Texture

COS 429: Computer Vision

Acknowledgment: slides from Antonio Torralba, Kristen Grauman, Jitendra Malik, Alyosha Efros, and Tom Funkhouser
Texture

What is a texture?

[Google search results for texture]
Texture

What is a texture?
Texture

What is a texture?
Texture

- Texture: stochastic pattern that is stationary ("looks the same" at all locations)
- May be structured or random
Texture

Stochastic     Stationary
Texture

Stochastic     Stationary
Goal

- Computational representation of texture
  - Textures generated by same stationary stochastic process have same representation
  - Perceptually similar textures have similar representations

Hypothetical texture representation

5, 7, 34, 2, 199, 12
Applications

- Segmentation
- 3D Reconstruction
- Classification
- Synthesis

http://animals.nationalgeographic.com/
Applications

- Segmentation
- 3D Reconstruction
- Classification
- Synthesis
Applications

- Segmentation
- 3D Reconstruction
- Classification
- Synthesis
Applications

- Segmentation
- 3D Reconstruction
- Classification
- Synthesis
Applications

- Segmentation
- 3D Reconstruction
- Classification
- Synthesis
Texture Representation?

- What makes a good texture representation?
  - Textures generated by same stationary stochastic process have same representation
  - Perceptually similar textures have similar representations
Approaches

• Statistics of filter banks
• Textons
• Markov Random Fields
Approaches

• Statistics of filter banks
• Textons
• Markov Random Fields
Filter-Based Texture Representation

- Research suggests that the human visual system performs **local** spatial frequency analysis (Gabor filters)

Texture Representation

• Analyze textures based on the responses of linear filters
  – Use filters that look like patterns (spots, edges, bars, …)
  – Compute magnitudes of filter responses

• Represent textures with statistics of filter responses within local windows
  – Histogram of feature responses for all pixels in window
Texture Representation Example

original image

derivative filter responses, squared

statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Win. #1</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>

Grauman
Texture Representation Example

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th>Window</th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows
Texture Representation Example

- Original image
- Derivative filter responses, squared
- Statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Window</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Grauman
Texture Representation Example

Statistics to summarize patterns in small windows.
Texture Representation Example

Statistics to summarize patterns in small windows.

<table>
<thead>
<tr>
<th>Window</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Far: dissimilar textures
Close: similar textures
Filter Banks

- Previous example used two filters, resulting in 2-dimensional feature vector
  - $x$ and $y$ derivatives revealed local structure
- Filter bank: many filters
  - Higher-dimensional feature space
  - Distance still related to similarity of local structure
Filter banks

- What filters to put in the bank?
  - Combination of different scales, orientations, patterns

<table>
<thead>
<tr>
<th>Scales</th>
<th>&quot;Edges&quot;</th>
<th>&quot;Bars&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Spots&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Grauman
You Try: Can you match the texture to the response?

Filters

1

2

3

Mean abs responses

A

B

C

Derek Hoiem
Filter Bank Texture Representation

- Pass image through filter bank
- **Analysis:** Compile statistics of filter outputs
  - Mean
  - Mean + variance
  - Histogram
- **Synthesis:**
  - Start with random noise image
  - Adjust histograms to match original image
  - Re-synthesize image from filter outputs
Histogram Equalization

- **Given**: two histograms of intensity $H_1$ and $H_2$

- **Goal**: function that remaps intensities to make new histogram $H_1'$ equal $H_2$
Histogram Equalization

1. Compute CDFs (integrals) of histograms

2. For each intensity, map through CDF 1 then look up inverse in CDF 2
Application: Texture Synthesis

Original Texture

Synthesized Texture

Heeger and Bergen
Application: Retrieval

• Retrieve similar images based on texture

Approaches

- Statistics of filter banks
- Textons
- Markov Random Fields
Textons

• Elements ("textons") either identical or come from some statistical distribution

• Can analyze in natural images
Clustering Textons

- Output of bank of $n$ filters can be thought of as a vector in $n$-dimensional space.
- Can *cluster* these vectors using *k*-means [Malik et al.]
- Result: dictionary of most common textures
Clustering Textons

Image

Clustered Textons

Texton to Pixel Mapping
Using Texture in Segmentation

- Compute histogram of how many times each of the $k$ clusters occurs in a neighborhood.
- Define similarity of histograms $h_i$ and $h_j$ using $\chi^2$

$$\chi^2 = \frac{1}{2} \sum_k \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}$$

- Different histograms $\rightarrow$ separate regions
Application: Segmentation
Approaches

• Statistics of filter banks
• Textons
• Markov Random Fields
Markov Random Fields

• Different way of thinking about textures
• Premise: probability distribution of a pixel depends on values of neighbors
• Probability the same throughout image
  – Extension of Markov chains
Motivation from Language

- Shannon (1948) proposed a way to synthesize new text using N-grams
  - Use a large text to compute probability distributions of each letter given N–1 previous letters
  - Starting from a seed repeatedly sample the conditional probabilities to generate new letters
  - Can do this with image patches!
Texture Synthesis Based on MRF

• For each pixel in destination:
  – Take already-synthesized neighbors
  – Find closest match in original texture
  – Copy pixel to destination

• Efros & Leung 1999
  – Speedup by Wei & Levoy 2000
  – Extension to copying whole blocks by Efros & Freeman 2001
Efros & Leung Algorithm

- Compute output pixels in scanline order (top-to-bottom, left-to-right)
Efros & Leung Algorithm

- Find candidate pixels based on similarities of pixel features in neighborhoods.
Efros & Leung Algorithm

- Similarities of pixel neighborhoods can be computed with squared differences (SSD) of pixel colors and/or filter bank responses
Efros & Leung Algorithm

- For each pixel $p$:
  - Find the best matching $K$ windows from the input image
  - Pick one matching window at random
  - Assign $p$ to be the center pixel of that window
Synthesis Results
Synthesis Results

white bread

brick wall
Hole Filling

- Fill pixels in “onion skin” order
  - Within each “layer”, pixels with most neighbors are synthesized first
  - Normalize error by the number of known pixels
  - If no close match can be found, the pixel is not synthesized until the end
Hole Filling
Extrapolation