

Texture Synthesis over Arbitrary Manifold Surfaces

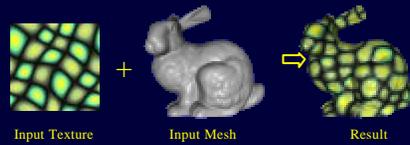
Li-Yi Wei Marc Levoy



Computer Graphics Group
Stanford University

Introduction

Synthesize a surface texture by coloring mesh vertices



Desirable Properties

- Share advantages of 2D algorithm [SIGGRAPH 2000]
 - Quality
 - Efficient
 - General
 - Easy to use
- Minimum distortion
- Minimum discontinuity



Previous Work

- Texture Mapping
 - Global [Catmull'74, Bier'86, Levy'98]
 - Local
 - Triangle tiles [Neyret'99]
 - Lapped textures [Praun'00]

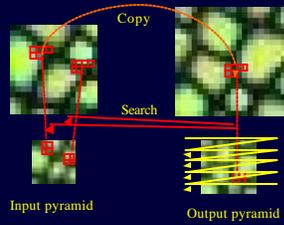
Previous Work

- 3D Texture synthesis
 - procedural [Perlin'85, Turk'91, Witkin'91]
 - from example
 - surface texture [Gagalowicz'86, Turk'01, Ying'01]
 - volume texture [Heeger'95, Ghazanfarpour'96]

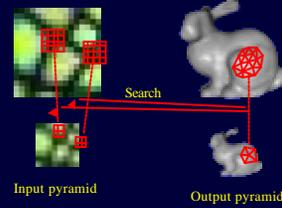
Previous Work

- 2D Texture from Example
 - pyramid matching [Heeger'95, Simoncelli'98]
 - block shuffling [De Bonet'97, Xu'00]
 - Markov Random Field [Popat'93]
 - neighborhood search [Efros'99, Wei'00]

Texture Synthesis by Neighborhood Search



Surface Texture Synthesis by Neighborhood Search



Differences between Images and Meshes

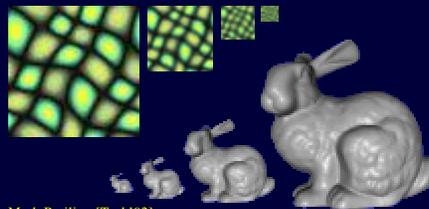


Process

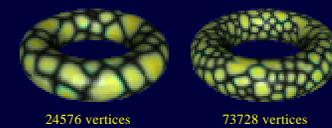
1. Build image/mesh pyramids
2. Assign texture orientation
3. Generate texture



Image & Mesh Pyramids



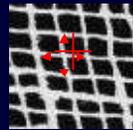
Retiling Density



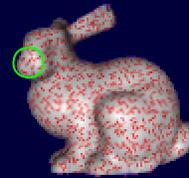
Texture Orientation

- Methods for orienting textures
 - user-specified
 - random (for isotropic textures)
 - smooth or symmetric (for anisotropic textures)
 - by relaxation

Texture Orientation



4-way symmetric texture

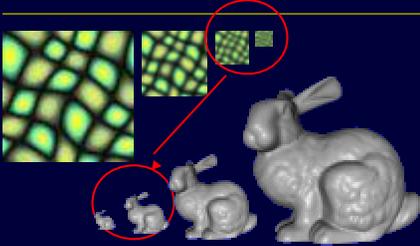


4-way symmetric vector field

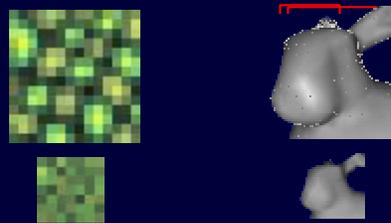
Texture Orientation by Relaxation

- Minimize an error function
 - similar to [Hertzmann'00, Pedersen'95]

