the texture mapped to a surface. Since we are also interested in the quantitative properties of the textures in relation to the image data, we can extend the above analysis to use the entire imaging transform \mathbf{M}_{Img} , which maps a texel to a sensel. Because we used an (s, t) parameterization of the object surface above, the relationship between texture and image has the familiar form:

$$\beta = \mathbf{M}_{Img} \cdot \alpha \tag{4.13}$$

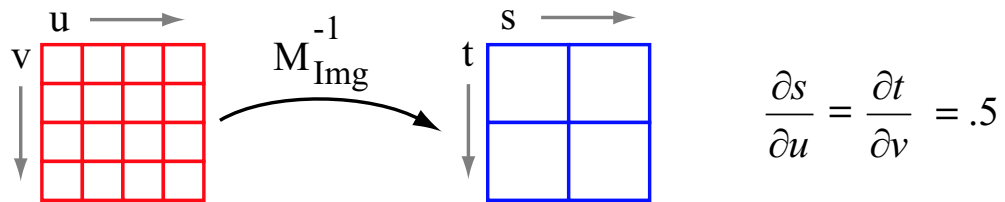
This relationship allows us to conveniently refer to the rate at which the sensor samples the scene in units of texels per sensel. Again, to stay with the conceptual model of projecting the image data into the environment, we deal primarily with the inverse of the imaging transform, \mathbf{M}_{Img}^{-1} . This transformation is again characterized by its Jacobian matrix, $\mathbf{J}(\mathbf{M}_{Img}^{-1})$.

We will now look at a slightly more involved example of interpreting the values

³Here, in keeping with the sampling nomenclature, a ‘feature’ is a complete cycle of an intensity profile sinusoid.



(a)



(b)

Image space

Texture space

Figure 4.6: A simple example illustrating the use of the elements of J to judge sample rate. In (a) the texture has more resolution than is necessary to capture the detail in the image. In (b) the texture has too little resolution and cannot capture all of the detail present in the image.

in the Jacobian matrix, than was performed in the last section. Evaluating the Jacobian matrix across the texture gives four measures per texel (the four partial derivatives). Values on the diagonal ($\frac{\partial s}{\partial u}$ and $\frac{\partial t}{\partial v}$) indicate the relative sample distances in the direction of the projected u and v axes. This is illustrated in Figure 4.6. The off-diagonal values ($\frac{\partial s}{\partial v}$ and $\frac{\partial t}{\partial u}$) can be thought of as indicating the rotation of the s and t axes relative to the projected u and v axes. In Figure 4.6 these values (not shown) would be zero. As Figure 4.6 (a) shows, values in the Jacobian matrix > 1 indicate that texture-space is discretized more densely than the projected image-space, while values < 1 indicate the opposite (as shown in (b)).

The values in the Jacobian matrix have a direct physical interpretation in terms of the frequencies from the image that the texture is capable of representing at a given resolution. In a sampling sense, values ≥ 1 indicate regions where the texture parameterization is sufficiently dense to represent all of the data captured by the image sensor (Figure 4.6 (a)). That is, projected sensels spread out to cover more than one texel. Values < 1 indicate that some data in the image cannot be represented with the given texture parameterization (Figure 4.6 (b)), or in other words, projected sensels cover less than one texel.

4.4.3 An example using the Jacobian metric

As an example of how the Jacobian can be used to characterize the sampling over some environment, consider the scene of brick cubes in Figure 4.7. The environment is shown in the image at the top. For visualization purposes, the single largest value of the Jacobian matrix evaluated at each pixel is shown in the next row. This value corresponds to the worst sampling rate in any direction for that

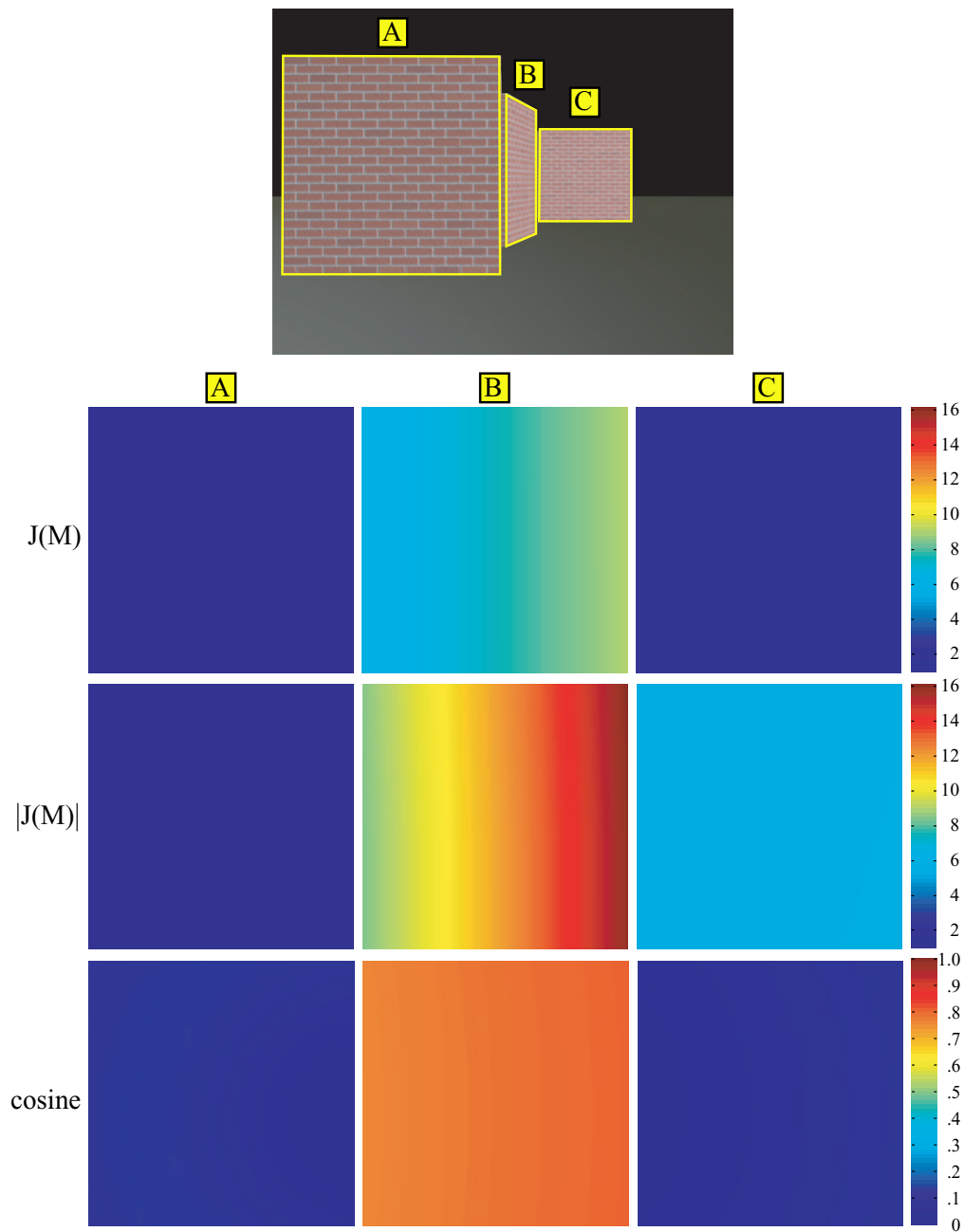


Figure 4.7: A simple example illustrating sampling metrics over an environment. The first row shows the Jacobian for each texel on the indicated faces. The bottom two rows show the determinant of the Jacobian matrix and the cosine of the angle between the view ray and the surface normal, respectively.

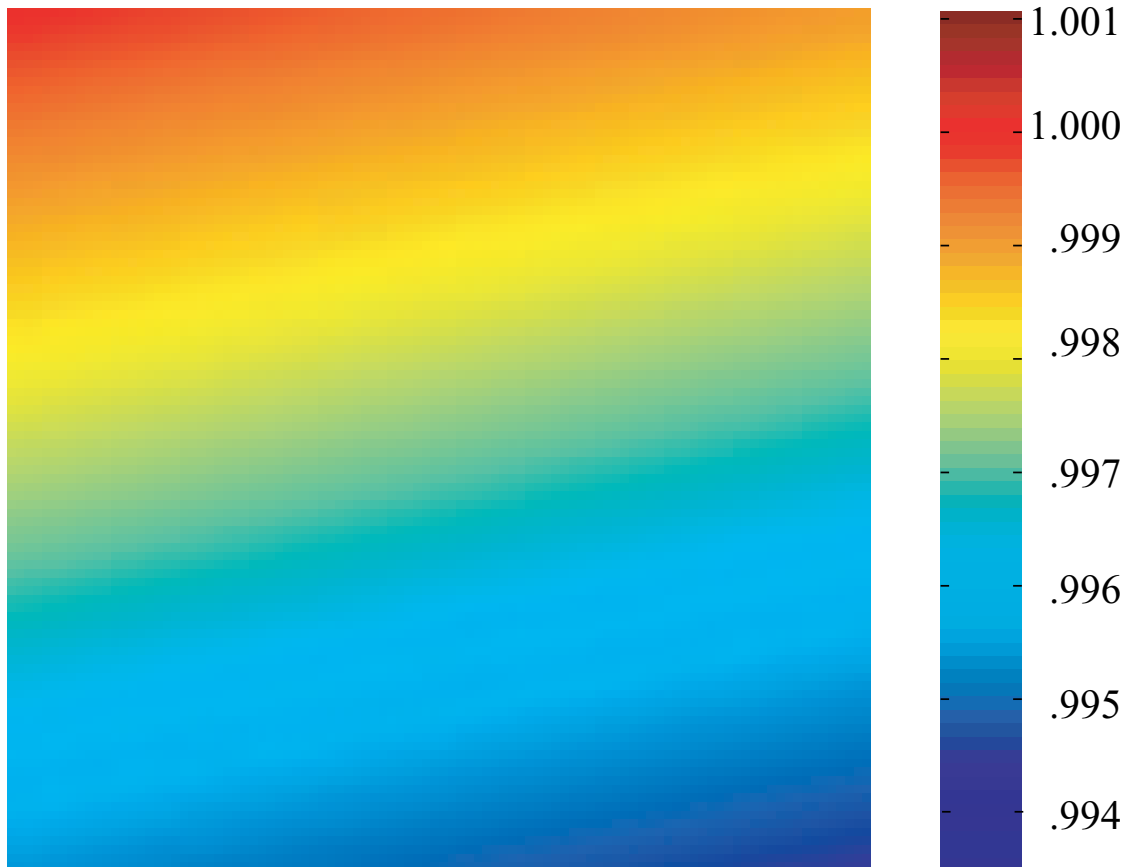


Figure 4.8: A rescaled visualization of the values of the Jacobian matrix for the front face shown in Figure 4.7. Although it is not immediately apparent in the original image, the front face is not parallel to the image plane – it is actually tilted slightly back and to the left. Notice the sensitivity of the metric to slight variations in the sampling rate induced by this very small rotation.

