Oral Presentation Multiscale Texture Synthesis

Theofanis Petkos, Rovira i Virgili University

Paper: Multiscale Texture Synthesis

Authors:

- Charles Han
- Eric Risser
- Ravi Ramamoorthi
- Eitan Grinspun

University of Columbia, 2008.

Source URL: https://www.cs.columbia.edu/cg/mts/index.html

Overview

- Example-based Texture Synthesis vs Multiscale Texture Synthesis

- Proposed Multiscale Texture Synthesis:
    - Infinite Zoom
    - Inconsistency Corrections
    - Optimized Performance

Example-Based Texture Synthesis

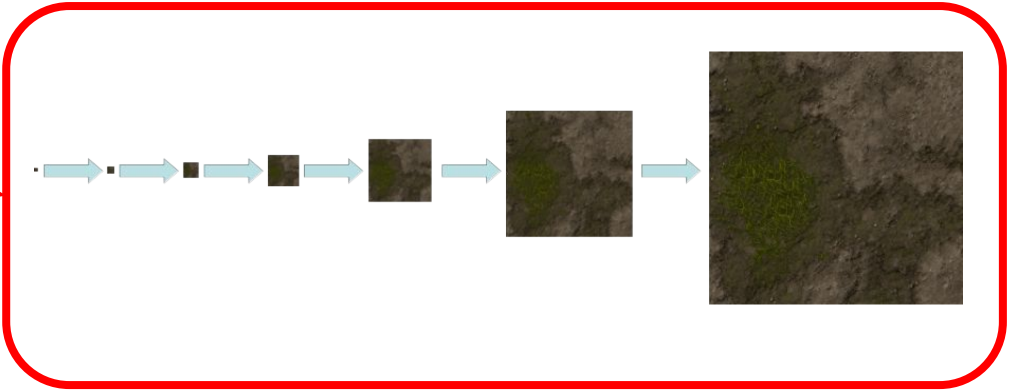
- Takes as input a single exemplar image.
- Specific limitations, usually resolution-based.
- Hierarchical Approach

Example-Based Texture Synthesis

- Takes as input a single exemplar image.

- Specific limitations, usually resolution-based.

