midterm review

16-385 Computer Vision
Carnegie Mellon University (Kris Kitani)
Filtering and convolution
\[ h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l] \]
Filtering vs Convolution

**Filtering**
\[ h = g \otimes f \]
(cross-correlation)

**Convolution**
\[ h = g \ast f \]

**Filter**
\[ h[m, n] = \sum_{k,j} g[k, l] f[m + k, n + l] \]

**Image**
\[ h[m, n] = \sum_{k,j} g[k, l] f[m - k, n - l] \]

What’s the difference?
Filtering vs Convolution

filtering  \( h = g \otimes f \)  
\( (\text{cross-correlation}) \)

convolution  \( h = g * f \)  

output  \( h[m, n] = \sum_{k,j} g[k, l] f[m + k, n + l] \)

filter flipped vertically and horizontally

image  \( h[m, n] = \sum_{k,j} g[k, l] f[m - k, n - l] \)
Filtering vs Convolution

**filtering**  
\( h = g \otimes f \)  
(cross-correlation)

**convolution**  
\( h = g \ast f \)  

### Output
\[ h[m, n] = \sum_{k, j} g[k, l] f[m + k, n + l] \]

### Filter
\[ \text{filter flipped vertically and horizontally} \]

### Image
\[ h[m, n] = \sum_{k, j} g[k, l] f[m - k, n - l] \]

Suppose \( g \) is a Gaussian filter. How does convolution differ from filtering?

Recall...
Match the image to the Fourier magnitude image:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

- 1: Square with a dot in the center.
- 2: Colorful gradient.
- 3: Two intersecting circles.
- 4: Spokes-like pattern.
- 5: Star-like pattern.

- a: Black and white pattern.
- b: White flowers.
- c: Dark image with bright lights.
- d: Dark spot.
- e: Beach scene with people.
Hough Transform
Image and parameter space

\[ y = mx + b \]

Variables:
- \( y \)
- \( x \)
- \( m \)
- \( b \)

Parameters:
- \( m \)
- \( b \)

A point becomes a line in the parameter space.

Image space

Parameter space
Problems with parameterization

What’s wrong with the parameterization \((m, c)\)?

How big does the accumulator need to be?

\[ A(m, c) \]
Problems with parameterization

What’s wrong with the parameterization \((m, c)\) ?

How big does the accumulator need to be?

\[
A(m, c)
\]

The space of \(m\) is huge! The space of \(c\) is huge!

\[-\infty \leq m \leq \infty \quad \text{and} \quad -\infty \leq c \leq \infty\]
Image and parameter space

\[ y = mx + b \]

variables

\[ x \cos \theta + y \sin \theta = \rho \]

parameters

(a point becomes a line)

Image space

Parameter space
Image and parameter space

\[ y = mx + b \]

variables

parameters

Image space

Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

variables

parameters

Image space

Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

variables

parameters

Image space

Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

variables

parameters

Image space

Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

Variables\n
Parameters

Image space\n
Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

variables

parameters

two points become?
Image Pyramids
Constructing a Gaussian Pyramid

repeat
  filter
  subsample
until min resolution reached

Whole pyramid is only 4/3 the size of the original image!
Constructing the Laplacian Pyramid

do( i = 0 : nScales-1 )
{  
  blur(fi)
  subSamp2(li)
  hi = fi-li
}

Feature Detector & Descriptor
Easily recognized by looking through a small window

Shifting the window should give large change in intensity

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

[Moravec 1980]
Error function approximation

Change of intensity for the shift \([u,v]\):

\[
E(u, v) = \sum_{x,y} w(x, y) \left[ I(x + u, y + v) - I(x, y) \right]^2
\]

Second-order Taylor expansion of \(E(u,v)\) about \((0,0)\)
(bilinear approximation for small shifts):

\[
E(u, v) \approx E(0,0) + [u \ v] \begin{bmatrix} E_u(0,0) \\ E_v(0,0) \end{bmatrix} + \frac{1}{2} [u \ v] \begin{bmatrix} E_{uu}(0,0) & E_{uv}(0,0) \\ E_{uv}(0,0) & E_{vv}(0,0) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}
\]

first derivative  
second derivative
Visualization of a quadratic

The surface $E(u,v)$ is locally approximated by a quadratic form

$$E(u,v) \approx [u \ v] \ M \ [u \ v]$$

$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$
interpreting eigenvalues

What kind of image patch does each region represent?

