

Detail Synthesis for Image-based Texturing

Ryan M. Ismert*
Program of Computer Graphics
Kavita Bala†
Department of Computer Science
Donald P. Greenberg‡
Program of Computer Graphics

PCG-03-1

January 10, 2003

Interactive and immersive environments which model the real world often use images of the environment to capture realistic visual complexity. Image-based modeling techniques permit the creation of visually interesting geometric models 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.

This paper introduces a novel technique to improve the quality of image-based texturing processes by introducing a physically-based metric that can be used to extend current texture synthesis methods. We propose a sampling-based metric of texture quality based on the Jacobian matrix of the imaging transform, which captures the interaction of the imaging system with the imaged environment. This metric suggests a physical interpretation of the multi-resolution image representations widely used in texture synthesis. Use of this metric enables synthesis 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 technique relieves the user of the burden of collection large numbers of images and increases the quality of user-driven image-based modeling systems. This improved quality is important in order to create compelling visual experiences in interactive environments.

*e-mail: mccloud@graphics.cornell.edu

†e-mail: kb@cs.cornell.edu

‡e-mail: dpg@graphics.cornell.edu

Detail Synthesis for Image-based Texturing

Ryan M. Ismert*
Program of Computer Graphics

Kavita Bala†
Department of Computer Science

Donald P. Greenberg‡
Program of Computer Graphics

Abstract

Interactive and immersive environments which model the real world often use images of the environment to capture realistic visual complexity. Image-based modeling techniques permit the creation of visually interesting geometric models 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.

This paper introduces a novel technique to improve the quality of image-based texturing processes by introducing a physically-based metric that can be used to extend current texture synthesis methods. We propose a sampling-based metric of texture quality based on the Jacobian matrix of the imaging transform, which captures the interaction of the imaging system with the imaged environment. This metric suggests a physical interpretation of the multi-resolution image representations widely used in texture synthesis. Use of this metric enables synthesis 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 technique relieves the user of the burden of collection large numbers of images and increases the quality of user-driven image-based modeling systems. This improved quality is important in order to create compelling visual experiences in interactive environments.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Imaging geometry I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Texture

Keywords: sampling, image-based modeling, texture mapping

1 Introduction

Many interactive and immersive environments demand high levels of visual realism. One way that this has traditionally been accomplished for environments which model the real world is by creating surface textures through resampling actual images the scene. We

*e-mail: mccloud@graphics.cornell.edu

†e-mail: kb@cs.cornell.edu

‡e-mail: dpg@graphics.cornell.edu

refer to this process as *image-based texturing*, noting its close relationship to image-based modeling. Image-based modeling and texturing are compelling methods for creating detailed and accurate models of real world structures. Such models are valuable for cultural heritage, tourism, urban planning, and entertainment purposes. Both computer- and user-driven geometric reconstruction projects have made a great deal of recent progress[Debevec et al. 1996; Dick et al. 2000; Dick et al. 2001; Pollefeys et al. 1998]. In addition to recent improvements in the quality of reconstructed geometry, shortcomings in geometric accuracy can often be masked by detailed textures[Heckbert 1986] derived from the imaged environment.

In general, however, an image from a single camera position yields an uneven sampling of the surface of an object (even though the sampling rate is constant across the image plane), thus creating unevenly sampled textures. This effect is due to the camera pose and position with respect to each object in the scene, as well as the effects of projection and lens distortion, and is demonstrated in Figure 1.

As the distance between the location of samples in the world gets larger, the sampling rate gets smaller. The result is that the minimum resolvable feature size becomes larger. The visual effect of this is progressive blurring in the resampled textures, as shown in Figure 1, columns (c) and (d). This blurring is obviously undesirable in interactive environments that support inspection of the world from viewpoints other than the original camera positions.

Current approaches attempt to improve the quality of a texture either by merging several textures or by constructing an alternative surface appearance model such as a View-dependent Texture Map[Debevec et al. 1996; Debevec et al. 1998] or a Surface Light Field[Wood et al. 2000]. 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, accurately representing a tall building requires 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 areas with high detail, but leaving most areas with much less detail.

Texture synthesis has also been suggested as a means of addressing the poor quality of textures resampled from images. To date, work on texture synthesis has focused on synthesizing entirely new textures in 2D or on 3D surfaces, and do not take the into account issues of partial data or projective resampling images of real world scenes.

In this paper we present a solution to the problem of poor texture quality due to uneven sampling by using a detail synthesis approach. We introduce a physical, sampling-based measure of the quality of a texture. We extend existing texture synthesis algorithms to generate only missing detail using the metric. This process injects statistically correct high frequency information into areas with poor detail, preserving any detail present. This paper introduces two important contributions to image-based texturing:

- a physical, sampling-based metric using the Jacobian matrix of the imaging transform for evaluating texture quality, and