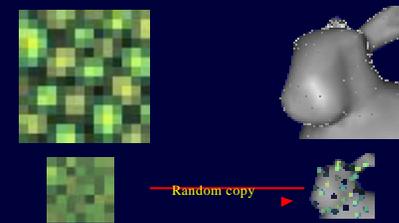
Synthesis

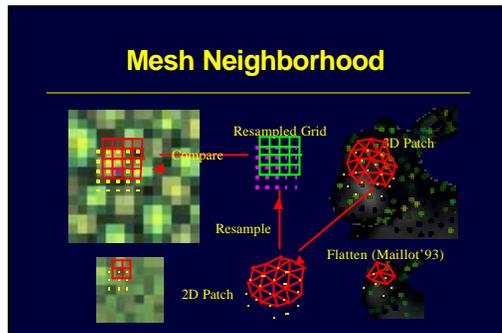
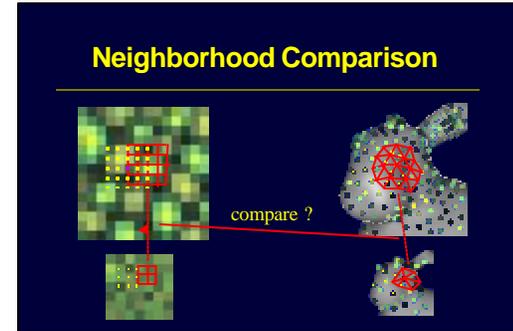
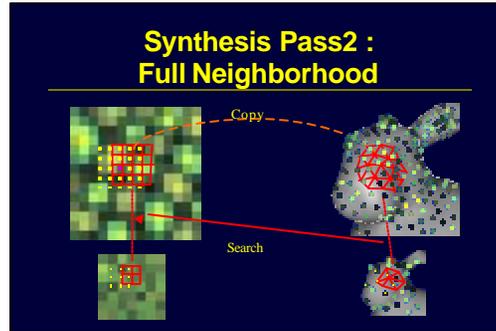
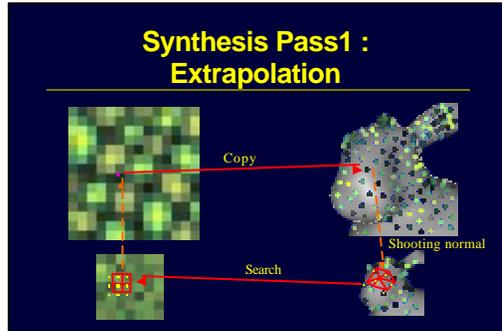


Synthesis : 2 Lowest Levels



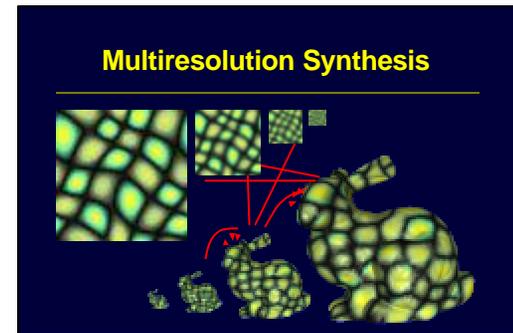
Synthesis : Lowest Level

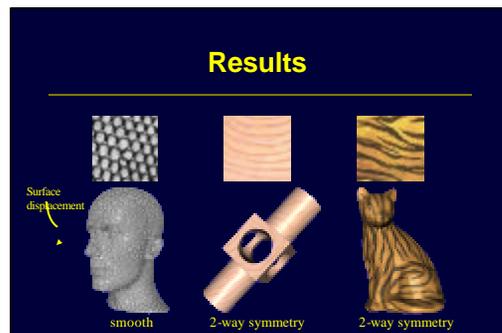
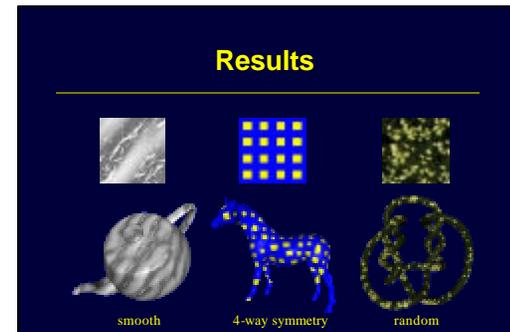
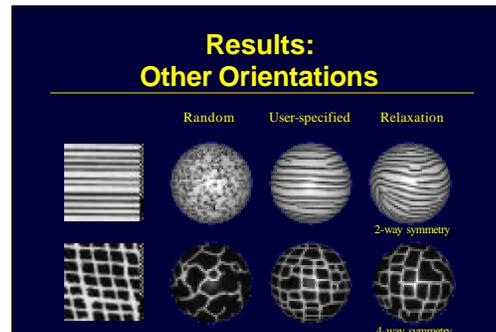
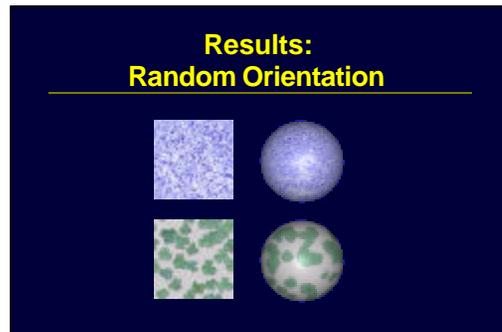




