ELEC 3210
Introduction to Mobile Robotics
Lecture 14

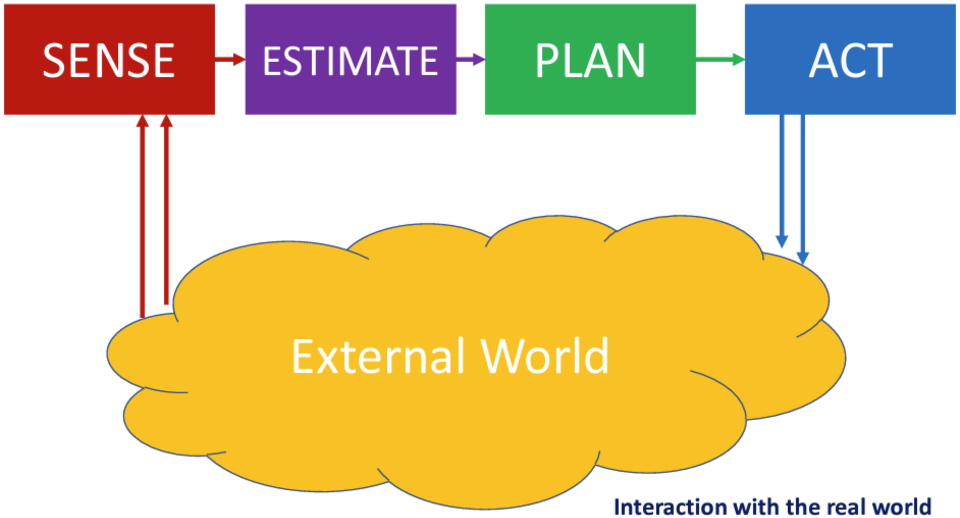
(Machine Learning and Infomation Processing for Robotics)

Huan YIN

Research Assistant Professor, Dept. of ECE

eehyin@ust.hk

Recap L7 to L12

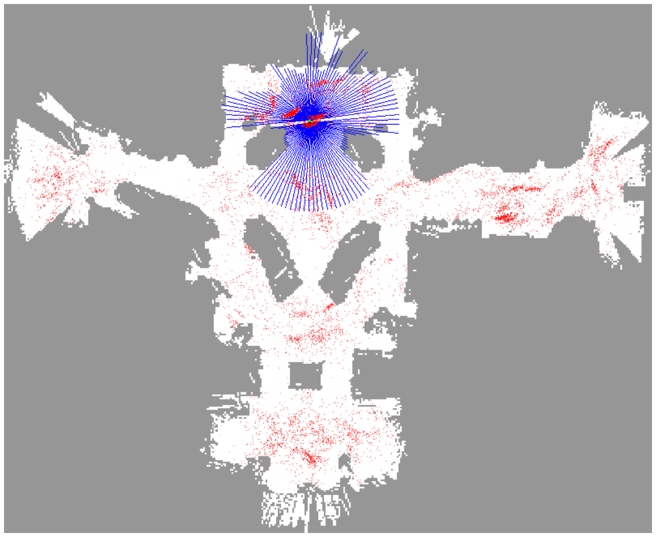


Interaction with the real world introduces uncertainty!

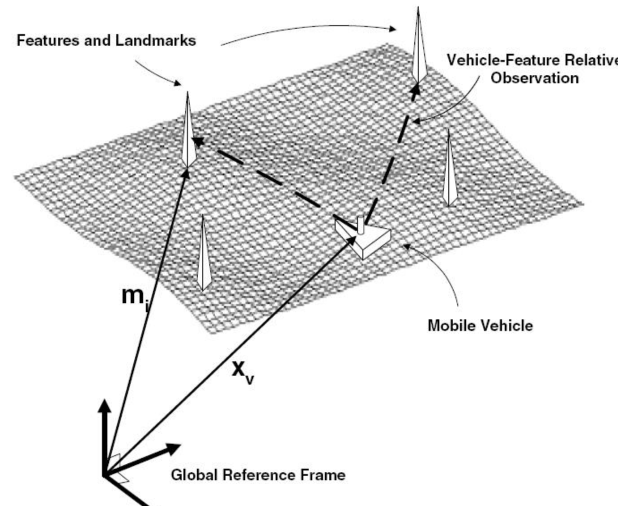
- Pose is the ``flow'' in the navigation paradigm
- With sensings, we estimate the poses using
- Bayes Filter
  - Particle Filter, Extended Kalman Filter, EKF SLAM
- Graph-based
  - Pose graph optimization, or with landmarks

Case Studies in L7 - L12

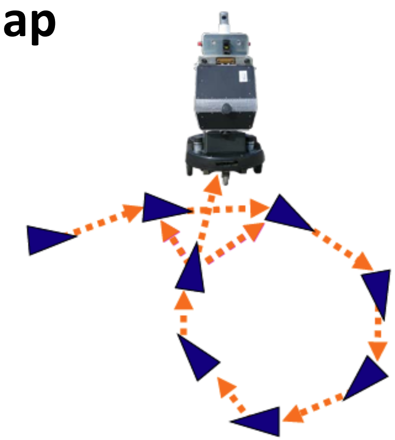
- All with laser sensings



Particle Filter on a Given Map



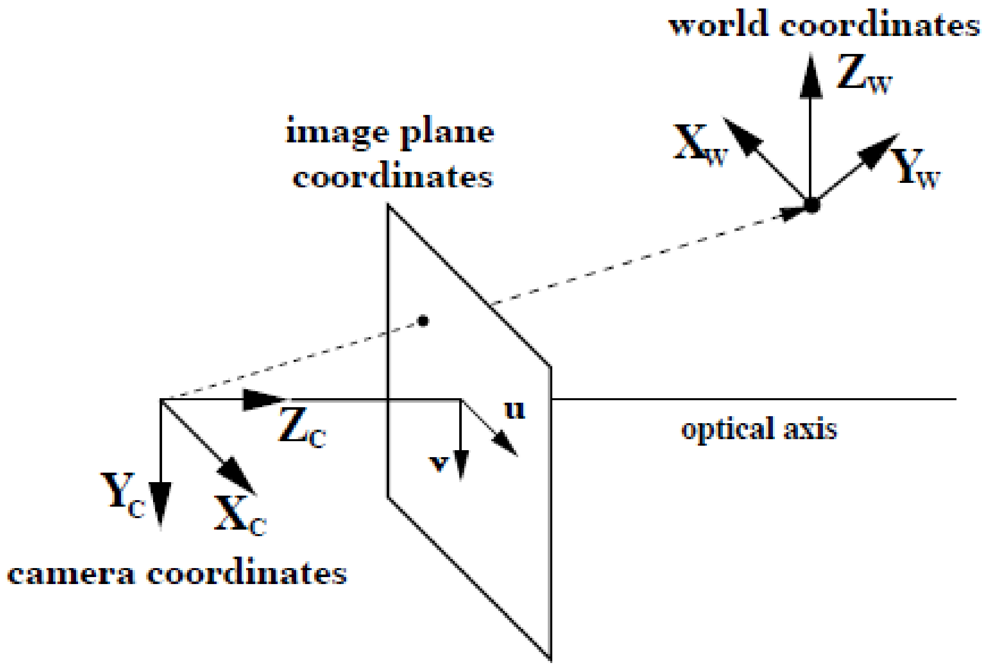
EKF SLAM



Pose Graph SLAM

Visual Perception This Week

- We **touch upon** the pose estimation using visual perception
- Cameras as sensors, Lecture 4 Sensors
- From 2D image plane to 3D pose estimation

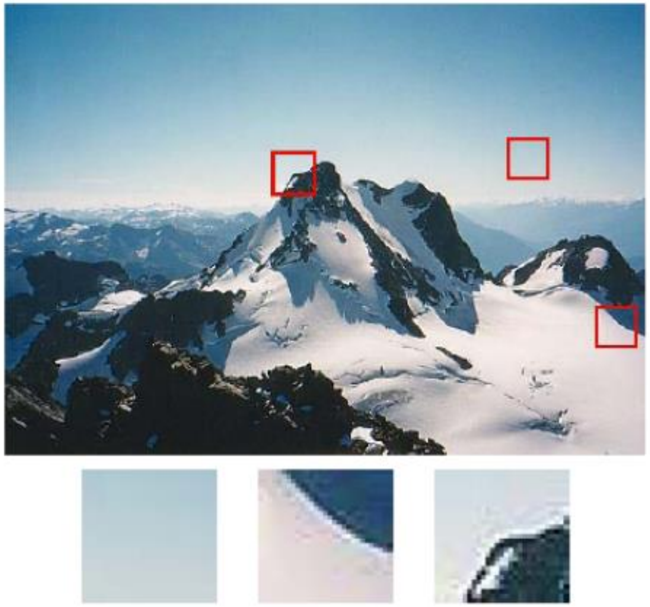


Feature Detection

- We cannot process the entire image directly
  - "landmark detection" in LiDAR scans

- Requirements:
  - Repeatability
  - Saliency
  - Locality
  - Compactness and efficiency

- Popular features
  - Corners (FAST, Harris☺, …)
  - Blob (SIFT☺, SURF, …)
  - Line (Canny, …)

