



明



明确发展定位 提升综合实

How to stitch them together?

SSE, Tongji Univ.



SSE, Tongji Univ.



Lecture 2

Local Interest Point Detectors

School of Software Engineering
Tongji University
Fall 2023



Content

- Local Invariant Features
 - Motivation
 - Requirements
 - Invariance
- Harris Corner Detector
- Scale Invariant Point Detection
 - Automatic scale selection
 - Laplacian-of-Gaussian detector
 - Difference-of-Gaussian detector

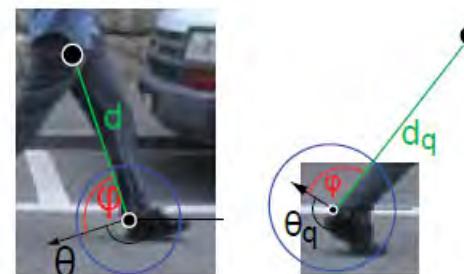


Motivation

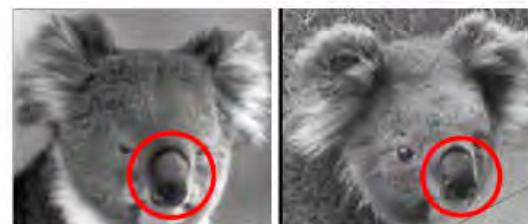
- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
 - Occlusions



- Articulation



- Intra-category variations





Motivation

Application: Image Matching



by [Diva Sian](#)



by [swashford](#)



Motivation

Application: Image Matching

Harder Case



by [Diva Sian](#)



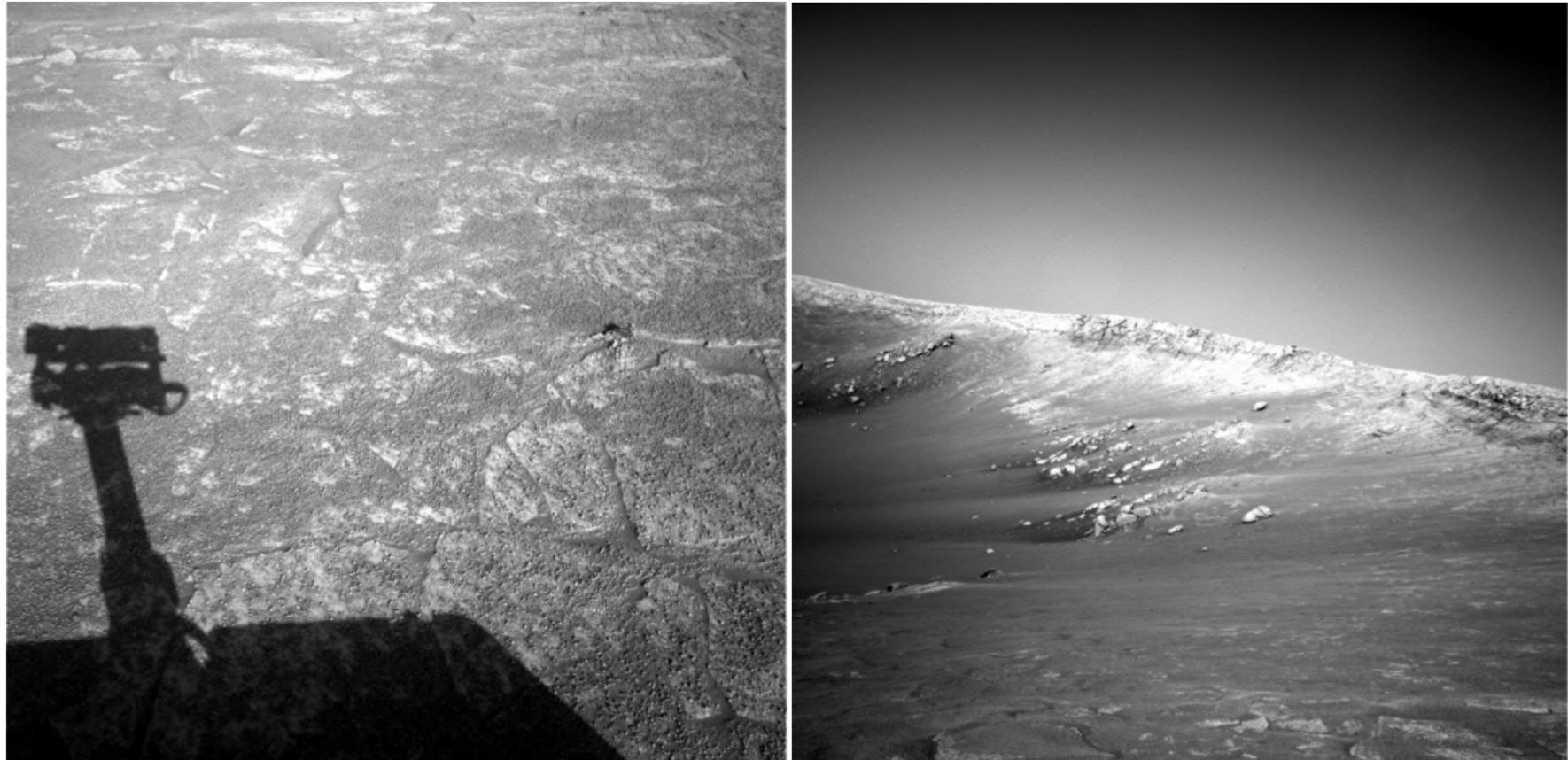
by [scgbt](#)

Steve Seitz



Motivation

Application: Image Matching

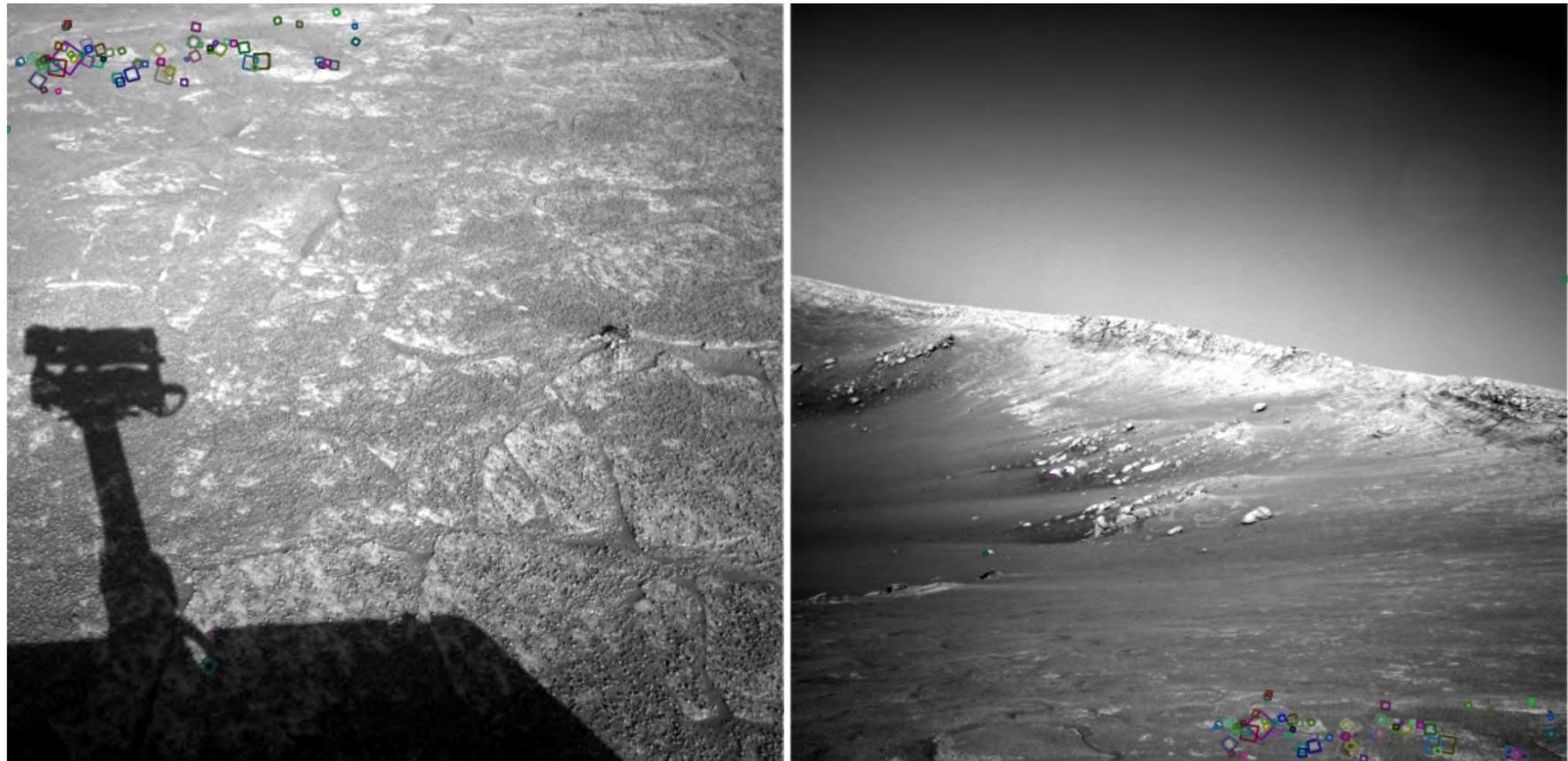


NASA Mars Rover Images



Motivation

Application: Image Matching (Look for tiny colored squares)

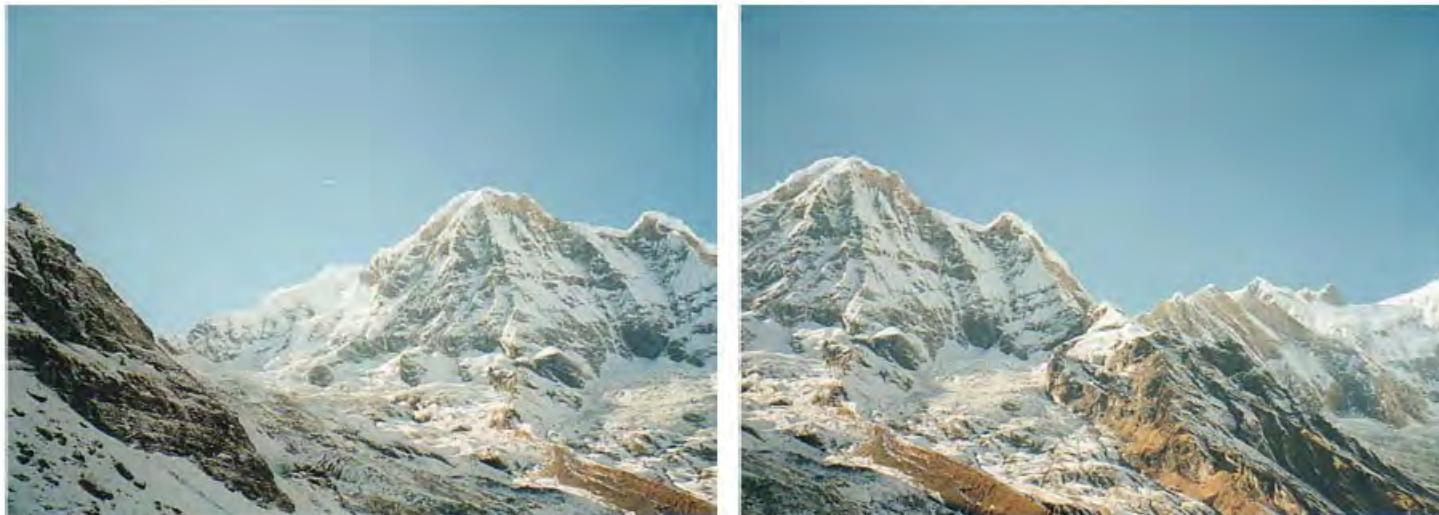


NASA Mars Rover images with SIFT matches



Motivation

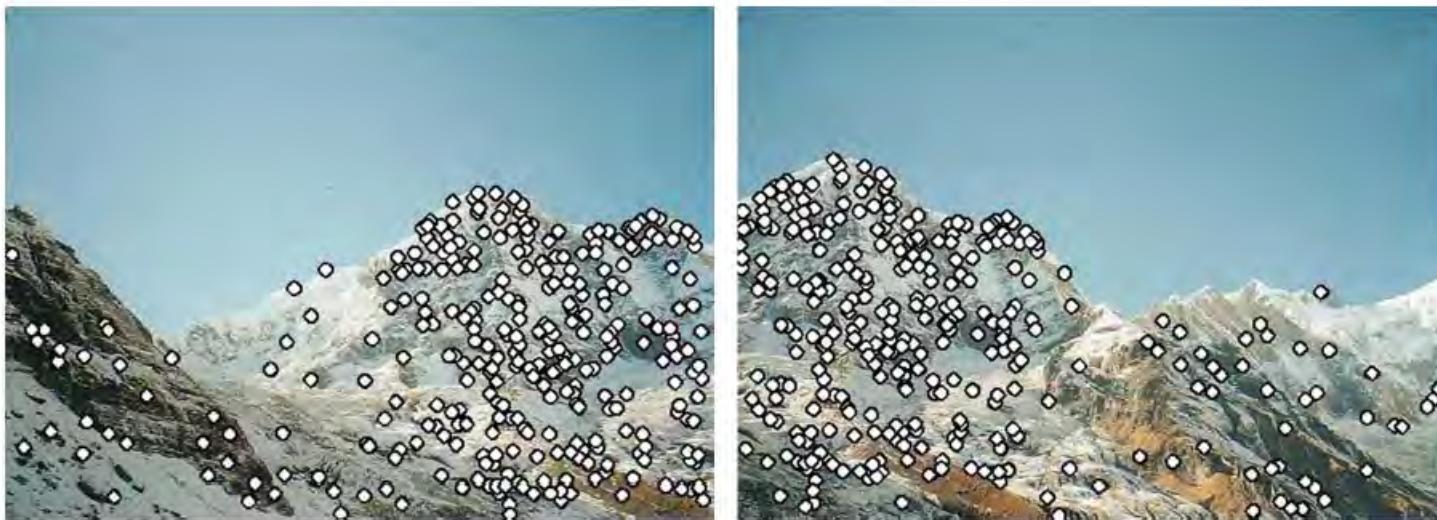
- Panorama stitching
 - We have two images – how do we combine them?





Motivation

- Panorama stitching
 - We have two images – how do we combine them?

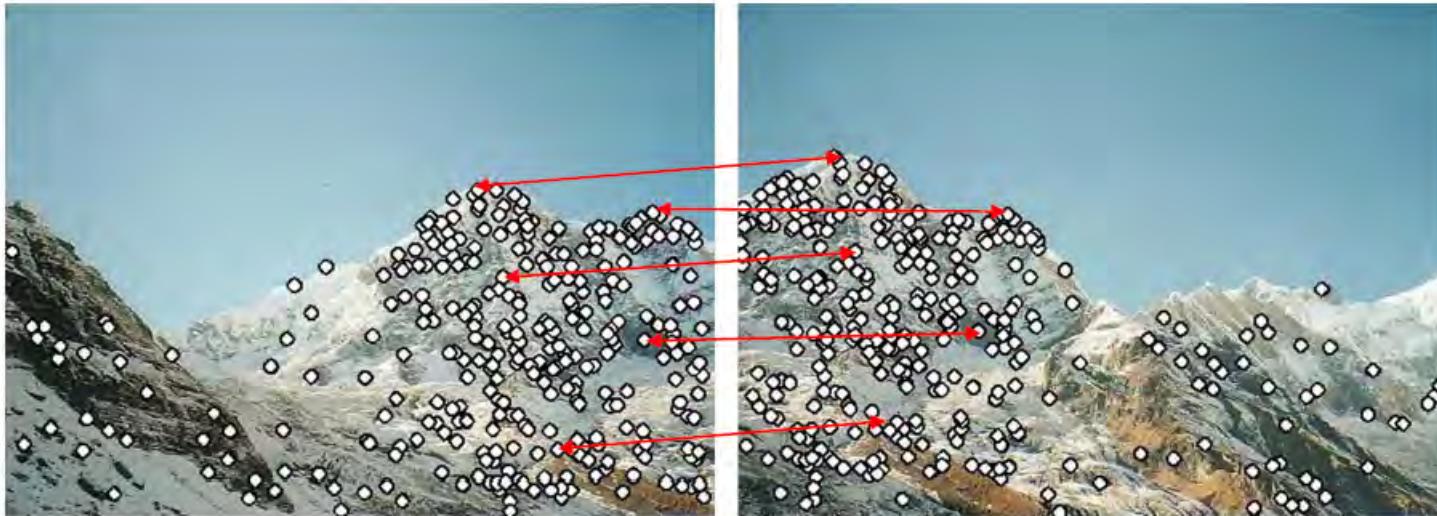


- Procedure:
 - Detect feature points in both images



Motivation

- Panorama stitching
 - We have two images – how do we combine them?