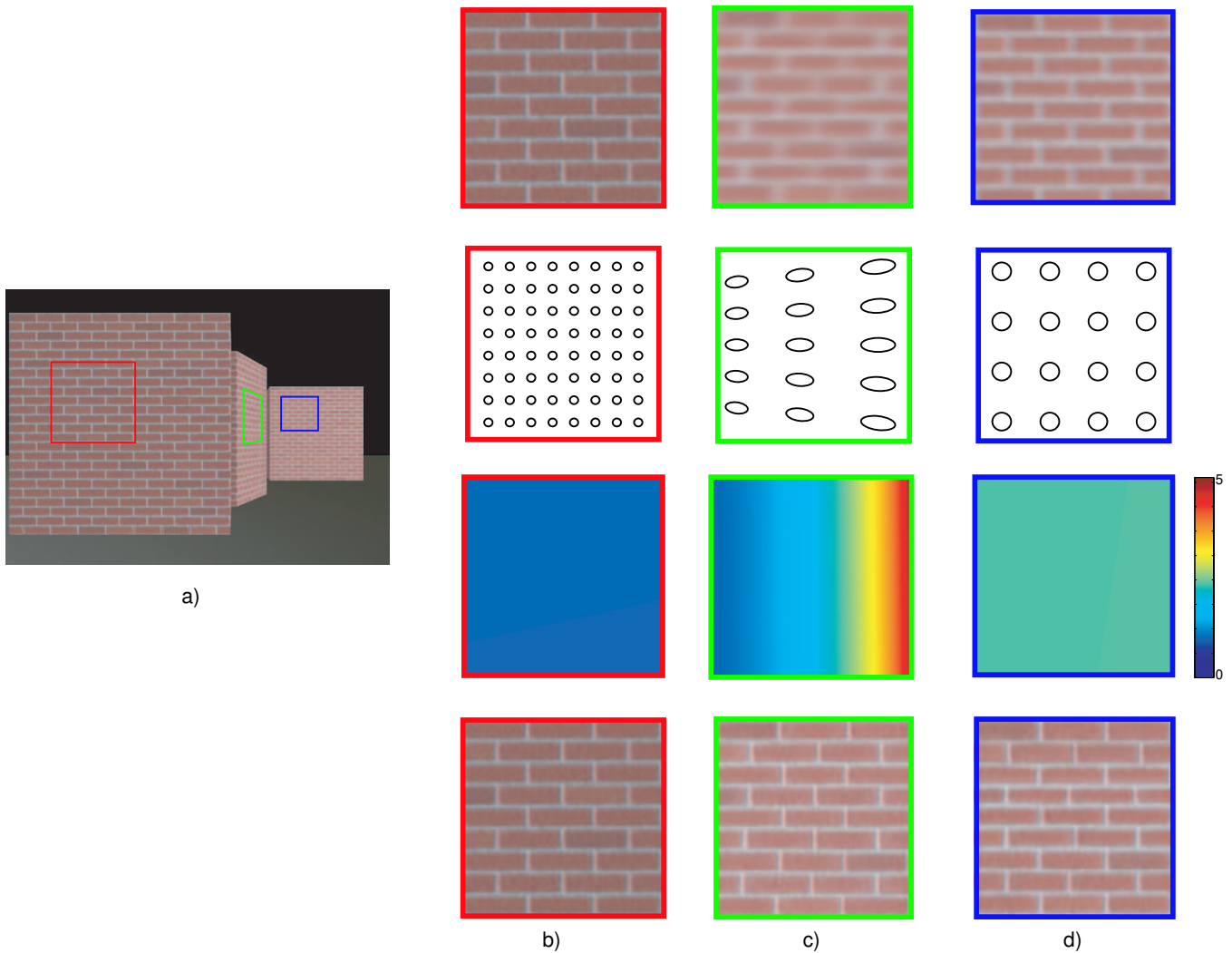


Figure 1: Three common sampling situations. (a) is the scene. Column (b) shows data from the front face, which is well sampled. Columns (c) and (d) show data from poorly sampled faces. The first row shows portions of textures from the front, middle, and far blocks, respectively. The second row illustrates the sampling patterns on each of these faces. The third row shows the maximum value of the entries in the Jacobian matrix evaluated across each face. The bottom row shows the results of applying our technique to synthesize detail onto poorly sampled faces (columns (c) and (d)) using data from the texture on the well sampled face (column (b)). Comparing the top row and the bottom row of columns (c) and (d), we see the high frequency detail inserted by our algorithm.

- a method that relates the multi-resolution pyramids used by many current texture synthesis techniques to the Jacobian-based metric, allowing synthesis driven by a physically-based, quantitative measure of texture quality.

The sampling-based metric of texture quality is based on and captures the interaction of imaging systems with the imaged environment (for example, see Figure 1 (a)). The metric uses the Jacobian matrix of the imaging transform which maps the surface of an object into the image from which a texture will be extracted. The top row of Figure 1 shows portions of textures extracted from an image. Note that the quality of the extracted textures varies widely. The quality variations are due to the way in which the surfaces were sampled (Figure 1, second row). The Jacobian-based metric enables us to quantify the sampling behavior across texture space (Figure 1, third row) and leads directly to a physical interpretation of the multi-resolution image representations widely used in texture synthesis, as described in section 4. We 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. The resulting synthesized texture using our technique is shown in Figure 1, columns (c) and (d), bottom row.

Our technique allows the creation of models with uniform, high-resolution textures from small input image sets, and significantly, from sets which are poorly distributed in the space of desired viewing positions. Generating textures with these qualities constitutes a significant improvement in the ability to generate highly detailed, reality-preserving models with reasonable input constraints for users. Our technique eases the burden of collecting large numbers of images and increases the quality of textures derived from images of the environment. This improved quality is important in order to create compelling visual experiences in interactive environments that model the real world.

The rest of this paper is organized as follows: Section 2 is a discussion of previous related work. Section 3 describes the use of the Jacobian as a texture quality metric. Section 4 describes how we apply the metric to the task of detail synthesis, and Section 5 presents the results of that technique.

