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

\[ P = [x, y, z] \]

\[ p = [x, y] \]

• Let’s now focus on 2D
How to represent images?
The big picture…

Feature Detection
- e.g. DoG

Feature Description
- e.g. SIFT

- Estimation
- Matching
- Indexing
- Detection
Estimation
Estimation
Object modeling and detection
• 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)
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

\[
\frac{d}{dx}(f \ast g) = \frac{dg}{dx} \ast f = "\text{derivative of Gaussian" filter.}
\]

Source: S. Seitz
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

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

\[ = \begin{bmatrix} g_x \\ g_y \end{bmatrix} \ast I = \begin{bmatrix} g_x \ast I \\ g_y \ast I \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \text{gradient vector} \]
Canny Edge Detection \textsuperscript{(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
Corners detectors
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
Blob detectors
Edge detection

\[ f \]

\[ g \]

\[ f * g \]

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

Source: S. Seitz
Edge detection

[f \ast g] = \text{“second derivative of Gaussian” filter} = \text{Laplacian of the gaussian}

[Eq. 7] \frac{d^2}{dx^2} (f \ast g)

[Eq. 8] f \ast \frac{d^2}{dx^2} g
Edge detection as zero crossing

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

[Eq. 8]

Edge = zero crossing of second derivative

\[ \frac{d^2}{dx^2} g \]
Edge detection as zero crossing

![Diagram of edge detection as zero crossing]
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 sizes and looking for the maximum response

increasing $\sigma$
Scale normalization

- To keep the energy of the response the same, must multiply Gaussian derivative by $\sigma$

- Laplacian is the second Gaussian derivative, so it must be multiplied by $\sigma^2$

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

\[ \sigma^2 \frac{d^2}{dx^2} g \]
Characteristic scale

We define the characteristic scale as the scale that produces peak of Laplacian response

Characteristic scale

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

Original signal

$\sigma = 1$  $\sigma = 2$  $\sigma = 4$  $\sigma = 8$  $\sigma = 16$

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

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

Scale-normalized:

\[ \nabla_{\text{norm}}^2 g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \]  

[Eq. 9]
Scale selection

- For a binary circle of radius $r$, the Laplacian achieves a maximum at

$$\sigma = \frac{r}{\sqrt{2}}$$
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
Approximating the Laplacian with a difference of Gaussians:

\[ L = \sigma^2 \left( G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right) \]  
(Laplacian)  
[Eq. 10]

\[ DoG = G(x, y, 2\sigma) - G(x, y, \sigma) \]  
Difference of gaussian with scales 2 \( \sigma \) and \( \sigma \)  
[Eq. 11]

In general:

\[ DoG = G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 L \]  
[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>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Scale-normalized: \( \nabla^2_{\text{norm}} g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \)
## 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</td>
<td>(Yes ’04)</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

1 x NM vector of pixel intensities

\[ w = [ \quad \ldots \quad ] \]
Normalized vector of intensities

\[ w = \begin{bmatrix} 1 \times NM \text{ vector of pixel intensities} \end{bmatrix} \]

\[ W_n = \frac{(w - \bar{w})}{\| (w - \bar{w}) \|} \quad \text{Makes the descriptor invariant with respect to affine transformation of the illumination condition} \]

[Eq. 13]
Illumination normalization

- **Affine intensity change:**
  \[ w \rightarrow w + b \]
  \[ \rightarrow a\ w + b \]  
  \[[\text{Eq. 14}]\]
  \[ w_n = \frac{(w - \bar{w})}{\| (w - \bar{w}) \|} \]

- 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:

\[ w_n \cdot w'_n = \frac{(w - \bar{w})(w' - \bar{w}')}{\| (w - \bar{w})(w' - \bar{w}') \|} \]
# 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>
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

Image window
SIFT descriptor

- Alternative representation for image regions
- Location and characteristic scale \( s \) given by DoG detector

- Compute gradient at each pixel
- \( N \times N \) spatial bins
- Compute an histogram \( h_i \) of \( M \) orientations for each bin \( i \)
SIFT descriptor

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

- Compute gradient at each pixel
- $N \times N$ spatial bins
- Compute an histogram $h_i$ of $M$ orientations for each bin $i$
- Concatenate $h_i$ for $i=1$ to $N^2$ to form a $1 \times MN^2$ vector $H$
SIFT descriptor

- Alternative representation for image regions
- Location and characteristic scale \( s \) given by DoG detector

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

Typically \( M = 8; N = 4 \)
\( H = 1 \times 128 \) descriptor
SIFT descriptor

• 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)
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>
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)
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)**
Next lecture:

Image Classification by Deep Networks by Ronan Collobert (Facebook)