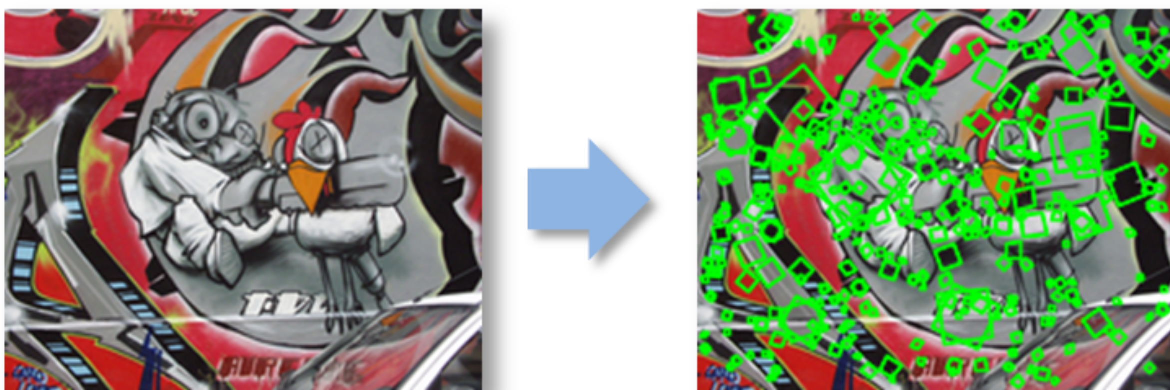
CSCI 497/597P: Computer Vision

Scott Wehrwein

Feature Detection:
Descriptors and Matching



Reading

- Szeliski: 4.1.2, 4.1.3

Happenings

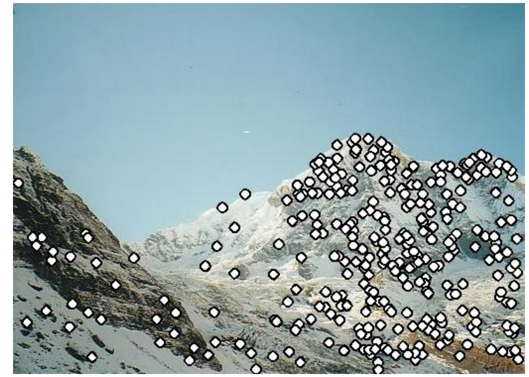
- Tuesday, 1/22 – CS SMATE Faculty Candidate **Gerald Raj**: Research Talk – 4 pm in CF 316
- Tuesday, 1/22 – ACM Research Talk: Computer Vision with Dr. Scott Wehrwein – 5 pm in CF 316
- Wednesday, 1/23 – CS SMATE Faculty Candidate **Gerald Raj**: Teaching Talk – 4 pm in CF 316
- Wednesday, 1/23 – Peer Lecture Series: CS Success Workshop – 5 pm in CF 420
- Thursday, 1/24 – CS SMATE Faculty Candidate **Cecily Heiner**: Research Talk – 4 pm in CF 316
- Friday, 1/25 – CS SMATE Faculty Candidate **Cecily Heiner**: Teaching Talk – 4 pm in **CF 115**

Goals

- Understand the invariance and covariance properties of the Harris operator
- Understand how to implement multi-scale keypoint detection and description.
- Understand the details of the MOPS feature descriptor
- Understand the general idea behind the SIFT feature descriptor

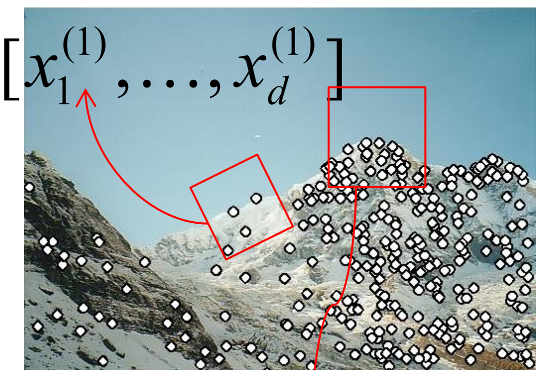
Local features: main components

1) **Detection:** Identify the interest points



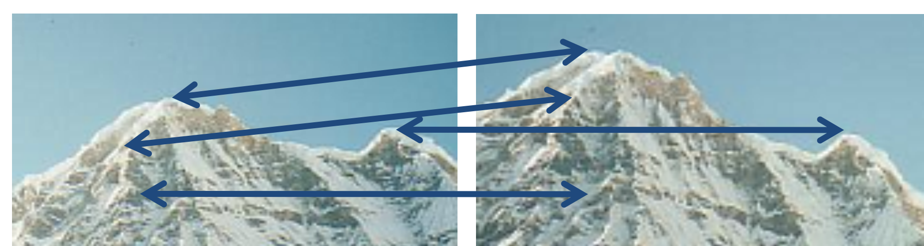
2) **Description:** Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



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

3) **Matching:** Determine correspondence between descriptors in two views



Kristen Grauman

What makes a good feature?

Invariance vs. uniqueness

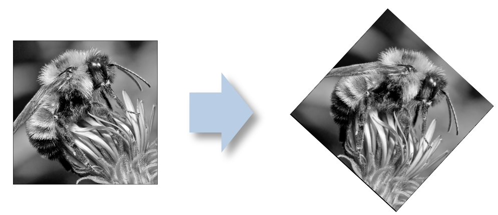
- Invariance:
  - Our ability to detect and match the feature shouldn’t change even if image is transformed

- Uniqueness:
  - The feature should be highly unique for each point: it should only match the same “real-world” feature.