pixel, and represents an upper bound on the non-directional sampling information present. The textures shown across the row are color-mapped visualizations of these Jacobian values for each face of interest. Notice that the middle, oblique face has a much higher maximum Jacobian value than the back face. This is due to the large difference between the viewing angle and the surface normal of this face. In contrast, although the back face is more distant, its maximum Jacobian value is only approximately twice that of the front face, indicating that the sampling rate across the back face is approximately half that of the front face. Figure 4.8 shows a rescaled visualization of the Jacobian for the front face. This figure demonstrates the sensitivity of the metric to even slight variations in the sample rate caused by the pose of the face relative to the camera. As the figure shows, the front face is tilted slightly back and to the side relative to the image plane. Next we compare the Jacobian metric to two other metrics proposed in the literature and used in existing systems.

4.5 Comparison of metrics

This section compares the performance of the physically-based Jacobian metric to that of two other common texture quality metrics that attempt to approximate variations in sampling. We compare our metric to a viewing angle metric and a projected sensel area metric. To derive a single value for comparison purposes, we again use the largest value in the matrix per texel as a conservative bound for the best non-directional sampling rate at that texel.

Texturing algorithms utilizing the cosine metric were surveyed in Chapter 2. This metric calculates the dot product of the viewing angle and the surface normal at a point. For ease of visualization, we have used $\frac{1}{\cos(\theta)}$. This metric, although

quick to compute, has the major drawback that it disregards depth. In fact, as the bottom row of Figure 4.7 reveals, the cosine metric evaluates *slightly better* across the distant face than across the front face. This is exactly the opposite of the actual sampling behavior. In the front face, we also see some slight falloff as the distance from the center of projection increases. This effect is not actually present in the physical sampling and is an artifact of this metric. Last, note that over the middle, oblique face in the scene, the cosine metric shows far too small of a range in values compared to the correct, physically derived Jacobian.

The projected sensel area is a much better measure than the cosine, and is very similar to the Jacobian metric presented in this thesis. In fact, the projected sensel area is equivalent to the determinant of the Jacobian. However, this metric conceals the directional information present in the full matrix form of the Jacobian. Because of this, there is no way to extract sample distances or orientation information from this metric. Figure 4.9 shows this problem graphically. Although the horizontal sample distance varies widely across the top row of the illustrated sampling profiles, the area of each projected sample is constant. In the second row, the angle of the sampling varies up to 90 degrees. Not only does the determinant assign the same value to each angle, it assigns the same value calculated for the sample patterns in the top row! In contrast, the full Jacobian metric describes each of these sampling patterns uniquely. Thus, although the determinant metric is similar to the Jacobian metric, is cannot be used to understand the directional frequency response of the imaging system for the environment. As shown in the middle row of Figure 4.7, the determinant metric also generates values that are roughly the square of the Jacobian values, due to its measure of area instead of distances. This has the unfortunate result that the values derived from the metric no longer have a

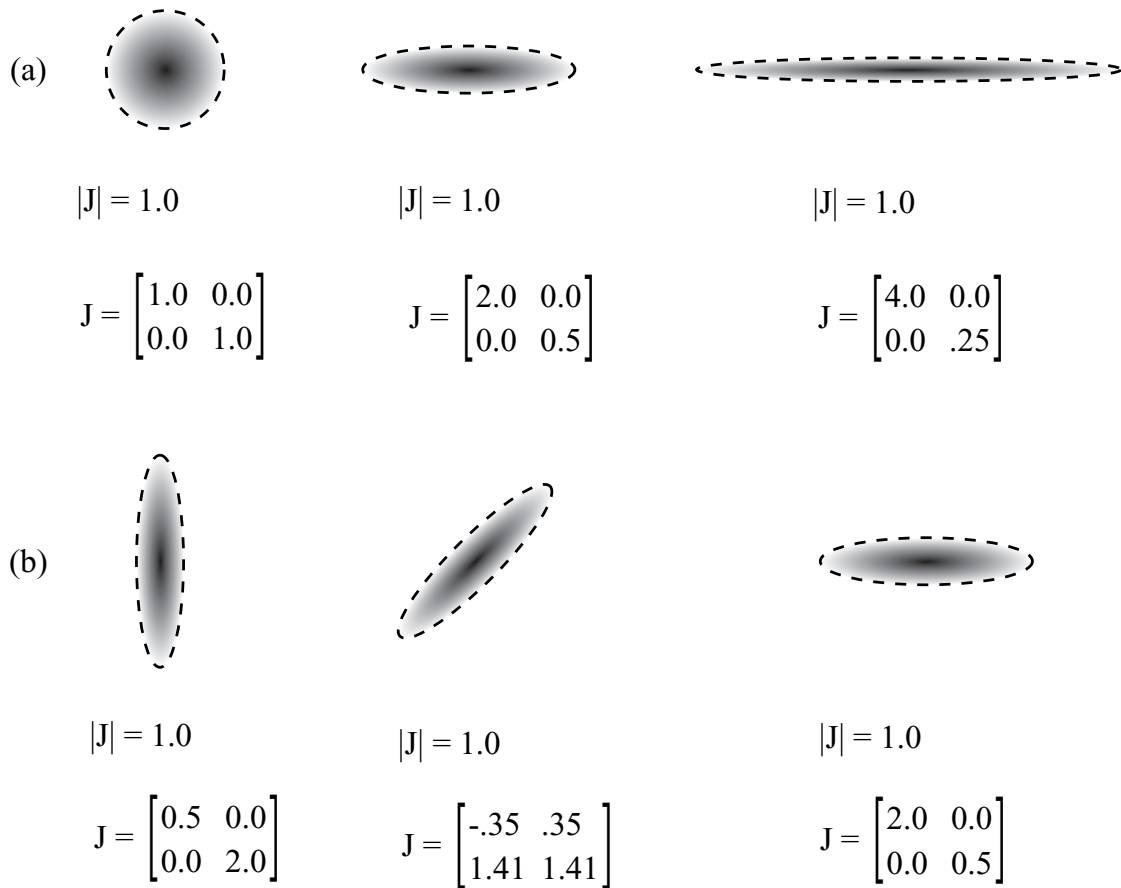


Figure 4.9: A comparison of the Jacobian metric to the determinant of the Jacobian for several sample projections. All of these patterns evaluate to the same determinant value, while the full Jacobian matrix preserves the actual sampling behavior.

direct relationship to the sample distance. Additionally, the directional component information necessary to get back to that information has been discarded.

The new Jacobian-based sampling quality metric described in this chapter is clearly superior to these other metrics. It is physically-based and measures actual sample distances, either in world units or in texels. It preserves directional information, and additionally, it is easy to compute, given a matrix for the imaging transform. One potential application of this metric – detail synthesis – is demonstrated in the next chapter.

Chapter 5

Detail Synthesis

In the last chapter, we introduced a metric for assessing the sampling quality of image-based textures, based on the Jacobian of the imaging transform. Given such a metric, we now return to the problem of extracting uniformly high quality textures from few images. Recall that when textures are resampled from an image some surfaces are well sampled – yielding good quality textures – but some (or even most) surfaces are not. When a surface, or portion of a surface is poorly sampled the extracted textures are of poor quality, appearing blurry or stretched. This effect was demonstrated in Figure 4.1, which is repeated here as Figure 5.1 for convenience. The middle row of images illustrates the sampling pattern over the three faces of interest. The bottom row shows a portion of the textures resampled from each of these faces. The textures for faces B and C show obvious visual blurring and stretching defects.

Texture synthesis has recently been proposed [PKVG98, WL00, FJP02] as an attractive way to correct deficiencies in resampled textures. However, existing applications of the technique have focused on either traditional texturing (in texture-space only) or image-editing (in image-space only with no corresponding geometry) – no research to date has demonstrated texture synthesis techniques for image-based texturing, operating over the surfaces in an environment.