- Hierarchical Approach



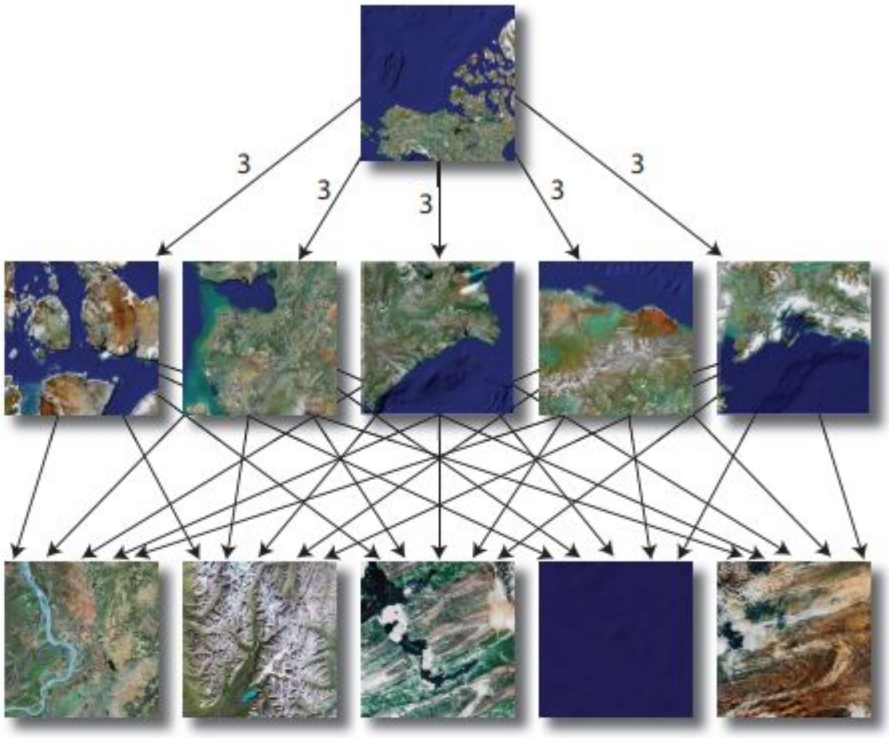
Multiscale Texture Synthesis

- Exemplar Graph instead of single exemplar input

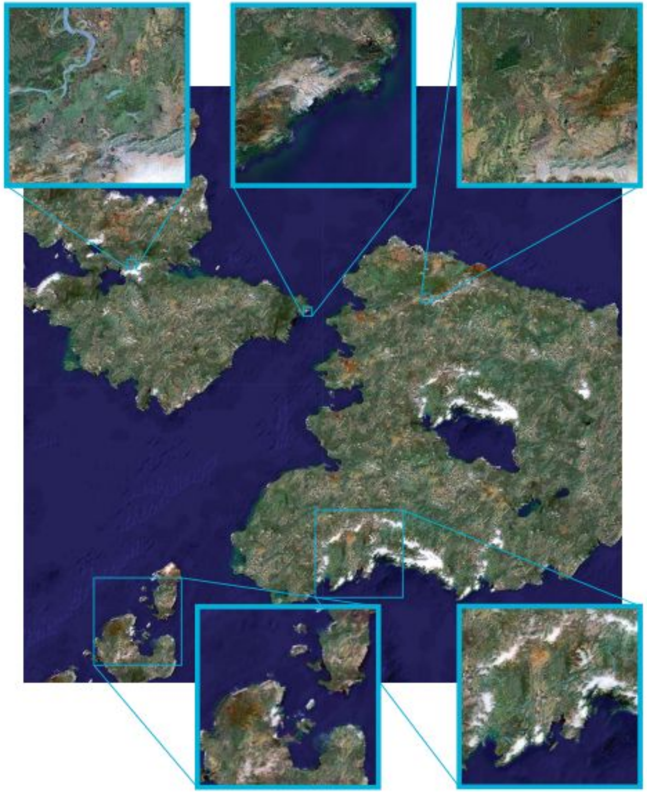
- Multiple samples are connected to create the synthesized result.

- Support for single exemplar graph by looping over its contents.

