CS231A Section 6: Problem Set 3

Jiahui Shi

Announcements

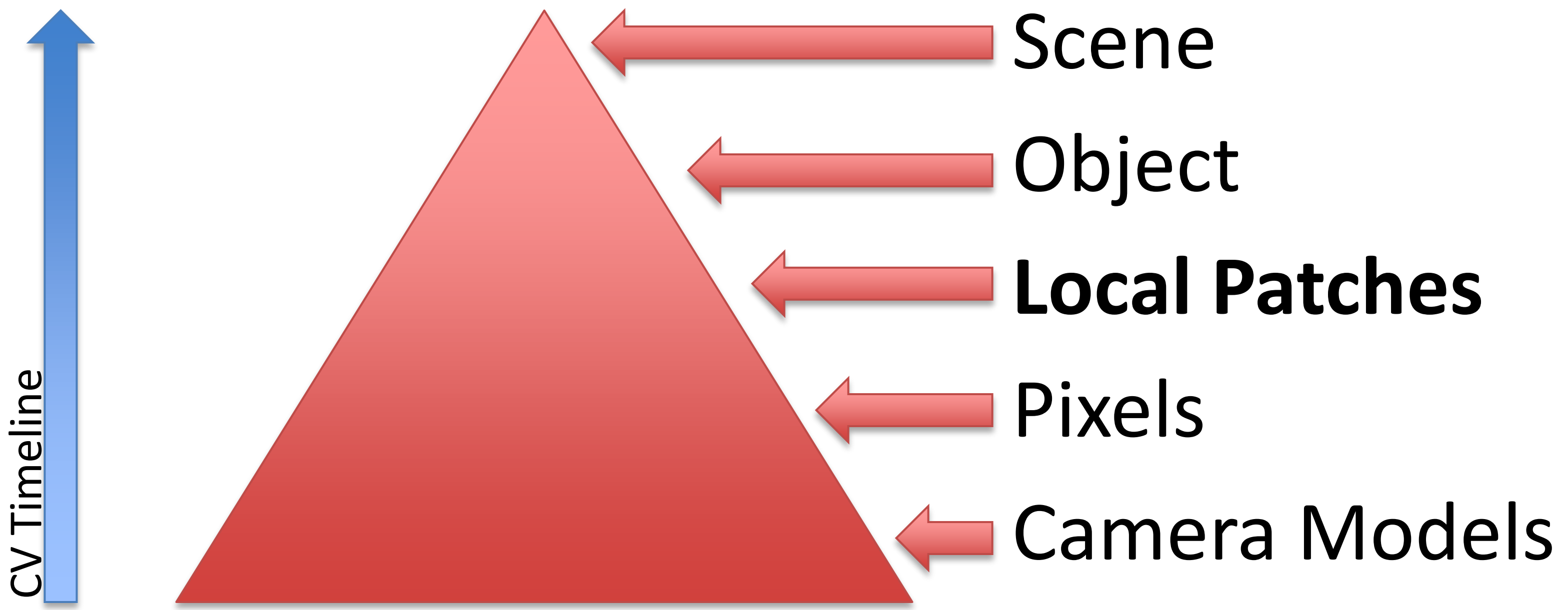
- PS3 Due 9:30am Monday, Nov 14

- Extra Office Hours:
  - Saturday 3 – 6pm Gates 104

SIFT

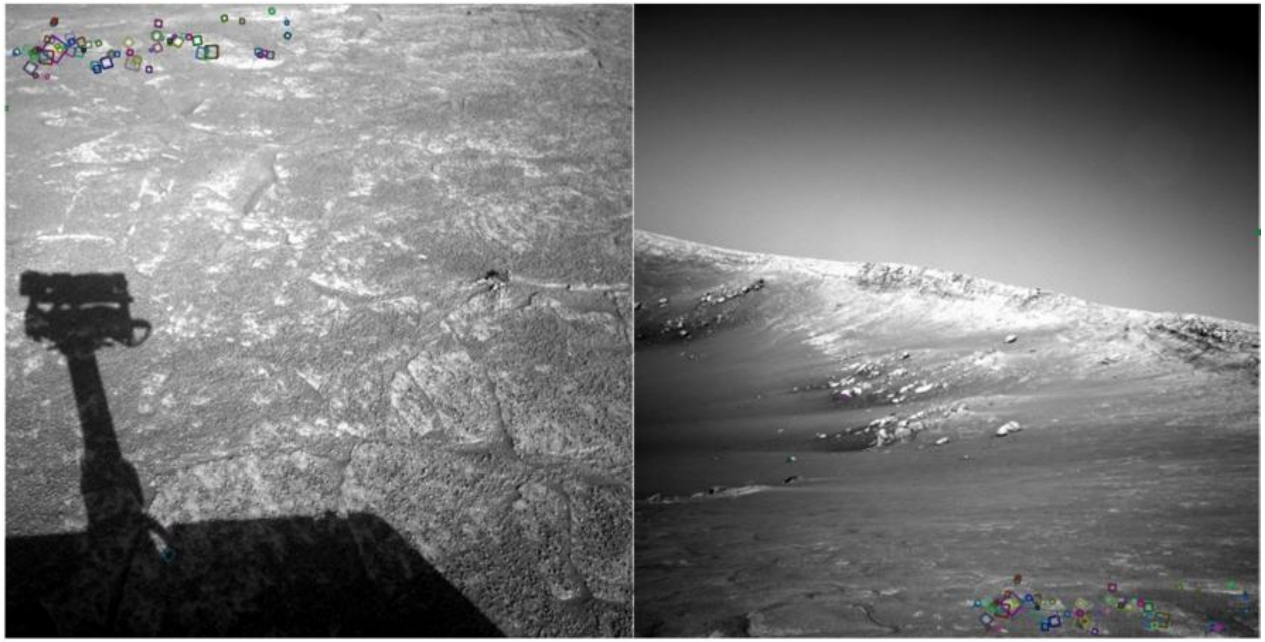
- Motivations
- What makes a point "interesting"
- What to do with Keypoints
- SIFT detector
- SIFT descriptor