This chapter introduces a method for applying current texture synthesis techniques to surface textures resampled from images, using the Jacobian-based sampling metric introduced in Chapter 4. Our method has several inherent strengths which differentiate it from current approaches. First, although we demonstrate our

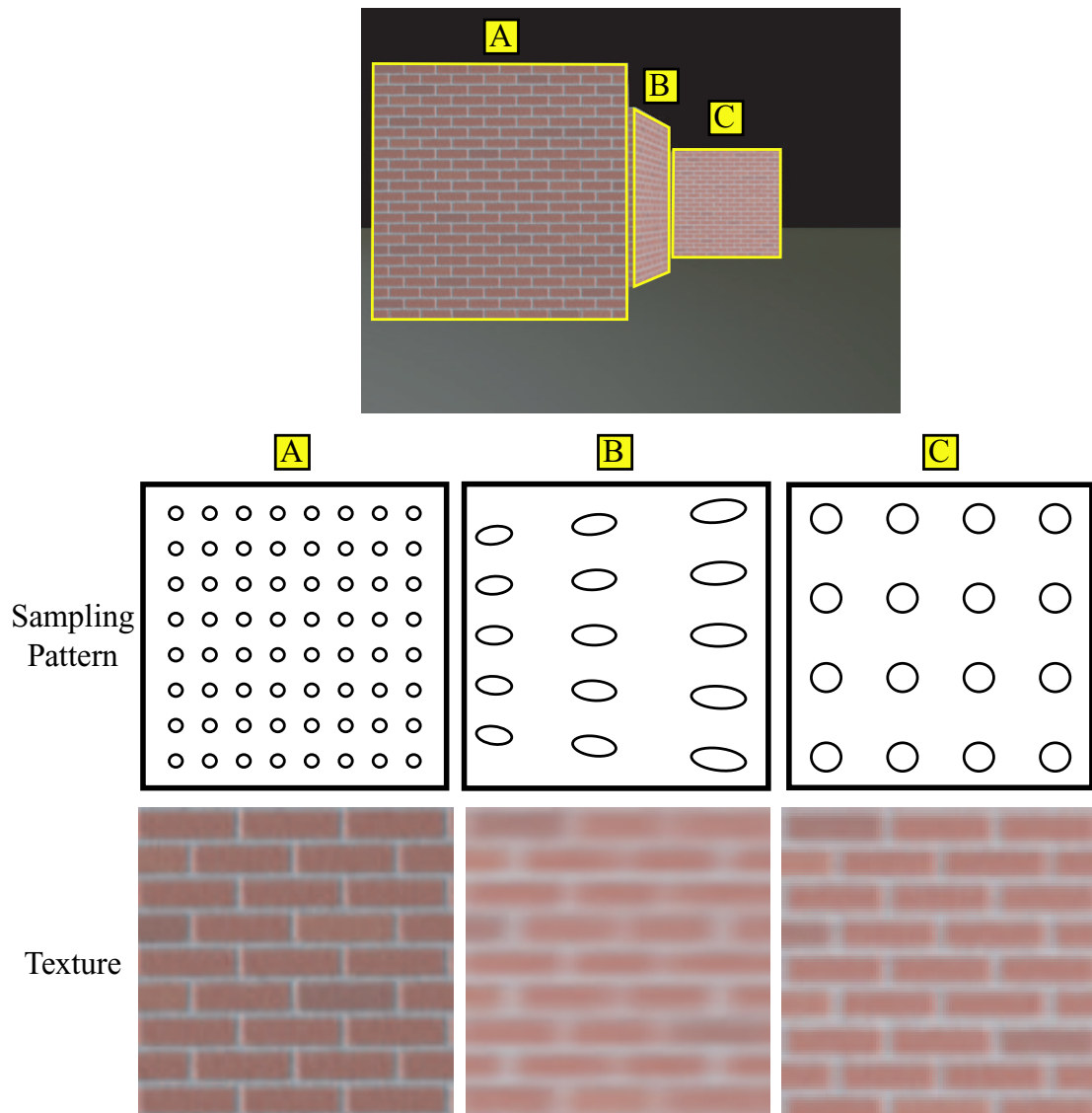


Figure 5.1: Sampling patterns in an environment. The top image shows the scene. The middle row illustrates the sampling patterns over each surface of interest. The bottom row shows the effects of the sampling pattern on the resampled textures.

approach using an extension of the method presented by Wei and Levoy[WL00], in theory any texture synthesis method may be used. Second, neither the technique, nor the Jacobian metric are restricted to planar surfaces. Last, and most importantly, our approach preserves the lower frequencies present in the resampled texture, adding only the correct high frequency detail. In this way, no good data from the image is ever discarded. For this reason we call our approach *detail synthesis*.

Because our approach relies on existing texture synthesis methods, we briefly summarize texture synthesis algorithms before describing our detail synthesis algorithm.

5.1 Texture Synthesis

Background work in texture synthesis was surveyed in Chapter 2. Recall that, in general, texture synthesis is the problem of generating an arbitrarily sized texture patch ‘similar to’ a given example patch, without tiling. We refer to the sample patch as the *source*, and the generated patch as the *target*. First, a pyramid representation is created for both the existing source texture and the potential target texture. Current synthesis approaches have used Gaussian, Laplacian, or steerable pyramids [KS96]. The source pyramid has data at every level and the target pyramid is seeded with noise. The noise is generated so that its histogram matches that of the corresponding level in the source pyramid. The pyramid for the source texture functions as a Markov Random Field (MRF) modeling the process which created the texture. This model is sampled using a simple best-match operator to insert data into the target pyramid. A sample in the MRF usually consists of some neighborhood at the level being filled, as well as some

number of corresponding neighborhoods at higher levels (*parent neighborhoods*). The target pyramid is filled from the top down, in scanline order, in order to take advantage of multiresolution information and local spatial coherence.

The synthesis process is illustrated in Figure 5.2. Parent levels are hidden to improve clarity. Figure 5.2 (a) shows a level of the source pyramid. Several samples of the MRF are indicated as red neighborhoods. Although only five neighborhoods are shown, an MRF sample (a neighborhood) exists for every pixel at this level. The images in row (b) show the progress of the synthesis algorithm in the corresponding level of target pyramid. For each texel in the target, the algorithm finds a texel in the source with the closest matching neighborhood, then copies that texel into the target. Note that since the algorithm proceeds in scanline order and uses causal neighborhoods, noise needs only to be injected into the target on the left and bottom borders. The algorithm’s result for this level is shown in (c). The resulting target level is significantly similar to the source level, but is in fact unique. For additional details we refer the reader to the original texture synthesis research referenced in Chapter 2.

Algorithms similar to the one described above work well for a variety of texture types, but break down for others. In addition, the literature provides little guidance regarding parameter tuning (e.g. neighborhood size and number of parent levels) to improve the results. Because we use standard texture synthesis techniques to obtain our results, our technique is limited by the same factors. In general, detail synthesis performs well where existing texture synthesis techniques perform well; it fails where they fail. However, our approach does not rely on any specific synthesis algorithm, thus we can take advantage of any future improvements in texture synthesis research to further improve the quality of our results. We now

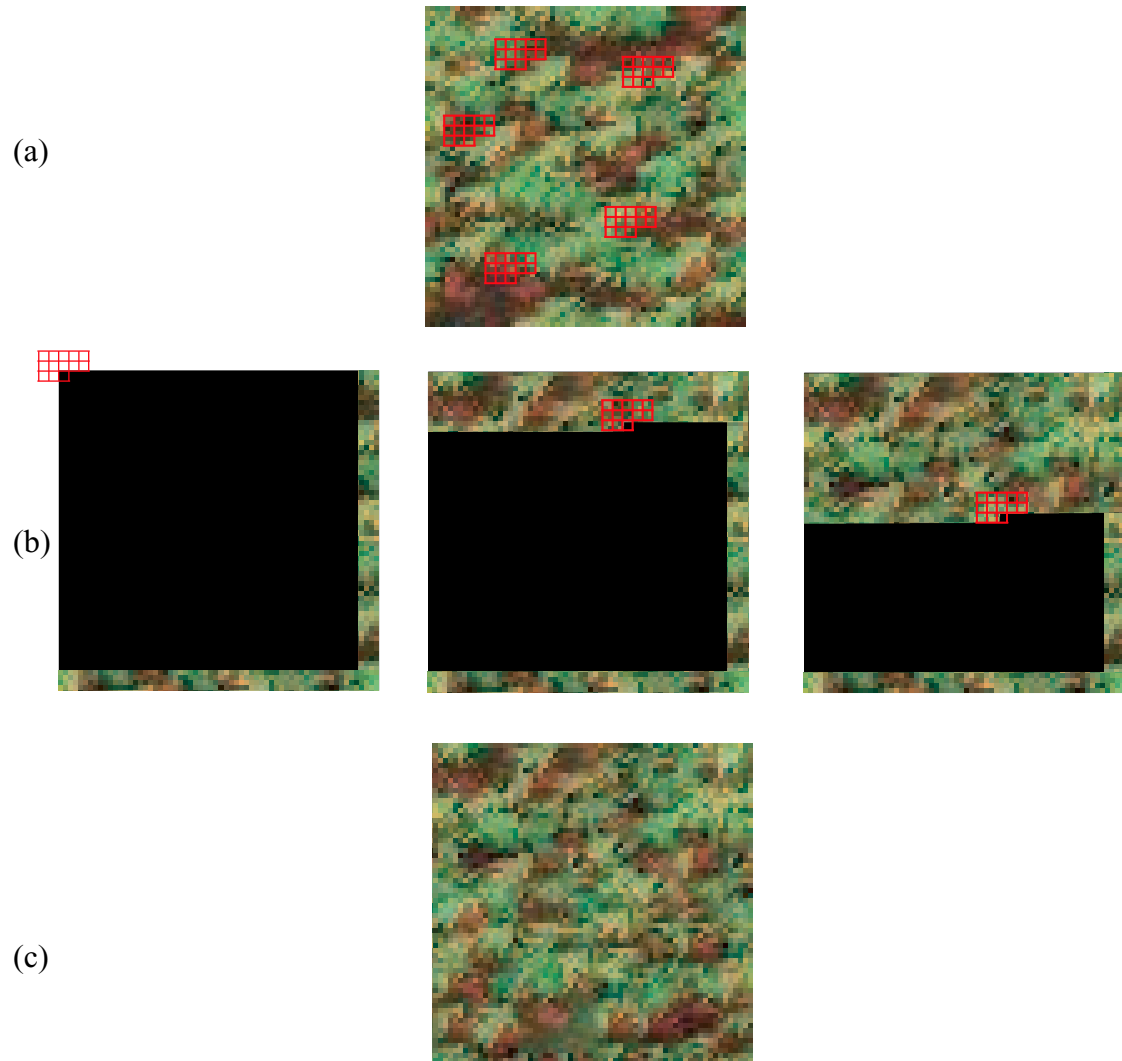


Figure 5.2: Traditional texture synthesis at work. (a) shows the source texture with several neighborhoods indicated. Row (b) illustrates the progress of the synthesis process. (c) shows the result of the synthesis.

explain the proposed detail synthesis algorithm.