Multiscale Texture Synthesis



Multiscale Texture Synthesis



Multiscale Texture Synthesis: Key Concepts

- Exemplar Graph

- Gaussian Stack

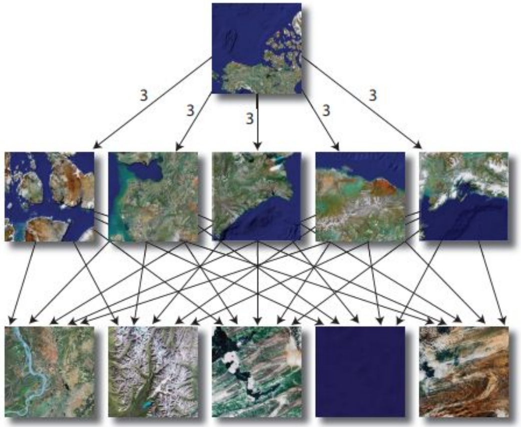
- Admissible Candidates

Multiscale Texture Synthesis: Key Concepts

- **Exemplar Graph**

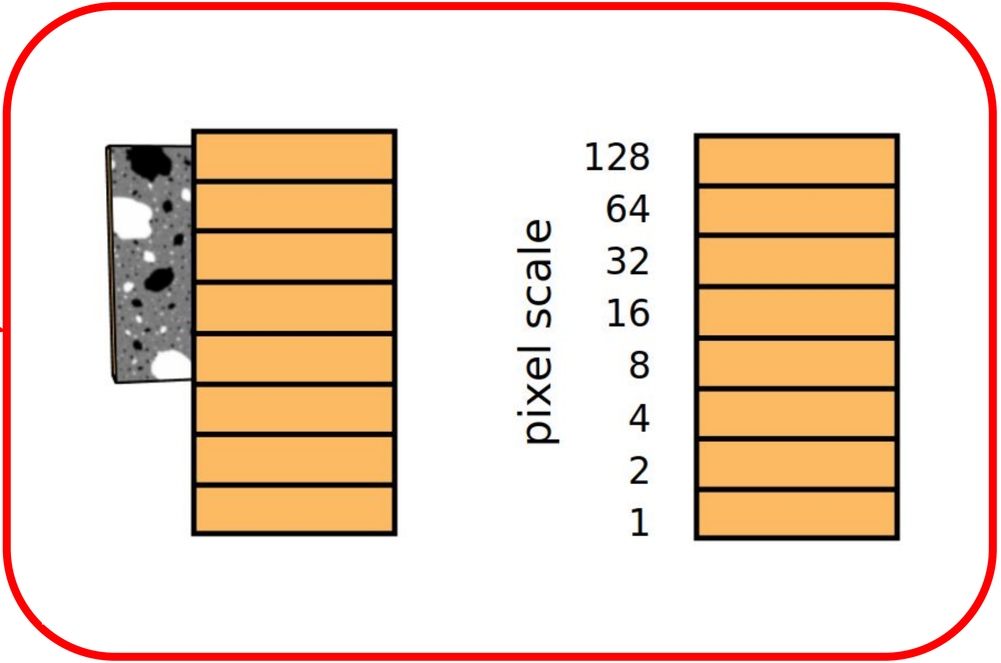
- Gaussian Stack

- Admissible Candidates



Multiscale Texture Synthesis: Key Concepts

- Exemplar Graph
- **Gaussian Stack**
- Admissible Candidates

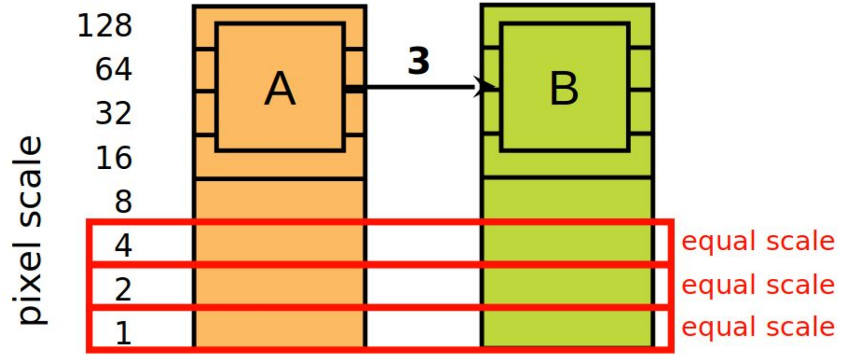


Multiscale Texture Synthesis: Key Concepts

- Exemplar Graph

- Gaussian Stack

- **Admissible Candidates**



$$\mathcal{A}(E^i_k) = \{\, E^j_l \mid \exists (i, j, k-l) \in \mathbb{E},\ 0 \le l < L \,\},$$

Multiscale Texture Synthesis: Algorithm

- Input
- Upsampling
- Jitter
- Correction

Multiscale Texture Synthesis: Algorithm

- **Input**
- Upsampling
- Jitter
- Correction

- Images are represented by coordinates
- Therefore they contain coordinates in format of *(i, k, u)*

Multiscale Texture Synthesis: Algorithm

- Input
- **Upsampling**
- Jitter
- Correction

- We upsample the given stack level to the next one:

$$S_t\left[2p+\Delta+\left(\tfrac{1}{2},\tfrac{1}{2}\right)\right] := \left(i, k+1, u+\lfloor h_k \Delta \rfloor \ (\mathrm{mod}\ m)\right) \quad ,$$

where $\Delta \in \left\{\left(\pm\frac{1}{2}, \pm\frac{1}{2}\right)\right\}$.

Multiscale Texture Synthesis: Algorithm

- Input
- Upsampling
- Jitter
- Correction

- Now that we have expanded the scale we need jittering:

$$S_l[p] := \left(i, k, u + J_l(p) \pmod{m}\right) ,$$

$$\text{where } J_l(p) = \left\lfloor h_k \mathcal{H}(p) \rho_l + \left(\tfrac{1}{2}, \tfrac{1}{2}\right) \right\rfloor$$

