Texture Synthesis

Given a texture, create more

Don’t expect too much

And amazing successes

How does it work?

Big idea
- statistical assumptions:
  - Markov random field model
  - stationarity and ergodicity
  - find pixels with similar neighbors
  - scanline order (causal neighborhood)
  - implementation: exhaustive search

In a Nutshell

Find most similar neighborhood
- high dimensional point/vector

**Details**

**Choices**
- neighborhood size
- hierarchy
- distance measure
- acceleration structures
- cut search space

**Neighborhood Size**

<table>
<thead>
<tr>
<th>Neighborhood Size</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3</td>
<td>423</td>
</tr>
<tr>
<td>5x5</td>
<td>538</td>
</tr>
<tr>
<td>7x7</td>
<td>739</td>
</tr>
<tr>
<td>9x9</td>
<td>1026</td>
</tr>
<tr>
<td>11x11</td>
<td>1445</td>
</tr>
<tr>
<td>41x41</td>
<td>24350</td>
</tr>
</tbody>
</table>

**Hierarchical Approach**

Smaller neighborhoods ok

- Single res: 5x5
- Single res: 11x11
- 3 levels: 5x5

**Applications**

What are instances of synthesis?
- repair (inpainting)
- image editing
- extrapolation
- user control (introducing bias)

**Resources**

**Papers**
- Wei & Levoy
  - Fast Texture Synthesis using TSVQ
- Ashikhmin
  - Synthesizing Natural Textures
- check their web pages

**Enhancements**

**Additional ideas**
- store coordinates not values
- better for upsampling
- upsample, jitter, correct
- multiple passes
- parallel subpasses
- search only shifted pixel nghbd.
**Enhancements**

- Tong et al.: k-way coherence search
  - preprocess exemplar with nearest list
- Lefebvre+Hoppe: appearance space
  - distance to features as addl. data
  - PCA on neighborhoods

**Transformed Exemplar**

- 5×5 pixel neighborhood
- (Nonlinear dim. reduction also possible)
- RGB exemplar → Appearance-space exemplar → Appearance vector → PCA → 4D/3D → Transformed exemplar

**Example for 75D to 3D**

**Results**

**Results**