Lecture 10

Detectors and descriptors

- Properties of detectors
  - Edge detectors
  - Harris
  - DoG

- Properties of descriptors
  - SIFT
  - HOG
  - Shape context
From the 3D to 2D & vice versa

Let's now focus on 2D
How to represent images?

Feature Detection

e.g. DoG
How to represent images?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection</td>
<td>e.g. DoG</td>
</tr>
<tr>
<td>Feature Description</td>
<td>e.g. SIFT</td>
</tr>
<tr>
<td>• Estimation</td>
<td></td>
</tr>
<tr>
<td>• Matching</td>
<td></td>
</tr>
<tr>
<td>• Indexing</td>
<td></td>
</tr>
<tr>
<td>• Detection</td>
<td></td>
</tr>
</tbody>
</table>
Estimation

Courtesy of TKK Automation Technology Laboratory
Estimation
Estimation
Matching

Image 1

Image 2

H
Object modeling and detection
Lecture 10
Detectors and descriptors

- Properties of detectors
  - Edge detectors
  - Harris
  - DoG

- Properties of descriptors
  - SIFT
  - HOG
  - Shape context
Edge detection
What causes an edge?

Identifies sudden changes in an image

- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., highlights; shadows)
Example of edge detection
Edge Detection

• Criteria for *optimal edge detection* (Canny 86):
  
  – **Good detection accuracy**:  
    • minimize the probability of false positives (detecting spurious edges caused by noise),  
    • false negatives (missing real edges)
  
  – **Good localization**:  
    • edges must be detected as close as possible to the true edges.
  
  – **Single response constraint**:  
    • minimize the number of local maxima around the true edge  
      (i.e. detector must return single point for each true edge point)
Edge Detection

• Examples:

- True edge
- Poor robustness to noise
- Poor localization
- Too many responses
Designing an edge detector

• **Two ingredients:**

• Use derivatives (in x and y direction) to define a location with high gradient.

• Need *smoothing* to reduce noise prior to taking derivative.
Designing an edge detector

\[
\frac{d}{dx} (f \ast g)
\]

[Eq. 1]

\[
= \frac{dg}{dx} \ast f = \text{"derivative of Gaussian" filter}
\]

[Eq. 2]

See CS231A, lecture 4 for details on convolution and linear filters.
Edge detector in 2D

• Smoothing

\[ I' = g(x, y) \ast I \]  \hspace{1cm} \text{[Eq. 3]}

\[ g(x, y) = \frac{1}{2\pi \sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \] \hspace{1cm} \text{[Eq. 4]}

• Derivative

\[ S = \nabla(g \ast I) = (\nabla g) \ast I = \]

\[ = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \ast I = \begin{bmatrix} g_x \ast I \\ g_y \ast I \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \text{gradient vector} \] \hspace{1cm} \text{[Eq. 5]}

\[ \nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \] \hspace{1cm} \text{[Eq. 6]}
Canny Edge Detection (Canny 86):

See CS131A for details

- The choice of $\sigma$ depends on desired behavior
  - large $\sigma$ detects large scale edges
  - small $\sigma$ detects fine features
Other edge detectors:

- Sobel
- Canny-Deriche
- Differential
Corner/blob detectors
Corner/blob detectors

- **Repeatability**
  - The same feature can be found in several images despite geometric and photometric transformations

- **Saliency**
  - Each feature is found at an “interesting” region of the image

- **Locality**
  - A feature occupies a “relatively small” area of the image;
Repeatability

Illumination invariance

Scale invariance

Pose invariance
- Rotation
- Affine
• Saliency

• Locality
Harris corner detector


See CS131A for details
Harris Detector: Basic Idea

Explore intensity changes within a window as the window changes location.

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions
Results
Harris corner doesn’t tell us the scale of the corner!
Blob detectors
Edge detection

Source: S. Seitz
Edge detection

\[ \frac{d^2}{dx^2} (f \ast g) \]

\[ f \ast \frac{d^2}{dx^2} g \]

= “second derivative of Gaussian” filter = Laplacian of the gaussian
Edge detection as zero crossing

Edge = zero crossing of the second derivative

**[Eq. 8]**
Edge detection as zero crossing

\[
\begin{array}{c}
\text{Kernel} \\
\begin{array}{c}
0 \\
0 \\
0
\end{array}
\end{array}
\]

= 

\[
\begin{array}{c}
\text{edge} \\
\text{edge}
\end{array}
\]
From edges to blobs

- Can we use the laplacian to find a blob (RECT function)?

