Designing descriptors
Overview of today’s lecture

- Why do we need feature descriptors?
- Designing feature descriptors.
- MOPS descriptor.
- GIST descriptor.
- Histogram of Textons descriptor.
- HOG descriptor.
- SURF descriptor.
- SIFT.
Why do we need feature descriptors?
If we know where the good features are, how do we match them?
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

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

vector of intensity values

What are the problems?
Image patch

Just use the pixel values of the patch

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

```
1 2 3
4 5 6
7 8 9
```

vector of x derivatives

‘binary descriptor’

Feature is invariant to absolute intensity values

What are the problems?
Image gradients

Use pixel differences

```
1 2 3
4 5 6
7 8 9
```

```
- + + - - +
```

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)

International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517
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)

International Conference on Computer Vision and Pattern Recognition (CVPR2005). pages 510-517

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
Haar wavelet responses can be computed with filtering.
Haar wavelet responses can be computed with filtering.
Haar wavelet responses can be computed with filtering.
Haar wavelet responses can be computed with filtering efficiently (in constant time) with integral images.
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)
GIST descriptor
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
Gabor Filters
(1D examples)

\[ e^{-\frac{x^2}{2\sigma^2}} \sin (2\pi \omega x) \]

\[ e^{-\frac{x^2}{2\sigma^2}} \cos (2\pi \omega x) \]
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

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

\[ \approx G^x_\sigma \quad \approx G^{xx}_\sigma \]
Directional edge detectors
What is the GIST descriptor encoding?
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?

Rough spatial distribution of image gradients
HOG descriptor
Dalal, Triggs. **Histograms of Oriented Gradients** for Human Detection. CVPR, 2005

- **Block** (2x2 cells)
- **Cell** (8x8 pixels)

- 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/

15 x 7 x 4 x 9 x 4 = 3780
(last x4 is block normalization)
SURF descriptor
SURF
(‘Speeded’ Up Robust Features)

Compute Haar wavelet response at each pixel in patch
SURF
(‘Speeded’ Up Robust Features)

4 x 4 cell grid

Each cell is represented by 4 values:

\[
\begin{bmatrix}
\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|
\end{bmatrix}
\]

Haar wavelets filters
(Gaussian weighted from center)

5 x 5 sample points

How big is the SURF descriptor?
SURF
(‘Speeded’ Up Robust Features)

Each cell is represented by 4 values:

$$\left[ \sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right]$$

Haar wavelets filters
(Gaussian weighted from center)

How big is the SURF descriptor?

64 dimensions
Integral Image

<table>
<thead>
<tr>
<th>$I(x, y)$</th>
<th>$A(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 5 2</td>
<td>1 6 8</td>
</tr>
<tr>
<td>2 4 1</td>
<td>3 12 15</td>
</tr>
<tr>
<td>2 1 1</td>
<td>5 15 19</td>
</tr>
</tbody>
</table>

$$A(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y')$$
Integral Image

original image

\[
I(x, y) = \begin{bmatrix}
1 & 5 & 2 \\
2 & 4 & 1 \\
2 & 1 & 1
\end{bmatrix}
\]

integral image

\[
A(x, y) = \begin{bmatrix}
1 & 6 & 8 \\
3 & 12 & 15 \\
5 & 15 & 19
\end{bmatrix}
\]

\[
A(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y')
\]

Can find the **sum** of any block using 3 operations

\[
A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1)
\]
What is the sum of the bottom right 2x2 square?

\[ A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1) \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

image

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>19</td>
<td></td>
</tr>
</tbody>
</table>

integral image

\[
A(1, 1, 3, 3) = A(3, 3) - A(1, 3) - A(3, 1) + A(1, 1) \\
= 19 - 8 - 5 + 1 \\
= 7
\]
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
4. Keypoint descriptor

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

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

Gaussian weighting
(sigma = half width)
Histogram of Textons descriptor
**Textons**

Julesz. Textons, the elements of texture perception, and their interactions. Nature 1981

*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

- Training image
- Filter Responses
- Texton Map
- Histogram of textons in image
- ‘encoding’
- ‘pooling’

Histogram of Textons descriptor

- Training image
- Filter Responses
- Texton Map
- Histogram of textons in image
- ‘encoding’
- ‘pooling’
Learning Textons from data

Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Dictionary / Thesaurus

Texton Dictionary

Learning Textons from data

Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Filter

patches

Dictionary

Texton Dictionary
Example of Filter Banks

Isotropic Gabor

Gaussian derivatives at different scales and orientations

'S'

'LM'

'MR8'
Learning Textons from data

Multiple training images of the same texture

Filter response over a bank of filters

Clustering

Texton Dictionary
Universal texton dictionary

histogram

References

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