Administrative
Feedback

• Homework
Mid-term

• Practice exam
  • POLL
Where we are?

• What is next?
• Homework
• Programming assignments - BONUS
• Mid-terms
• Final project

• Grading
Feedback

• How is it going so far?
• Help with any topic from class?
• Help in assignments?
• Any issues so far?
  • Office hours?
• General comments?
Features – I
Part - II
Lecture 9
Recap
Basic Idea in Corner Detection

- Recognize corners by looking at small window.
- Shift in *any direction* to 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
Corner Detection by Auto-correlation

Change in appearance of window \( w(x,y) \) for shift \([u,v] \):

\[
E(u,v) = \sum_{x,y} w(x,y) \left[ I(x+u, y+v) - I(x, y) \right]^2
\]

Window function

Shifted intensity

Intensity

Window function \( w(x,y) = \) 1 in window, 0 outside

or

Gaussian

Source: R. Szeliski
Corner Detection: Mathematics

The quadratic approximation simplifies to

$$E(u, v) \approx [u \ v] M [u \ v]^T$$

where $M$ is a second moment matrix computed from image derivatives:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \sum \nabla I (\nabla I)^T$$
0. Input image
   We want to compute $M$ at each pixel.

1. Compute image derivatives (optionally, blur first).

2. Compute $M$ components as squares of derivatives.

3. Gaussian filter $g()$ with width $s$
   
   
   $$= g(I_x^2), g(I_y^2), g(I_x \circ I_y)$$

4. Compute cornerness
   
   $$C = \det(M) - \alpha \text{trace}(M)^2$$
   
   $$= g(I_x^2) \circ g(I_y^2) - g(I_x \circ I_y)^2 - \alpha [g(I_x^2) + g(I_y^2)]^2$$

5. Threshold on $C$ to pick high cornerness

6. Non-maximal suppression to pick peaks.
Harris Detector: Steps
Histogram of Oriented Gradients

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)
  - The vote is the gradient magnitude
HOG

Input image

Histogram of Oriented Gradients
SIFT
Scale Invariant Feature Transform (SIFT)

• Lowe., D. 2004, IJCV

Distinctive Image Features from Scale-Invariant Keypoints

DAVID G. LOWE
Computer Science Department, University of British Columbia, Vancouver, B.C., Canada
Lowe@cs.ubc.ca

Received January 10, 2003; Revised January 7, 2004; Accepted January 22, 2004

Abstract. This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Routh transform to identify clusters belonging to a single object, and finally performing verification through least-squares solution for consistent pose parameters. This approach to recognition can robustly identify objects among clutter and occlusion while achieving near real-time performance.

Keywords: Invariant features, object recognition, scale invariance, image matching

1. Introduction

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition. The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Following are the major stages of computation used to generate the set of image features:

1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
Scale Invariant Feature Transform (SIFT)

• Image content is transformed into local feature coordinates
• Invariant to
  • translation
  • rotation
  • scale, and
  • other imaging parameters
Scale Invariant Feature Transform (SIFT)

• Image content is transformed into local feature coordinates
Overall Procedure at a High Level

- **Scale-Space Extrema Detection**: Search over multiple scales and image locations.
- **KeyPoint Localization**: Fit a model to determine location and scale. Select KeyPoints based on a measure of stability.
- **Orientation Assignment**: Compute best orientation(s) for each keyPoint region.
- **KeyPoint Description**: Use local image gradients at selected scale and rotation to describe each keyPoint region.
Automatic Scale Selection

\[ f(I_{i_1\ldots i_m}(x, \sigma)) = f(I_{i_1\ldots i_m}(x', \sigma')) \]

How to find patch sizes at which \( f \) response is equal?

What is a good \( f \)?
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

• Function responses for increasing scale (scale signature)
What Is A Useful Signature Function $f$?

- **Single Gaussian**
- **1st Derivative of Gaussian**
- **Laplacian of Gaussian**

10/9/2020
What Is A Useful Signature Function $f$?

“Blob” detector is common for corners
- Laplacian (2\textsuperscript{nd} derivative) of Gaussian (LoG)
Find local maxima in position-scale space

\[
\begin{align*}
\sigma^5 & \quad \sigma^4 \\ 
\sigma^3 & \quad \sigma^2 \\ 
\sigma & 
\end{align*}
\]

\(\Rightarrow\) List of \((x, y, s)\)

Find maxima/minima

Scale
Alternative kernel

Approximate LoG with Difference-of-Gaussian (DoG).
Alternative kernel

Approximate LoG with Difference-of-Gaussian (DoG).

1. Blur image with $\sigma$ Gaussian kernel
2. Blur image with $k\sigma$ Gaussian kernel
3. Subtract 2. from 1.
Scale-space
Find local maxima in position-scale space of DoG

\[ \text{Find maxima/minima} \]

\[ \Rightarrow \text{List of } (x, y, s) \]
Results: Difference-of-Gaussian

- Larger circles = larger scale
- Descriptors with maximal scale response
SIFT Orientation estimation

• Compute gradient orientation histogram
• Select dominant orientation $\Theta$
SIFT Orientation Normalization

- Compute gradient orientation histogram
- Select dominant orientation $\Theta$
- Normalize: rotate to fixed orientation
SIFT descriptor formation

- Compute on local 16 x 16 window around detection.
- Rotate and scale window according to discovered orientation $\Theta$ and scale $\sigma$ (gain invariance).
- Compute gradients weighted by a Gaussian of variance half the window (for smooth falloff).

Actually 16x16, only showing 8x8
SIFT descriptor formation

• 4x4 array of gradient orientation histograms weighted by gradient magnitude.

• Bin into 8 orientations x 4x4 array = 128 dimensions.
SIFT Descriptor Extraction

Gradient magnitude and orientation

8 bin ‘histogram’ - add magnitude amounts!

Utkarsh Sinha
Reduce effect of illumination

• 128-dim vector normalized to 1

• Threshold gradient magnitudes to avoid excessive influence of high gradients
  • After normalization, clamp gradients > 0.2
  • Renormalize
Review: Local Descriptors

• Most features can be thought of as
  • templates,
  • histograms (counts),
  • or combinations

• The ideal descriptor should be
  – Robust and Distinctive
  – Compact and Efficient

• Most available descriptors focus on edge/gradient information
  – Capture texture information
  – Color rarely used
Questions?

Sources for this lecture include materials from works by Mubarak Shah, S. Seitz, James Tompkin and Ulas Bagci