- Object Recognition with SIFT

Where are we?


CV Timeline
Scene
Object
Local Patches
Pixels
Camera Models

What to identify? (Keypoints)

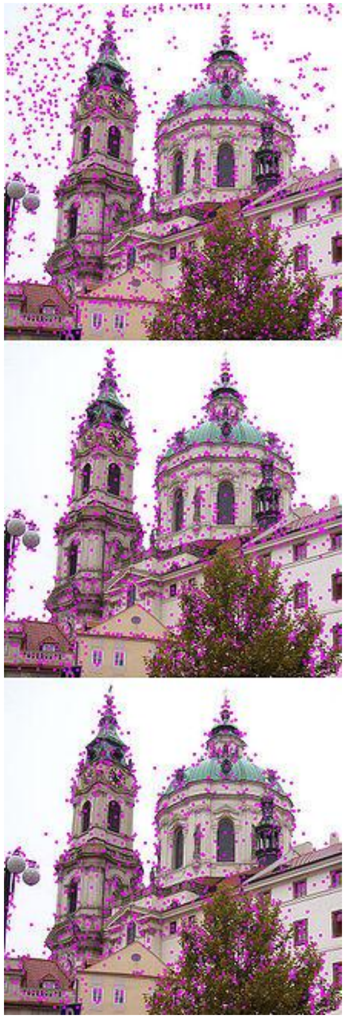
- Salient points that would be present across a set of (reasonably) related images
  - Repeatable
  - Distinctive



What information? (Descriptors)