5.2 Detail Synthesis

5.2.1 Overview

The goal of the proposed detail synthesis approach is to create image-based textures that have uniform frequency content for surfaces which are non-uniformly sampled. Additionally, the process should not modify or delete any data in frequency bands which are properly resampled by the texture reconstruction process. That is, data should be added where it is missing, while preserving all of the texture data that is present. This section gives an overview of our detail synthesis algorithm.

We begin with a texture of some surface with a high constant sampling rate, which we designate as the source, and with a texture of some similar surface with a lower, possibly varying, sampling rate, which we designate as the target. Observe that the Jacobian-based metric proposed in Chapter 4 can be used to determine which texture (among all similar textures in the scene) has the highest possible uniform sampling rate.

In extending texture synthesis algorithms to image-based texturing we perform the following steps:

1. **Determine the best scaling of the source texture.** This process uses the Jacobian values to determine the scaling which produces the optimal source texture resolution. A source texture of the indicated resolution is extracted from the image.
2. **Normalize the target texture with respect to the source texture scaling.** This results in a target texture with the same number of texels

per world unit as the source texture. A target texture of the indicated resolution is extracted from the image. The Jacobian metric for the target is also recalculated, using the rescaled imaging transform. This yields a set of values describing the sampling rate of the target relative to that of the source.

3. Construct Laplacian pyramids for the source and target textures.

The height of the pyramids are determined by the lowest entirely valid level of the target pyramid, as well as the number of parent levels (if any) used by the synthesis algorithm.

4. Construct a validity pyramid for the target texture.

The new pyramid structure will store values indicating the validity of the corresponding texels in the target Laplacian pyramid. The validity pyramid is initialized using the Jacobian values of the normalized target texture.

5. Perform texture synthesis.

Using the complete source pyramid and the incomplete target pyramid (together with the validity pyramid), synthesize the data still absent into the target pyramid, then reconstruct the texture.

Note the importance of the Jacobian metric both in selecting the optimal texture resolution, as well as in indicating degraded texture areas which require detail synthesis. Specifically, the correlation between the maximum value of the Jacobian metric and levels of a bandpass image pyramid is the key insight which allows the extension of texture synthesis techniques to image-based texturing. The following sections describe the entire process in greater depth.

5.2.2 Determining the source texture scaling

The purpose of this step is to refine the mapping \mathbf{M}_{Tex} (see Section 4.3), so that the source texture is of such a resolution that it captures exactly the frequencies present in the image data. Note that this is in general impossible, as the sampling rate will generally not be constant across the entire surface, thus we use the conservative resolution determination described below.

We begin by calculating the Jacobian matrix across the texture assuming that \mathbf{M}_{Tex} is the identity matrix (effectively $\mathbf{J}(M_{Proj}^{-1})$), using the world space length of the surface’s s and t directions as the initial texture resolution. Because the values in the Jacobian matrix can be interpreted directly as sample distances, we can take the largest of any of the four values across the surface as the largest distance between any two samples from the image, in either the s or t directions. This distance, k , is used as the conservative resolution scaling factor. We apply $\frac{1}{k}$ to the texture size values to get the new texture resolution. To adjust the matrix scaling we use:

$$\mathbf{M}_{Tex} = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Once k is determined, a source texture is extracted from the image. Intuitively, we have normalized the source texture such that all of the elements in the texture have a Jacobian of ≤ 1 . Therefore, the lowest level of the pyramid is fully populated.

There is a side effect of using this conservative value, k , to bound the Jacobian values over the surface. The resulting texture has only enough resolution to capture

the data from the image at the point where the surface is most poorly sampled, but not where it is most densely sampled, which implies that data is lost when extracting the texture data from the image. This behavior is, however, desirable for texture synthesis. If we were instead to select the minimum Jacobian value, and thus had a texture resolution high enough to capture the most densely sampled regions of the object surface, then there could be portions of (at least) the lowest level of the Laplacian pyramid that would not contain any data. This would cause portions of the MRF for the source texture to contain incomplete data and would cause the synthesis algorithm to perform poorly.

5.2.3 Normalizing the target texture

We now wish to determine how the target texture sampling rate relates to that of the source texture. We do this for two related purposes - first to extract a target texture of an appropriate resolution, and second, to determine where texture data in the target pyramid is valid.

For the target texture we use the same source scaling factor, k , and the source \mathbf{M}_{Tex} . After calculating the new target resolution using the scaling factor k , a target texture is extracted from the image. Next the Jacobian metric is calculated over the surface using the scaled imaging transform. Using the maximum of the calculated Jacobian values over the target surface, the lowest completely valid level of the target pyramid is calculated. The values of the Jacobian matrix across the target texture now yield a quantitative comparison of the sampling between the source and target surfaces. Note that the target texture is defined as the more poorly sampled texture, so the values of the target Jacobian will generally be larger than those of the source (i.e., the samples are more spread out).

For example, if the Jacobian matrix for the target texture is constant across the texture and is:

$$J(M_{Img}^{-1}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (5.2)$$

then the sample distance in the target texture's s and t directions is twice that of the source texture. Thus the sample rate for the target is half that of the source, as is the Nyquist frequency. In other words, the target is missing the highest octave of frequency data present in the source.

5.2.4 Creating the Laplacian pyramids

The technique presented in this paper uses Laplacian pyramids [BA83] as the multi-resolution image pyramid. Laplacian pyramids are formed using a ‘difference of Gaussians’ operator, and are thus (approximately) bandpass at each level. The frequency composition of an example pyramid is illustrated in Figure 5.3. The bottom level (level 0) contains data from the spatial frequency band from the Nyquist frequency, f_N , to $\frac{f_N}{2}$. The next level up (level 1) contains data from the spatial frequency band from $\frac{f_N}{2}$ to $\frac{f_N}{4}$. In a pyramid with levels 0 to n (bottom to top), level $m : 0 \leq m < n$ contains data in the spatial frequency band from $\frac{f_N}{2^m}$ to $\frac{f_N}{2^{m+1}}$. Level n contains data in the spatial frequency band from $\frac{f_N}{2^n}$ to 0.

Because of the bandpass nature this data structure, another major benefit of using Laplacian pyramids for detail synthesis lies in the method of reconstructing a final texture from a pyramid. For a Laplacian pyramid, the texture is recovered by repeatedly upsampling and merging levels, from the top down. Because the levels are bandpass, this means that the data from higher levels ‘show through’ the lower

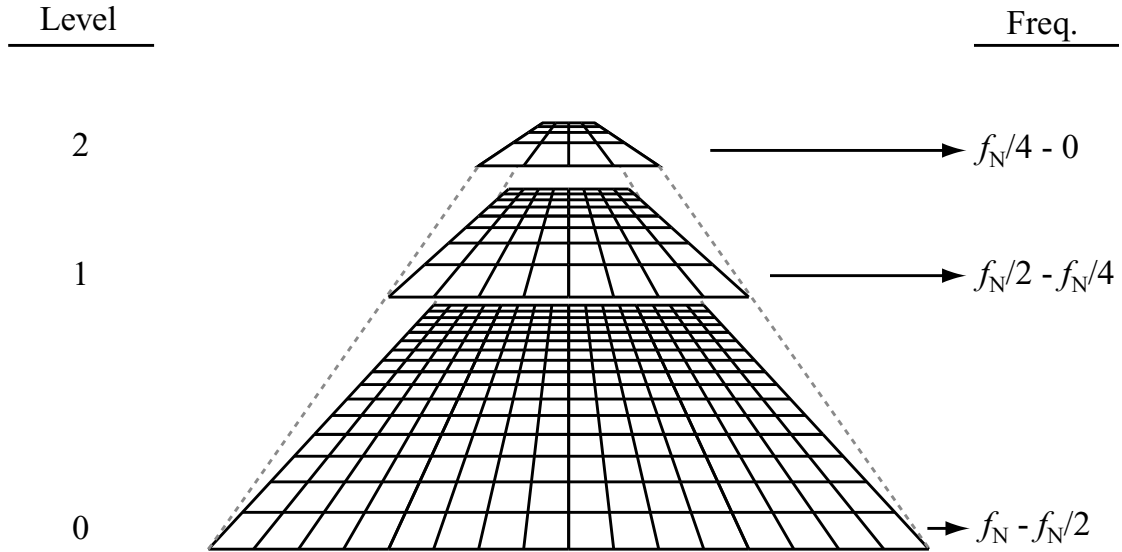


Figure 5.3: An illustration of a Laplacian pyramid with three levels. Level 0 contains the frequencies from f_N , to $\frac{f_N}{2}$. Level 1 contains the frequencies from $\frac{f_N}{2}$ to $\frac{f_N}{4}$. Level 2 contains the frequencies from $\frac{f_N}{4}$ to 0. Note that illustration does not show pyramid data in order to preserve clarity.

levels. Thus, if there is valid pre-existing data in, for example, the top 3 levels of a 5 level pyramid, and synthesized data in the bottom 2 levels, then the pre-existing low resolution data is preserved in the final texture, no matter what the results of the synthesis for the high frequency data in levels 0 and 1. This characteristic of the Laplacian pyramid is important in order for image-based textures to preserve all of the information present in the images.

For the synthesis algorithm to compare similar frequency bands, the height of the source and target pyramids must be equal. The pyramid height is calculated by taking the lowest completely valid level of the target pyramid with data, as determined above, and adding the number of parent levels that the synthesis al-

gorithm will consider. Creation of the source and target Laplacian pyramids then follows the standard algorithm. Figure 5.4 shows a Laplacian pyramid (a) for a well sampled texture (c), as well as a pyramid (b) for a poorly sampled texture (d). Note that there is significantly less information in the lower levels of (b) than there is in the lower levels of (a). This is due to the lack of data in the higher frequency bands.