Multiscale Texture Synthesis: Algorithm

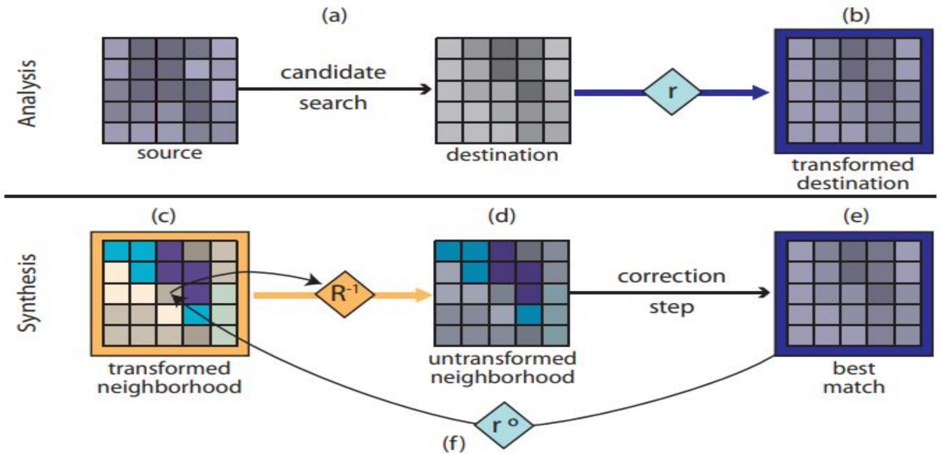
- Input
- Upsampling
- Jitter
- **Correction**

- Final step is to correct the new level
- Re-arranging neighborhoods
- *k-coherence algorithm*
- Pick admissible candidates that minimize error functional:

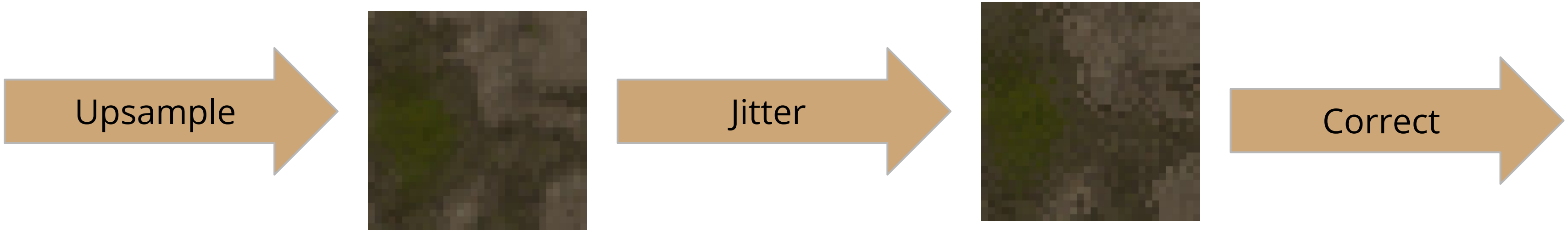
$$\sum_{\delta\in\{-2\ldots+2\}^2} \left\| E_k^i[u+\delta h_k] - E_l^j[v+\delta h_l] \right\|^2$$

