

# Texture Mapping with a Jacobian-Based Spatially-Variant Filter

Ke Deng\* Jingdan Zhang\* Lifeng Wang† Baining Guo†

† Microsoft Research Asia  
Beijing, P.R.China

\*Tsinghua University  
Beijing, P.R.China

## Abstract

In this paper, we describe a new method to map a texture on a surface with a spatially-variant filter. Our filter takes into consideration the effects of anisotropy using a Jacobian approximation while computing the sampling rate, and the interpolation weights are computed with a sinc function. We also discuss how to do forward and backward mapping with the filter, and extend our algorithms to 3D meshes. Our experimental results verify our analysis.

**Keywords:** texture mapping, antialiasing, spatially-variant filtering

**Introduction** Texture mapping is an established rendering technique used to enhance the realism of 3D models. It has been extensively analyzed in the field of image processing and computer graphics [4, 7]. Texture aliasing may occur during this process, as mentioned by Wolberg [7]. Spatially-invariant kernels produce aliasing when warping images or textures with high frequencies. Using spatially-variant filters can help to reduce the aliasing effect with a variant sampling rate (e.g. MIPMAP [6] and EWA [2, 3, 1]). Recently Wang *et al.* [5] analyzed the issues of sampling rate, and used an affine-approximated filter for reconstruction from multiple views. In this paper we propose a new texture mapping method which gives better results than existing ones.

**Texture filter** Texture filtering involves the sampling rate and the kernel shape. Here we start from texture mapping with a homography. A mapping of a point  $(x', y')$  in the destination image to its correspondent point  $(x, y)$  in the source texture can be shown as:

$$\begin{cases} x = \frac{h_{11}x' + h_{12}y' + h_{13}}{h_{31}x' + h_{32}y' + h_{33}} \\ y = \frac{h_{21}x' + h_{22}y' + h_{23}}{h_{31}x' + h_{32}y' + h_{33}} \end{cases} \quad (1)$$

\*The work was done when Ke Deng and Jingdan Zhang were visiting at Microsoft Research Asia.

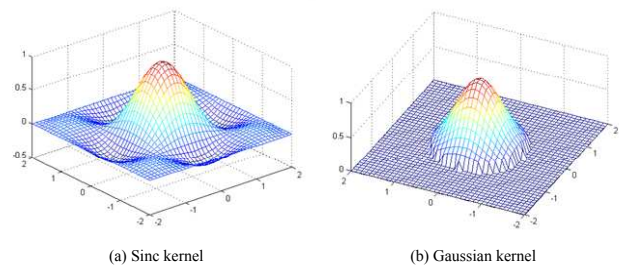


Figure 1. Comparison of *sinc* kernel with a Gaussian kernel in the 2D case.

The sampling rate is illustrated by the Jacobian of the homography:

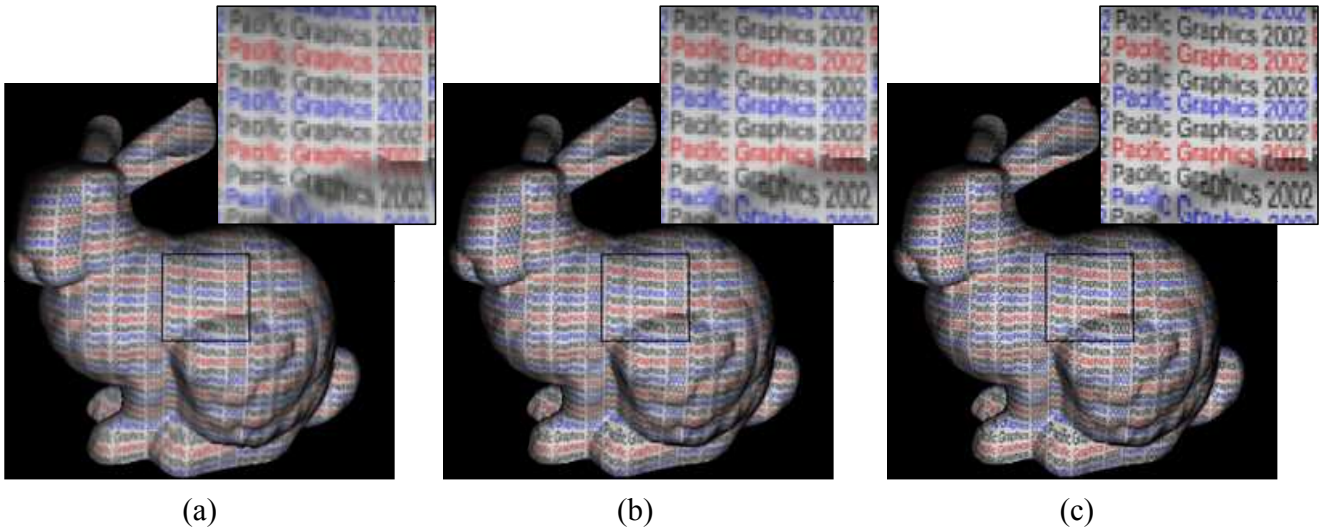
$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} dx' \\ dy' \end{bmatrix}, \quad (2)$$

