Local Feature Descriptors

SIFT

Various slides from previous courses by:
D.A. Forsyth (Berkeley / UIUC), I. Kokkinos (Ecole Centrale / UCL), S. Lazebnik (UNC / UIUC), S. Seitz (MSR / Facebook), J. Hays (Brown / Georgia Tech), A. Berg (Stony Brook / UNC), D. Samaras (Stony Brook), J. M. Frahm (UNC), V. Ordonez (UVA).
Last Class

- Corner Detection - Harris
- Interest Points
- Blob Detection
Today’s Class

• Interest Points (DoG extrema operator)
• SIFT Feature descriptor
• Feature matching
Corner Detection: Basic Idea

• We should easily recognize the point by looking through a small window
• Shifting a window in any direction should give a large change in intensity

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

Source: A. Efros
Harris Corner Detection

- Compute the following matrix of squared gradients for every pixel.

\[
M = \sum_{\text{patch}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}
\]

$I_x$ and $I_y$ are gradients computed using Sobel or some other approximation.

\[
M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad M = \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \quad M = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix} \quad M = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}
\]
\[ I_x I_y \]
If both \( a \) and \( b \) are large then this is a corner, otherwise it is not. Set a threshold and this should detect corners.

Problem: Doesn’t work for these corners:
Harris Corner Detection

\[ M = \sum_{\text{patch}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \]

Under a rotation \( M \) can be diagonalized

\[ M = R_m^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R_m \]

\( \lambda_1 \) and \( \lambda_2 \) are the eigenvalues of \( M \)

From your linear algebra class finding them requires solving

\[ \det(M - \lambda I) = 0 \]
However no need to solve $\det(M - \lambda I) = 0$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$
Corner response function

\[ R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \]

\( \alpha \): constant (0.04 to 0.06)
Harris Detector Summary [Harris88]

- Second moment matrix

\[ \mu(\sigma_I, \sigma_D) = g(\sigma_I) \star \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Image derivatives (optionally, blur first)

\[ \det M = \lambda_1 \lambda_2 \]
\[ \text{trace } M = \lambda_1 + \lambda_2 \]

2. Square of derivatives

3. Gaussian filter \( g(\sigma_i) \)

4. Cornerness function – both eigenvalues are strong

\[ har = \det[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))^2] = g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2 \]

5. Non-maxima suppression
Alternative Corner response function

“edge”:
\[ \lambda_1 \gg \lambda_2 \]
\[ \lambda_2 \gg \lambda_1 \]

“corner”:
\[ \lambda_1 \text{ and } \lambda_2 \text{ are large, } \lambda_1 \sim \lambda_2 ; \]

“flat” region
\[ \lambda_1 \text{ and } \lambda_2 \text{ are small;} \]

\[ f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \quad \text{Szeliski Harmonic mean} \]

\[ \det M = \lambda_1 \lambda_2 \]
\[ \text{trace } M = \lambda_1 + \lambda_2 \]
Harris Detector: Steps

Compute corner response $R$
Harris Detector: Steps

Find points with large corner response: $R > \text{threshold}$
Harris Detector: Steps

Take only the points of local maxima of $R$
Harris Detector: Steps
Invariance and covariance

• We want corner locations to be *invariant* to photometric transformations and *covariant* to geometric transformations
  • **Invariance**: image is transformed and corner locations do not change
  • **Covariance**: if we have two transformed versions of the same image, features should be detected in corresponding locations
Keypoint detection with scale selection

- We want to extract keypoints with characteristic scale that is *covariant* with the image transformation.
Basic idea

- Convolve the image with a “blob filter” at multiple scales and look for extrema of filter response in the resulting scale space.

Blob detection

- Find maxima and minima of blob filter response in space and scale

Source: N. Snavely
Blob filter

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

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
Scale-space blob detector: Example
Scale-space blob detector: Example

Slide by Svetlana Lazebnik
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
Eliminating edge responses

- Laplacian has strong response along edge
Approximating the Laplacian with a difference of Gaussians:

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

(Laplacian)

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

(Difference of Gaussians)
Efficient implementation...

Eliminating rotation ambiguity

- To assign a unique orientation to circular image windows:
  - Create histogram of local gradient directions in the patch
  - Assign canonical orientation at peak of smoothed histogram
SIFT keypoint detection

From keypoint detection to keypoint representation (feature descriptors)

Figure by Svetlana Lazebnik
SIFT descriptors

- Inspiration: complex neurons in the primary visual cortex

From keypoint detection to keypoint representation (feature descriptors)

Compare SIFT feature vectors instead
SIFT Feature Matching

Rice Hall at UVA
The method has been integrated into OpenCV library (see xfeatures2d in opencv_contrib).
A hard keypoint matching problem

NASA Mars Rover images
Answer below (look for tiny colored squares...)

NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely
Feature Descriptors Zoo

• SIFT (under a patent) Proposed around 1999
• SURF (under a patent too – I think)
• BRIEF
• ORB (seems free as it is OpenCV’s preferred)
• BRISK
• FREAK
• FAST
• KAZE
• LIFT (Most recently proposed at ECCV 2016)
<table>
<thead>
<tr>
<th>TITLE</th>
<th>CITED BY</th>
<th>YEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distinctive image features from scale-invariant keypoints</td>
<td>45496</td>
<td>2004</td>
</tr>
<tr>
<td>DG Lowe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>International journal of computer vision 60 (2), 91-110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object recognition from local scale-invariant features</td>
<td>14817</td>
<td>1999</td>
</tr>
<tr>
<td>DG Lowe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>International Conference on Computer Vision, 1999, 1150-1157</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?