Symmetry-Guided Texture Synthesis and Manipulation

Vladimir G. Kim  Yaron Lipman  Thomas Funkhouser
Princeton University  Weizmann Institute  Princeton University
Motivation
Motivation

- Fine-scale details
- Large-scale pattern
Motivation

Fine-scale details

Large-scale pattern
Motivation

- Fine-scale details
- Large-scale pattern
Goal

Control large-scale patterns while preserving fine-scale details

- Synthesize textures with desirable patterns
- Manipulate patterns in textures and images
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
- Near-Regular Textures
- Symmetry Detection
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
  - Near-Regular Textures
  - Symmetry Detection

Efros and Lleung ‘99
Efros and Freeman ‘01,
Kwattra et al. ‘03,
Kwattra et al. ‘05,
Lefebvre et al. ‘10
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
  - Near-Regular Textures
  - Symmetry Detection

Additional Objectives

Texture Transfer

Interactive Merging

Efros and Lleung ‘99
Efros and Freeman ’01,
Kwatra et al. ‘03,
Kwatra et al. ’05,
Lefebvre et al. ‘10
Related Work

Analyzing, manipulating, and synthesizing textures

- **Texture Synthesis**
- **Near-Regular Textures**
- **Symmetry Detection**

Near-Regular Texture Analysis and Manipulation. *Liu et al. ‘04*

Image De-Fencing. *Liu et al. ‘08, Park et al. ‘10*
Related Work

Analyzing, manipulating, and synthesizing textures
- Texture Synthesis
- Near-Regular Textures
- Symmetry Detection

Symmetry Detection from Real World Images. *Ingmar Rauschert, Kyle Brocklehurst, Somesh Kashyap, Jingchen Liu, Yanxi Liu, ’12*

Partial and approximate symmetry detection…
*Mitra et al. ‘06*
Discovering Structural Regularity…
*Pauly et al. ‘08*
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
- Near-Regular Textures
  - Symmetry Detection

Lattice detection is hard…

Deformed Lattice Detection… Park et al. ’09,
Discovering Texture Regularity… Hays et al. ’06
Related Work

Analyzing, manipulating, and synthesizing textures

- Texture Synthesis
- Near-Regular Textures
  - Symmetry Detection

Lattice detection is hard…
especially if there is none!
Our Approach

Approximate Symmetry Representation:

Approximate Symmetry Representation:
Our Approach

Approximate Symmetry Representation:

Identity

Approximate Symmetry Representation:

Identity
Our Approach

Approximate Symmetry Representation:

Approximate Symmetry Representation:
Our Approach

Approximate Symmetry Representation:

Approximate Symmetry Representation:
Our Approach

Approximate Symmetry Representation:

Approximate Symmetry Representation:
Our Approach

Approximate Symmetry Representation:

“Reflective symmetry descriptor”, Kazhdan et al. 2003,
“Planar reflective symmetry transform”, Podolak et al. 2006, ...
Typical Texture Synthesis Pipeline

Image Space

Input

Texture Perturbation Model

Optimizer

Objective Function

Target
Our Framework Overview

Image Space

- Input
- Target
- Texture Perturbation Model
- Compute Symmetry Representation

Symmetry Space

- Optimizer
- Objective Function
Pattern Processing

Image Space

Texture Perturbation Model

Compute Symmetry Representation

Optimizer

Objective Function

Filter

Symmetry Space

Pattern Processing
Pattern Processing: Goal

Filter the symmetry representation: e.g.

○ Blur = make less symmetric
○ Sharpen = make more symmetric
○ Scale = scale pattern (same scale of texture details)
○ ...
Pattern Processing: Scale

Input

Image

Symmetry Representation
Pattern Processing: Scale

Input

Image

Symmetry Representation

Filter

Scale x2
Pattern Processing: Scale

Input

Image

Symmetry Representation

Filter

Scale x2

Result
Pattern Processing: Scale

Input

Filter

Result

Image

Symmetry

Representation

Scale x2

Result
Pattern Processing: Reduce Contrast

Input
Image

Filter
Symmetry Representation
Contrast +4

Result
Pattern Processing: Increase Contrast

Input

Image

Symmetry Representation

Filter

Contrast -4

Result
Pattern Processing: Identity

Input

Image

Filter

Symmetry Representation

No filter

Result
Pattern Processing: Identity

**Input**

- Image

**Not Symmetry-guided**

- Result

**Symmetry Representation**

- Image
Optimization Example

Represent large-scale patterns with approximate symmetry representations
Optimization Example

Represent large-scale patterns with approximate symmetry representations
Optimization Example

Represent large-scale patterns with approximate symmetry representations

Image Space

Symmetry Space

Autocorrelation

Filter

Scale x2
Optimization Example

Represent large-scale patterns with approximate symmetry representations

$\theta$: Patch-Based Synthesized Textures

Autocorrelation

Symmetry Space

Filter

Scale x2
Represent large-scale patterns with approximate symmetry representations

Optimization Example

Random Patches

Iteration 0

Image Space

Symmetry Space

θ: Patch-Based Synthesized Textures

Autocorrelation

Scale x2
Represent large-scale patterns with approximate symmetry representations

