Published at Interactive 3D Graphics (I3D) 2003, Monterey, CA

Detail Synthesis for Image-based Texturing

Ryan M. Ismert*
Program of Computer Graphics
Cornell University

Kavita Bala†
Department of Computer Science
Cornell University

Donald P. Greenberg‡
Program of Computer Graphics
Cornell University

Abstract

Image-based modeling techniques permit the creation of visually interesting geometric models from photographs. But traditional image-based texturing (IBT) techniques often result in extracted textures of poor, uneven quality. This paper introduces a novel technique to improve the quality of image-based textures. We compute a simple and efficient texture quality metric based on the Jacobian of the imaging transform. We identify the correlation between the values of the Jacobian metric and the levels of an image pyramid, allowing us to formulate a novel texture synthesis approach which operates over textures from 3D surfaces in a scene. Our technique allows the creation of uniform, high-resolution textures, relieving the user of the burden of collecting large numbers of images while increasing the visual quality of image-based models. This improved quality is important to create compelling visual experiences in interactive environments.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Texture; I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Texture

Keywords: Image-based modeling, texture mapping.

1 Introduction

Many interactive and immersive environments use image-based texturing (IBT) to achieve high levels of visual realism, by creating textures from images of the real world. These methods create compelling, detailed and accurate models of real world structures, which are valuable for cultural heritage, tourism, urban planning, and entertainment applications. In general, using a single image to texture a surface results in blurry or stretched resampled textures, due to uneven sampling of the surface of an object. 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. To achieve reasonable texture quality a uniform sampling of every surface in the scene is required – that is, a dense, well-distributed sampling of all possible viewpoints. For example, accurately texturing a tall building requires difficult to obtain images of the top of the building facade. The blurring and loss of detail that results from

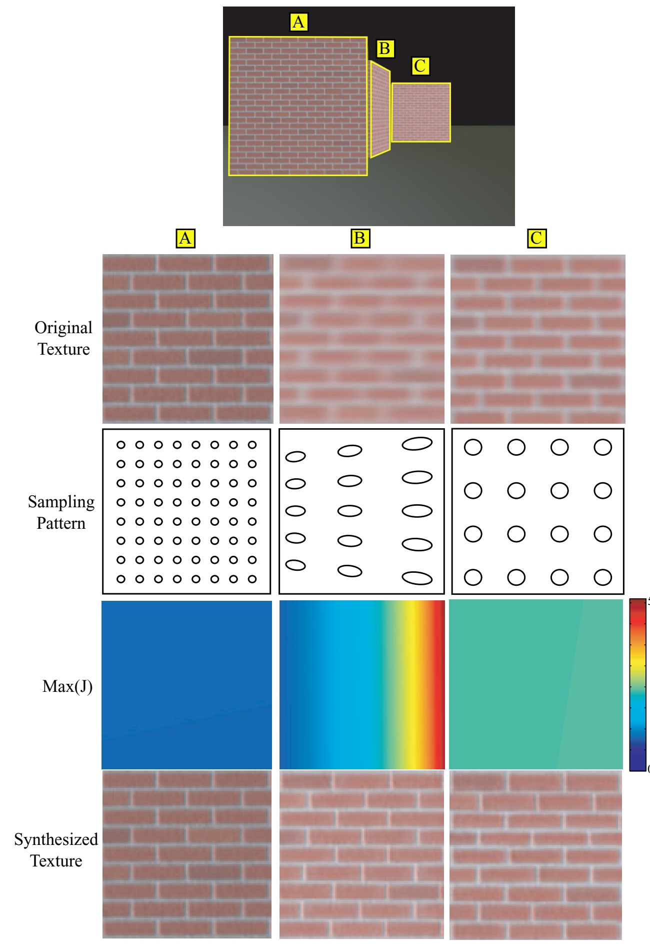


Figure 1: Detail synthesis for the synthetic brick scene at the top. The first two rows show resampled texture regions and the corresponding camera sampling patterns. The next row shows our Jacobian metric across each surface. Using this metric to drive synthesis, we achieve the results shown in the bottom row. Comparing the top row and the bottom row for faces B and C, we see the high frequency detail inserted by our algorithm. Low frequency information, such as the color of the individual bricks, is also preserved.

using fewer images is obviously undesirable in interactive environments that support inspection of the world from viewpoints other than the original camera positions.