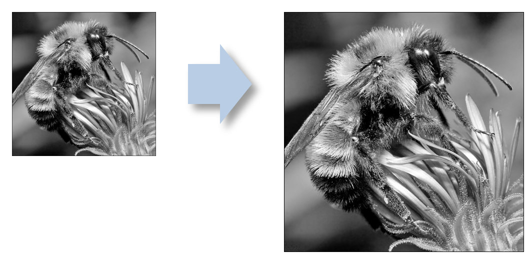
Image transformations

- Geometric

Rotation

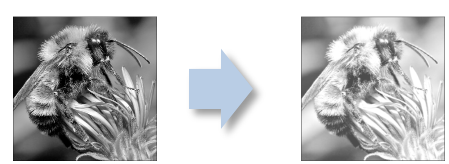


Scale



- Photometric

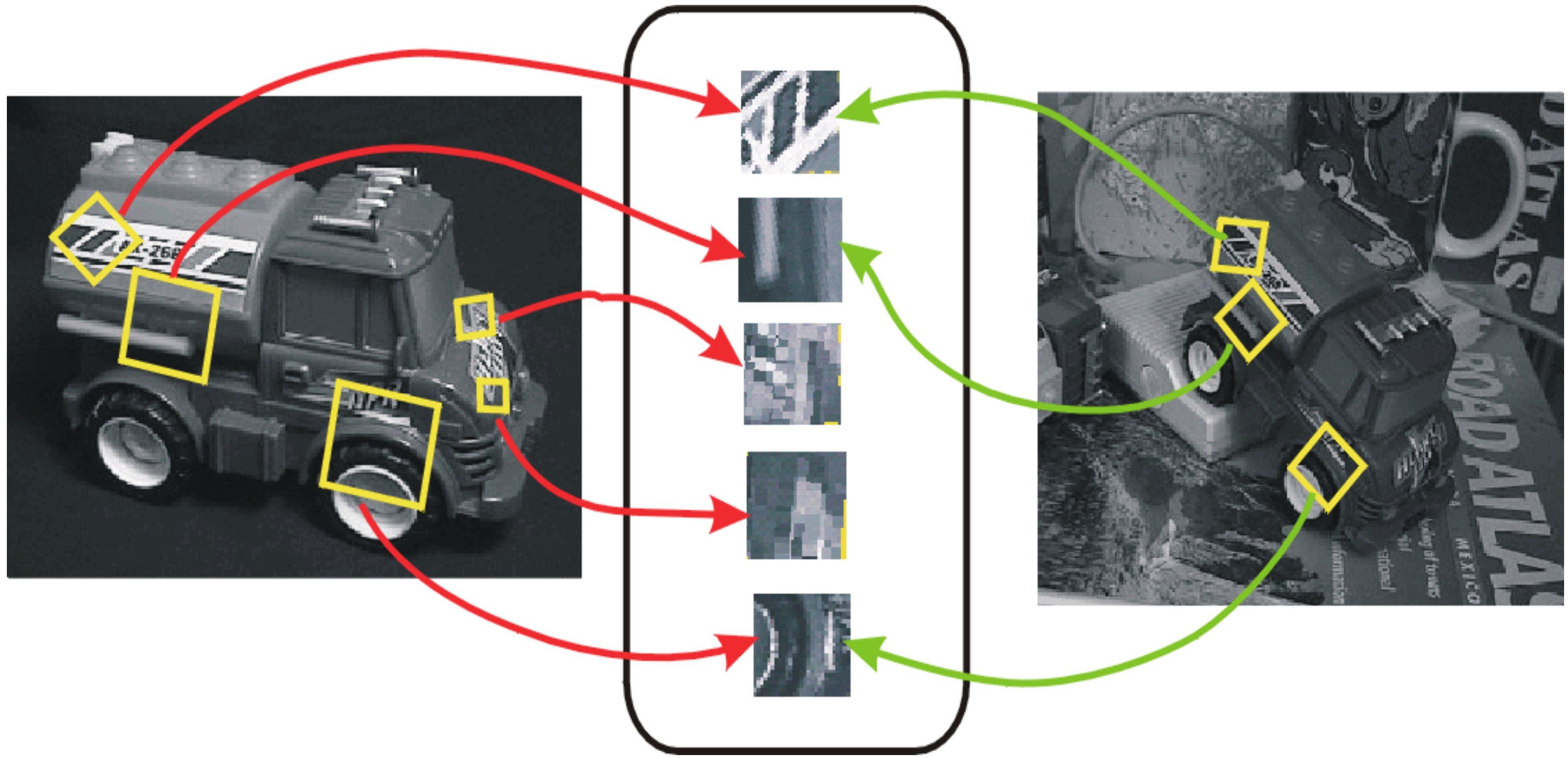
Intensity change



Invariance in local features

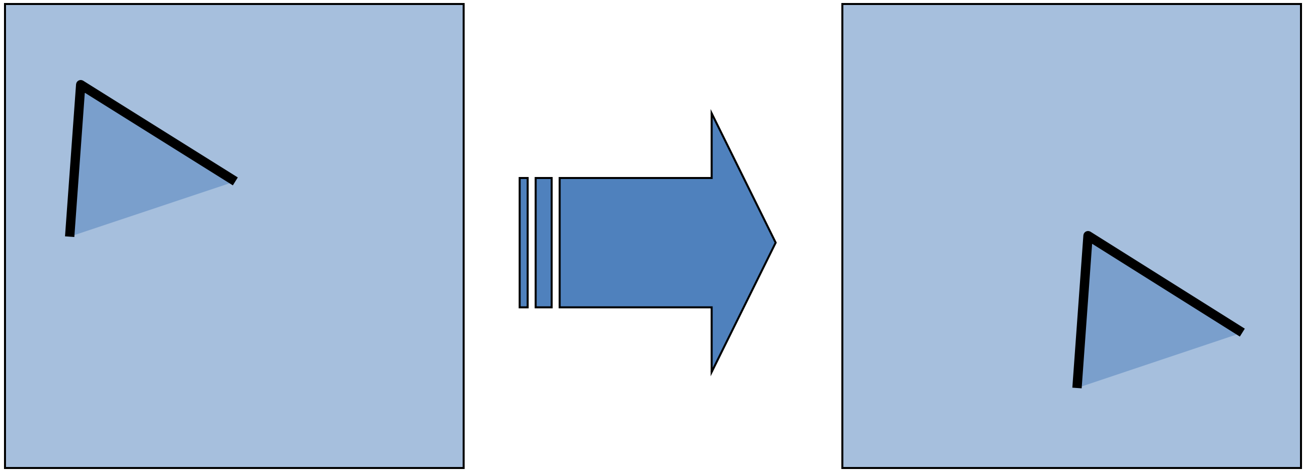
Ideally, we want to find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



Feature Descriptors

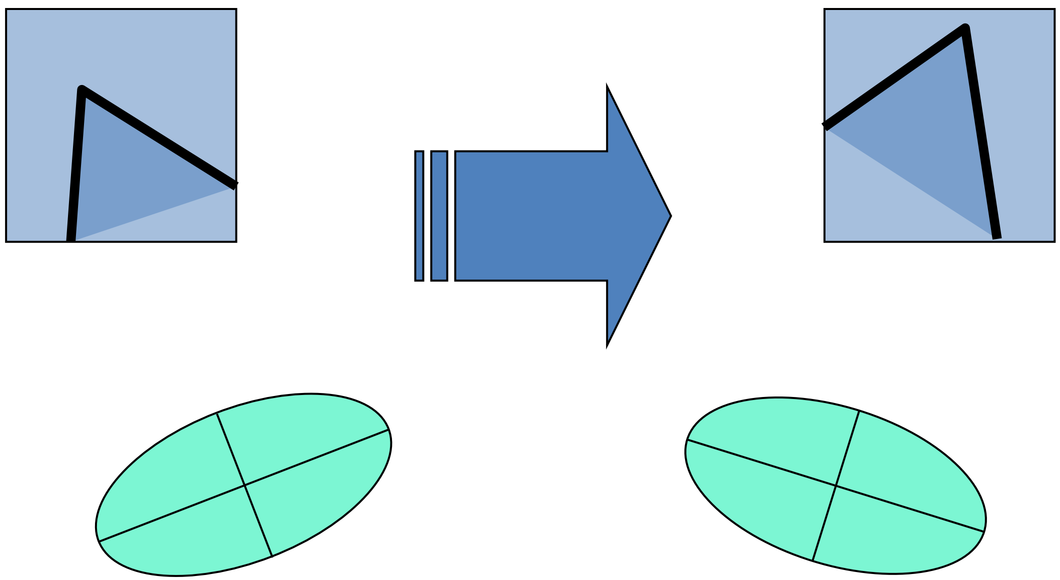
Harris detector: Invariance properties -- Image translation



- Derivatives and window function are shift-invariant

Corner location is covariant w.r.t. translation

Harris detector: Invariance properties -- Image rotation

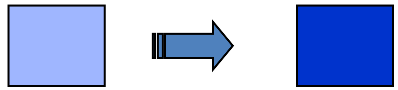


Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Or: *direction* of max and min error increase rotates, but *amount* does not.

Corner location is covariant w.r.t. rotation

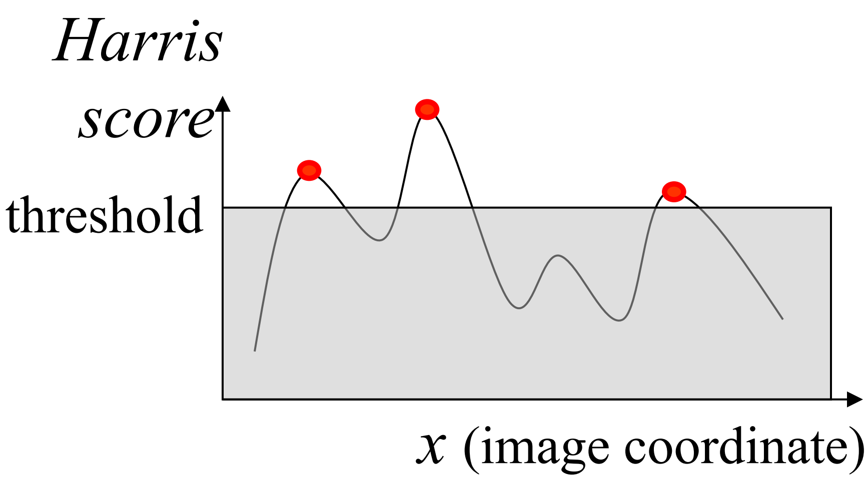
Harris detector: Invariance properties – Affine intensity change

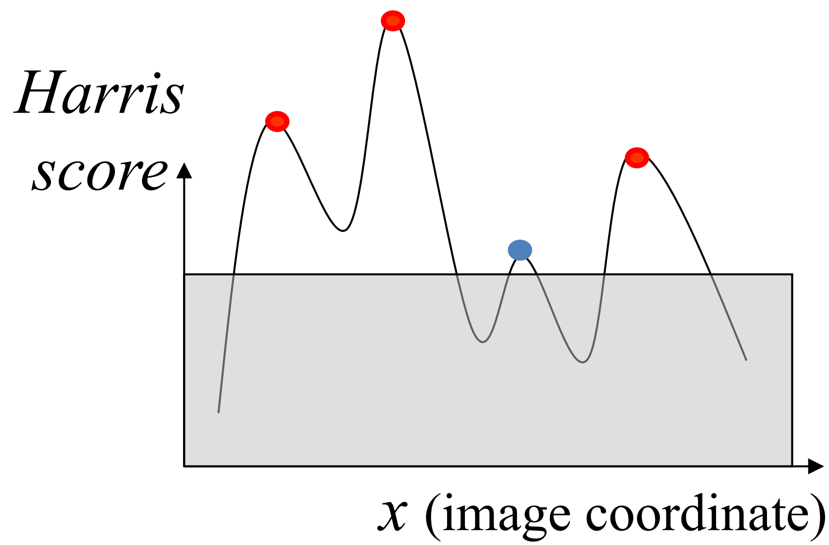


$$I \to a\,I + b$$

- How about intensity shift $I \to I + b$?