From the analysis in [5], we can find that a *sinc* kernel is an ideal sampling kernel to avoid frequency overlapping in the sampling process. With the Jacobian-based sampling rate and the *sinc* kernel, our filter is taken as:

$$\begin{aligned} r(x, y) &= \text{sinc}((J_{11} + J_{21})x, (J_{12} + J_{22})y) \otimes \text{sinc}(x, y) \\ &= \text{sinc}(x \cdot \max(J_{11} + J_{21}, 1), y \cdot \max(J_{12} + J_{22}, 1)) \end{aligned} \quad (3)$$

where  $\otimes \text{sinc}(x, y)$  means a necessary prefilter to eliminate frequency that is higher than destination resolution.

**Comparison with EWA** EWA is a well-known spatially-variant filter. Figure 1 shows the difference between a *sinc* kernel and a Gaussian kernel in 2D case. Our kernel in 2D is also a *sinc* kernel and takes the form of  $\text{sinc}(x, y) = \text{sinc}(x)\text{sinc}(y)$ , and the Gaussian kernel takes the form of  $f(x, y) = e^{-2(x^2+y^2)}$ . It can be observed that a Gaussian will not be a good approximation of a *sinc* kernel when extended beyond the area  $\delta(-1..1, -1..1)$  in 2D, which makes the result more blurred than ours in figure 2.



**Figure 2. Texture mapping on a 3D bunny with shading.**

(a) The mapping result using OpenGL with MIPMAP on. (b) The mapping result using EWA. (c) The mapping result using our method.

**Forward mapping and backward mapping** There are two approaches for mapping a texture to a surface: backward mapping and forward mapping. When doing backward mapping, we compute each pixel in the destination image by resampling the source texture. Destination pixels in scan-line order are inversely mapped and sampled on the source texture. We can also splat source pixels to destination [8], which is called forward mapping [1]. We apply our filter to both approaches, and find that backward mapping gives better result than forward mapping in that forward mapping tends to create artifacts at the boundary because of the weight leakage at boundaries.

**Experimental result** Figure 2 is an example of mapping a “Pacific Graphics 2002” texture to a complicated 3D bunny mesh. We apply shading after rendering each pixel on screen in a manner similar to [8]. (a) is the mapping result using a linear interpolation kernel by OpenGL with MIPMAP on, and it looks blurred. (b) is the mapping result using EWA. (c) is the result using our method.

**Discussion and future work** Our filter is based on a Jacobian sampling rate and a *sinc* kernel function, which gives better results than traditional filters. But our method consumes more computation time, because the filter is spatially-variant for each pixel. Accelerating is necessary for extending our method to real-time rendering. Our analysis is useful not only for texture mapping on surfaces, but also for applications such as analysis by synthesis, point-based rendering, super-resolution techniques, and VDTM in image-based rendering.

## References

- [1] B. Chen, F. Dacheille, and A. Kaufman. Forward image warping. *IEEE Visualization '99*, pages 89–96, 1999.
- [2] N. Greene and P. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 6(6):21–27, June 1986.
- [3] P. Heckbert. Fundamentals of texture mapping and image warping. *Master's thesis, CS Division, UCB/CSD 89/516, U.C. Berkeley*, 1989.
- [4] P. S. Heckbert. Survey of texture mapping. *Proceedings of Graphics Interface '86*, pages 207–212, 1986.
- [5] L. Wang, S. Kang, H. Shum, and R. Szeliski. Optimal texture map reconstruction from multiple views. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2001)*, 2001.
- [6] L. Williams. Pyramidal parametrics. *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, pages 1–11, 1983.
- [7] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, California, 1990.
- [8] M. Zwicker, H. Pfister, J. Baar, and M. Gross. Surface splatting. *Computer Graphics proceeding, Annual Conf. Series, ACM SIGGraph*, pages 371–378, 2001.