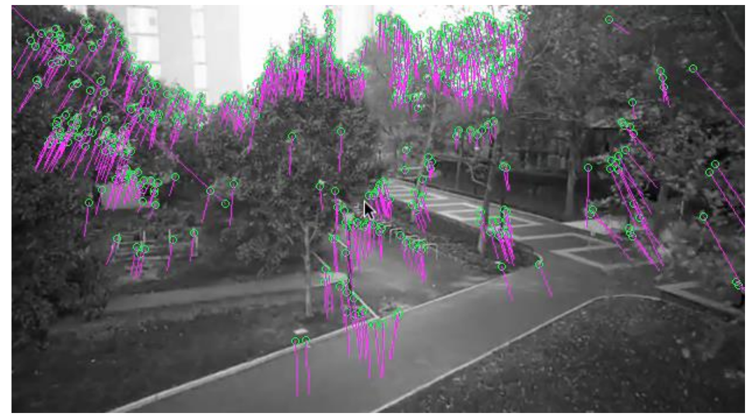


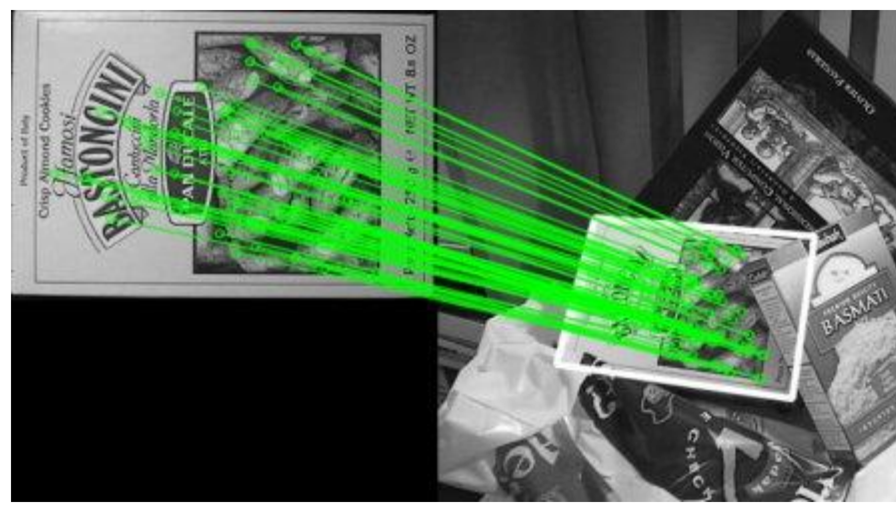
Courtesy: Shaojie Shen

Feature Matching

- Match features in different images
  - Across multiple cameras
  - Across time

- Common methods:
  - Descriptor matching ☺
  - Optical flow



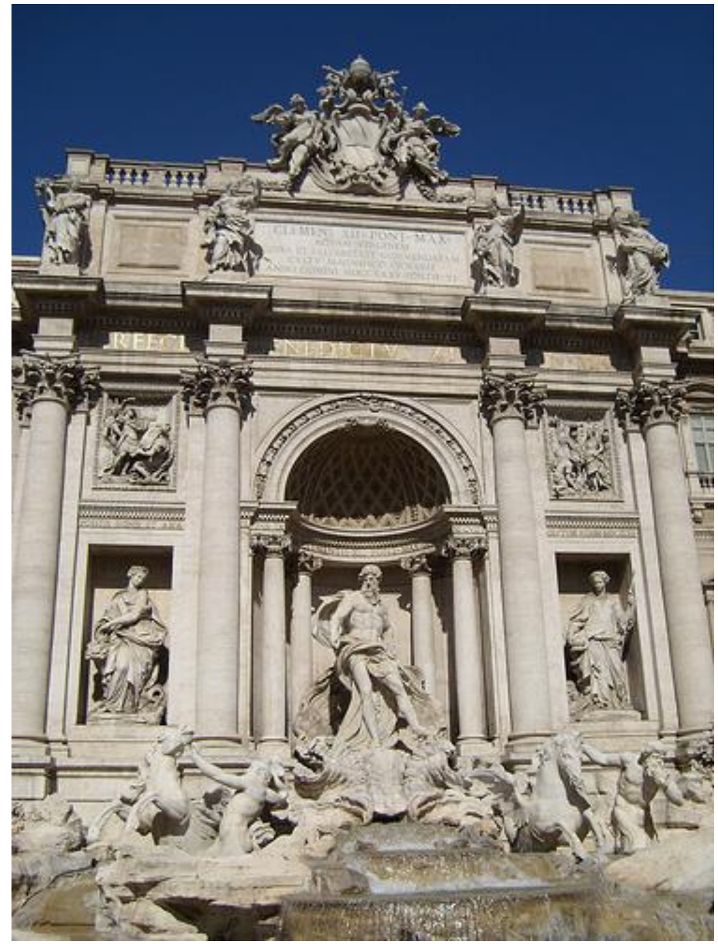


Courtesy: Shaojie Shen

Feature (Keypoint) Detection

Image Matching

by Divan Sian



by swashford

Courtesy: Steve Seitz and Richard Szeliski

Harder Case

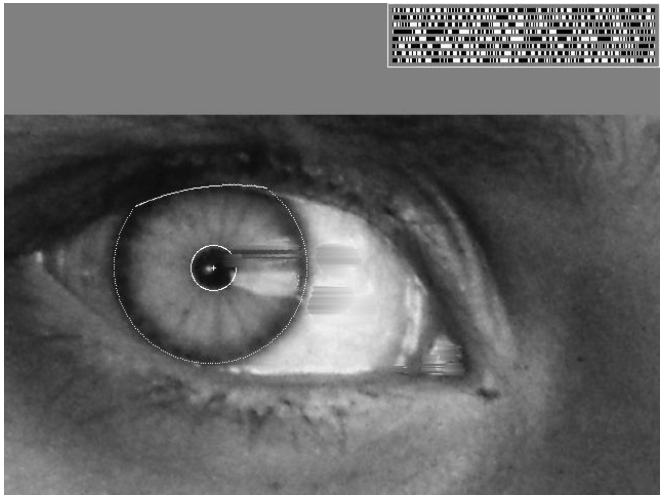
by Divan Sian

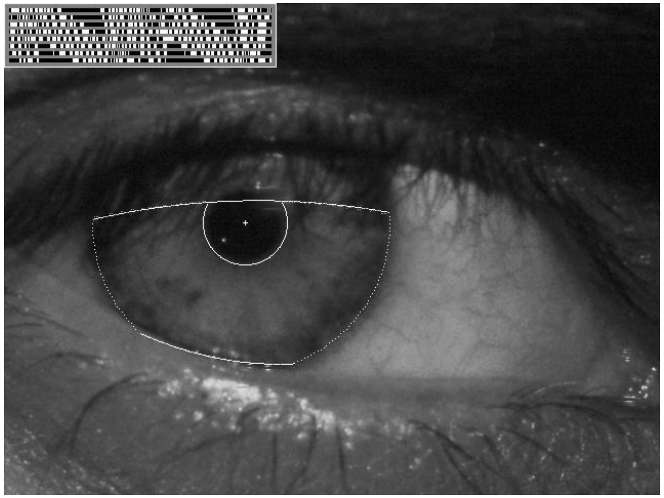
by scgbt

Courtesy: Steve Seitz and Richard Szeliski

Even Harder Case

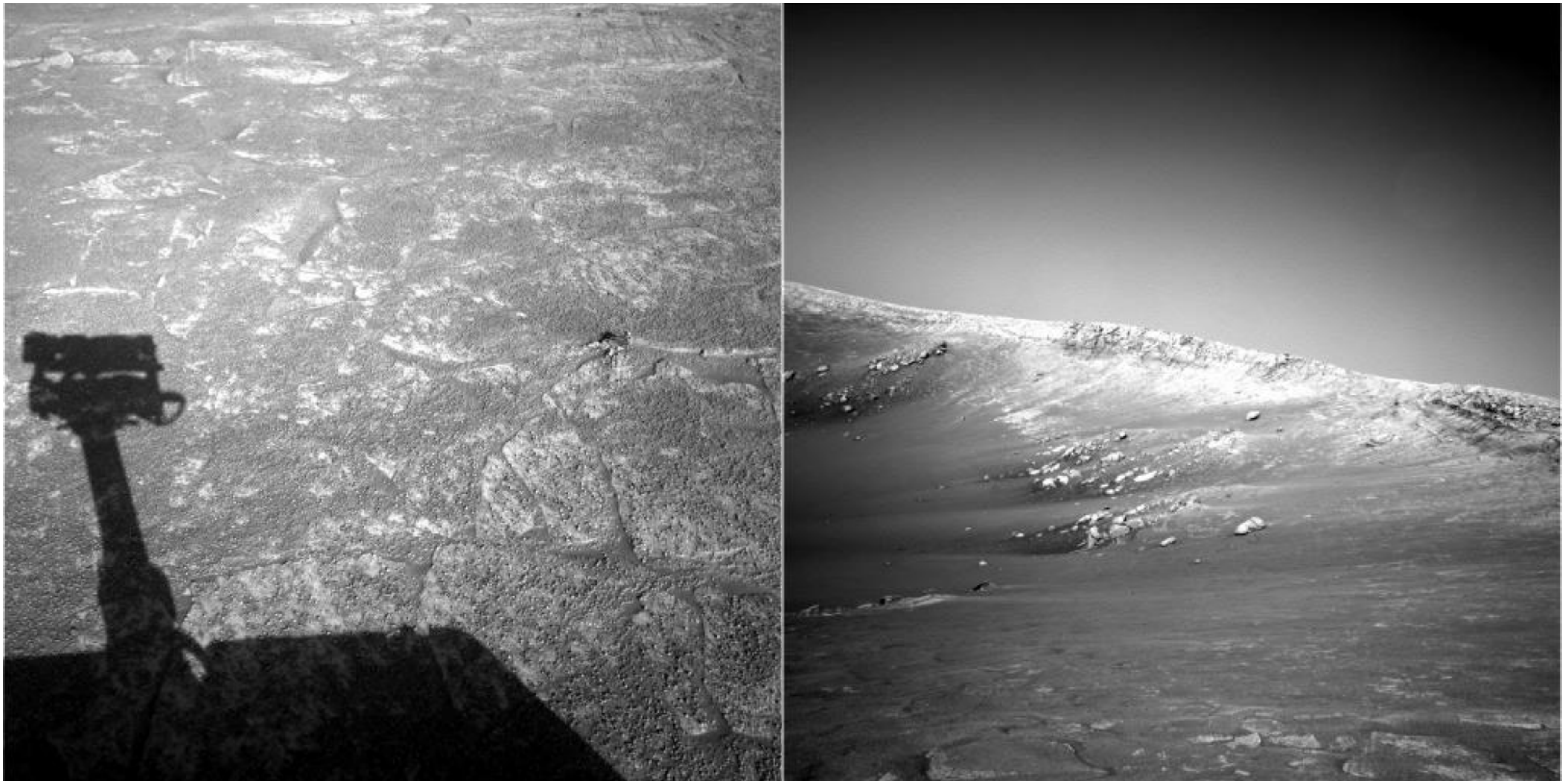
“How the Afghan Girl was Identified by Her Iris Patterns” Read the story





Courtesy: Steve Seitz and Richard Szeliski

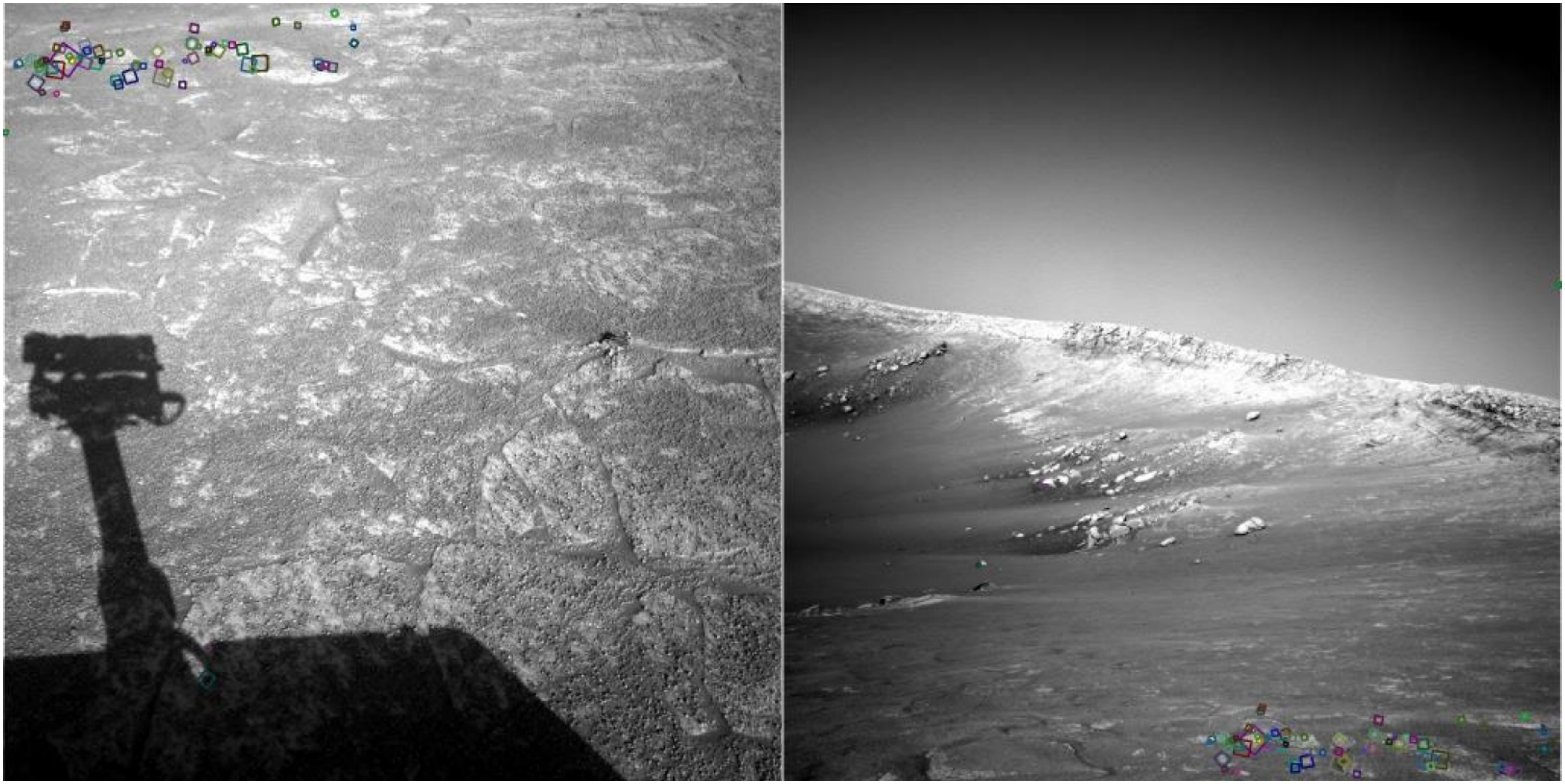
Harder Still?



