Announcements

• Write your CS login ID on the pset hardcopy

Review: last time

• Edge detection:
  – Filter for gradient
  – Threshold gradient magnitude, thin

• Binary image analysis
  – Connected components to find regions
  – Morphological operators to “clean up”

Texture

What defines a texture?

Includes: more regular patterns

Includes: more random patterns
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

Shape from texture

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

Analysis vs. Synthesis

Images: Bill Freeman, A. Efros
What kind of response will we get with an edge detector for these images?

Images from Malik and Perona, 1990

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]
Capturing the local patterns with image measurements


- Scale of patterns influences discriminability
- Size-tuned linear filters

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
    - Mean, standard deviation
    - Histogram
    - Histogram of “prototypical” feature occurrences

Texture representation: example

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

Win. #2

<table>
<thead>
<tr>
<th>Win. #</th>
<th>dx mean value</th>
<th>dy mean value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#2</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

Statistics to summarize patterns in small windows

original image
derivative filter responses, squared
Texture representation: example

original image

derivative filter responses, squared

statistics to summarize patterns in small windows

Texture representation: example

original image

derivative filter responses, squared

statistics to summarize patterns in small windows

Texture representation: example

original image

derivative filter responses, squared

visualization of the assignment to texture “types”

Texture representation: example

original image

derivative filter responses, squared

statistics to summarize patterns in small windows
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

Multivariate Gaussian

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

\[
\Sigma = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}
\]
Filter bank
We can form a feature vector from the list of responses at each pixel.

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

\[ r_1, r_2, \ldots, r_{38} \]

General definition of inter-point Euclidean distance ($L_2$).

Example uses of texture in vision: analysis
Classifying materials, “stuff”

Figure by Varma & Zisserman

Texture features for image retrieval


Characterizing scene categories by texture


Segmenting aerial imagery by textures


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


Markov Chains

Markov Chain

• a sequence of random variables \(X_1, X_2, \ldots, X_n\)

• \(X_t\) is the state of the model at time \(t\)

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5 \]

• Markov assumption: each state is dependent only on the previous one
  – dependency given by a conditional probability:
    
Markov Chain Example: Text

“*A dog is a man’s best friend. It’s a dog eat dog world out there.*”

\[
P(X_t|X_{t-1})
\]

Text synthesis

Create plausible looking poetry, love letters, term papers, etc.

Most basic algorithm

1. Build probability histogram
   – find all blocks of \(N\) consecutive words/letters in training documents
   – compute probability of occurrence \(p(x_t|x_{t-1}, \ldots, x_{t-(N-1)})\)

Text synthesis

• Results:
  – “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”

Synthesizing Computer Vision text

• What do we get if we extract the probabilities from the F&P chapter on Linear Filters, and then synthesize new statements?


Source: S. Seitz

Synthesizing Computer Vision text

Check out Yisong Yue’s website implementing text generation: build your own text Markov Chain for a given text corpus. [http://www.yisongyue.com/shaney/index.php]
Synthesized text

- This means we cannot obtain a separate copy of the best studied regions in the sum.
- All this activity will result in the primate visual system.
- The response is also Gaussian, and hence isn’t bandlimited.
- Instead, we need to know only its response to any data vector, we need to apply a low pass filter that strongly reduces the content of the Fourier transform of a very large standard deviation.
- It is clear how this integral exist (it is sufficient for all pixels within a $2k + 1 \times 2k + 1 \times 2k + 1 \times 2k + 1$ — required for the images separately.

Markov Random Field

A Markov random field (MRF)
- generalization of Markov chains to two or more dimensions.

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

\[
P(X|A, B, C, D)
\]

Texture Synthesis [Efros & Leung, ICCV 99]

Can apply 2D version of text synthesis

Texture corpus (sample)

Output

Texture synthesis: intuition

Before, we inserted the next word based on existing nearby words...

Now we want to insert pixel intensities based on existing nearby pixel values.

Sample of the texture ("corpus")

Place we want to insert next

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

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

**Growing Texture**

- Starting from the initial image, “grow” the texture one pixel at a time

**Synthesis results**

- french canvas
- rafia weave

**Synthesis results**

- white bread
- brick wall

**Synthesis results**

**Failure Cases**

- Growing garbage
- Verbatim copying
• The Efros & Leung algorithm
  – Simple
  – Surprisingly good results
  – Synthesis is easier than analysis!
  – ...but very slow

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

Minimal error boundary
  overlapping blocks
  vertical boundary
  overlap error
  min. error boundary
Texture Transfer

- Take the texture from one object and "paint" it onto another object
  - This requires separating texture and shape
  - That's HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading
- Then, just add another constraint when sampling: similarity to underlying image at that spot

(Manual) texture synthesis in the media
(Manual) texture synthesis in the media


Synthesizing textures when constructing 3d models of archaeological sites

A. Zalesny et al., Realistic Textures for Virtual Anastylosis

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

So far: features and filters

Transforming and describing images; textures, colors, edges
Next: Grouping & fitting

Clustering, segmentation, fitting; what parts belong together?

Coming up

• Next time:
  Segmentation and grouping
  – For Thurs: read F&P Chapter 14

• Reminder:
  – Problem set 1 due Sept 21 (Monday) 11:59 PM