Accelerating Neighborhood Search

- Tree-structured Vector Quantization [Wei'00]





Summary of Differences

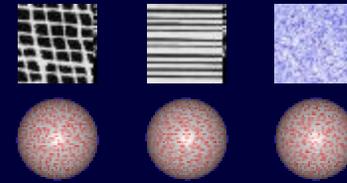
	Turk's approach	Our approach
Vector field	smooth	random, symmetric
Traversal order	sweeping	random
Neighborhood	surface marching	flattening/resampling
Mesh hierarchy	explicit parent/child	shooting normal

- ### Future Work
- Texture transfer between models
 - Texturing animated models
 - Mesh signal processing

More Information

<http://graphics.stanford.edu/projects/texture/>

Texture Orientation Examples

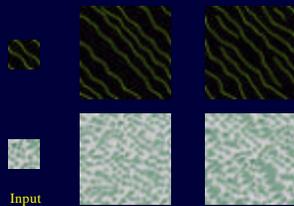


4-way symmetric

2-way symmetric

Random

Synthesis Order



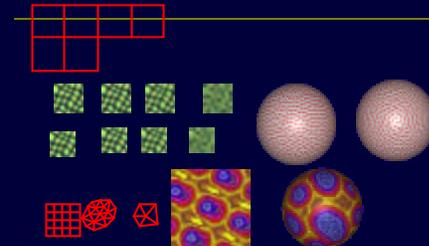
Input

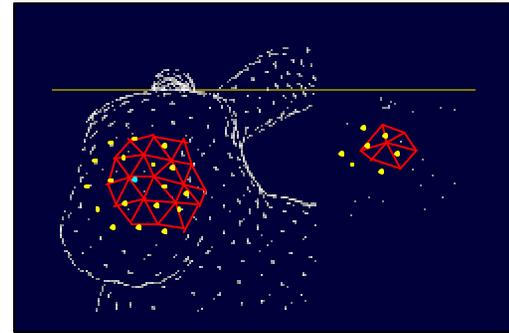
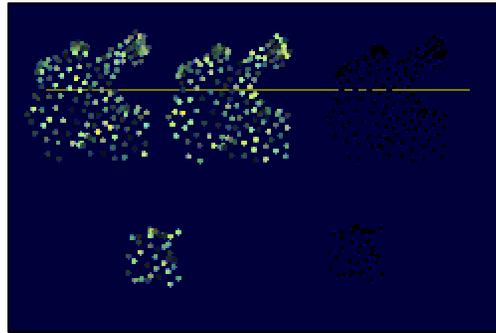
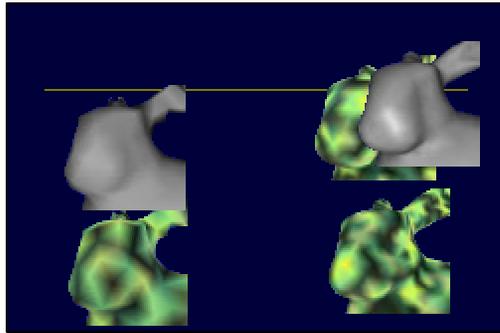
Scanline

Random

Differences between Turk'01 and Our Approach

	Turk's approach	Our approach
Vector field	smooth	random, symmetric
Traversal order	sweeping	random
Neighborhood	surface marching	flattening + resampling
Mesh hierarchy	explicit parent/child	shooting normal





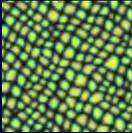
Introduction



Input

Texture Synthesis





Result

Introduction

Synthesize a surface texture by coloring mesh vertices



Input Texture

+



Input Mesh

Synthesis





Result

Previous Work

<p>Synthesis from Example Texture mapping</p>	<p>Procedural texturing Lapped textures</p>
<p>General Applicable to surfaces Efficient Low distortion No discontinuity</p>	<div style="border: 1px solid white; width: 100%; height: 100%; position: relative;"> <div style="position: absolute; top: 10px; left: 10px; color: red; font-size: 2em;">✗</div> <div style="position: absolute; bottom: 10px; left: 10px; color: green; font-size: 2em;">✓</div> </div>

