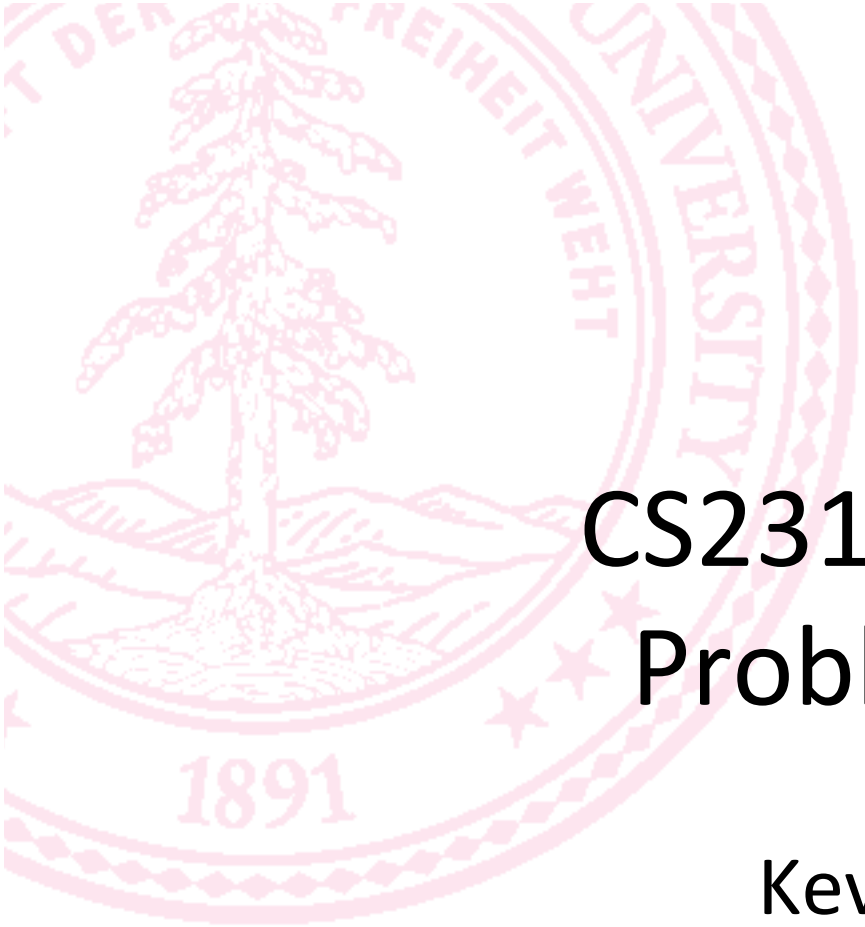


CS231A Section: Problem Set 3

Kevin Wong

Announcements

- PS3 Due 5pm Friday, Feb 21

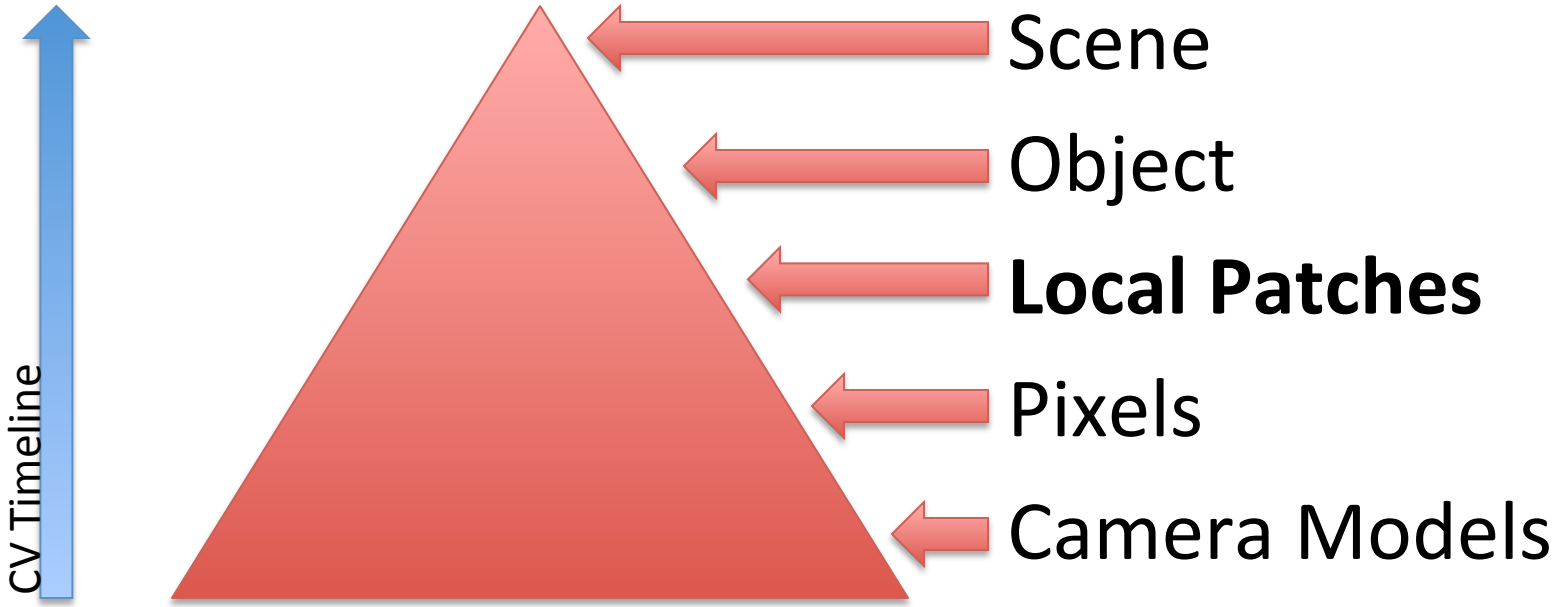
Topics

- PS3
  - Keypoints and features
  - SIFT Matching
  - RANSAC
  - Hough Transforms