NASA Mars Rover Images

Courtesy: Steve Seitz and Richard Szeliski

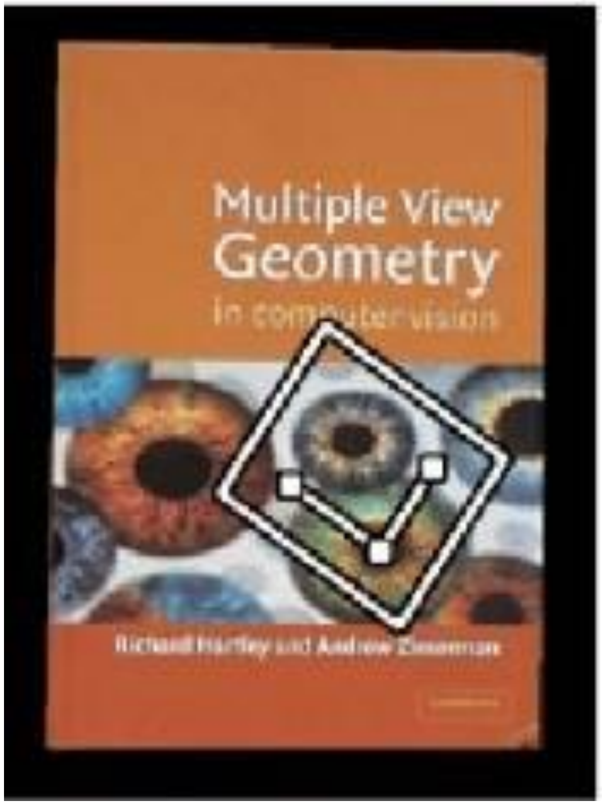
Look for Tiny Colored Squares…



NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely

Courtesy: Steve Seitz and Richard Szeliski

Image Matching



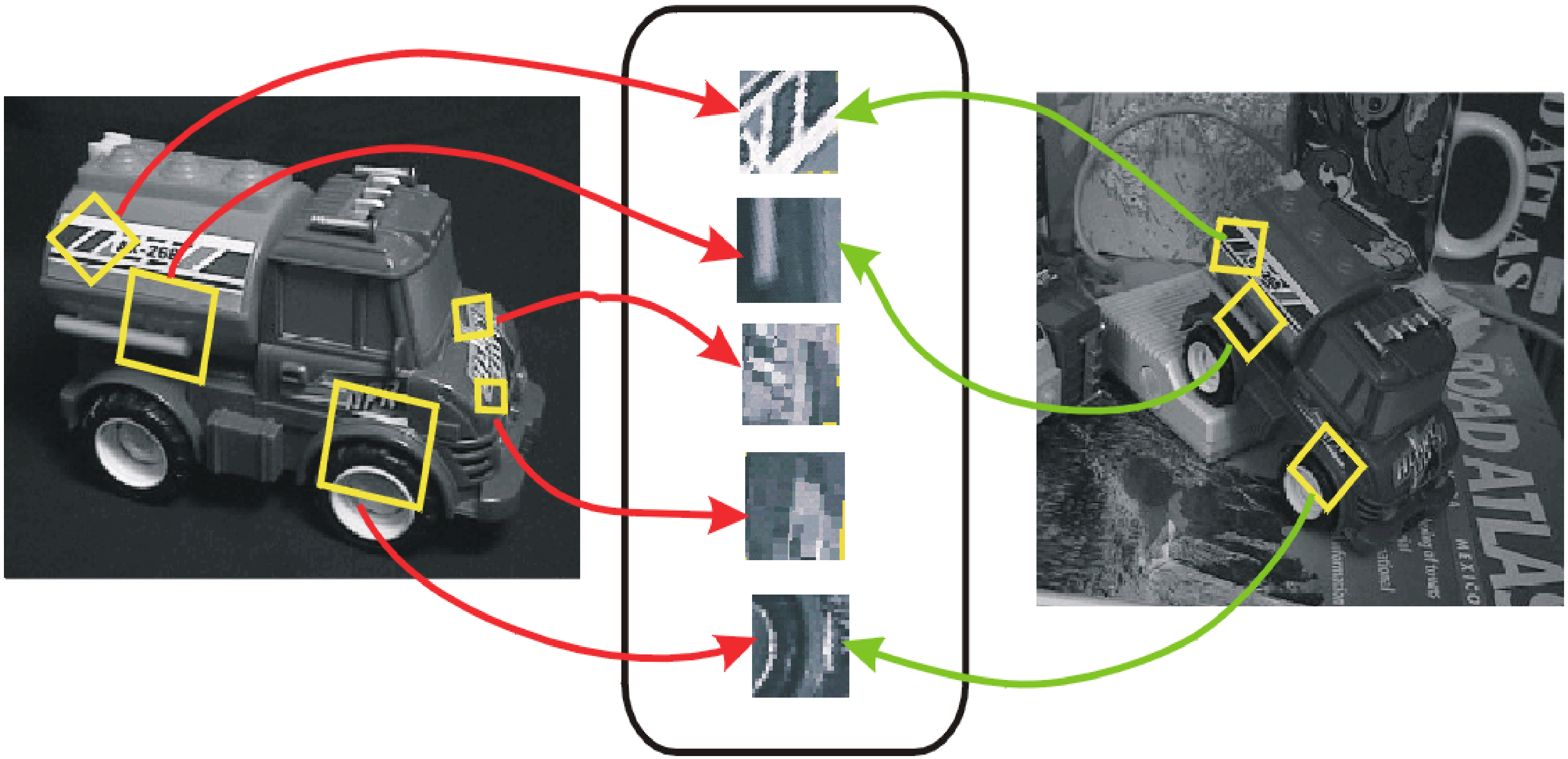
Courtesy: Steve Seitz and Richard Szeliski

Image Matching

Courtesy: Steve Seitz and Richard Szeliski

Invariant Local Features

- Find features that are invariant to transformations
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure, …



Courtesy: Steve Seitz and Richard Szeliski

Advantages of Local Features

- Locality
  - Features are local, so robust to occlusion and clutter
- Distinctiveness
  - Can differentiate a large database of objects
- Quantity
  - Hundreds or thousands in a single image
- Efficiency
  - Real-time performance achievable
- Generality
  - Exploit different types of features in different situations

Courtesy: Steve Seitz and Richard Szeliski

More Motivation…

- Feature points are used for:
  - Image alignment (e.g., mosaics)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … other

Courtesy: Steve Seitz and Richard Szeliski

Want Uniqueness

- Look for image regions that are unusual
    - Lead to unambiguous matches in other images

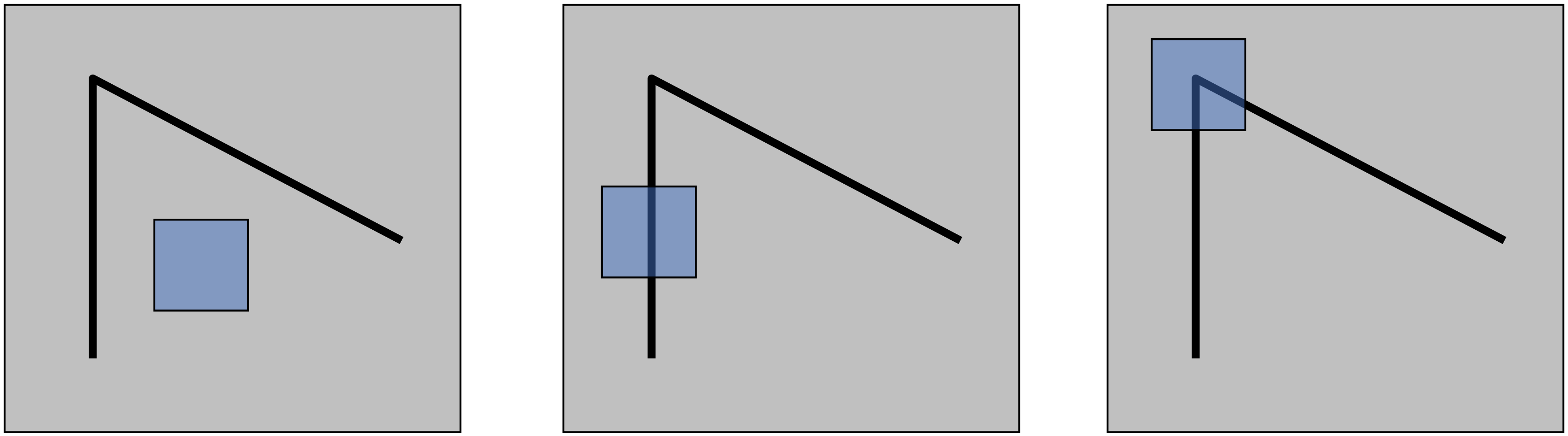
- How to define “unusual”?

Courtesy: Steve Seitz and Richard Szeliski

Local Measures of Uniqueness

Suppose we only consider a small window of pixels

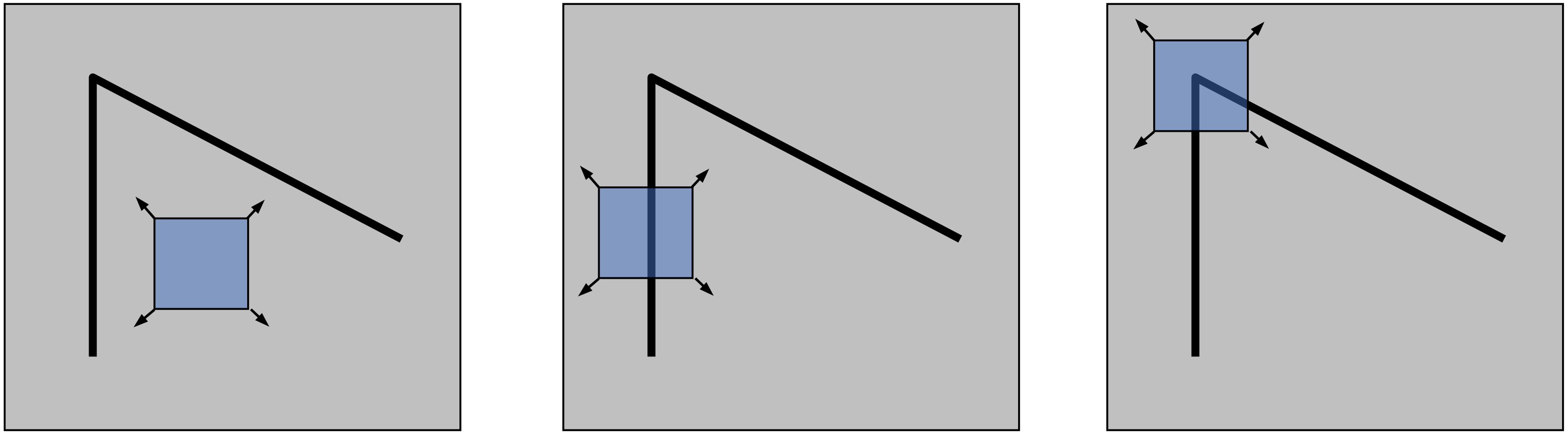
- What defines whether a feature is a good or bad candidate?



Courtesy: Steve Seitz and Richard Szeliski

Feature Detection

- Local measure of feature uniqueness
  - How does the window change when you shift it?
  - Shifting the window in any direction causes a big change



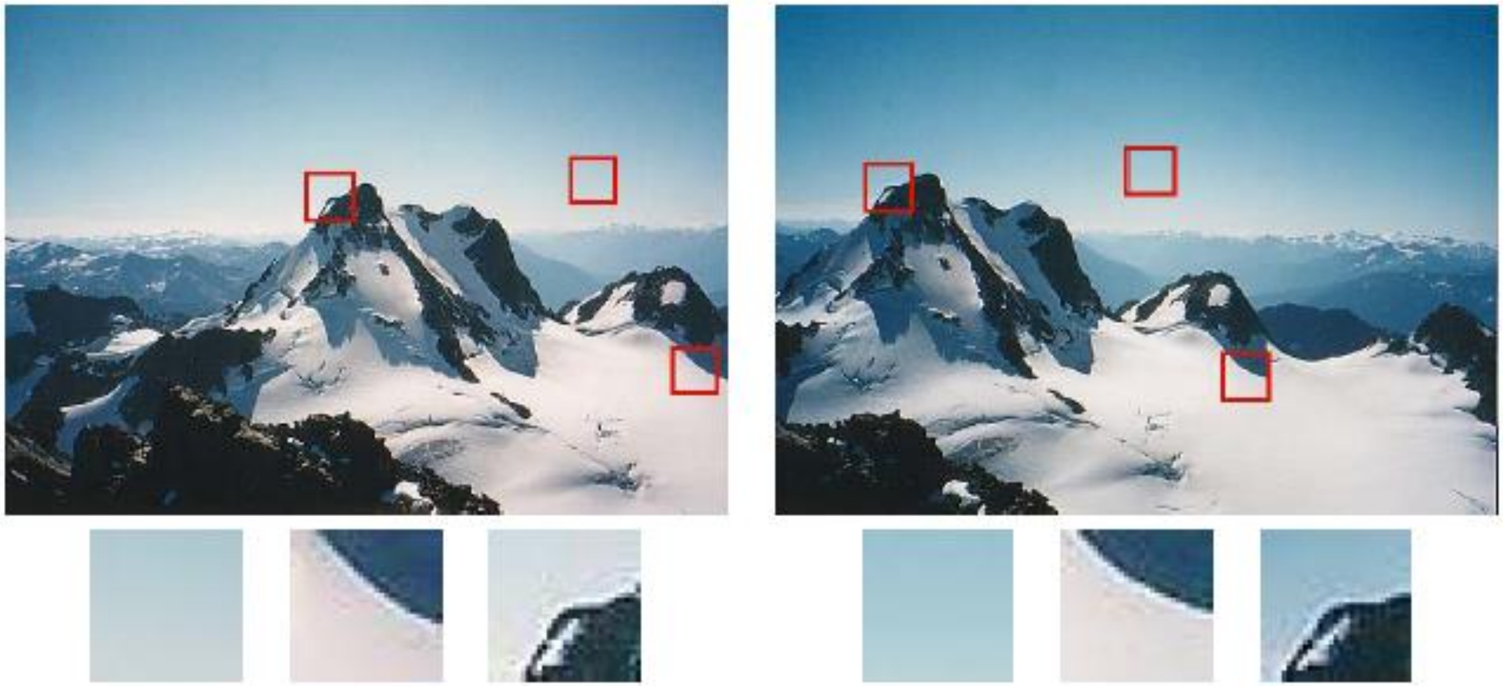
"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Courtesy: Steve Seitz and Richard Szeliski