5.2.5 Creating the validity pyramid

The knowledge that Laplacian pyramids are (approximately) bandpass, and that the Jacobian gives a measure of sampling rate, suggests that there is a direct and meaningful physical interpretation of their relationship. Simply, since each level of the pyramid contains an octave of data, a texel with the largest entry in its Jacobian matrix equal to r can only possibly contribute to data present at or above the $\log_2(r)$ level of the pyramid. This straightforward relationship gives us a simple decision rule for deciding if a texel at a certain level of the pyramid contains valid data by inspecting the corresponding value of the Jacobian metric for the target texture.

Observe that one major benefit of using Laplacian pyramids is the direct relationship between the Jacobian values for a texture and the levels of the Laplacian pyramid. As stated previously, the Nyquist frequency, f_N is $\frac{\text{sampling rate}}{2}$, and the Jacobian texture quality metric yields values equal to the sample distance. So, if η is the largest value of the Jacobian metric across the entire texture, then $f_N = \frac{1}{2*\eta}$.

If we assume that in some portion of the texture, the Jacobian metric is equal to 1 (i.e., the sampling is perfect, and $f_N = .5$), then the data in the remainder of the texture can easily be assigned to the appropriate pyramid level. Regions with

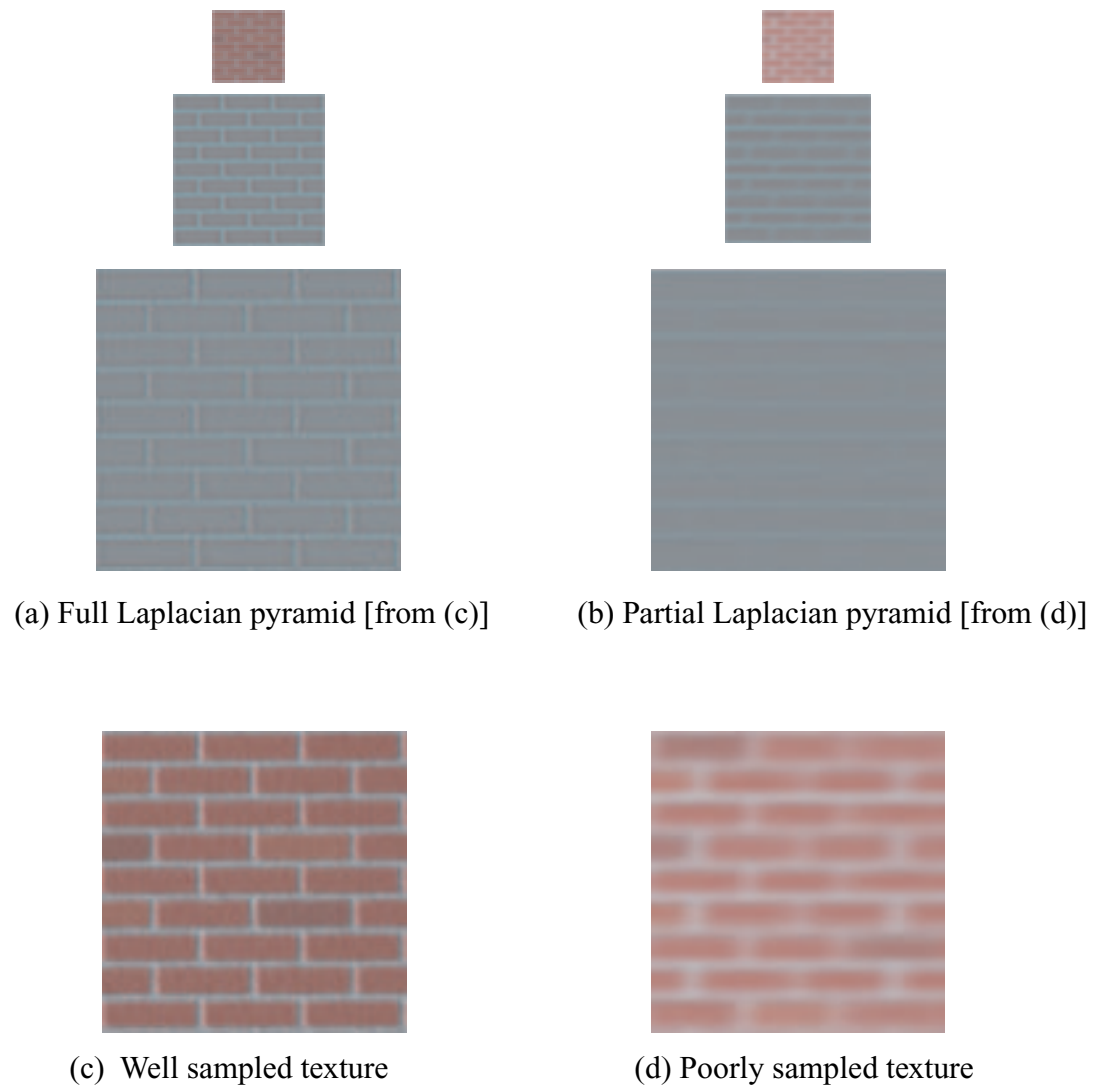


Figure 5.4: Laplacian pyramids for a good source texture and a degraded target texture.

sampling rates between 1.0 and .5 cycles per pixel (capturing frequencies between .5 and .25 cycles per pixel) would have data in and above level 0. Regions with sampling rates between .5 and .25 cycles per pixel (capturing frequencies between .25 and .125 cycles per pixel) would have data in and above level 1. Regions with sampling rates between .25 and .125 cycles per pixel (capturing frequencies between .125 and .0625 cycles per pixel) would have data in and above level 2. For example, a region of an image with a Jacobian value of 3.2 has a sampling rate of $\frac{1}{2*3.2} = .15625$ would have no data in the Laplacian pyramid lower than level 1.

With this information, creating a pyramid data structure containing validity information for the target texture is relatively straightforward. An image pyramid equal in size to the target Laplacian pyramid is created. The validity pyramid is marked with a 1 if the frequency band corresponding to that level of the pyramid could be sampled by the sensor, given the value of the Jacobian metric for the corresponding location on the target texture. Otherwise it is 0.

The validity pyramid for a degraded texture is shown in Figure 5.5. The target Laplacian pyramid is shown in (a) for the texture in (c). The validity pyramid (b) was calculated from the Jacobian values for the surface displayed in (d).

5.2.6 Synthesis implementation

The actual synthesis is based on the algorithm by Wei and Levoy[WL00]. We utilize a multi-pass sliding neighborhood approach that minimizes the sum of the squared Euclidian distance between the pixel values in the target neighborhood and pixel values in the neighborhoods of the search space. RGB values for all pixels in the neighborhood are packed sequentially into vectors of pixel values. The actual search uses a k-d tree to accelerate matching of the pixel vectors.

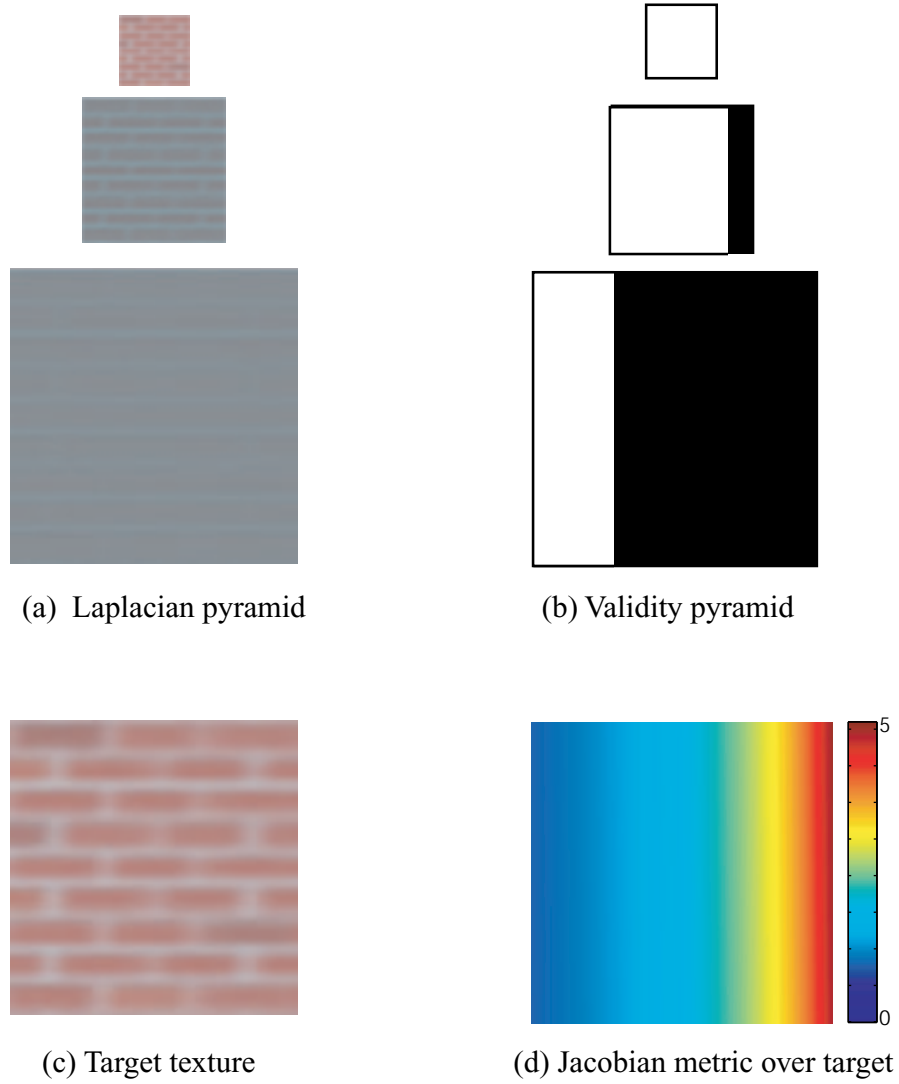


Figure 5.5: The target Laplacian pyramid is shown in (a) for the texture in (c). The validity pyramid (b) was calculated from the Jacobian values for the surface displayed in (d). The Jacobian values range from approximately 1.0 to 5.2, as shown by the color bar to the right of (d).