Magnitude of the Laplacian response achieves a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob.
From edges to blobs

- Can we use the laplacian to find a blob (RECT function)?

What if the blob is slightly thicker or slimmer?
Scale selection

Convolve signal with Laplacians at several scales and looking for the maximum response. How in increase the scale??

By increasing $\sigma$
Scale normalization

- To keep the energy of the response the same, must multiply Gaussian kernel by $\sigma$
- Laplacian is the second Gaussian derivative, so it must be multiplied by $\sigma^2$

$$g(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$\sigma^2 \frac{d^2}{dx^2} g_n$$
The characteristic scale is the scale that produces peak of Laplacian response. This procedure allows us to:

1) detect the blob
2) estimate the size of the blob!

Characteristic scale

Here is what happens if we don’t normalize the Laplacian:

This should give the max response 😞
Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

\[ \nabla^2 g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) = \sigma^2 (g_{xx} + g_{yy}) \]

[Eq. 9]
Scale selection

- For a binary circle of radius $r$, the Laplacian achieves a maximum at $\sigma = r / \sqrt{2}$
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales

2. Find maxima of squared Laplacian response in scale-space

The maxima indicate that a blob has been detected and what’s its intrinsic scale
Scale-space blob detector: Example
Scale-space blob detector: Example

sigma = 11.9912
Scale-space blob detector: Example
**Difference of Gaussians (DoG)**

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), 04

- Approximating the Laplacian with a difference of Gaussians:

\[
L = \sigma^2 \left( g_{xx}(x, y, \sigma) + g_{yy}(x, y, \sigma) \right)
\]

(Laplacian) \[\text{[Eq. 10]}\]

\[
DoG = g(x, y, 2\sigma) - g(x, y, \sigma)
\]

Difference of gaussian with scales 2 \(\sigma\) and \(\sigma\) \[\text{[Eq. 11]}\]

In general:

\[
DoG = g(x, y, k\sigma) - g(x, y, \sigma) \approx (k - 1)\sigma^2L
\]

[Eq. 12]
Affine invariant detectors


Similarly to characteristic scale, we can define the characteristic shape of a blob
### Properties of detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>View point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowe ‘99 (DoG)</td>
<td>Yes*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ f \rightarrow f + b \]
Properties of detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>Viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowe ’99 (DoG)</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

\[ f \rightarrow f + b \]
## Properties of detectors

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>View point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowe ’99 (DoG)</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Harris corner</td>
<td>Yes*</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Mikolajczyk &amp; Schmid ’01, ’02</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tuytelaars, ’00</td>
<td>Yes*</td>
<td>Yes</td>
<td>No (Yes ’04 )</td>
<td>Yes</td>
</tr>
<tr>
<td>Kadir &amp; Brady, 01</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes</td>
<td>no</td>
</tr>
<tr>
<td>Matas, ’02</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes</td>
<td>no</td>
</tr>
</tbody>
</table>
Lecture 10

Detectors and descriptors

- Properties of detectors
  - Edge detectors
    - Harris
    - DoG

- Properties of descriptors
  - SIFT
  - HOG
  - Shape context
The big picture…

Feature Detection
- e.g. DoG

Feature Description
- e.g. SIFT
- Estimation
- Matching
- Indexing
- Detection
Properties

Depending on the application a descriptor must incorporate information that is:

• Invariant w.r.t:
  • Illumination
  • Pose
  • Scale
  • Intraclass variability

• Highly distinctive (allows a single feature to find its correct match with good probability in a large database of features)
The simplest descriptor...
Normalized vector of intensities

\[ w = \begin{bmatrix} \vdots \end{bmatrix} \]

\[ \begin{bmatrix} \vdots \end{bmatrix} \]

1 x NM vector of pixel intensities

\[
W_n = \frac{(w - \bar{w})}{\| (w - \bar{w}) \|}
\]

Makes the descriptor invariant with respect to affine transformation of the illumination condition

[Eq. 13]
Illumination normalization

• **Affine intensity** change:

\[ w \rightarrow w + b \quad \text{[Eq. 14]} \]
\[ \rightarrow a \, w + b \]

\[ w_n = \frac{(w - \bar{w})}{\|(w - \bar{w})\|} \]

• Make each patch zero mean: remove \( b \)
• Make unit variance: remove \( a \)
Why can’t we just use this?