- Procedure:
 - Detect feature points in both images
 - Find corresponding pairs



Motivation

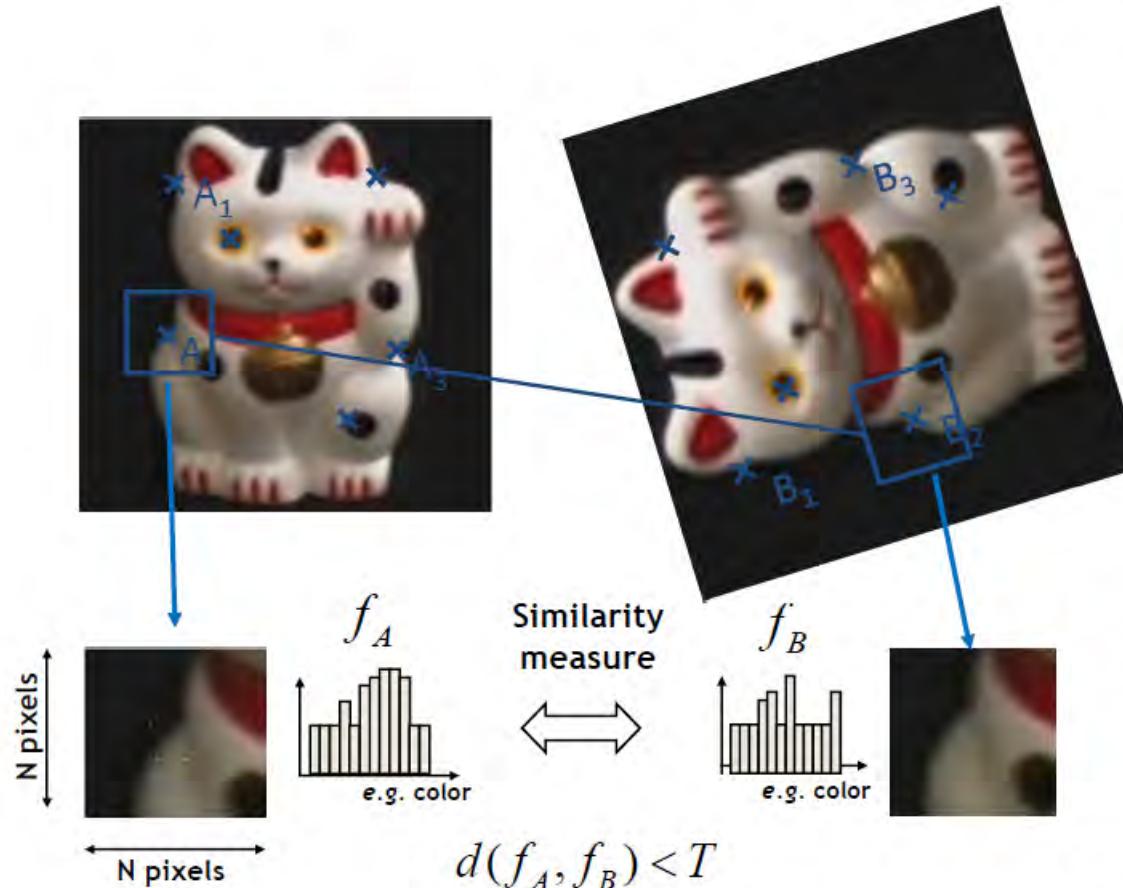
- Panorama stitching
 - We have two images – how do we combine them?



- Procedure:
 - Detect feature points in both images
 - Find corresponding pairs
 - Use these pairs to align the images



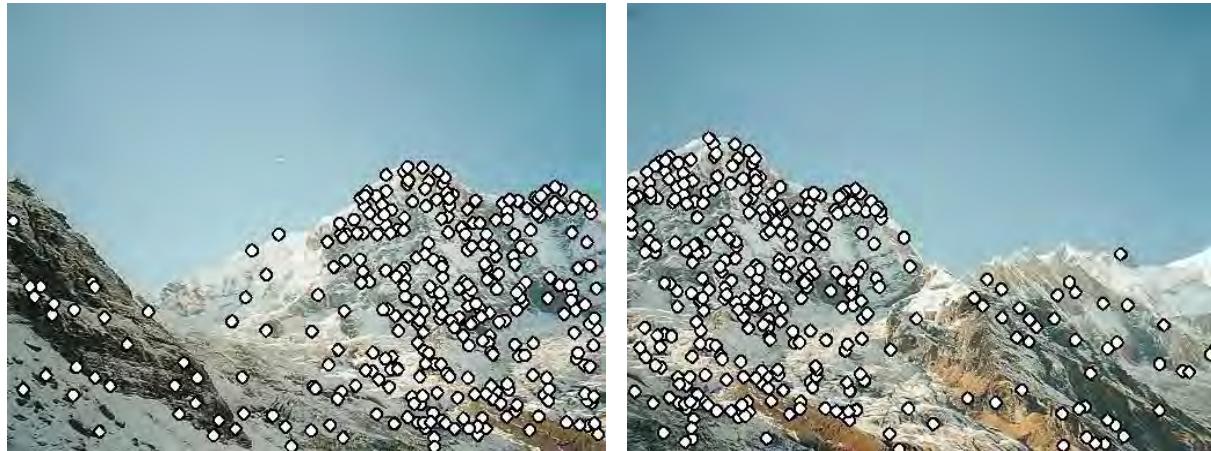
General Approach for Image Matching



Source: B. Leibe



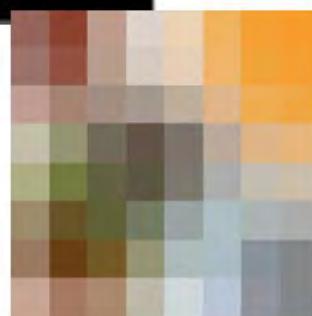
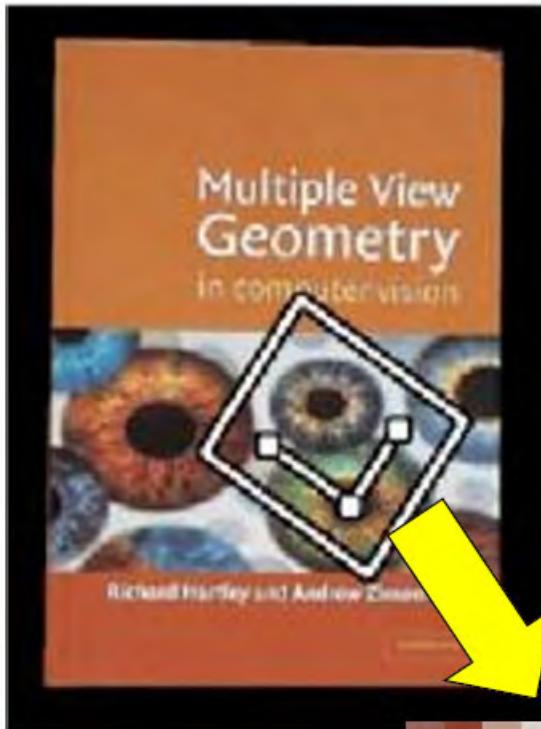
Characteristics of Good Features



- **Repeatability**
 - The same feature can be found in several images despite geometric and photometric transformations
- **Saliency**
 - Each feature has a distinctive description
- **Compactness and efficiency**
 - Many fewer features than image pixels
- **Locality**
 - A feature occupies a relatively small area of the image; robust to clutter and occlusion

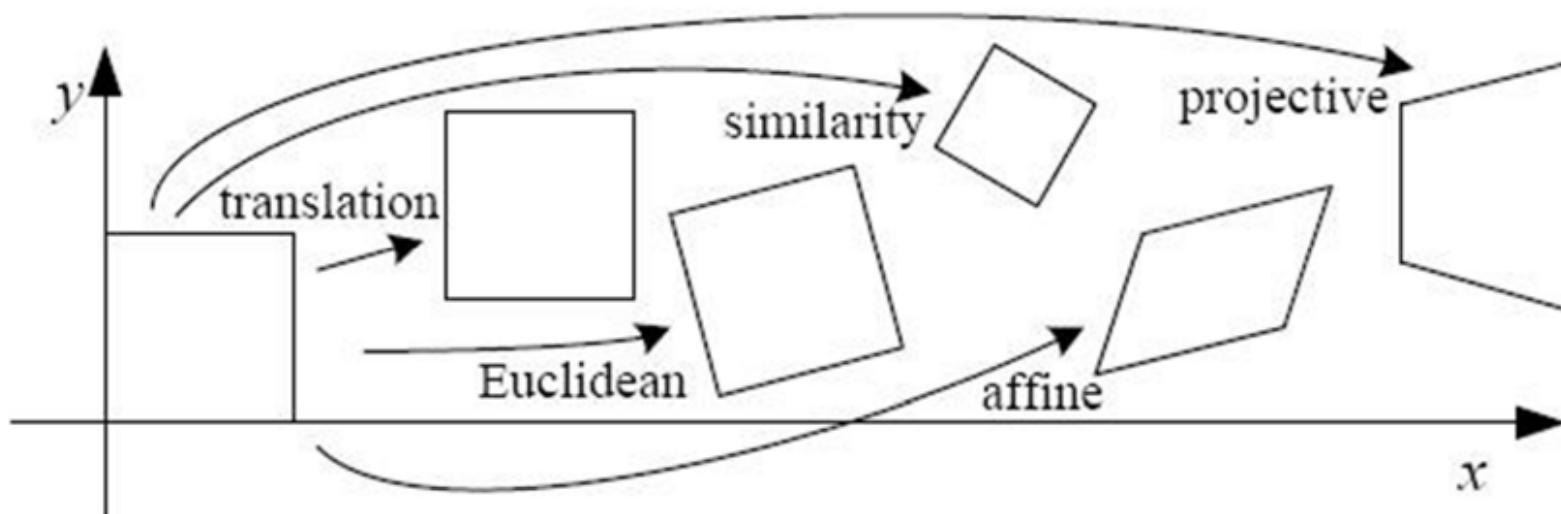


Invariance: Geometric Transformations



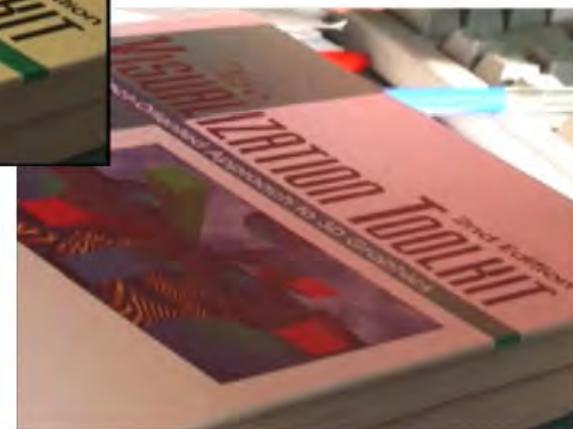
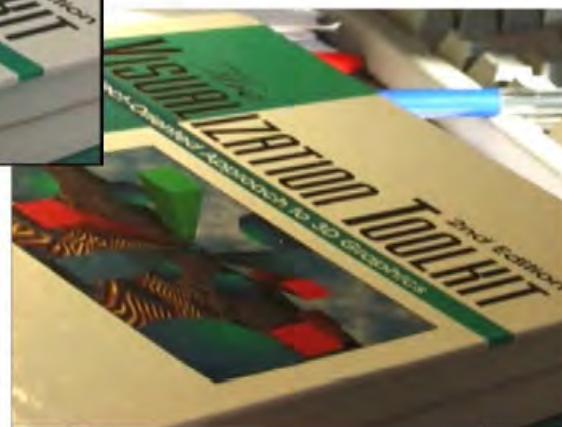
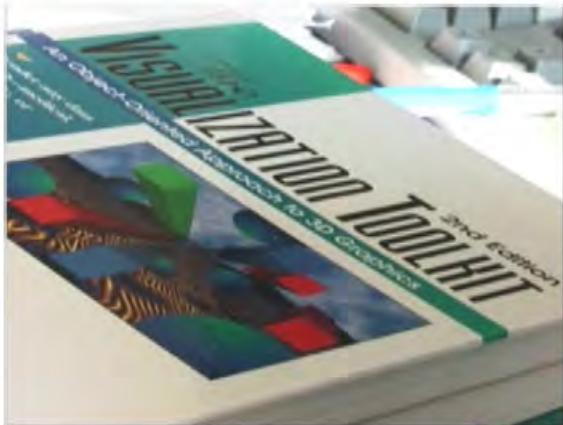


Level of Geometric Invariance





Invariance: Photometric Transformations



- Often modeled as a linear transformation:
 - Scaling + Offset



Applications

Feature points are used for:

- Motion tracking
- Image alignment
- 3D reconstruction
- Object recognition
- Indexing and database retrieval
- Robot navigation



Content

- Local Invariant Features
 - Motivation
 - Requirements
 - Invariance
- Harris Corner Detector
- Scale Invariant Point Detection
 - Automatic scale selection
 - Laplacian-of-Gaussian detector
 - Difference-of-Gaussian detector



Finding Corners



My office,
5:30PM, Sep. 18, 2011



SSE, Tongji Univ.



Finding Corners

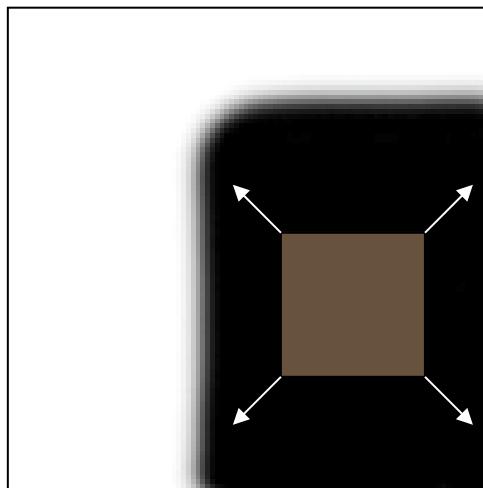
- Key property: in the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive

C. Harris and M. Stephens. [A Combined Corner and Edge Detector.](#)
Proceedings of the 4th Alvey Vision Conference: pages 147—151, 1988.