2 Related Work

Our research 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 has been studied extensively from the standpoint of generating an arbitrarily sized texture patch from a small example patch. Heeger and Bergen [Heeger and Bergen 1995] and DeBonet [DeBonet 1997] presented the basic approach using steerable pyramids. Wei and Levoy [Wei and Levoy 2000] demonstrated synthesis with Gaussian pyramids and introduced a Tree-structured Vector Quantization accelerated search method. These approaches have not been directly applied to image-based texturing.

Freeman et al. [Freeman et al. 2002] describe a synthesis approach for sharpening images. Their approach is shown synthesizing one octave of data, and 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 [Zalesny and Gool 2001] demonstrate a method of synthesizing oriented textures for surfaces, but their ap-

proach does not model texture quality, and thus replaces the low resolution texture, rather than using it to guide the synthesis.

The approach introduced in this paper builds on these algorithms (primarily Wei and Levoy [Wei and Levoy 2000]) 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.

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 [Lensch et al. 2000], as well as larger items such as statuary [Bernardini et al. 2001]); and large-scale objects such as buildings [Debevec 1996]. Although we are interested in applying the technique presented in this paper exclusively 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 [Bernardini et al. 2001; Lensch et al. 2000; Neugebauer and Klein 1999; Rocchini et al. 1999]. These techniques do not perform detail synthesis, but instead merge 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 [Wood et al. 2000], and Bi-directional Texture Functions [Liu et al. 2001] (BTFs) can also 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. And although BTFs have been demonstrated using smaller numbers of input 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 [Debevec et al. 1996; Debevec et al. 1998] (VDTMs) have been proposed as a way to improve the appearance of these often complex surfaces. 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 [Debevec 1996]. 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 of the top of a facade of a tall building or for the sides of many structures in constricted urban settings.

2.2.3 Texture quality metrics

Texture quality metrics are often used by image-based texturing techniques, which need to select or weight contributions to a texture from pixels in several images. 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. [Debevec et al. 1996], 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 [Lensch et al. 2000; Rocchini et al. 1999] also use the same metric.

Bernardini et al. [Bernardini et al. 2001] 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. [Ofek et al. 1997] and Neugebauer and Klein [Neugebauer and Klein 1999] 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.

3 Texture Quality Metric

In this section we describe the problem of characterizing the sampling behavior of an image sensor during the image formation process. We then show how the values of the Jacobian matrix of the imaging transform can be interpreted as sample distances within the environment, and how the Jacobian values then serve to measure texture quality.

3.1 Terminology

We orient our discussion of imaging and image-based texturing by noting that they can both be considered as sampling problems. A camera samples a scene to produce a digital *image* - a computer file containing digital values for colors on a regular grid. We refer to pixels in the image as *sensor elements* or *sensels*. We refer to sensel positions using coordinates of *image space*, a discrete two dimensional space parameterized in (u, v) , with an origin at the top left of the image. We assume that the 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 $\langle \text{imagewidth} \rangle$ and 0 to $\langle \text{imageheight} \rangle$, respectively.

The spatial frequencies in the scene that can be captured by the image sensor are determined by projecting the centers of adjacent sensels into the environment, and then determining the distance between them. We call this distance the *sample distance*. More frequently we refer to this quantity in terms of the *sampling rate*, $\frac{1}{\text{sample distance}}$. Basic sampling theory tells us that the highest spatial frequency that can be captured, the *Nyquist frequency*, is $\frac{\text{sampling rate}}{2}$. As the sampling rate varies across the scene so does the Nyquist frequency.

Note that although the sample values recorded by the camera are uniformly distributed in image space, it is generally not the case that the scene locations that were sampled by the camera are uniformly distributed in the scene, due to the pose and position of the camera and the effects of perspective. This is illustrated in Figure 2. Figure 2 (a) shows the view from the camera with sensel boundaries shown in red. The camera is looking toward a rectangle in the environment, being viewed in perspective, which is shown outlined in blue. Figure 2 (b) shows a view of the same rectangle viewed orthogonally, with the world space units shown in solid blue lines. The outlines of the sensels as they project onto the rectangle are shown with dashed red lines. Although arranged regularly in the image, the sensels clearly do not sample the rectangle regularly, due to perspective and the position of the camera.

The sampling of the scene by the camera (i.e., the image formation process) can be represented as the mapping of textures onto objects, then onto the image sensor. This mapping can be described by the *imaging transform*, \mathbf{M}_{Img} :

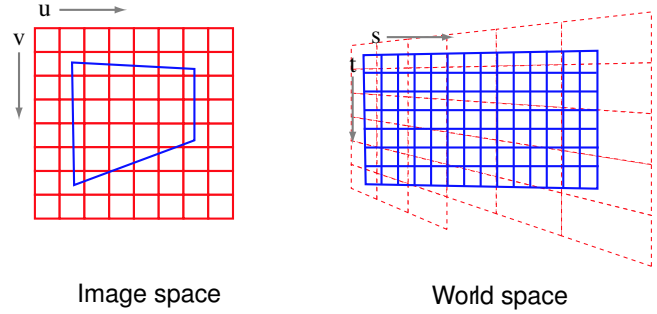


Figure 2: At left an image (red) captures a surface (blue) in perspective. At right, the same surface is viewed from an orthogonal direction with the image sensor sampling overlaid in dashed red lines.

