Texture

COS 429
Princeton University
Texture

What is a texture?
Texture

What is a texture?
Texture

What is a texture?
Texture

- Texture is stochastic and stationary (same regardless of position)
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?
Applications

Segmentation
3D Reconstruction
Classification
Synthesis
etc.

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

Input ➔ Output

Alyosha Efros
Texture Representation?

What’s a good texture representation?

- Textures generated by same stationary stochastic process have same representation
- Perceptually similar textures have similar representations
Texture Representation?

Julesz conjectured that the putative units of pre-attentive human texture perception are related to local features, such as edges, line ends, blobs, etc.

B. Julesz. Textons, the Elements of Texture Perception, and their Interactions. 
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

<table>
<thead>
<tr>
<th>statistics to summarize patterns in small windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean d/dx value</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Win. #1</td>
</tr>
</tbody>
</table>

Kristen 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>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Statistics to summarize patterns in small windows

Kristen Grauman
Texture Representation Example

Original image → derivative filter responses, squared → statistics to summarize patterns in small windows

<table>
<thead>
<tr>
<th>Window</th>
<th>$\text{mean } d/dx$ value</th>
<th>$\text{mean } d/dy$ 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>

Kristen Grauman
Texture Representation Example

---

**original image**

**derivative filter responses, squared**

<table>
<thead>
<tr>
<th>Window</th>
<th>$\text{mean } d/dx$ value</th>
<th>$\text{mean } d/dy$ 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>

... statistics to summarize patterns in small windows

---

Kristen Grauman
Texture Representation Example

Dimension 1 (mean d/dx value)

Dimension 2 (mean d/dy value)

<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>

... statistics to summarize patterns in small windows
Texture Representation Example

Windows with primarily horizontal edges

Windows with small gradient in both directions

Windows with primarily vertical edges

Both

Dimension 1 (mean d/dx value)

Dimension 2 (mean d/dy value)

Table:

<table>
<thead>
<tr>
<th>Windows</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>

Statistics to summarize patterns in small windows

Kristen Grauman
Texture Representation Example

original image

derivative filter responses, squared

visualization of the assignment to texture “types”
Texture Representation Example

Dimension 1 (mean d/dx value)

Dimension 2 (mean d/dy value)

Far: dissimilar textures

Close: similar textures

<table>
<thead>
<tr>
<th></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>

statistics to summarize patterns in small windows
Texture Representation Example

$D(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$
Distance reveals how dissimilar texture from window a is from texture in window b.
Filter banks

• Our previous example used two filters, and resulted in a 2-dimensional feature vector to describe texture in a window.
  – x and y derivatives revealed something about local structure.

• We can generalize to apply a collection of multiple \((d)\) filters: a “filter bank”

• Then our feature vectors will be \(d\)-dimensional.
  – still can think of nearness, farness in feature space
Filter banks

- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

Matlab code available for these examples: http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Kristen Grauman
Multivariate Gaussian

\[
p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right).
\]

\[
\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}
\]

\[
\Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix}
\]

\[
\Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}
\]
Filter bank
Showing magnitude of responses

Kristen Grauman
You try: Can you match the texture to the response?

Filters

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mean abs responses

1

2

3

Derek Hoiem
Representing texture by mean abs response

Filters

Mean abs responses

Derek Hoiem
Form a feature vector from the list of responses at each pixel.

[r1, r2, ..., r38]
Texture Representation Summary

• Analyze textures based on the responses of linear filters
  – Use oriented filters at multiple scales
  – Compute magnitudes of filter responses
• Represent textures with statistics of filter responses within local windows
  – Histogram of feature responses for all pixels in window
Example uses of texture analysis in computer vision
Similarity

• Predict perceptual similarity of textures
  – based on Euclidean distance ($L_2$) in d-dimensional feature space

\[ D(a, b) = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2} \]
Segmentation

Segment images with color and texture
Segmentation

Segment aerial imagery by textures

Kristen Grauman
Classification

Figure by Varma & Zisserman
Classification

Characterizing scene categories by texture

Texture synthesis

Create new image (e.g., of different size or shape) with texture of an input image
Texture synthesis

How can we do this?

Input

Output

Alyosha Efros
Texture synthesis

Copying texture multiple times produces seams and repetitions

Input

Bad Output
Texture Synthesis

Can synthesize new texture by sampling from the probability distribution of local neighborhoods

$P(x|\text{neighborhood of pixels around } x)$

Input image

Synthesized image
Texture Synthesis

Texture is stochastic and stationary
- $p(\text{pixel}) = p(\text{pixel} | \text{neighborhood})$
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 words too …
Mark V. Shaney (Bell Labs)

• Results (using alt.singles corpus):
  – “As I've commented before, really relating to someone involves standing next to impossible.”
  – “One morning I shot an elephant in my arms and kissed him.”
  – “I spent an interesting evening recently with a grain of salt.”

• Notice how well local structure is preserved!

Now let’s try this in 2D...
Efros & Leung Algorithm

Compute output pixels in scanline order (top-to-bottom, right-to-left)

completed portion (grey)

output image
Efros & Leung Algorithm

For each pixel, 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

\[ \left\| (A - B) \right\|^2 \]
Efros & Leung Algorithm

- For each pixel $x$:
  - Find the best matching $K$ windows from the input image
  - Pick one matching window at random
  - Assign $x$ to be the center pixel of that window
Efros & Leung Algorithm

For each pixel \( x \):
- Find the best matching \( K \) windows from the input image
- Pick one matching window at random
- Assign \( x \) to be the center pixel of that window
Efros & Leung Algorithm

• For each pixel $x$:
  – Find the best matching $K$ windows from the input image
  – Pick one matching window at random
  – Assign $x$ to be the center pixel of that window
Synthesis results

french canvas

rafia weave

Slide from Alyosha Efros, ICCV 1999
Synthesis results

white bread

brick wall

Slide from Alyosha Efros, ICCV 1999
Synthesis results
Failure Cases

Growing garbage

Verbatim copying

Alyosha Efros

Slide from Alyosha Efros, ICCV 1999
Example Applications

• Hole filling and extrapolation
  – 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

Slide from Alyosha Efros, ICCV 1999
Extrapolation
Summary

• Texture is a useful property that is often indicative of materials, appearance cues

• **Texture representations** attempt to summarize repeating patterns of local structure

• **Filter banks** useful to measure redundant variety of structures in local neighborhood
  – Feature spaces can be multi-dimensional

• Neighborhood statistics can be exploited to “sample” or **synthesize** new texture regions
  – Example-based technique