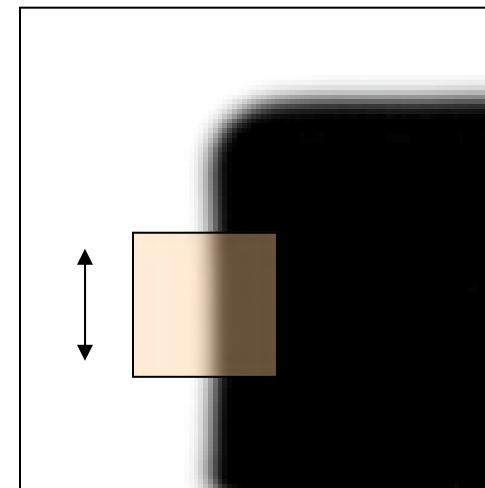


Corner Detection: Basic Idea

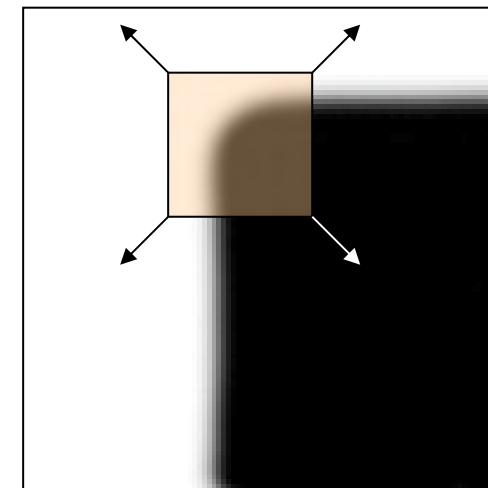
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



“flat” region:
no change in
all directions



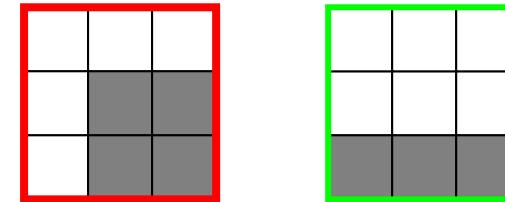
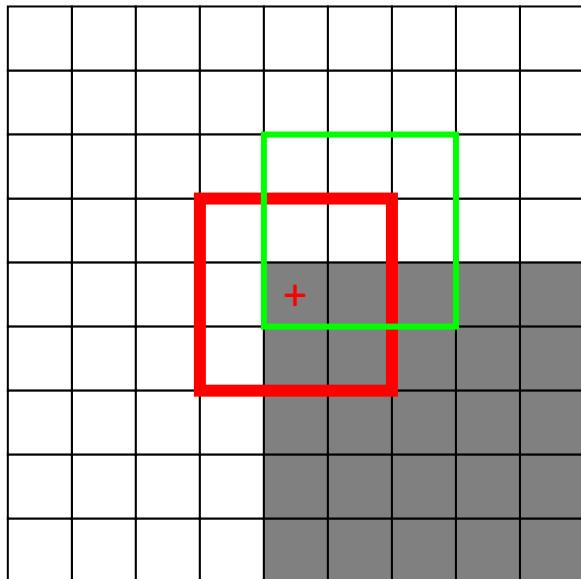
“edge”:
no change along
the edge
direction



“corner”:
significant change
in all directions



Harris Detector: Basic Idea



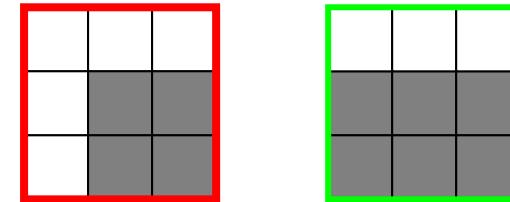
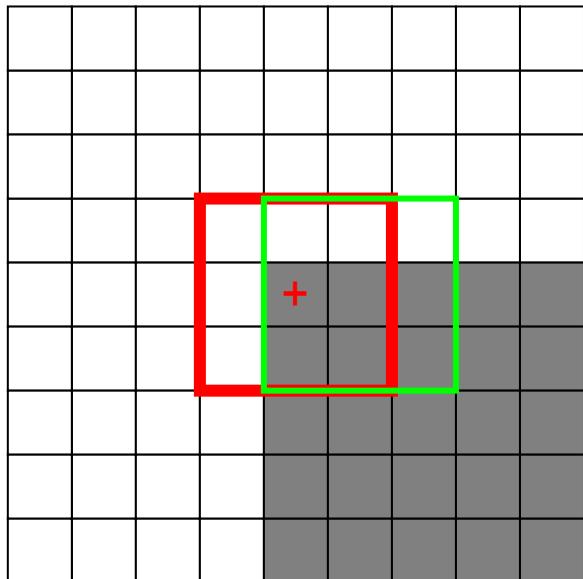
Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



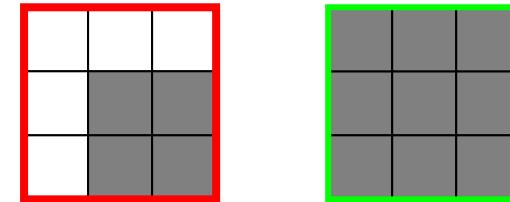
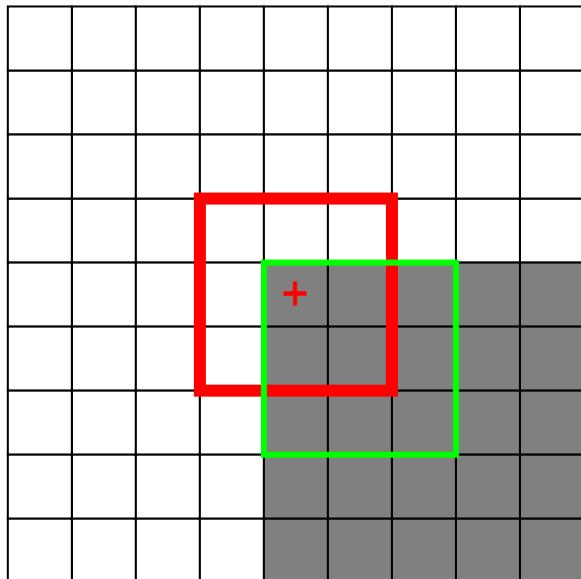
Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



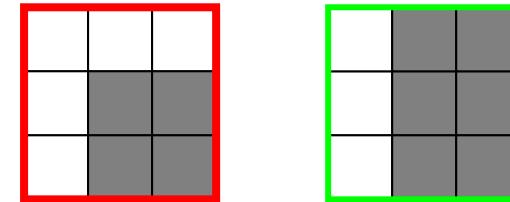
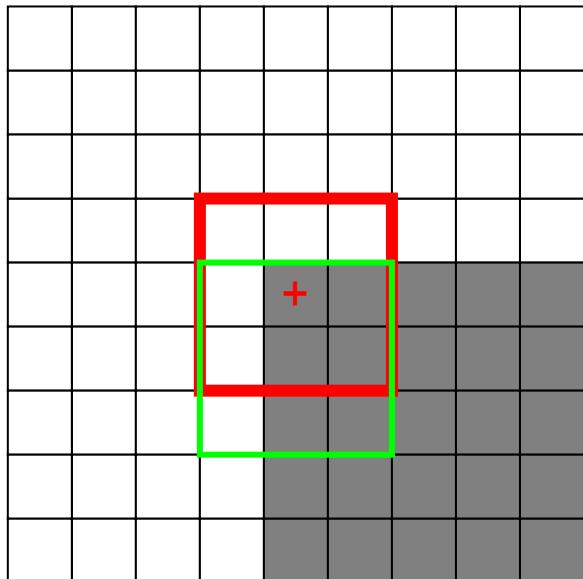
Difference = 5

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



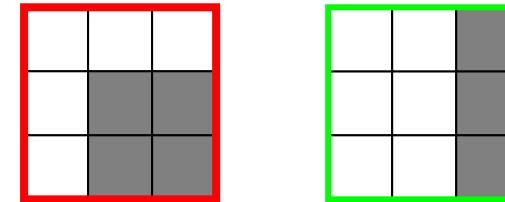
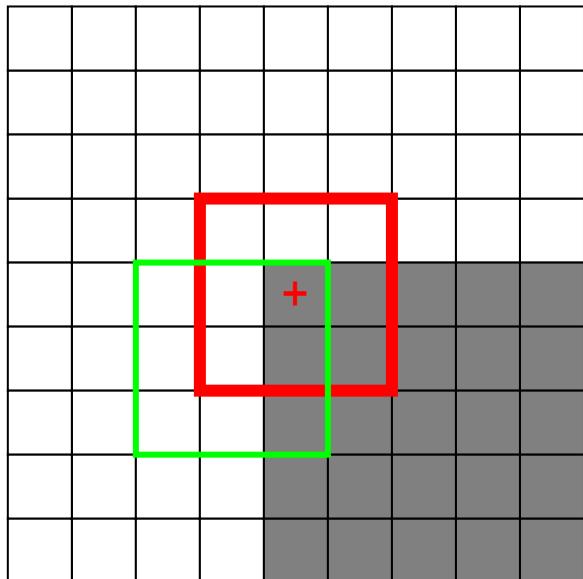
Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



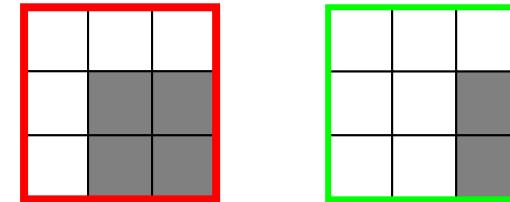
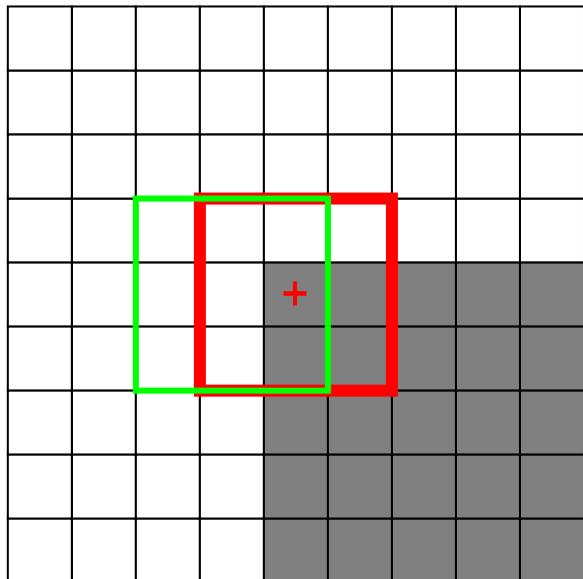
Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



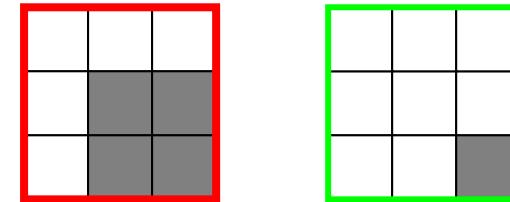
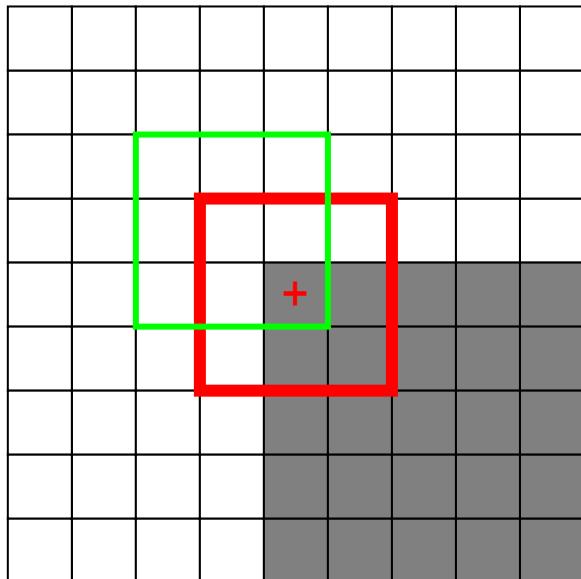
Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



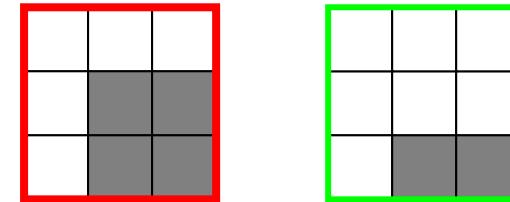
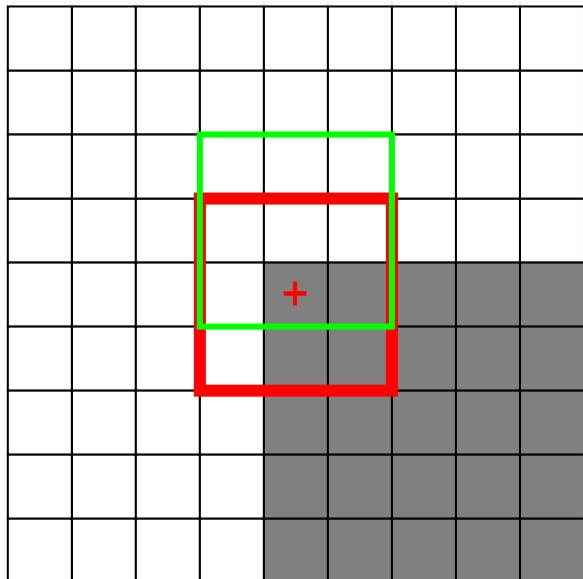
Difference = 3

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Detector: Basic Idea



Difference = 2

Demo of a point + with well distinguished neighborhood.

Moving the window in any direction will result in a large intensity change.



Harris Corner Detection: Mathematics

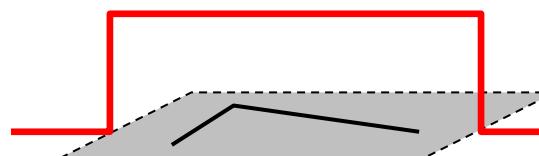
Change in appearance of a local patch (defined by a window w) centered at p for the shift $(\Delta x, \Delta y)$:

$$S_w(\Delta x, \Delta y) = \sum_{(x_i, y_i) \in w} (f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y))^2$$

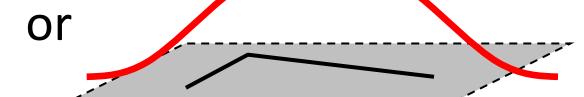
↑
↑
↑

Window function Intensity Shifted intensity

Window function \mathcal{W} =



1 in window, 0 outside



Gaussian



Harris Corner Detection: Mathematics

$$S_w(\Delta x, \Delta y) = \sum_{(x_i, y_i) \in w} (f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y))^2 \quad (1)$$

$$\approx \sum_{(x_i, y_i) \in w} \left(f(x_i, y_i) - f(x_i, y_i) - \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \quad (2)$$

$$= \sum_{(x_i, y_i) \in w} \left(\left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y} \right] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \quad (\text{Due to } |\mathbf{u}|^2 = \mathbf{u}^T \mathbf{u})$$

$$= [\Delta x, \Delta y] \left(\sum_{(x_i, y_i) \in w} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} \\ \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)$$

$$= [\Delta x \ \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$



Harris Corner Detection

$M =$

$$\begin{bmatrix} \sum_{(x_i, y_i) \in w} \left(\frac{\partial f(x_i, y_i)}{\partial x} \right)^2 & \sum_{(x_i, y_i) \in w} \left(\frac{\partial f(x_i, y_i)}{\partial x} \cdot \frac{\partial f(x_i, y_i)}{\partial y} \right) \\ \sum_{(x_i, y_i) \in w} \left(\frac{\partial f(x_i, y_i)}{\partial x} \cdot \frac{\partial f(x_i, y_i)}{\partial y} \right) & \sum_{(x_i, y_i) \in w} \left(\frac{\partial f(x_i, y_i)}{\partial y} \right)^2 \end{bmatrix}$$



Harris Corner Detection

$$S(\Delta x, \Delta y) \cong [\Delta x, \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

where, $M = \begin{bmatrix} \sum_{(x_i, y_i) \in w} (I_x)^2 & \sum_{(x_i, y_i) \in w} (I_x I_y) \\ \sum_{(x_i, y_i) \in w} (I_x I_y) & \sum_{(x_i, y_i) \in w} (I_y)^2 \end{bmatrix}$

$S(\Delta x, \Delta y) = 1$ actually is the ellipse equation.

The shape of the ellipse is determined by M .

M can be proved to be a positive semi-definite matrix

Assignment

In practice, M is positive definite nearly for sure, then $[\Delta x \ \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 1$ represents an ellipse



Harris Corner Detection

The “cornerness” of the window w is reflected in M

Suppose there are two local windows w_1 and w_2 ; consider the cases when the moving of the two windows leads to the intensity change equals to 1. The moving vector $[\Delta x, \Delta y]$ of each window satisfies the ellipse equation. Thus,

For w_1 ,

$$[\Delta x, \Delta y] M_1 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 1 \rightarrow \text{Elliptical shape}$$

Which window has higher cornerness?

