Feature detectors and descriptors
Course announcements

• Homework 1 is due tonight at 23:59!

• Homework 2 will be posted tonight and will be due on Monday, February 25th.

• There are additional office hours today: 3-5pm, covered by Abhay, at the graphics lounge in Smith Hall.
Overview of today’s lecture

Leftover from lecture 5:

• Finish Harris corner detector.
• Multi-scale detection.

New in lecture 6:

• Why do we need feature descriptors?
• Designing feature descriptors.
• MOPS descriptor.
• GIST descriptor.
• Histogram of Textons descriptor.
• HOG descriptor.
• SIFT.
Most of these slides were adapted from:

Why do we need feature descriptors?
If we know where the **good** features are, how do we **match** them?
Object instance recognition

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002
Image mosaicing
How do we describe an image patch?

Patches with similar content should have similar descriptors.
Designing feature descriptors
Photometric transformations
Geometric transformations

objects will appear at different scales, translation and rotation
What is the best descriptor for an image feature?
Image patch

Just use the pixel values of the patch

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{pmatrix} 
\rightarrow 
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 \\
\end{pmatrix}
\]

vector of intensity values

Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)
Tiny Images
Just down-sample it!
Simple, fast, robust to small affine transforms.
Image patch

Just use the pixel values of the patch

![Image patch example]

Perfectly fine if geometry and appearance is unchanged
(a.k.a. template matching)

What are the problems?
Image patch

Just use the pixel values of the patch

\[
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{array}
\]

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\end{array}
\]

vector of intensity values

Perfectly fine if geometry and appearance is unchanged (a.k.a. template matching)

What are the problems?
How can you be less sensitive to absolute intensity values?
Image gradients

Use pixel differences

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{pmatrix}
\rightarrow
\begin{pmatrix}
- & + & + \\
- & - & - \\
+ & + & +
\end{pmatrix}
\]

vector of x derivatives

‘binary descriptor’

Feature is invariant to absolute intensity values

What are the problems?
Image gradients

Use pixel differences

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

\[ \begin{pmatrix} - & + & + & - & - & + \end{pmatrix} \]

vector of x derivatives

Feature is invariant to absolute intensity values

What are the problems?
How can you be less sensitive to deformations?
Color histogram

Count the colors in the image using a histogram

Invariant to changes in scale and rotation

What are the problems?
Color histogram

Count the colors in the image using a histogram

Invariant to changes in scale and rotation

What are the problems?
Color histogram

Count the colors in the image using a histogram

Invariant to changes in scale and rotation

What are the problems?
How can you be more sensitive to spatial layout?
Spatial histograms

Compute histograms over spatial ‘cells’

Retains rough spatial layout
Some invariance to deformations

What are the problems?
Spatial histograms

Compute histograms over spatial ‘cells’

Retains rough spatial layout
Some invariance to deformations

What are the problems?
How can you be completely invariant to rotation?
Orientation normalization

Use the dominant image gradient direction to normalize the orientation of the patch

save the orientation angle $\theta$ along with $(x, y, s)$

What are the problems?
MOPS descriptor
Multi-Scale Oriented Patches (MOPS)

Multi-Scale Oriented Patches (MOPS)

Given a feature \((x, y, s, \theta)\)

Get 40 x 40 image patch, subsample every 5th pixel
(what’s the purpose of this step?)

Subtract the mean, divide by standard deviation
(what’s the purpose of this step?)

Haar Wavelet Transform
(what’s the purpose of this step?)
Multi-Scale Oriented Patches (MOPS)


Given a feature $(x, y, s, \theta)$

Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)

Subtract the mean, divide by standard deviation
(what’s the purpose of this step?)

Haar Wavelet Transform
(what’s the purpose of this step?)
Multi-Scale Oriented Patches (MOPS)


Given a feature \((x, y, s, \theta)\)

Get 40 x 40 image patch, subsample every 5th pixel (low frequency filtering, absorbs localization errors)

Subtract the mean, divide by standard deviation (removes bias and gain)

Haar Wavelet Transform

(what’s the purpose of this step?)
Multi-Scale Oriented Patches (MOPS)


Given a feature \((x, y, s, \theta)\)

Get 40 x 40 image patch, subsample every 5th pixel
(low frequency filtering, absorbs localization errors)

Subtract the mean, divide by standard deviation
(removes bias and gain)

Haar Wavelet Transform
(low frequency projection)
Haar Wavelets

(actually, Haar-like features)