- Sensitive to small variation of:
  - location
  - Pose
  - Scale
  - intra-class variability

- Poorly distinctive
Sensitive to pose variations

Normalized Correlation:

\[ w_n \cdot w'_n = \frac{(w - \overline{w})(w' - \overline{w}')}{{\|}(w - \overline{w})(w' - \overline{w}'){\|}} \]
## Properties of descriptors

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>Good</td>
<td>Poor</td>
<td>Poor</td>
</tr>
</tbody>
</table>
Bank of filters

More robust but still quite sensitive to pose variations

## Properties of descriptors

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>Good</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>FILTERS</td>
<td>Good</td>
<td>Medium</td>
<td>Medium</td>
</tr>
</tbody>
</table>
SIFT descriptor

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), 04

- Alternative representation for image regions
- Location and characteristic scale $s$ given by DoG detector
- Compute gradient at each pixel
SIFT descriptor

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), 04

- Alternative representation for image regions
- Location and characteristic scale $s$ given by DoG detector

1. Compute gradient at each pixel
2. $N \times N$ spatial bins
3. Compute an histogram $h_i$ of $M$ orientations for each bin $i$
SIFT descriptor

David G. Lowe. "Distinctive image features from scale-invariant keypoints." IJCV 60 (2), 04

• Alternative representation for image regions
• Location and characteristic scale s given by DoG detector

1 Compute gradient at each pixel
2 N x N spatial bins
3 Compute an histogram $h_i$ of M orientations for each bin i
4 Concatenate $h_i$ for $i=1$ to $N^2$ to form a $1 \times MN^2$ vector $H$
5 Gaussian center-weighting
6 Normalize to unit norm

Typically $M = 8$; $N = 4$
$H = 1 \times 128$ descriptor
Rotational invariance

- Find dominant orientation by building an orientation histogram
- Rotate all orientations by the dominant orientation

This makes the SIFT descriptor rotational invariant
Properties of descriptors

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>Good</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>FILTERS</td>
<td>Good</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>SIFT</td>
<td>Good</td>
<td>Good</td>
<td>Medium</td>
</tr>
</tbody>
</table>

- SIFT is robust w.r.t. small variation in:
  - Illumination (thanks to gradient & normalization)
  - Pose (small affine variation thanks to orientation histogram )
  - Scale (scale is fixed by DOG)
  - Intra-class variability (small variations thanks to histograms)
HoG = Histogram of Oriented Gradients

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

• Like SIFT, but…
  – Sampled on a dense, regular grid around the object
  – Gradients are contrast normalized in overlapping blocks
Shape context descriptor

Belongie et al. 2002

Histogram (occurrences within each bin)

13\textsuperscript{th}
Shape context descriptor

descriptor 1

descriptor 2

descriptor 3
Other detectors/descriptors

- **HOG: Histogram of oriented gradients**
  Dalal & Triggs, 2005

- **SURF: Speeded Up Robust Features**

- **FAST (corner detector)**

- **ORB: an efficient alternative to SIFT or SURF**
  Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. ICCV 2011

- **Fast Retina Key- point (FREAK)**
Using CNNs to detect and describe features

Layer 1

Layer 2

Layer 3

[Zeiler & Fergus ECCV 14]
Object detection using CNN features!

Next lecture:

Introduction to recognition

We’ll run a “Small Group” Feedback Session by the office of Teaching and Learning from 4-4:20pm
Pose normalization

View 1

View 2

Scale, rotation & sheer
Pose normalization

- Keypoints are transformed in order to be invariant to translation, rotation, scale, and other geometrical parameters [Lowe 2000]

Change of scale, pose, illumination…
Pose normalization

NOTE: location, scale, rotation & affine pose are given by the detector or calculated within the detected regions.