Spectral Texturing for Real-Time Applications

Daniel R. Berger*
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

1 Introduction

In this sketch we present a new method for rendering large-scale, high-resolution, non-repetitive textures in real-time using multi layer texturing. The basic idea of spectral texturing is to construct the final texture by multiple texture layers where each layer provides a certain range of the spectrum of the texture's spatial frequencies. Alpha channels are used to introduce statistical dependencies between the frequency bands.

This approach extends a method called detail texturing, which does not use alpha channels to model higher statistical properties of the resulting texture. Other approaches to generate textures with specified statistical properties, like [DeBonet 1997], are not suitable for real-time use and would require storage of the generated texture.

Spectral texturing is very easy to implement, runs on all contemporary 3d graphics cards, and is especially suitable for naturalistic textures in real-time applications which are viewed from a large range of distances.

2 Method

Figure 1 shows a groundplane with a five-layer spectral texture (top). The texture of all five layers is the same low-pass filtered random noise texture tile of 512*512 pixels (bottom left), which tiles seamlessly. The texture contains a textured transparency channel (bottom right), which has in this case been generated by applying an emboss filter to the color texture.

The layer with the smallest tiles is rendered first. On top of this the next-larger scaled layer is rendered using alpha blending, and so on. The transparency map of each layer defines where the previously rendered layers (containing higher frequency components of the texture) will be visible. It can, for example, make the amplitude of higher-frequency content depend on the phase of the texture layer's color map, as in Figure 1.

In this example there are 64x64 tiles per km^2 at the smallest scale; at the largest scale the texture tile is stretched over 10x10 km. This results in a virtual texture size of 327680^2 pixels (a texture of 10x10 km at a resolution of approximately 3x3 cm per pixel). In texture memory, such a texture would take up 300 GB. As a spectral texture, it uses just 1 MB, but rendering needs five textured layers.

Both mip-mapping and bilinear filtering are used to assure smoothness of the texture at all magnification levels. The textures themselves are low-pass filtered to such a degree that they do not produce artifacts when magnified. As the different layers of the spectral texture are magnified differently at the same distance, they also go into mip-mapping and bilinear interpolation at different distances. This results in a texture which has fine detail at a large range of distances without producing aliasing artifacts.

The examples shown were all generated by manually adjusting the texture parameters. Currently we are working on a method to analyze a given image for its statistical properties, and to use these properties to generate texture tiles and mixing parameters for a similar-looking spectral texture. For the analysis a method similar to the one shown in [DeBonet 1997] could be used.

*e-mail: daniel.berger@tuebingen.mpg.de

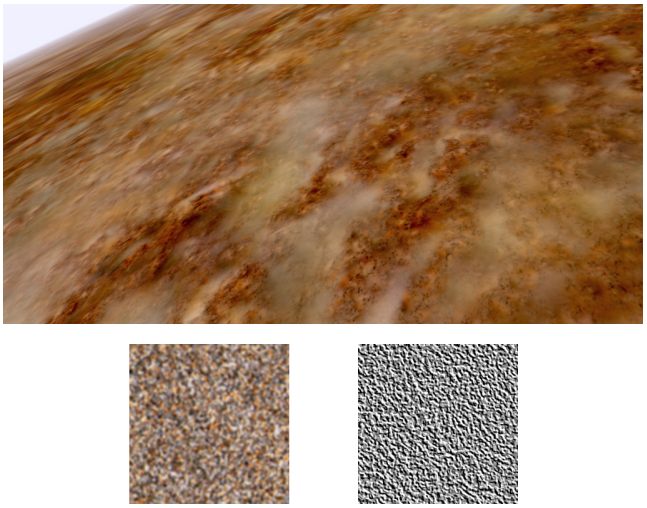

Figure 1: A five-layer spectral texture (top), and color and alpha channels used for this texture (bottom)

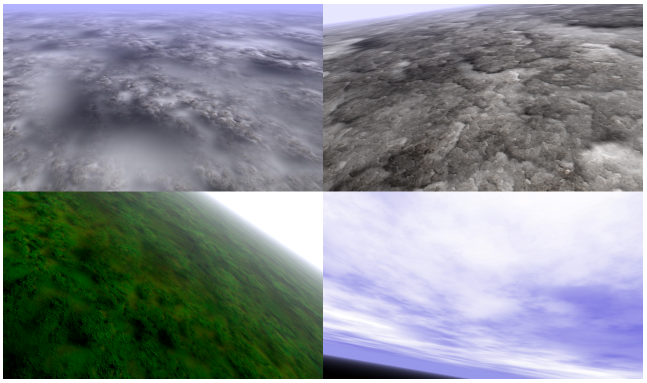

Figure 2: More examples of real-time spectral textures

3 Conclusion

Spectral texturing can be used to simulate different sorts of clouds, gravel, stone, concrete, mud, vegetation, and other naturalistic textures with random and fractal characteristics in real-time. Therefore this approach should be especially advantageous for 3d computer games and VR environments.
See also: *http://www.tuebingen.mpg.de/∼berger/spectraltexturing/*

References

DEBONET, J. S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of SIGGRAPH 1997*, ACM Press / ACM SIGGRAPH, T. Whitted, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 361–368.