Feature Detection

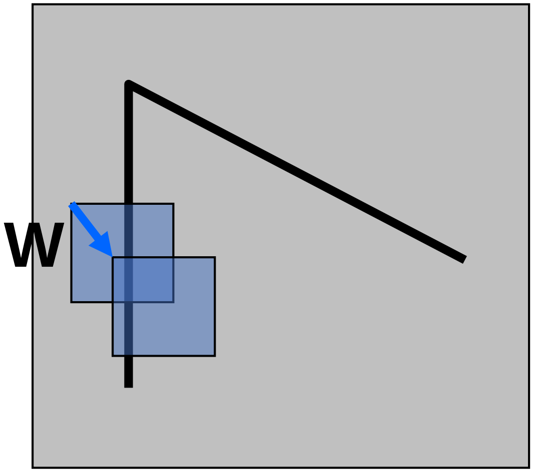


Courtesy: Steve Seitz and Richard Szeliski

Feature Detection: the Math

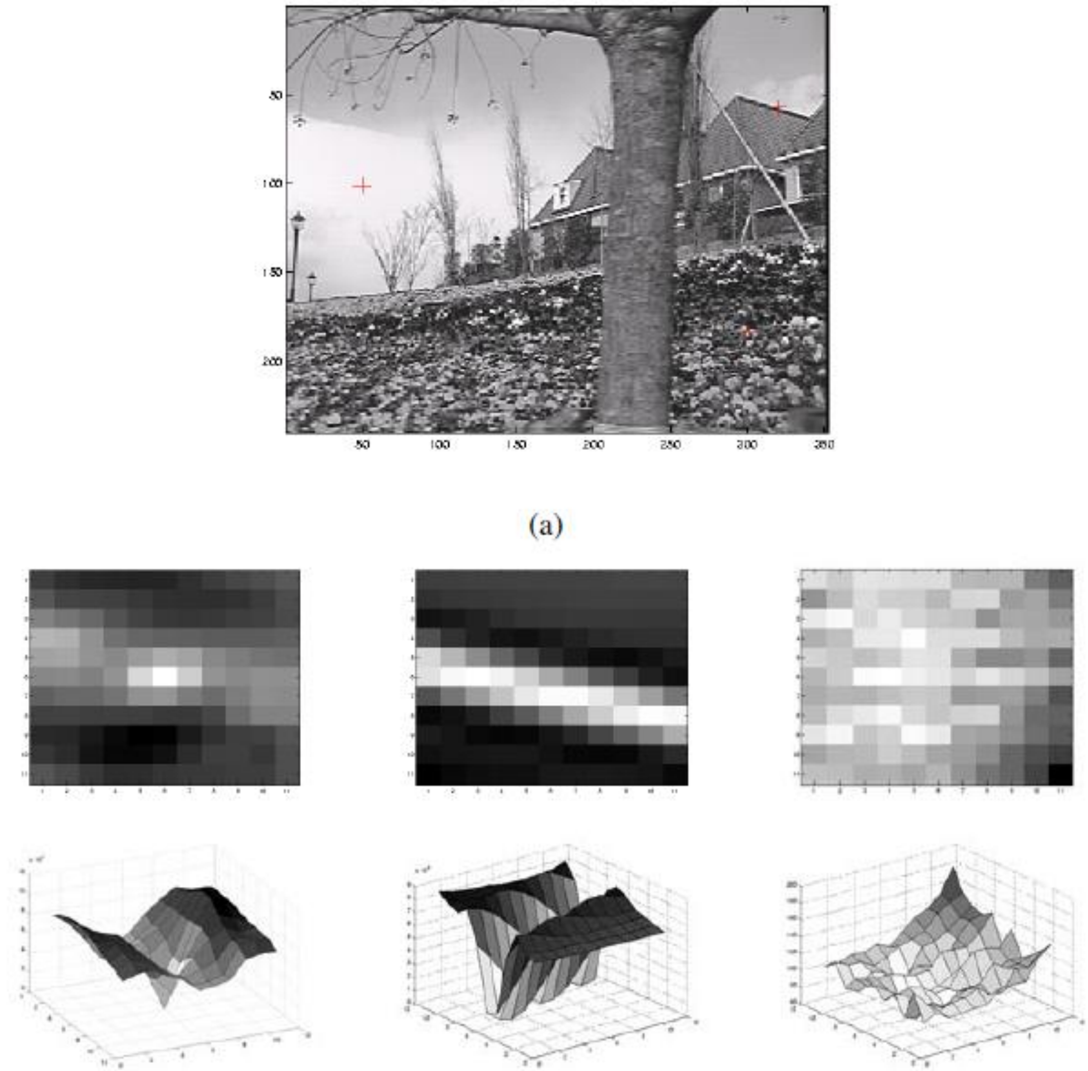
- Consider shifting the window **W** by (u,v)
  - How do the pixels in **W** change?
  - Compare each pixel before and after by summing up the squared differences (SSD)
  - This defines an SSD "error" of *E(u,v)*:

$$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v) - I(x,y)]^2$$



Courtesy: Steve Seitz and Richard Szeliski

Feature Detection



(a)

Courtesy: Steve Seitz and Richard Szeliski

Small Motion Assumption

- Taylor Series expansion of I:

$$I(x+u, y+v) = I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

- If the motion (u,v) is small, then first order approximation is good

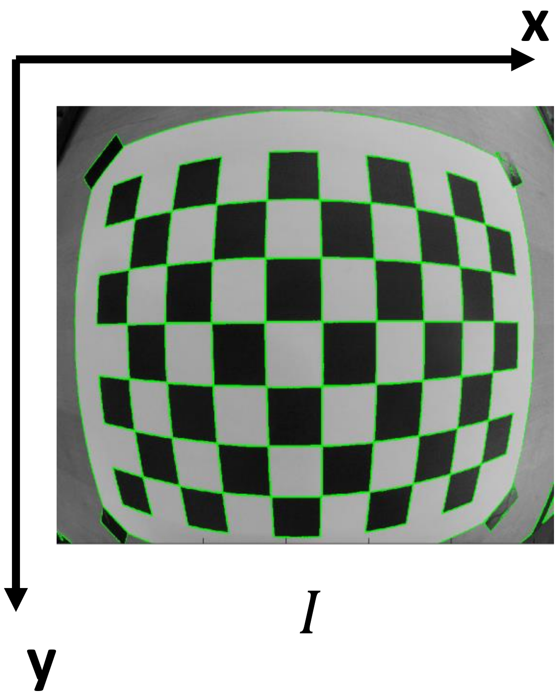
$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x,y) + \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

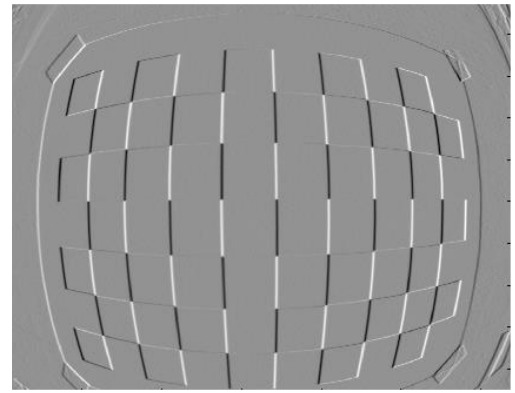
$$\text{shorthand: } I_x = \frac{\partial I}{\partial x}$$

Courtesy: Steve Seitz and Richard Szeliski

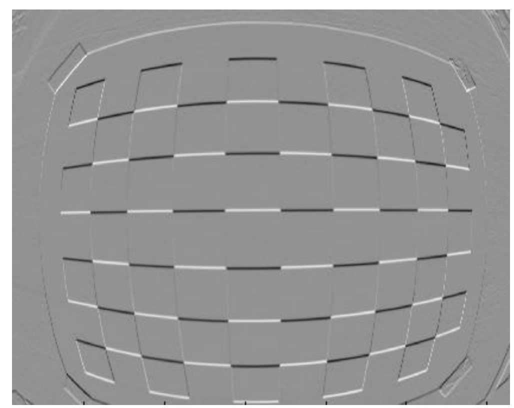
Image Gradients



I



$$I_x = I(x+1, y) - I(x-1, y)$$



$$I_y = I(x, y+1) - I(x, y-1)$$

Courtesy: Shaojie Shen

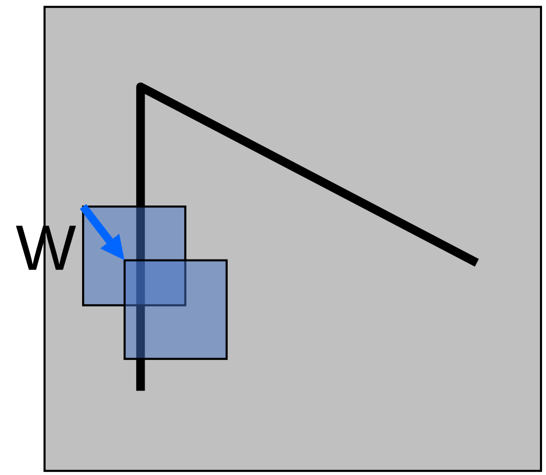
Feature Detection: the Math

- Plugging gradients into the formula on the previous slide…
- Consider shifting the window **W** by (u,v)
  - How do the pixels in **W** change?
  - Compare each pixel before and after by summing up the squared differences
  - This defines an "error" of *E(u,v)*:

$$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v) - I(x,y)]^2$$

$$\approx \sum_{(x,y)\in W} \left[I(x,y) + [I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x,y)\right]^2$$

$$\approx \sum_{(x,y)\in W} \left[[I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix}\right]^2$$

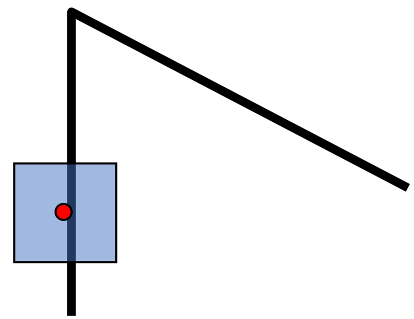


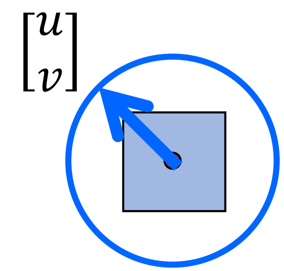
Courtesy: Steve Seitz and Richard Szeliski

Feature Detection: the Math

- This can be rewritten:

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





- For the example above
  - You can move the center of the window to anywhere on the blue unit circle
  - Which directions will result in the largest and smallest E values?
  - We can find these directions by looking at the **eigenvectors** of *H*

Quick Eigenvalue/vector Review

- The **eigenvectors** of a matrix **A** are the vectors **x** that satisfy:

$$Ax = \lambda x$$

- The scalar λ is the **eigenvalue** corresponding to **x**
  - The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

  - In our case, ***A*** = ***H*** is a 2x2 matrix, so we have

$$det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$

  - The solution:

$$\lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$$

- Once you know λ, you find **x** by solving

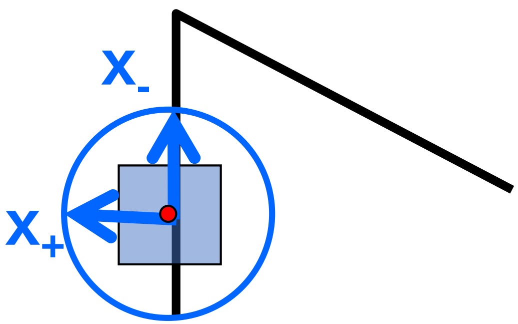
$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

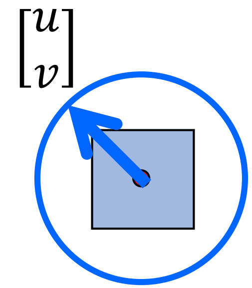
Courtesy: Steve Seitz and Richard Szeliski

Feature Detection: the Math

- This can be rewritten:

$$E(u,v) = \sum_{(x,y)\in W} [u \quad v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} \begin{bmatrix} u \\ v \end{bmatrix}$$