- What about intensity scaling: $I \to a\,I$

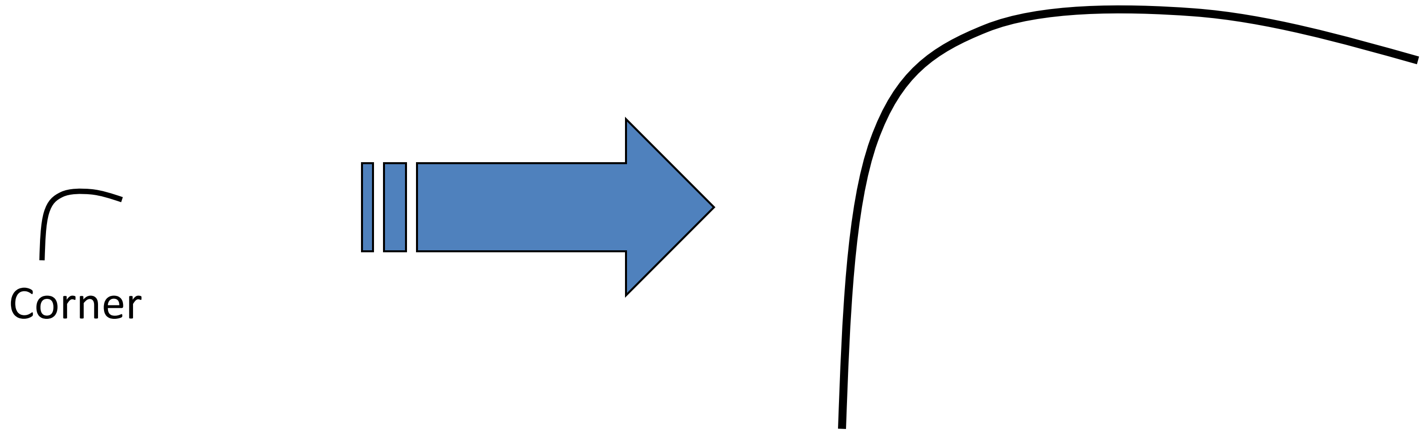




Partially invariant to affine intensity change

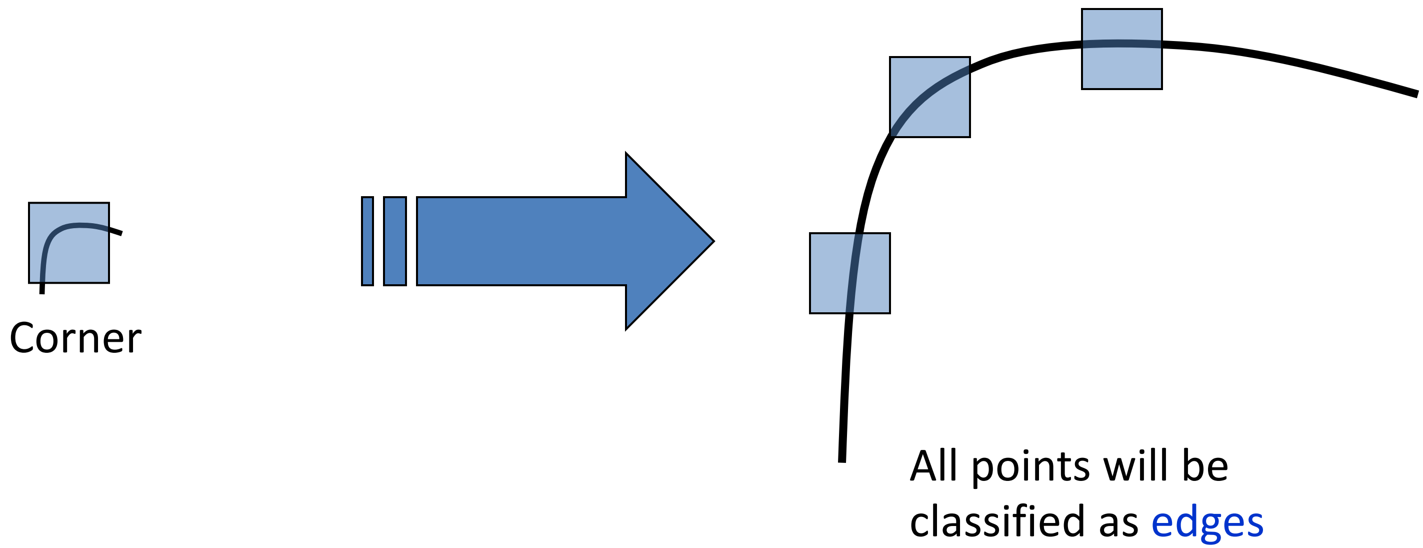
Harris Detector: Invariance Properties

- Invariant to scaling (e.g, upsampling?)



Harris Detector: Invariance Properties

- Invariant to scaling (e.g, upsampling?)

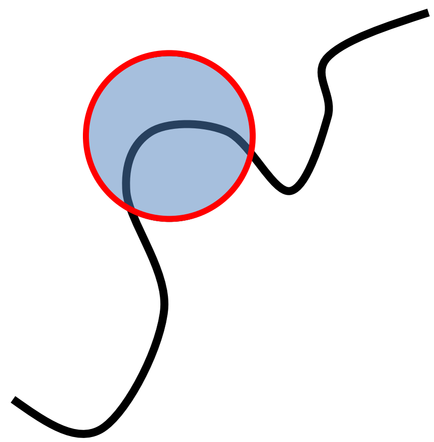


More on this next week…

Not invariant to scaling

Scale invariant detection

Suppose you're looking for corners:



Keypoint detection:

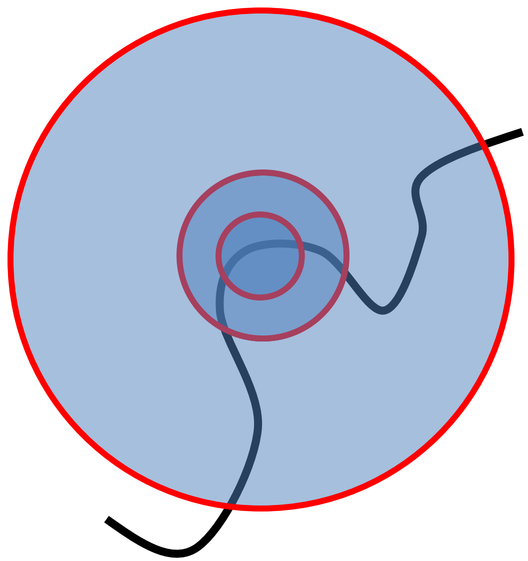
- find local maxima in H(x,y)

Scale-invariant keypoint detection:

- find local maxima in H(x,y,scale)

Scale invariant detection

Suppose you're looking for corners

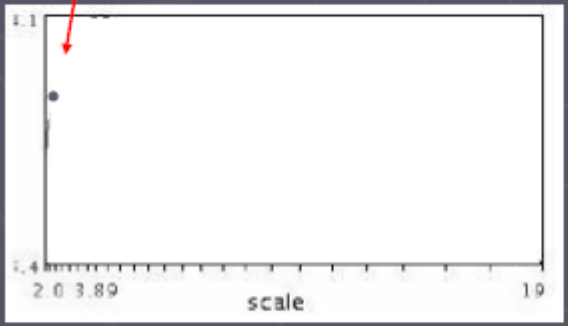


Key idea: find scale that gives local maximum of H(x,y)