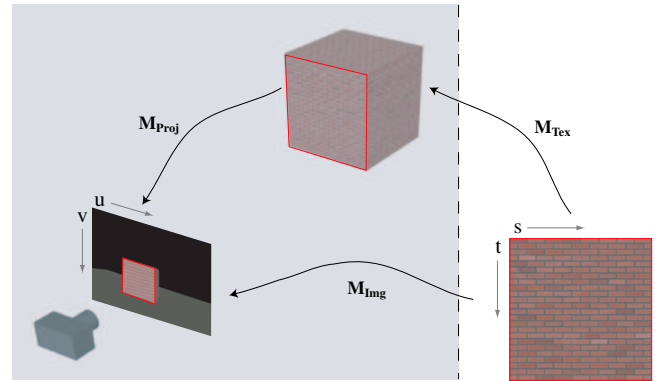


Figure 3: \mathbf{M}_{Img} , the imaging transform, is composed of \mathbf{M}_{Proj} and \mathbf{M}_{Tex} .

$$\mathbf{M}_{\text{Img}} = \mathbf{M}_{\text{Proj}} \mathbf{M}_{\text{Tex}} \quad (1)$$

As shown in Figure 3, \mathbf{M}_{Img} is composed of two transforms: \mathbf{M}_{Proj} , which projects an object into the image plane, and \mathbf{M}_{Tex} , which projects the surface texture onto the object.

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

$$\mathbf{M}_{\text{Proj}} = \mathbf{M}_{\text{Dist}} \mathbf{M}_{\text{Int}} \mathbf{M}_{\text{Ext}} \mathbf{M}_{\text{Obj}} \quad (2)$$

where \mathbf{M}_{Dist} is the lens distortion model matrix, \mathbf{M}_{Int} is the transform associated with the intrinsic camera parameters, \mathbf{M}_{Ext} is the transform associated with the extrinsic camera parameters which, together with \mathbf{M}_{Int} , maps the world into image space, and \mathbf{M}_{Obj} is the transform that maps the object into world space.

We store the appearance of an object in a *texture*, which is a discrete, two dimensional representation parameterized in (s, t) . Pixels in the texture are referred to as *texels*. A texture maps to a surface by the *texture transform*, \mathbf{M}_{Tex} , which defines a sampling of the surface with a constant sampling rate. It is constant because we assume that the surface parameterization matches the texture parameterization except for a scaling factor, so \mathbf{M}_{Tex} represents only 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 for the texture.

3.2 Physically-based sampling metric

Optimally, image-based textures should be constantly sampled at a constant, high rate. Current approaches attempt to approximate this sampling, but do not actually reconstruct textures with constant sampling behavior and have other undesirable properties, as noted previously. In order to reconstruct textures with constant sampling rates, we require a *physical, sampling-based texture quality measure*.

3.2.1 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 2D coordinates (s, t) , since this is required for the texture mapping operation in any case. The object vertices are given in (homogeneous) coordinates of some world units. So we have:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{M}_{Proj} * \begin{pmatrix} s \\ t \\ 1 \end{pmatrix} \quad (3)$$

This transformation is illustrated in Figure 3. As noted previously, this is a standard way to express the “imaging” of the environment in computer graphics. However, we are interested in framing the problem slightly differently. If \mathbf{M}_{Proj} projects an object surface into the image, then we are interested in the inverse problem: how does \mathbf{M}_{Proj}^{-1} project the image data onto an object surface? In keeping with this perspective, from this point forward we deal primarily with the inverse of \mathbf{M}_{Proj} .

This projection of image data onto an object surface is isomorphic to the sampling of the scene by the camera. The sampling induced by the transformation can be characterized by the Jacobian matrix [Kaplan 1984], $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$:

$$\mathbf{J}(\mathbf{M}_{Proj}^{-1}) = \begin{bmatrix} \frac{\partial s}{\partial u} & \frac{\partial s}{\partial v} \\ \frac{\partial t}{\partial u} & \frac{\partial t}{\partial v} \end{bmatrix} \quad (4)$$

The values in the Jacobian matrix, as partial derivatives, indicate 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. 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 of $\mathbf{J}(\mathbf{M}_{Proj}^{-1})$.

For example, given some \mathbf{M}_{Proj}^{-1} such that at some point on the surface:

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

If world coordinates are given in units of centimeters, the Jacobian matrix 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 $\frac{\text{sampling rate}}{2}$, or in this case $\frac{1}{5*2} \frac{\text{cycles}}{\text{cm}}$, in both the horizontal and vertical directions. Thus the smallest feature¹ that can be detected without aliasing is 10cm in height or width.

¹Here, in keeping with the sampling nomenclature, a ‘feature’ is a complete cycle of an intensity profile sinusoid.

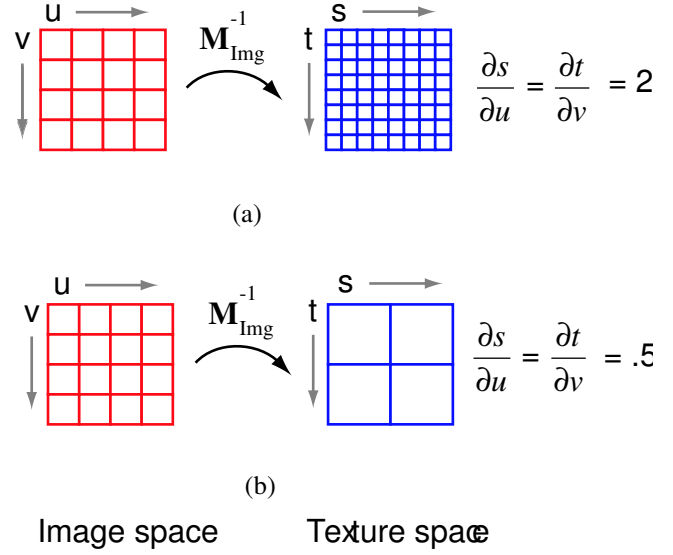


Figure 4: A simple example illustrating the use of the elements of \mathbf{J} to judge sample rate. In (a) the texture has more resolution than is necessary to capture the detail in the image. In (b) the texture has too little resolution and cannot capture all of the detail present in the image.

3.2.2 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 world units are assumed to have a constant scaling factor relationship to the resolution 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 same form:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{M}_{Img} * \begin{pmatrix} s \\ t \\ 1 \end{pmatrix} \quad (6)$$

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 texture representation we deal primarily with the inverse of the imaging transform, \mathbf{M}_{Img}^{-1} . This transformation is characterized by the Jacobian matrix, $\mathbf{J}(\mathbf{M}_{Img}^{-1})$.

Evaluating the Jacobian matrix across the texture gives four measures per texel. Values on the diagonal indicate the relative sample distances in the direction of the projected u and v axes. This is illustrated in Figure 4. The off-diagonal values 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, these values (not shown) are zero. As Figure 4 (a) shows, values in the Jacobian matrix > 1 indicate that the 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 parametrization is

sufficiently dense to represent all of the data captured by the image sensor (Figure 4 (a)); values < 1 indicate that some data in the image cannot be represented with the given texture parametrization (Figure 4 (b)).

4 Detail Synthesis

Given a metric for assessing the sampling quality of image-based textures, we can now return to the problem of extracting uniformly high quality textures from few images. As noted previously, when textures are extracted from a few images, one is faced with the situation shown in Figure 1 - some faces are well sampled, but some (or even most) are not. Consequently, in this situation, the extracted textures are of poor quality. We now show how to correct the poor texture quality through detail synthesis.

4.1 Multi-resolution image pyramids

The technique presented in this paper uses Laplacian pyramids [Burt and Adelson 1983] 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. 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.

One can see that there is a 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 values equal to the sample distance. Thus, for distributing data to the levels of the pyramid, $f_N = \frac{1}{2 * \text{Jacobian_value}}$.

If we assume that in some portion of the texture, the Jacobian metric is equal to 1 (i.e., the sampling is perfect), then the data in the remainder of the texture can easily be assigned to the appropriate pyramid level. Regions with sampling rates between .5 and .25 cycles per pixel would have data in and above level 0. Regions with sampling rates between .25 and .125 cycles per pixel would have data in and above level 1. Regions with sampling rates 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.

The primary 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 levels. So 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. The preservation of existing information is important for image-based textures to be reality preserving, and is why we refer to our method as detail synthesis. We now consider the entire process in greater detail.

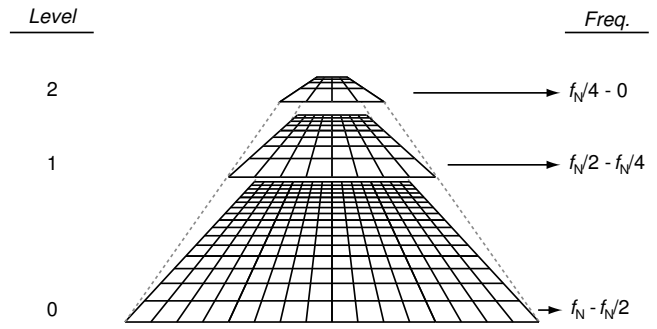


Figure 5: An illustration of a Laplacian pyramid with three levels. Level 0 contains the frequencies from f_N , to $\frac{f_N}{2}$. Level 1 contains the frequencies from $\frac{f_N}{2}$ to $\frac{f_N}{4}$. Level 2 contains the frequencies from $\frac{f_N}{4}$ to 0. Note that illustration does not show pyramid data in order to preserve clarity.

4.2 Synthesis Overview

Given two textures from similar surfaces, one with a high sampling rate (the *source texture*) and one with a low sampling rate (the *target texture*), we wish to synthesize missing higher frequency data into the target texture, using the source texture as a model.

Current approaches to multi-resolution texture synthesis begin by creating some pyramid representation (for example, Gaussian, Laplacian, or steerable) of a source texture that has the desired frequency content. Then a target pyramid is created with a pyramid height equal to that of the source pyramid. Each level of the target pyramid is usually seeded with noise. The noise is generated so that its histogram matches that of the corresponding level in the source pyramid. Correct texture data is then generated for each level of the target pyramid (from the top down) based on statistical matching with the data in the source pyramid, usually considering a neighborhood in the current level and some number of corresponding ‘parent’ neighborhoods in higher levels. For more detail we refer the reader to the texture synthesis work referenced in section 2.1.

In extending texture synthesis algorithms to image-based texturing we perform the steps briefly illustrated in Figure 6. First a Laplacian pyramid is constructed for the source texture as shown in Figure 6 (a). Next the pyramid for the target texture is constructed by inserting the texture into the pyramid at the level indicated by the texture’s maximum Jacobian value. Lower levels of the pyramid are seeded with noise. As shown in Figure 6 (b) the bottom level is seeded with noise, and the target texture – with the maximum value of the Jacobian metric equal to 3.0 – is inserted into level 1. 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. Finally, texture synthesis is performed in the target texture image pyramid to generate the high quality result texture, as demonstrated in Figure 6 (c).

Of course this is a simplified overall description and several initialization and normalization steps are required. The remainder of this section discusses the process in greater detail. The complete detail synthesis process using the Jacobian metric is as follows:

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.
2. **Normalize the target texture with respect to the source texture scaling**. This yields a set of Jacobian values that describe

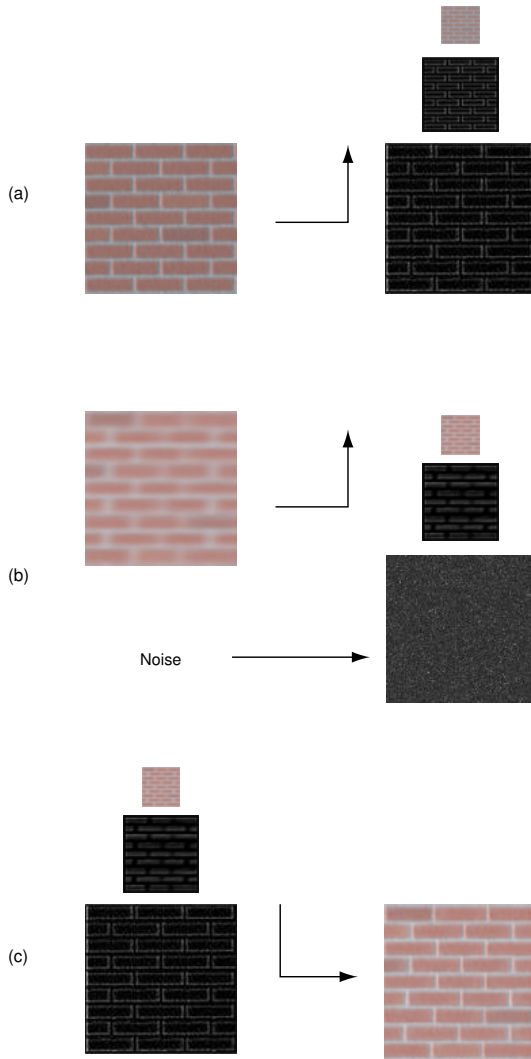


Figure 6: The detail synthesis process. The Laplacian pyramid for source texture is constructed in (a). The Laplacian pyramid for the target is constructed in (b). The synthesis and resulting texture are shown in (c).

the sampling rate of the target relative to the source. That is, it answers the question, ‘where in the target pyramid does this data from the target texture appear?’

3. **Construct a pyramid representation of the source texture.** The height of the pyramids are determined by the lowest level in the target pyramid at which data appears, as well as the number of parent levels (if any) used by the synthesis algorithm.
4. **Construct a pyramid representation of the target texture.** This process must correctly insert data from the target texture into the appropriate level of the target pyramid.
5. **Perform texture synthesis.** Using the complete source pyramid and the partial target pyramid, synthesize the data still absent into the target pyramid, then reconstruct the texture.

We now discuss each of these steps in more detail.

4.3 Determining the source texture scaling

The purpose of this step is to refine the mapping \mathbf{M}_{Tex} , such 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 assume that the texture parameterization has a constant scaling relation to the underlying surface parameterization, thus the determination of the optimal source texture representation involves computing a factor to scale the transform \mathbf{M}_{Tex} and then applying the inverse of that factor to scale the texture resolution extents. We begin by calculating the Jacobian matrix across the texture assuming that \mathbf{M}_{Tex} is the identity matrix (i.e., take the Jacobian matrix of \mathbf{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. This value, 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:

$$\mathbf{M}_{Tex} = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

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

There is a side effect of using the conservative value, k , to bound the Jacobian values over the surface. The resulting texture is generally only of sufficient 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 when one considers that we will ultimately construct a Laplacian pyramid from the source texture. If we selected 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 would be portions of (at least) the lowest level of the pyramid that would not contain any data. This would cause portions of the statistical model for the source texture to contain incorrect (or incomplete) data and would cause the synthesis algorithm to perform poorly.

4.4 Normalizing the target texture

The optimal parametrization of the source texture space is calculated using the scale factor, k , derived directly from the calculated Jacobian values across the source surface. Now we 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 determine how target texture data should be inserted into the target pyramid, and second to extract a target texture of an appropriate resolution.

For the target we apply the source scaling factor, k , to the target texture resolution and use the source \mathbf{M}_{Tex} . Calculating the values of the Jacobian matrix across the target texture now yields 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:

$$J(M_{Proj}^{-1}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (8)$$

then the sample distance in the target texture's s and t directions is twice that of the source. 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.

The above example together with the analysis of multiresolution pyramids provided in section 4.1 suggests a straightforward way to relate the target image pyramid to the maximum Jacobian value over the target texture. Recall that the pyramids are bandpass, and the Jacobian gives a measure of sampling rate, thus there is a direct and meaningful physical interpretation of their relationship. Each level of the pyramid contains an octave of data, thus a texel with the largest entry in its Jacobian equal to r , can only possibly contribute to data present at or above the $\log_2(r)$ level of the pyramid.

Using the maximum of the calculated Jacobian values over the target surface, the lowest level of the pyramid that can contain all of the texture data is calculated. Note that this is a conservative calculation, like the one used to determine the source texture scaling - i.e. we want the lowest *full* level of the Laplacian pyramid, even if it means that some higher frequency data is not represented. The texture resolution and the target M_{Tex} are recalculated based on the size of this level, and an appropriate texture is extracted from the image data.

4.5 Creating the Laplacian pyramids

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 level of the target pyramid with data, as determined above, and adding the number of parent levels that the synthesis algorithm will consider. Creation of the source pyramid then follows the standard algorithm for creating a Laplacian pyramid.

To populate the target pyramid, we begin with an empty pyramid with the same height as the source pyramid. Population of the target pyramid with data is performed as follows: The extracted target texture is inserted directly into the pyramid at the appropriate level. That level and all higher levels are then processed as per the standard Laplacian pyramid algorithm. Levels below the lowest valid level are populated with histogram-equalized noise, matching the histogram of the corresponding level of the source pyramid. This process is shown in Figure 6 (b).