In this paper we present a solution to the problem of poor image-based texture quality, using a detail synthesis approach. This paper introduces two important contributions to image-based texturing. First, we identify the relationship between a simple, efficient, physically-based texture quality metric (using the Jacobian of the imaging transform), and the levels of an image pyramid. Second,

*e-mail: mccloud@graphics.cornell.edu
†e-mail:kb@cs.cornell.edu
‡e-mail:dpg@graphics.cornell.edu

we show how this relationship provides the additional information necessary to use texture synthesis to improve the quality of degraded and perspective distorted textures on 3D object surfaces. Our process, which we refer to as *detail synthesis*, creates high frequency texture data into areas with poor detail, while preserving any data present. It allows the creation of models with uniform, high-resolution textures from small, poorly distributed input image sets. 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 discusses related work, Section 3 describes the Jacobian-based texture quality metric, Section 4 describes how the metric is used for detail synthesis, Section 5 presents results, and we conclude in Section 6.

2 Related Work

Our research is related to previous research in both texture synthesis and image-based texturing (IBT).

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 [1995] and DeBonet [1997] presented the basic approach using steerable pyramids. Efros and Leung [1999] demonstrated synthesis with Gaussian pyramids, and Wei and Levoy [2000] introduced tree-structured vector quantization to accelerate the process. These approaches have not been directly applied to image-based texturing.

Freeman et al.[2002] describe a synthesis approach for sharpening images. Their approach is shown synthesizing one octave of data in image space. We expect that our texture metric could be useful for extending their training based approach to 3D surfaces.

Zalesny and Van Gool[2000] demonstrate a method of synthesizing oriented textures for surfaces by estimating the oriented appearance of the surface for each new view.

The approach introduced in this paper builds on these algorithms (primarily Efros and Leung [1999]) to allow higher quality image-based texturing from 3D surfaces using real camera information. Our algorithm synthesizes only the detail missing from the original images, not the entire texture.

Image-based texturing. 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 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.

Another class of image-based texturing approaches, such as surface light fields[Wood et al. 2000], bi-directional texture functions[Liu et al. 2001] (BTFs), and view-dependent texture maps[Debevec et al. 1996] (VDTMs) construct higher dimensional representations of the surface appearance to capture view dependence. These techniques generally require large sets of input images with a dense sampling of all possible viewpoints.

Texture quality metrics. Texture quality metrics are often used by image-based texturing techniques, to evaluate the quality of the sampling for a texel. Many methods [Debevec et al. 1996; Lensch et al. 2000; Rocchini et al. 1999] use a metric involving the viewing angle. As noted by 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. 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. Ofek et al.[1997] and Neugebauer and Klein[1999] use projected pixel area to evaluate texture quality. Their metric is closely related to the Jacobian metric used in this paper, but our metric additionally preserves directional information.

3 Texture Quality Metric

In this section we describe the problem of evaluating the quality of reconstruction for the data at a given texel on a surface. We show how the values of the Jacobian matrix of the imaging transform provide a physical measure of texture quality.

3.1 The imaging transform

The image formation process is represented as the mapping of textures onto objects, then onto the image sensor. This mapping can be described by the *imaging transform*, $\mathbf{M}_{Img}$, which maps texture space to image space ($\Re^2_{(s,t)} \rightarrow \Re^2_{(u,v)}$). $\mathbf{M}_{Img}$ is composed of $\mathbf{M}_{Proj}$, which projects an object into the image plane, and $\mathbf{M}_{Tex}$, which projects the surface texture onto the object. Thus $\mathbf{M}_{Img} = \mathbf{M}_{Proj}\mathbf{M}_{Tex}$. $\mathbf{M}_{Proj}$ is composed as $\mathbf{M}_{Proj} = \mathbf{M}_{Dist}\mathbf{M}_{Int}\mathbf{M}_{Ext}\mathbf{M}_{Obj}$, 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.