- Now searching over both position and scale

Automatic scale selection

Lindeberg et al., 1996



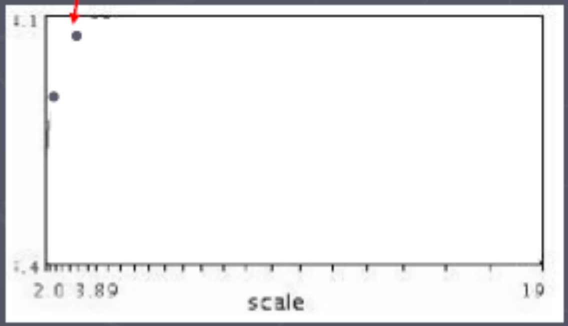
$f(I_{i_1 \dots i_m}(x,\sigma))$

Slide from Tinne Tuytelaars

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

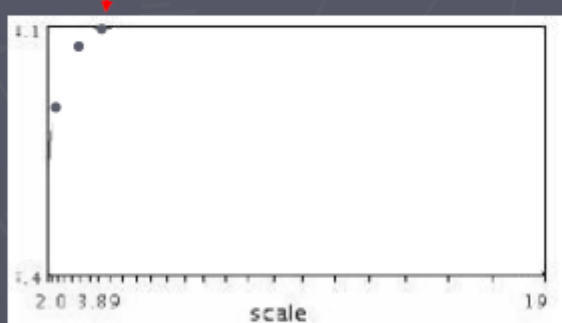


$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

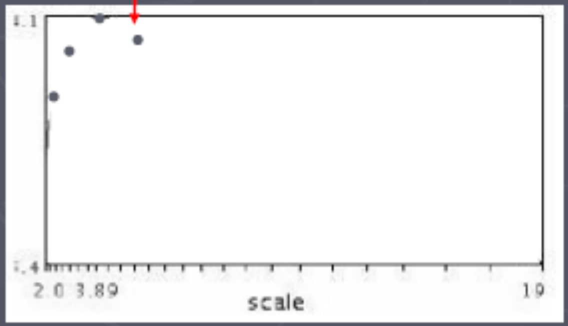


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

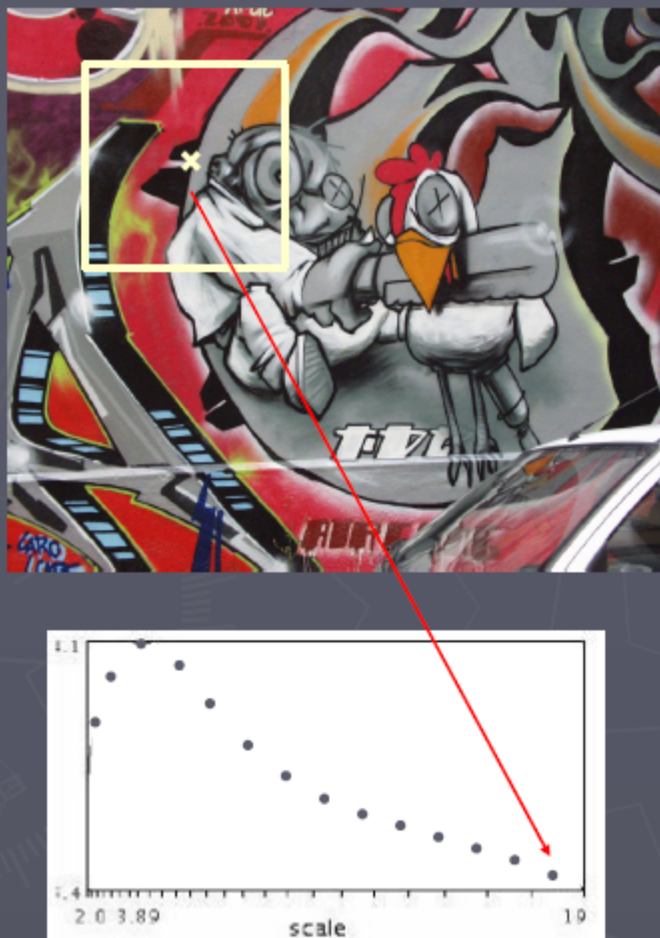


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

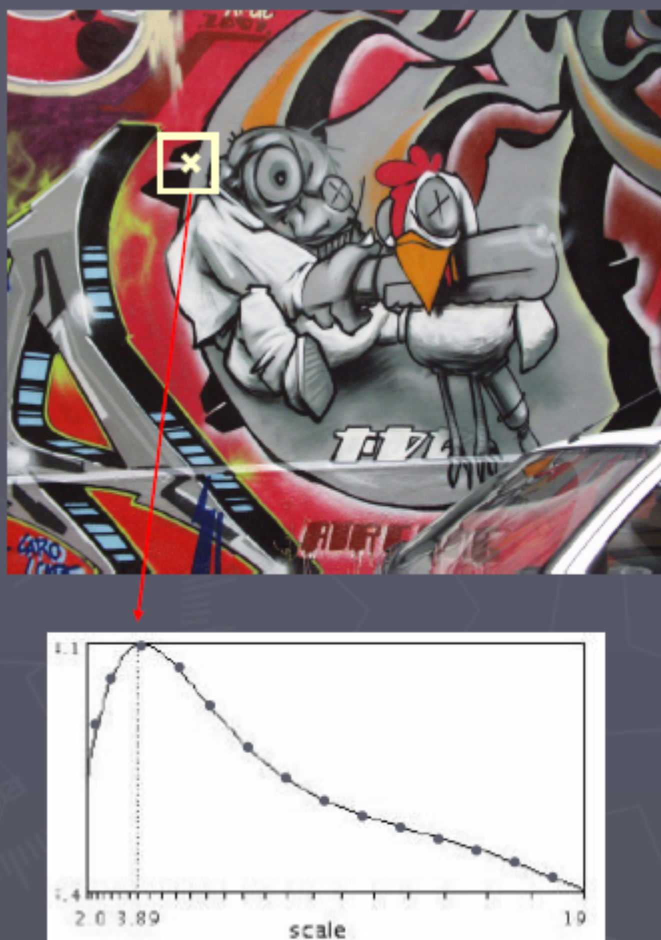


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection

Function responses for increasing scale

Scale trace (signature)

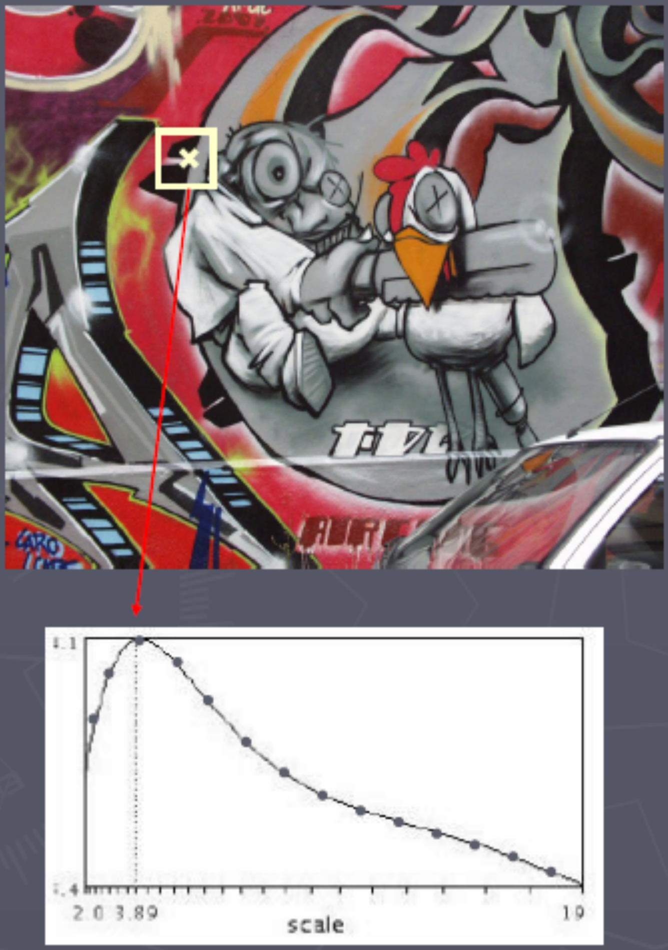


$f(I_{i_1 \ldots i_m}(x,\sigma))$

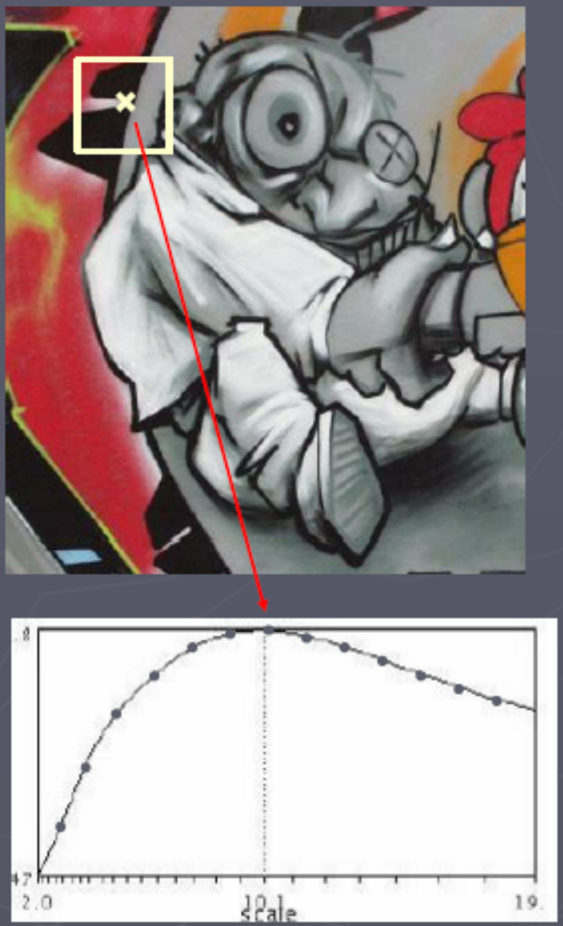
Automatic scale selection

Function responses for increasing scale

Scale trace (signature)