  - Shape Context
  - Voxel Coloring
- Projects
  - OpenCV with Matlab

SIFT

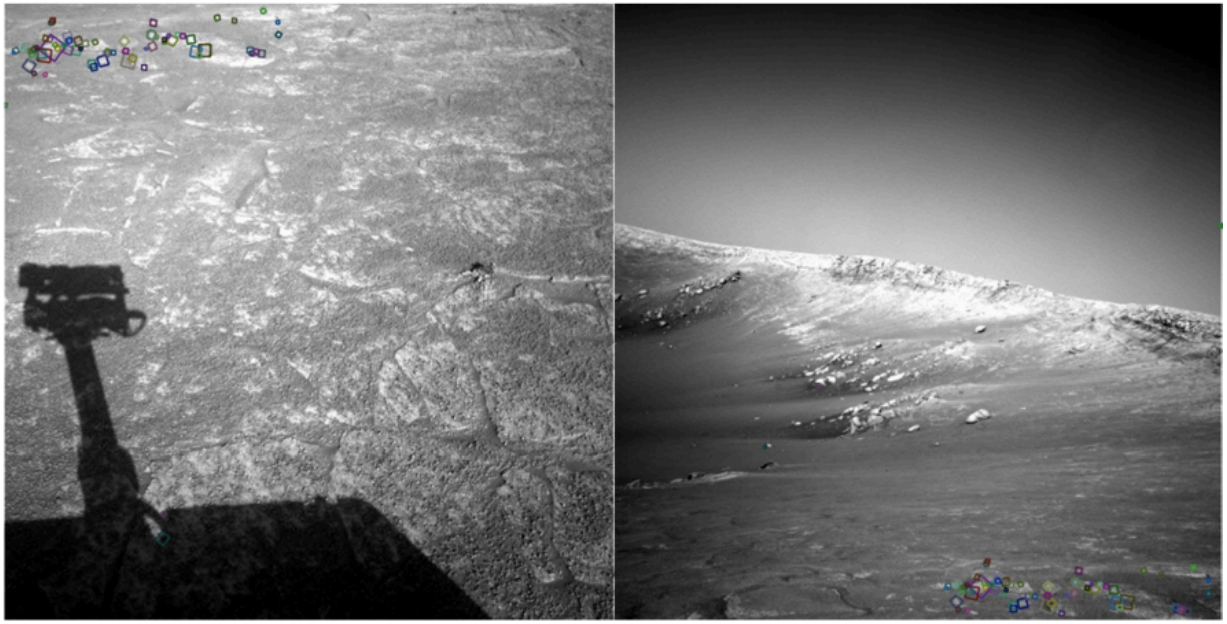
- Motivations
- What makes a point “interesting”
- What to do with Keypoints
- SIFT detector
- SIFT descriptor
- Object Recognition with SIFT

Where are we?