Optimization Example

- Perturb Patches to minimize $L_2$ distance
- $\theta$: Patch-Based Synthesized Textures
- Image Space
- Symmetry Space
- Autocorrelation

Iteration 6

Scale x2
Optimization Example

Represent large-scale patterns with approximate symmetry representations

Iteration 11

Perturb Patches to minimize $L_2$ distance

Image Space

Symmetry Space

$\theta$: Patch-Based Synthesized Textures

Autocorrelation

Scale x2
Optimization Example

Represent large-scale patterns with approximate symmetry representations

Iteration 25

Perturb Patches to minimize $L_2$ distance

$\theta$: Patch-Based Synthesized Textures

Image Space

Symmetry Space

Autocorrelation

Scale x2
Pattern Processing: Warping Texture

As-rigid-as-possible deformation

Input

Texture Perturbation Model

Optimizer

Compute Symmetry Representation

Objective Function

Filter

Image Space

Symmetry Space
Pattern Processing: Sharpen

Input

Filter

Symmetry Representation

Sharpen
Pattern Processing: Sharpen

Input

Filter

Texture Perturbation

Symmetry Representation

Sharpen
Pattern Processing: Sharpen

Input

Filter

Texture Perturbation

Symmetry Representation

Sharpen

Iteration 2
Pattern Processing: Sharpen

Input

Filter

Texture Perturbation

Symmetry Representation

Sharpen

Iteration 5
Pattern Processing: Sharpen

Input  |  Filter  |  Result

Symmetry Representation  |  Sharpen  |  

Result
Pattern Processing: Blur G(2)
Pattern Processing: Blur G(4)

Input

Filter

Result

Symmetry Representation

Blur G(4)
Pattern Processing: Sharpen / Blur

Input

Sharpen
Blur G(2)
Blur G(4)
Applications

Pattern Processing

Pattern Transfer

Pattern Optimization
Pattern Transfer: Goal

Transfer pattern of texture $g$ to texture $f$

Texture: $f$

Pattern: $g$

Result: $f'$
Pattern Transfer

Input

Texture Perturbation Model

Optimizer

Compute Symmetry Representation

Objective Function

Target

Image Space

Symmetry Space
Pattern Transfer From a Sketch

Target Pattern

Source Texture:

Result

© shallowend24401

Result

© euart
Pattern Transfer From a Sketch

Target Pattern

Source Texture:

Result
© shallowend24401

Result
© euart
Pattern Transfer From a Sketch

Source Texture:

Target Pattern

© shallowend24401

© euart

Result

Result
Pattern Transfer From an Image

Source Texture:

Target Pattern

Result

© shallowend24401

© 100kr

Result
Pattern Transfer From an Image

Target Pattern

Source Texture: © shallowend24401

Result

Texture:
Symmetry

Result © 100kr
Pattern Transfer From an Image

Target Pattern

Source Texture:

© shallowend24401

© 100kr

Result

Result
Applications

Pattern Processing

Pattern Transfer

Pattern Optimization
Pattern Optimization: Goal

Optimize function defined over symmetry representation, e.g.

- Symmetrize
Pattern Optimization

Maximize standard Deviation of symmetry representation

Input

Texture Perturbation Model

Compute Symmetry Representation

Optimizer

Objective Function
Pattern Optimization: Tileable Texture

Blend edges + tile
Pattern Optimization: Tileable Texture

Blend edges + tile → Symmetrize
Pattern Optimization: Tileable Texture

Symmetrize

Blend edges + tile

Blend edges + tile
Pattern Optimization: Tileable Texture

Blend edges + tile

Symmetrize

Blend edges + tile
Pattern Optimization: Tileable Texture

© David Brennan
Pattern Optimization

Image Space

Input

Texture Perturbation Model

Symmetry Space

Optimizer

Compute Symmetry Representation

Objective Function

Rotational symmetry

Reflective symmetry
Pattern Optimization: Rotational

Input Image
Pattern Optimization: Rotational

Symmetrized

© Martin Heigan
Pattern Optimization: Reflectional

Input Image

© John Perivolaris
Most of the time is spent on iterative computation of a symmetry transform.

Total Time ≈ 1hr

Total Time ≈ 30min
Summary

General framework for operating on textures in symmetry space

Investigated applications: pattern transfer, pattern filtering and pattern optimization
Limitations

Must be gap in size between fine-scale details and large-scale pattern
Future work

Other applications

- Symmetry-guided manipulation of 3D geometries
Acknowledgements + Project Website

Code and Data:
- Flickr: shallowend, dolescum, C. Lewis (cloois), 100kr, snappa2006, cwazymandy, D. Brennan (davidbrennan), J. Perivolaris (dr_john2005), M. Heigan (martin_heigan)
- CMU NRT database
- Code (for comparison): J. Hays, M. Park, Y. Liu

Funding:
- Google, Intel, Adobe, NSERC, NSF

Project Website (code, data, and examples):

http://www.cs.princeton.edu/~vk/SymmetryTexture

Thank You!