Multiscale Texture Synthesis: Algorithm

- Input
- Upsampling
- Jitter
- **Correction**



Multiscale Texture Synthesis: Algorithm



Upsample

Jitter

Correct

Multiscale Texture Synthesis: Inconsistencies

- Corrections is possible to lead to inconsistencies

- Transfer functions

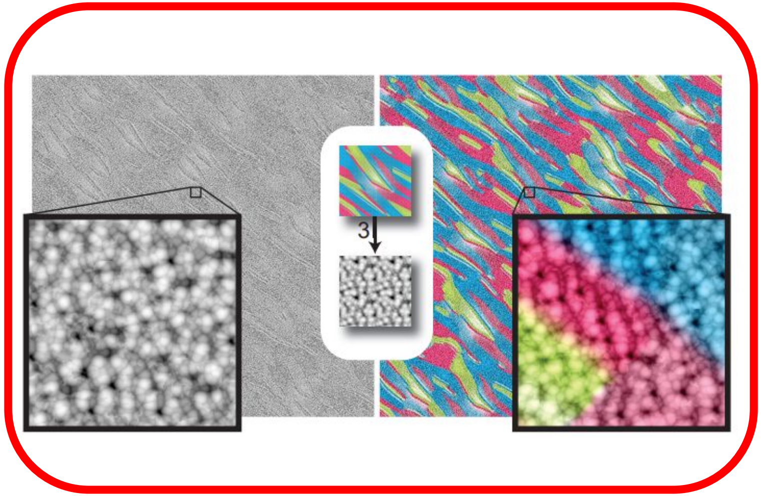
- Analysis and synthesis process

Multiscale Texture Synthesis: Inconsistencies

- **Corrections is possible to lead to inconsistencies**

- Transfer functions

- Analysis and synthesis process



Multiscale Texture Synthesis: Inconsistencies

- Corrections is possible to lead to inconsistencies

- **Transfer functions**

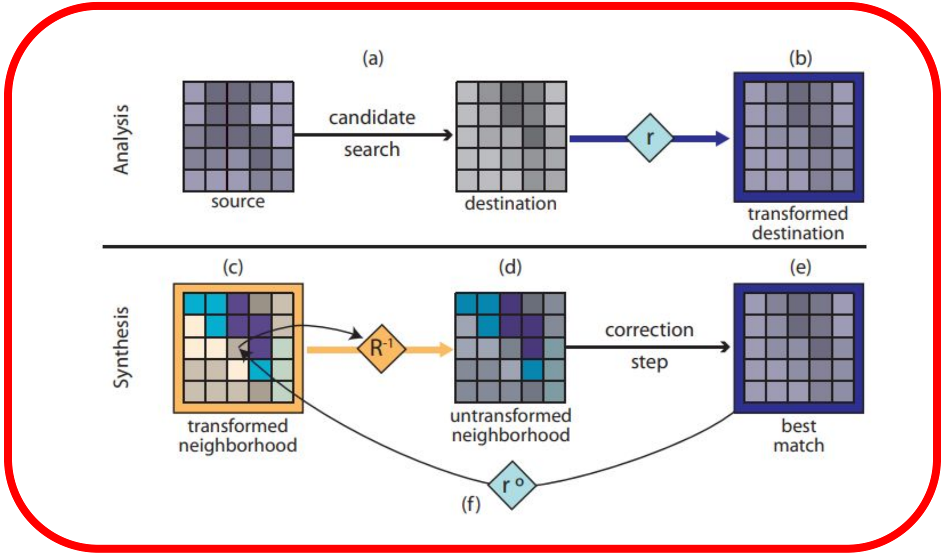
- Analysis and synthesis process

Multiscale Texture Synthesis: Inconsistencies

- Corrections is possible to lead to inconsistencies

- **Transfer functions**

- Analysis and synthesis process

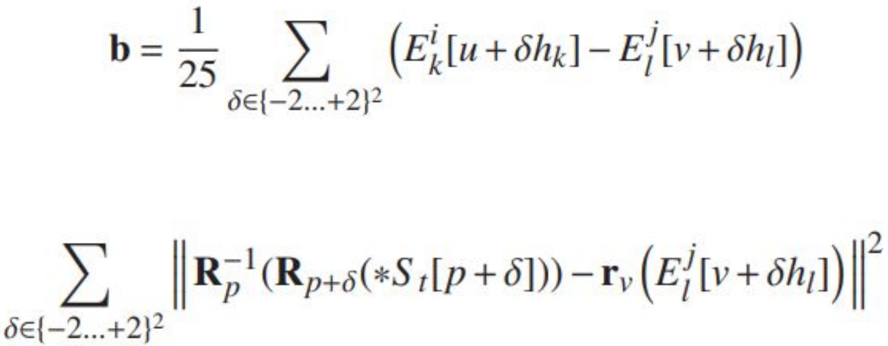


Multiscale Texture Synthesis: Inconsistencies

- Corrections is possible to lead to inconsistencies

- Transfer functions

- Analysis and synthesis process



$$\mathbf{b} = \frac{1}{25} \sum_{\delta \in \{-2 \ldots +2\}^2} \left(E_k^i[u + \delta h_k] - E_l^j[v + \delta h_l] \right)$$

$$\sum_{\delta \in \{-2 \ldots +2\}^2} \left\| \mathbf{R}_p^{-1}(\mathbf{R}_{p+\delta}(*S_t[p+\delta])) - \mathbf{r}_v \left(E_l^j[v + \delta h_l] \right) \right\|^2$$