CV Timeline
Scene
Object
Local Patches
Pixels
Camera Models

What to identify? (Keypoints)

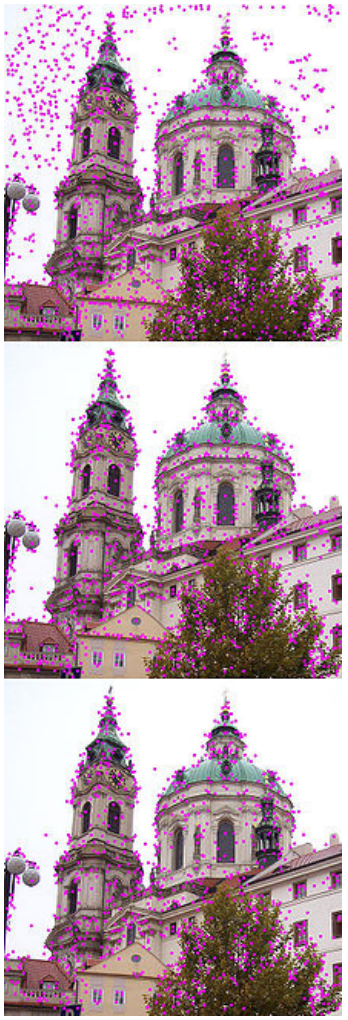
- Salient points that would be present across a set of (reasonably) related images
  - Repeatable
  - Distinctive



What information? (Descriptors)

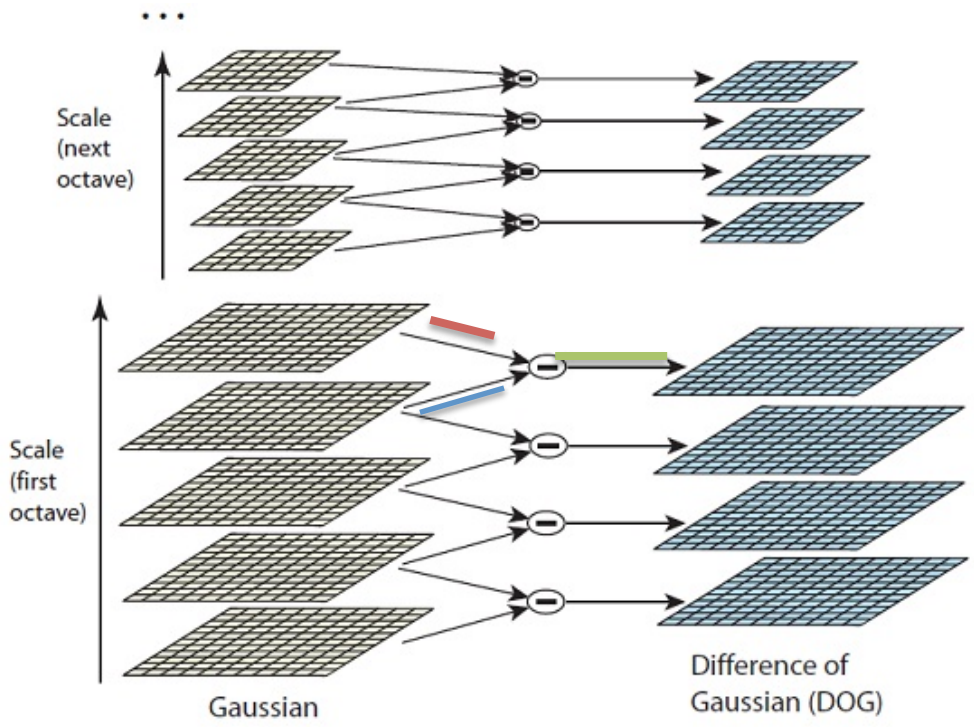
SIFT

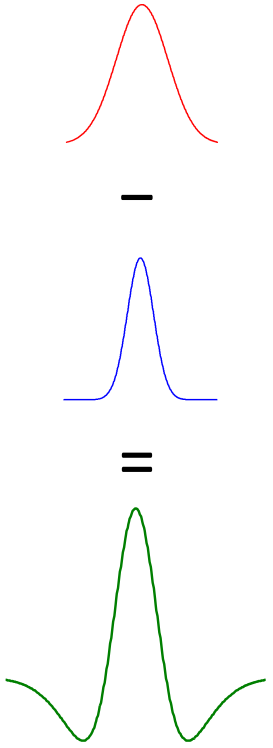
- Ubiquitous: 10,000+ Citations
- David Lowe, 2001 and 2004, from UBC
- Both a detector and a descriptor
- **Invariant to:**
  - **Illumination**
  - **Scale**
  - **Rotation**
  - **Affine**
  - **Perspective**