For w_2 ,

$$[\Delta x, \Delta y] M_2 \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 1 \rightarrow \text{Elliptical shape}$$





Harris Corner Detection

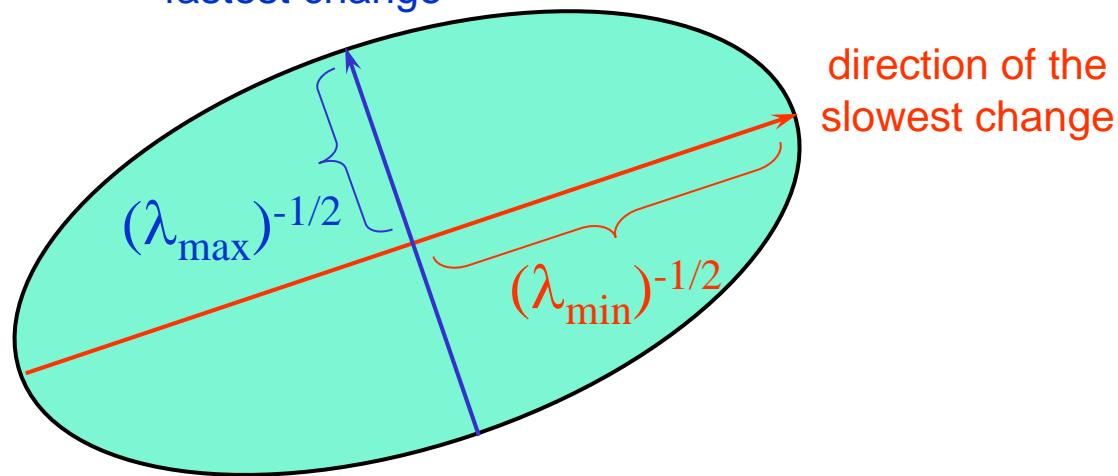
Diagonalization of M :

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$



The axis lengths of the ellipse are determined by the eigenvalues and the orientation is determined by R

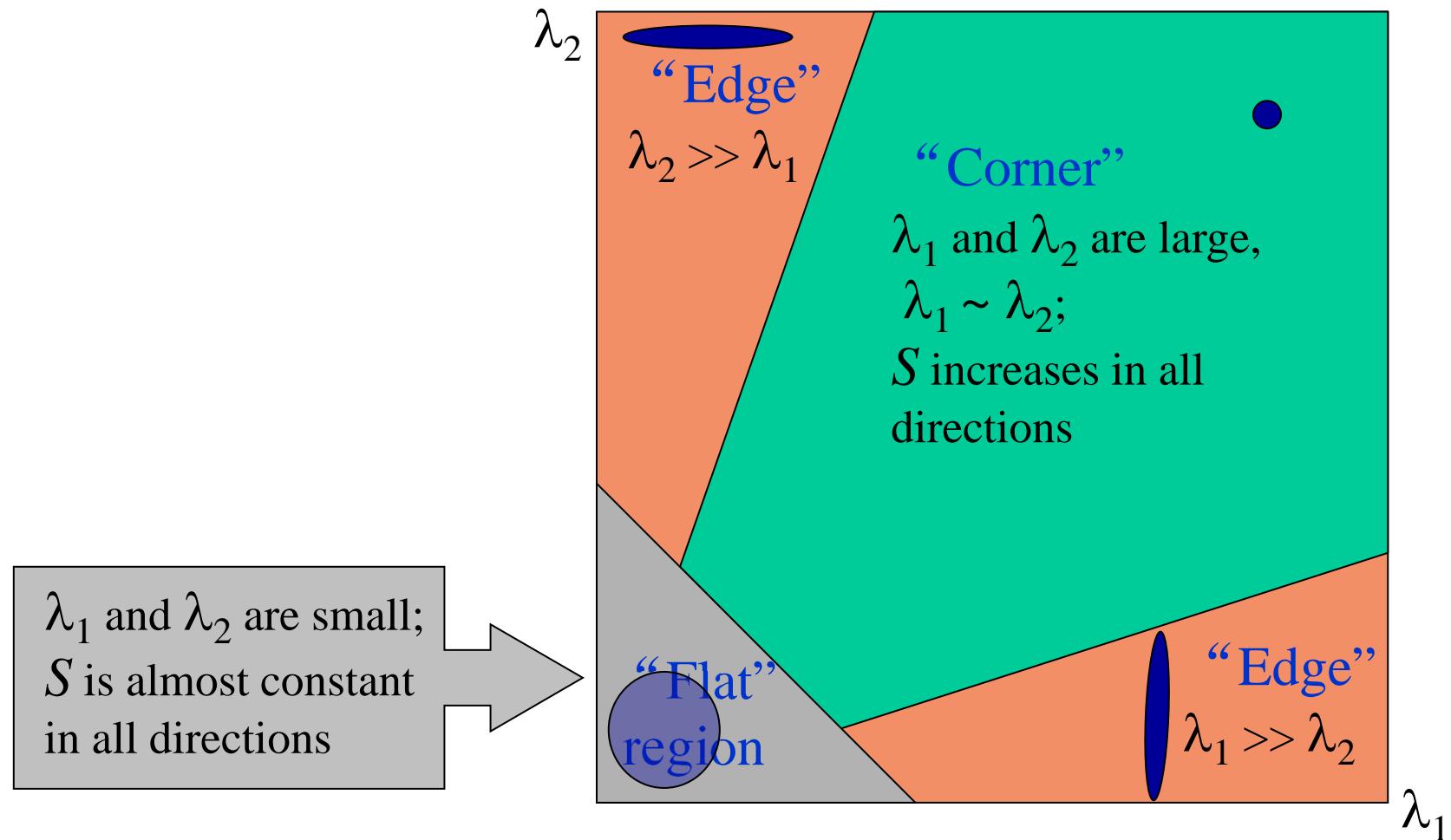
direction of the
fastest change





Interpreting the eigenvalues

Classification of image points using eigenvalues of M :





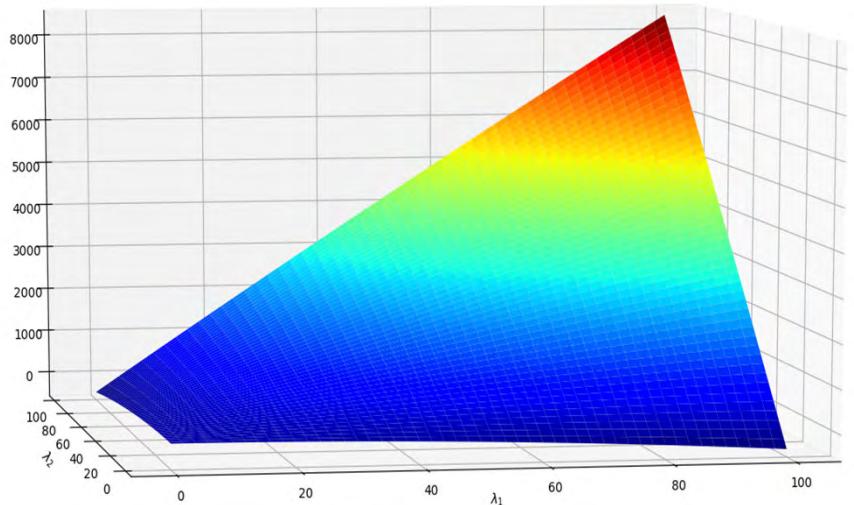
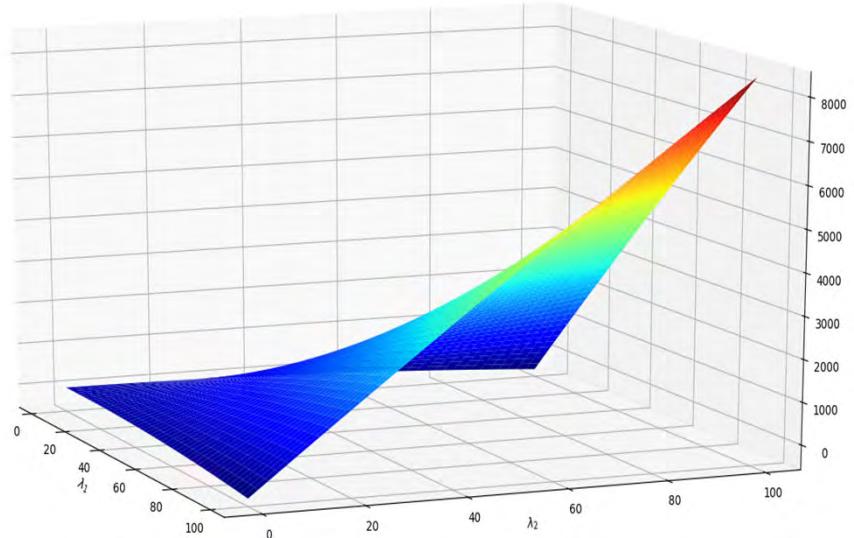
Corner response function

Measure of corner response:

$$R = \det \mathbf{M} - k(\text{trace} \mathbf{M})^2$$

$$\det \mathbf{M} = \lambda_1 \lambda_2$$
$$\text{trace} \mathbf{M} = \lambda_1 + \lambda_2$$

(k – empirical constant, $k = 0.04\text{-}0.06$)



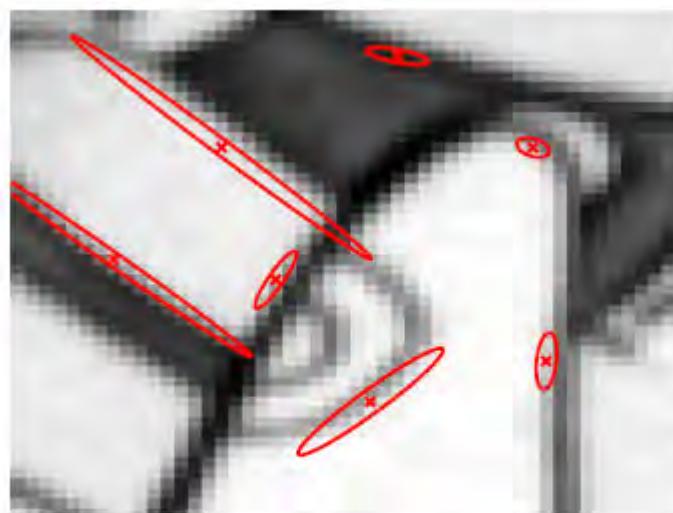


Harris corner detector--illustration

Ellipse with equation : $[\Delta x, \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 1$



flat region
both eigenvalues small



edge
one small, one large



Harris corner detector--illustration

Ellipse with equation : $[\Delta x, \Delta y] M \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = 1$



corner
both eigenvalues large



Harris corner detector-Algorithm

- Compute second moment matrix
(autocorrelation matrix)

$$M(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives



2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$



4. Cornerness function - two strong eigenvalues

$$\begin{aligned} R &= \det[M(\sigma_I, \sigma_D)] - \alpha[\text{trace}(M(\sigma_I, \sigma_D))]^2 \\ &= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2 \end{aligned}$$

5. Perform non-maximum suppression





Harris Detector: Steps

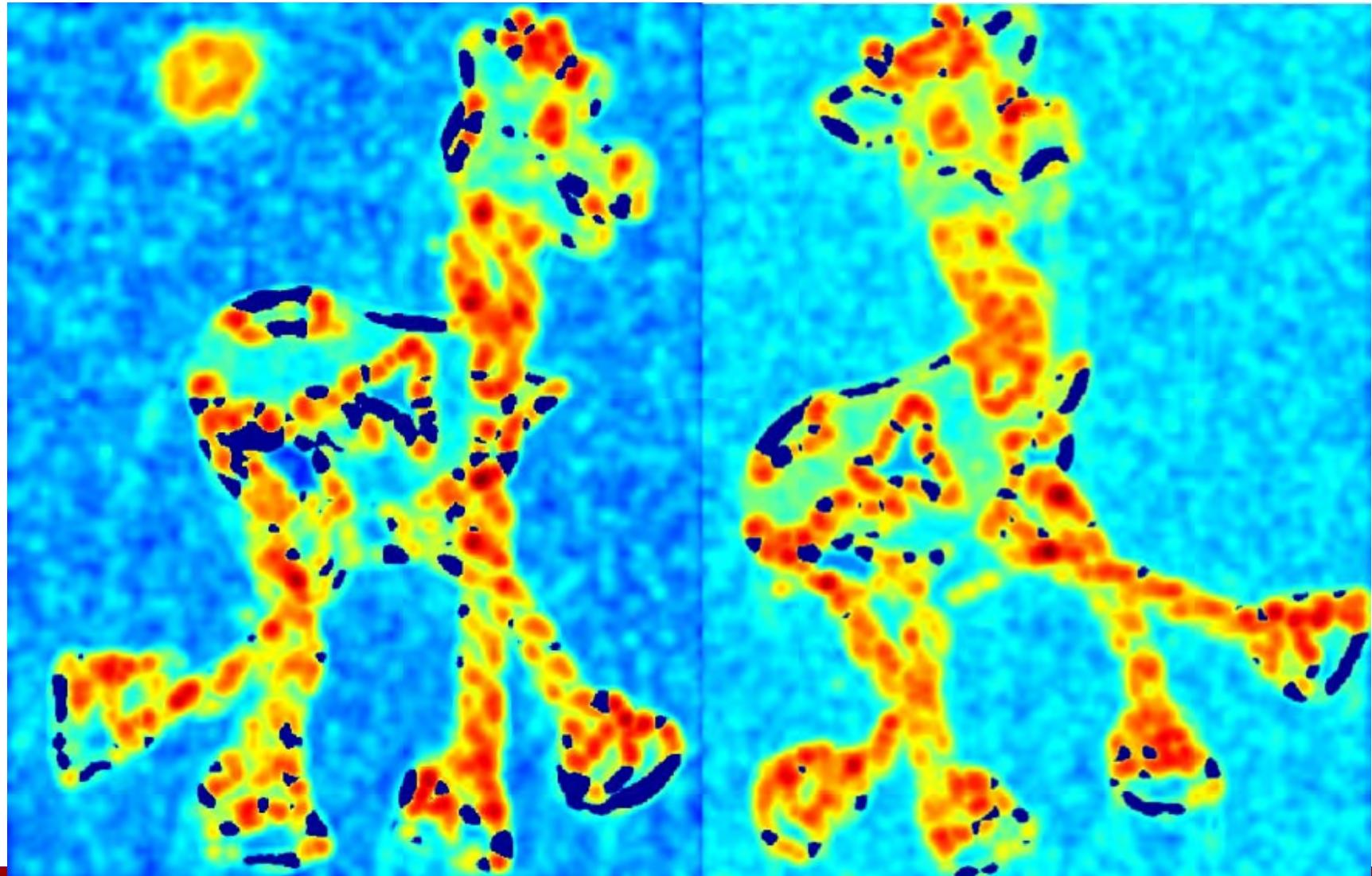


SSE, Tongji Univ.



