

Example-based Texture Synthesis on Disney’s *Tangled*

Christian Eisenacher¹

Chuck Tappan²

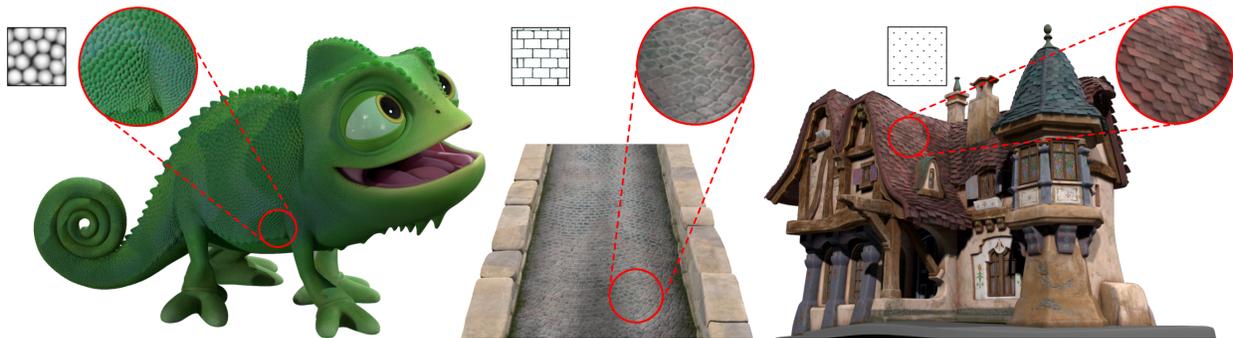
Brent Burley²

Daniel Teece²

Arthur Shek²

¹University of Erlangen-Nuremberg

²Walt Disney Animation Studios



© 2010 Disney Enterprises

Figure 1: Left, displacement for scales, synthesized onto a chameleon with varying size. Middle, displacement for stones on a road, synthesis oriented by a vector flow field. Right, a texture to specify instancing sites for tile geometry, synthesized to arrange tiles organically on a roof.

1 Overview

Look development on Walt Disney’s animated feature *Tangled* called for artists to paint hundreds of organic elements with high-resolution textures on a tight schedule. With our Ptex format [Burley and Lacewell 2008], we had the infrastructure to handle massive textures within our pipeline, but the task of manually painting the patterned textures would still involve tedious effort.

We found that example-based texture synthesis, where an artist paints a small *exemplar* texture indicating a desired pattern that the system synthesizes over arbitrary surfaces, would alleviate some of the burden. In this talk we describe how we adapted existing synthesis methods to our Ptex-based workflow, scaled them to production-sized textures, and addressed other engineering challenges.

2 Synthesis Details

Our typical models use up to one giga-texel per surface, and complex models easily exceed that. However, few algorithms are even capable of synthesizing more than one mega-texel on a continuous surface. Our system is based on neighborhood matching, inspired by Lefebvre and Hoppe [2006], but addresses a number of challenges, including support for the texture sizes we require. We chose their method because it is inherently parallel, trades pre-processing for very fast synthesis, and has shown excellent synthesis quality over a wide range of inputs.

Exemplar Analysis: Like Lefebvre and Hoppe, we pre-process the exemplar to speed up runtime synthesis. We create a set of box-filtered images, extract neighborhoods, perform dimensionality reduction using PCA, and prune the search space using k -coherence. However, instead of determining the k most similar candidates for each neighborhood using a kd-tree, we pseudo-randomly sample the exemplar several times and keep the most similar neighborhood of 64, 16, and 4 samples. That way, we obtain $k=3$ sufficiently similar candidates with increasing randomness. This is considerably faster and allows significantly larger exemplars: up to 4096x4096 compared to the typical 64x64 or 128x128. The added variation turns out to be beneficial for synthesis quality with the hand painted exemplars we use. Our analysis is multi-threaded and a 1024x1024 exemplar is analyzed in about one minute on an 8-core machine.

Synthesis: We cannot fit all temporary data required to synthesize such large textures into main memory. Therefore, we directly operate on Ptex data and perform synthesis out-of-core, taking advan-

tage of Ptex’s per-face granularity. We create the texture hierarchically from coarse to fine. For each level, we request each face and its adjacent faces from disk, perform a correction pass with proper filtering, and send the result back to disk. Synthesizing 600 million color texels requires 14 GB of temporary data and 30 min on an 8-core machine. We are I/O bandwidth bound, as the complete texture needs to be streamed to/from disk for each iteration, but manage an average 7x multithreaded speedup with a two disk RAID 0.

3 Workflow Integration

Texture synthesis integrates nicely into our proprietary digital painting tool. Exposing it there complements existing methods for procedural texture generation via a custom shader expression language. Artists can conveniently browse from a shared library or paint an exemplar, and synthesize a texture across a surface with one click. The texture is immediately available for manual correction of problem areas.

Our system allows artists to have fine control over varying feature size by either painting or procedurally generating a scale map, as done for the chameleon body in Fig. 1. Artists can also specify the orientation of the synthesized texture locally, using an interactively brushed vector flow field. It is represented as an oriented point cloud for easy interaction with external generators. Note the subtle interweaved curve-pattern of the stones on the road in Fig. 1.

4 Conclusion and Future Work

We have had great success synthesizing many organic and semi-regular textures for *Tangled*. Artists enjoy the increased texture painting efficiency and high level of art-directability. Next, we would like to explore ways to improve synthesis quality for very regular patterns like stone courses on irregular geometry, better integrate coarse and fine detail when using higher-res exemplars, and provide more intuitive tools for generating effective exemplars.

References

- BURLEY, B., AND LACEWELL, D. 2008. Ptex: Per-face texture mapping for production rendering. *Computer Graphics Forum* 27, 4 (June), 1155–1164.
- LEFEBVRE, S., AND HOPPE, H. 2006. Appearance-space texture synthesis. *ACM Transactions on Graphics* 25, 3, pp. 541–548.