Multiscale Texture Synthesis: Optimizations

- Superexemplar

- PCA Projection

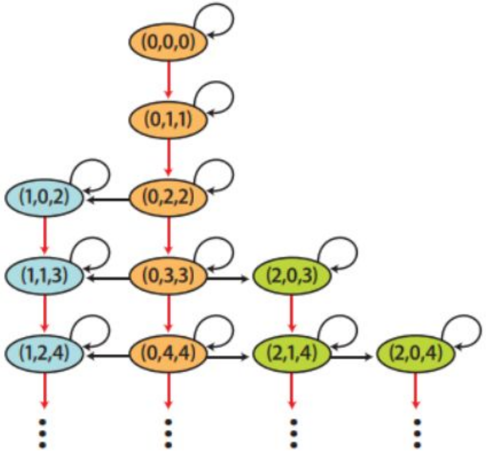
- Texture Packing

Multiscale Texture Synthesis: Optimizations

- **Superexemplar**

- PCA Projection

- Texture Packing



Multiscale Texture Synthesis: Optimizations

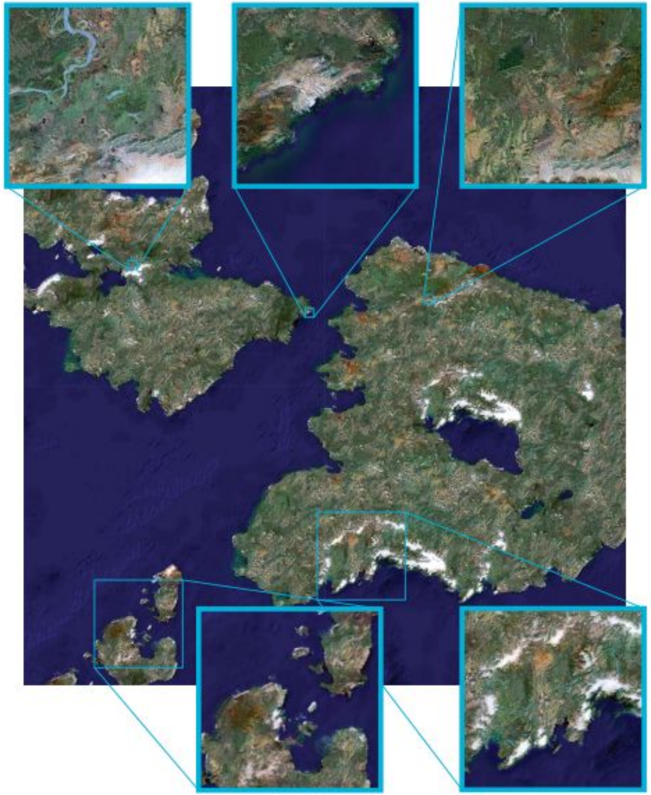
- Superexemplar
- **PCA Projection**
- Texture Packing

Multiscale Texture Synthesis: Optimizations

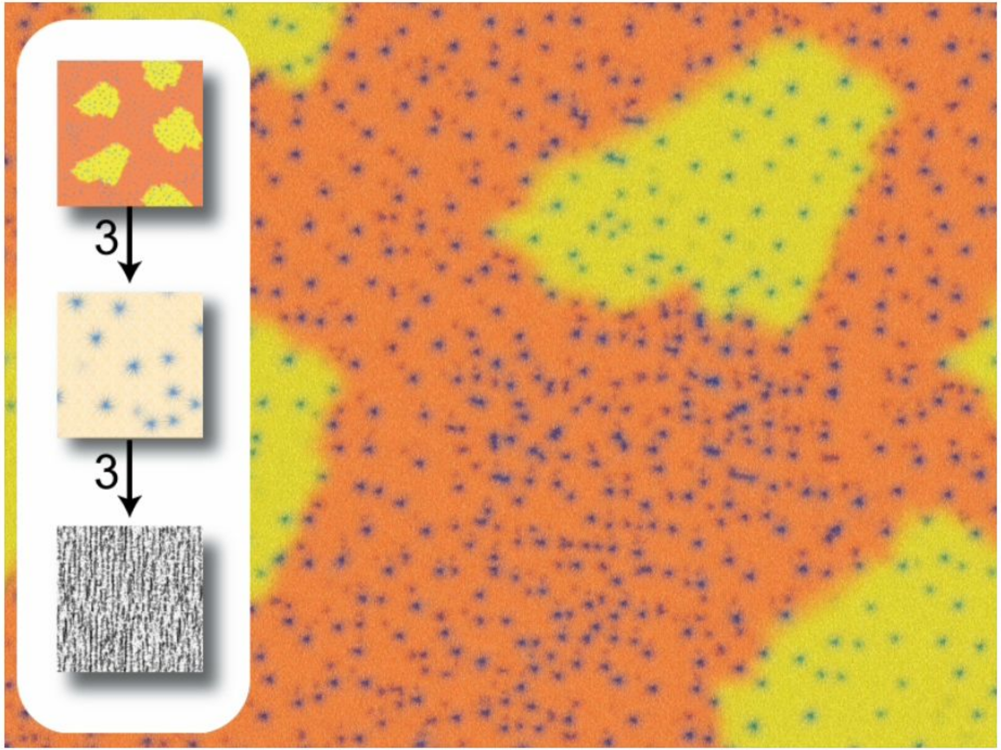
- Superexemplar
- PCA Projection
- **Texture Packing**