SIFT: Keypoint Detector

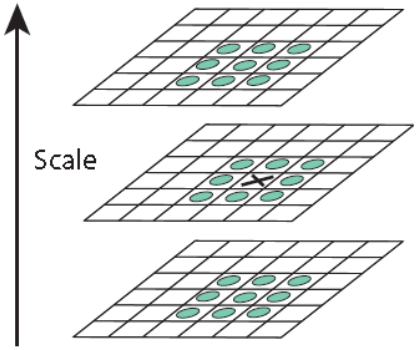
- Difference of Gaussians using Scale Space Pyramid
- Section 3 and 4 of Lowe, 2004



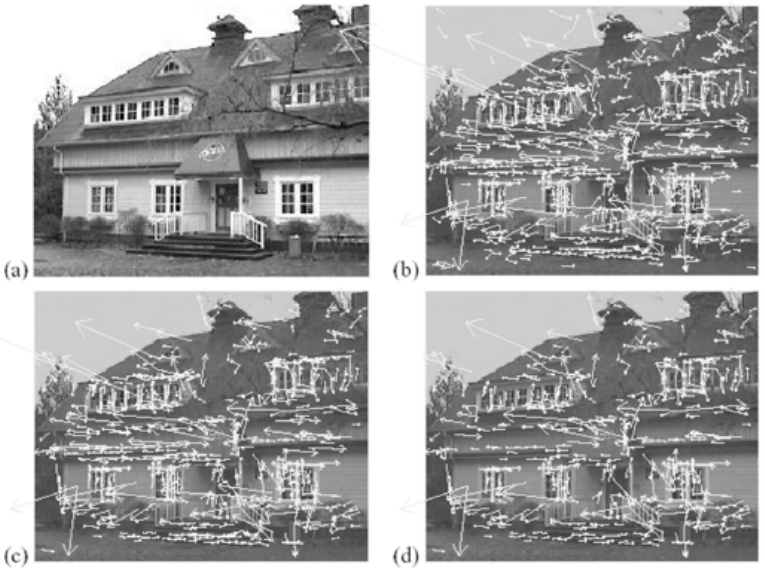


Extract and Prune *max(DoG)*

- 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

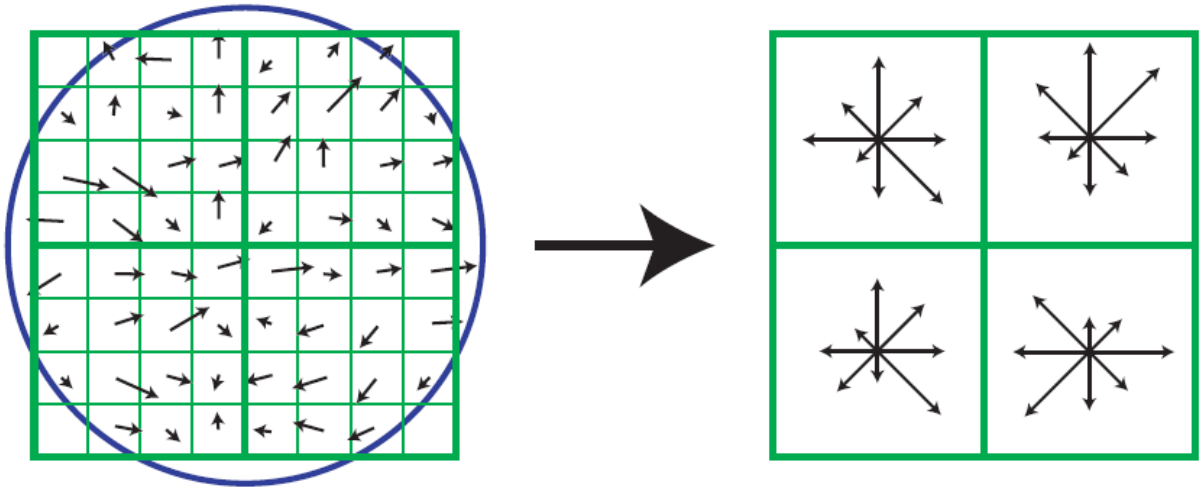


Image gradients

Keypoint descriptor

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: Descriptor 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 for general 3d poses
  - bin size of 30 degrees for orientation, a factor of 2 for scale, etc. For the problem, you can use a uniform bins, which isn't optimal, good enough for the problem.