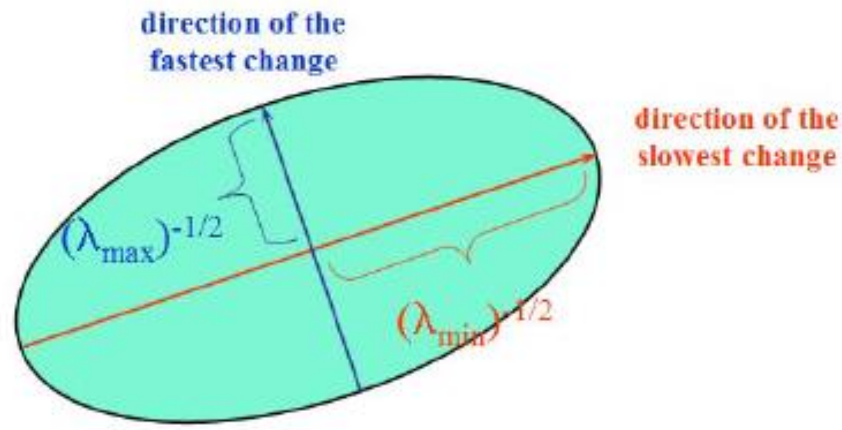
- Eigenvalues and eigenvectors of H (structure matrix)
  - Define shifts with the smallest and largest change (E value)
  - x_+ = direction of **largest** increase in E.
  - λ_+ = amount of increase in direction x_+
  - x_- = direction of **smallest** increase in E.
  - λ- = amount of increase in direction x_+

$$Hx_+ = \lambda_+ x_+$$
$$Hx_- = \lambda_- x_-$$

Courtesy: Steve Seitz and Richard Szeliski

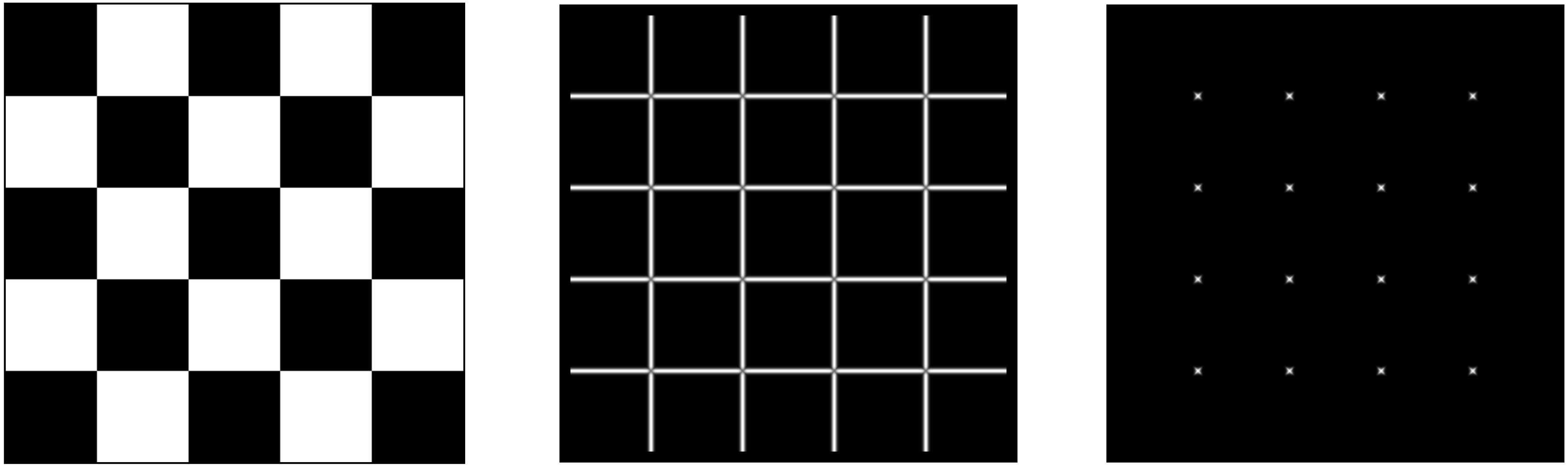
Feature Detection: the Math

- How are λ_+, $\mathbf{x}_+$, λ_-, and $\mathbf{x}_-$ relevant for feature detection?
  - What's our feature scoring function?



Feature Detection: the Math

- How are λ_+, $\mathbf{x}_+$, λ_-, and $\mathbf{x}_-$ relevant for feature detection?
  - What's our feature scoring function?
- Want *E(u,v)* to be ***large*** for small shifts in ***all*** directions
  - the minimum of *E(u,v)* should be large, over all unit vectors [u v]
  - this minimum is given by the smaller eigenvalue (λ_-) of ***H***

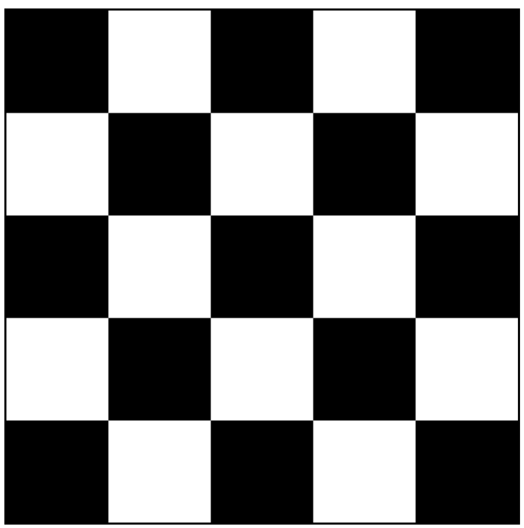


I λ_+ λ_-

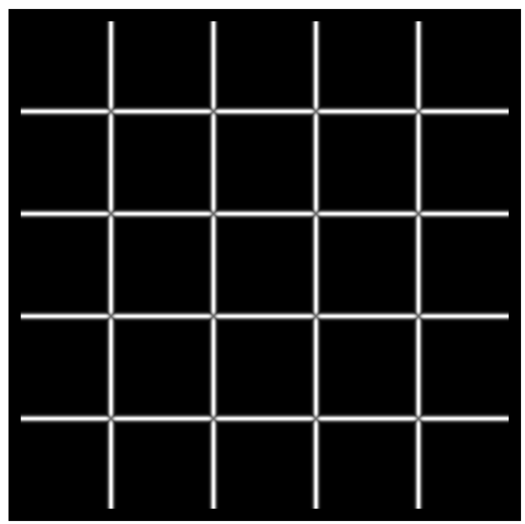
Courtesy: Steve Seitz and Richard Szeliski

Feature Detection Summary

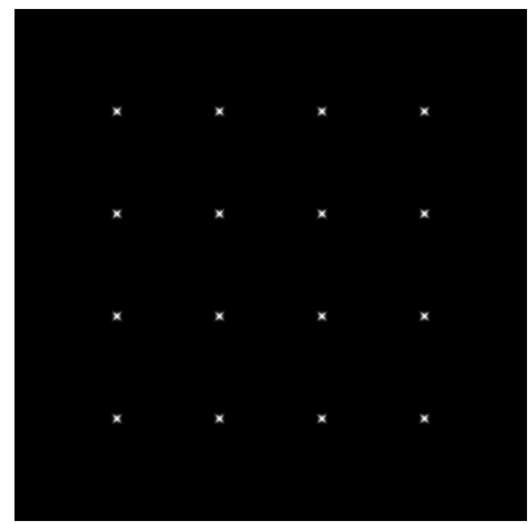
- Here's what you do
  - Compute the gradient at each point in the image
  - Create the ***H*** matrix from the entries in the gradient
  - Compute the eigenvalues
  - Find points with large response (λ_- > threshold)
  - Choose those points where λ_- is a local maximum as features



I



λ_+

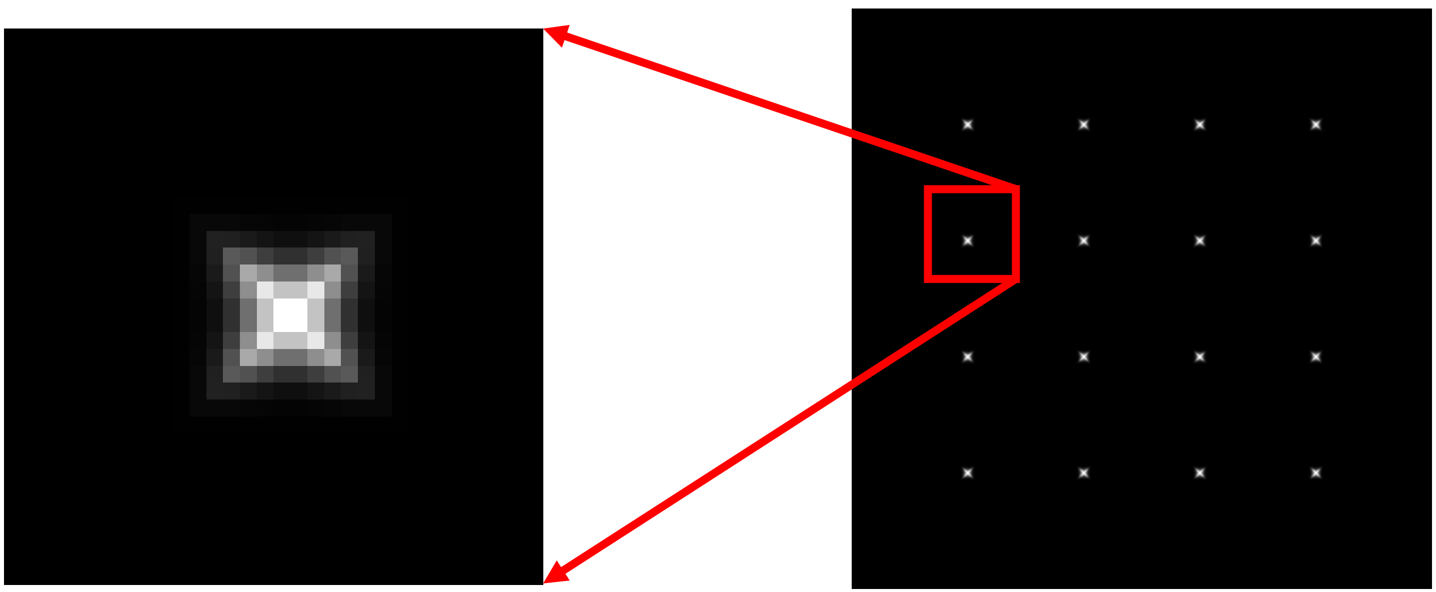


λ_-

Courtesy: Steve Seitz and Richard Szeliski

Feature Detection Summary

- Here's what you do
  - Compute the gradient at each point in the image
  - Create the ***H*** matrix from the entries in the gradient
  - Compute the eigenvalues.
  - Find points with large response (λ_- > threshold)
  - Choose those points where λ_- is a local maximum as features



λ_-

Courtesy: Steve Seitz and Richard Szeliski

Harris Corner

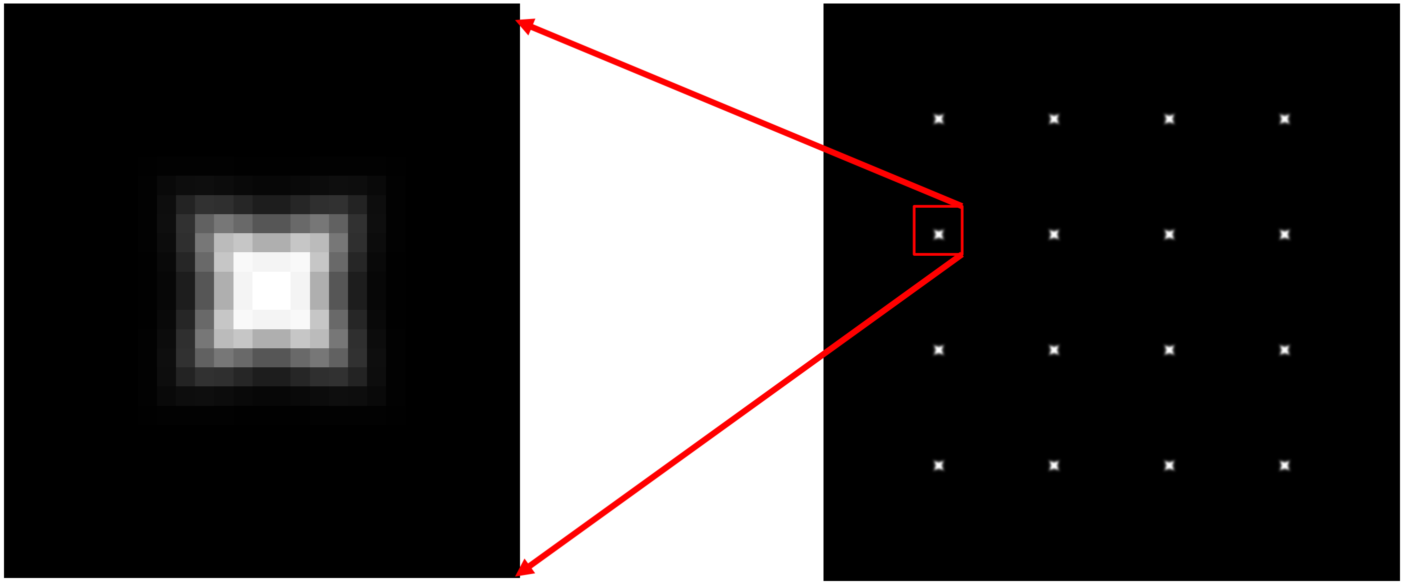
- λ_- is a variant of the "Harris operator" for feature detection