We believe that this application of detail synthesis to image-based model textures is unique. Wei and Levoy[WL00] and Freeman, et al.[FJP02] have demonstrated the use of synthesis techniques as image editing tools, where the synthesis was performed in image space. The quality metric and pyramid creation technique presented in this thesis are significant in allowing the application of these methods to textures on the surfaces of imaged objects.

5.3 Results

5.3.1 Image acquisition and modeling

All of the real-world examples were photographed with a Canon EOS D30 digital camera. Images were acquired in RAW mode (12 bits per pixel) at 2160x1440 resolution, and were demosaiced and converted to gamma-corrected 24-bit as well as linear 48-bit TIFs using the Canon-provided software with no contrast or saturation adjustment. The camera intrinsic parameters and distortion term were calibrated with the Intel Camera Calibration Toolbox for Matlab[Cal], and the optoelectronic conversion function was confirmed to be linear using HDRShop[HDR] and ISO14524[ISO]. Although the general camera model can account for sensor elements that have non-orthogonal and differently sized x and y dimensions, our particular image sensor was verified to have square sampling aperture and orthogonal sensel rows and columns. Geometrically, the camera is modelled as a standard projective pinhole camera, and includes radial and tangential distortion parameters. As shown above, the Jacobian measure can account for image distortion; however, to accommodate our image-based modeling software we undistorted the images prior to modeling.

The real-world scenes were modelled with commercially available image-based modeling software (RealVIZTMImageModelerTM), and exported to our system using VRML. The camera extrinsic parameters were automatically calculated by ImageModelerTM during the modeling process and exported with the VRML file.

The linear, 48-bit images were used for the actual detail synthesis. These images were found to give superior quality when compared to the gamma-corrected, 24-bit images.

5.3.2 Rendered bricks

The brick cubes shown in Figure 5.6 were rendered at 800-by-600 using Discreet 3DSMaxTM, then modelled again using ImageModelerTM. The front face of the front cube (A) was used as the source texture; the side of the middle cube (B) and the front of the back cube (C) were the target textures. After re-scaling \mathbf{M}_{Tex} the maximum Jacobian value for the side face was approximately 4.89, with a large range due to the angle of the surface. The maximum Jacobian value for the back face was approximately 2.35, and was fairly constant across the surface.

Synthesis results are shown in the bottom row of Figure 5.6, for textures B and C. Texture A was used as the source, and is thus unchanged. The synthesis algorithm used pyramids with 5 and 4 levels, for the middle and back faces, respectively. The algorithm used 2 passes with 7x7 neighborhoods to generate the results shown. Our technique was able to inject enough high frequency data in a correct manner to noticeably sharpen the output textures. Notice also that the algorithm preserves the low frequency information (present in the color of the bricks) instead of writing over this with data from the source texture.

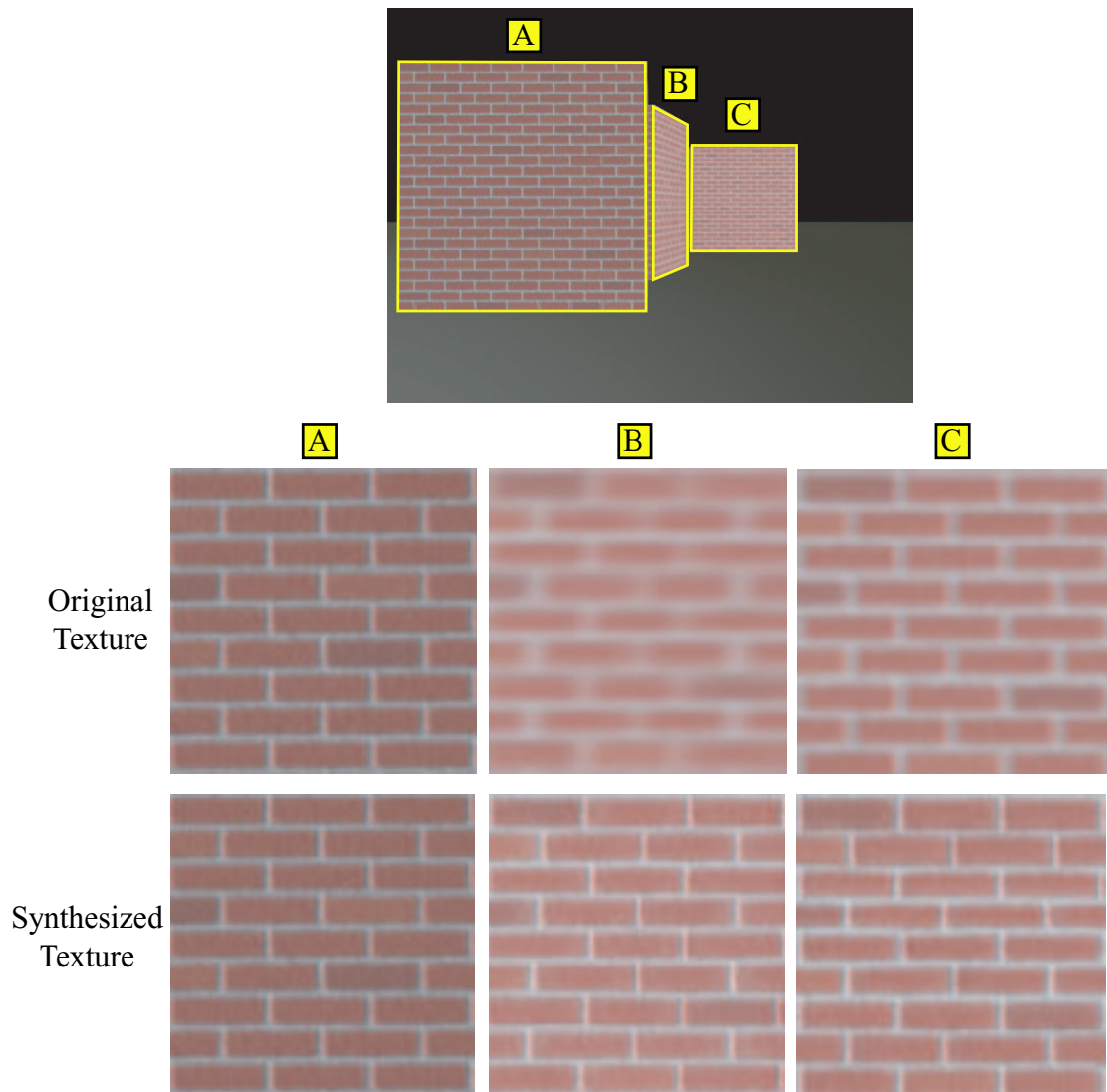


Figure 5.6: Detail synthesis results for the synthetic brick scene. The middle row shows the original textures resampled from the image at top. The bottom row shows the results after detail synthesis. The texture for A was used as the source texture, and thus was not changed. The textures for B and C had significant degradation which was corrected by our technique.

5.3.3 Building columns

The results shown in Figure 5.7 are from Olin Library on the Cornell University campus. The image was taken with the Canon D30 camera described above. The front-facing side of the nearest column (A) was used as the source. The corresponding faces of the third (B) and fourth (C) columns were used as the targets. After re-scaling \mathbf{M}_{Tex} the maximum Jacobian values for the target textures were approximately 2.4 and 3.1, respectively, with only a small range across each surface. The synthesis algorithm used pyramids with 4 levels, and used 4 passes with 5x5 neighborhoods to generate the results shown.

As the results show, the synthesis correctly inserts higher frequency data into the target texture. The overall brightness of the target is preserved, even though the source texture is much brighter than the target. Although the mortar lines between the stones are not as clear as in the source, we believe that the result is promising, given the difficulty of this type of surface for current texture synthesis algorithms. The effects are subtle, but present. The strong differences in illumination and coloration between the source and the targets, as well as differences in structure, have all been overcome.