Use responses of a bank of filters as a descriptor

We will see later in class how to compute Haar wavelet responses efficiently (in constant time) with integral images
GIST descriptor
GIST

1. Compute filter responses (filter bank of Gabor filters)

2. Divide image patch into $4 \times 4$ cells

3. Compute filter response averages for each cell

4. Size of descriptor is $4 \times 4 \times N$, where $N$ is the size of the filter bank
Gabor Filters
(1D examples)

\[ e^{-\frac{x^2}{2\sigma^2}} \sin (2\pi \omega x) \quad e^{-\frac{x^2}{2\sigma^2}} \cos (2\pi \omega x) \]
2D Gabor Filters

\[ e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(2\pi (k_x x + k_y y)) \]
Odd Gabor filter

... looks a lot like...

Gaussian Derivative
Even Gabor filter

... looks a lot like...

Laplacian
If scale small compared to inverse frequency, the Gabor filters become derivative operators.

\[ \sigma = 2 \quad f = 1/6 \]

\[ \approx G^x_\sigma \approx G^{xx}_\sigma \]
Directional edge detectors
GIST

1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

What is the GIST descriptor encoding?
1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

What is the GIST descriptor encoding?

Rough spatial distribution of image gradients
Histogram of Textons descriptor
Texture is characterized by the repetition of basic elements or textons.

For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters.
Histogram of Textons descriptor

1. **Training image**
2. **Filter Responses**
3. **Texton Map**
4. **Encoding**
5. **Pooling**
6. **Histogram of textons in image**

- **Training image**: An input image used for training.
- **Filter Responses**: The output of applying filters to the training image.
- **Texton Map**: A representation of textons in the image.
- **Encoding**: The process of converting textons into a histogram.
- **Pooling**: A technique to aggregate texton responses.
Learning Textons from data


Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Dictionary / Thesaurus

Textons, Contours and Regions: Cue Integration in Image Segmentation.
Learning Textons from data

Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Dictionary / Thesaurus

Texton Dictionary
Example of Filter Banks

Isotropic Gabor

Gaussian derivatives at different scales and orientations
Learning Textons from data

Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Texton Dictionary

Filter

patches

We will learn more about clustering later in class (Bag of Words lecture).
HOG descriptor
Dalal, Triggs. **Histograms of Oriented Gradients** for Human Detection. CVPR, 2005

- **Cell** (8x8 pixels)
- **Block** (2x2 cells)
- **Histogram of ‘unsigned’ gradients**
- **Soft binning**
- **Gradient magnitude histogram** (one for each cell)
- **Concatenate and L-2 normalization**

**Single scale, no dominant orientation**
Pedestrian detection

1 cell step size

128 pixels
16 cells
15 blocks

64 pixels
8 cells
7 blocks

Redundant representation due to overlapping blocks

How many times is each inner cell encoded?

http://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/
SIFT
SIFT
(Scale Invariant Feature Transform)

SIFT describes both a detector and descriptor

1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor
1. Multi-scale extrema detection

First octave

Second octave

Gaussian

Difference of Gaussian (DoG)
Gaussian

Laplacian
Scale-space extrema

Selected if larger than all 26 neighbors

Scale of Gaussian variance

Difference of Gaussian (DoG)
2. Keypoint localization

2nd order Taylor series approximation of DoG scale-space

\[ f(x) = f + \frac{\partial f^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 f}{\partial x^2} x \]

\[ x = \{x, y, \sigma\} \]

Take the derivative and solve for extrema

\[ x_m = -\frac{\partial^2 f^{-1}}{\partial x^2} \frac{\partial f}{\partial x} \]

Additional tests to retain only strong features
3. Orientation assignment

For a keypoint, \( L \) is the \textbf{Gaussian-smoothed} image with the closest scale, 

\[
m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}
\]

\[
\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1))/(L(x + 1, y) - L(x - 1, y)))
\]

Detection process returns 

\[
\{x, y, \sigma, \theta\}
\]

location  scale  orientation
4. Keypoint descriptor

**Image Gradients**
(4 x 4 pixel per cell, 4 x 4 cells)

- Gaussian weighting
  (sigma = half width)

**SIFT descriptor**
(16 cells x 8 directions = 128 dims)
Discriminative power

- Raw pixels
- Sampled
- Locally orderless
- Global histogram

Generalization power
References

Basic reading:
• Szeliski textbook, Sections 4.1.2, 14.1.2.