Feature Extraction
(Part 2)

Image Processing
CSE 166
Lecture 18
Announcements

• Assignment 7 is due today, 11:59 PM
• Quiz 7 is Dec 11
• Final exam is Dec 14
• Course and Professor Evaluations (CAPE)
  – https://cape.ucsd.edu/
  – Must be completed before Dec 14, 8:00 AM
• Reading
  – Chapter 12: Feature extraction
    • Sections 12.7
Feature extraction

- Feature extraction is comprised of
  - Feature detection
  - Feature description
    - A feature descriptor is
      - **Invariant** with respect to a set of transformations if its value remains unchanged after the application of any transformation from the family
      - **Covariant** with respect to a set of transformations if applying any transformation from the set produces the same result in the descriptor
Features

• Features
  – Local (a member of a set)
  – Global (the entire set)

• Categories
  – Boundaries
  – Regions
  – Whole images
Scale invariant feature transform (SIFT)

- SIFT features are called *keypoints*
- Keypoints are invariant to
  - Scale
  - Rotation
- Keypoints are robust to
  - Changes in viewpoint
  - Changes in illumination
  - Noise
Scale invariant feature transform (SIFT)

• SIFT feature descriptors are $n$-dimensional feature vectors
  – Elements are invariant feature descriptors
Scale invariant feature transform (SIFT)

• Steps
  1. Construct the scale space
  2. Obtain the initial keypoints
  3. Improve the accuracy of the location of the keypoints
  4. Delete unsuitable keypoints
  5. Compute keypoint orientations
  6. Compute keypoint descriptors
SIFT, construct the scale space

• Search for stable features across all possible scales
  – Use a function of scale known as scale space
• Achieves scale invariance
Scale space

• Scale space theory is a formal theory for image structures at different scales
  – Image is represented by a one-parameter family of Gaussian low pass filtered images
  – A Gaussian filter meets all scale space axioms
    • Linearity, shift invariance, semi-group structure, non-creation of local extrema (zero-crossings), non-enhancement of local extrema, rotational symmetry, and scale invariance
• The scale parameter is the variance of the Gaussian filter
• Image details significantly smaller than (two times) the standard deviation (square root of variance) are removed from the image at that scale parameter
SIFT, construct the scale space

- Each octave corresponds to doubling the standard deviation
  - First image at each new octave is downsampled third image (octave image) from previous octave
SIFT, construct the scale space

**Figure 12.57**
Illustration using images of the first three octaves of scale space in SIFT. The entries in the table are values of standard deviation used at each scale of each octave. For example, the standard deviation used in scale 2 of octave 1 is $k \sigma_1$, which is equal to 1.0.

(The images of octave 1 are shown slightly overlapped to fit in the figure space.)
SIFT, obtain the initial keypoints

- First, difference two adjacent scale-space images in an octave.
SIFT, obtain the initial keypoints

• Second, detect extrema in the differences

The point (in black) is selected as an extremum point if its value is larger than the values of all its neighbors (in blue) or smaller than the values of all its neighbors (in blue).
SIFT, obtain the initial keypoints

**FIGURE 12.50** How Eq. (12-69) is implemented in scale space. There are $s + 3$ $L(x, y, \sigma)$ images and $s + 2$ corresponding $D(x, y, \sigma)$ images in each octave.
SIFT, improve the accuracy of the location of the keypoints

• Interpolate the values of the difference images about extrema
• Determine subpixel coordinates of extrema using interpolated values
SIFT, delete unsuitable keypoints

- Determine difference at subpixel keypoints
- Eliminate keypoints with low contrast and/or are poorly localized
- Additionally, delete keypoints associated with edges
  - Only keep corner-like features
    - Equivalent to thresholding the minor eigenvalue
Scale invariant feature transform (SIFT)

• Steps
  1. Construct the scale space
  2. Obtain the initial keypoints
  3. Improve the accuracy of the location of the keypoints
  4. Delete unsuitable keypoints
     – So far, we have computed the location of each keypoint in scale space (i.e., location and scale of each keypoint)
        • Scale invariance
     – Next is rotation invariance
  5. Compute keypoint orientations
  6. Compute keypoint descriptors
SIFT, compute keypoint orientations

• For each keypoint
  – At its scale, compute the magnitude and orientation of the keypoint
  – Form a histogram of orientations
    • 36 bins (10 degrees each)
    • Weight an orientation by its associated magnitude and a Gaussian, when adding it to an orientation bin
  – Initial orientation is largest bin
    • Create an additional keypoint for other bins within 80% the size of the largest bin
  – Improve orientation estimate using interpolation
    • Fit a parabola to values of the largest bin and its two neighboring bins
SIFT, compute keypoint orientations

• Keypoints
  – Location and scale (scale invariant)
  – Orientation (rotation invariant)
    • Length of arrow is histogram of orientations interpolated bin value
    • (Useful in matching keypoints across images)
Local regions

• The local region about each **oriented** keypoint is invariant to
  – Scale, orientation, illumination, and image viewpoint
SIFT, compute keypoint descriptors

- Feature descriptor
  - 16x16 region about keypoint
    - Gradient magnitude (Gaussian weighted) and direction at each point in region
    - Quantize gradient directions in each 4x4 subregion to 45 degree increments
      - Interpolate each of the 16 gradients directions to distribute it over all 8 bins (8 * 45 degrees = 360 degrees)
  - Concatenate the 8 8-directional histograms bins to form a 128-dimensional feature vector
SIFT, compute keypoint descriptors

• Rotation invariance
  – Rotate the 8-directional histograms relative to the keypoint orientation

• Robustness to changes in illumination
  – Unitize the 128-dimensional feature vector
  – Threshold to reduce the influence of large gradient magnitudes
  – Unitize again
Matching keypoints across images

- First image is whole image
- Second image is darker version of red rectangle
Matching keypoints across images using SIFT features and feature descriptors
CSE 166 topics

- Image acquisition
- Image filtering and enhancement
- Image restoration
- Wavelets and other image transforms
- Color image processing
- Image compression and watermarking
- Morphological image processing
- Image segmentation
- Feature extraction
Geometric transformations

**Euclidean transformation**

\[
x' = x \cos \theta - y \sin \theta + t_x \\
y' = x \sin \theta + y \cos \theta + t_y
\]

\[
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & -\sin \theta & t_x \\
\sin \theta & \cos \theta & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

**Similarity transformation**

\[
x' = sx \cos \theta - sy \sin \theta + t_x \\
y' = sx \sin \theta + sy \cos \theta + t_y
\]

\[
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} =
\begin{bmatrix}
s \cos \theta & -s \sin \theta & t_x \\
s \sin \theta & s \cos \theta & t_y \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

**Affine transformation**

\[
x' = a_{11}x + a_{12}y + a_{13} \\
y' = a_{21}x + a_{22}y + a_{23}
\]

\[
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} =
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]
Image filtering and enhancement

- Intensity transformations
- Spatial filtering

Low-pass filter

Gamma correction
Image filtering and enhancement

- Filtering in the frequency domain
Image restoration

• Noise models
• Noise reduction
Color image processing

• Color models
• Color transformations
Wavelets and other transforms

Basis vectors
Image compression and watermarking

- Lossless vs lossy compression
Morphological image processing

• Dilation and erosion
• Opening and closing

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.
Image segmentation

• Thresholding
Feature extraction

- Feature detection
CSE 166

• Image acquisition
• Geometric transformations and image interpolation
• Intensity transformations
• Spatial filtering
• Fourier transform and filtering in the frequency domain
• Image restoration
• Color image processing

• Basis vectors
• Matrix-based transforms
• Basis images
• Wavelet transform
• Image compression
• Image watermarking
• Morphological image processing
• Edge detection
• Image segmentation
• Feature extraction