5.3.4 Pavement

The input images for these results were also imaged with the Canon D30. The image shown in Figure 5.8 was used to resample the target texture, which is shown outlined in red. A portion of the original resampled texture is shown in b). After re-scaling \mathbf{M}_{Tex} the maximum Jacobian value for the target texture was approximately 7.6, with a large range across the surface. The synthesis algorithm used a pyramid with 5 levels, and used 2 passes with 5x5 neighborhoods to generate the

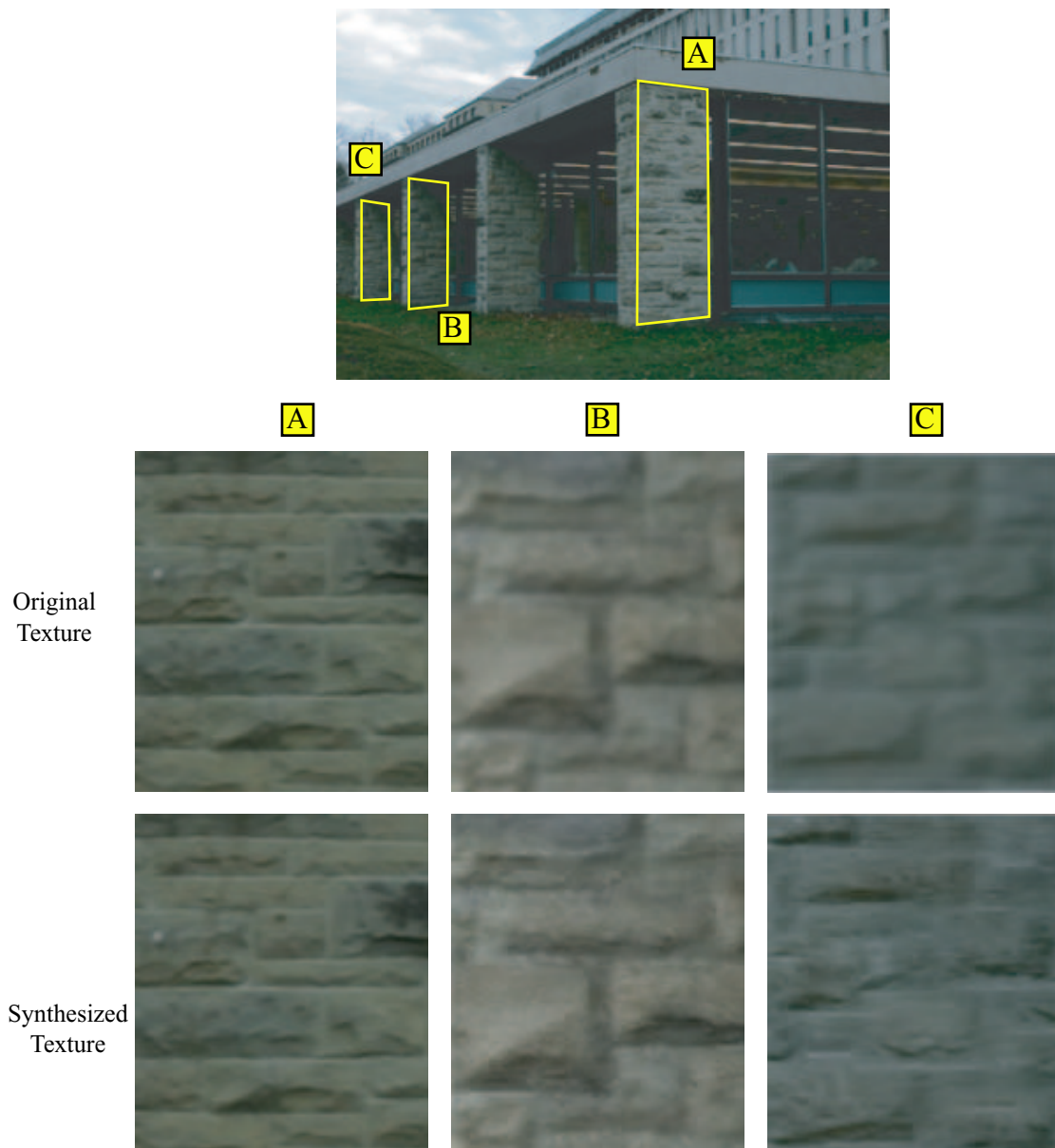


Figure 5.7: Results for columns along the face of Olin library. The front facing side of the nearest column (A) was the source. Both the third column (B) and the fourth column (C) have been synthesized. Although improvement is subtle, the result textures have a full extra octave of data.

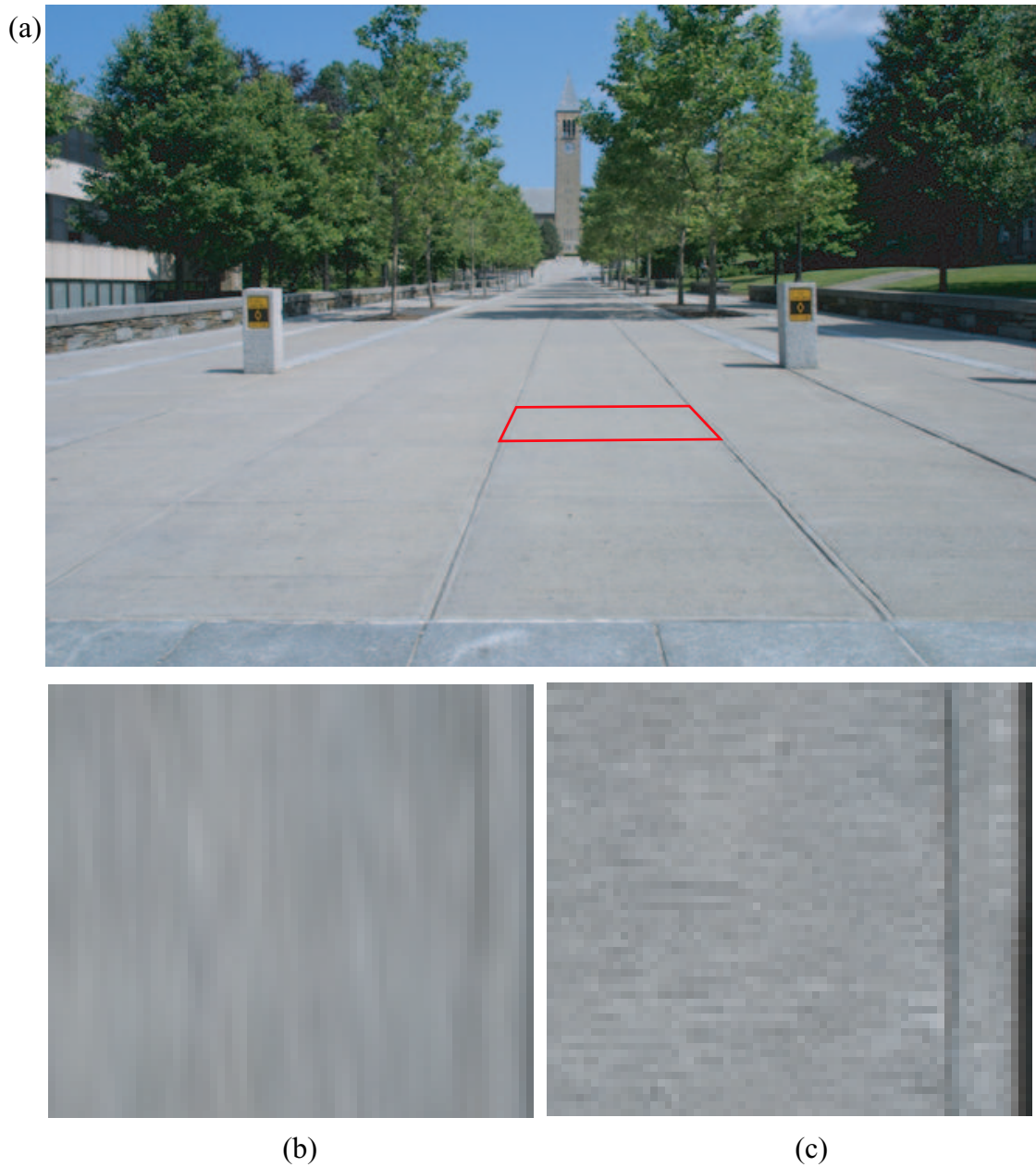


Figure 5.8: Results for a paved pedestrian walkway. The image in a) was used to resample the texture for the target pavement section (outlined in red). A portion of the original texture for this pavement section is shown in b). The results after synthesis are shown in c). The texture portion shown in d) shows the source texture (resampled from another image).

results shown.

Once again, the results show that the synthesis correctly inserts higher frequency data into the target texture. For example, the sharp border of the cement on the right is correctly synthesized by our algorithm. Also, unique low frequency characteristics of the target, such as the general lightness of the cement, are preserved. We would expect texture synthesis algorithms to perform well on this type of surface, which is supported by the quality of our results. However, without our technique significant user intervention would be required to accurately reproduce appearance characteristics unique to this imaged environment, for example, the high frequency data on the borders.

5.3.5 Moroccan Doors

The input images for these results were scanned using a Umax Powerlook 2100XL flatbed scanner. One image, shown in Figure 5.9 a) provides an overview of the doors, from which a region was cropped for use as the target texture. The detail photograph shown in Figure 5.9 b) was used as the source texture. The scale difference, and thus the maximum relative Jacobian value, between the door overview and the detail photo, was calculated to be approximately 4.0. The synthesis algorithm used a pyramid with 4 levels, and used 1 pass with 5x5 neighborhoods to generate the results shown.

A comparison of the resulting texture to the original texture is shown in c) and d). Figure 5.9 c) shows the target texture at the scanned resolution, with a region zoomed to illustrate the loss of detail under magnification. Figure 5.9 d) shows the result texture rescaled to be the same size as c). However, an examination of the same zoomed region shows that the detail synthesis algorithm has successfully