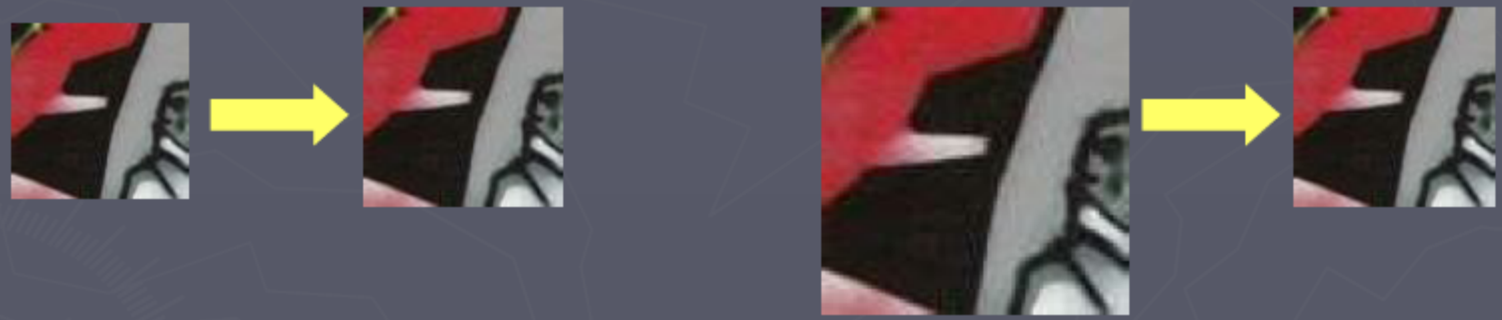
$f(I_{i_1 \ldots i_m}(x,\sigma))$



$f(I_{i_1 \ldots i_m}(x',\sigma'))$

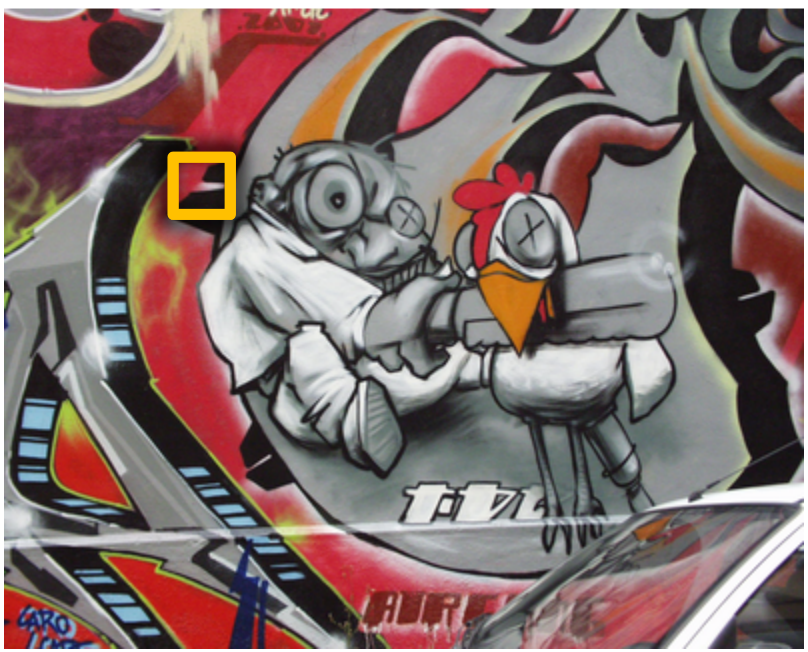
Automatic scale selection

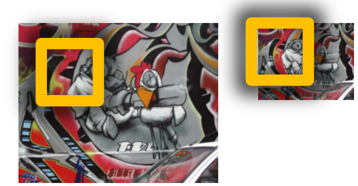
Normalize: rescale to fixed size



Implementation

- Instead of computing *H* for larger and larger windows, we can implement using a fixed window size with a Gaussian pyramid



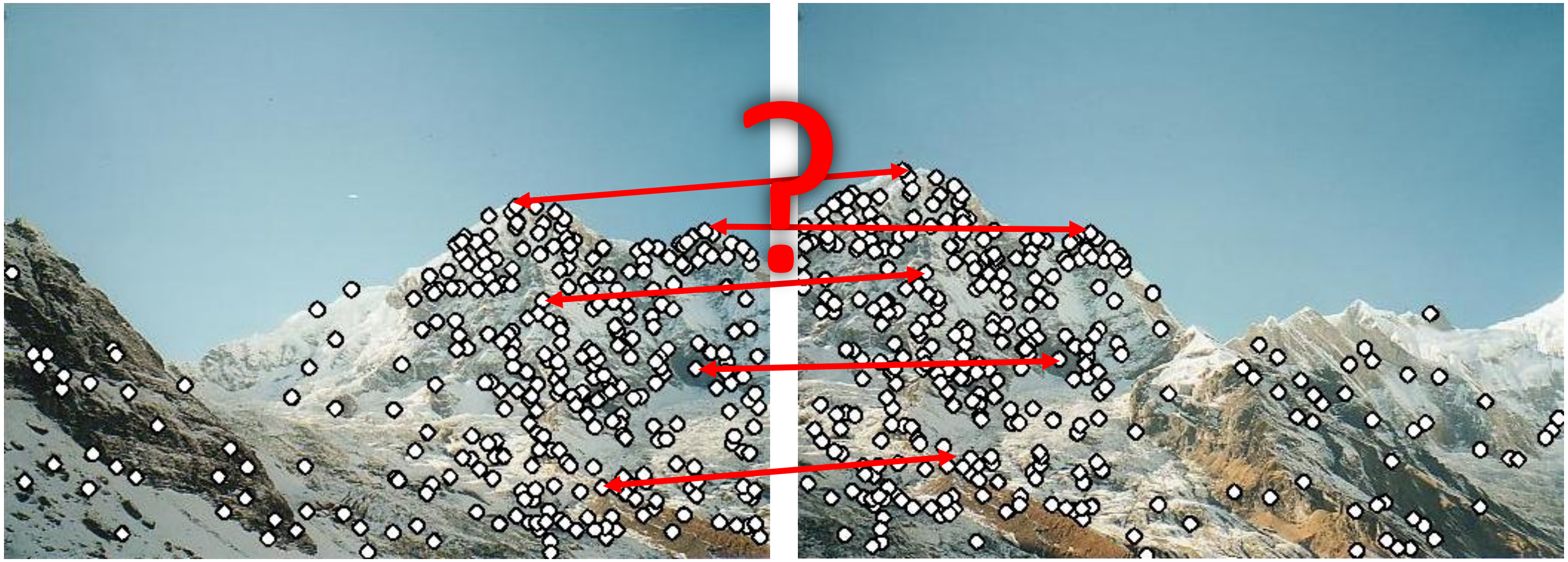


(may perform better with in-between levels, e.g. a ¾-size image)

Feature descriptors

We know how to detect good points

Next question: **How to match them?**

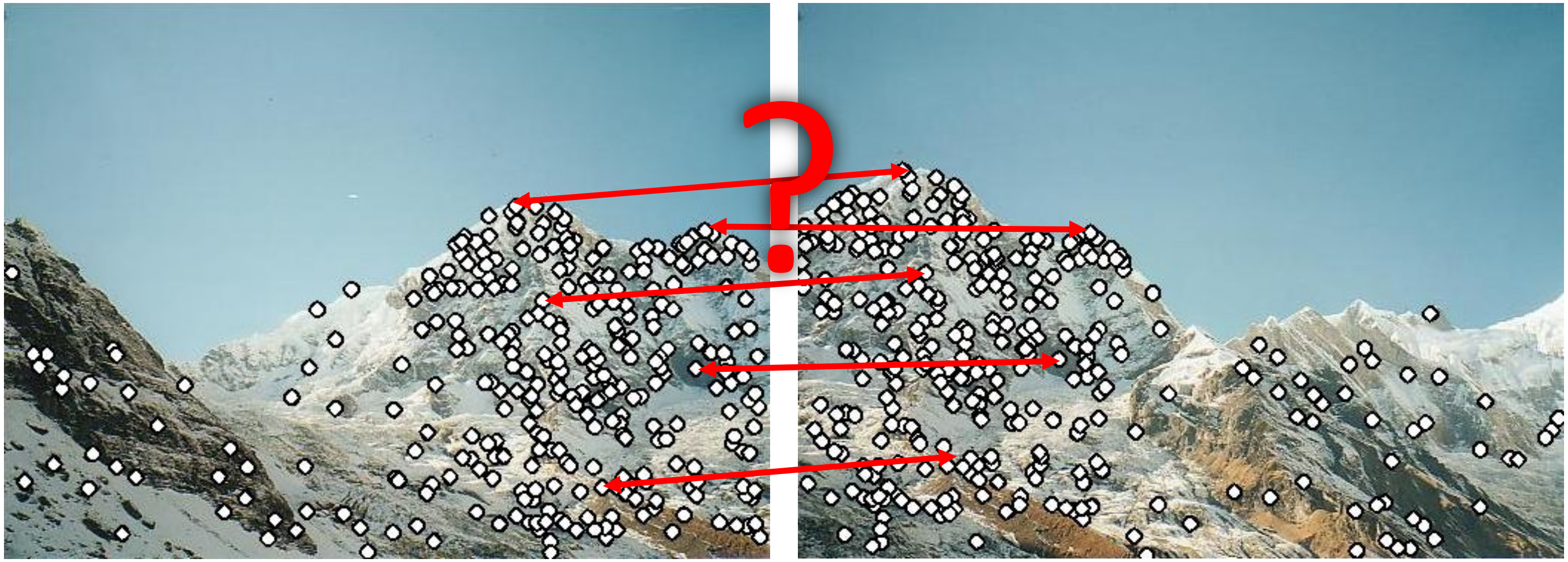


Answer: Come up with a *descriptor* for each point, find similar descriptors between the two images

Feature descriptors

We know how to detect good points

Next question: **How to match them?**



Lots of possibilities

- Simple option: match square windows around the point
- State of the art approach: SIFT
  - David Lowe, UBC http://www.cs.ubc.ca/~lowe/keypoints/