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$
$$= \frac{determinant(H)}{trace(H)}$$

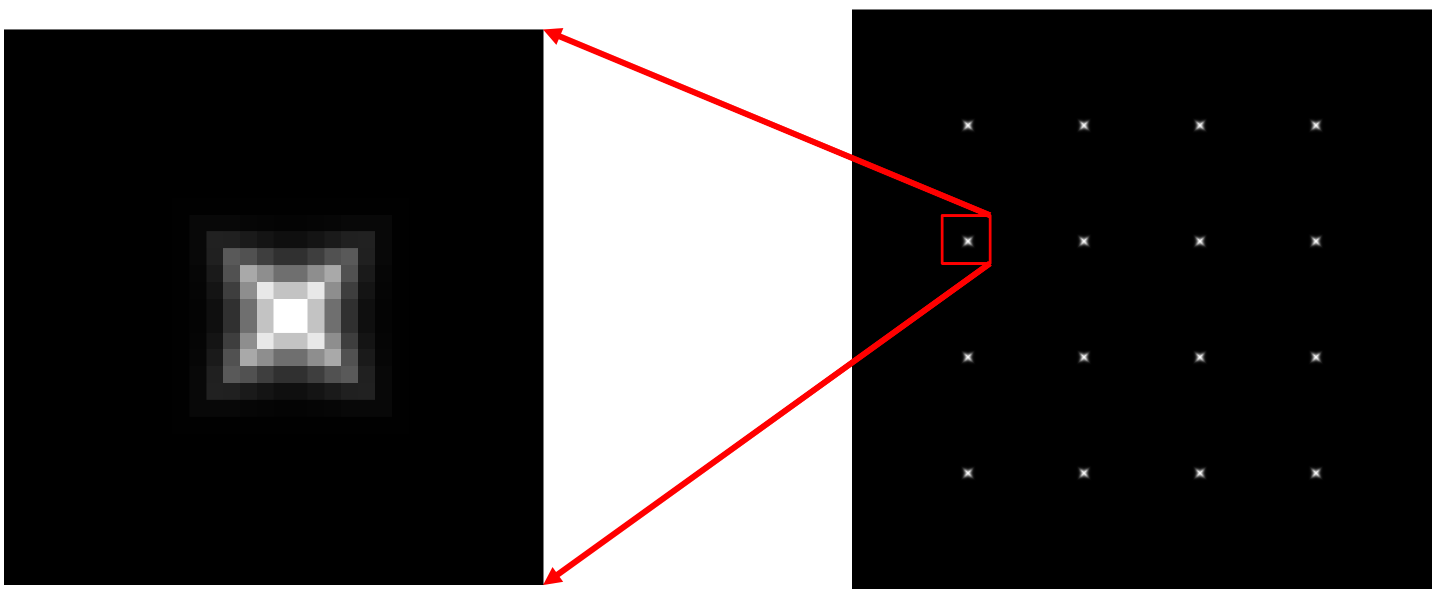
- The *trace* is the sum of the diagonals, i.e., $trace(H) = h_{11} + h_{22}$
- Very similar to λ_- but less expensive (no square root)
- Called the "Harris Corner Detector" or "Harris Operator"
- Lots of other detectors, this is one of the most popular

Courtesy: Steve Seitz and Richard Szeliski

The Harris Operator



Harris operator

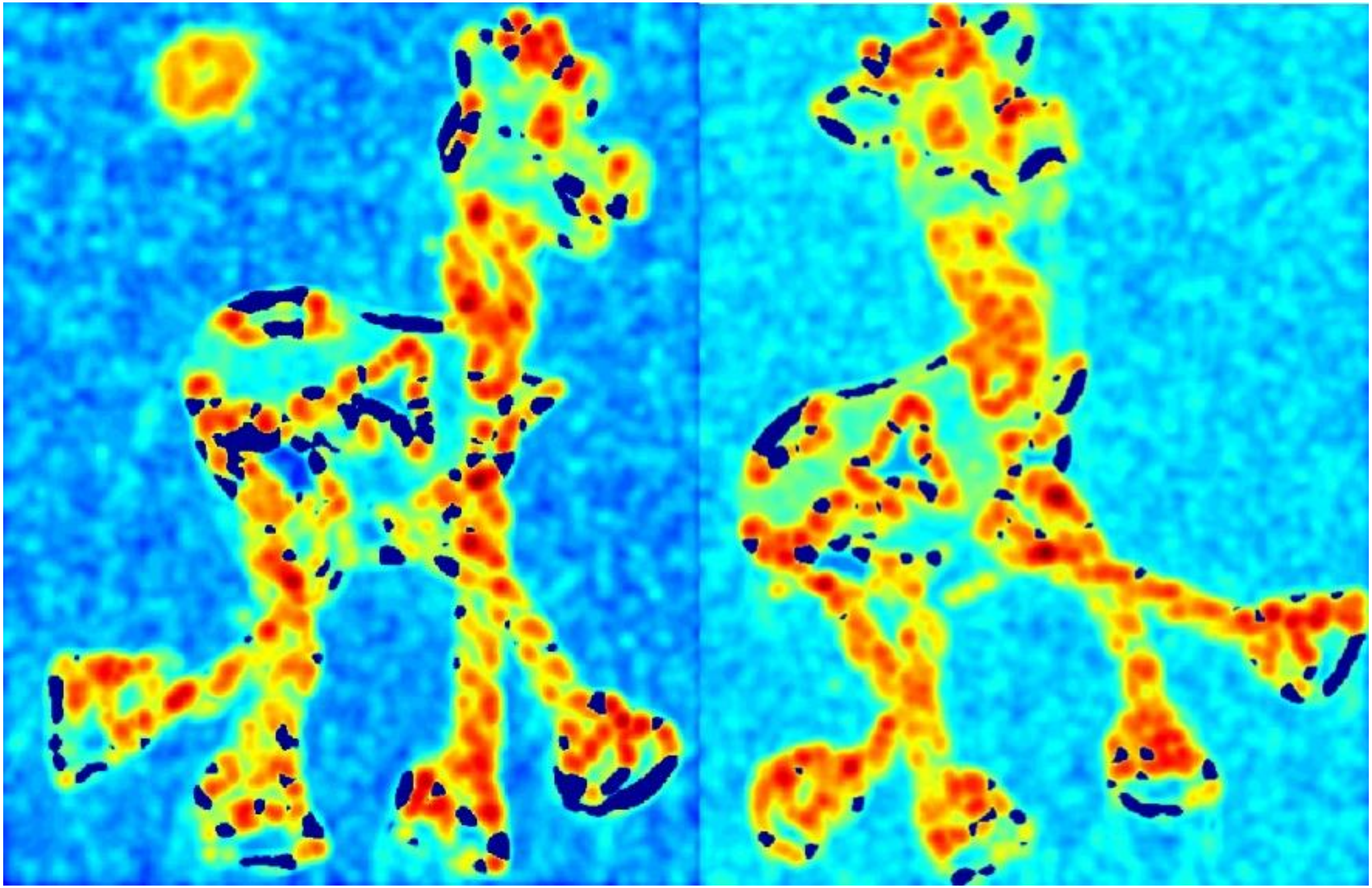


λ_

Harris Detector Example

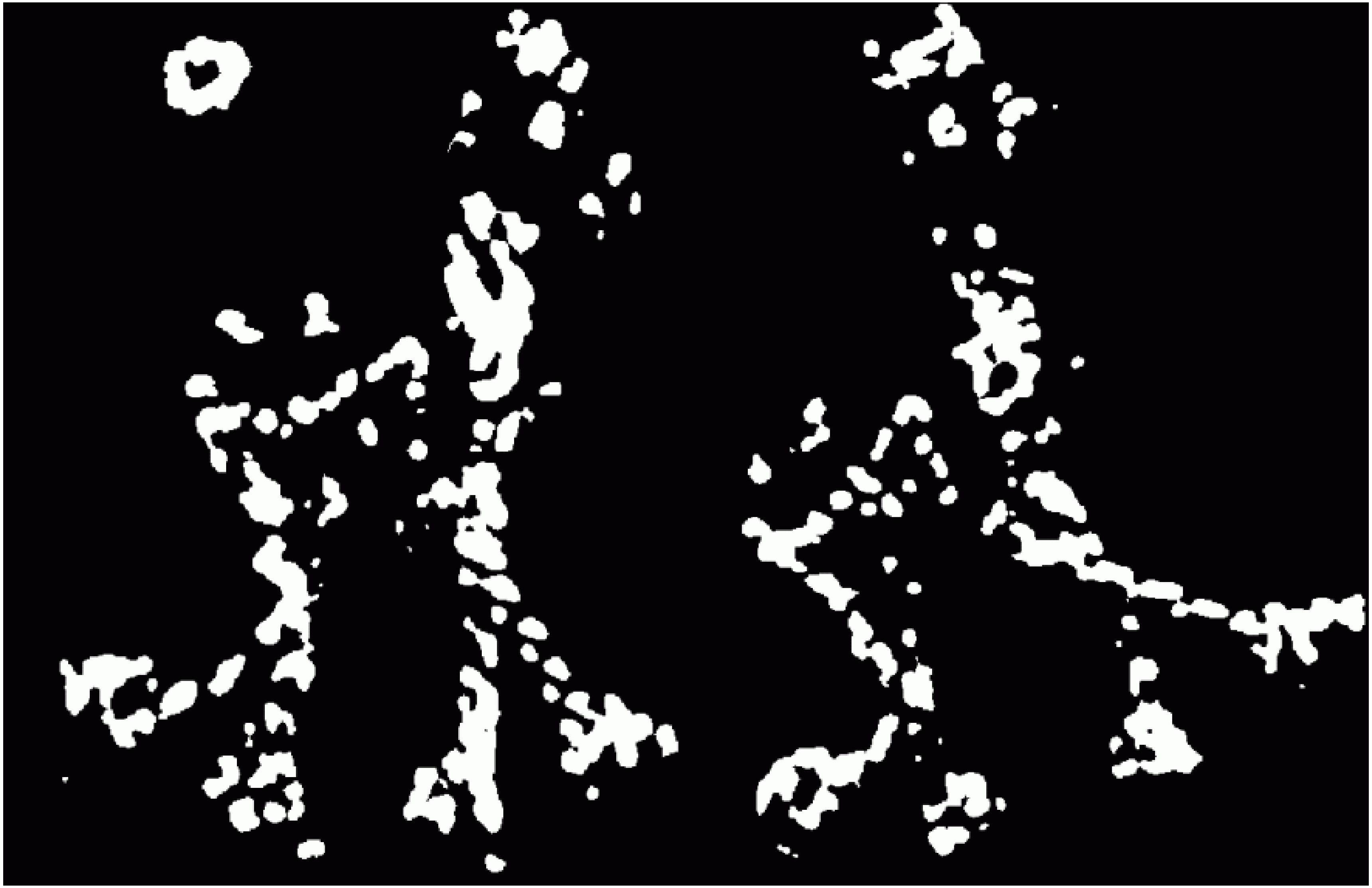
Courtesy: Steve Seitz and Richard Szeliski

f value (red high, blue low)