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:

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

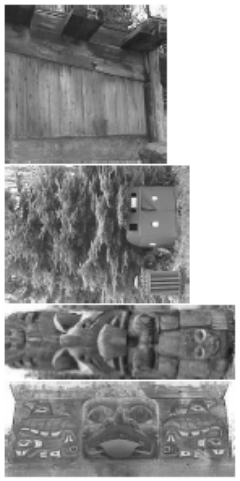
- Least-squares solution for the best affine projection parameters

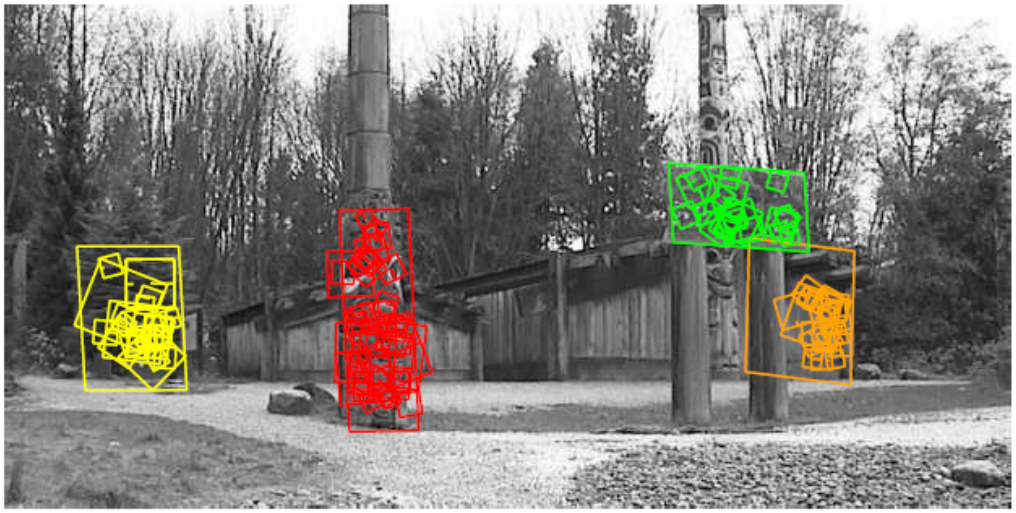
$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

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

Results





Object Matching in PS3

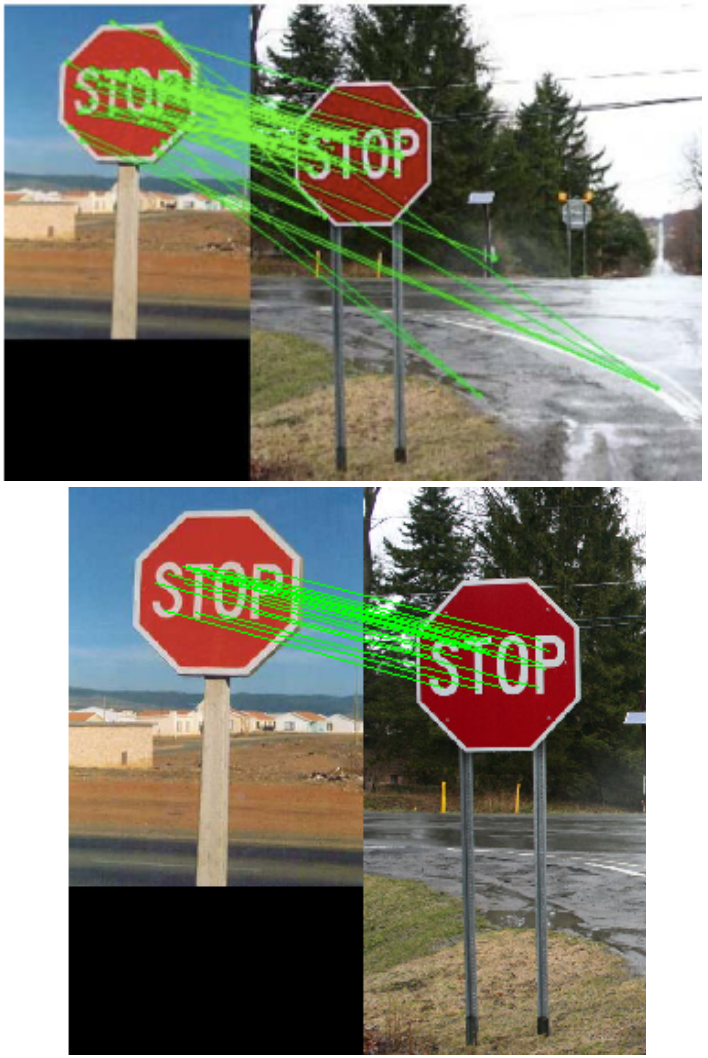
- PS3 uses a simplified version of object matching
  - Match on keypoints descriptors
  - Use Hough to determine bounding box parameters.

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

RANSAC

Algorithm

1. Select a random sample of SIFT matches. (What is the minimum?)
2. Compute Homography from matches.
3. Calculate set of inliers, thresholding the L2 reprojection error



RANSAC

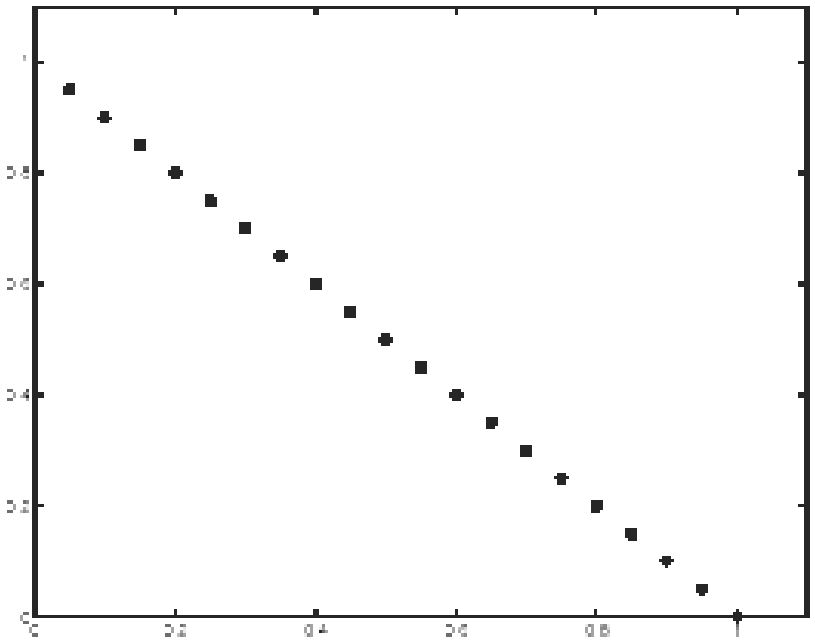
- Sampling Points
  - How many?
  - Point Restrictions?

Ransac

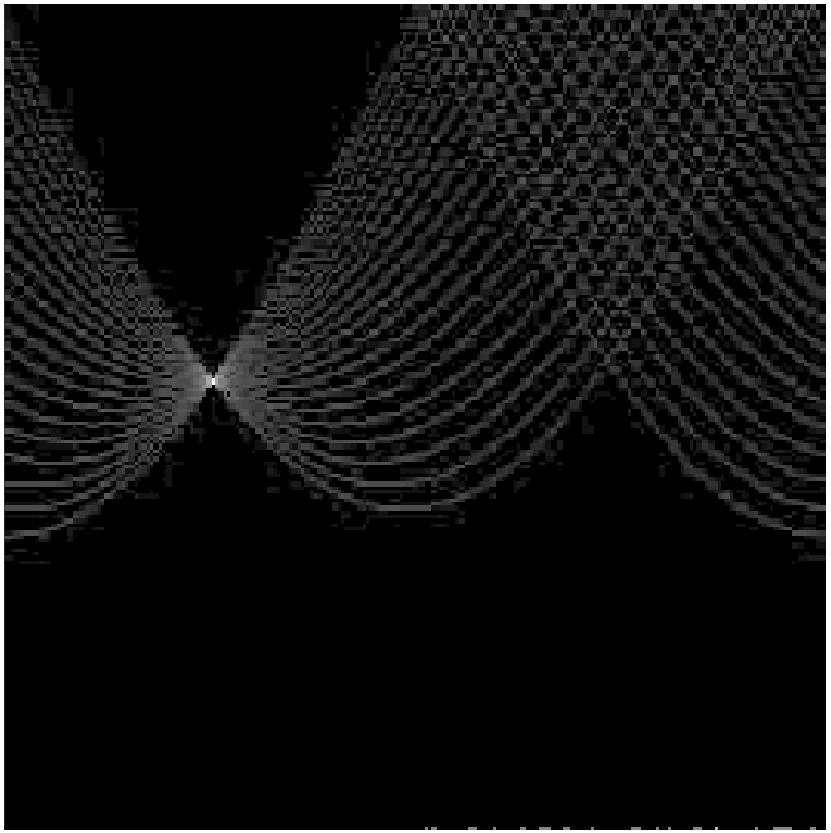
- Calculating the Homography:

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}$$

Hough Transforms



features

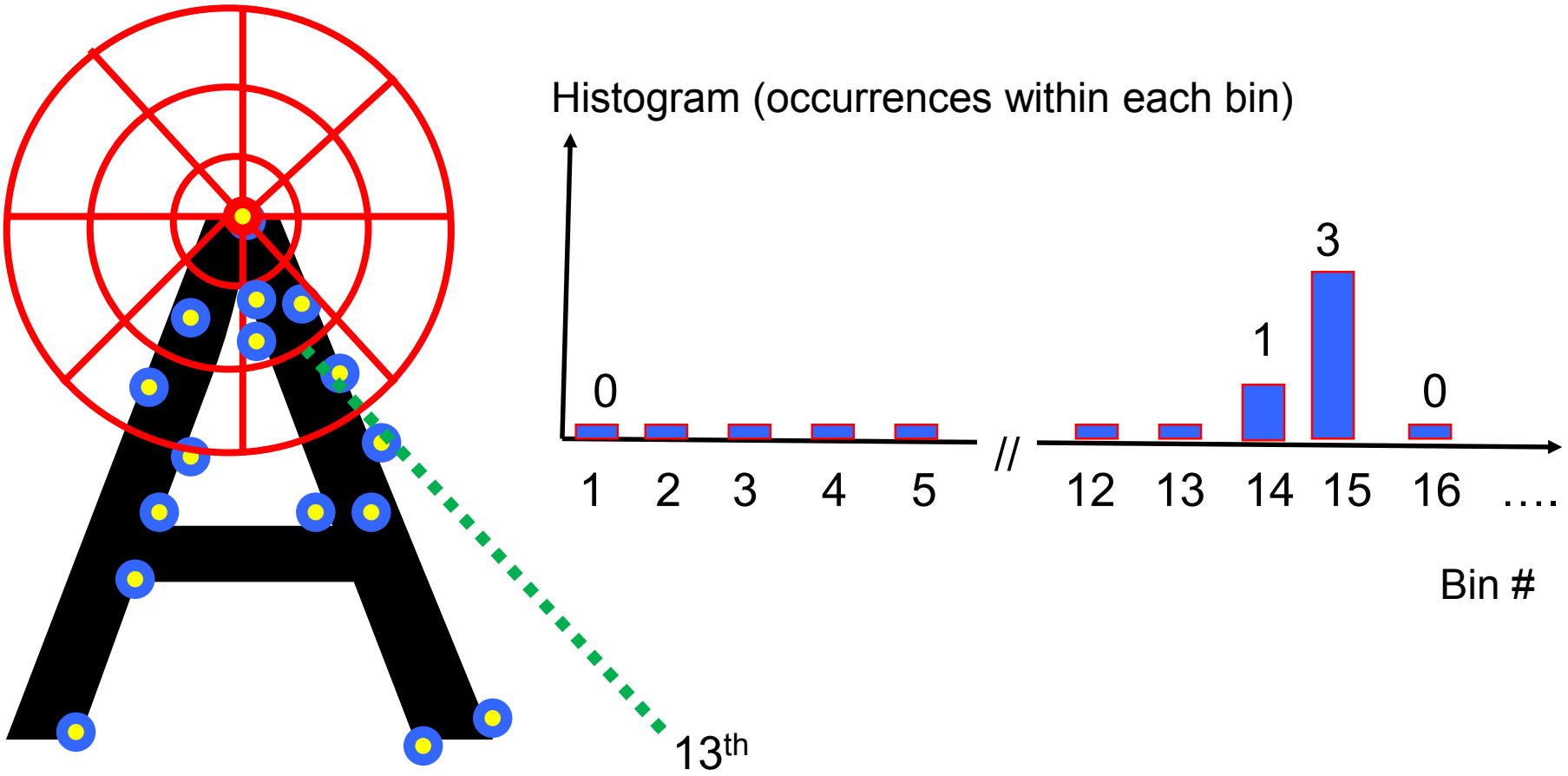


votes

PS3 Hough Transform

- Binning bounding box parameters instead of poses. (x, y, orientation, scale(width))
- For each match, compute relative bounding boxes in second image, bounding boxes might be rotated.
- Each image cases a vote in 4D parameter spaced of bounding boxes
- Find bins with more votes than the threshold.
- How do you combine votes in a bin?

Shape Context



Shape Context Tips

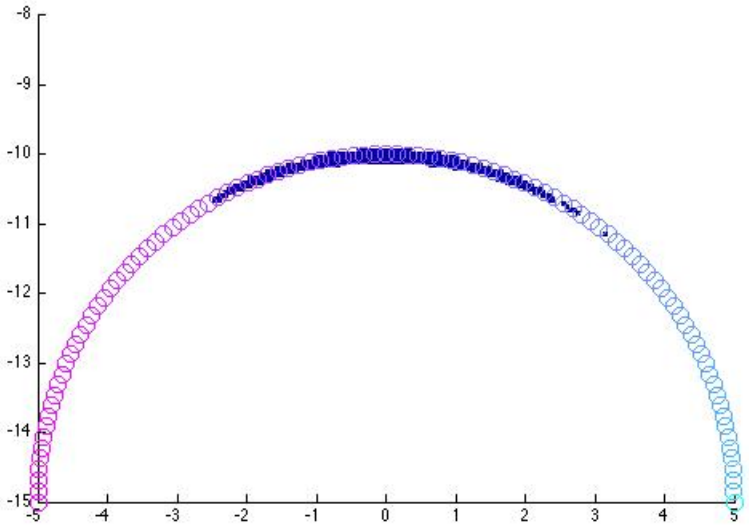
- We use a slightly different radial binning definition, see the comments.

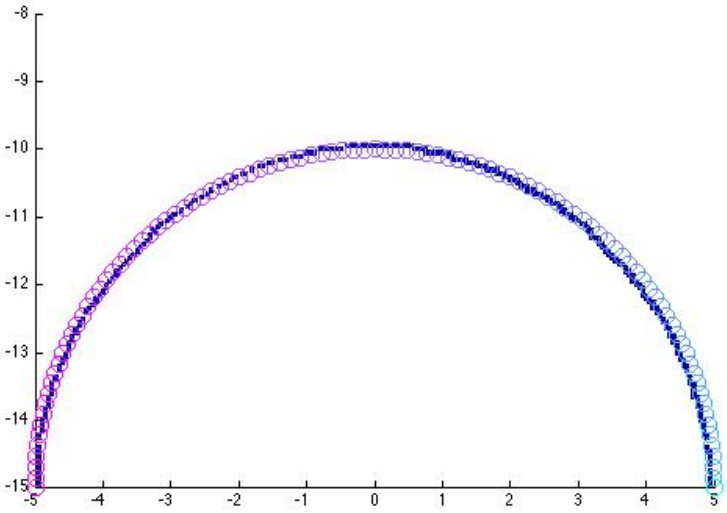
Voxel Coloring

- We handle the grid transversal and plotting functions, but we ask you to implement three functions.
  - keep: Updates occlusions, in voxel_coloring.m
  - Photoconsistent: Checks for voxel photo consistancy, see paper.
  - voxel_projections: Finds pixels representing the projection of a voxel into each of the images, taking occlusions into consideration.

Voxel Coloring

Bad occlusion mask





Voxel Coloring Tips

- Camera matrices given are general 3x4 camera matrices, not 3x3 K matrices.
- Grid_spacing is the voxel spacing, so given a voxel corner, you can compute the location of the other corners.
- Photoconsistancy, the sigma_0 parameter is not the sensor standard deviation.

Projects

- OpenCV with Matlab
- http://xanthippi.ceid.upatras.gr/people/evangelidis/matlab_opencv/
  - Guide to building OpenCV Mex files for windows that can be called in Matlab.
  - Note that the Guide is for OpenCV 2.1.
  - Macs: Tested using OSX 10.8, Xcode 4.5, OpenCV 2.4.2.
- Other options:
  - Use files for I/O to standalone OpenCV applications
  - Use OpenCV exclusively, with CGAL for potential matrix operations.

Reminders

- Midterm coming up, 48 hour take home.
- PS4 Released next week.
- Next TA session: Midterm Review