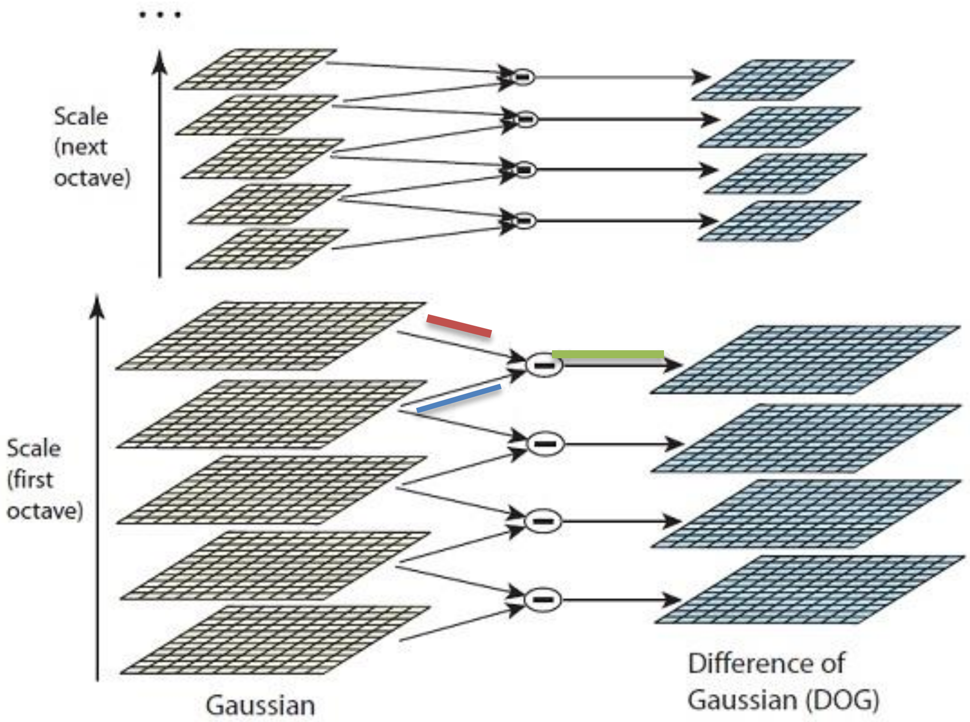
SIFT

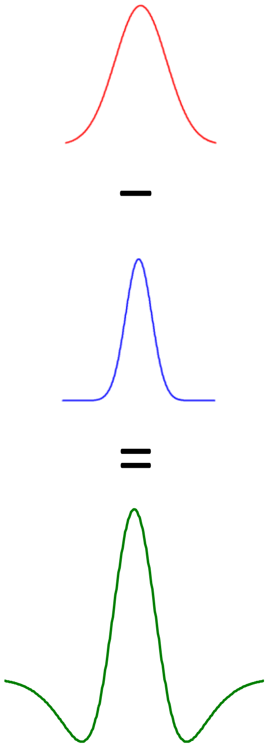
- Ubiquitous: 10,000+ Citations
- David Lowe, 2001 and 2004, from UBC
- Both a detector and a descriptor
- **Invariant to:**
  - **Illumination**
  - **Scale**
  - **Rotation**
  - **Affine**
  - **Perspective**



SIFT: Keypoint Detector

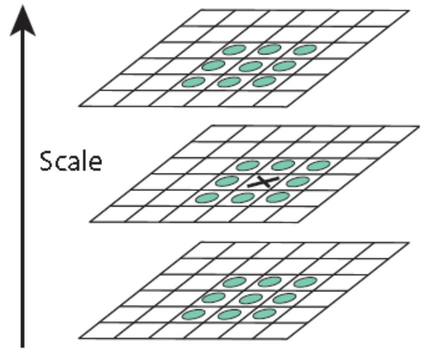
- Difference of Gaussians using Scale Space Pyramid
- Section 3 and 4 of Lowe, 2004