\[ \lambda_2 \gg \lambda_1 \]

\[ \lambda_1 \approx 0 \]
\[ \lambda_2 \approx 0 \]

\[ \lambda_1 \gg \lambda_2 \]
interpreting eigenvalues

\[ \lambda_2 \gg \lambda_1 \]

\[ \lambda_1 \sim \lambda_2 \]

\[ \lambda_1 \gg \lambda_2 \]

horizontal edge

corner

flat

vertical edge
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)
Scale-space extrema

Selected if larger than all 26 neighbors

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

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

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

1. Compute gradients over image patch
2. Compute histogram over gradients
3. Take the histogram bin (orientation) with the greatest count
4. Rotate the image patch by that orientation
5. Compute descriptor
Bag-of-Words
Some local features are very informative.

An object as

A collection of local features (bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant
But what about layout?

All of these images have the same color histogram. Bag-of-words (histogram) representations remove spatial information.
Do you know how to build a visual vocabulary?
K-means Clustering

• Given k, the k-means algorithm consists of four steps:

  • Select initial centroids at random.

  • Assign each object to the cluster with the nearest centroid.

  • Compute each centroid as the mean of the objects assigned to it.

  • Repeat previous 2 steps until no change.
• How do you determine an optimal number of visual words?
Classification
Graphical model

$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$

$p(z)$

class
(random variable)

edge
(dependence relation)

$p(x_1 | z)$ $p(x_2 | z)$ $p(x_3 | z)$ $p(x_4 | z)$

$\mathbf{z}$
Each $x$ is an observed feature (e.g., visual words).

This is called the posterior: the probability of a class $z$ given the observed features $X$.

$$p(z | x_1, \ldots, x_N)$$

For classification, $z$ is a discrete random variable (e.g., car, person, building).

Each $x$ is an observed feature (e.g., visual words).

(it's a function that returns a single probability value)
The posterior can be decomposed according to **Bayes’ Rule**

\[
p(A|B) = \frac{p(B|A)p(A)}{p(B)}
\]

In our context...

\[
p(z|x_1, \ldots, x_N) = \frac{p(x_1, \ldots, x_N|z)p(z)}{p(x_1, \ldots, x_N)}
\]
The naive Bayes’ classifier is solving this optimization

\[ \hat{z} = \arg \max_{z \in \mathcal{Z}} p(z | X) \]

MAP (maximum a posteriori) estimate

\[ \hat{z} = \arg \max_{z \in \mathcal{Z}} \frac{p(X|z)p(z)}{p(X)} \]  

Bayes’ Rule

\[ \hat{z} = \arg \max_{z \in \mathcal{Z}} p(X|z)p(z) \]  

Remove constants
A naive Bayes’ classifier assumes all features are \textit{conditionally independent}

\[
p(x_1, \ldots, x_N | z) = p(x_1 | z)p(x_2, \ldots, x_N | z) \\
= p(x_1 | z)p(x_2 | z)p(x_3, \ldots, x_N | z) \\
= p(x_1 | z)p(x_2 | z) \cdots p(x_N | z)
\]
Find hyperplane $\mathbf{w}$ such that …

the gap between parallel hyperplanes \( \frac{2}{\|\mathbf{w}\|} \) is maximized
Can be formulated as a maximization problem

\[
\max_{\mathbf{w}} \frac{2}{\|\mathbf{w}\|}
\]

subject to \( \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \) if \( y_i = +1 \) for \( i = 1, \ldots, N \)

\[
\leq -1 \quad \text{if} \quad y_i = -1
\]

Equivalently,

\[
\min_{\mathbf{w}} \|\mathbf{w}\|
\]

subject to \( y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \leq 0 \) for \( i = 1, \ldots, N \)
‘soft’ margin

\[ \min_{w, \xi} \|w\|^2 + C \sum_{i} \xi_i \]

subject to

\[ y_i(w^T x_i + b) \geq 1 - \xi_i \]

for \( i = 1, \ldots, N \)

- Every constraint can be satisfied if slack is large
- \( C \) is a regularization parameter
  - Small \( C \): ignore constraints (larger margin)
  - Big \( C \): constraints (small margin)
- Still QP problem (unique solution)