Lecture 2: SIFT
Recap
Correspondence estimation

- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract features
Step 2: match features
The Harris corner detector

Compute second-moment matrix:

Sum over a small region around the hypothetical corner

Gradient with respect to x, times gradient with respect to y

\[ C = \begin{bmatrix}
    \sum I_x^2 & \sum I_x I_y \\
    \sum I_x I_y & \sum I_y^2
\end{bmatrix} \]

Matrix is symmetric

Slide credit: David Jacobs
Harris Detector: Workflow

Slide credit: http://vims.cis.udel.edu/~chandra/
Harris Detector: Workflow

Compute corner response $R$
The Harris corner detector

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

\[ R = \det(C) - k \times (\text{trace}(C))^2 \]

\[ \det(C) = \lambda_1 \lambda_2 \]

\[ \text{trace}(C) = \lambda_1 + \lambda_2 \]

- When \(|R|\) is small, which happens when \(\lambda_1\) and \(\lambda_2\) are small, the region is flat.
- When \(R < 0\), which happens when \(\lambda_1 >> \lambda_2\) or vice versa, the region is edge.
- When \(R\) is large, which happens when \(\lambda_1\) and \(\lambda_2\) are large and \(\lambda_1 \sim \lambda_2\), the region is a corner.
Harris Detector: Workflow

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

Slide credit: http://vims.cis.udel.edu/~chandra/
Harris Detector: Workflow

Take only the points of local maxima of $R$
Harris Detector: Workflow

Slide credit: http://vims.cis.udel.edu/~chandra/
Simple matching methods

- **SSD (Sum of Squared Differences)**
  \[ \sum_{x,y} |W_1(x, y) - W_2(x, y)|^2 \]

- **NCC (Normalized Cross Correlation)**
  \[ \frac{\sum_{x,y} (W_1(x, y) - \bar{W}_1)(W_2(x, y) - \bar{W}_2)}{\sigma_{W_1} \sigma_{W_2}} \]

  where
  \[ \bar{W}_i = \frac{1}{n} \sum_{x,y} W_i \]
  \[ \sigma_{W_i} = \sqrt{\frac{1}{n} \sum_{x,y} (W_i - \bar{W}_i)^2} \]

- What advantages might NCC have over SSD?
Desirable property: invariance

Find features that are invariant to transformations
- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, ...
**Scale Invariant Feature Transform**

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

Adapted from slide by David Lowe
The distance threshold affects performance

- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?
Choosing a match

- First approach: use $SSD = || f_2 - f_1 ||$

- Better approach: ratio distance $= \frac{|| f_2 - f_1 ||}{|| f_2' - f_1 ||}$
  - $f_2$ is best SSD match to $f_1$ in $I_2$
  - $f_2'$ is second best SSD match to $f_1$ in $I_2$
  - Gives large values (close to 1) for ambiguous matches.
How do we perceive depth in images?

Relate projections of the same point in two or more images of the scene.

Song and Chandraker, CVPR 2015

Newcombe et al., CVPR 2015
Active Correspondence
Structured Light Scanning
Structured Light Scanning
Structured Light Scanning

Camera  Projector

[Image of structured light scanning diagram]
Structured Light Scanning

Codeword of this pixel: 101....

Pattern 1
Pattern 2
Pattern 3

Projected over time

Slide courtesy: Narasimhan, CMU
Structured Light Scanning

Slide courtesy: Narasimhan, CMU
Microsoft Kinect works on similar principles

- Infra-red instead of visible light
- Random dot patterns instead of bar codes.

Now In Your Xbox!
More on SIFT

Slides adapted from David Lowe
Feature Matching
SIFT: Motivation

- The Harris operator is not invariant to scale and correlation is not invariant to rotation.

- For better image matching, need to develop an interest operator invariant to scale and rotation.

- Also, need a descriptor robust to typical variations. The descriptor is the most-used part of SIFT.
Idea of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.
Overall Procedure at a High Level

1. Scale-space extrema detection
   Search over multiple scales and image locations.

2. Keypoint localization
   Fit a model to determine location and scale.
   Select keypoints based on a measure of stability.

3. Orientation assignment
   Compute best orientation(s) for each keypoint region.

4. Keypoint description
   Use local image gradients at selected scale and rotation to describe each keypoint region.
1. Scale-space extrema detection

- **Goal:** Identify locations and scales that can be reliably assigned under different views of the same scene or object.

- **Method:** search for stable features across multiple scales using a continuous function of scale.

- **The scale space of an image is a function** $L(x,y,\sigma)$ **that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.**
Aside: Gaussian Smoothing

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

A 1D Gaussian with mean 0, variance 1

\[ G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

A 2D Gaussian with mean (0,0), variance 1
Aside: Gaussian Smoothing

A discrete approximation to Gaussian kernel with variance 1

\[
\begin{array}{ccccc}
1 & 4 & 7 & 4 & 1 \\
4 & 16 & 26 & 16 & 4 \\
7 & 26 & 41 & 26 & 7 \\
4 & 16 & 26 & 16 & 4 \\
1 & 4 & 7 & 4 & 1 \\
\end{array}
\]

\[
\begin{array}{ccccc}
1 \\
4 \\
7 \\
4 \\
1 \\
\end{array}
\]

Convolution
Convolution

Original image

Convolutional filter 1

Convolving the image

Result

Convolution operation:

\[ I(x, y) \ast h = \sum_{i=-a}^{a} \sum_{j=-b}^{b} I(x - i, y - j) \cdot h(i, j) \]
Convolution

Original image

Convolutional filter 1

Convolving the image

Result

$$I(x, y) * h = \sum_{i=-a}^{a} \sum_{j=-b}^{b} I(x - i, y - j) \cdot h(i, j)$$
Convolution

We will look at convolutions again as a basic operation for deep neural networks.
Example: Gaussian Smoothed Scale Space
Aside: Image Pyramids

- Bottom level is the original image.
  
- 2nd level is derived from the original image according to some function.
  
- 3rd level is derived from the 2nd level according to the same function.
  
And so on.
Aside: Gaussian Pyramid
At each level, image is smoothed and reduced in size.

At 2\textsuperscript{nd} level, each pixel is the result of applying a Gaussian mask to the first level and then subsampling to reduce the size.

Bottom level is the original image.

And so on.
Example: Subsampling with Gaussian Pre-filtering
Scale Space Pyramid

The parameter $s$ determines the number of images per octave.
Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space.

- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below.

For each max or min found, output is the location and the scale.
Difference of Gaussians

- Scale-space detection
  - Find local maxima across scale/space
  - A good “blob” detector

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

\[
G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.
\]
3. Orientation assignment

- Create histogram of local gradient directions at selected scale
- Assign canonical orientation at peak of smoothed histogram

If 2 major orientations, use both.
Keypoint localization with orientation

233x189 keypoints

832 keypoints
4. Keypoint Descriptors

- At this point, each keypoint has
  - location
  - scale
  - orientation

- Next is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive
  - as invariant as possible to variations such as changes in viewpoint and illumination
Normalization

- Rotate the window to standard orientation

- Scale the window size based on the scale at which the point was found.
SIFT Keypoint Descriptor
(shown with 2 X 2 descriptors over 8 X 8)

In implementation, 4x4 arrays of 8 bin histogram are used, a total of 128 features for one keypoint.
Uses for SIFT

- Feature points are used also for:
  - Panorama stitching
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … many others
Cases where SIFT does not work

- Strong illumination changes
- Large out-of-plane rotations
- Non-rigid deformations or articulations
- Semantic correspondence