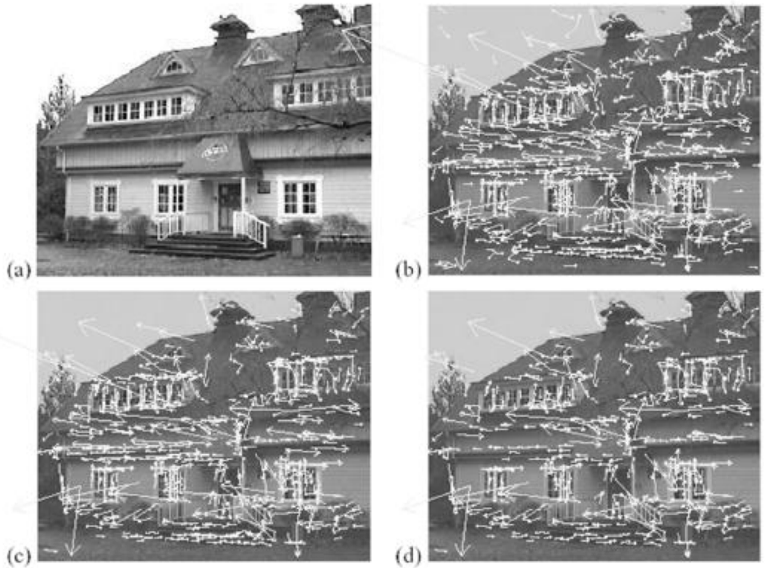


Extract and Prune *max(DoG)*

- A point is extreme if it larger/smaller than its 26 neighboring points



- Prune for:
  – Low Contrast
  – Edge Points

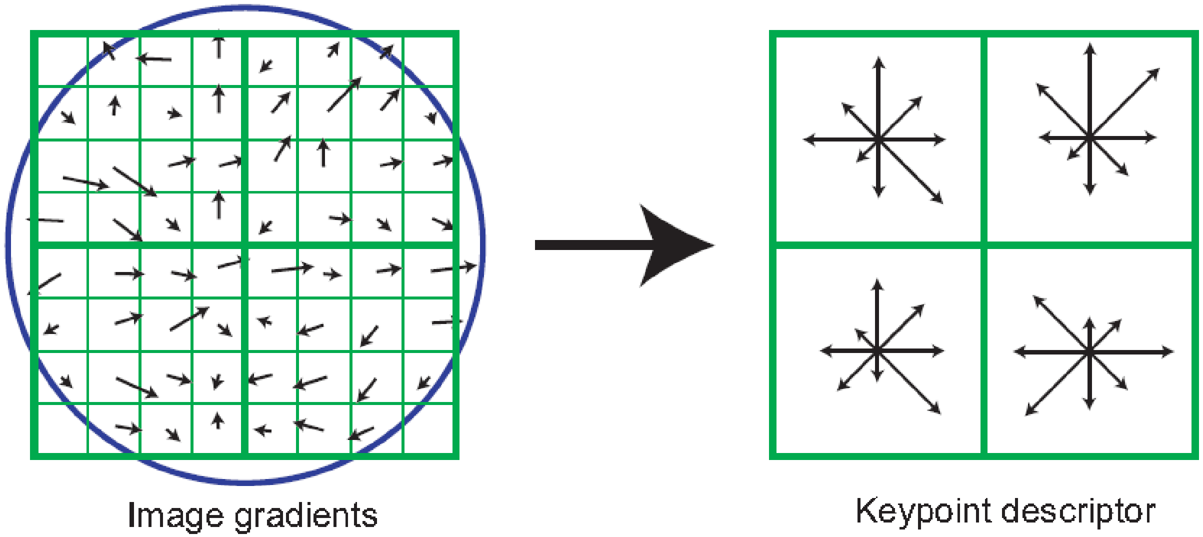


(a) 233x189 image
(b) 832 DoG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures (removing edge responses)

What can we do with keypoints?

SIFT Descriptor

- Sections 5 and 6 in Lowe, 2004.
- Inspired by neurological research and models
- Keypoint is center of square patch of pixels, blurred at the scale of the keypoint
- Construction of Orientation Histogram for each 4x4 set of pixels
- All pixels are rotated by the orientation of the keypoint



Steps of finding descriptor (for PS3)

Given a keypoint P

1. Find gradient:
    1. Apply Gaussian filter to the image using the scale of P as sigma (standard deviation)
    2. Find gradient in x and y directions (down - up, right - left)
    3. Magnitude: sqrt(Ix^2+Iy^2); Theta: atan2(Iy, Ix)
2. Find the patch associated with P
3. For each bin b of the 4 x 4 bins in the patch
    1. Initialize the histogram of the 8 angle bins
    2. For each pixel p in b
        1. Find the angle bin it falls into
        2. In the histogram, increase the value of this angle bin by the gradient magnitude of p
    3. Append this histogram to the end of the descriptor