4.6 Synthesis

The actual synthesis is based on the algorithm by Wei and Levoy [Wei and Levoy 2000]. 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. (Pixel neighborhoods are used instead of patches, but Freeman et al. [Freeman et al. 2002] demonstrate a similar approach using patches.) The actual search uses a k-d tree to accelerate matching of the pixel vectors.

We believe that this application of detail synthesis to image-based model textures is unique. Wei and Levoy [Wei and Levoy 2000] and Freeman, et al. [Freeman et al. 2002] 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 paper allows the application of these methods to textures on the surfaces of imaged objects for the creation of interactive environments.

5 Preliminary Results

5.1 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 de-mosaiced 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 n. d.], and the opto-electronic conversion function was confirmed to be linear using HDRShop [HDR n. d.] and ISO14524 [ISO n. d.b]. Although the 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 modeled 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 modeled with commercially available image-based modeling software (RealVIZTM ImageModelerTM), 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.

5.2 Rendered bricks

The brick cubes shown in Figure 1 were rendered at 800-by-600 using Discreet 3DSMaxTM, then modeled again using ImageModelerTM. The front face of the front cube (column b) was used as the source texture; the side of the middle cube (column c) and the front of the back cube (column d) were the target textures. After re-scaling 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 1, columns c and d. 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.

5.3 Building columns

The input images for the results shown in Figure 7 were imaged with the Canon D30 camera described above. The front-facing side of the nearest column was used as the source. The corresponding face of the fourth column was used as the target. After re-scaling M_{Tex} the maximum Jacobian value for the target texture was approximately 3.1, with only a small range across the surface. The

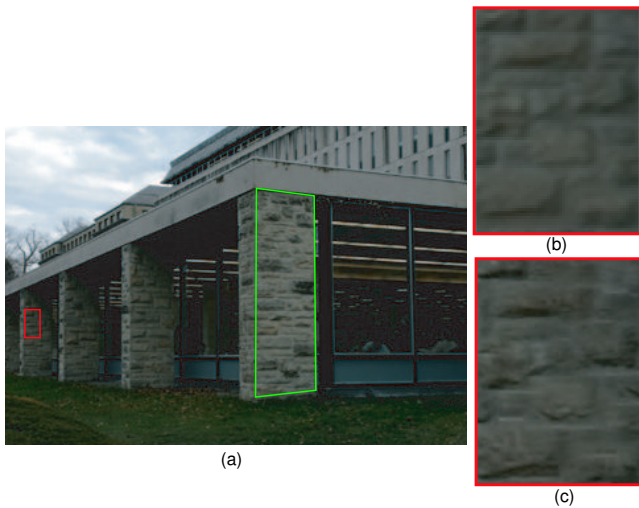


Figure 7: Results for columns along the face of a library. The front facing side of the nearest column was the source (shown in green). A portion of the target (the third column) is shown in red. (b) is the original target texture. (c) is the resulting target texture after synthesis.

synthesis algorithm used a pyramid with 4 levels, and used 4 passes with 5×5 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, and the strong differences in appearance between the source and the target.

5.4 Pavement

The input images for these results were also imaged with the Canon D30. The image shown in Figure 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 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 5×5 neighborhoods to generate the results shown.

As the results show, 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.

6 Conclusions and Future Work

This paper has presented two important contributions to aid the creation of high-quality textured environments from imaged data. This improved texture quality is important for interactive walkthroughs where the user's viewpoint can differ from the original sampled

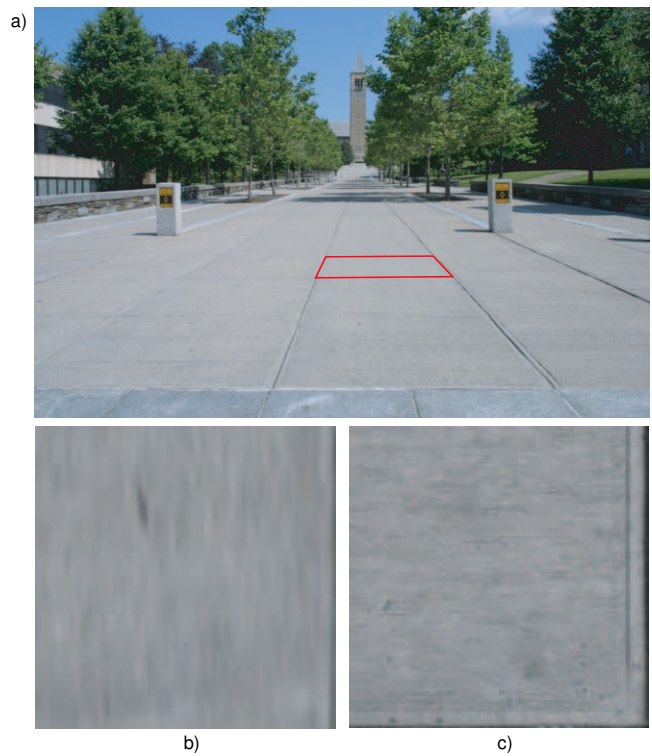


Figure 8: Results for a paved pedestrian walkway. The image in a) was used to resample the texture for the target pavement section (outlined in red). A portion of the original texture for this pavement section is shown in b). The results after synthesis are shown in c).

camera locations. First, it has shown how the Jacobian matrix of the imaging transform can be used to determine sampling behavior in a scene, and thus provides a simple, physically-based texture quality metric. Second, it has shown how this texture quality metric enables our novel technique for synthesizing high frequency detail into degraded regions of image-based textures using standard texture synthesis techniques. In contrast to most previous techniques, this synthesis occurs in texture space – on the surface of some object in the environment – rather than in image space. This technique provides a way to extract uniformly high quality textures of a model from no more images than are required to model the geometry of that model.