Rotation invariance for feature descriptors

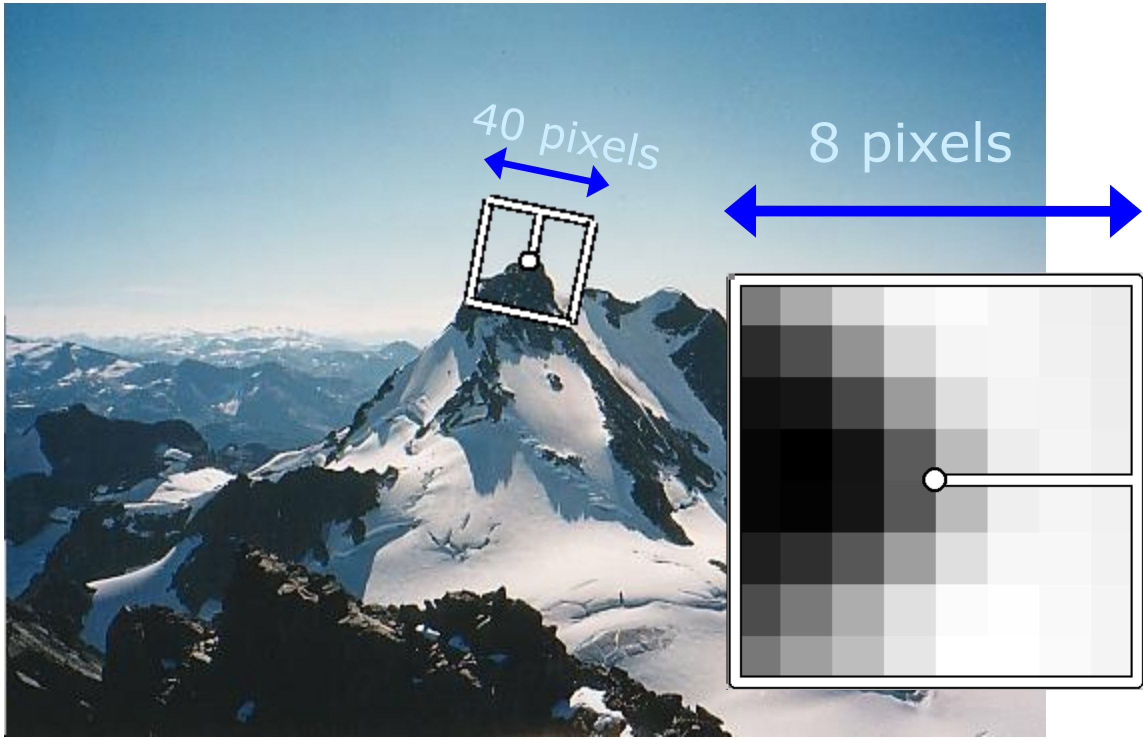
- Find dominant orientation of the image patch
  - E.g., given by $\mathbf{x}_{\mathbf{max}}$, the eigenvector of $\mathbf{H}$ corresponding to $\lambda_{\max}$ (the *larger* eigenvalue)
  - Or simply using the direction of the gradient.
  - Rotate the patch according to this angle

Figure by Matthew Brown

Multiscale Oriented PatcheS descriptor

Take 40x40 square window around detected feature, and:

1. Scale to 1/5 size (filter first to avoid aliasing!)
2. Rotate to horizontal
3. Sample 8x8 square window centered at feature
4. Normalize the intensity values by subtracting the mean and dividing by the standard deviation in the window



Adapted from slide by Matthew Brown

Detections at multiple scales

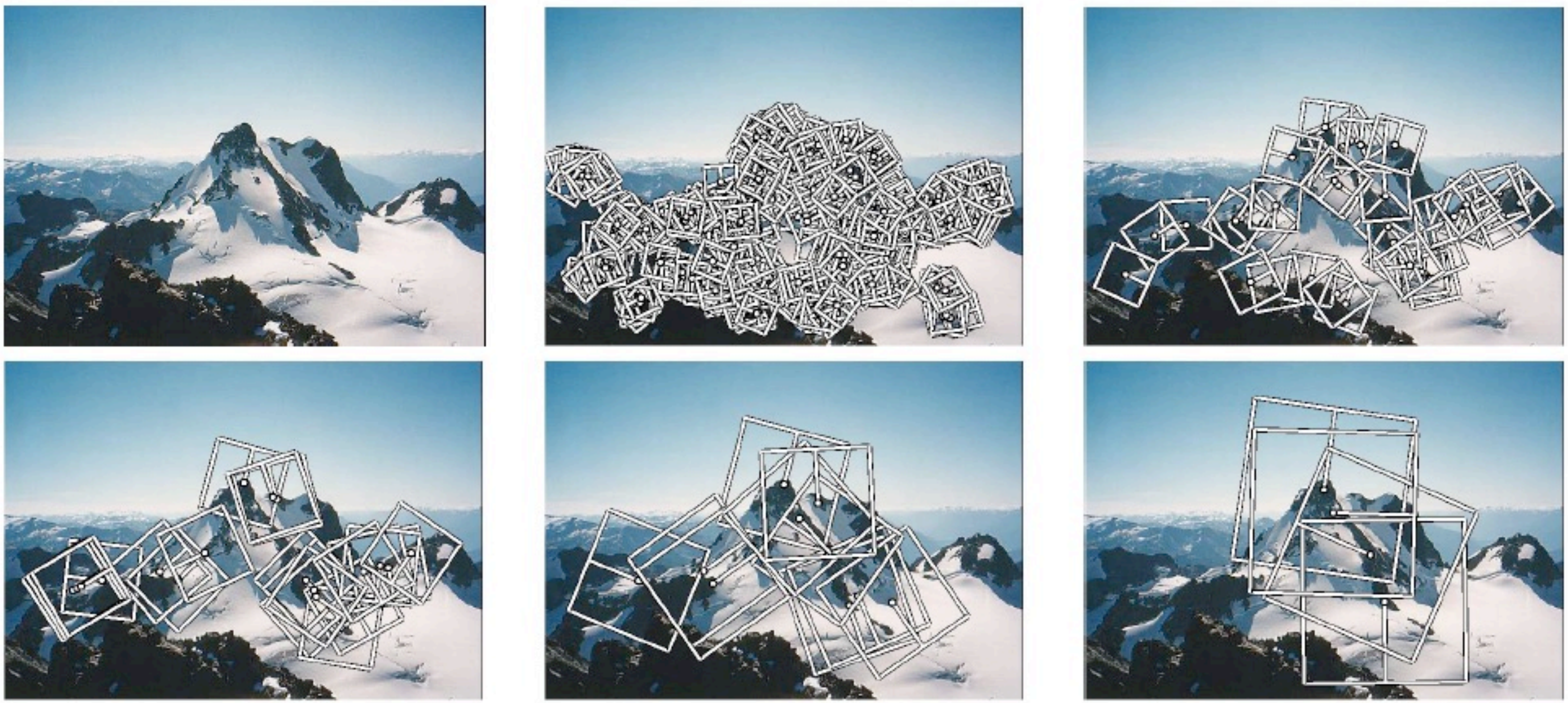
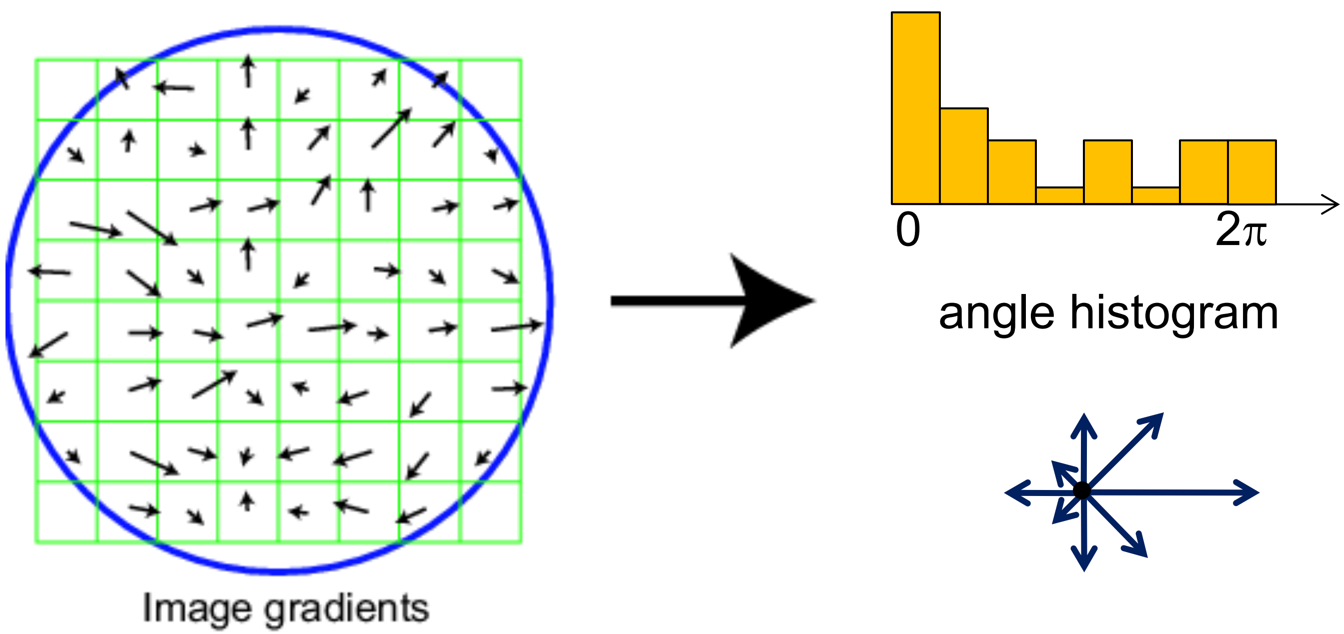


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

Scale Invariant Feature Transform

Basic idea behind the SIFT **descriptor**:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