Harris Detector: Steps

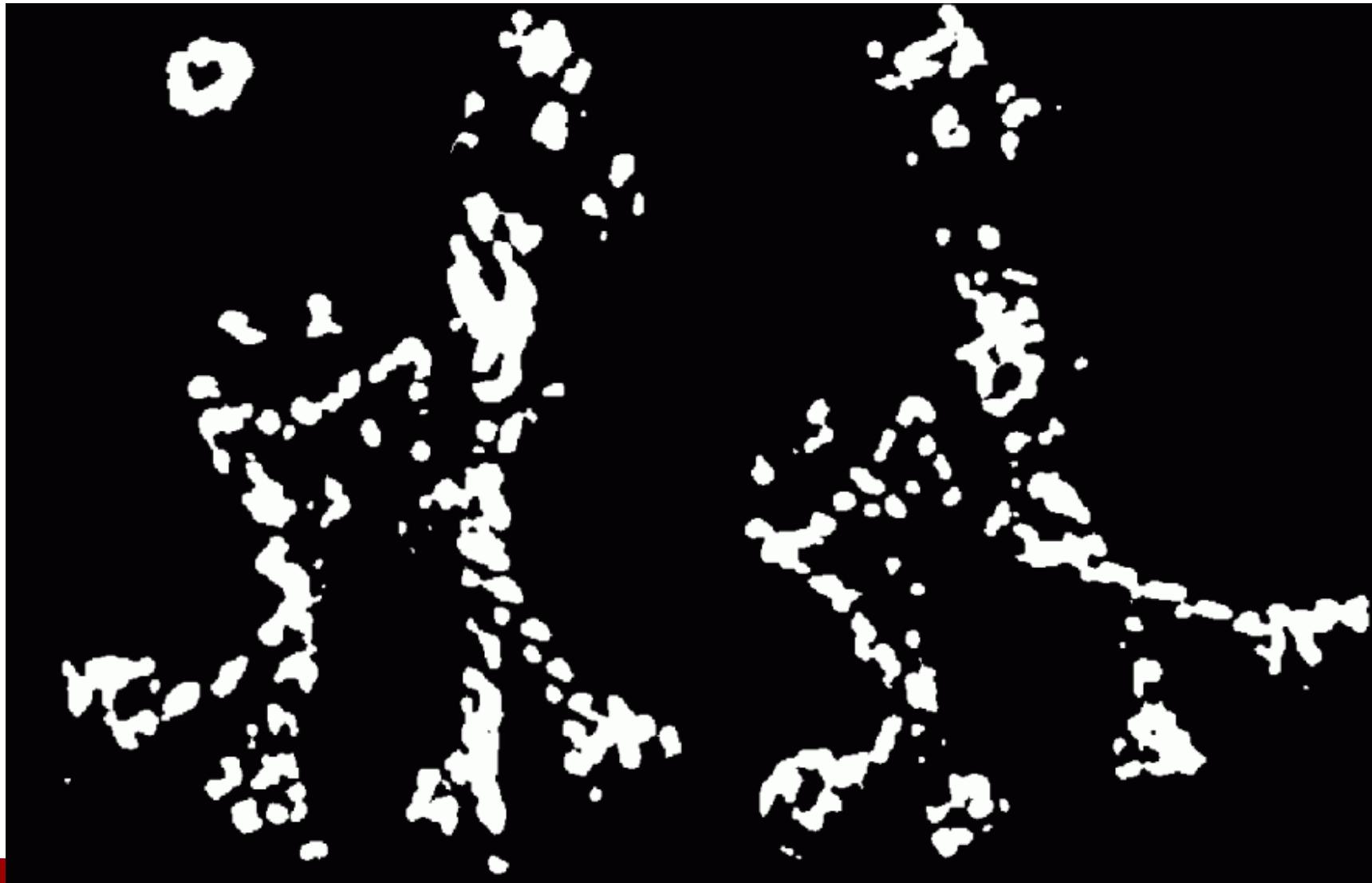
Compute corner response R





Harris Detector: Steps

Find points with large corner response: $R > \text{threshold}$





Harris Detector: Steps

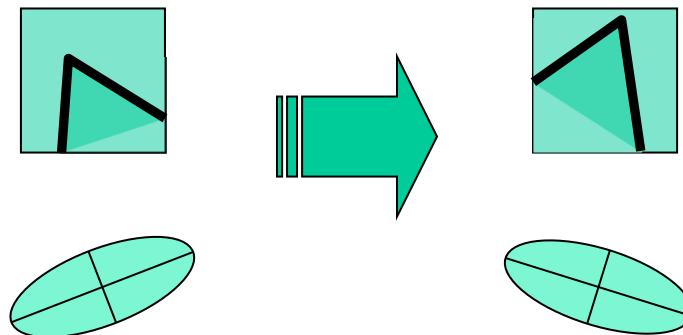
Take only the points of local maxima of R





Harris Detector: Some Properties

Rotation invariance



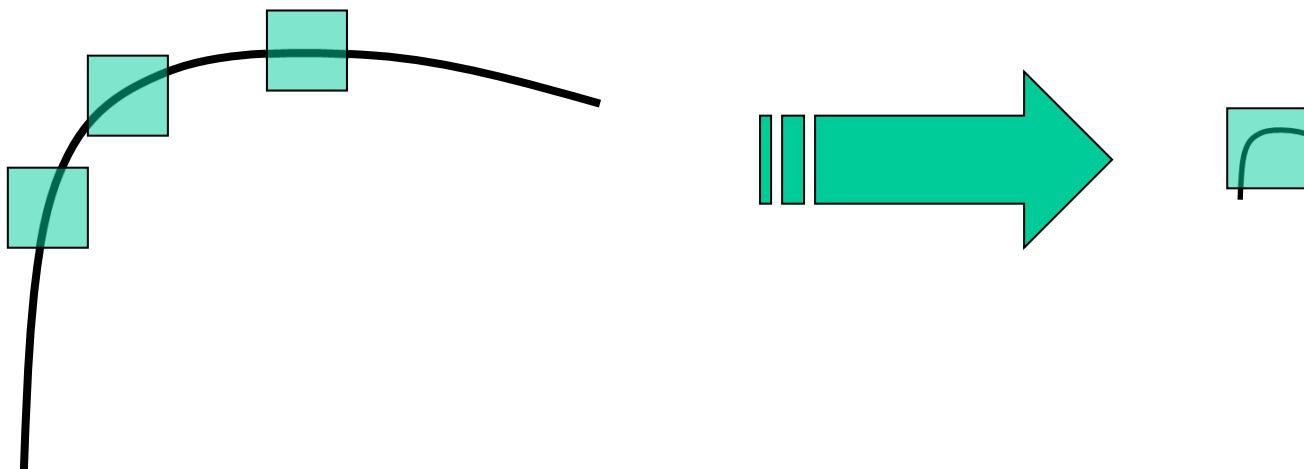
Ellipse rotates but its shape (i.e. eigenvalues)
remains the same

Corner response R is invariant to image rotation



Harris Detector: Some Properties

Not invariant to *image scale*!



All points will be
classified as **edges**

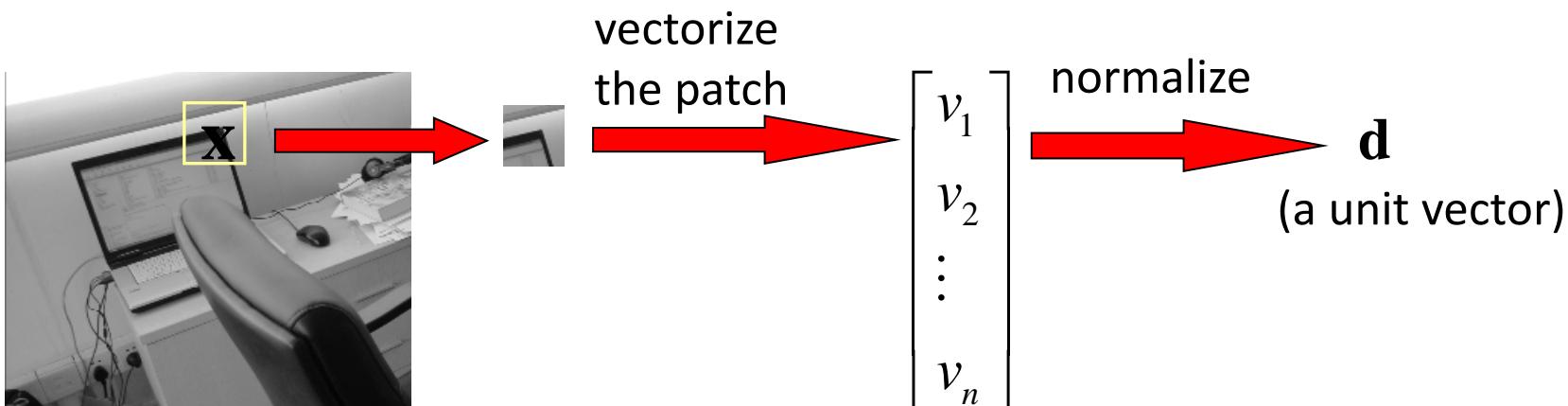
Corner !

The underlying reason is that Harris corner detection scheme does not provide an automatic and appropriate window size selection method!



“block” descriptor (usually used for Harris corner)

- “Block” descriptor for a Harris corner \mathbf{x}
 - Take a region with a fixed size around \mathbf{x}
 - Stack the region into a vector and normalize it as \mathbf{d}
 - This vector \mathbf{d} serves as the descriptor for \mathbf{x}





Distance between two descriptors (not limited to block descriptors)

Given two descriptors $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{R}^n$, their distance can be computed in different ways

Sum of squared differences (SSD):

$$SSD_{dist}(\mathbf{d}_1, \mathbf{d}_2) = \|\mathbf{d}_1 - \mathbf{d}_2\|_2^2 = \sum_{i=1}^n (d_1^i - d_2^i)^2$$

Or sum of absolute differences (SAD):

$$SAD_{dist}(\mathbf{d}_1, \mathbf{d}_2) = \sum_{i=1}^n |d_1^i - d_2^i|$$

Or normalized cross correlation (NCC):

$$NCC_{dist}(\mathbf{d}_1, \mathbf{d}_2) = 1 - \frac{1}{n} \frac{(\mathbf{d}_1 - \mu(\mathbf{d}_1)) \cdot (\mathbf{d}_2 - \mu(\mathbf{d}_2))}{std(\mathbf{d}_1) std(\mathbf{d}_2)}$$

or

$$NCC_{dist}(\mathbf{d}_1, \mathbf{d}_2) = \arccos \left(\frac{1}{n} \frac{(\mathbf{d}_1 - \mu(\mathbf{d}_1)) \cdot (\mathbf{d}_2 - \mu(\mathbf{d}_2))}{std(\mathbf{d}_1) std(\mathbf{d}_2)} \right)$$

where $\mu(\mathbf{d}_1)$ returns the mean value of \mathbf{d}_1 and $std(\mathbf{d}_1)$ returns the standard deviation of \mathbf{d}_1



Key points matching based on their descriptors

Suppose I_1 and I_2 are two images

I_1 's key points and the associated descriptors are $\{\mathbf{x}_i\}_{i=1}^m$ and $\mathcal{P} = \{\mathbf{d}_i\}_{i=1}^m$

I_2 's key points and the associated descriptors are $\{\mathbf{y}_j\}_{j=1}^n$ and $\mathcal{Q} = \{\mathbf{e}_j\}_{j=1}^n$

If and only if \mathbf{x}_i 's and \mathbf{y}_j 's descriptors \mathbf{d}_i and \mathbf{e}_j satisfy the following conditions, we say the key points \mathbf{x}_i and \mathbf{y}_j match and form a correspondence pair,

- 1) $dist(\mathbf{d}_i, \mathbf{e}_j) < t_1$, where t_1 is a predefined threshold
- 2) \mathbf{d}_i and \mathbf{e}_j satisfy the “two-direction” confirmation criteria, i.e.,

$$\forall \mathbf{v} \in \mathcal{P}, \mathbf{v} \neq \mathbf{d}_i, dist(\mathbf{v}, \mathbf{e}_j) > dist(\mathbf{d}_i, \mathbf{e}_j)$$

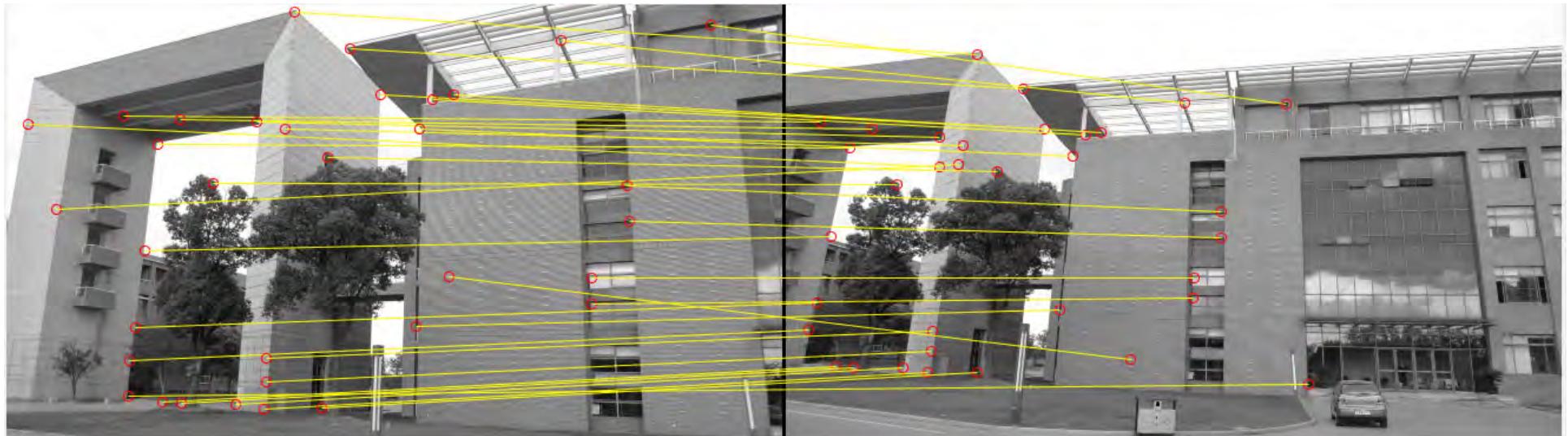
$$\forall \mathbf{v} \in \mathcal{Q}, \mathbf{v} \neq \mathbf{e}_j, dist(\mathbf{d}_i, \mathbf{v}) > dist(\mathbf{d}_i, \mathbf{e}_j)$$

- 3) \mathbf{d}_i and \mathbf{e}_j 's matching is unambiguous

Let $d_1 = dist(\mathbf{d}_i, \mathbf{e}_j)$. Suppose \mathbf{e}_k is the second best matching descriptor to \mathbf{d}_i and let $d_2 = dist(\mathbf{d}_i, \mathbf{e}_k)$. Then, $d_1 / d_2 < t_2$, where t_2 is another predefined threshold



Key points matching based on their descriptors



Matching Harris corners using block descriptors



Re-investigate the “block” descriptor

- “Block” descriptor for a Harris corner \mathbf{x}
 - Take a region with a fixed size around \mathbf{x}
 - Stack the region into a vector and normalize it as \mathbf{d}
 - This vector \mathbf{d} serves as the descriptor for \mathbf{x}
- Deficiencies of such simple descriptors
 - Not rotation invariant
 - Not scale invariant





Re-investigate the “block” descriptor

- “Block” descriptor for a Harris corner \mathbf{x}
 - Take a region with a fixed size around \mathbf{x}
 - Stack the region into a vector and normalize it as \mathbf{d}
 - This vector \mathbf{d} serves as the descriptor for \mathbf{x}
- We want:
 - Rotation and scale invariant feature points
 - Rotation and scale invariant feature descriptors



Content

- Local Invariant Features
 - Motivation
 - Requirements
 - Invariance
- Harris Corner Detector
- Scale Invariant Point Detection
 - Automatic scale selection
 - Laplacian-of-Gaussian detector
 - Difference-of-Gaussian detector



From Points to Regions

- The Harris corner detector defines interest points
 - Precise localization
 - High repeatability



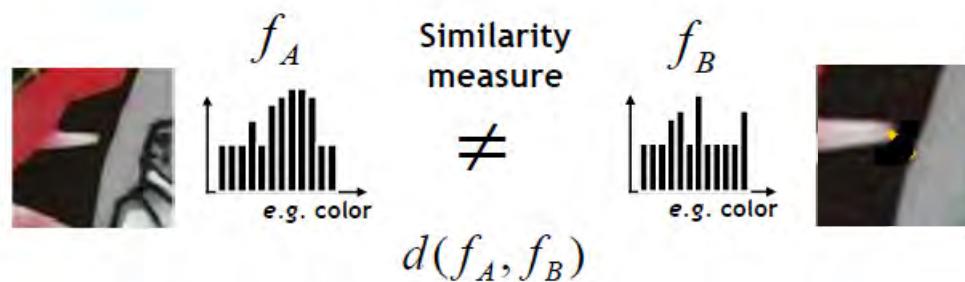
- In order to match those points, we need to compute a descriptor over a region
 - How can we define such a region in a scale invariant manner?
 - That is how can we detect scale invariant regions?