In the future, we will explore the use of alternate multi-resolution image representations (e.g. wavelets) to better address the highly anisotropic nature of the sampling in many scenes. We will also explore approaches combining several images based on Jacobian values, and synthesizing only detail missing in the merged result.

Acknowledgements

Special thanks to Steve Westin for his assistance to us in understanding all manner of imaging and sampling topics.

References

- BERNARDINI, F., MARTIN, I. M., AND RUSHMEIER, H. 2001. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics* 7, 4 (October - November), 318–332. ISSN 1077-2626.

2001. *Modulation transfer function in optical and electro-optical systems*. SPIE Press.
- BURT, P. J., AND ADELSON, E. H. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31,4*, 532–540.
- Camera calibration toolkit for matlab.
www.vision.caltech.edu/bouguetj/calib_doc/.
- DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH 96*, ACM SIGGRAPH / Addison Wesley, New Orleans, Louisiana, Computer Graphics Proceedings, Annual Conference Series, 11–20. ISBN 0-201-94800-1.
- DEBEVEC, P. E., YU, Y., AND BORSHUKOV, G. D. 1998. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Rendering Workshop 1998*, Springer Wein / Eurographics, Vienna, Austria, 105–116. ISBN 3-211-83213-0.
- DEBEVEC, P. 1996. *Modeling and Rendering Architecture from Photographs*. PhD thesis, University of California at Berkeley.
- DEBONET, J. S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of SIGGRAPH 97*, ACM SIGGRAPH / Addison Wesley, Los Angeles, California, Computer Graphics Proceedings, Annual Conference Series, 361–368. ISBN 0-89791-896-7.
- DICK, A., P.TORR, , AND CIPOLLA, R. 2000. Automatic 3d modelling of architecture. In *Proc. 11th British Machine Vision Conference (BMVC'00)*, 372–381.
- DICK, A., TORR, P., RUFFLE, S., AND CIPOLLA, R. 2001. Combining single view recognition and multiple view stereo for architectural scenes. In *Proc. 8th IEEE International Conference on Computer Vision (ICCV'01)*, 97–106.
- FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. Example-based super-resolution. *IEEE Computer Graphics & Applications* 22, 2 (March-April), 56–65. ISSN 0272-1716.
- Hdrshop - high dynamic range image processing and manipulation.
<http://www.debevec.org/HDRShop/>.
- HECKBERT, P. S. 1986. Survey of texture mapping. *IEEE Computer Graphics & Applications* 6, 11 (November), 56–67.
- HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, 229–238.
- HOLST, G. 1998. *CCD Arrays Camera and Displays*. JCD Publishing and SPIE Optical Engineering Press.
- HOLST, G. 2000. *Electro-Optical Imaging System Performance*. JCD Publishing and SPIE Optical Engineering Press.
- Iso 12233:2000. Photography – Electronic still-picture cameras – Resolution measurements.
- Iso 14524:1999. Photography – Electronic still-picture cameras – Methods for measuring opto-electronic conversion functions (OECFs).
- KAPLAN, W. 1984. *Advanced Calculus, 3rd ed.* Addison-Wesley.
- LENSCH, H. P. A., HEIDRICH, W., AND SEIDEL, H.-P. 2000. Automated texture registration and stitching for real world models. In *8th Pacific Conference on Computer Graphics and Applications*, IEEE, 317–327. ISBN 0-7695-0868-5.
- LIU, X., YU, Y., AND SHUM, H.-Y. 2001. Synthesizing bidirectional texture functions for real-world surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, 97–106.
- MATSUSHITA, K., AND KANEKO, T. 1999. Efficient and handy texture mapping on 3d surfaces. *Computer Graphics Forum* 18, 3 (September), 349–358. ISSN 1067-7055.
- NEUGEBAUER, P. J., AND KLEIN, K. 1999. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum* 18, 3 (September), 245–256. ISSN 1067-7055.
- OFEK, E., SHILAT, E., RAPPOPORT, A., AND WERMAN, M. 1997. Multiresolution textures from image sequences. *IEEE Computer Graphics & Applications* 17, 2 (March - April), 18–29. ISSN 0272-1716.
- POLLEFEYS, M., KOCH, R., VERGAUWEN, M., AND GOOL, L. V. 1998. 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*, Springer-Verlag, vol. 1506 of *Lecture Notes in Computer Science*, 138–153.
- ROCCHINI, C., CIGNONI, P., AND MONTANI, C. 1999. Multiple textures stitching and blending on 3d objects. In *Eurographics Rendering Workshop 1999*, Springer Wein / Eurographics, Granada, Spain.
- TSAI, R. 1987. 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).
- WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 479–488. ISBN 1-58113-208-5.
- WOOD, D. N., AZUMA, D. I., ALDINGER, K., CURLESS, B., DUCHAMP, T., SALESIN, D. H., AND STUETZLE, W. 2000. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 287–296.
- ZALESNY, A., AND GOOL, L. J. V. 2001. A compact model for viewpoint dependent texture synthesis. In *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*, Springer, M. Pollefeys, L. J. V. Gool, A. Zisserman, and A. W. Fitzgibbon, Eds., vol. 2018 of *Lecture Notes in Computer Science*, 124–143.