What is a Texture?
# Texture Spectrum

<table>
<thead>
<tr>
<th>regular</th>
<th>near-regular</th>
<th>irregular</th>
<th>near-stochastic</th>
<th>stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Pattern" /></td>
<td><img src="image2" alt="Pattern" /></td>
<td><img src="image3" alt="Pattern" /></td>
<td><img src="image4" alt="Pattern" /></td>
<td><img src="image5" alt="Pattern" /></td>
</tr>
<tr>
<td><img src="image6" alt="Pattern" /></td>
<td><img src="image7" alt="Pattern" /></td>
<td><img src="image8" alt="Pattern" /></td>
<td><img src="image9" alt="Pattern" /></td>
<td><img src="image10" alt="Pattern" /></td>
</tr>
<tr>
<td><img src="image11" alt="Pattern" /></td>
<td><img src="image12" alt="Pattern" /></td>
<td><img src="image13" alt="Pattern" /></td>
<td><img src="image14" alt="Pattern" /></td>
<td><img src="image15" alt="Pattern" /></td>
</tr>
<tr>
<td><img src="image16" alt="Pattern" /></td>
<td><img src="image17" alt="Pattern" /></td>
<td><img src="image18" alt="Pattern" /></td>
<td><img src="image19" alt="Pattern" /></td>
<td><img src="image20" alt="Pattern" /></td>
</tr>
</tbody>
</table>
Texture scandals!!
Bush campaign digitally altered TV ad

President Bush’s campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.

Original photograph
Two crucial algorithmic points

- Nearest neighbors
  - again and again and again

- Dynamic programming
  - likely new; we’ll use this again, too
Texture Synthesis

Efros & Leung ICCV99
How to paint this pixel?
Ask Neighbors

- What is the conditional probability distribution of $p$, given it’s neighbors?

Efros & Leung ICCV99
• Don’t bother to model the distribution
  • It’s already there, in the image

Efros & Leung ICCV99
Efros & Leung Algorithm

Synthesizing a pixel

non-parametric sampling

Efros & Leung ICCV99
Concerns

- Distance metric
- Neighborhood size
- Order to paint
Distance metric

- Normalized sum of squared distances
- Not all the neighbors worth the same
  - Gaussian mask

- Preserve the local structure
- Pick among reasonably similar neighborhoods
Neighborhood size

Efros & Leung ICCV99
Varying Window Size

Efros & Leung ICCV99
The Order matters
Some Results

Efros & Leung ICCV99
More Results
More Results

french canvas

rafia weave

Efros & Leung ICCV99
More Results

wood

granite

Efros & Leung ICCV99
More Results

white bread

brick wall

Efros & Leung ICCV99
Growing Regions
Hole Filling

Efros & Leung ICCV99
Hole Filling

Efros & Leung ICCV99
Extrapolation

Efros & Leung ICCV99
Failure Cases

Growing garbage

Verbatim copying

Efros & Leung ICCV99
Pros and Cons

• Very simple
• Easy to implement
• Promising results

• Very slooooonnnnnoowwww
• Idea:
  • Patches instead of pixels
Patch based

Observation
- neighbouring pixels are highly correlated

Idea:
- unit of synthesis = block

Synthesizing a block

Efros & Freeman SIGGRAPH01
Input texture

Block

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut

Efros & Freeman SIGGRAPH01
Minimal error boundary

overlapping blocks

vertical boundary

\[
\text{overlap error} \quad \quad 2 \quad \quad = \quad \quad \text{min. error boundary}
\]

Efros & Freeman SIGGRAPH01
Dynamic Programming

\[ e_{ij} = (B1_{ij}^{ov} - B2_{ij}^{ov})^2 \]

\[ E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}) \]
Dynamic Programming

\[ e_{ij} = (B1_{ij}^{ov} - B2_{ij}^{ov})^2 \]

\[ E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}) \]
Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut

Efros & Freeman SIGGRAPH01
More Results
Failures
Texture Transfer

- Take the texture from one object and paint it on another object

Decomposing shape and texture
  Very challenging
  Walk around
  Add some constraint to the search

Efros & Freeman SIGGRAPH01
Texture Transfer

Efros & Freeman SIGGRAPH01
parmesan

+ rice

=  

Effros & Freeman SIGGRAPH01
Image Analogies

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Image Analogies

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Image Analogies

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Image Analogies

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Training

Unfiltered source \((A)\)  

Filtered source \((A')\)
Learn to Blur

Unfiltered source ($A$)  Filtered source ($A'$)

Unfiltered target ($B$)  Filtered target ($B'$)

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Texture by Numbers

Unfiltered source ($A$)

Filtered source ($A'$)

Unfiltered ($B$)

Filtered ($B'$)

Hertzman, Jacobs, Oliver, Curless, and Salesin. SIGGRAPH01
Colorization

Unfiltered source (A)  Unfiltered target (B)

Filtered source (A')  Filtered target (B')

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Super-resolution

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Super-resolution (result!)

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Training images

Hertzman, Jacobs, Oliver, Curless, and Salesin, SIGGRAPH01
Order of inpainting matters
Choosing the order

Given a patch $\Psi_p$ centred at the point $p$ for some $p \in \delta\Omega$ (see fig. 3), its priority $P(p)$ is defined as the product of two terms:

$$P(p) = C(p)D(p). \quad (1)$$

We call $C(p)$ the confidence term and $D(p)$ the data term, and they are defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \overline{\Omega}} C(q)}{|\Psi_p|}, \quad D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}$$

where $|\Psi_p|$ is the area of $\Psi_p$, $\alpha$ is a normalization factor (e.g., $\alpha = 255$ for a typical grey-level image), and $n_p$ is a unit vector orthogonal to the front $\delta\Omega$ in the point $p$. The
Constraining the match region

- We don’t have to look for matches in the whole image
  - idea: allow user to “paint” good sources of matches on top of the image
Nearest Neighbor search

The core of most of the patch based methods
Very slow

Smarter neighborhood search

Barnes et al., SIGGRAPH09
Applications

(b) hole+constraints

(c) hole filled

(d) constraints

(e) constrained retarget

(f) reshuffle

Barnes et al., SIGGRAPH09
Retargeting

• Make an image bigger or smaller in one direction
  • eg change aspect ratio
• Traditional
  • cut off pixels
  • difficulty: lousy results
• Strategy
  • cut out a curve of pixels that “doesn’t matter much”
    • low energy at pixels
    • many energy functions, eg

\[ e_1(I) = |\frac{\partial}{\partial x} I| + |\frac{\partial}{\partial y} I| \]
Finding a seam=DP

Avidan, Shamir. SIGGRAPH07
Different energies give different results
- $e_1 = \text{abs gradient (as above)}$
- $e_{\text{hog}} = \text{(look for gradients in patch)}$
- $e_{\text{entropy}} = \text{(entropy of patch)}$
- $e_{\text{seg}} = \text{(segment image, $e_1$ in segments, 0 on boundaries)}$

(a) Original
(b) $e_1$
(c) $e_{\text{Entropy}}$
(d) $e_{\text{HoG}}$
(e) Segmentation and $L_1$
Retargeting
Seam removal
Scaling
Cropping

Avidan, Shamir, SIGGRAPH07
Retargeting

Avidan, Shamir, SIGGRAPH07
Avidan, Shamir, SIGGRAPH07
Can use constraints in retargeting
Constrained retargeting
Local scale editing

(a) building marked by user
(b) scaled up, preserving texture
(c) bush marked by user
(d) scaled up, preserving texture.
reshuffling

(a) input  (b) our reshuffling

Barnes et al., SIGGRAPH09