Scale Invariant Region Selection

Naïve Approach: Exhaustive Search

- Multi-scale procedure
 - Compare descriptors while varying the patch size



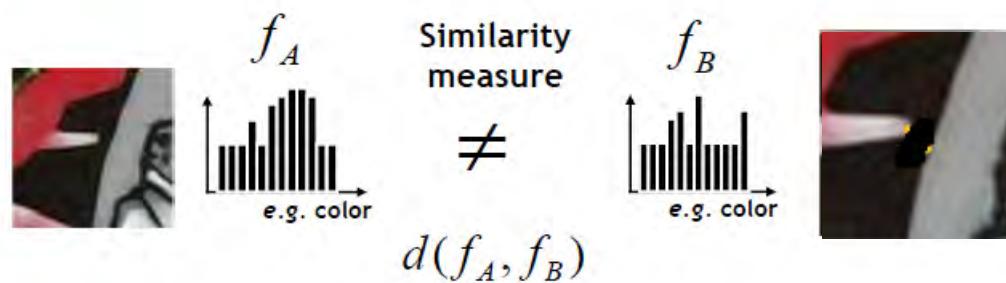
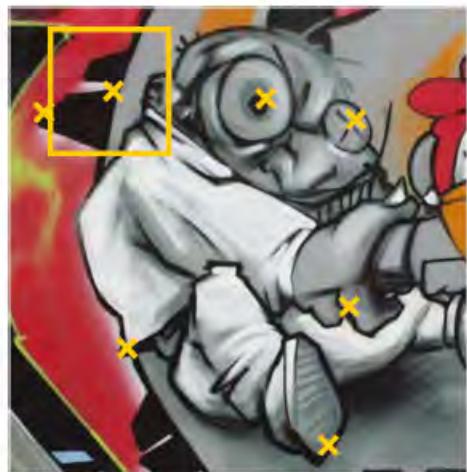
Slide credit: Krystian Mikolajczyk



Scale Invariant Region Selection

Naïve Approach: Exhaustive Search

- Multi-scale procedure
 - Compare descriptors while varying the patch size



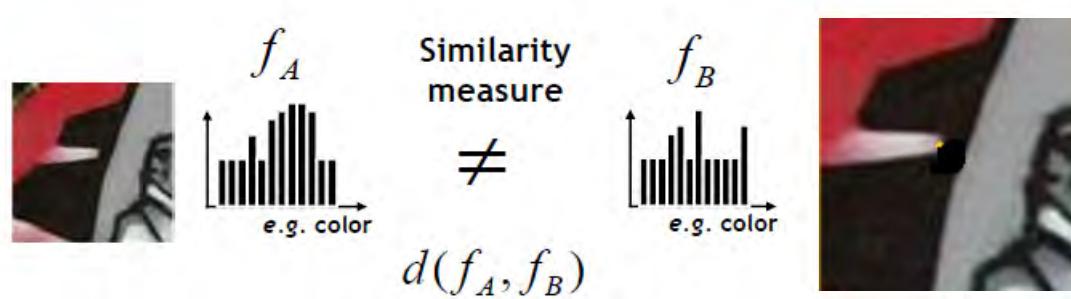
Slide credit: Krystian Mikolajczyk



Scale Invariant Region Selection

Naïve Approach: Exhaustive Search

- Multi-scale procedure
 - Compare descriptors while varying the patch size



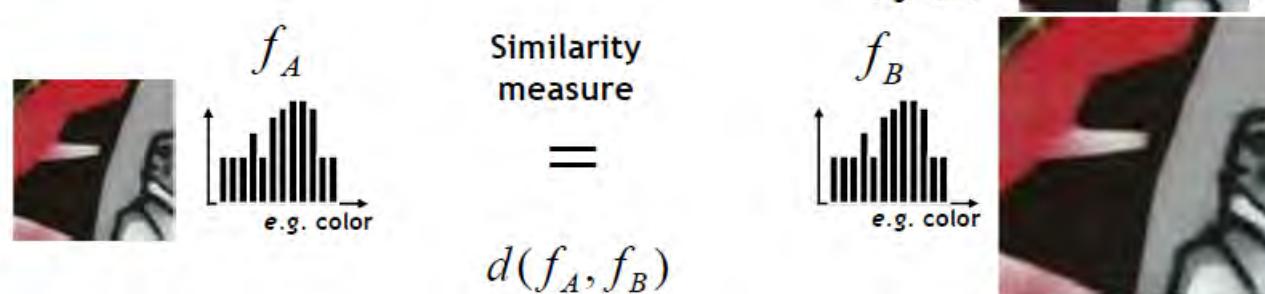
Slide credit: Krystian Mikolajczyk



Scale Invariant Region Selection

Naïve Approach: Exhaustive Search

- Comparing descriptors while varying the patch size
 - Computationally inefficient
 - Inefficient but possible for matching
 - Prohibitive for retrieval in large databases
 - Prohibitive for recognition



Slide credit: Krystian Mikolajczyk



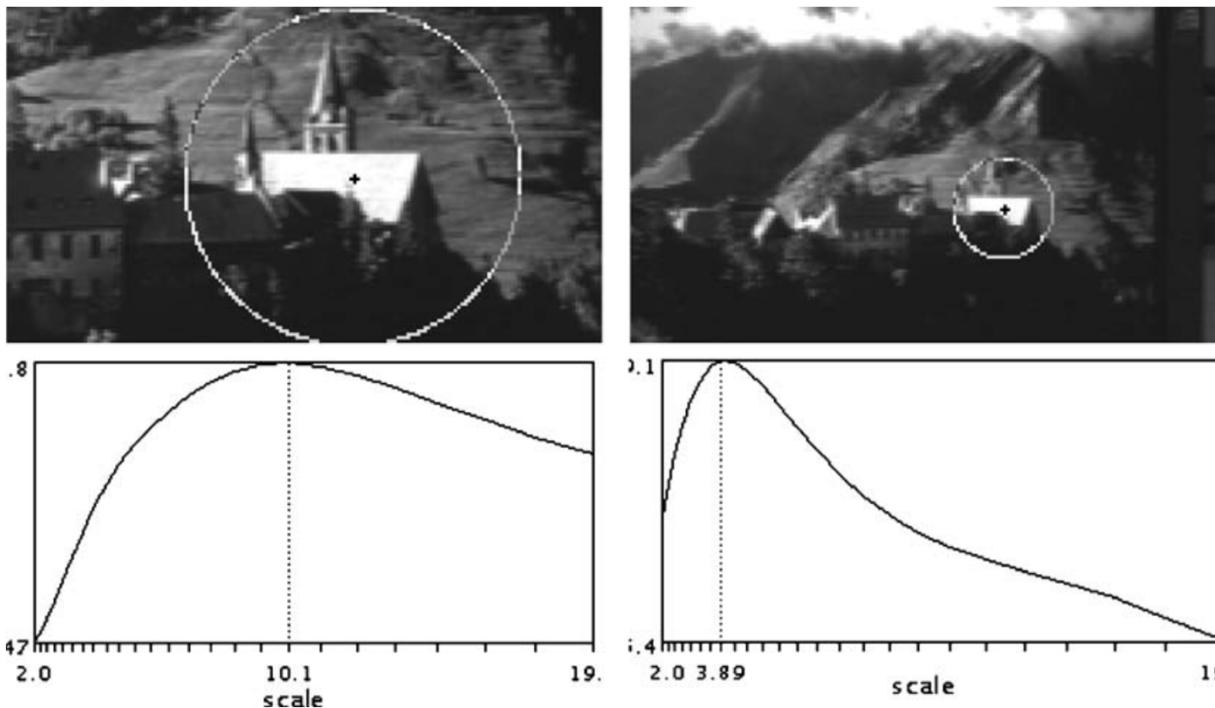
What do we want to do next?

- Naïve approach for scale invariant local description is not efficient (Detect Harris corners first, and then exhaustively searching for regions with appropriate sizes)
- Now we want to:
 - Find scale invariant points in the image (location)
 - At the same time, we want to know their **characteristic scales** (used to determine the neighborhood for local description)



Achieving scale covariance

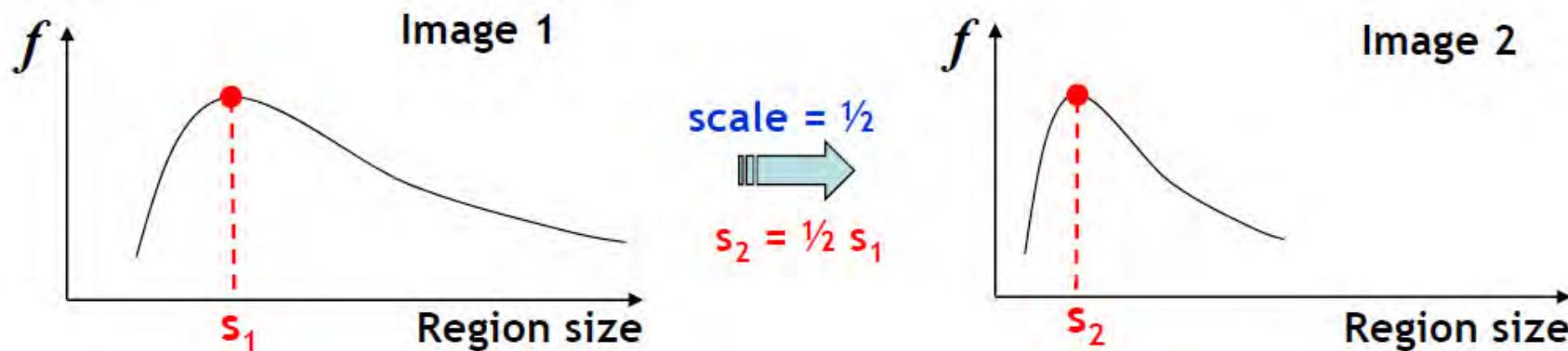
- Goal: independently detect corresponding regions in scaled versions of the same image
- Need *scale selection* mechanism for finding characteristic region size that is *covariant* with the image transformation





Automatic Scale Selection

- Common approach
 - Take a local extremum of this function
 - Observation: region size for which the extremum is achieved should be covariant to image scale; this scale covariant region size is found in each image independently

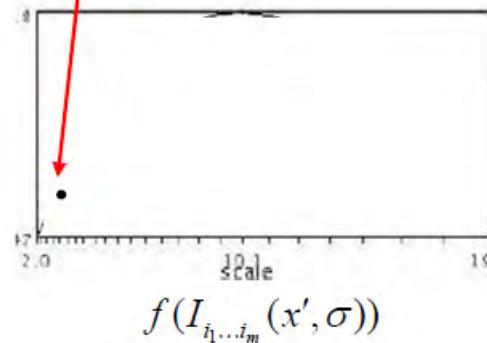
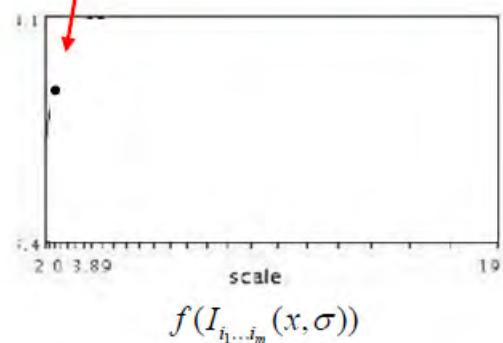


Slide credit: Kristen Grauman



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

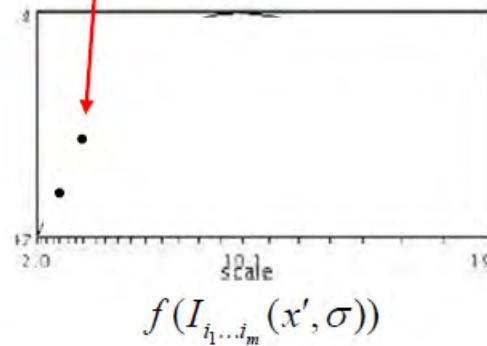
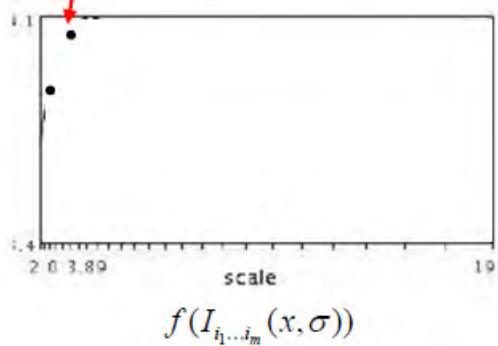


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

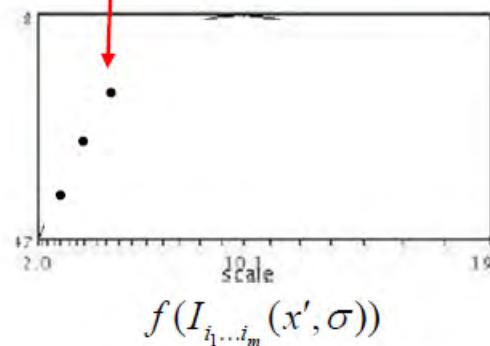
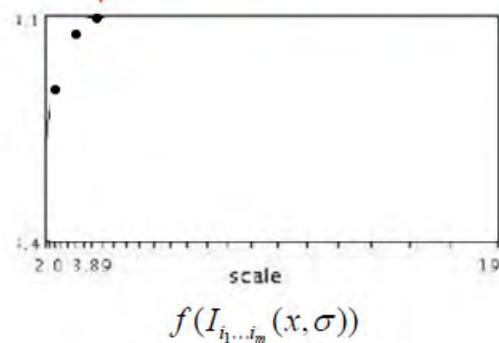


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

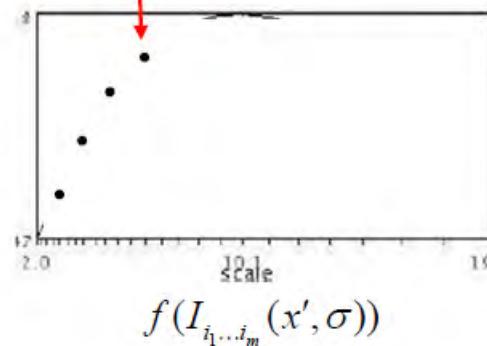
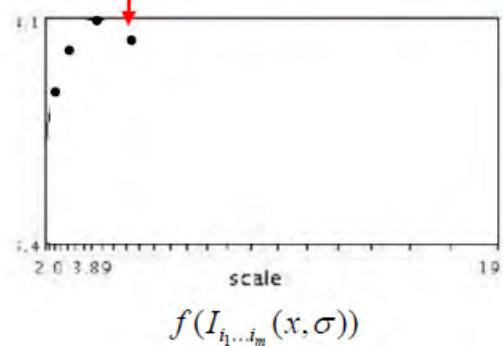


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

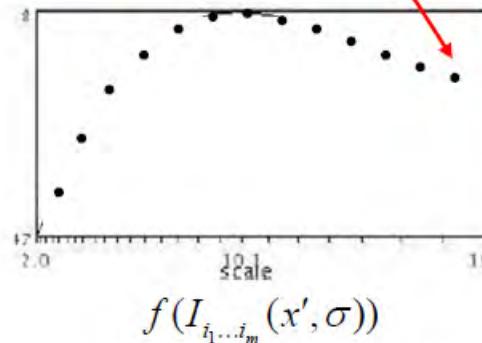
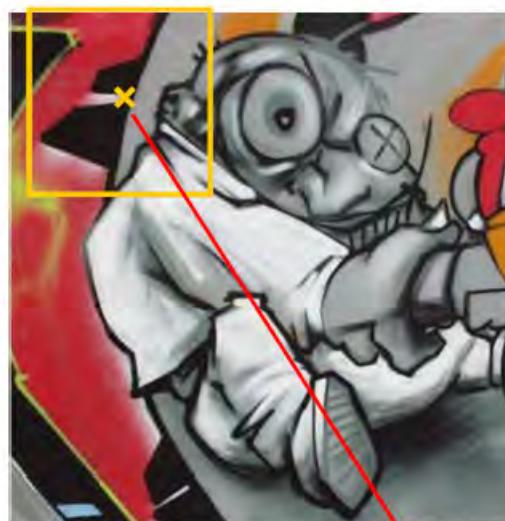
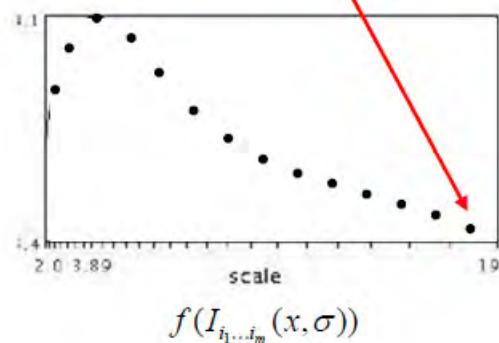
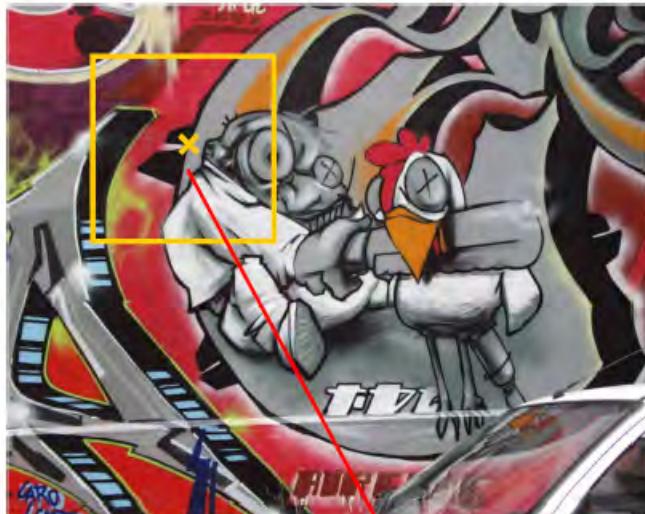