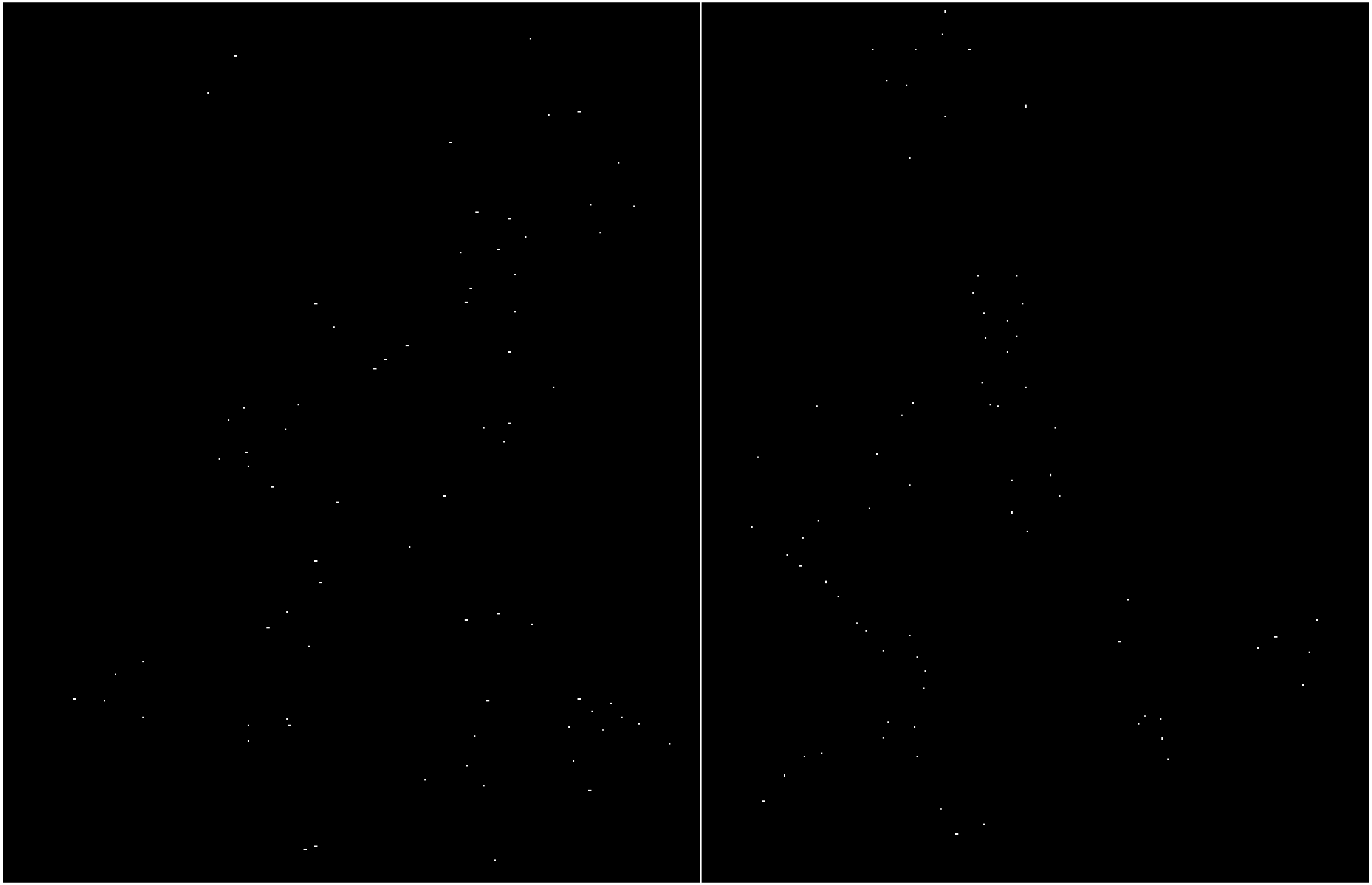
Courtesy: Steve Seitz and Richard Szeliski

Threshold (f > value)



Courtesy: Steve Seitz and Richard Szeliski

Find Local Maxima of f



Courtesy: Steve Seitz and Richard Szeliski

Harris Features (in red)

Courtesy: Steve Seitz and Richard Szeliski

Invariance

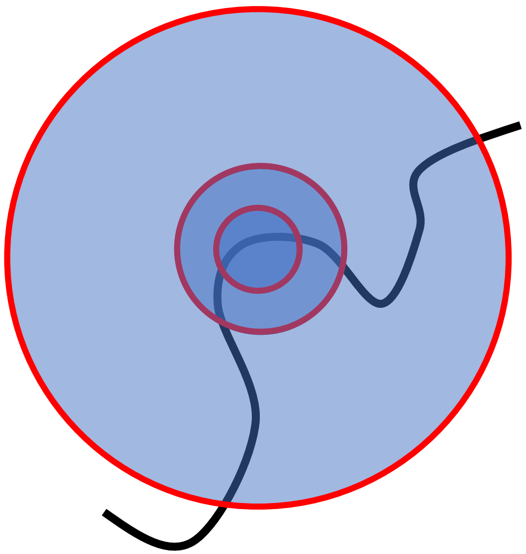
- Suppose you **rotate** the image by some angle
  - Will you still pick up the same features?

- What if you change the brightness?

- Scale?

Scale invariant detection

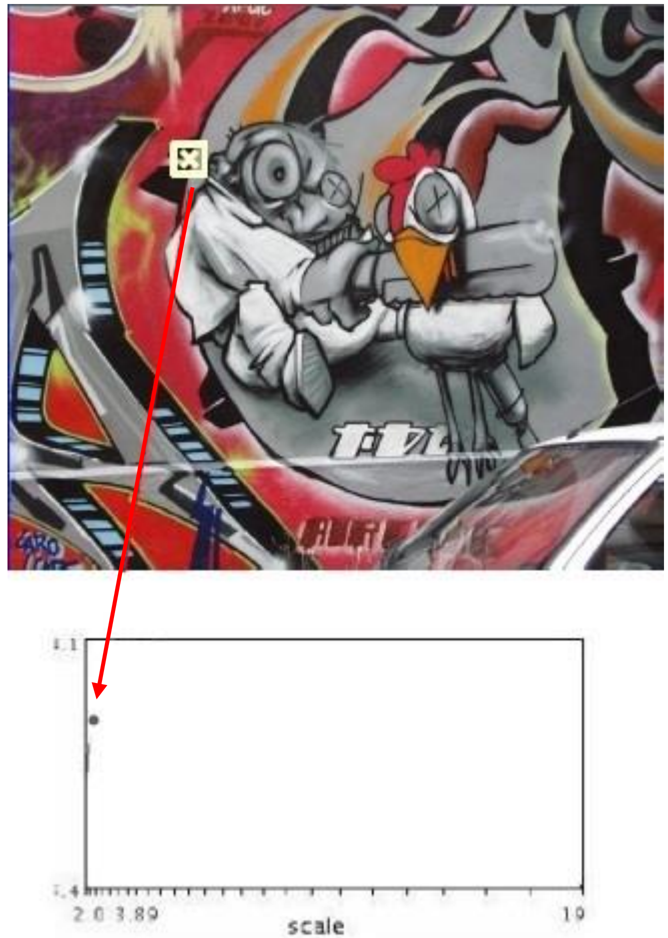
- Suppose you're looking for corners



- Key idea: find scale that gives local maximum of f
  - f is a local maximum in both position and scale

