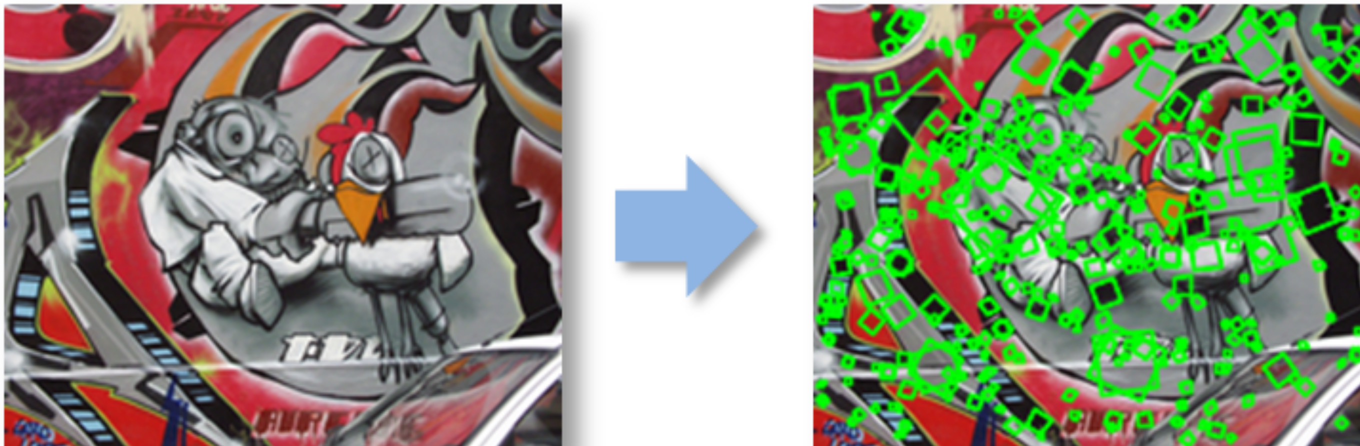
CSCI 497P/597P: Computer Vision



Lecture 10: Image Features

Harris Detector: Computing it

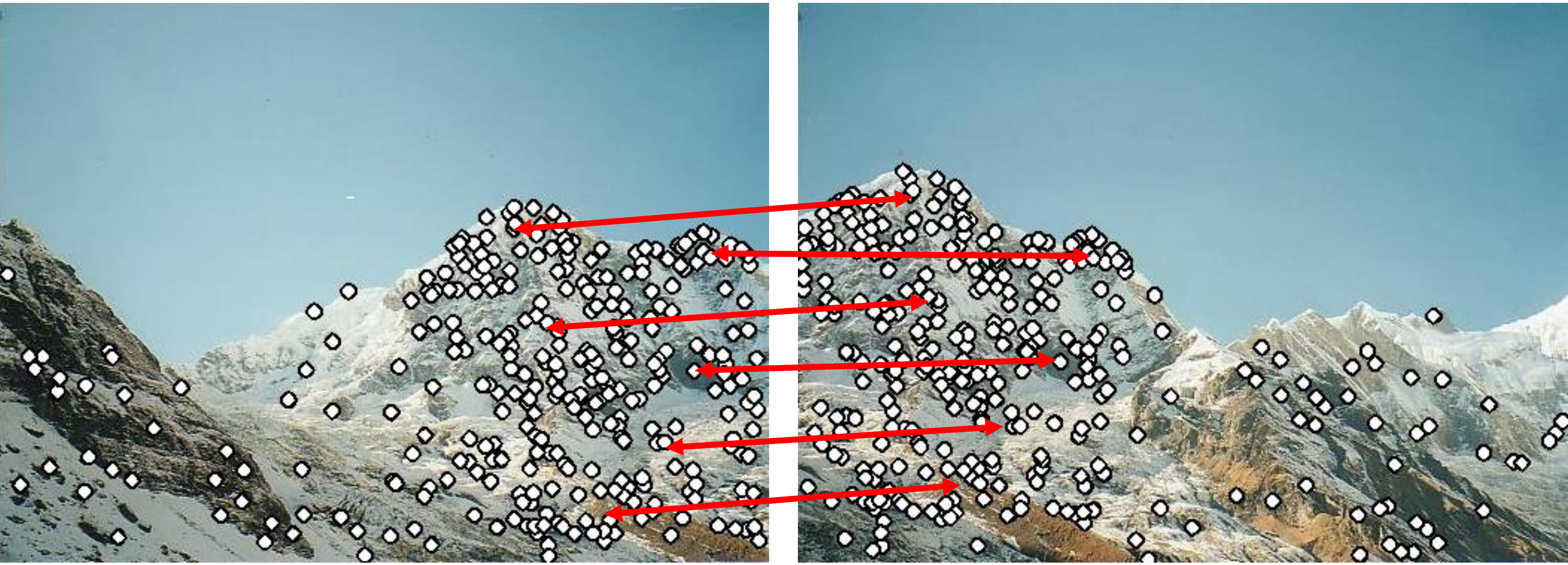
Feature Descriptors

Announcements

Goals

- Know how to compute Harris corners
- Understand the concept of invariance as pertains to detectors and descriptors
- Understand how to compute the MOPS feature descriptor
- Understand the gist of the SIFT descriptor

Running motivational example: Panorama Stitching



too long ; didn't math

Harris Corners: TL;DM

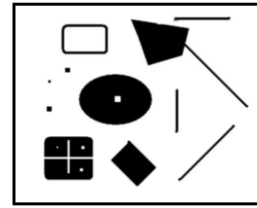
- **Goal**: Find "unique" patches.
  Proxy for uniqueness: not like neighboring patches
- ~~**Approach**: Eigenanalysis of SSD error on locally-linearized image windows.~~
- **Upshot**:
  The smaller eigenvalue of this 2x2 matrix indicates cornerishness:

$$H = \begin{bmatrix} \sum_{(x,y)\in W} I_x^2 & \sum_{(x,y)\in W} I_x I_y \\ \sum_{(x,y)\in W} I_x I_y & \sum_{(x,y)\in W} I_y^2 \end{bmatrix}$$

I_x = x gradient $\frac{\partial I}{\partial x}$

W = window

Harris Corners: TL;DM



Algorithm:

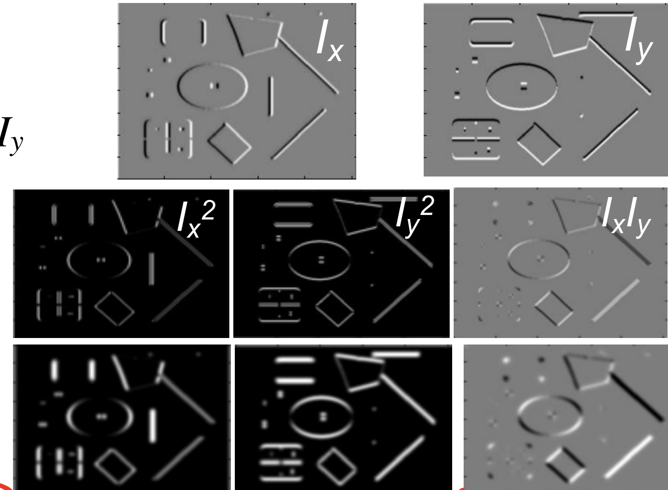
Convolution x2

1. Use Sobel filter to estimate gradients I_x, I_y

2. Compute I_x^2, I_y^2, I_xI_y np ops

Convolution x3

3. Filter each with a K x K mean filter



$A \; \sum_{x,y \in W} I_x^2 \qquad C \; \sum_{x,y \in W} I_y^2 \qquad B \; \sum_{x,y \in W} I_x I_y$

4. Approximate smallest eigenvalue as:
   det(H) / tr(H)

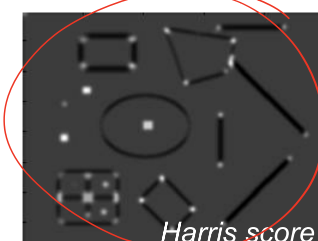
np ops

$$h = \frac{AC - BB}{A + C}$$

$$\begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

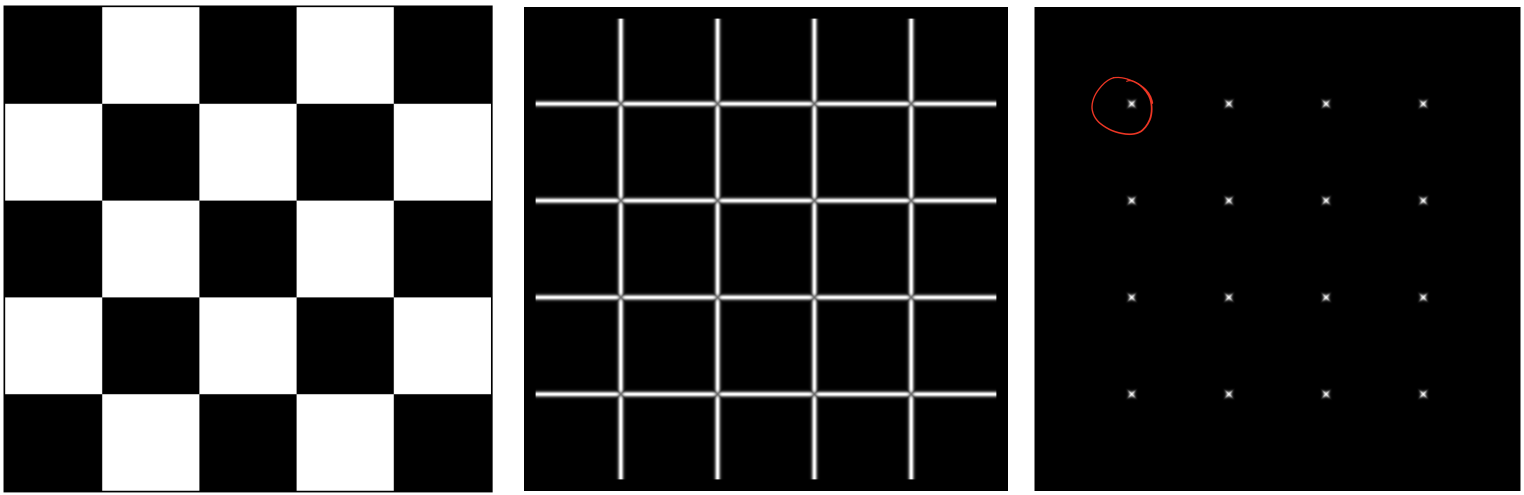
det(H) = A*C - B*B

tr(H) = A+C



Corner Detection: Upshot

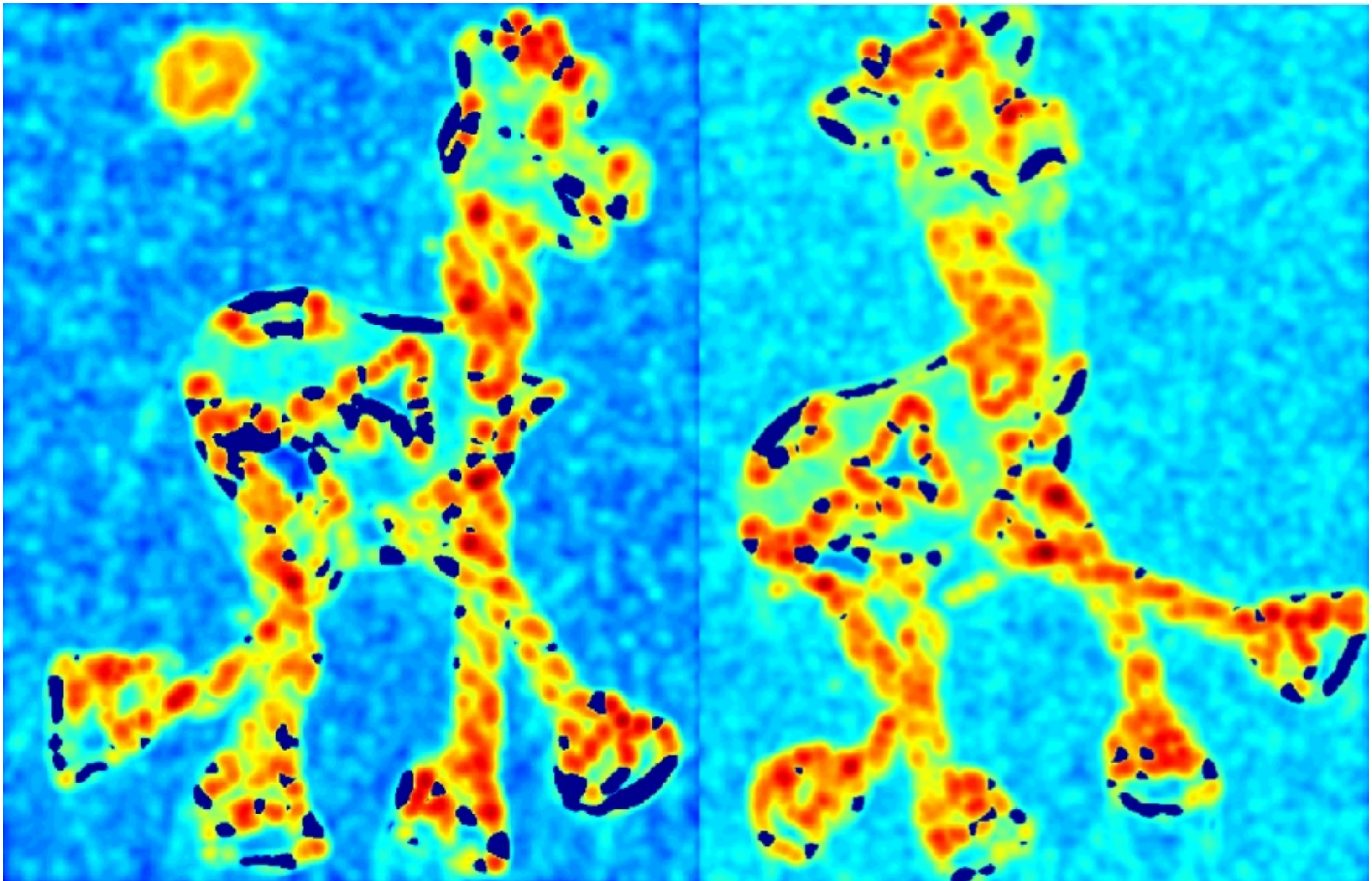
- The **smaller** eigenvalue of H is **large** when the patch is centered on a corner.



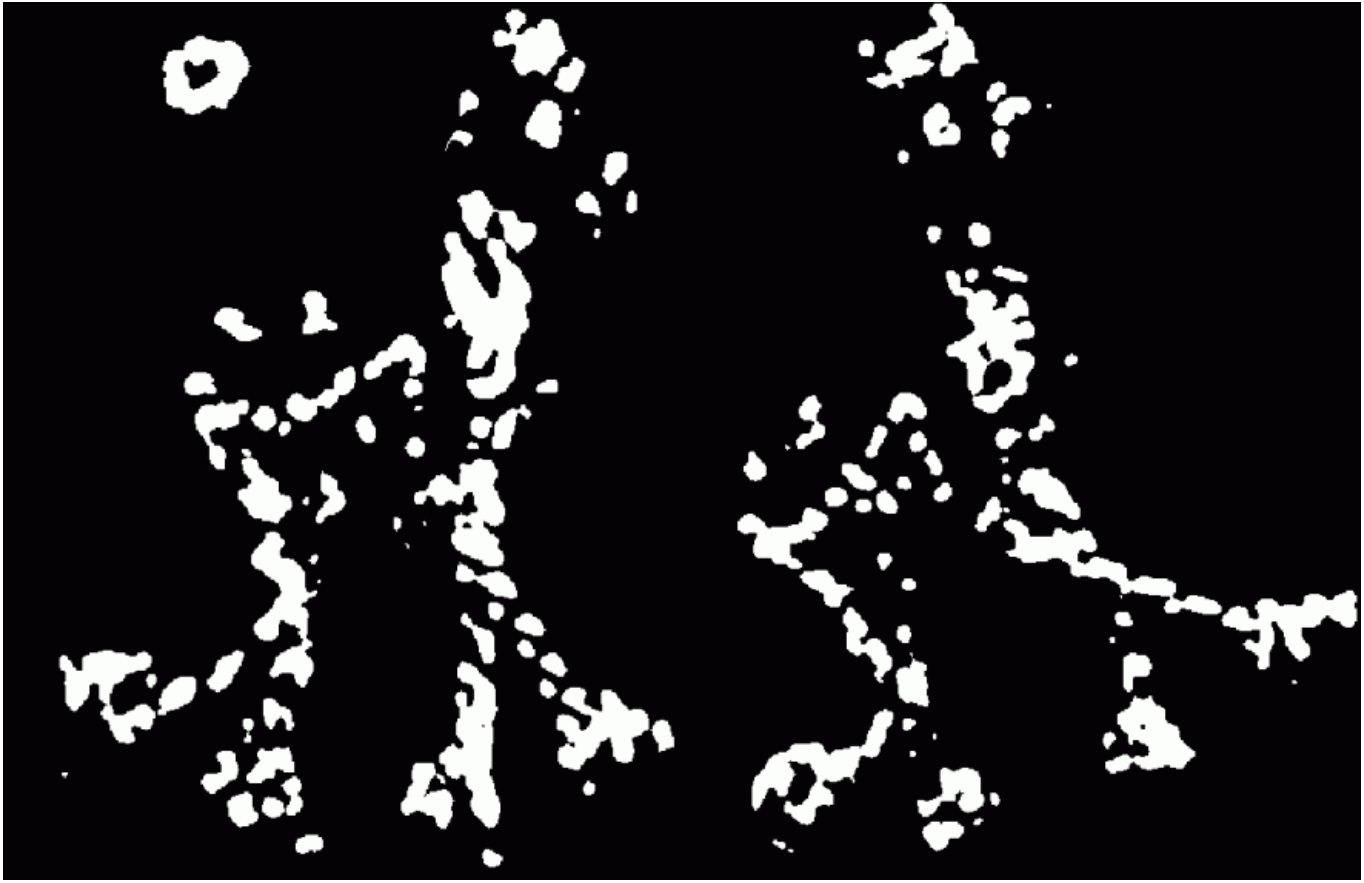
I $\lambda_{\max}$ $\lambda_{\min}$

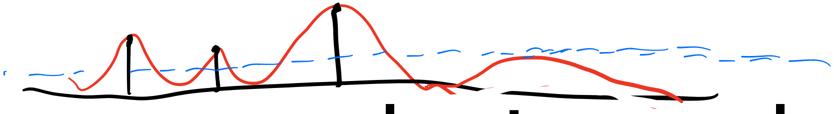
Input

Smallest eigenvalue

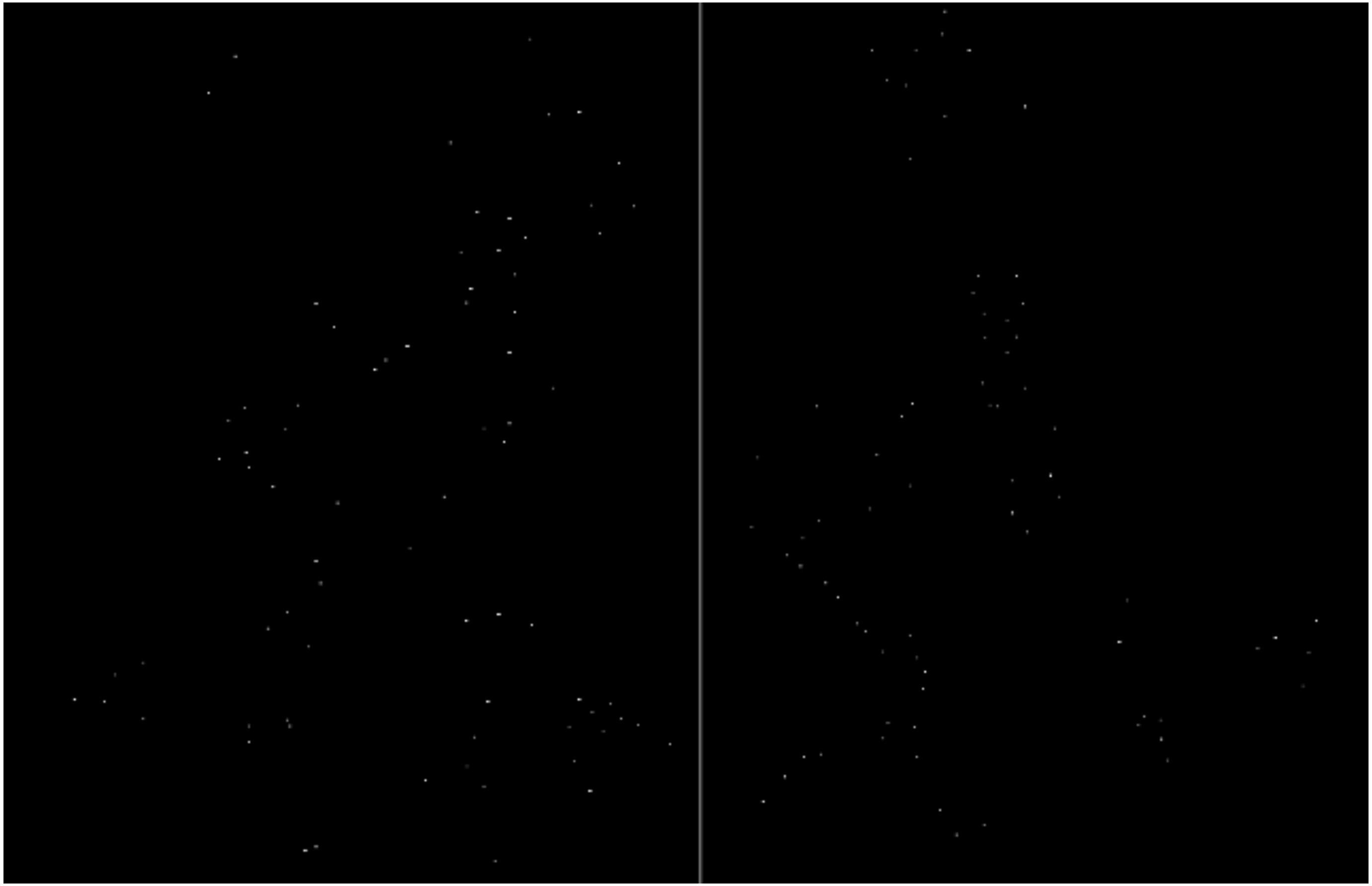


Thresholded

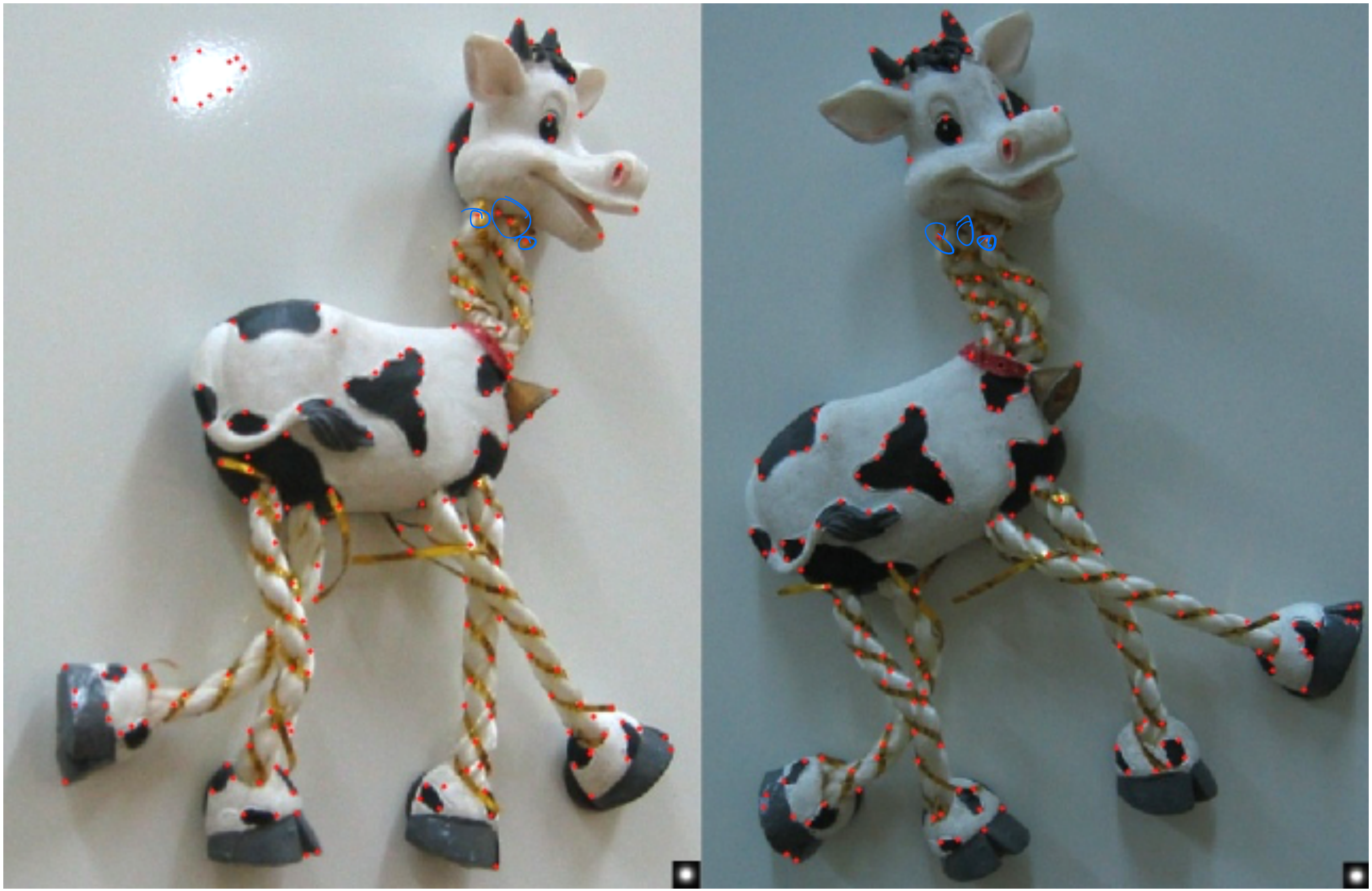




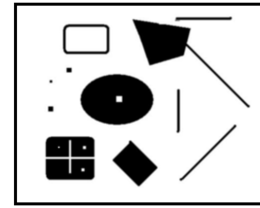
Keep only Local Maxima



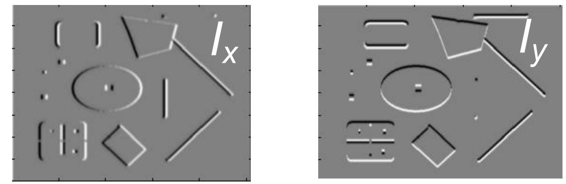
Resulting Corners



Harris Corners: TL;DM

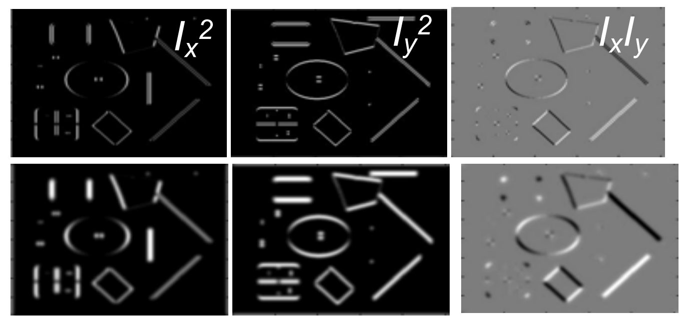


Algorithm:



1. Use Sobel filter to estimate gradients I_x, I_y

2. Compute $I_x{}^2, I_y{}^2, I_xI_y$



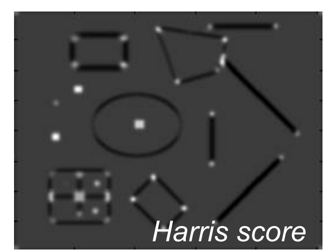
3. Filter each with a K x K mean filter

$$\sum_{x,y \in W} I_x^2 \qquad \sum_{x,y \in W} I_y^2 \qquad \sum_{x,y \in W} I_x I_y$$

4. Approximate smallest eigenvalue as:
   det(H) / tr(H)

5. Threshold   h[h<t]=0

6. Maximum filter   np.maximum_filter



Two desirable properties:

- Uniqueness: features **shouldn't** match if they're from different points in the scene.

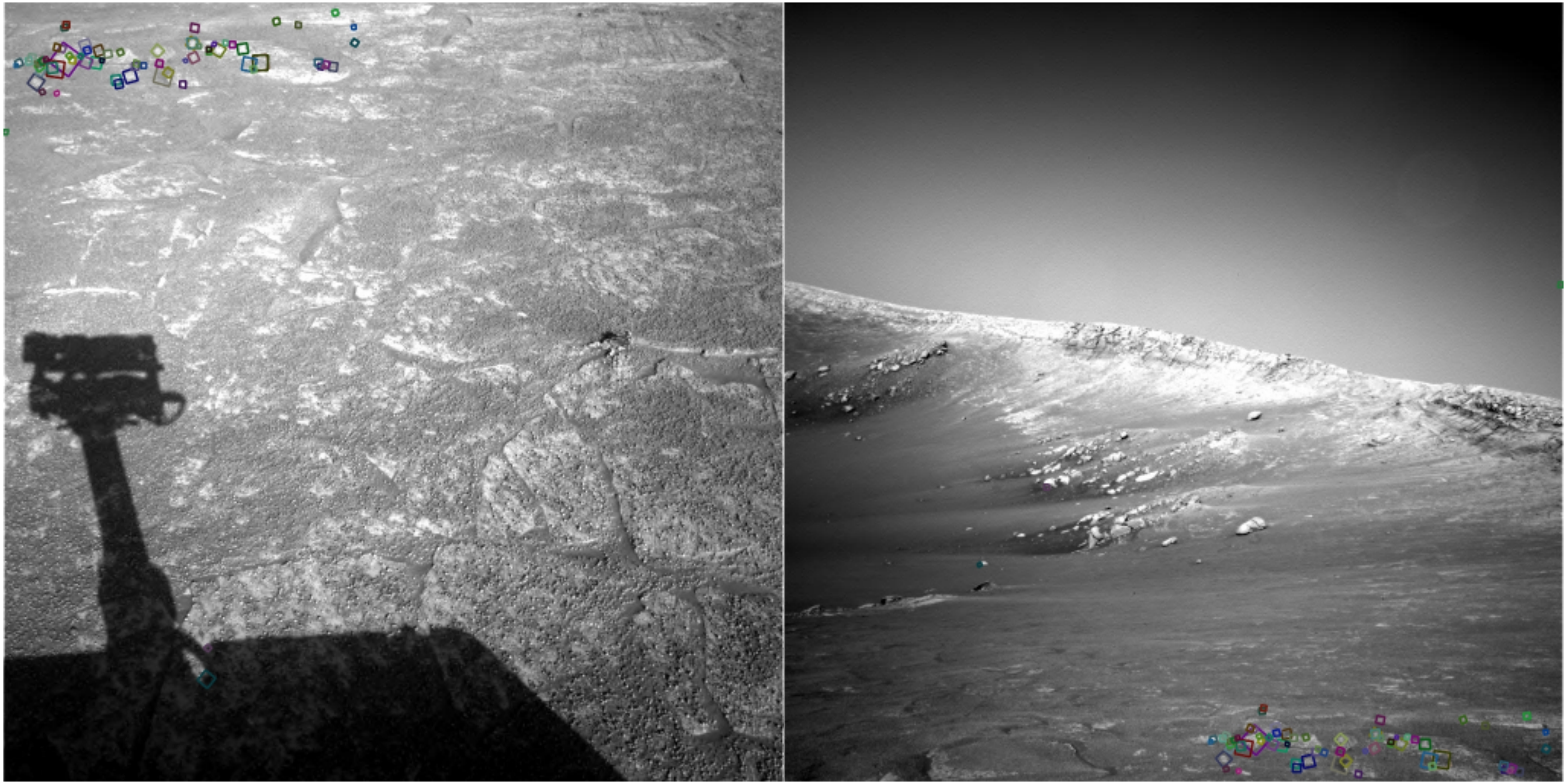
- Invariance: features **should** match if they do come from the same point in the scene.

Invariance

Invariance: Hard mode

Invariance: Mars Mode

Invariance: Mars Mode



Invariance

socrative.com
Room: CSCI497P

- Suppose we're comparing two images of the same scene. What kinds of transformations could relate the two images if:
  - They are part of a panorama sequence?
  - They were taken at different times of day?
  - They were taken by different cameras?
  - They were taken from different viewpoints?

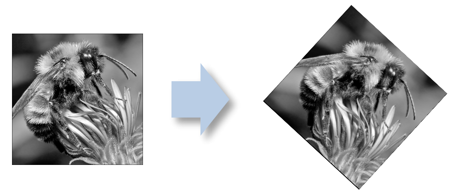
translation
rotation

brightness/exposure

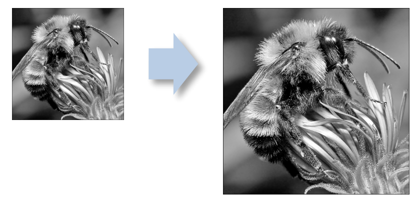
intensity changes
color
lighting, shadows
focal length(?)
noise
magnification
viewpoint-dependent appearance

Desirable Invariances

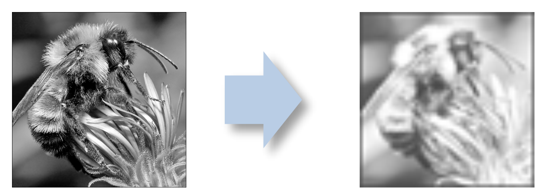
- Geometric transformations



  - Rotation

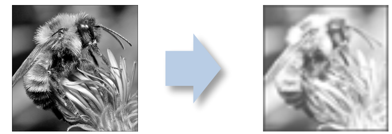


  - Translation
  - Scale
  - Skew
- Photometric transformations:
  - brightness shift $I \leftarrow I + 20$
  - brightness scale $I \leftarrow I \cdot 1.02$
  - contrast

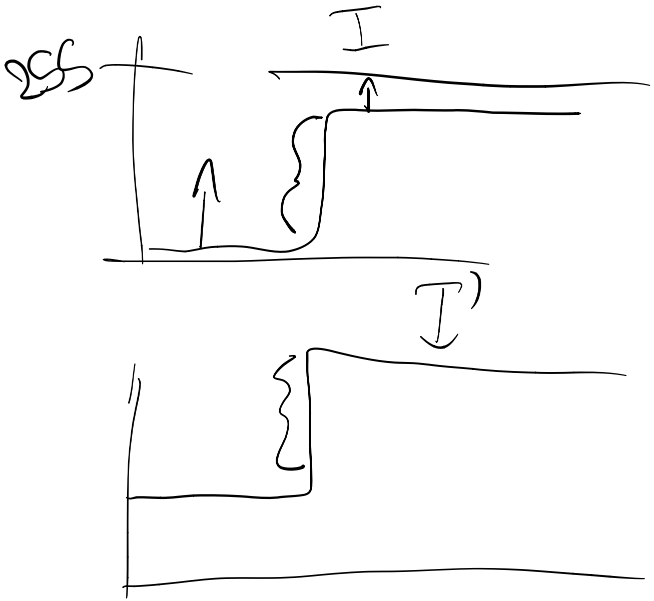


Harris detector: invariance

- Invariant to intensity shift?



$$I' = I + 20$$

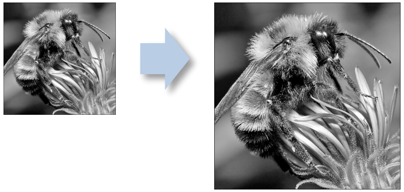


yes! (ish)

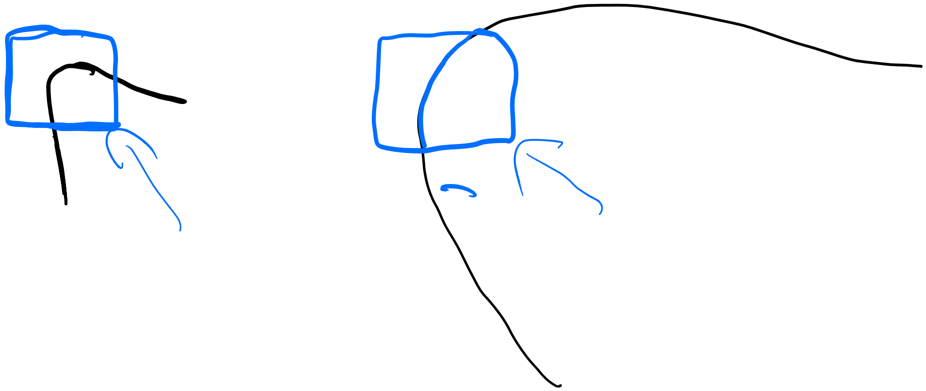
Harris detector: invariance

- Invariant to scaling?

no

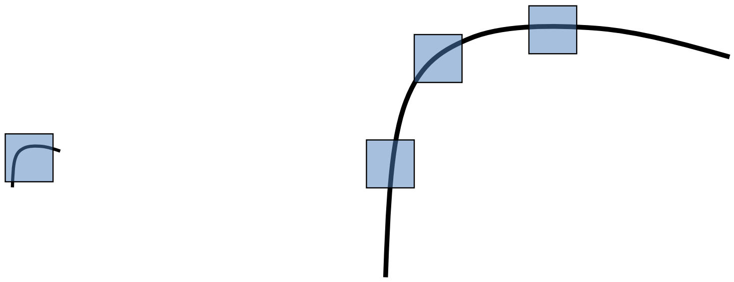


I



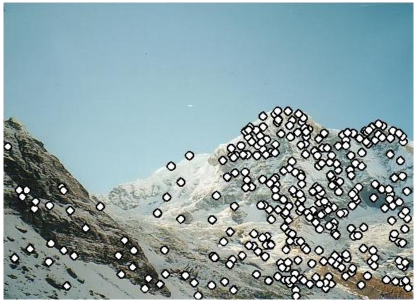
Harris detector: invariance

- Invariant to scaling?



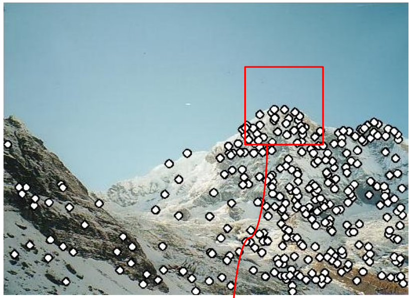
Features - Overview

1. Detect



detect *unique* points

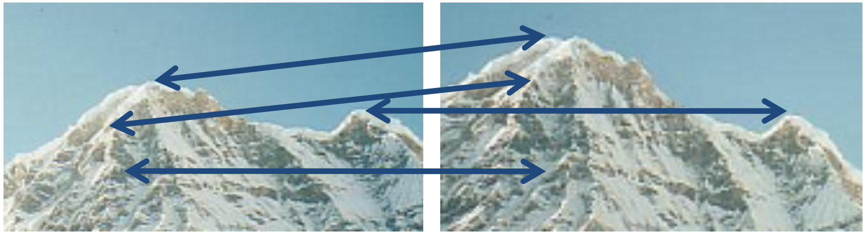
2. **Describe**



describe using *invariant* representation

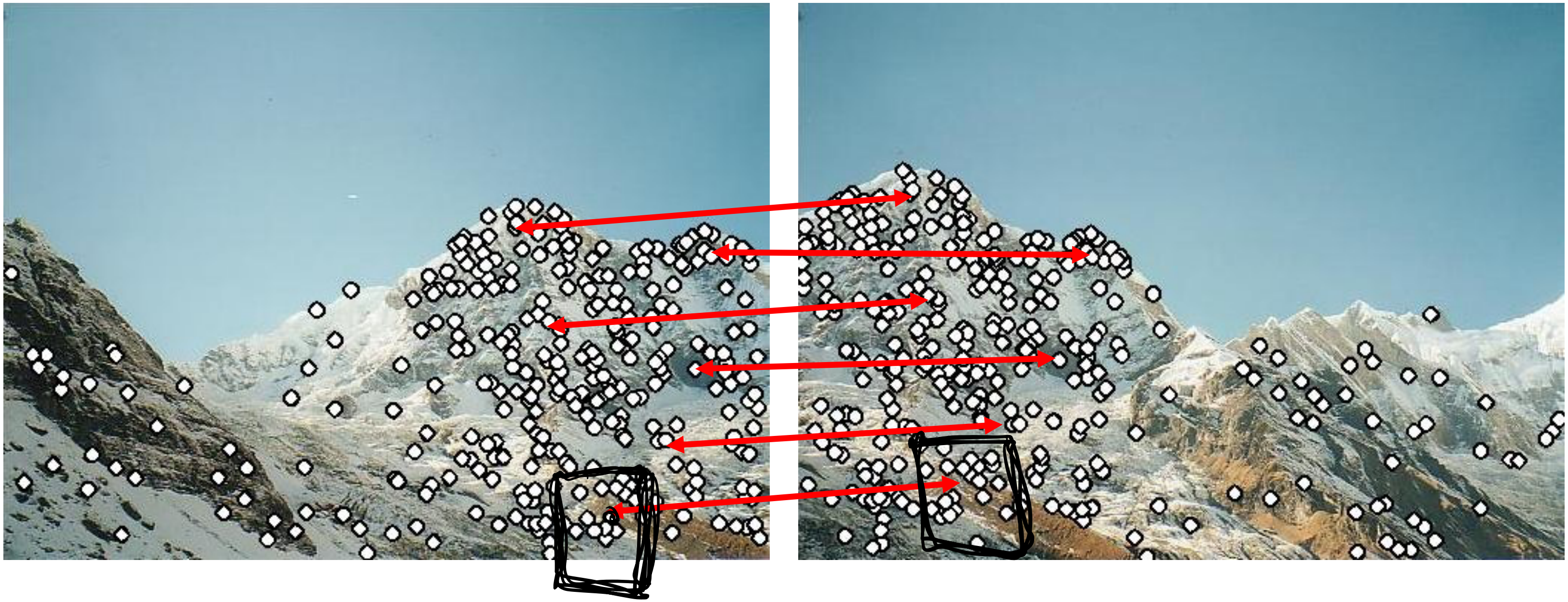
$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3. Match

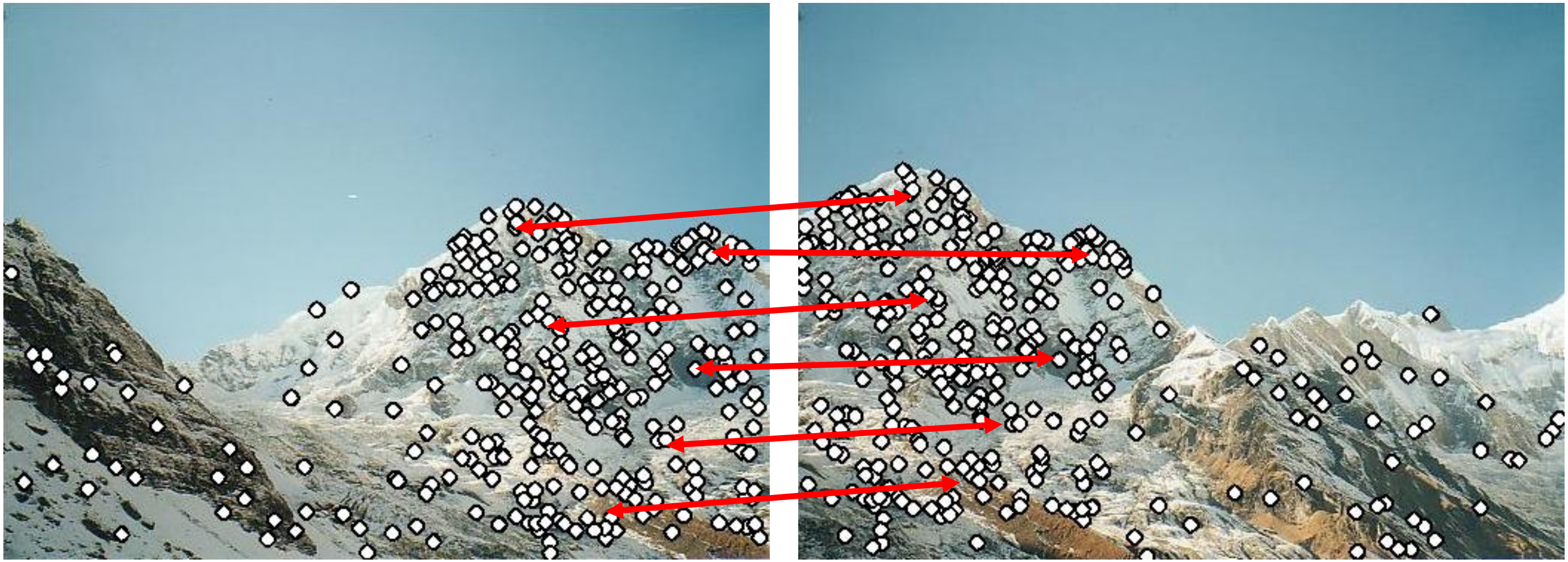


match them *robustly*

Feature Descriptors



Feature Descriptors



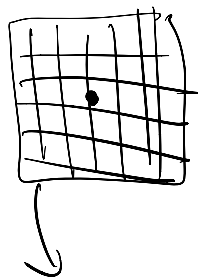
Simple starting point: window of pixels around the point.

Feature Descriptors

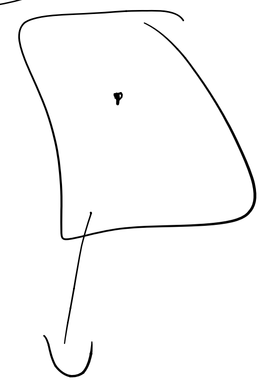
- Starting with a window of pixels, let's add invariances:
    - Brightness (scale and/or shift - "*affine* invariance")

$$x_2 = x_1 \cdot \underline{s} + \underline{t}$$

$$\frac{x_1 - \mu_{x_1}}{\sigma_{x_1}}$$



$\vec{x}_1$



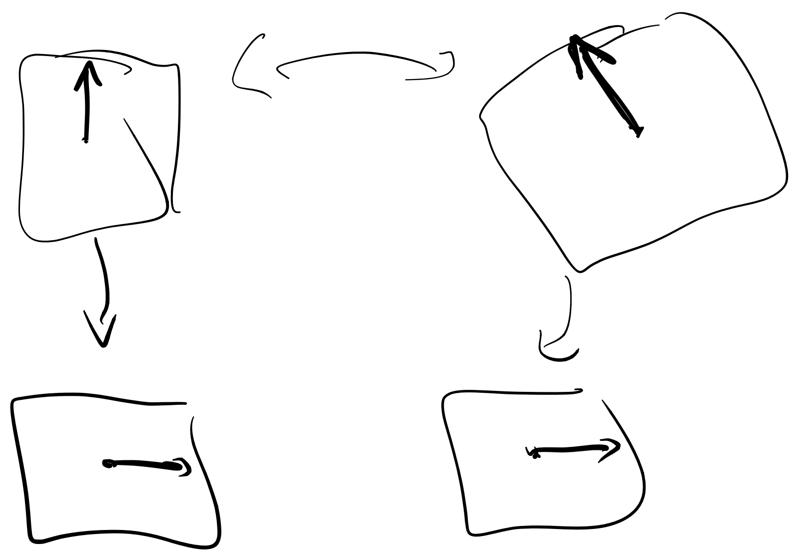
$\vec{x}_2$

$$\frac{x_2 - \mu_{x_2}}{\sigma_{x_2}}$$

$$\underline{\| x_1 - x_2 \|^2}$$

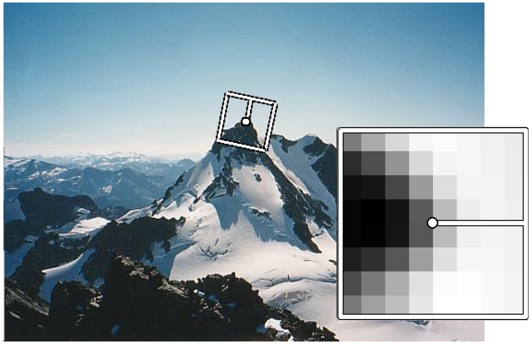
Feature Descriptors

- Starting with a window of pixels, let's add invariances:
  - Brightness (scale and/or shift - "*affine* invariance")
  - Rotation



Feature Descriptors

- Starting with a window of pixels, let's add invariances:
  - Brightness (scale and/or shift - "*affine* invariance")
  - Rotation

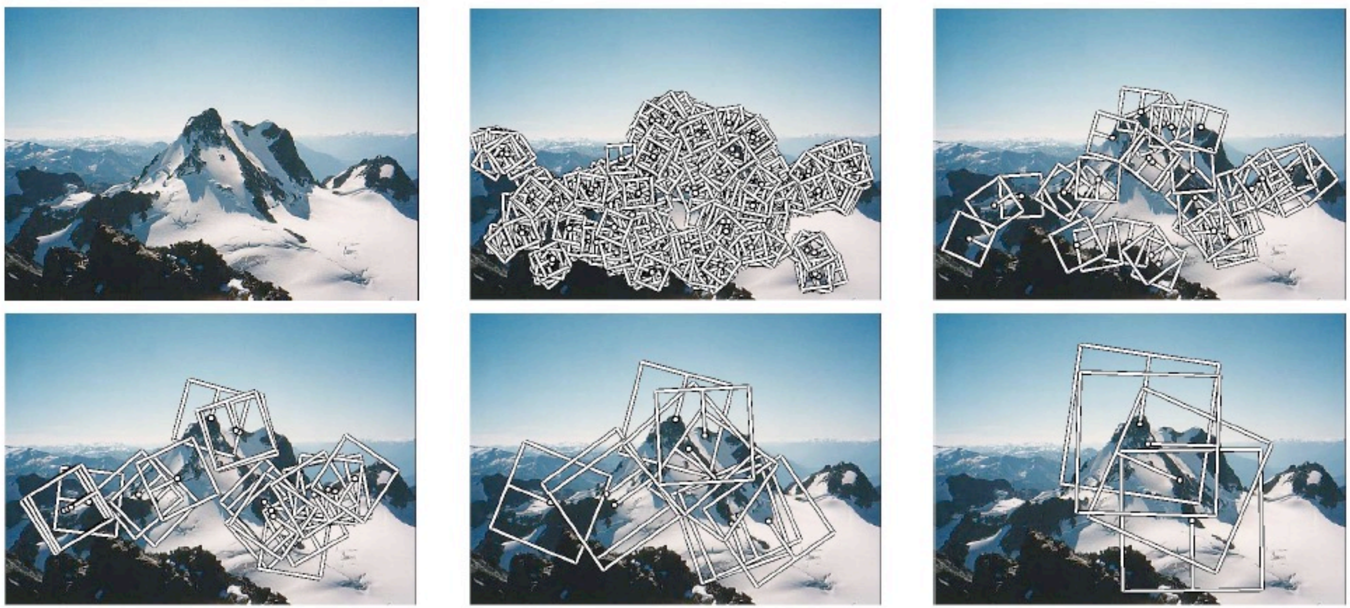


Feature Descriptors

- Starting with a window of pixels, let's add invariances:
    - Scale

Feature Descriptors

- Starting with a window of pixels, let's add invariances:
  - Scale

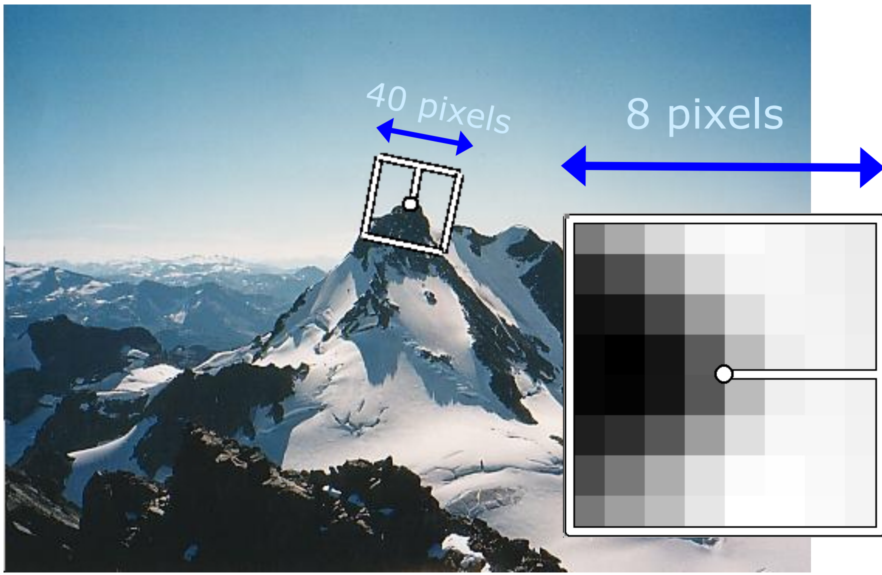


Feature Descriptors

- Starting with a window of pixels, let's add invariances:
  - Scale

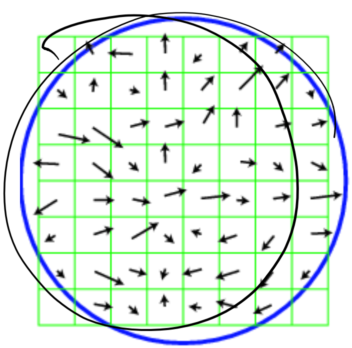
Multiscale Oriented PatcheS: The MOPS Descriptor

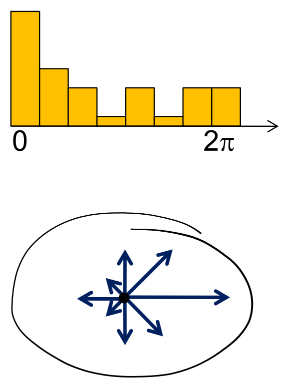
- Scale to 1/5 size
- Rotate to horizontal
- Normalize intensity:
  - subtract mean
  - divide by standard dev
- Run it on a Gaussian pyramid



Fancy, industrial-strength feature descriptors: SIFT

- Take a 16x16 window
- Compute edge orientation at each pixel
- Discard weak edges
- Create a histogram of remaining edge orientations

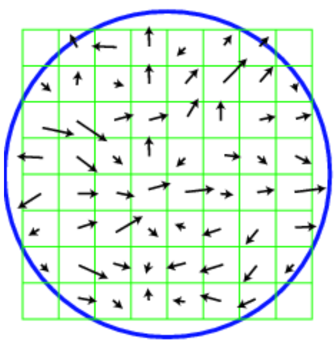


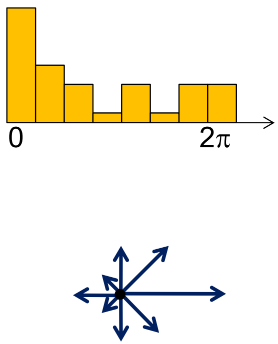


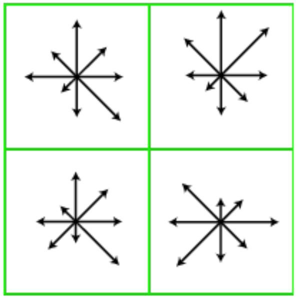
Scale Invariant Feature Transform: SIFT

Real-deal, industrial-strength feature descriptors.

- Take a 16x16 window
- Compute edge orientation at each pixel
- Discard weak edges
- Create a histogram of remaining edge orientations
- Actually do this for each of 4 quadrants of the window

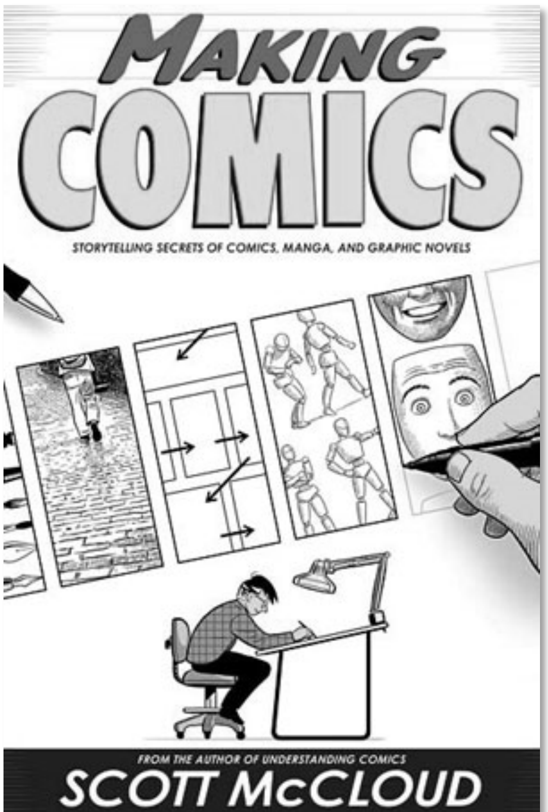




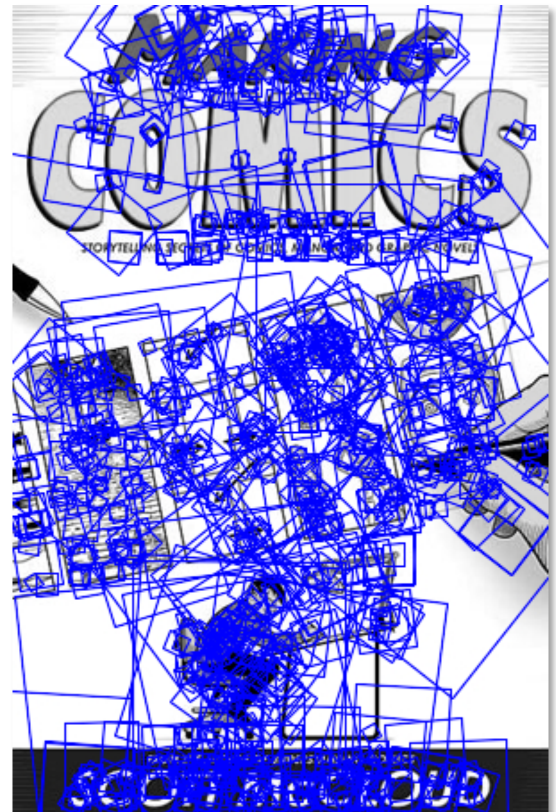


4 histograms - unroll into vector

SIFT: Example



sift



SIFT: Properties

Remarkably invariant to:

- Viewpoint, illumination, rotation, scale (via pyramid)