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

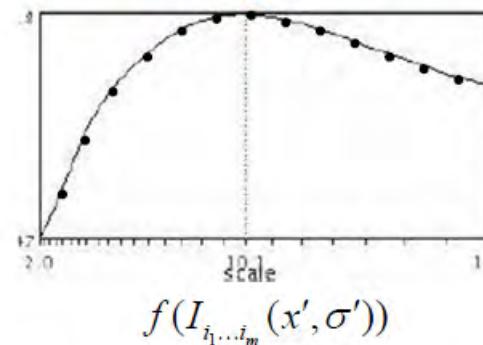
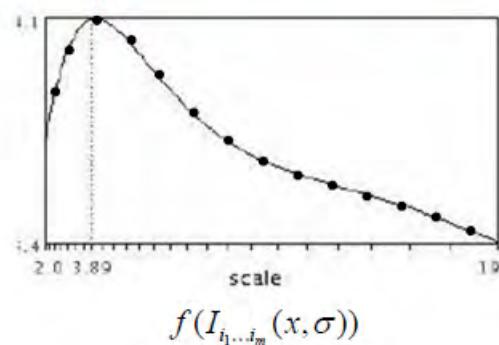
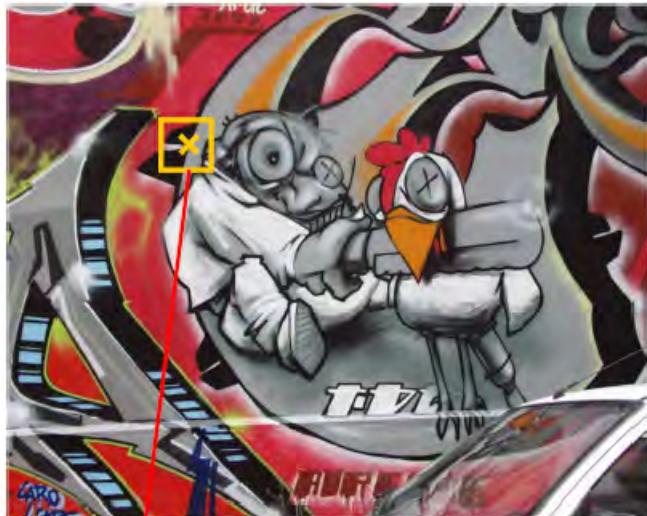


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Function responses for increasing scale (scale signature)

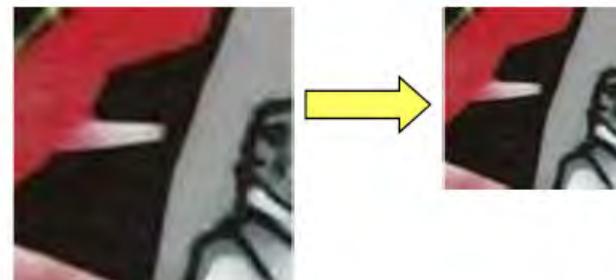
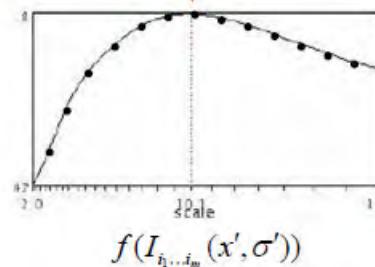
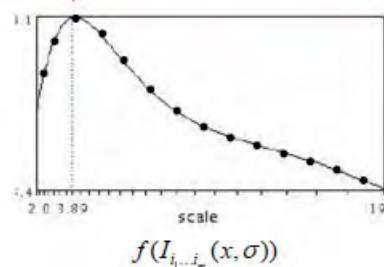
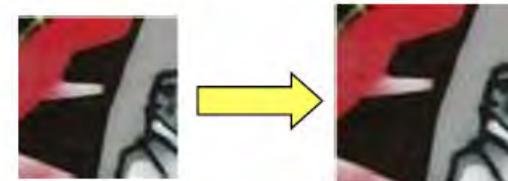
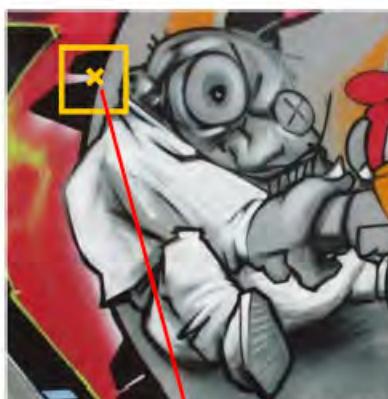


Slide credit: Krystian Mikolajczyk



Automatic Scale Selection

- Normalize: Rescale to fixed size

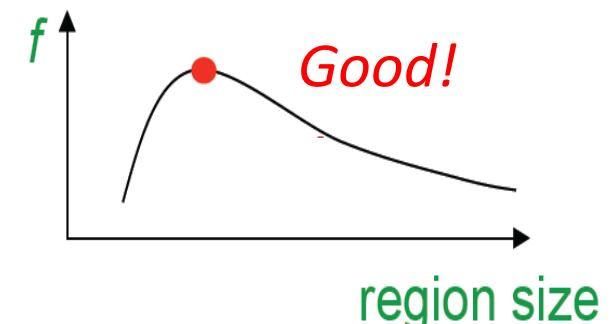
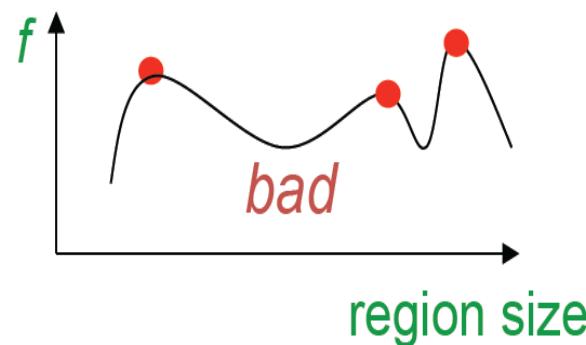
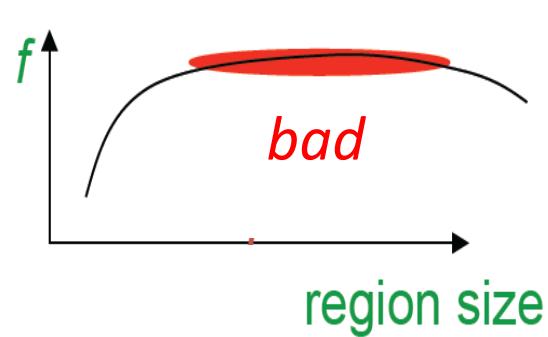


Slide credit: Tinne Tuytelaars



Automatic Scale Selection

- A good function for scale selection
 - It should have one stable sharp peak response to region size



The answer is scale-normalized Laplacian of Gaussian!



Scale-normalized LoG

2D isotropic Gaussian function,

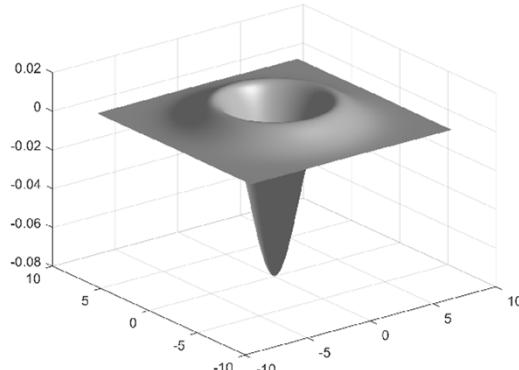
$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

LoG,

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Scale-normalized LoG,

$$\sigma^2 \nabla^2 g = \frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

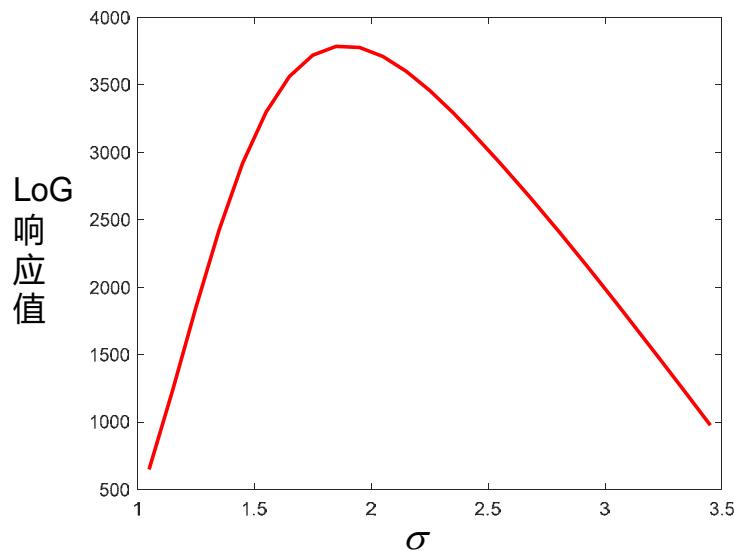


It can be used for blob structure detection



Blob detection using scale-normalized LoG

- For blob structures detection on image I
 - We need to know their centers and their spatial extensions (blob sizes)
 - For a blob point, its responses to scale-normalized LoGs with various scales has a unique peak (valley) , and thus we can use the “peak scale” to determine its size

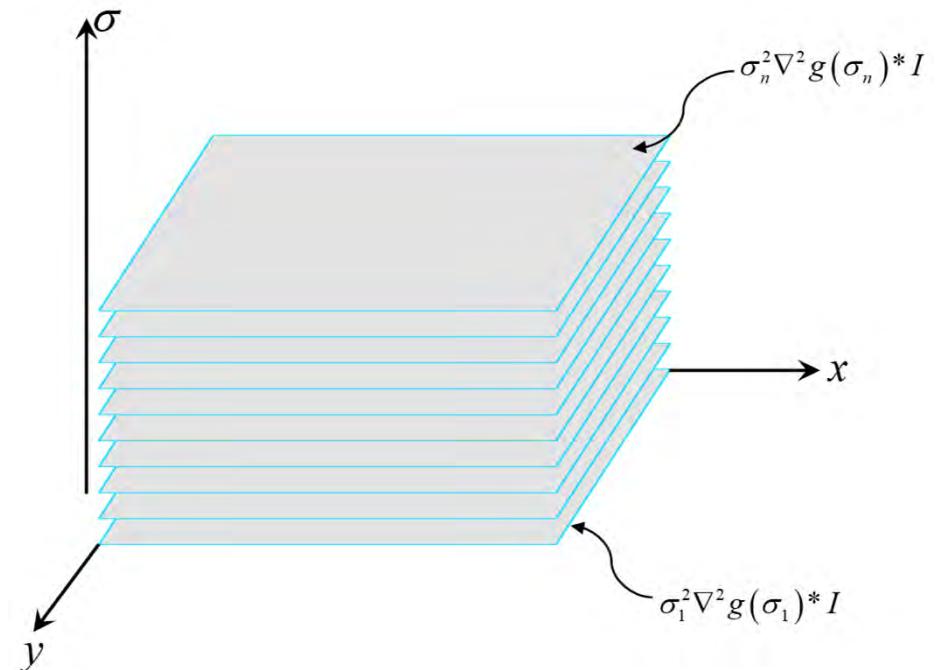


Responses of a blob point to a series of scale-normalized LoGs with various scales



Blob detection using scale-normalized LoG

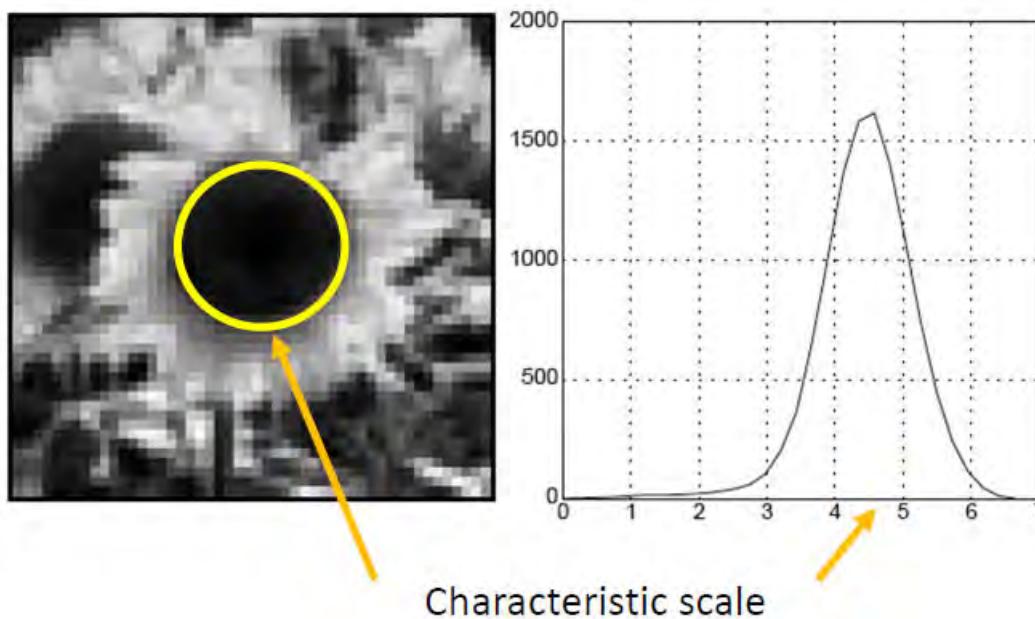
- For blob structures detection on image I
 - We need to know their centers and their spatial extensions (blob sizes)
 - For a blob point, its responses to scale-normalized LoGs with various scales has a unique peak (valley) , and thus we can use the “peak scale” to determine its size
 - Thus, we can use I ’s responses to a set of scale-normalized LoGs with various scales to find blobs





Characteristic scale

- We define the characteristic scale as the scale that produces peak (or valley) of scale-normalized Laplacian-of-Gaussian response



T. Lindeberg (1998). ["Feature detection with automatic scale selection."](#) *International Journal of Computer Vision* 30 (2): pp 77--116.