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 The sampling metric

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. In order to measure this variation, and to reconstruct textures with constant sampling rates, we require a physical, sampling-based texture quality measure.

The sampling induced by the imaging transform can be characterized by the Jacobian matrix[Kaplan 1984], $\mathbf{J}(\mathbf{M}^{-1}_{Img})$:

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

Recall that $\mathbf{M}_{Img}$ maps (s,t) to (u,v). 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 $\mathbf{M}^{-1}_{Img}$, $\mathbf{J}(\mathbf{M}^{-1}_{Img})$ accounts for all of the factors affecting the sampling rate, i.e. projective effects, camera pose and position, and lens distortion. Since the sample distance determines the Nyquist frequency, quantitative claims may be made regarding the frequencies present under $\mathbf{M}^{-1}_{Img}$, based on the values of $\mathbf{J}(\mathbf{M}^{-1}_{Img})$.

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. 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. 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.

Our metric is the maximum value of the Jacobian matrix per texel. This is a conservative measure of the texture quality at a point because it is the largest distance in world units between any two adjacent samples. Our metric is defined as:

$$\max[\mathbf{J}(\mathbf{M}_{Img}^{-1})] = \max(\frac{\partial s}{\partial u}, \frac{\partial s}{\partial v}, \frac{\partial t}{\partial u}, \frac{\partial t}{\partial v}) \qquad (2)$$

4 Detail Synthesis

Given a metric for assessing the sampling quality of image-based textures, we now return to the problem of extracting uniformly high quality textures from few images. 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, resulting in extracted textures 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 approximately bandpass at each level. Each level contains one octave of frequency data, with the highest level containing the residual data.

There is a direct relationship between the Jacobian values for a texture and the levels of the bandpass Laplacian pyramid. If the texture resolution is scaled so that the minimum value of the Jacobian metric over the surface is equal to 1 (i.e., the sampling is perfect), then the texture data can easily be assigned to the appropriate pyramid level. A texel with a metric (i.e., 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.

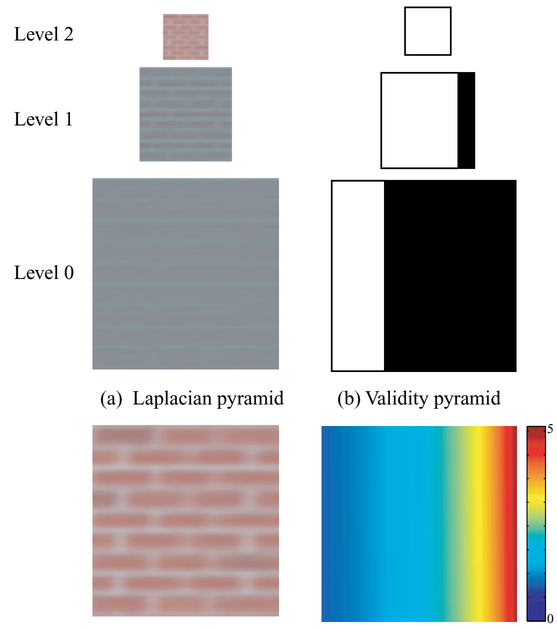
Another benefit of using Laplacian pyramids for detail synthesis lies in the method of reconstructing a final texture from a pyramid. Because the texture is recovered by repeatedly upsampling and merging bandpass levels, from the top down, data from higher levels 'show through' the lower levels. The 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.

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. 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.

In extending texture synthesis algorithms to image-based texturing we utilize the data structures illustrated in Figure 2 (taken from



(a) Laplacian pyramid (b) Validity pyramid

(c) Target texture (d) Jacobian metric over target

Figure 2: The detail synthesis data structures. The Laplacian pyramid in (a) corresponds to the texture in (c). The validity pyramid in (b), with valid texels indicated in white, is calculated from the target Jacobian in (d).

Figure 1, face B). A Laplacian pyramid (Figure 2 (a)) is constructed for the target texture (Figure 2 (c)). A pyramid containing validity information is created (Figure 2 (b)) using information in the Jacobian metric evaluated over the target surface. Our detail synthesis algorithm operates only in regions of the target Laplacian pyramid marked as invalid. Note that the Laplacian pyramid for the source is not shown in this figure.

The complete detail synthesis process using the Jacobian metric is as follows: determine the best scaling of the source texture, normalize the target texture with respect to the source texture scaling, construct a pyramid representation of the source texture, construct a pyramid representation of the target texture, and last, perform texture synthesis in regions of the target containing invalid data. We now discuss each of these steps in more detail.

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 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 involves computing a factor to scale the texture resolution. We begin by calculating the Jacobian matrix across the source texture assuming that $\mathbf{M}_{Tex}$ is the identity matrix, using the world space length of the surface's s and t directions as the initial texture resolution.

We can interpret the metric (i.e., the largest of any of the four Jacobian values) at any texel on 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 $1/k$ to the texture size values to get the new texture resolution, and scale $\mathbf{M}_{Tex}$ by k, then extract a source texture 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, so that 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. Thus, data could be lost when extracting the source texture from the image. This behavior is, however, desirable when one considers that we will use the source pyramid for synthesis; therefore, all levels of the pyramid must be full for proper synthesis to take place.

Normalizing the target texture. In order to to determine how the target texture quality relates to that of the source texture, we rescale the target $\mathbf{M}_{Tex}$ by the same factor calculated for the source. This is necessary in order to determine the data in the target pyramid, as well as to guarantee that the final result has the desired resolution.

After applying the scaling factor to the target texture, 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. Due to the rescaling, the values of the Jacobian metric for the target will generally be greater than 1, and the corresponding Laplacian pyramid will have missing data in its lower levels. It is this missing data that we will create through detail synthesis.

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 completely valid level of the target pyramid, as determined below, and adding the number of parent levels that the synthesis algorithm will consider. Creation of the pyramids then follows the standard algorithm for creating a Laplacian pyramid.

Using the analysis of Laplacian pyramids provided in Section 4.1, the validity of data at each level in the pyramid is easily determined. Recall that 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. We store this validity information in another pyramid structure to guarantee that data is only synthesized in invalid regions of the Laplacian pyramid.

Synthesis. The actual synthesis is based on the algorithm by Efros and Leung [1999]. Pixel neighborhoods are used instead of patches, but 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[2000] and 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

Detailed description of our image acquisition and model creation process can be found in [Ismert et al. 2003]. Briefly, all real world images were acquired at 16 bits/channel (linear) with a Canon D30 digital camera. Models were created using commercial image-based modeling software.

Rendered bricks. Our only synthetic scene, the brick cubes shown in Figure 1, was modeled and rendered using Discreet 3DSMax$^{\text{TM}}$. 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 $\mathbf{M}_{Tex}$ 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.