Image Texture Synthesis

- Exhaustively search neighborhoods



Image Texture Synthesis

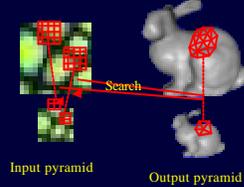


Mesh Pyramid



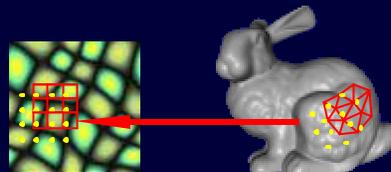
Key Idea

- Reuse the neighborhood-search framework

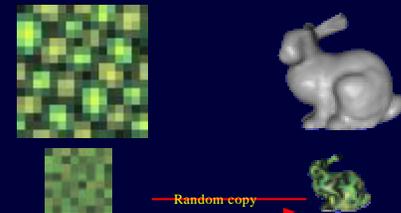


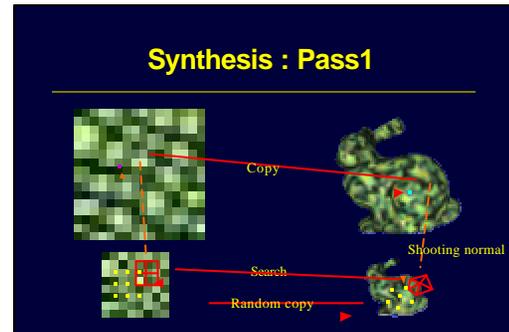
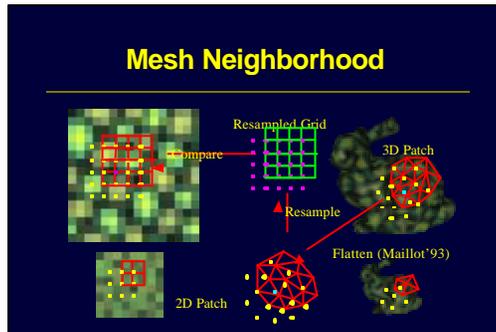
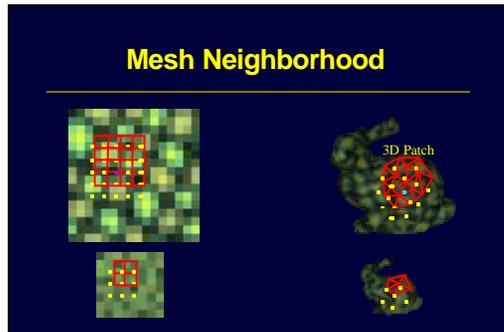
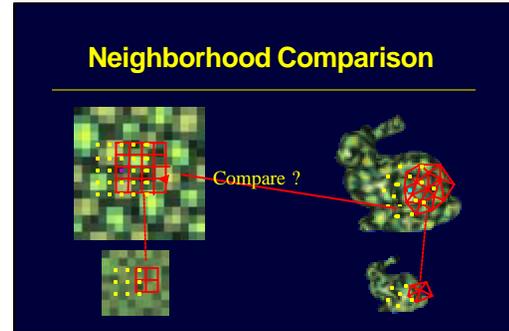
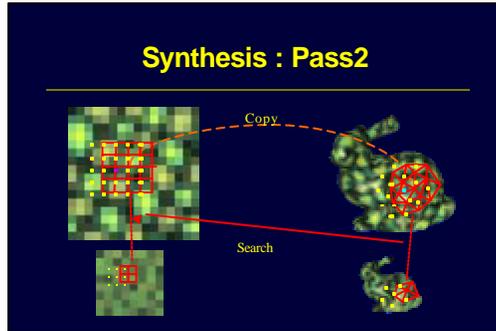
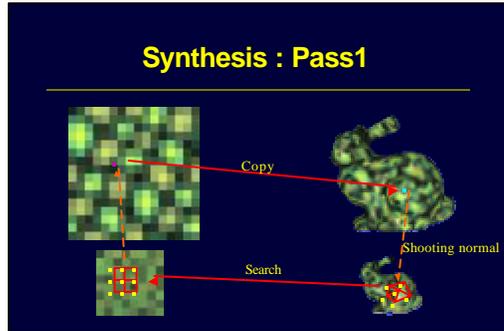
Key Idea

- Reuse the neighborhood-matching framework



Synthesis Lowest Level





You accidentally hit the END button!