Slide credit: Svetlana Lazebnik

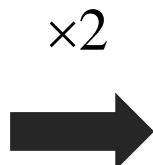


Characteristic scale

An example



x's characteristic scale obtained by scale-normalized LoGs is 1.5



y's characteristic scale obtained by scale-normalized LoGs is 3.0



scale-normalized LoG satisfy our requirement of automatic scale selection



Another Fact

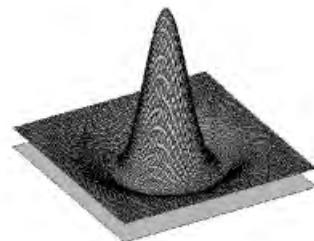
Spatial selection: the magnitude of the scale-normalized Laplacian-of-Gaussian response will achieve an extremum at the center of the blob, provided that its scale is “matched” to the scale of the blob



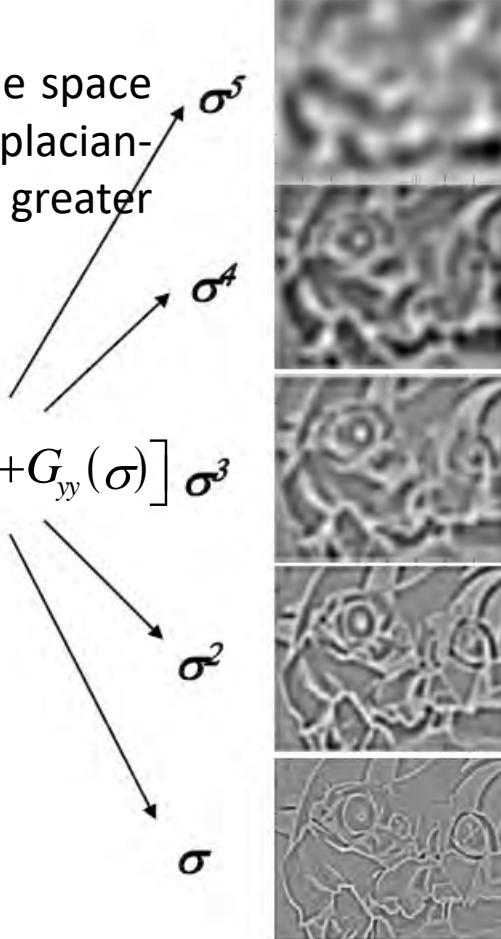
Scale-Invariant Point Detection

- Interest points:

- Local extremum in scale space
of scale-normalized Laplacian-of-Gaussian and be greater
than a threshold



$$\sigma^2 [G_{xx}(\sigma) + G_{yy}(\sigma)] \sigma^3$$



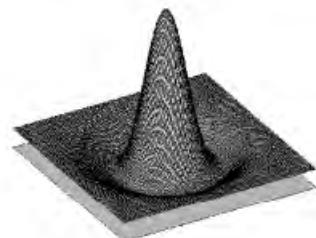
Slide adapted from Krystian Mikolajczyk



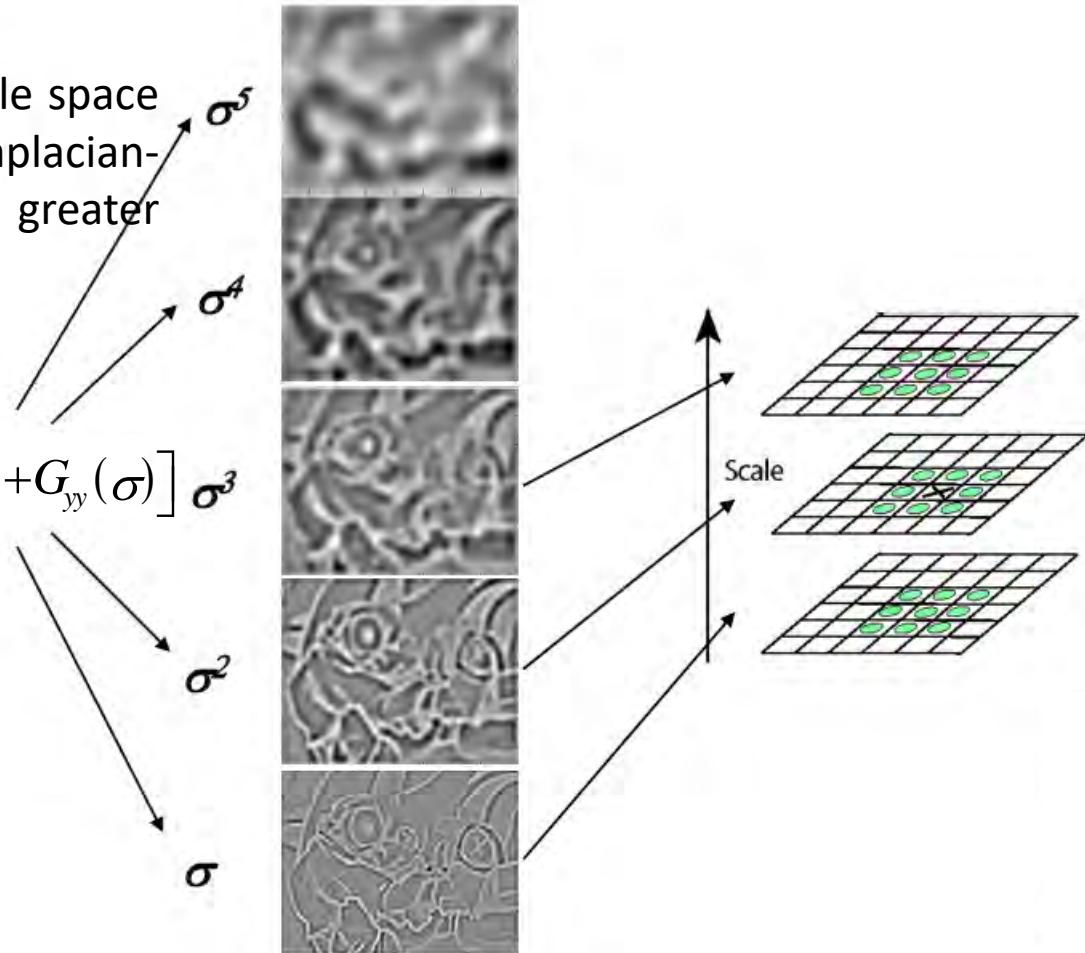
Scale-Invariant Point Detection

- Interest points:

- Local extremum in scale space
of scale-normalized Laplacian-of-Gaussian and be greater
than a threshold



$$\sigma^2 [G_{xx}(\sigma) + G_{yy}(\sigma)] \sigma^3$$

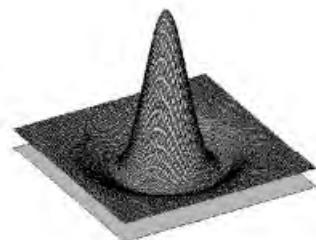




Scale-Invariant Point Detection

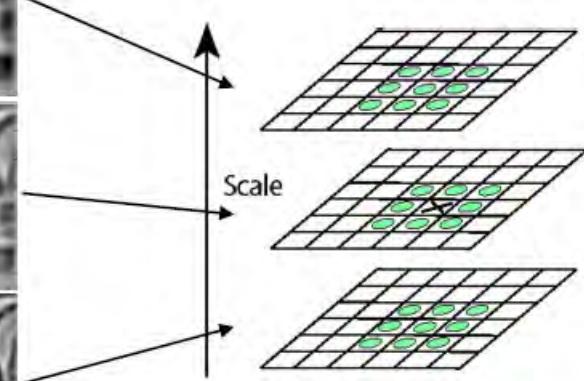
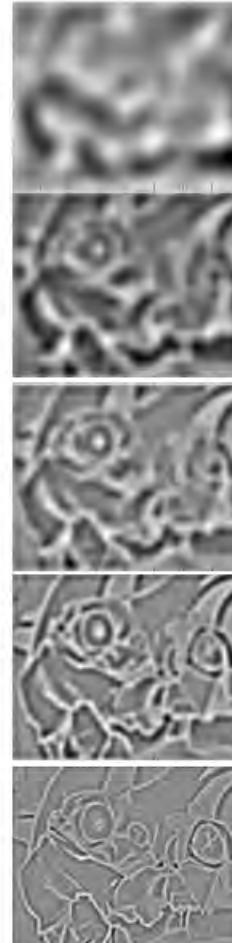
- **Interest points:**

- Local extremum in scale space
of scale-normalized Laplacian-of-Gaussian and be greater
than a threshold



$$\sigma^2 [G_{xx}(\sigma) + G_{yy}(\sigma)] \sigma^3$$

σ^5
 σ^4
 σ^3
 σ^2
 σ

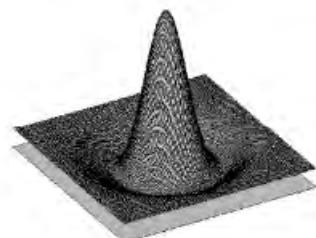




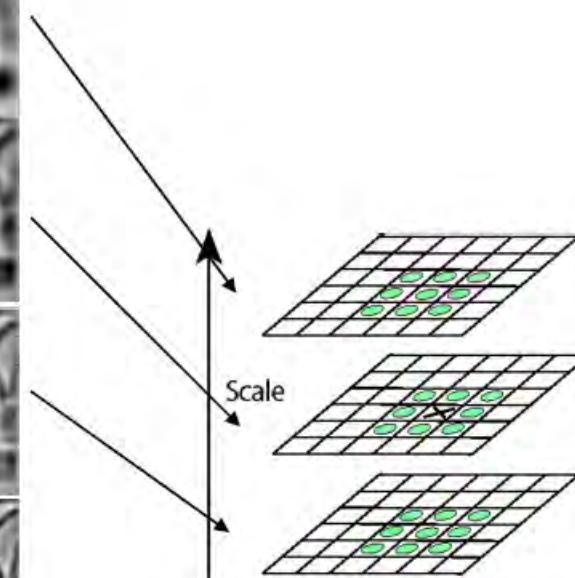
Scale-Invariant Point Detection

- Interest points:

- Local extremum in scale space
of scale-normalized Laplacian-of-Gaussian and be greater
than a threshold



$$\sigma^2 [G_{xx}(\sigma) + G_{yy}(\sigma)] \sigma^3$$



⇒ List of (x, y, σ)
(Positions of extrema in
the scale-spatial space)



We have got want we want!

Note: local extrema is obtained as key points by comparing the examined location with all the other 26 points around it in the scale-space

If the local extrema of scale-normalized LoG is achieved at p , two things of p can be determined: its spatial location and characteristic scale!



Scale-Invariant Point Detection: Example





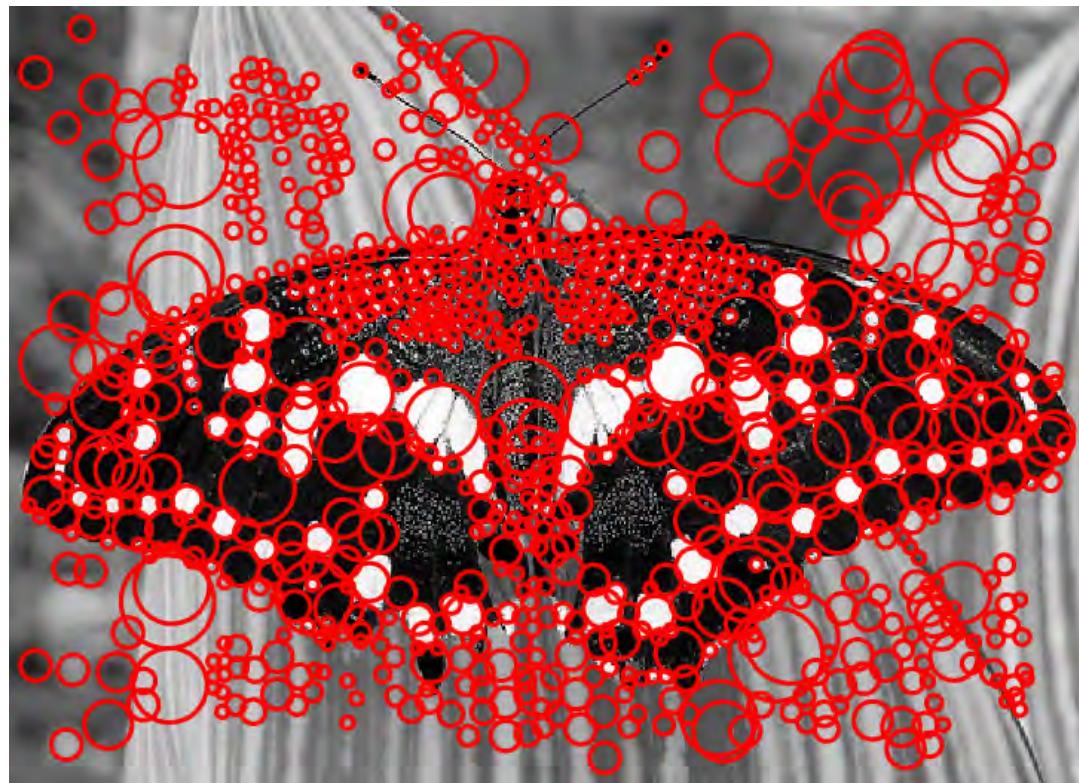
Scale-Invariant Point Detection: Example



$\sigma = 11.9912$



Scale-Invariant Point Detection: Example





Efficient implementation

Approximating the scale-normalized LoG with a difference of Gaussians:

$$L = \sigma^2 (G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma))$$

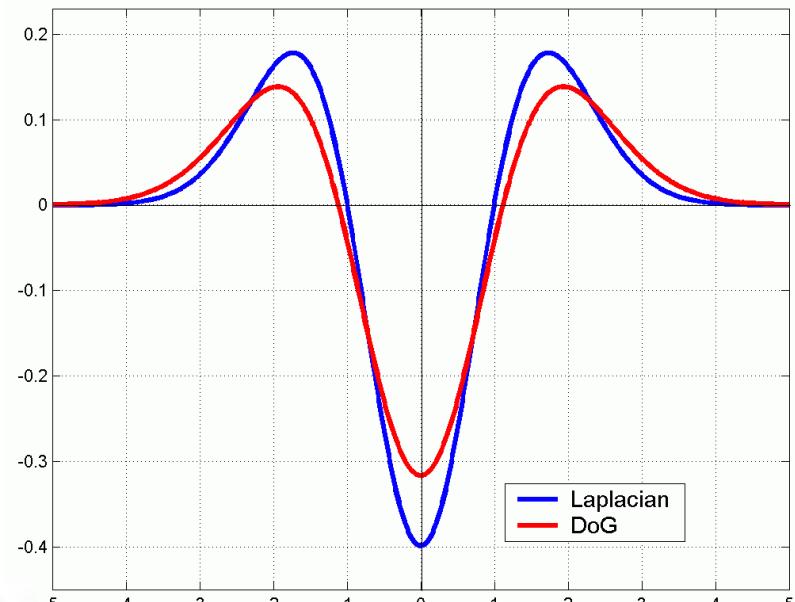
(scale-normalized LoG)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian is

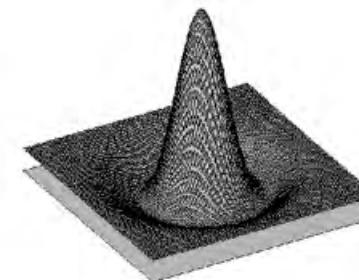
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$





DoG

- Difference of Gaussians as approximation of scale-normalized LoG
 - This is used e.g. in Lowe's SIFT pipeline for feature detection.
- Advantages
 - No need to compute 2nd derivatives
 - Gaussians are computed anyway, e.g. in a Gaussian pyramid.



Slide credit: Bastian Leibe

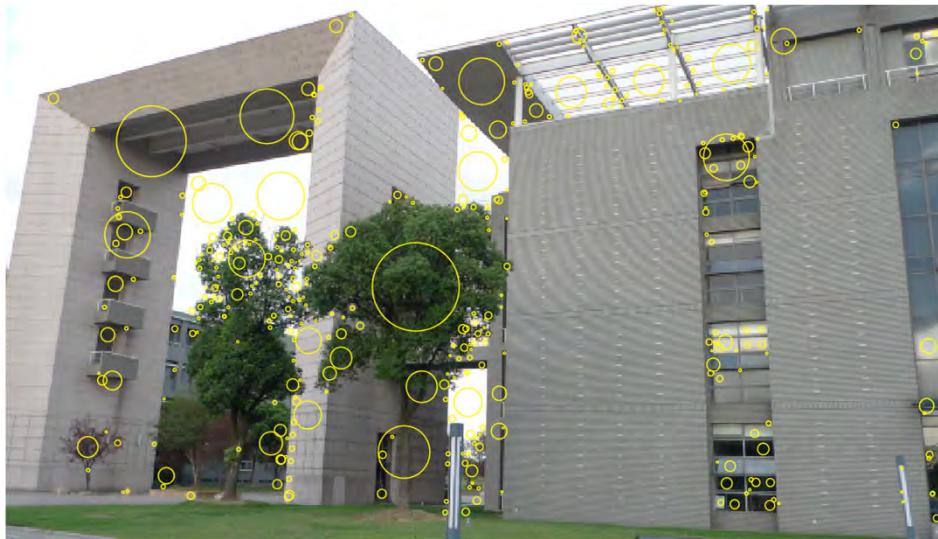


Scale-Invariant Point Detection

- **Given:** Two images of the same scene with a *large scale difference* between them.
- **Goal:** Find *the same* interest points *independently* in each image.
- **Solution:** Search for *maxima* of suitable functions in *scale* and in *space* (over the image).
- Two strategies
 - scale-normalized LoG
 - Difference-of-Gaussian (DoG) as a fast approximation
 - *These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).*



Examples





SSE, Tongji Univ.