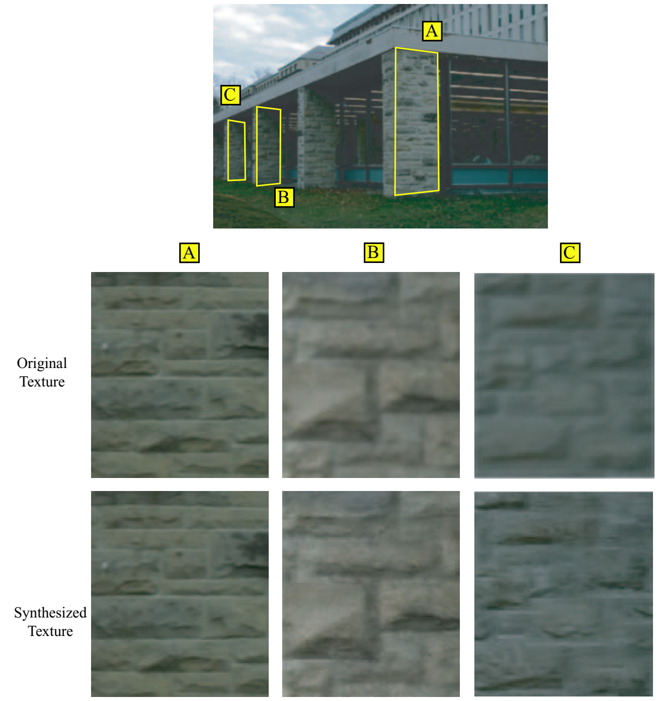


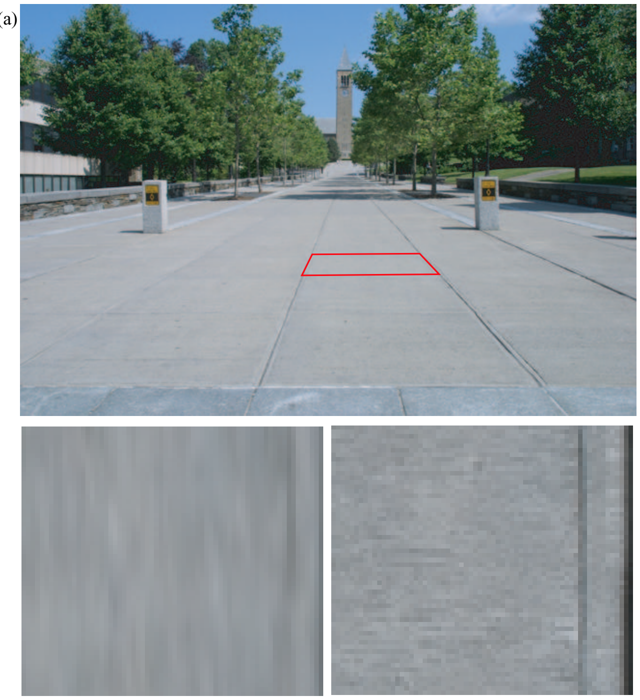
Figure 3: Results for columns along the face of a library. Face A was the source; textures for faces B and C were synthesized.

Synthesis results are shown in the bottom row of Figure 1, for faces B and C. 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 correct high frequency data 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.

Building columns. The input images for the results shown in Figure 3 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 $\mathbf{M}_{Tex}$ the maximum Jacobian value for the target texture was approximately 3.1, with only a small range across the surface. The synthesis algorithm used a pyramid 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, the results are 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.

Pavement. The input images for these results were also imaged with the Canon D30. The image shown in Figure 4 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



(a) (b) (c)

Figure 4: 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)

5x5 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. 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 few images. This improved texture quality is important for interactive walkthroughs where the user's viewpoint can differ from the original sampled camera locations.

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, 318–332.

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.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. ACM SIGGRAPH 1996*, 11–20.

DEBONET, J. S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. ACM SIGGRAPH 1997*, 361–368.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, 1033–1038.

FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. Example-based super-resolution. *IEEE Computer Graphics & Applications 22*, 2, 56–65.

HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proc. ACM SIGGRAPH 1995*, 229–238.

ISMERT, R., BALA, K., AND GREENBERG, D. 2003. Detail synthesis for imaghe-based texturing. Tech. Rep. PCG-03-001, Cornell University, Program of Computer Graphics.

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 *Pacific Graphics 2000*, 317–327.

LIU, X., YU, Y., AND SHUM, H.-Y. 2001. Synthesizing bidirectional texture functions for real-world surfaces. In *Proc. ACM SIGGRAPH 2001*, 97–106.

NEUGEBAUER, P. J., AND KLEIN, K. 1999. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum 18*, 3, 245–256.

OFEK, E., SHILAT, E., RAPPOPORT, A., AND WERMAN, M. 1997. Multiresolution textures from image sequences. *IEEE Computer Graphics & Applications 17*, 2, 18–29.

ROCCHINI, C., CIGNONI, P., AND MONTANI, C. 1999. Multiple textures stitching and blending on 3d objects. In *Eurographics Rendering Workshop 1999*.

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proc. ACM SIGGRAPH 2000*, 479–488.

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 *Proc. ACM SIGGRAPH 2000*, 287–296.

ZALESNY, A., AND GOOL, L. J. V. 2000. A compact model for viewpoint dependent texture synthesis. In *Proc. SMILE 2000*, 124–143.