CS376 Computer Vision
Lecture 5: Texture

Qixing Huang
Feb. 6th 2019
Today: Texture

What defines a texture?
Includes: more regular patterns
Includes: more random patterns
Scale and texture
Texture-related tasks

• **Shape from texture**
  – Estimate surface orientation or shape from image texture
Shape from texture

• Use deformation of texture from point to point to estimate surface shape


Slide Credit: Kristen Grauman
Analysis vs. Synthesis

Why analyze texture?

Images: Bill Freeman, A. Efros

Slide Credit: Kristen Grauman
Texture-related tasks

• Shape from texture
  – Estimate surface orientation or shape from image texture

• Segmentation/classification from texture cues
  – Analyze, represent texture
  – Group image regions with consistent texture

• Synthesis
  – Generate new texture patches/images given some examples
What kind of response will we get with an edge detector for these images?

Images from Malik and Perona, 1990

Slide Credit: Kristen Grauman
...and for this image?

Image credit: D. Forsyth

Slide Credit: Kristen Grauman
Why analyze texture?

Importance to perception:
• Often indicative of a material’s properties
• Can be important appearance cue, especially if shape is similar across objects
• Aim to distinguish between shape, boundaries, and texture

Technically:
• Representation-wise, we want a feature one step above “building blocks” of filters, edges.
Psychophysics of texture

• Some textures distinguishable with preattentive perception—without scrutiny, eye movements [Julesz 1975]

Same or different?
Capturing the local patterns with image measurements


Scale of patterns influences discriminability

Size-tuned linear filters

Slide Credit: Kristen Grauman
Texture representation

- Textures are made up of repeated local patterns, so:
  - Find the patterns
    - Use filters that look like patterns (spots, bars, raw patches...)
    - Consider magnitude of response
  - Describe their statistics within each local window, e.g.,
    - Mean, standard deviation
    - Histogram
    - Histogram of “prototypical” feature occurrences
Texture representation: example

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th>Win. #1</th>
<th>mean ( \frac{d}{dx} ) value</th>
<th>mean ( \frac{d}{dy} ) value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide credit: Kristen Grauman
Texture representation: example

![Original Image](image1)

![Derivative Filter Responses, Squared](image2)

<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

Slide credit: Kristen Grauman
Texture representation: example

original image

derivative filter responses, squared

mean \( \frac{d}{dx} \) value | mean \( \frac{d}{dy} \) value
--- | ---
Win. #1 | 4 | 10
Win. #2 | 18 | 7

statistics to summarize patterns in small windows

Slide credit: Kristen Grauman
Texture representation: example

original image

derivative filter responses, squared

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

Slide credit: Kristen Grauman
Texture representation: example

- Mean $d/dx$ value
- Mean $d/dy$ value

Win. #1: 4, 10
Win. #2: 18, 7
Win. #9: 20, 20

Statistics to summarize patterns in small windows

Slide credit: Kristen Grauman
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>
<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

Slide credit: Kristen Grauman
Texture representation: example

original image

derivative filter responses, squared

visualization of the assignment to texture “types”

Slide credit: Kristen Grauman
Texture representation: example

Far: dissimilar textures
Close: similar textures

<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

Slide credit: Kristen Grauman
Texture representation: example

\[ D(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \]
Texture representation: example

Distance reveals how dissimilar texture from window a is from texture in window b.

Slide credit: Kristen Grauman
Texture representation: window scale

• We’re assuming we know the relevant window size for which we collect these statistics.

Possible to perform scale selection by looking for window scale where texture description not changing.
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
Multivariate Gaussian

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

\[ \Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix} \]

Slide credit: Kristen Grauman
Filter bank

Slide credit: Kristen Grauman
Showing magnitude of responses

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

Filters

A

B

C

Slide Credit: Derek Hoiem
Representing texture by mean abs response

Filters

Mean abs responses

Slide Credit: Derek Hoiem
We can form a feature vector from the list of responses at each pixel.

[r1, r2, ..., r38]
**d-dimensional features**

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

Euclidean distance \((L_2)\)

Slide credit: Kristen Grauman
Example uses of texture in vision: analysis
Classifying materials, “stuff”

Figure by Varma & Zisserman

Slide Credit: Kristen Grauman
Texture-related tasks

• **Shape from texture**
  – Estimate surface orientation or shape from image texture

• **Segmentation/classification** from texture cues
  – Analyze, represent texture
  – Group image regions with consistent texture

• **Synthesis**
  – Generate new texture patches/images given some examples
Texture synthesis

• Goal: create new samples of a given texture
• Many applications: virtual environments, hole-filling, texturing surfaces
The Challenge

• Need to model the whole spectrum: from repeated to stochastic texture


Slide Credit: Kristen Grauman
Markov Random Field

First-order MRF:

- probability that pixel $X$ takes a certain value given the values of neighbors $A$, $B$, $C$, and $D$:

$$P(X|A, B, C, D)$$

Source: S. Seitz
Texture Synthesis \cite{Efros&Leung, ICCV 99}

- Can apply 2D version of text synthesis

Texture corpus (sample)

Output

Slide Credit: Kristen Grauman
Texture synthesis: intuition

We want to insert **pixel intensities** based on existing nearby pixel values.

Distribution of a value of a pixel is conditioned on its neighbors alone.
Synthesizing One Pixel

- What is $P(x | \text{neighborhood of pixels around } x)$?
- Find all the windows in the image that match the neighborhood
- To synthesize $x$
  - pick one matching window at random
  - assign $x$ to be the center pixel of that window
- An exact neighbourhood match might not be present, so find the best matches using SSD error and randomly choose between them, preferring better matches with higher probability
Neighborhood Window

Slide from Alyosha Efros, ICCV 1999
Varying Window Size

Increasing window size

Slide from Alyosha Efros, ICCV 1999
Synthesis results

french canvas

rafi weave

Slide from Alyosha Efros, ICCV 1999
Synthesis results

white bread

brick wall

Slide from Alyosha Efros, ICCV 1999
Image Quilting [Efros & Freeman 2001]

• **Observation:** neighbor pixels are highly correlated

**Idea:** unit of synthesis = block

• Exactly the same but now we want $P(B|N(B))$

• Much faster: synthesize all pixels in a block at once
Input texture

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut
Minimal error boundary

overlapping blocks

vertical boundary

overlap error

min. error boundary
(Manual) texture synthesis in the media

Slide Credit: Kristen Grauman
(Manual) texture synthesis in the media