- All stack levels: RGB Texture
- 3 RGBA Textures for PCA-reduced candidate neighborhoods
- 16-bit RGBA for transfer functions

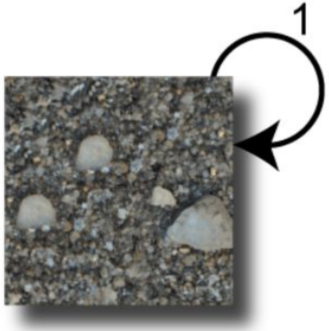
Multiscale Texture Synthesis: Results

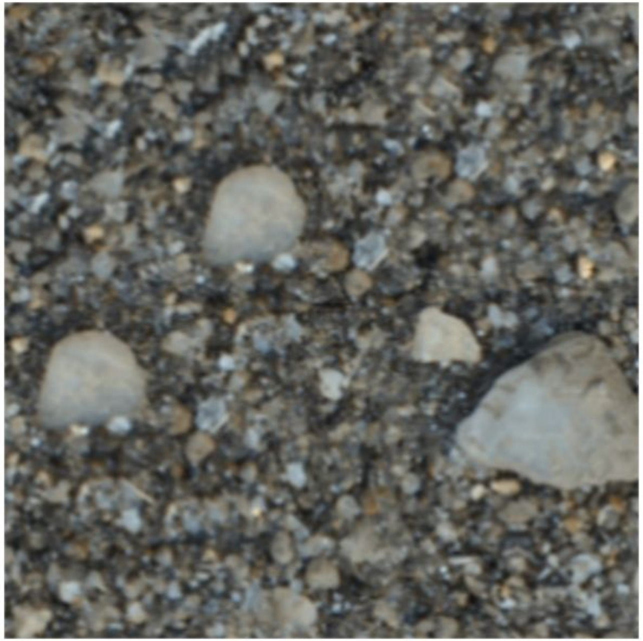


Multiscale Texture Synthesis: Results

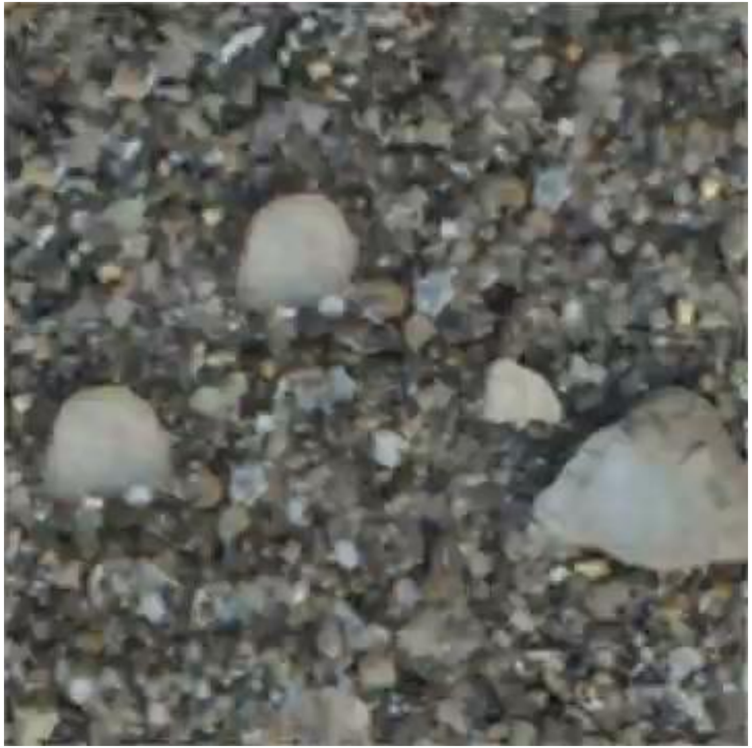


Multiscale Texture Synthesis: Results





Multiscale Texture Synthesis: Results



Thank you!