EECS 442 – Computer vision

Detectors part II
Descriptors

• Blob detectors
• Invariance
• Descriptors

Some slides of this lectures are courtesy of prof F. Li, prof S. Lazebnik, and various other lecturers
Goal:
Identify interesting regions from the images (edges, corners, blobs…)

Descriptors

Matching / Indexing / Recognition
e.g. SIFT
• **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 Detector
Invariance

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>View point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris corner</td>
<td>partial</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Extract useful building blocks: blobs
Edge detection

$f$

$\frac{d}{dx} g$

$f \ast \frac{d}{dx} g$

Source: S. Seitz
Edge detection as zero crossing

Edge = zero crossing of second derivative of Gaussian (Laplacian)

Source: S. Seitz
Edge detection as zero crossing

![Edge detection graphs](image)
From edges to blobs

- Blob = superposition of nearby edges

Ok, great, but what if the blob is slightly thicker or slimmer?
From edges to blobs

• Blob = superposition of nearby edges

Spatial selection: magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob.
Scale selection

• We want to find the characteristic scale of the blob by convolving it with Laplacians at several scales and looking for the maximum response.

• However, Laplacian response decays as scale increases:

Why does this happen?
Scale normalization

- The response of a derivative of Gaussian filter to a perfect step edge decreases as $\sigma$ increases.
Scale normalization

- To keep response the same (scale-invariant), must multiply Gaussian derivative by \( \sigma \)
- Laplacian is the second Gaussian derivative, so it must be multiplied by \( \sigma^2 \)
Effect of scale normalization

Original signal

Unnormalized Laplacian response

Scale-normalized Laplacian response

Maximum 😊
Blob detection in 2D

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D

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

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

- For a binary circle of radius $r$, the Laplacian achieves a maximum at $\sigma = \frac{r}{\sqrt{2}}$. 

![Diagram of an image with a circle and a graph showing the Laplacian response in scale ($\sigma$)]
Characteristic scale

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

Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space
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)

\[ \text{DoG} = G(x, y, k\sigma) - G(x, y, \sigma) \]  

(Difference of Gaussians)

\[ G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 L \]
Output: location, scale, orientation (more later)
Example of keypoint detection
## Invariance

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>Viewpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris corner</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lowe '99 (DoG)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Harris-Laplace
[Mikolajczyk & Schmid ’01]

• Collect locations \((x,y)\) of detected Harris features for \(\sigma = \sigma_1 \ldots \sigma_2\) (the sigma is here comes from \(g_x, g_y\))

• For each detected location \((x,y)\) and for each \(\sigma\), reject detection if Laplacian\((x,y, \sigma)\) is not a local maximum

Output: location, scale
## Invariance

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>View point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris corner</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lowe '99 (DoG)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Mikolajczyk &amp; Schmid '01</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Repeatability

Illumination invariance

Scale invariance

Pose invariance
- Rotation
- Affine
Affine invariance


Similarly to characteristic scale selection, detect the **characteristic shape** of the local feature
**Affine adaptation**

\[
M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R
\]

We can visualize \( M \) as an ellipse with axis lengths determined by the eigenvalues and orientation determined by \( R \).

**Ellipse equation:**

\[
[u \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix} = \text{const}
\]

The second moment ellipse can be viewed as the “characteristic shape” of a region.
Affine adaptation

1. Detect initial region with Harris Laplace
2. Estimate affine shape with M
3. Normalize the affine region to a circular one
4. Re-detect the new location and scale in the normalized image
5. Go to step 2 if the eigenvalues of the M for the new point are not equal [detector not yet adapted to the characteristic shape]
Affine adaptation

Output: location, scale, affine shape, rotation (more later)
Affine adaptation example

Scale-invariant regions (blobs)
Affine adaptation example

Affine-adapted blobs
## Invariance

<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Rotation</th>
<th>Scale</th>
<th>View point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris corner</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lowe '99 (DoG)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Mikolajczyk &amp; Schmid '01</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Mikolajczyk &amp; Schmid '02</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Detector</td>
<td>Illumination</td>
<td>Rotation</td>
<td>Scale</td>
<td>Viewpoint</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>Harris corner</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Lowe ’99 (DoG)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</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>
• Blob detectors
• Invariance
• Descriptors
Goal:
Identify interesting regions from the images (edges, corners, blobs…)

Descriptors

Matching / Indexing / Recognition

E.g. SIFT
Matching Features
(stitching images)

• Detect feature points in both images
Matching Features
(stitching images)

- Detect feature points in both images
- Find corresponding pairs
Matching Features
(stitching images)

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images
Matching Features
(estimating F)

• Detect feature points in both images
• Find corresponding pairs
• Use these pairs to estimate F
Matching Features
(recognizing objects)

• Detect feature points in both images
• Find corresponding pairs
• Use these pairs to match different object instances
Challenges

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)
Illumination normalization

- **Affine intensity change:**
  \[ I \rightarrow I + b \]
  \[ \rightarrow a I + b \]

- Make each patch zero mean:
  \[ \mu = \frac{1}{N} \sum_{x,y} I(x, y) \]
  \[ Z(x, y) = I(x, y) - \mu \]

- Then make unit variance:
  \[ \sigma^2 = \frac{1}{N} \sum_{x,y} Z(x, y)^2 \]
  \[ ZN(x, y) = \frac{Z(x, y)}{\sigma} \]
Pose normalization

- Keypoints are transformed in order to be invariant to translation, rotation, scale, and other geometrical parameters.

Change of scale, pose, illumination…

Courtesy of D. Lowe
Pose normalization

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

View 1

View 2

Scale, rotation & shear
The simplest descriptor

1 x NM vector of pixel intensities

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

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

Makes the descriptor invariant with respect to affine transformation of the illumination condition
Why can’t we just use this?

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

• Poorly distinctive
Stereo systems

Normalized Correlation:

\[ w_n \cdot w'_n = \frac{(w - \bar{w})(w' - \bar{w}')}{{\left\| (w - \bar{w})(w' - \bar{w}') \right\|}} \]
<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>***</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Bank of filters

More robust but still quite sensitive to pose variations
Bank of filters - Steerable filters

Gaussian derivatives up to 4\textsuperscript{th} order. The remaining derivatives can be computed by rotation of 90 degrees.
<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>***</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>FILTERS</td>
<td>***</td>
<td>**</td>
<td>**</td>
</tr>
</tbody>
</table>
SIFT descriptor

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

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

![Image of SIFT descriptor](image-url)
SIFT descriptor

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

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

- Compute gradient at each pixel
- $N \times N$ spatial bins
- Compute an histogram of $M$ orientations for each bean
SIFT descriptor

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

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

- Compute gradient at each pixel
- $N \times N$ spatial bins
- Compute an histogram of $M$ orientations for each bean
- Gaussian center-weighting
SIFT descriptor

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

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

- Compute gradient at each pixel
- $N \times N$ spatial bins
- Compute an histogram of $M$ orientations for each bean
- Gaussian center-weighting
- Normalized unit norm

Typically $M = 8$; $N = 4$

$1 \times 128$ 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 smoothed orientation histogram
- Rotate all orientations by the dominant orientation

This makes the SIFT descriptor rotational invariant
Rotational invariance
Rotational invariance
Matching using SIFT

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

David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), 04
<table>
<thead>
<tr>
<th>Detector</th>
<th>Illumination</th>
<th>Pose</th>
<th>Intra-class variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATCH</td>
<td>***</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>FILTERS</td>
<td>***</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>SIFT</td>
<td>***</td>
<td>***</td>
<td>***</td>
</tr>
</tbody>
</table>
Shape context

Belongie et al. 2002
Shape Context

Count the number of points inside each bin, e.g.:

\[ \text{Count} = 4 \]
\[ \vdots \]
\[ \text{Count} = 10 \]

Compact representation of distribution of points relative to each point.
Matching different instances
EECS 442 – Computer vision

Next lecture:
Mid level representations
Other interest point detectors

Scale Saliency [Kadir & Brady '01, '03]
Other interest point detectors

Scale Saliency [Kadir & Brady ’01, ’03]

- Uses entropy measure of local pdf of intensities:

\[ H_D (s, x) = - \int_{d \in D} p(d, s, x) \log_2 p(d, s, x).dd \]

- Takes local maxima in scale

- Weights with ‘change’ of distribution with scale:

\[ W_D (s, x) = s \int_{d \in D} \left| \frac{\partial}{\partial s} p(d, s, x) \right|.dd \]

- To get saliency measure:

\[ Y_D (s, x) = H_D (s, x) \times W_D (s, x) \]
Other interest point detectors

Scale Saliency [Kadir & Brady ’01, ’03]

Just using $H_D(s, x)$

Using $Y_D(s, x) = H_D W_D$

Most salient parts detected
Other interest point detectors
maximum stable extremal regions [matas et al. 02]

- Sweep threshold of intensity from black to white
- Locate regions based on stability of region with respect to change of threshold
Creating features stable to viewpoint change

- Edelman, Intrator & Poggio (97) showed that complex cell outputs are better for 3D recognition than simple correlation.
Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise.

- Figure of 30
Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise and affine distortion.

- Figure 30
Geometric blur

Berg et al. 2001
Geometric blur