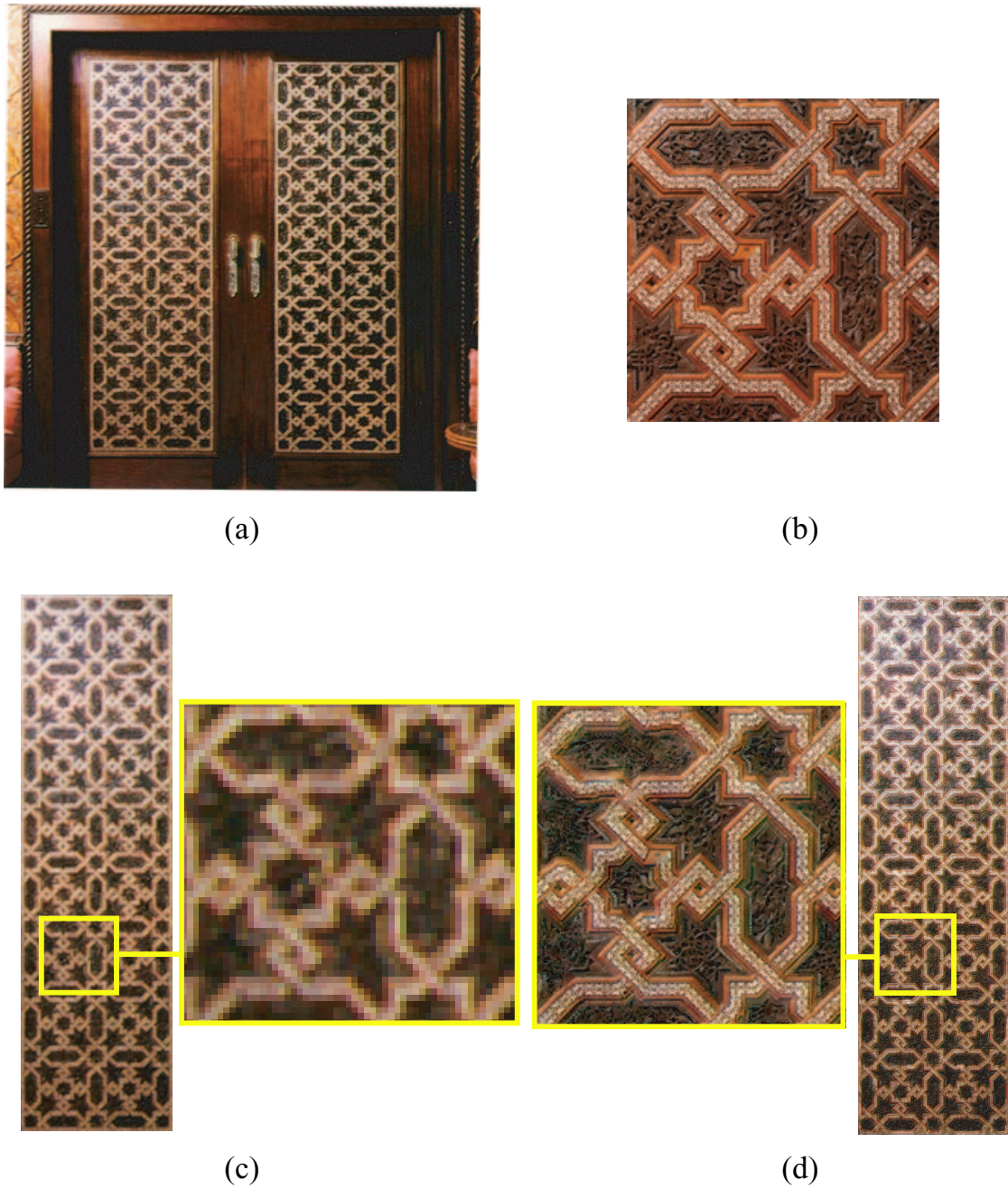


Figure 5.9: Results for a Moroccan door. An image of the set of doors is shown in a). A detail image is shown in b). A cropped image of the door panel is shown in c), with a blow up showing the lack of detail. The results of detail synthesis are shown in d). Here, the synthesized door panel reveals the intricacies of the inlaid wood pattern.

added a great deal of the detail of the tiny, intricate, inlaid wood pattern.

This set of results not only provides positive validation of our detail synthesis technique, but also illustrates how the Jacobian metric can be used to extend the similar image sharpening approach introduced by Freeman, et al [FJP02].

In summary, this chapter has demonstrated the first application of texture synthesis to image-based texturing that accounts for and preserves all information in the original image. This is made possible only because the Jacobian metric presented in this thesis has a direct physical interpretation in terms of sample distances in the scene, and therefore also has a direct relationship to levels of a bandpass image pyramid.

Chapter 6

Future Work and Conclusions

The results shown in this thesis have both theoretical and practical implications. As a theoretical formulation, the Jacobian-based sampling metric has a powerful descriptive value. Although this thesis has presented only the metric only in a form suitable for planar surfaces, the analytical forms for higher order surfaces should be straightforward to derive. Furthermore, the Jacobian metric contains far more information than we have made use of with our ‘worst-case’ single value measure. A close analysis of the directional components of the Jacobian will be of great use for any application that requires access to data on the orientation and eccentricity of the sampling. For example, some applications may find value in using the SVD of the Jacobian, which can be used to yield three matrices: two rotation matrices (one relative to each basis) and a scaling matrix. Other applications may wish to use data structures or techniques (such as wavelets) which might preserve orientation information across multiple scales.

The practical results of our detail synthesis technique show that the method has great promise, despite suffering from the shortcomings inherent in all texture Markov Random Field-based texture synthesis methods. The results show obvious, realistic sharpening of visually degraded textures, even for scenes with repeating structures and large intensity differences – generally pathological situations for current approaches. It achieves this performance by utilizing the correctly sampled low frequency portions of the texture to drive the synthesis of the high-frequency bands. We expect the results to improve even further with future advances in texture synthesis. Specifically, research providing metrics for neighborhood size

and the number of parent levels required by the algorithms would remove the main source of errors in the texture synthesis process.

In summation, this thesis has introduced two contributions to the field of image-based computer graphics. First, we have shown a physically-based metric of sampling quality, based on the Jacobian matrix of the imaging transform, which captures the interaction of the imaging system with the imaged environment. The use of the metric has been demonstrated for extending current texture synthesis methods to image-based texturing processes. Second, the use of the sampling metric enables a process we refer to as detail synthesis, which creates of high spatial frequency detail into a poorly sampled texture, while preserving existing low frequency texture data. This technique allows the creation of uniform, high-resolution textures. As we have noted previously, our synthesis approach relieves the user of the burden of collecting large numbers of images and increases the quality of user-driven image-based modeling systems. As a result of our research, it is now possible to achieve better, more accurate renderings of surfaces for use in texture-mapping routines commonly used in image-based computer graphics and image synthesis.

BIBLIOGRAPHY

- [BA83] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983.
- [BMR01] Fausto Bernardini, Ioana M. Martin, and Holly Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, October - November 2001. ISSN 1077-2626.
- [Bro66] D.C. Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.
- [BW93] M. Born and E. Wolf. *Principles of Optics*. Pergamon Press, Oxford, 1993.
- [Cal] Camera calibration toolkit for matlab. www.vision.caltech.edu/bouguetj/calib_doc/.
- [CRZ99] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece,*, pages 434–442, Sep, 1999.
- [Deb96] Paul Debevec. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley, 1996.
- [DeB97] Jeremy S. DeBonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pages 361–368, Los Angeles, California, August 1997. ACM SIGGRAPH / Addison Wesley. ISBN 0-89791-896-7.
- [DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pages 11–20, New Orleans, Louisiana, August 1996. ACM SIGGRAPH / Addison Wesley. ISBN 0-201-94800-1.
- [DYB98] Paul E. Debevec, Yizhou Yu, and George D. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Rendering Workshop 1998*, pages 105–116, Vienna, Austria, June 1998. Springer Wein / Eurographics. ISBN 3-211-83213-0.
- [EL99] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.

- [FJP02] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics & Applications*, 22(2):56–65, March–April 2002. ISSN 0272-1716.
- [HB95] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM Press, 1995.
- [HDR] Hdrshop - high dynamic range image processing and manipulation. <http://www.debevec.org/HDRShop/>.
- [Hec86] Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics & Applications*, 6(11):56–67, November 1986.
- [Hol98] Gerald Holst. *CCD Arrays Camera and Displays*. JCD Publishing and SPIE Optical Engineering Press, 1998.
- [Hol00] Gerald Holst. *Electro-Optical Imaging System Performance*. JCD Publishing and SPIE Optical Engineering Press, 2000.
- [HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition (CVPR'97)*, pages 1106–1112, 1997.
- [ISO] Iso 14524:1999. Photography – Electronic still-picture cameras – Methods for measuring opto-electronic conversion functions (OECFs).
- [Kap84] W. Kaplan. *Advanced Calculus, 3rd ed.* Addison-Wesley, 1984.
- [KS96] A. Karasaridis and E. Simoncelli. A filter design technique for steerable pyramid image transforms. In *Int'l Conf. Acoustics, Speech, and Signal Processing*, May 1996.
- [LHS00] Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans-Peter Seidel. Automated texture registration and stitching for real world models. In *8th Pacific Conference on Computer Graphics and Applications*, pages 317–327. IEEE, October 2000. ISBN 0-7695-0868-5.
- [LYS01] Xinguo Liu, Yizhou Yu, and Heung-Yeung Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 97–106. ACM Press, 2001.
- [NK99] Peter J. Neugebauer and Konrad Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18(3):245–256, September 1999. ISSN 1067-7055.

- [OSRW97] Eyal Ofek, Erez Shilat, Ari Rappoport, and Michael Werman. Multiresolution textures from image sequences. *IEEE Computer Graphics & Applications*, 17(2):18–29, March - April 1997. ISSN 0272-1716.
- [PKVG98] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *3D Structure from Images - SMILE 1998, Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 138–153. Springer-Verlag, 1998.
- [RCM99] C. Rocchini, P. Cignoni, and C. Montani. Multiple textures stitching and blending on 3d objects. In *Eurographics Rendering Workshop 1999*, Granada, Spain, June 1999. Springer Wein / Eurographics.
- [Sha49] C. E. Shannon. Communication in the presence of noise. *Proc. IRE*, 37:10–21, Jan 1949.
- [Smi97] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.
- [Tsa87] R. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), Aug 1987.
- [WAA⁺00] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296. ACM Press/Addison-Wesley Publishing Co., 2000.
- [Wil94] R. Willson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Carnegie Mellon University, 1994.
- [WL00] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, pages 479–488. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, July 2000. ISBN 1-58113-208-5.
- [ZG01] Alexey Zalesny and Luc J. Van Gool. A compact model for viewpoint dependent texture synthesis. In Marc Pollefeys, Luc J. Van Gool, Andrew Zisserman, and Andrew W. Fitzgibbon, editors, *3D Structure from Images - SMILE 2000, Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments Dublin, Ireland, July 12, 2000, Revised Papers*, volume 2018 of *Lecture Notes in Computer Science*, pages 124–143. Springer, 2001.

- [Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision (ICCV'99)*, pages 666–673, Sept. 1999.