Courtesy: Steve Seitz and Richard Szeliski

Automatic Scale Selection

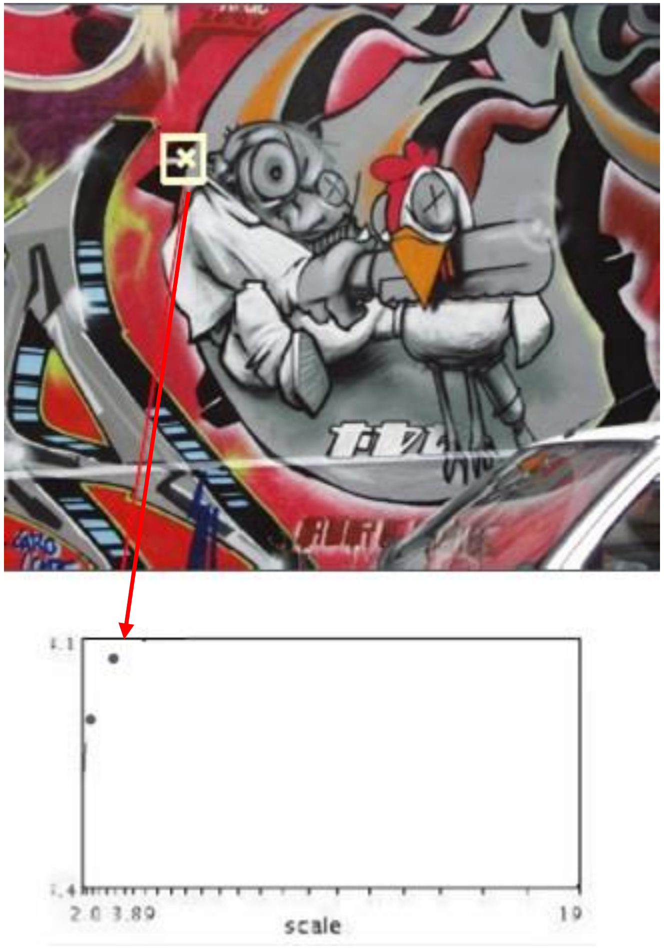


Lindeberg et al., 1996

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

Courtesy: Steve Seitz and Richard Szeliski

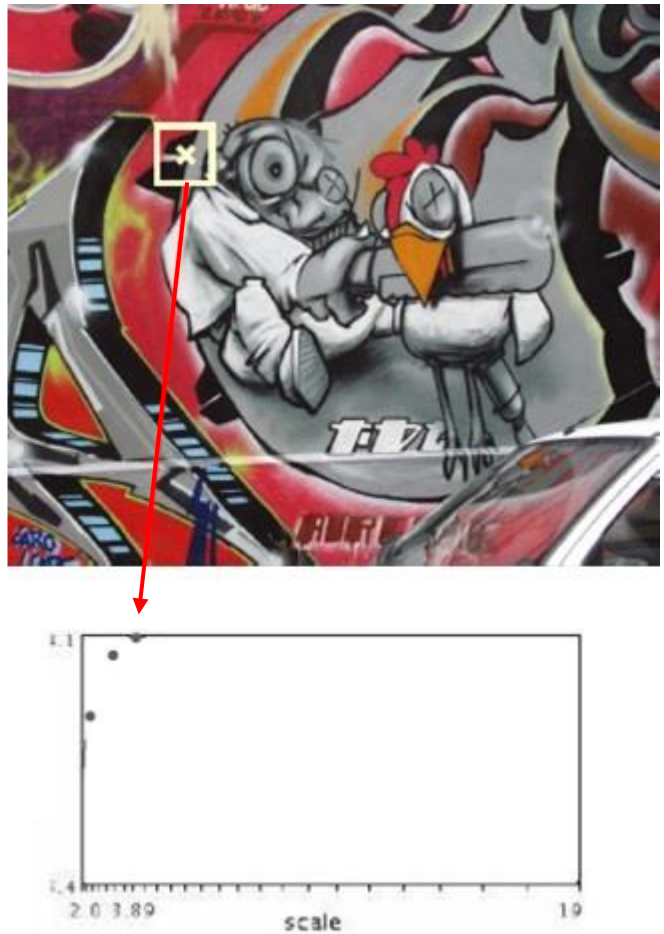
Automatic Scale Selection



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

Courtesy: Steve Seitz and Richard Szeliski

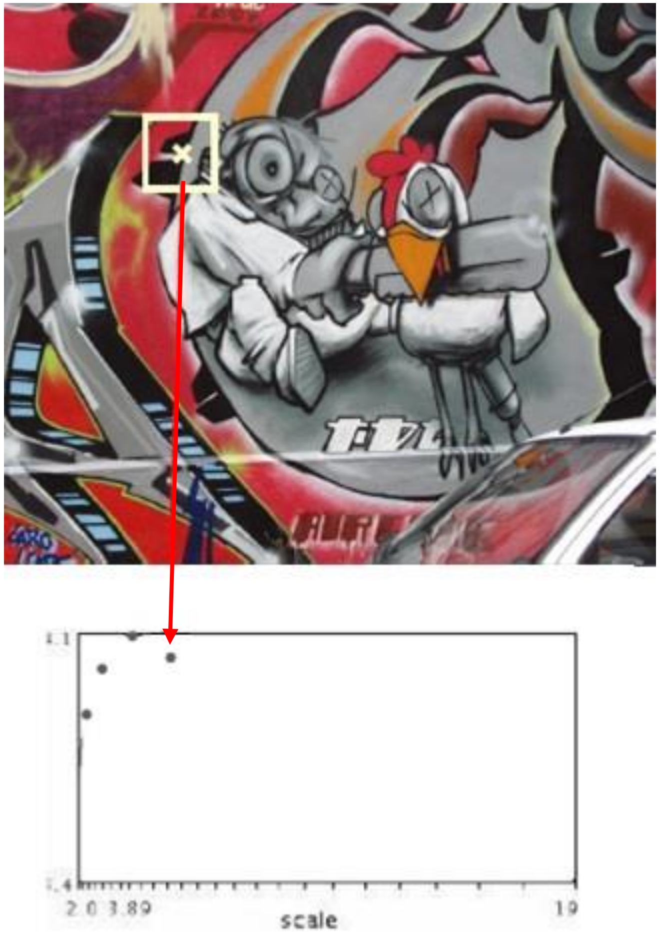
Automatic Scale Selection



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

Courtesy: Steve Seitz and Richard Szeliski

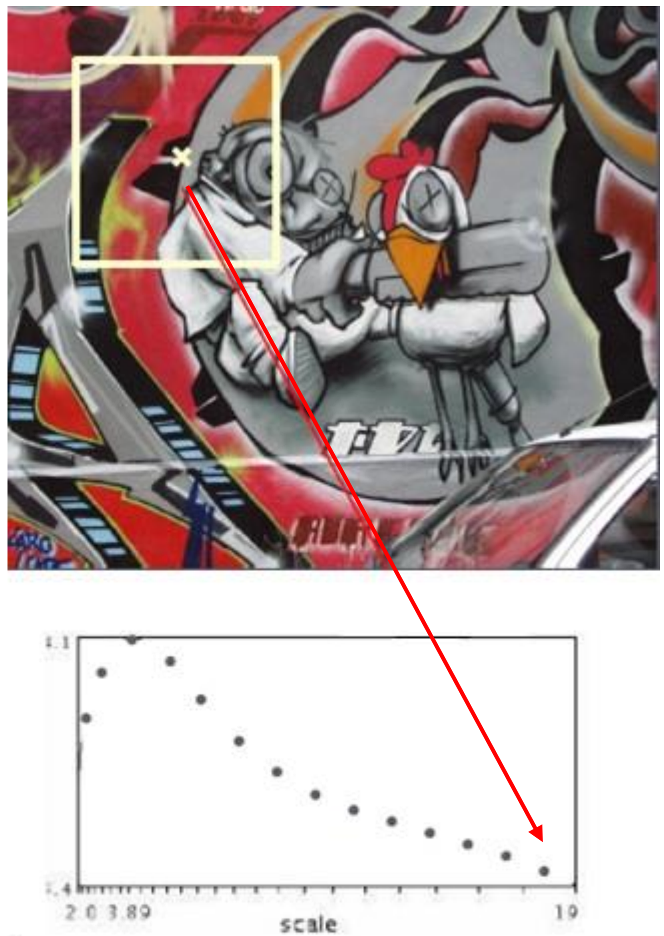
Automatic Scale Selection



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

Courtesy: Steve Seitz and Richard Szeliski

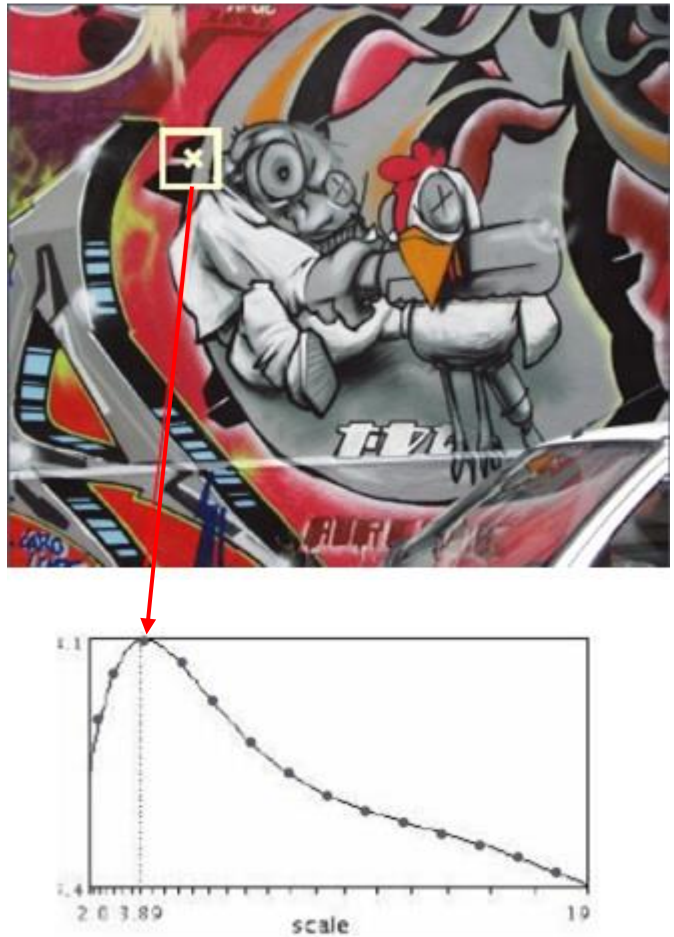
Automatic Scale Selection



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

Courtesy: Steve Seitz and Richard Szeliski

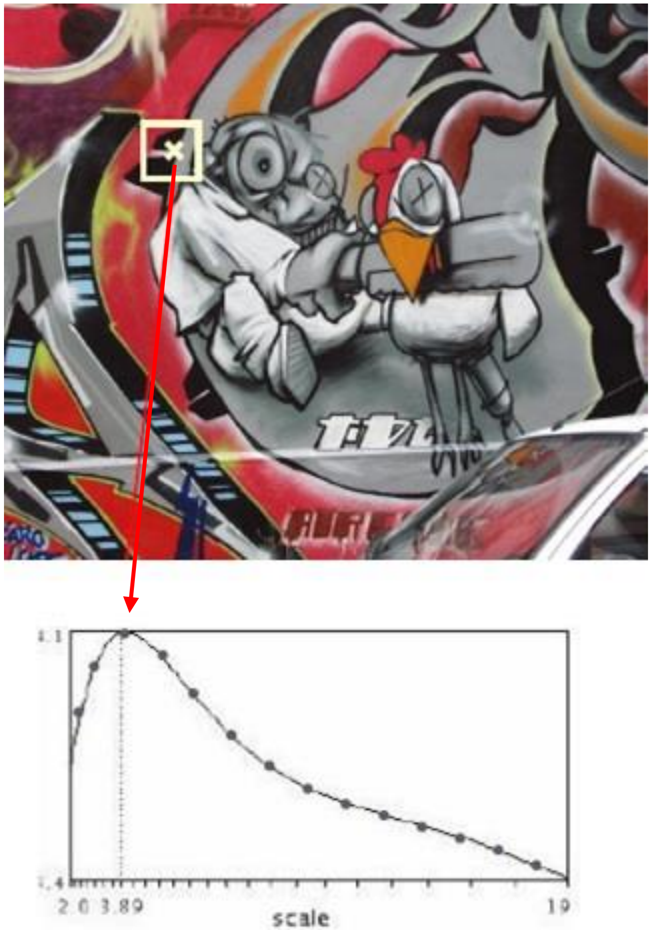
Automatic Scale Selection



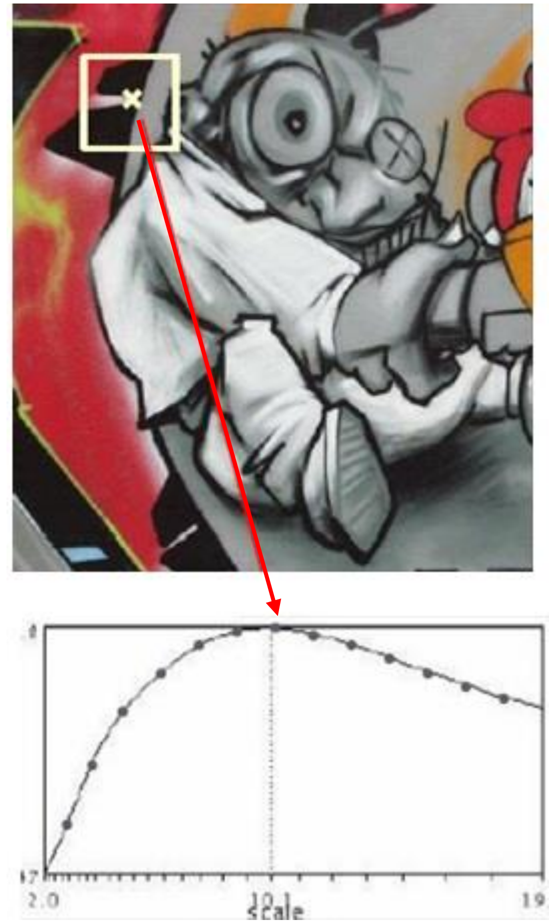
$$f(I_{i_1 \dots i_m}(x, \sigma))$$

Courtesy: Steve Seitz and Richard Szeliski

Automatic Scale Selection



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

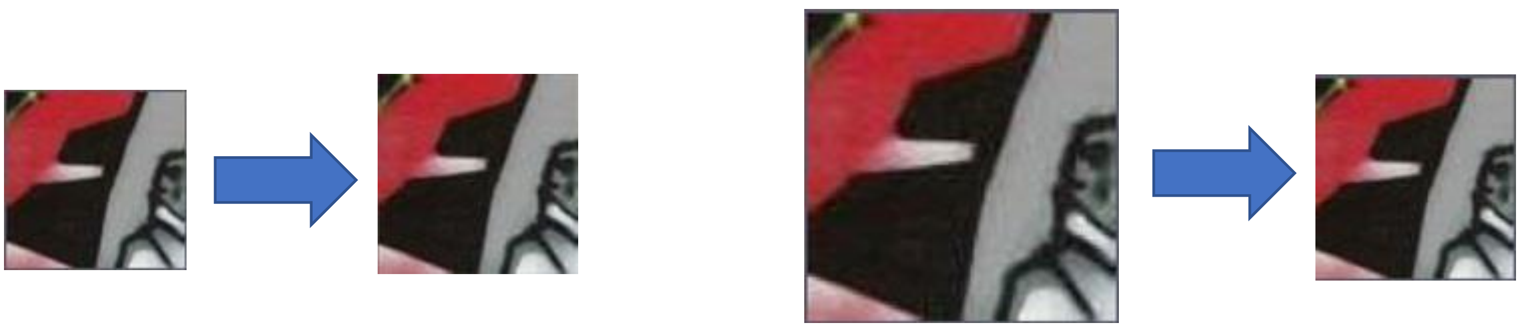


$$f(I_{i_1 \dots i_m}(x', \sigma'))$$

Courtesy: Steve Seitz and Richard Szeliski

Automatic Scale Selection

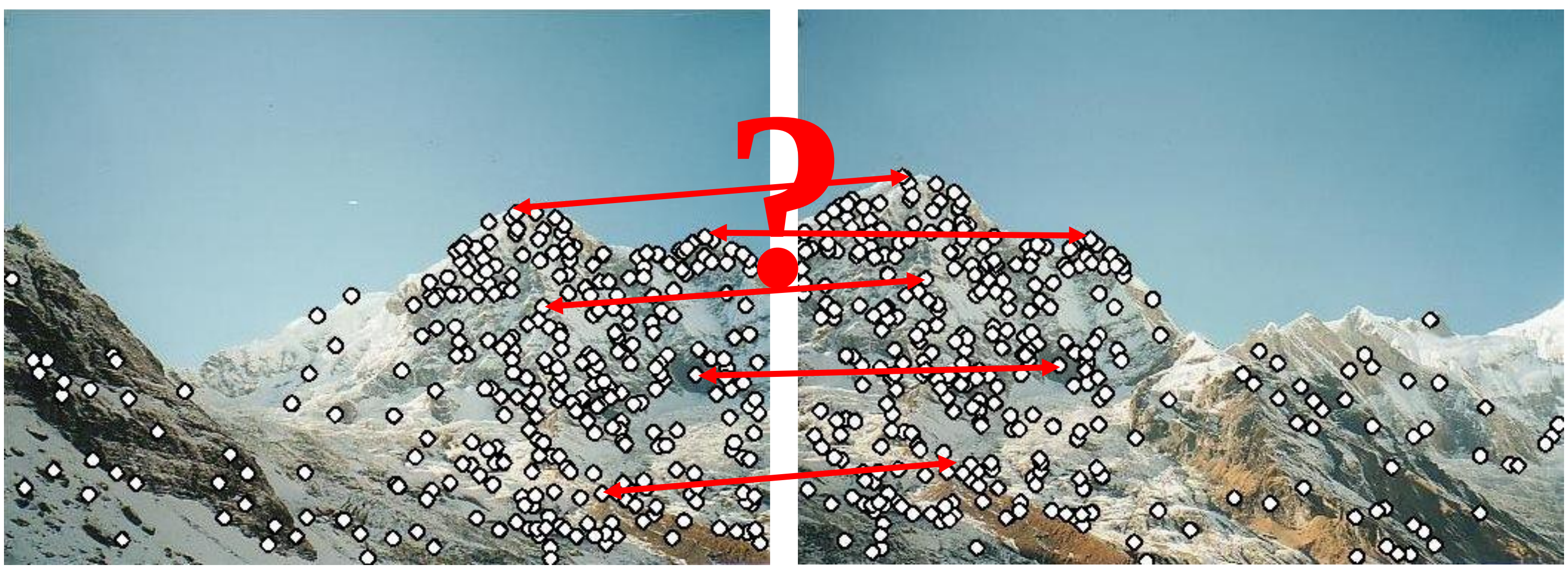
- Normalize: rescale to fixed size



Courtesy: Steve Seitz and Richard Szeliski

Feature Descriptors

- We know how to detect good points
- Next question: How to match them?



Courtesy: Steve Seitz and Richard Szeliski

Next Lecture

- Visual Descriptor and Matching
  - SIFT Descriptor
  - RANSAC
  - Introduction to PnP (2D-3D pose estimation)