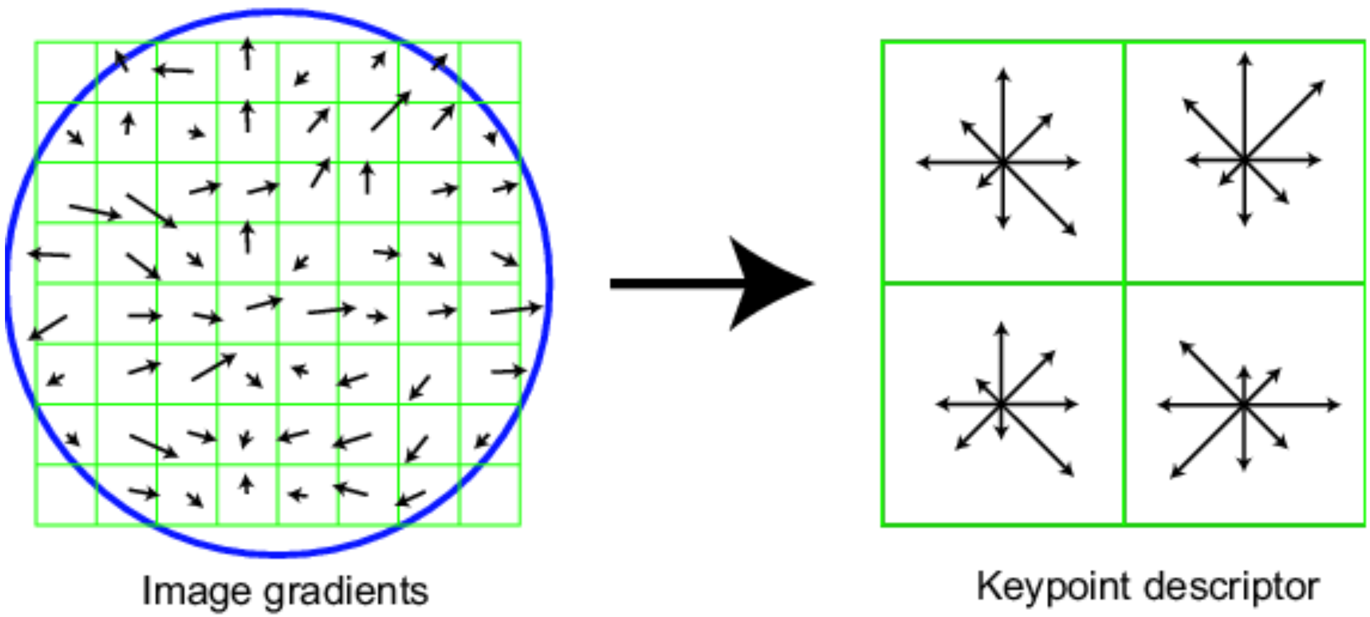


Adapted from slide by David Lowe

SIFT descriptor

Full version of the descriptor:

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor

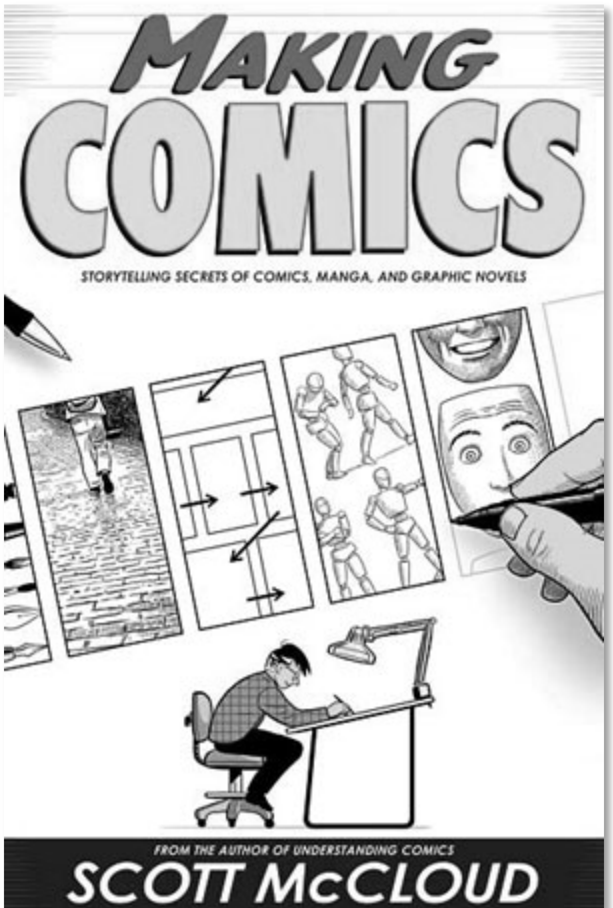


Adapted from slide by David Lowe

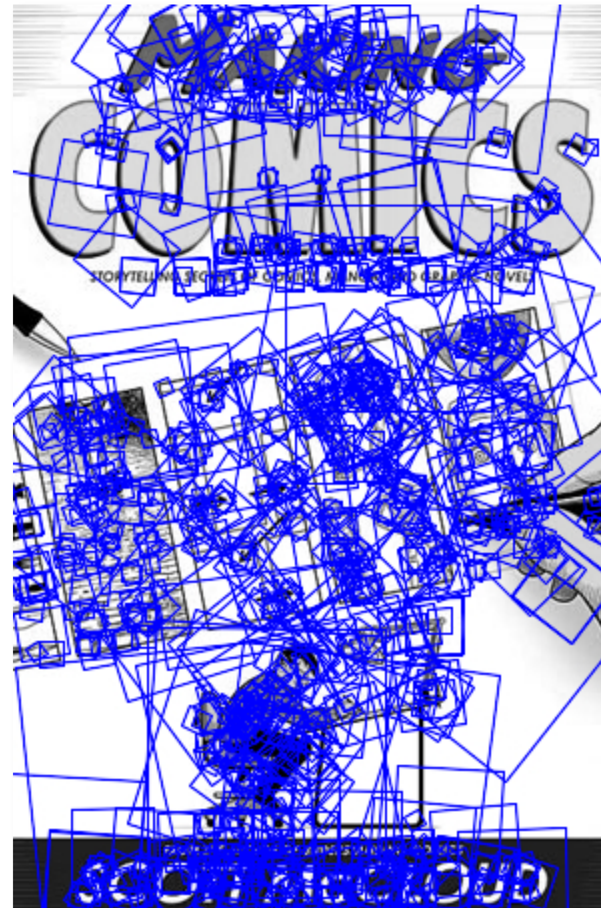
Scale Invariant Feature Transform

- The SIFT paper actually gives a full feature detection and description pipeline.
- Details in the paper: http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf
- Quickly became a workhorse in computer vision, remains a benchmark for faster / fancier algorithms.
- Different applications have different tradeoffs:
  - Gigapixel panorama stitching
  - Panorama stitching on phones
  - SLAM

SIFT Example



sift



868 SIFT features

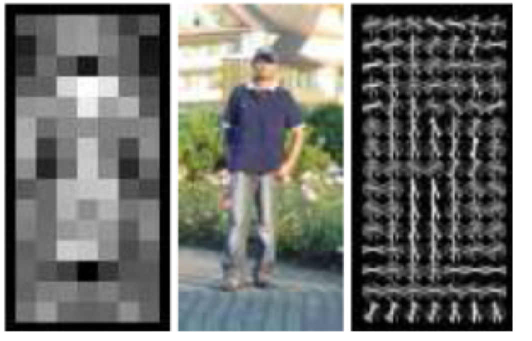
Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

Other descriptors

- HOG: Histogram of Gradients (HOG)
  - Dalal/Triggs
  - Sliding window, pedestrian detection

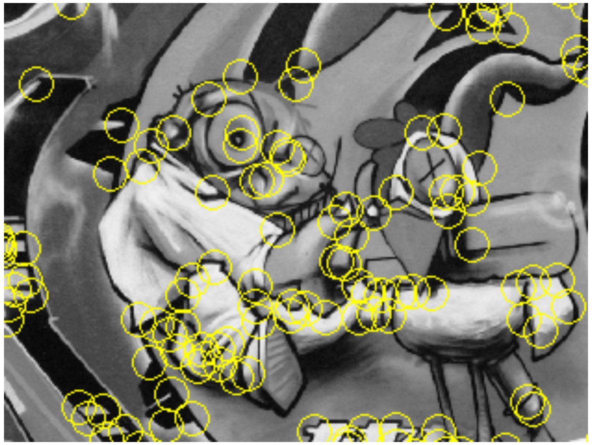


- FREAK: Fast Retina Keypoint
  - Perceptually motivated
  - Used in Visual SLAM
- LIFT: Learned Invariant Feature Transform
  - Learned via deep learning

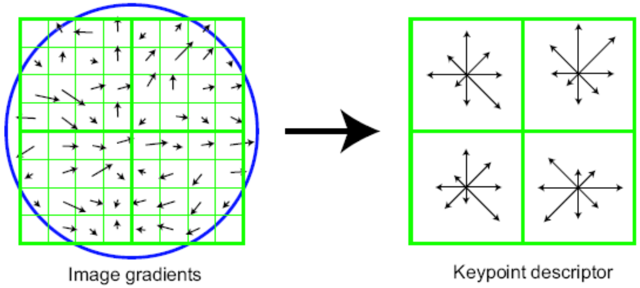
    https://arxiv.org/abs/1603.09114

Summary: Detection and Description

- Keypoint detection: repeatable and distinctive
  - Corners are most common
  - Can also use edges, blobs, …

