Templates, Image Pyramids, and Filter Banks

Computer Vision
James Hays, Brown
Review

1. Match the spatial domain image to the Fourier magnitude image
Reminder

• Project 1 due in one week
Today’s class

• Template matching

• Image Pyramids

• Filter banks and texture

• Denoising, Compression
Template matching

• Goal: find in image

• Main challenge: What is a good similarity or distance measure between two patches?
  – Correlation
  – Zero-mean correlation
  – Sum Square Difference
  – Normalized Cross Correlation
Matching with filters

• **Goal:** find \( \text{\includegraphics[width=10pt]{eye}} \) in image

• **Method 0:** filter the image with eye patch

\[
h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]
\]

\( f = \text{image} \)
\( g = \text{filter} \)

What went wrong?
Matching with filters

- **Goal:** find 🌟 in image

- **Method 1:** filter the image with zero-mean eye

\[
h[m,n] = \sum_{k,l} (f[k,l] - \bar{f}) (g[m+k,n+l])
\]

True detections

False detections

Input

Filtered Image (scaled)

Thresholded Image
Matching with filters

- **Goal:** find 🕶️ in image
- **Method 2:** SSD

\[
h[m,n] = \sum_{k,l} (g[k,l] - f[m+k,n+l])^2
\]
Matching with filters

• Goal: find 🕶️ in image

• Method 2: SSD

\[ h[m, n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2 \]

What’s the potential downside of SSD?
Matching with filters

- **Goal**: find \( \begin{figure}[h] \end{figure} \) in image
- **Method 3**: Normalized cross-correlation

\[
h[m, n] = \sum_{k,l} (g[k, l] - \bar{g})(f[m - k, n - l] - \bar{f}_{m,n})
\]

\[
\left( \sum_{k,l} (g[k, l] - \bar{g})^2 \sum_{k,l} (f[m - k, n - l] - \bar{f}_{m,n})^2 \right)^{0.5}
\]

**Matlab**: `normxcorr2(template, im)`
Matching with filters

• Goal: find 🕍 in image
• Method 3: Normalized cross-correlation
Matching with filters

• Goal: find ☁️ in image

• Method 3: Normalized cross-correlation

Input

Normalized X-Correlation

Thresholded Image

True detections
Q: What is the best method to use?

A: Depends

- SSD: faster, sensitive to overall intensity
- Normalized cross-correlation: slower, invariant to local average intensity and contrast
Q: What if we want to find larger or smaller eyes?

A: Image Pyramid
Review of Sampling

Image → Gaussian Filter → Low-Pass Filtered Image → Sample → Low-Res Image
Gaussian pyramid

Source: Forsyth
Template Matching with Image Pyramids

Input: Image, Template
1. Match template at current scale

2. Downsample image

3. Repeat 1-2 until image is very small

4. Take responses above some threshold, perhaps with non-maxima suppression
Coarse-to-fine Image Registration

1. Compute Gaussian pyramid
2. Align with coarse pyramid
3. Successively align with finer pyramids
   - Search smaller range

Why is this faster?

Are we guaranteed to get the same result?
Laplacian filter

unit impulse

Gaussian

Laplacian of Gaussian

Source: Lazebnik
2D edge detection filters

\[ h_\sigma(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2 + v^2}{2\sigma^2}} \]

\[ \frac{\partial}{\partial x} h_\sigma(u, v) \]

\[ \nabla^2 h_\sigma(u, v) \]

\[ \nabla^2 \text{ is the } \textbf{Laplacian} \text{ operator:} \]

\[ \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \]
Laplacian pyramid

Source: Forsyth
Computing Gaussian/Laplacian Pyramid

Can we reconstruct the original from the laplacian pyramid?

Hybrid Image
Hybrid Image in Laplacian Pyramid

High frequency $\rightarrow$ Low frequency
Image representation

- Pixels: great for spatial resolution, poor access to frequency

- Fourier transform: great for frequency, not for spatial info

- Pyramids/filter banks: balance between spatial and frequency information
Major uses of image pyramids

- Compression

- Object detection
  - Scale search
  - Features

- Detecting stable interest points

- Registration
  - Course-to-fine
Denoising

Additive Gaussian Noise

Gaussian Filter
Reducing Gaussian noise

Smoothing with larger standard deviations suppresses noise, but also blurs the image.

Source: S. Lazebnik
Reducing salt-and-pepper noise by Gaussian smoothing

3x3  5x5  7x7
Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window.

![Median filter diagram]

- Is median filtering linear?

Source: K. Grauman
Median filter

- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers

Source: K. Grauman
Median filter

Salt-and-pepper noise  
Median filtered

• MATLAB: medfilt2(image, [h w])

Source: M. Hebert
Median vs. Gaussian filtering

Gaussian

Median
Other non-linear filters

- Weighted median (pixels further from center count less)

- Clipped mean (average, ignoring few brightest and darkest pixels)

- Bilateral filtering (weight by spatial distance and intensity difference)

Image: [http://vision.ai.uiuc.edu/?p=1455](http://vision.ai.uiuc.edu/?p=1455)
Review of last three days
Review: Image filtering

\[ f[\cdot, \cdot] \]

\[ g[\cdot, \cdot] \]

\[
h[m, n] = \sum_{k, l} f[k, l] g[m + k, n + l]
\]
Image filtering

\[ f[\ldots] \]

\[ h[\ldots] \]

\[ h[m,n] = \sum_{k,l} f[k,l] g[m+k, n+l] \]
Image filtering

\[ f[\cdot, \cdot] \]

\[ h[\cdot, \cdot] \]

\[ h[m, n] = \sum_{k,l} f[k, l] g[m+k, n+l] \]
Filtering in spatial domain

\[
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{bmatrix}
\]
Filtering in frequency domain
Review of Last 3 Days

• Linear filters for basic processing
  – Edge filter (high-pass)
  – Gaussian filter (low-pass)

\[
\begin{bmatrix}
-1 & 1
\end{bmatrix}
\]

FFT of Gradient Filter

FFT of Gaussian

Gaussian

Slide: Hoiem
Review of Last 3 Days

- Derivative of Gaussian
Review of Last 3 Days

• Applications of filters
  – Template matching (SSD or Normxcorr2)
    • SSD can be done with linear filters, is sensitive to overall intensity
  – Gaussian pyramid
    • Coarse-to-fine search, multi-scale detection
  – Laplacian pyramid
    • More compact image representation
    • Can be used for compositing in graphics
  – Downsampling
    • Need to sufficiently low-pass before downsampling
Next Lectures

• Machine Learning Crash Course