Demo on Paper Pad

Many little details

- Many, many experimentally determined parameters
- The standard patch size is 16
  - Feature Vector = 16 patches * 8 bins = 128 elements
- Normalize each feature vector
- Avoid large (and potentially erroneous) gradients by capping each element at 0.2
- Add extra blur for camera sensor
- [More steps…]
- *We try to make the process as linear as possible…*

The Matching Problem

- Locate arbitrary objects despite environmental difficulties

Object Recognition with SIFT (Sec 7, Lowe 2004)

- Step 1: Feature matching
- Step 2: Hough transform in pose space
- Step 3: Geometric verification via affine transformation

Step 1: Feature matching

- A match is determined by distance between closest neighbor and second closest neighbor
  - Euclidean distance between descriptor vectors
- Nearest Neighbor problem
  - k-d tree is inefficient
  - Best-Bin-First (BBF) (Beis and Lowe, 1997), modified from k-d tree, giving approximated result

Step 2: Hough transform in pose space

- Goal: given test image and training image, find object pose
- Input: keypoint matches (p_i -> p_i') of an object, many are false matches
- Output: estimated object pose

- A match is specified by 4 parameters <x, y, scale, orientation>

Step 2: Hough transform in pose space

1. Discrete bins in 4D space;
2. Assign each match to the bin whose object pose is consistent with it;
3. Find bins with > 3 votes.

Step 2: Hough transform in pose space

- Have to use broad bins since there are only 4 parameters but 6 dof
  - bin size of 30 degrees for orientation, a factor of 2 for scale, etc.
- Match to 2 closest bins in each dimension

Step 3: Geometric verification via affine transformation

- Verify each bin with at least 3 entries
- 3 matches determine an affine transformation

  (An approximation of finding the fundamental matrix which requires more matches)

Parameters of Affine Transformation

- Affine transformation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

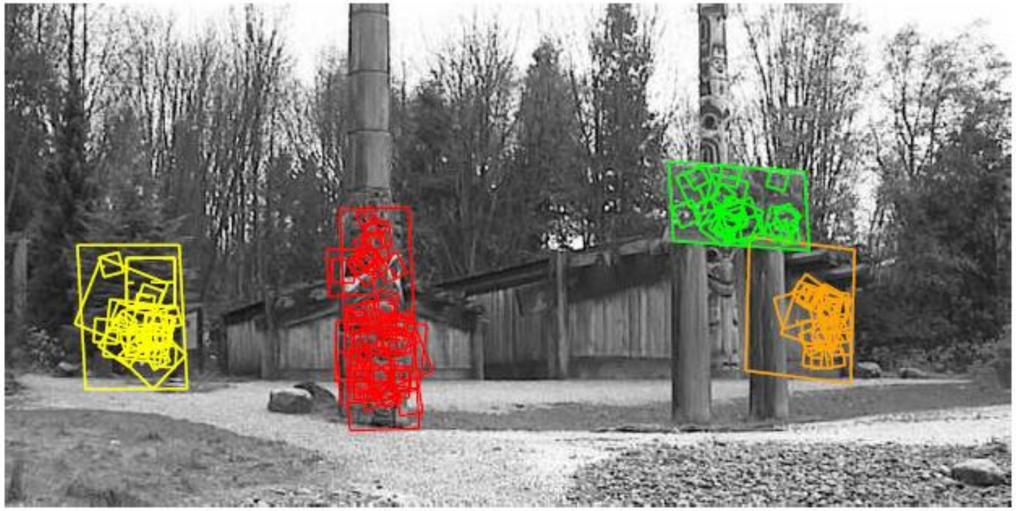
- Least-squares solution for the best affine projection parameters

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} = [\mathbf{A}^{\mathrm{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Results



Object Matching in PS3

- PS3 uses a simplified version of object matching
  - Match on keypoints (not descriptors)
  - Use Hough to determine bounding box parameters.

http://www.cs.washington.edu/homes/ankit/research_projects/sura/sura-website/sift.htm