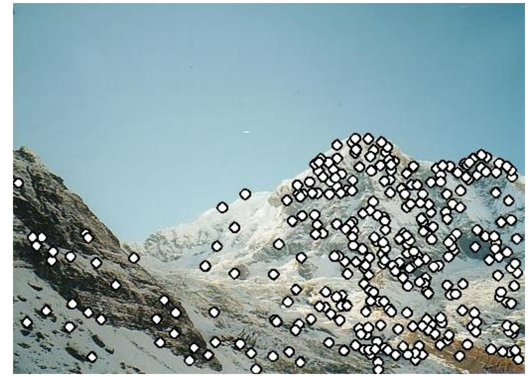


- Descriptors: robust and selective
  - spatial histograms of orientation
  - SIFT and variants are typically good for stitching and recognition
  - But, need not stick to one



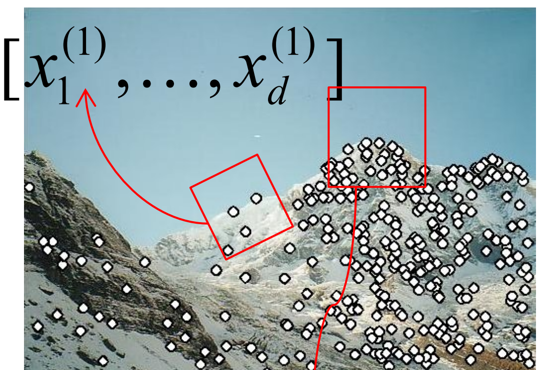
Local features: main components

1) Detection: Identify the interest points



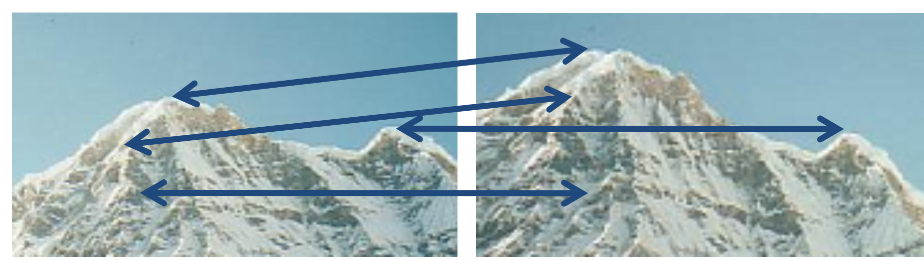
2) Description: Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



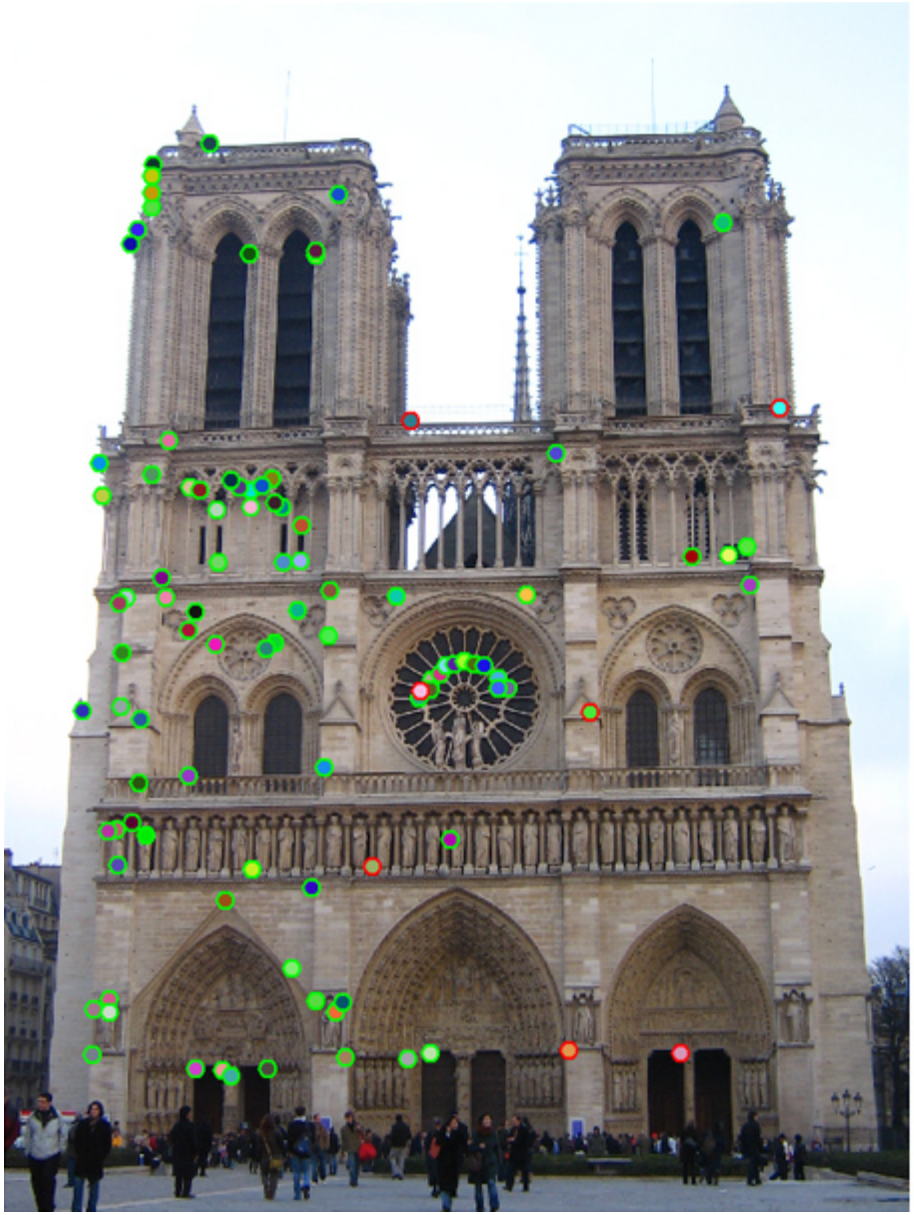
$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

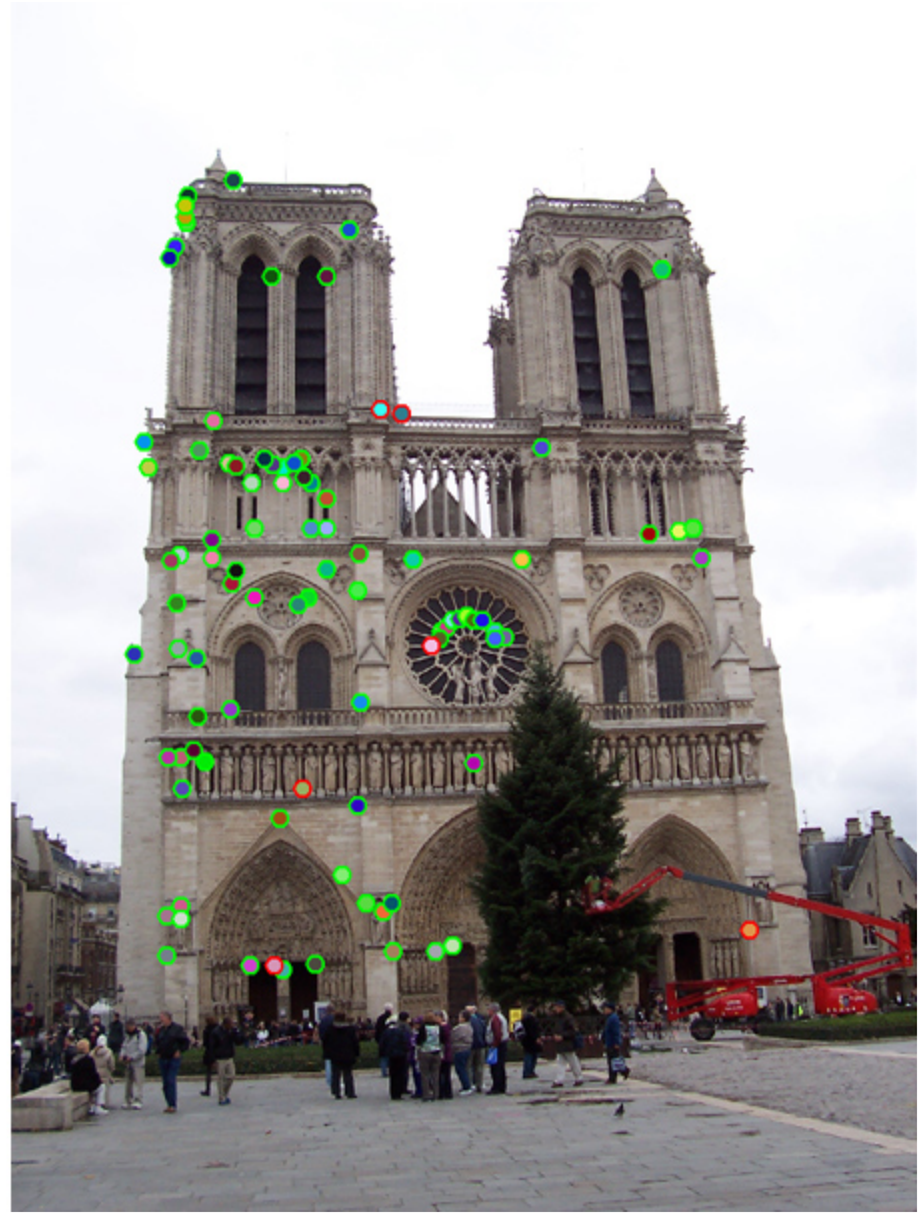
3) Matching: Determine correspondence between descriptors in two views



Kristen Grauman

Which features match?





Feature matching

Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance