CTT310: Digital Image Processing

Local Features Detection and Representation

Dr. Nguyen Ngoc Thao
Department of Computer Science, FIT, HCMUS

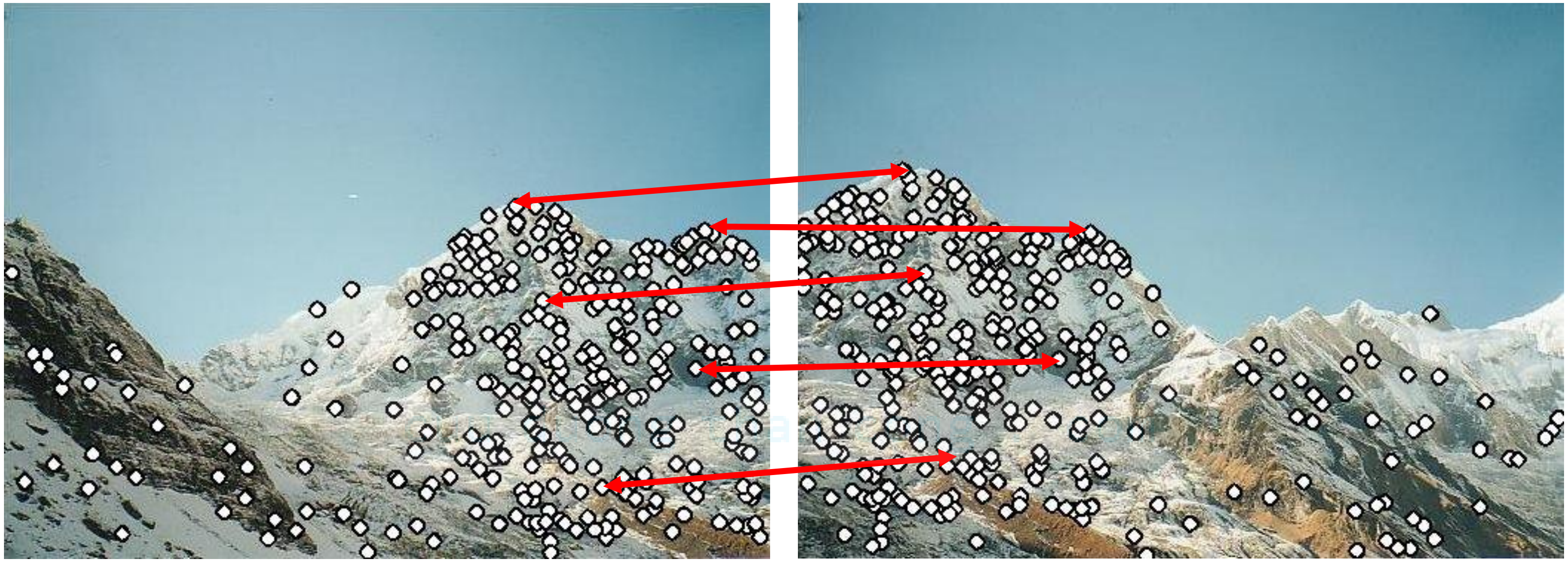
Outline

- The image matching problem
- Region detectors
  - The Harris-corner detector
  - Affine-covariant region detector
- Region descriptors
  - Types of local features for region representation
  - Scale-Invariant Feature Transform (SIFT)
  - Local Binary Patterns (LBP)
- Distance measures for feature matching

Section 8.1

THE IMAGE MATCHING PROBLEM

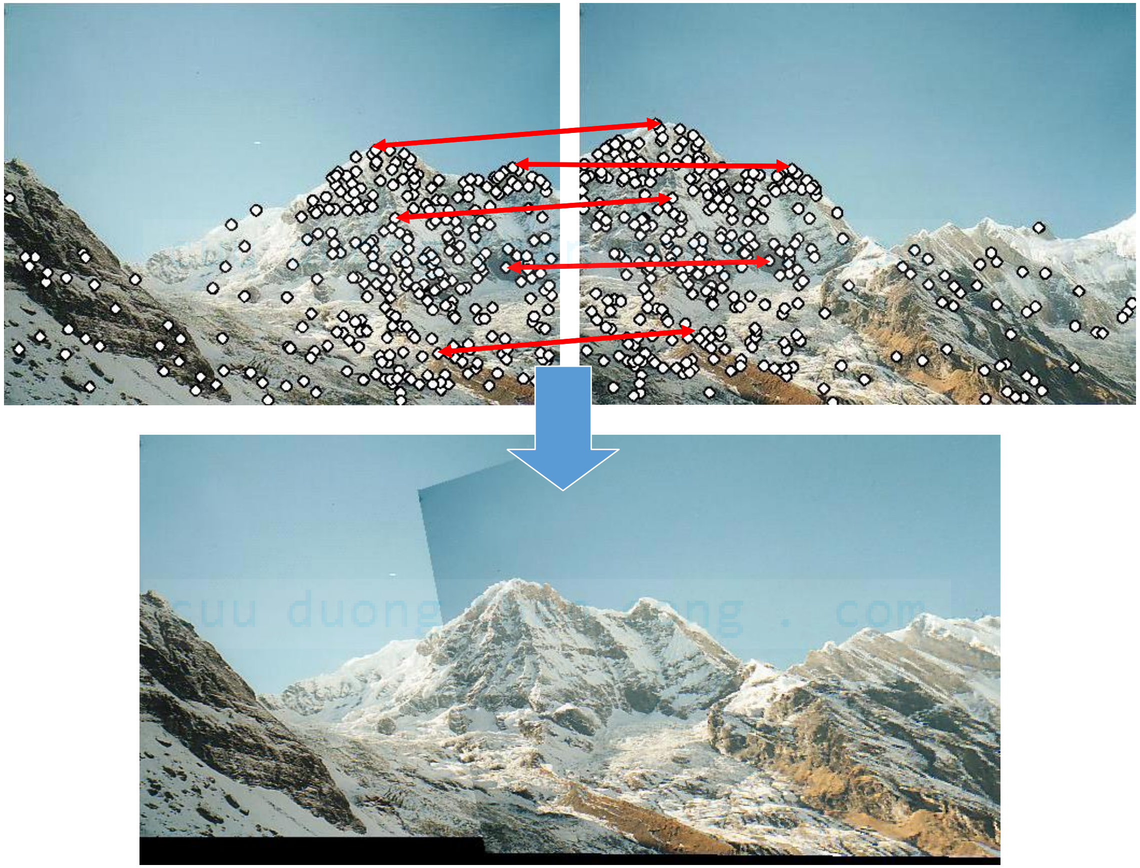
Feature matching



"What stuff in the left image matches with stuff on the right?"

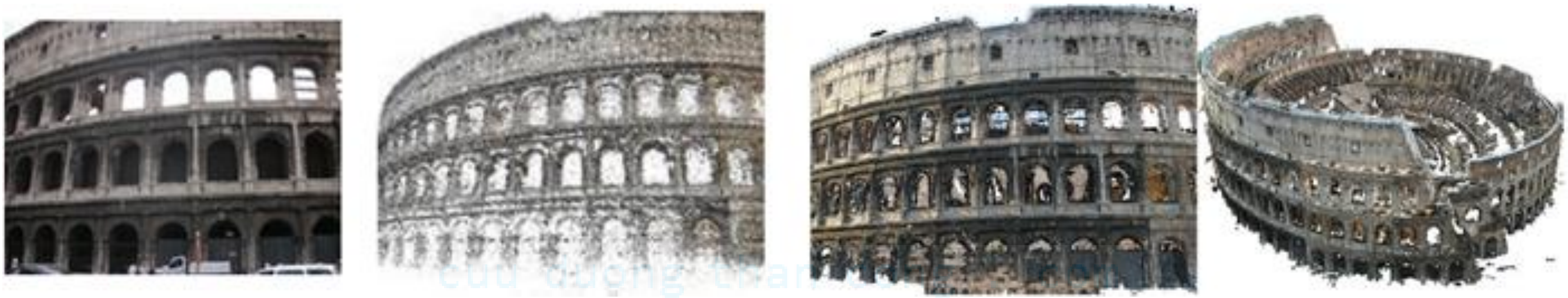
Feature matching: Applications

- Feature matching is an essential step for automatic panorama stitching



Feature matching: Applications

- It is also the very first step of high-level problems, such as
  - 3D reconstructed model



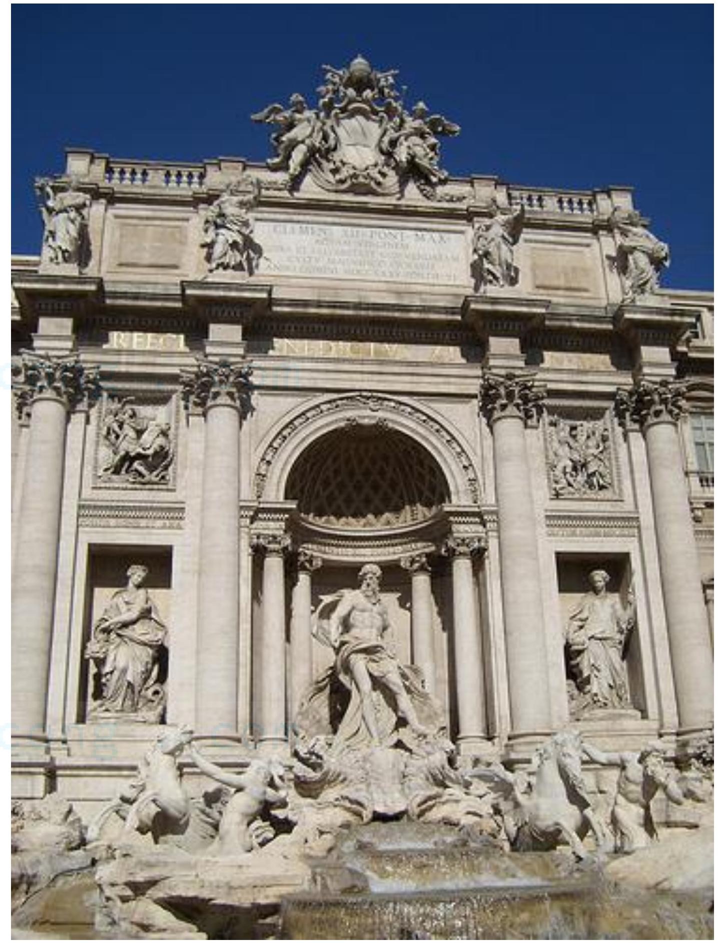
https://www.youtube.com/watch?v=HrgHFDPJHXo&index=3&list=PLDFDB5B8C80DB3AD6

  - Epipolar geometry estimation, texture classification, object recognition, and human action recognition

Feature matching

- This matching task seems to be easy…

by Diva Sian



by swashford

Feature matching

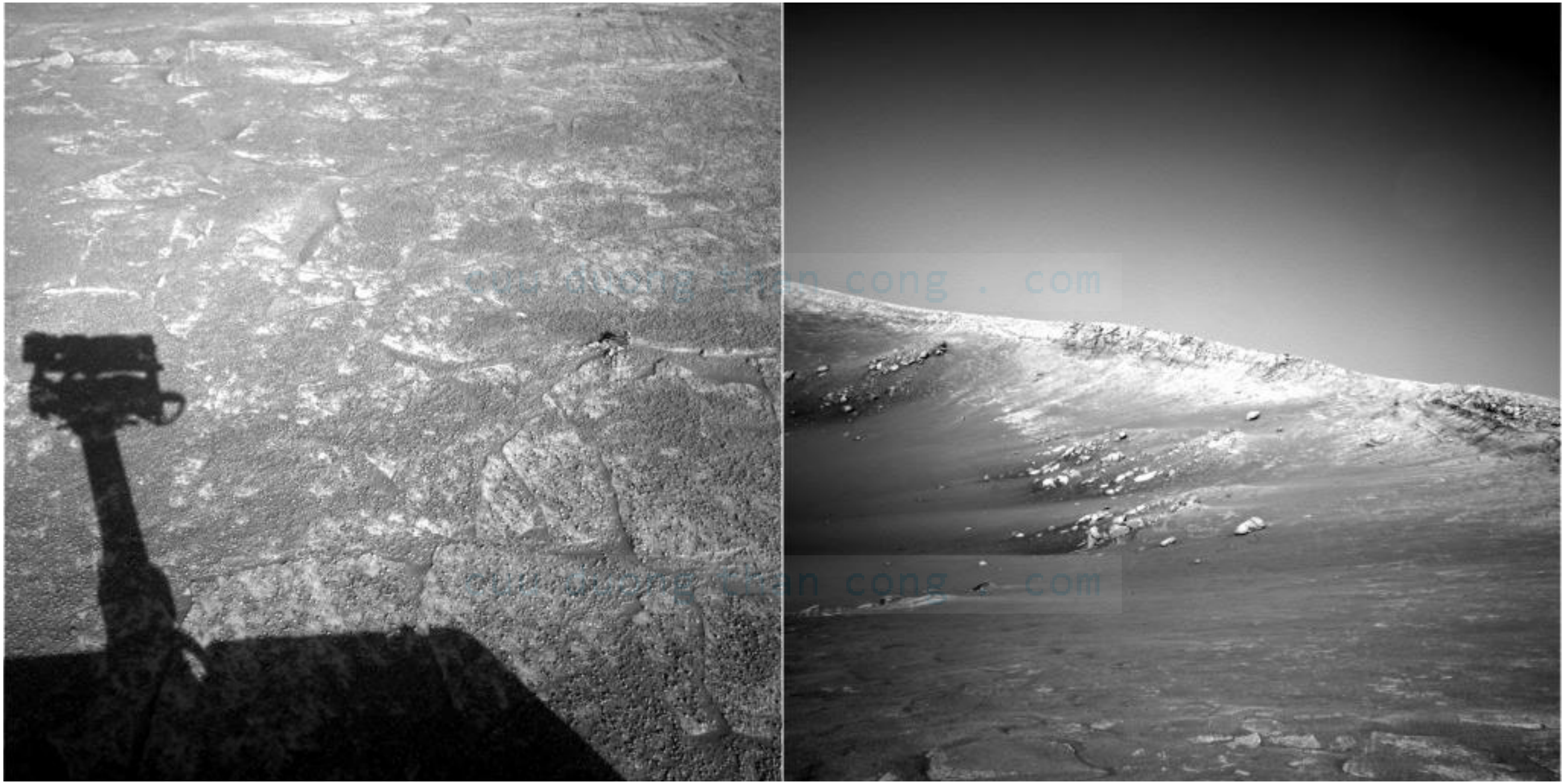
- It becomes harder because the two viewpoints are quite different from each other.

by Diva Sian

by scgbt

Feature matching

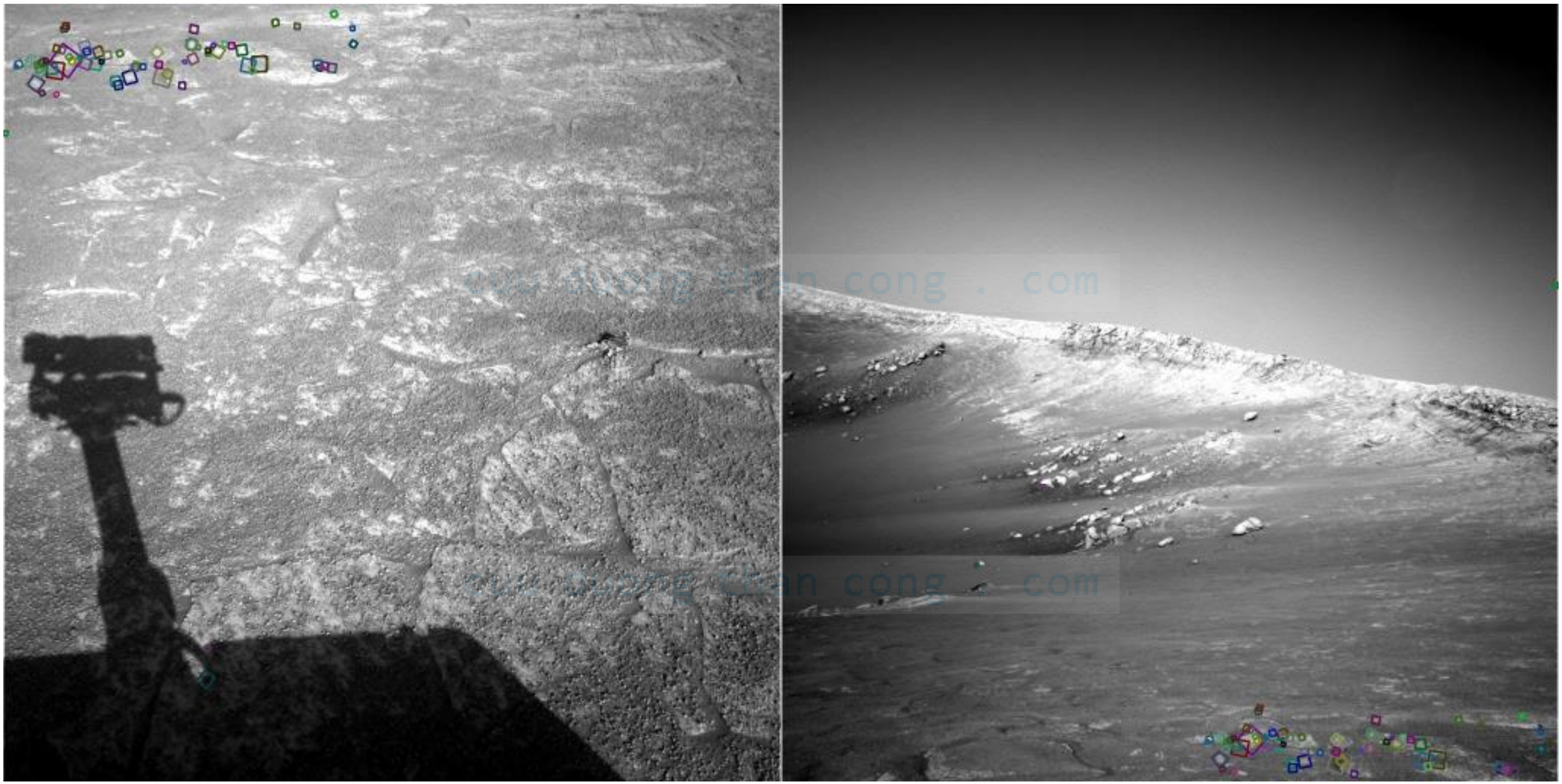
- Two images seem to have no common details.



NASA Mars Rover images

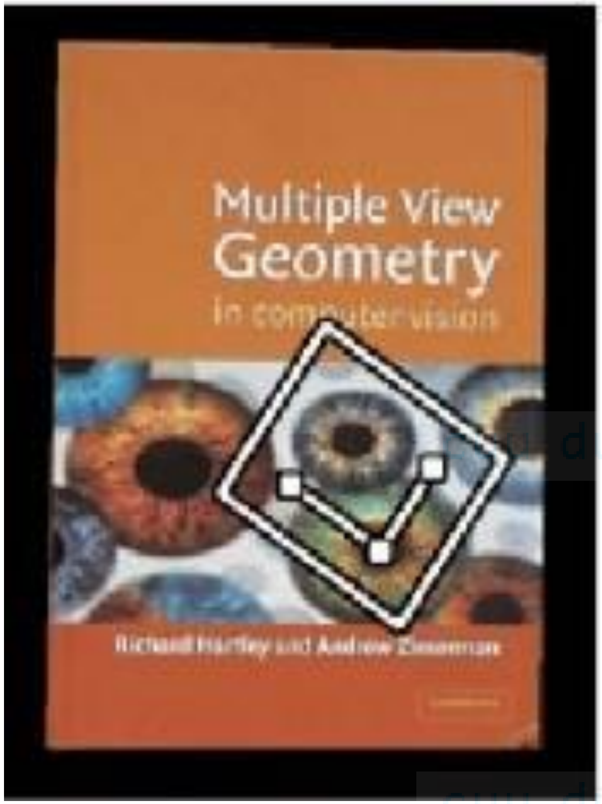
Feature matching

- They actually share some “very very small” details.



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

The image matching problem

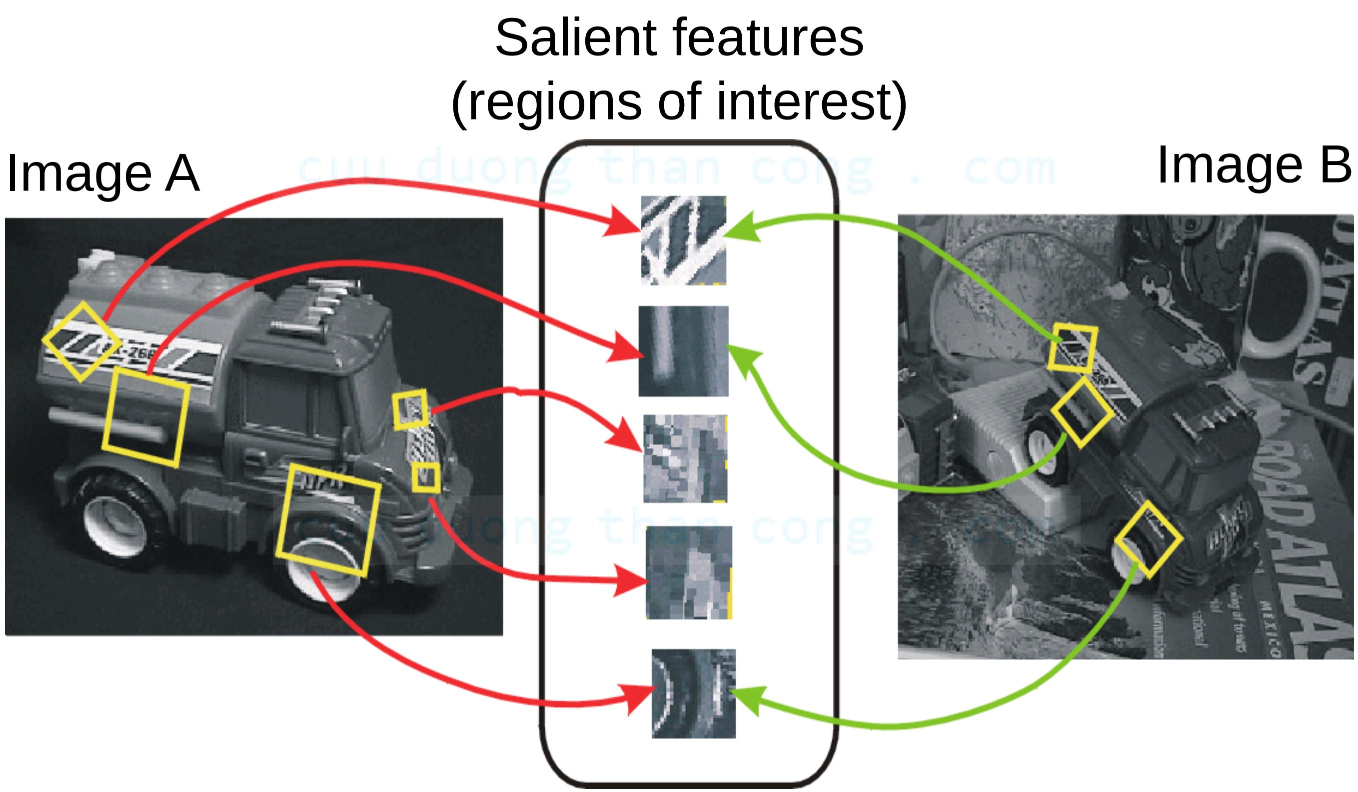


- At an interesting point, define a coordinate system
- Use the coordinate system to pull out a patch at that point

The image matching problem

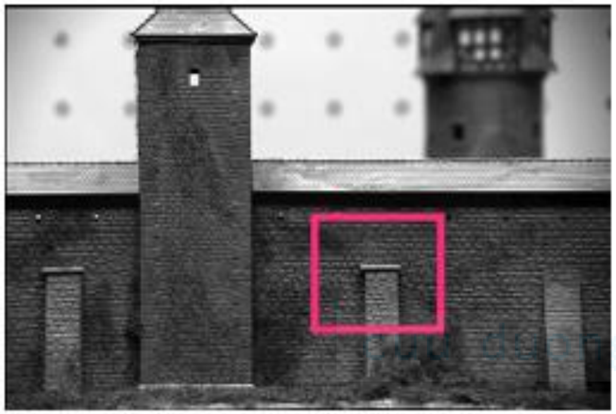
The image matching problem

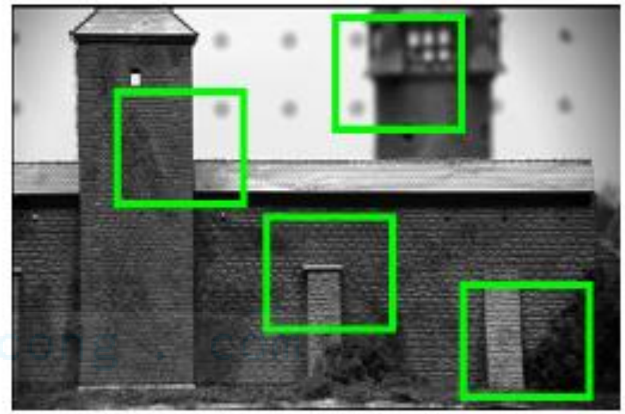
- Establish a set of correspondences from features in the first image to those in the second image



The image matching problem

- Elements to be matched are image patches of fixed size





- Task: find the best (most similar) patch in a second image

The image matching problem

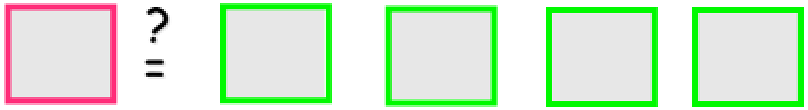
- Elements to be matched are image patches of fixed size

- Inituition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar)

Patch matching

- Elements to be matched are image patches of fixed size

- Inituition: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame)



Section 8.2

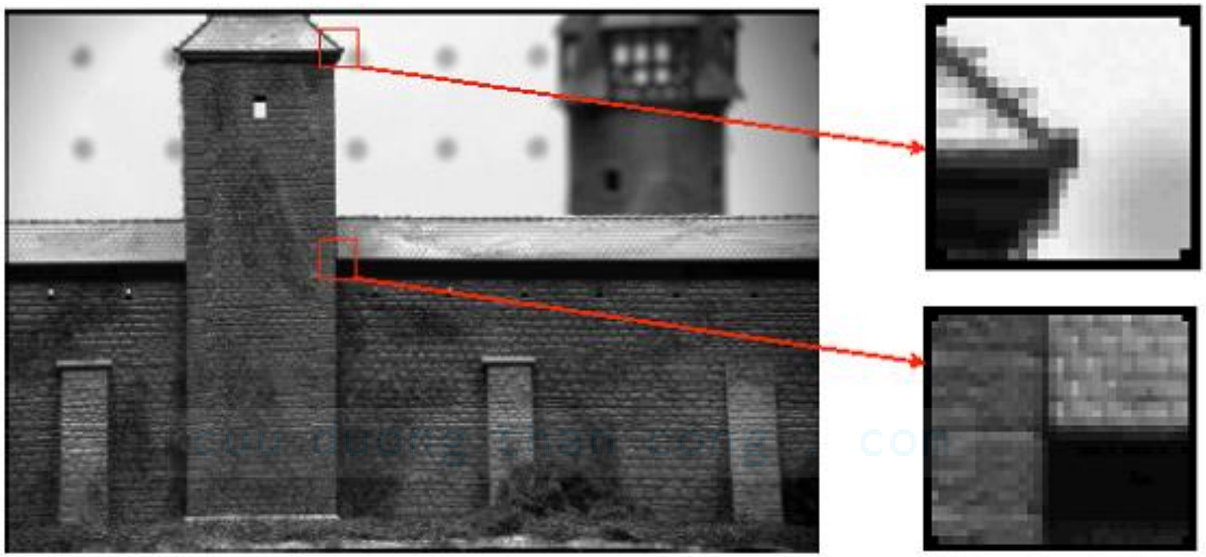
REGION DETECTORS

Invariant local features

- Algorithm for finding points and representing their patches should produce similar results even when conditions vary
- Buzzword is "invariance"
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure

- Look for image regions that are unusual
  - Lead to unambiguous matches in other images
- How to define "unusual"?

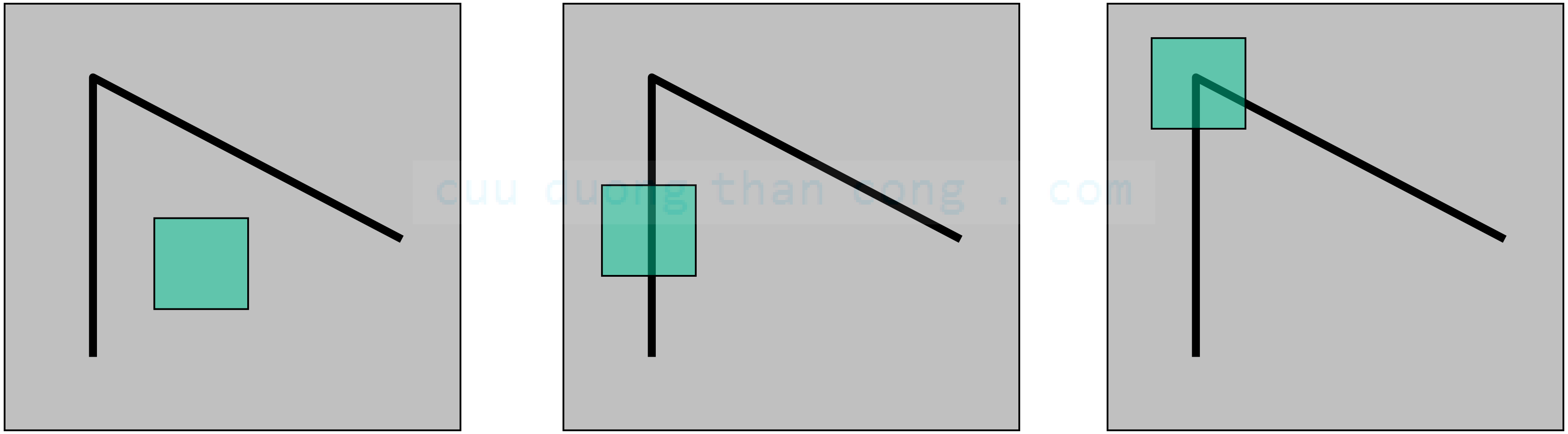
How about corners?



- Corners are junctions of contours
  - More stable features over changes of viewpoint
  - Large variations in the neighborhood of the point in all directions
- Corners are good features to match

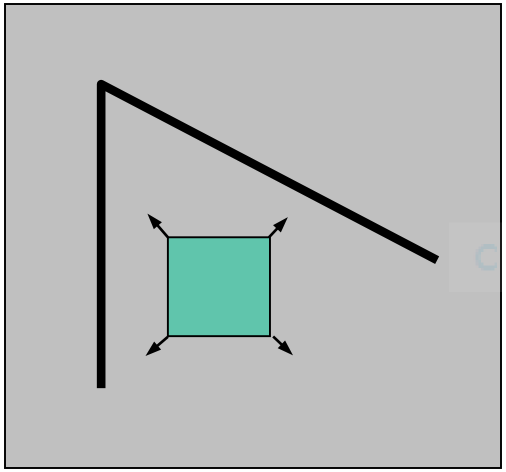
Local measures of uniqueness

- Suppose we only consider a small window of pixels
  - What defines whether a feature is a good or bad candidate?

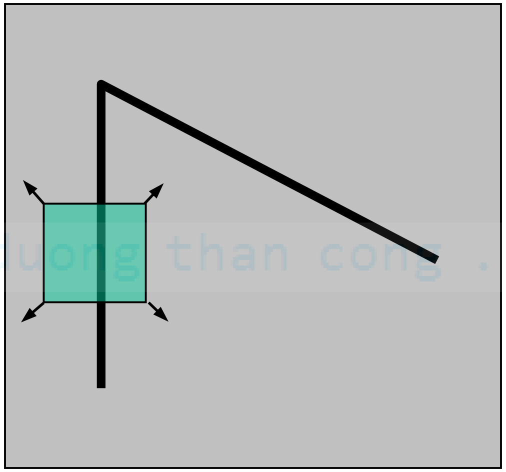


Local measures of uniqueness

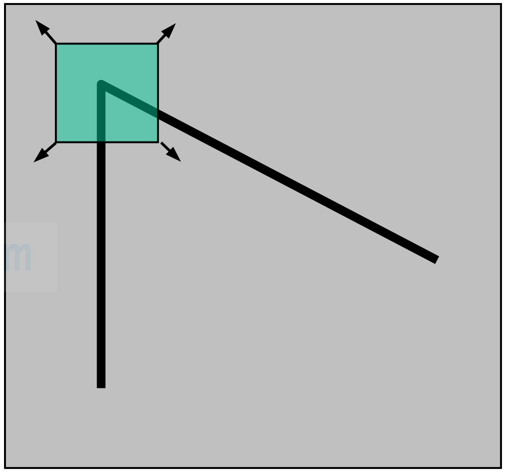
- How does the window change when you shift it?
- Shifting the window in any direction causes a big change



"flat" region:
no change in all directions



"edge":
no change along the edge direction

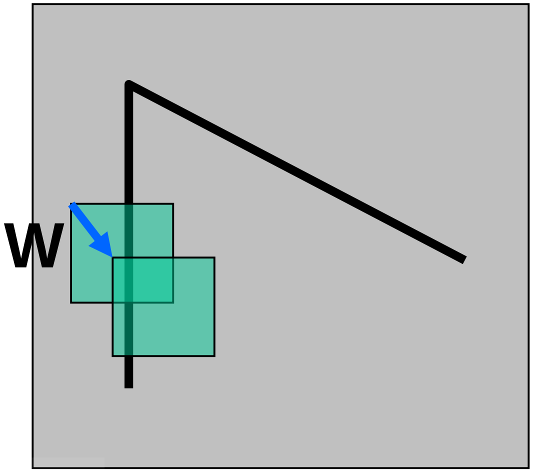


"corner":
significant change in all directions

Harris corner detector gives a mathematical approach for determining which case holds.

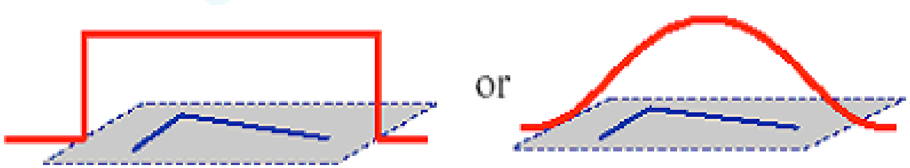
Harris Detector: Mathematical foundation

- Change of intensity for the shift $[u, v]$:



$$E(u,v) = \sum_{(x,y)\in W} \underbrace{w(x,y)}_{\text{Window function}} [\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}} - \underbrace{I(x,y)}_{\text{Intensity}}]^2$$

- Window function $w(x,y) =$



1 in window, 0 outside

Gaussian

Harris Detector: Mathematical foundation

- The Taylor Series for 2D Functions

$$f(x+u, y+v) = f(x,y) + uf_x(x,y) + vf_y(x,y) +$$

First partial derivatives

$$\frac{1}{2!}\left[u^2 f_{xx}(x,y) + uvf_{xy}(x,y) + v^2 f_{yy}(x,y)\right] +$$

Second partial derivatives

$$\frac{1}{3!}\left[u^3 f_{xxx}(x,y) + u^2 v f_{xxy}(x,y) + uv^2 f_{xyy}(x,y) + v^3 f_{yyy}(x,y)\right]$$

Third partial derivatives

$$+ \cdots (Higher\ order\ terms)$$

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

$$f(x+u, y+v) \approx f(x,y) + uf_x(x,y) + vf_y(x,y)$$

Harris Detector: Mathematical foundation

- Then, the derivation of Harris Corner is

$$\sum[I(x+u,y+v)-I(x,y)]^2$$

$$\approx \sum\left[I(x,y)+uI_x+vI_y-I(x,y)\right]^2 \quad \text{First order approx.}$$

$$=\sum\left(u^2{I_x}^2+2uvI_xI_y+v^2{I_y}^2\right)$$

$$=\sum[u \quad v]\begin{bmatrix} {I_x}^2 & I_xI_y \\ I_yI_x & {I_y}^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation}$$

$$=[u \quad v]\left(\sum\begin{bmatrix} {I_x}^2 & I_xI_y \\ I_yI_x & {I_y}^2 \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector: Mathematical foundation

- For small shifts (u, v), we have a bilinear approximation

$$R(u,v) \cong [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

  - where M is a 2×2 matrix computed from image derivatives

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_y I_x & {I_y}^2 \end{bmatrix}$$

Window function – computing a weighted sum (simplest case, $w = 1$)

These are just products of components of the gradient, I_x and I_y

Interpreting the second moment matrix

- First, consider an axis-aligned corner:

$$M = \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_y I_x & {I_y}^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

  - Dominant gradient directions align with x or y axis
  - If either λ is close to 0, then this is not a corner, so look for locations where both are large.

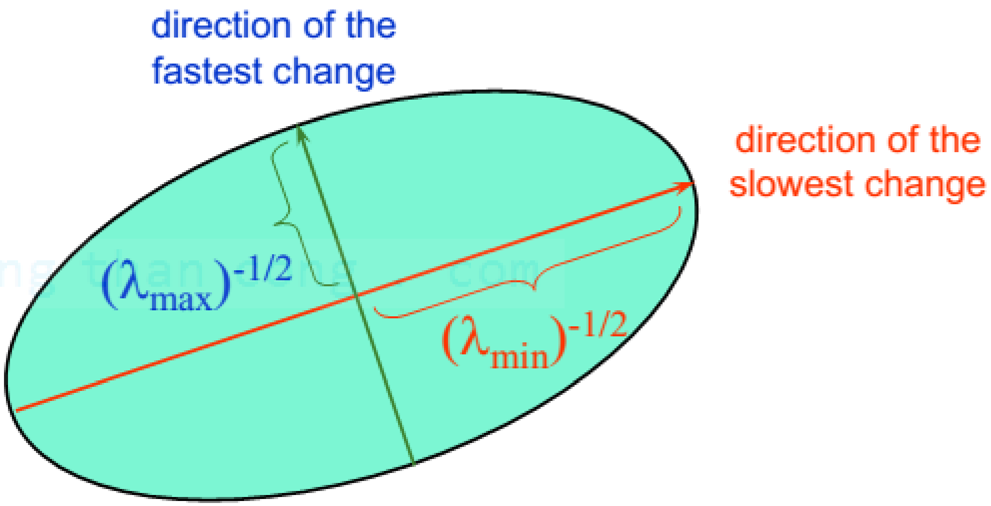
- Since M is symmetric, we have $M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$

Interpreting the second moment matrix

- M can be visualized as an ellipse with axis lengths determined by the eigenvalues and orientation by R

Ellipse $E(u, v) = const$

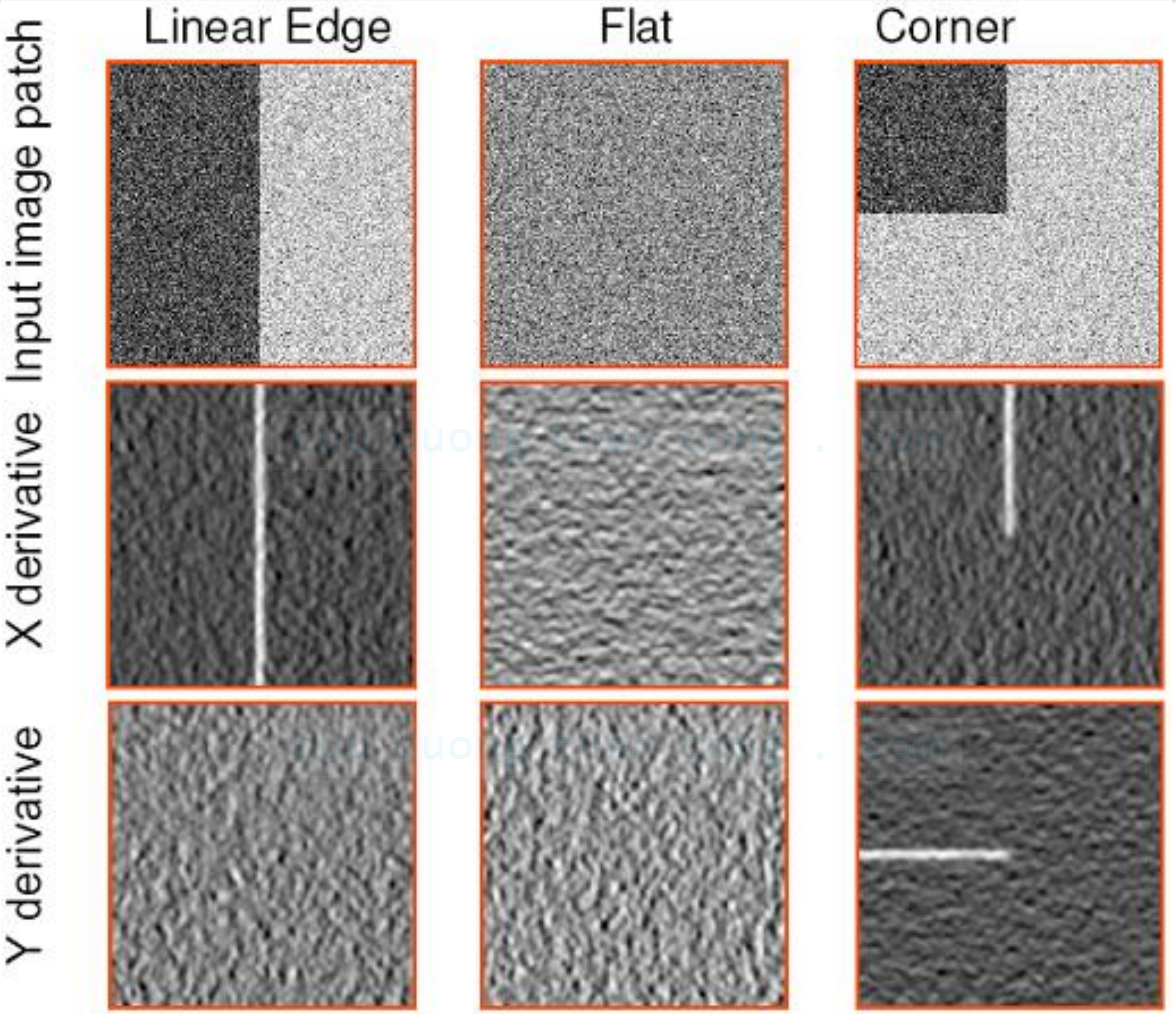
$$[u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix} = const$$



Intuitive way to understand Harris

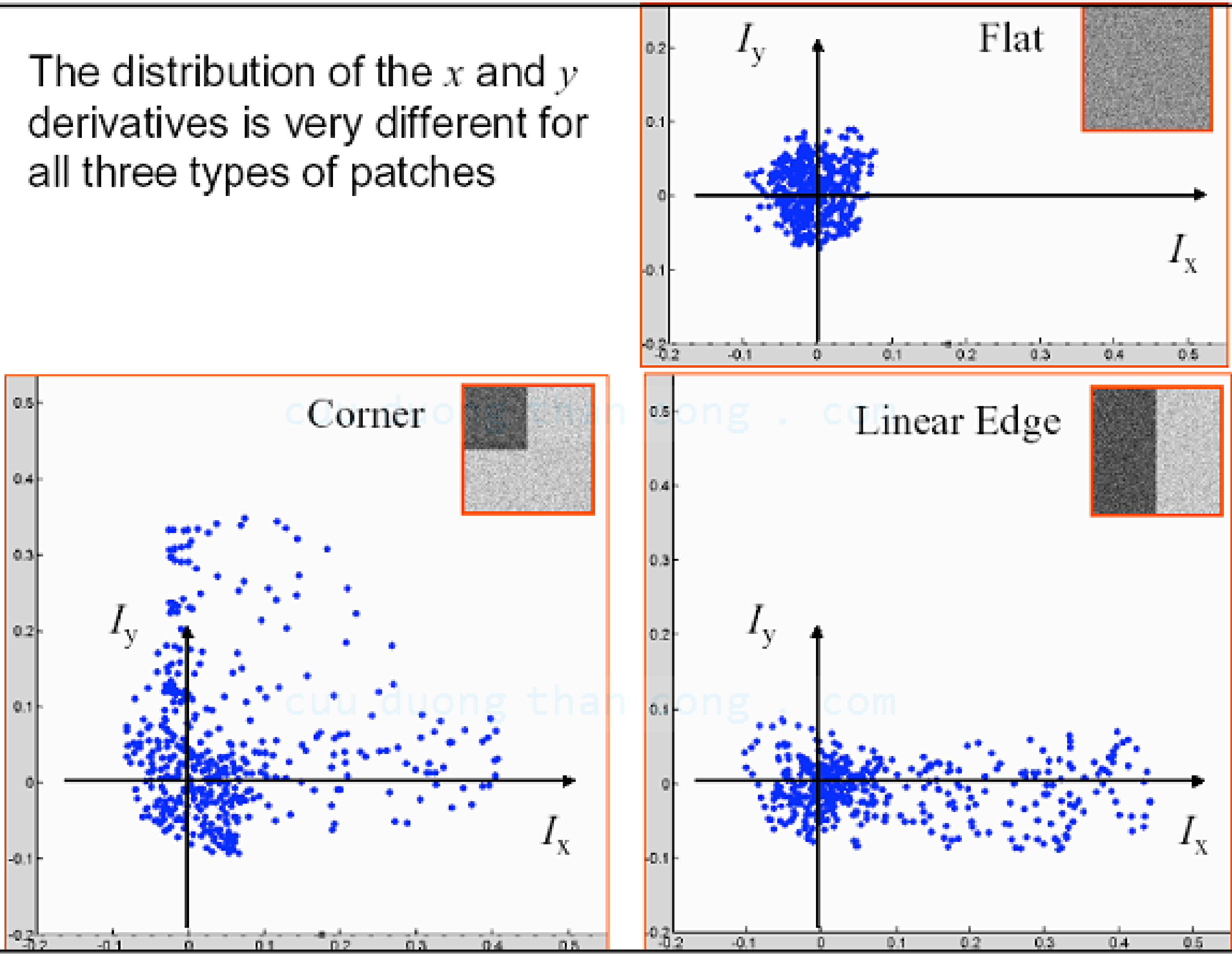
- Treat gradient vectors as a set of (dx, dy) points with a center of mass defined as being at $(0,0)$
- Fit an ellipse to that set of points via scatter matrix
- Analyze ellipse parameters for varying cases…

Example: Cases and 2D derivatives



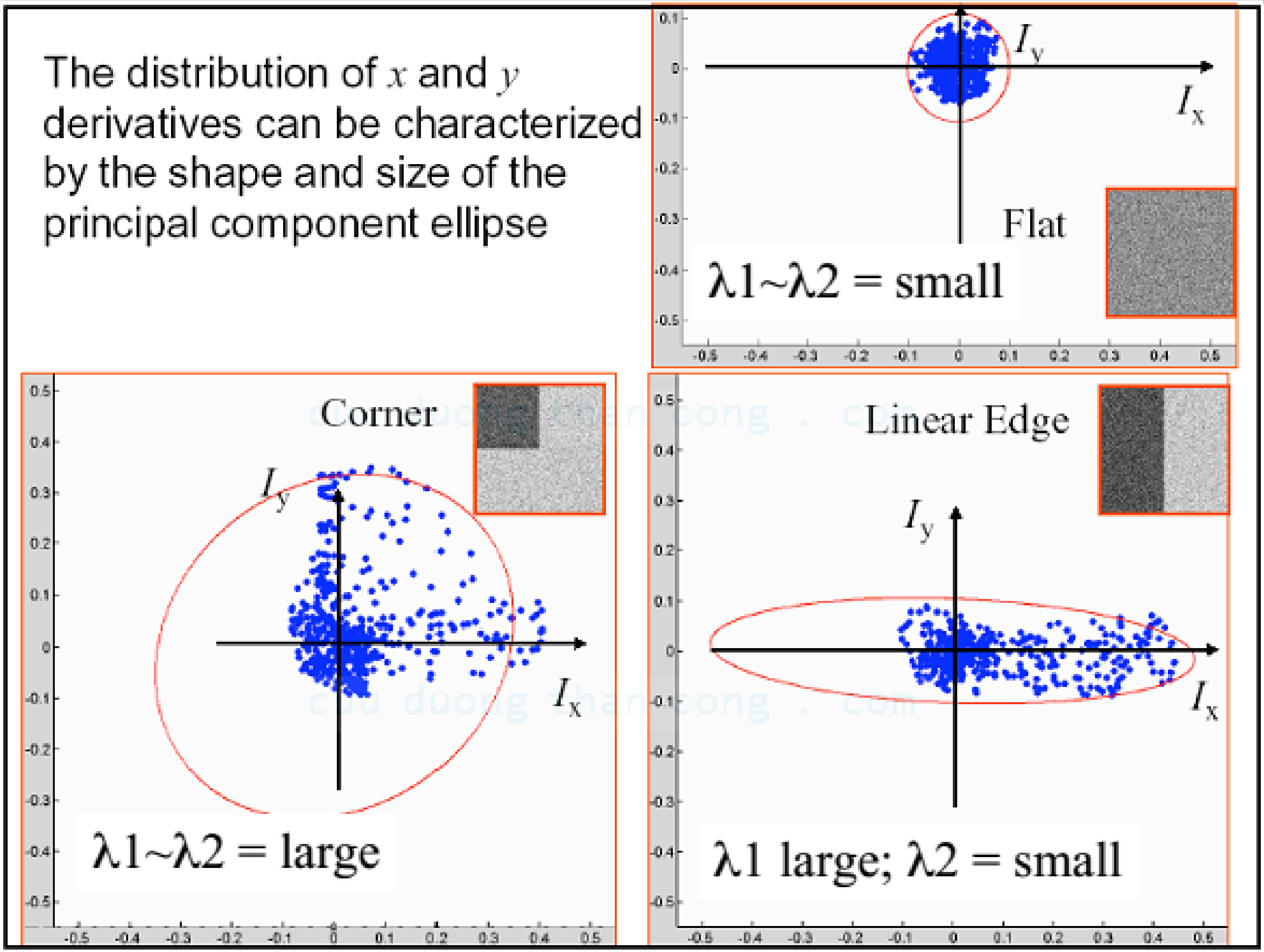
Plotting derivatives as 2D points

The distribution of the x and y derivatives is very different for all three types of patches



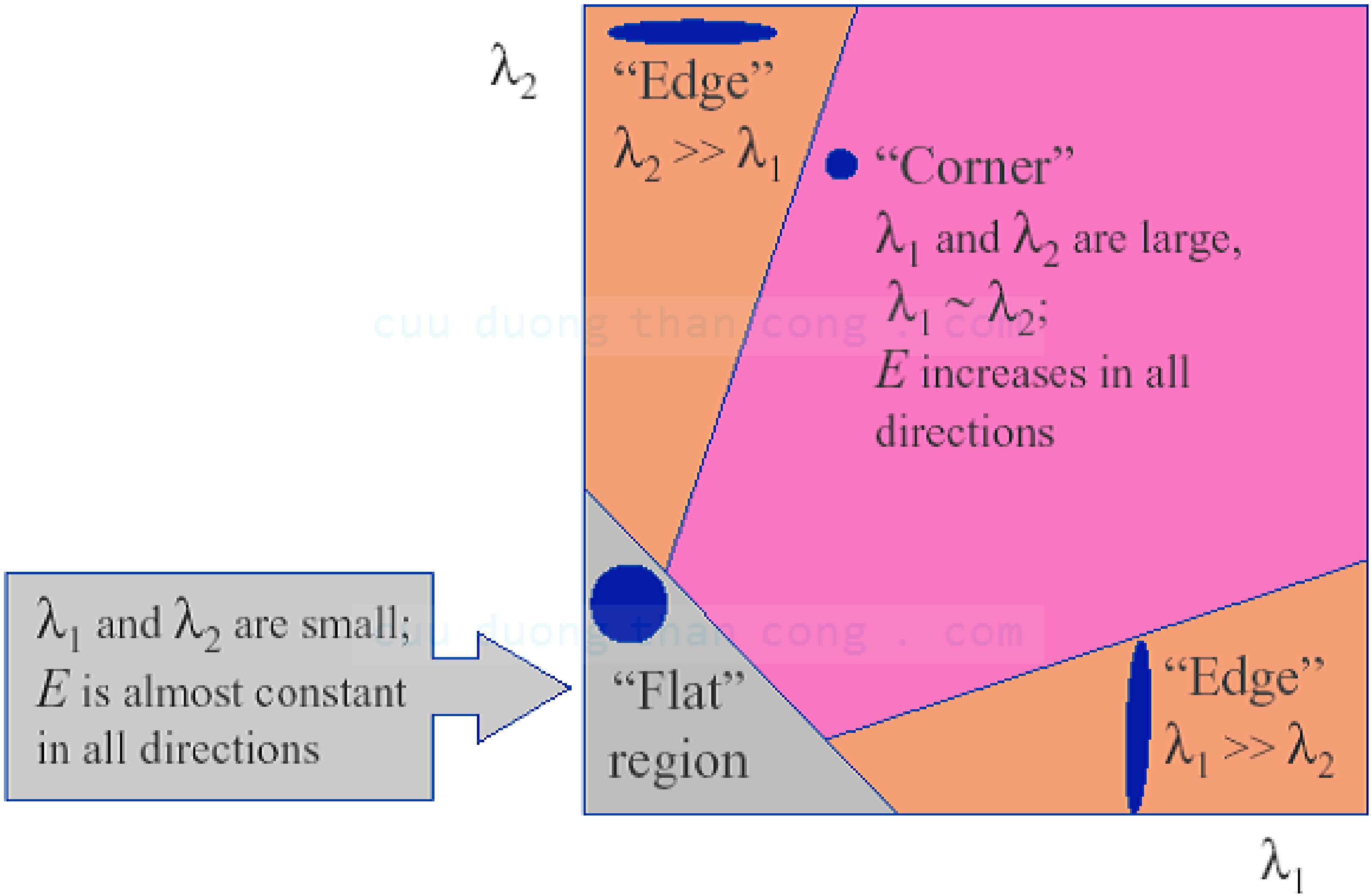
Fitting ellipse to each set of points

The distribution of x and y derivatives can be characterized by the shape and size of the principal component ellipse



Classification via eigenvalues

- Classification of image points using eigenvalues of M



Corner response measure

- Measure of corner response

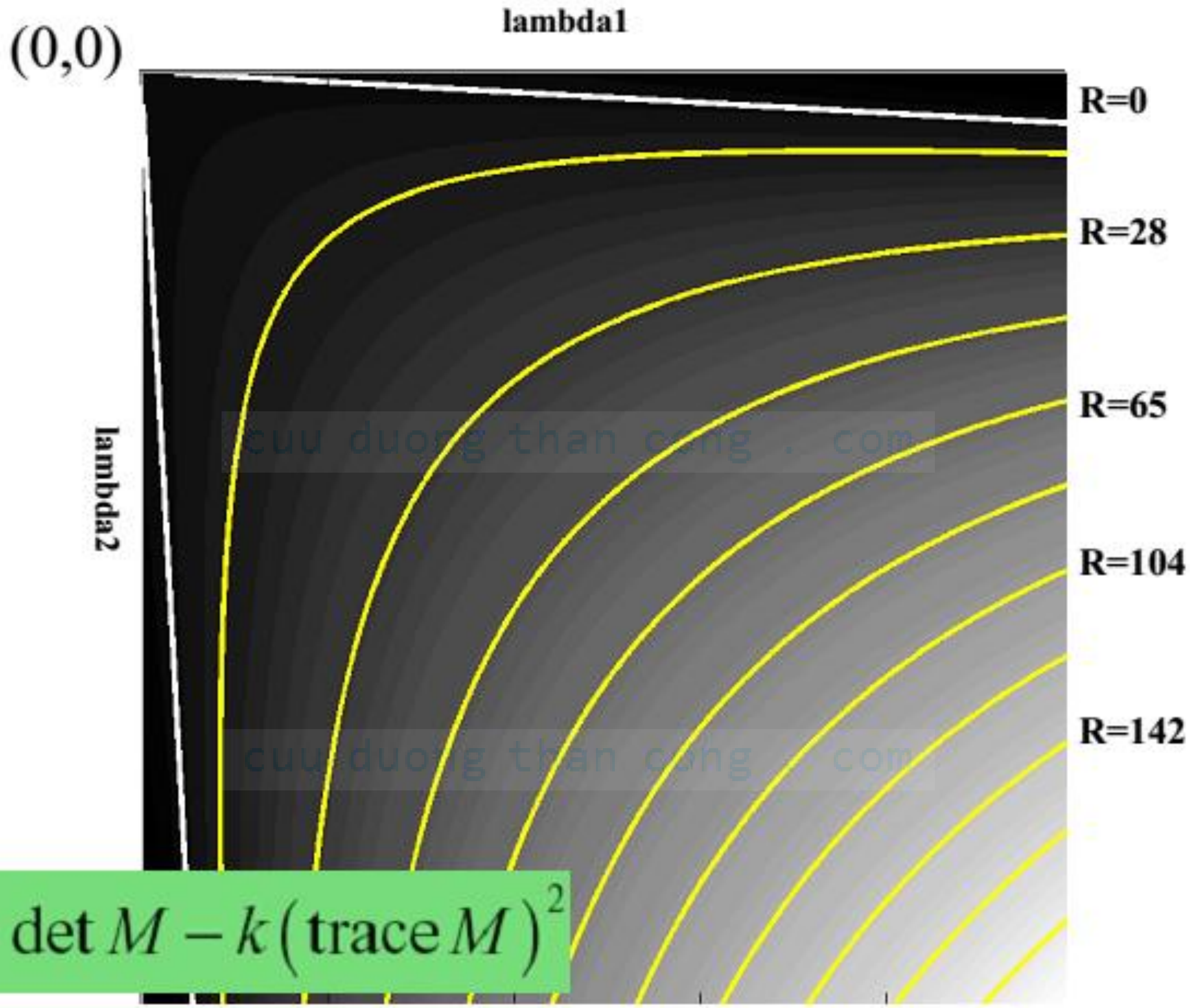
$$R = \det M - k(\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

  - where k is an empirically determined constant, $k = 0.04 - 0.06$

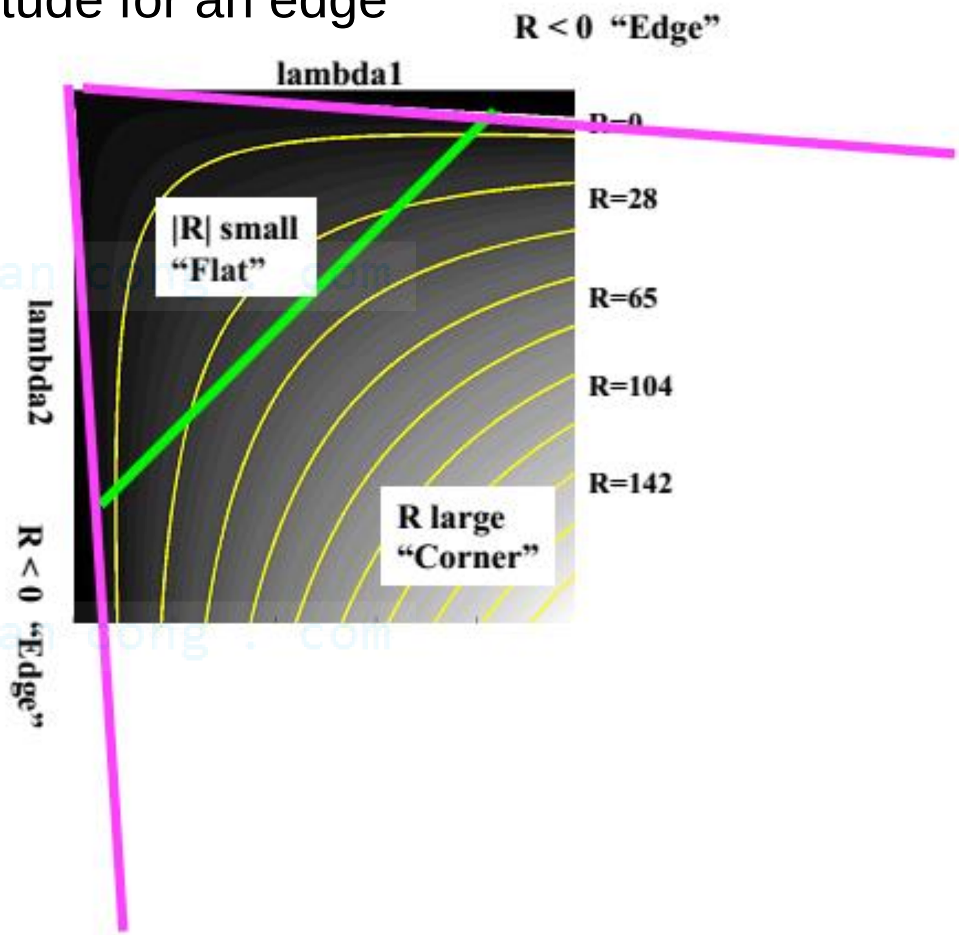
Corner response map



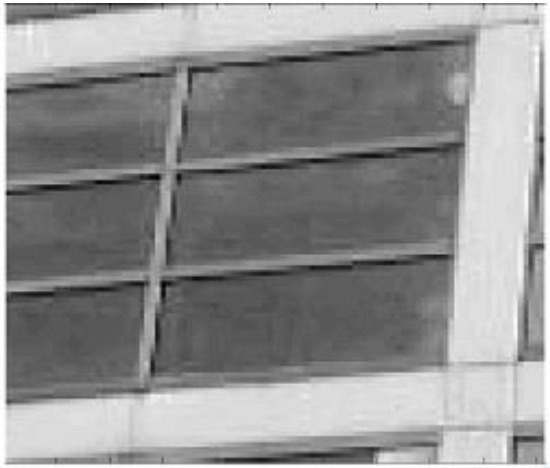
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

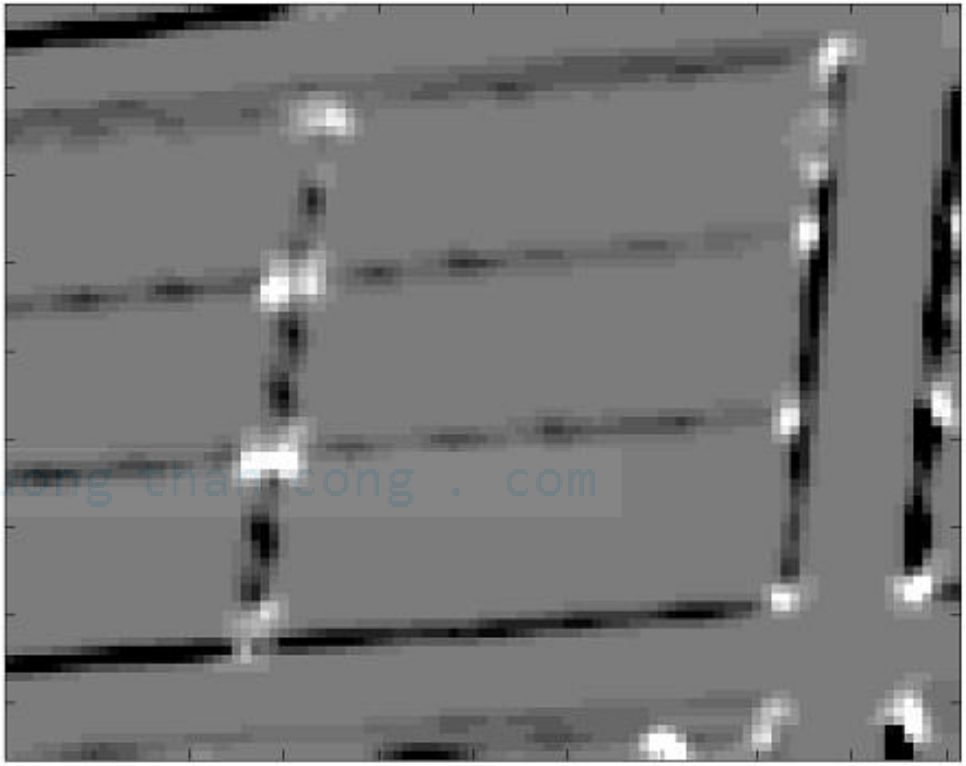
Corner response map

- R depends only on eigenvalues of M
  - R is large for a corner
  - R is negative with large magnitude for an edge
  - $|R|$ is small for a flat region



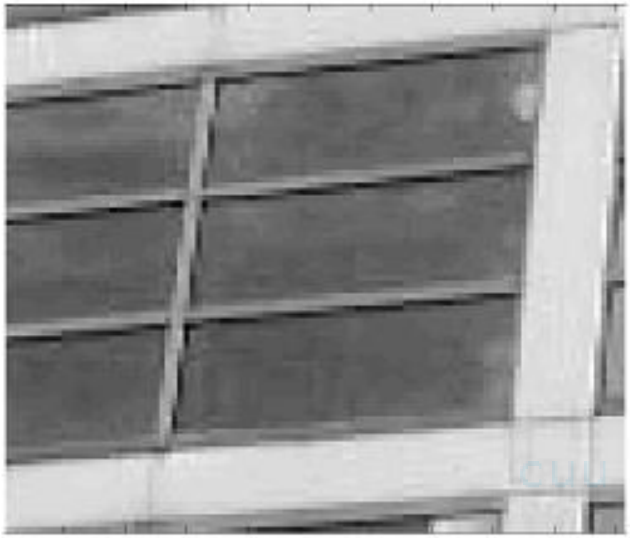
Corner response: An example

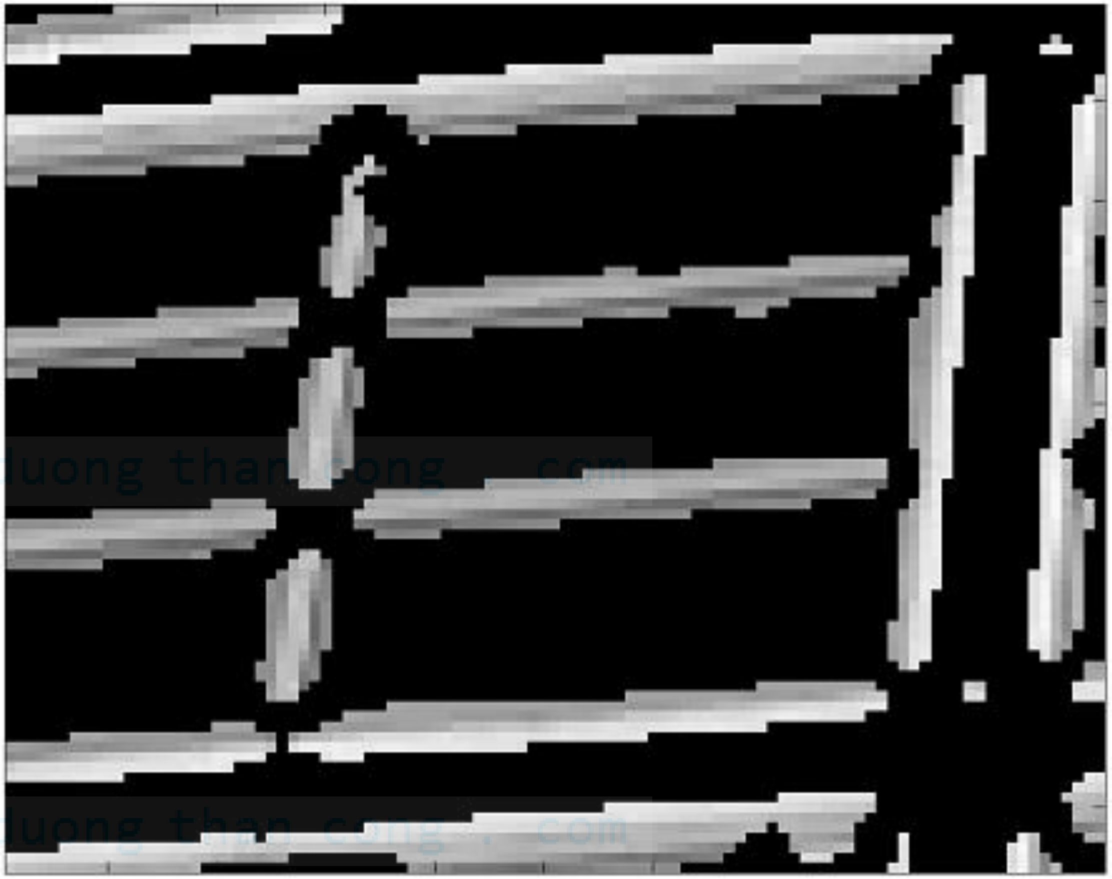




- Harris R score
  - I_x and I_y are computed using Sobel operator
  - Windowing function $w = Gaussian,\ \sigma = 1$

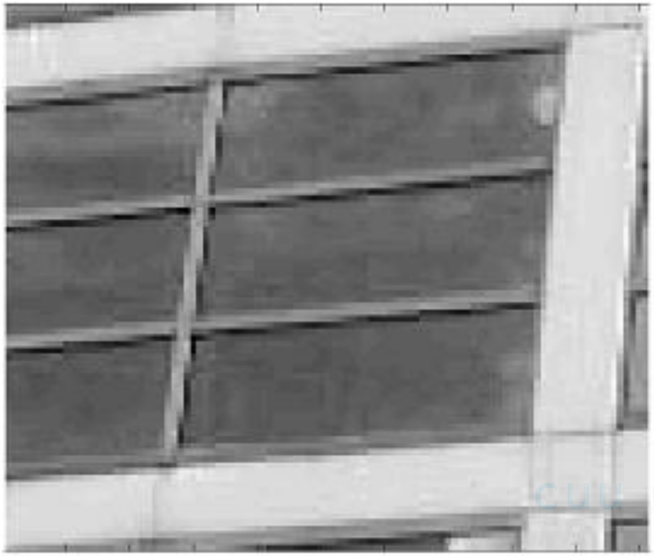
Corner response: An example





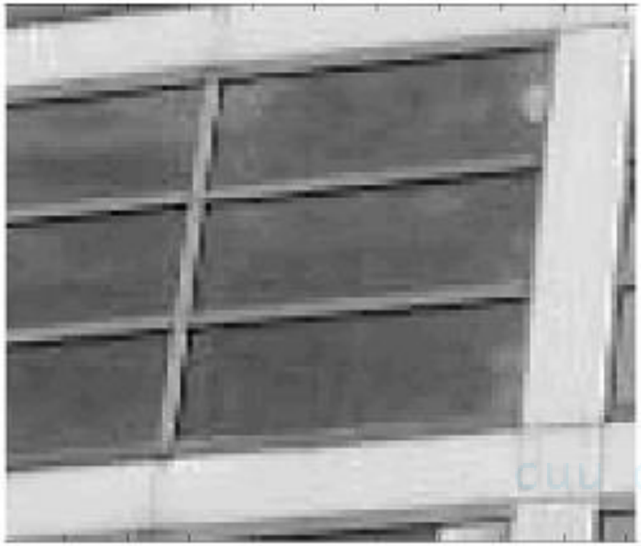
- Threshold $R < -10000$ (edges)

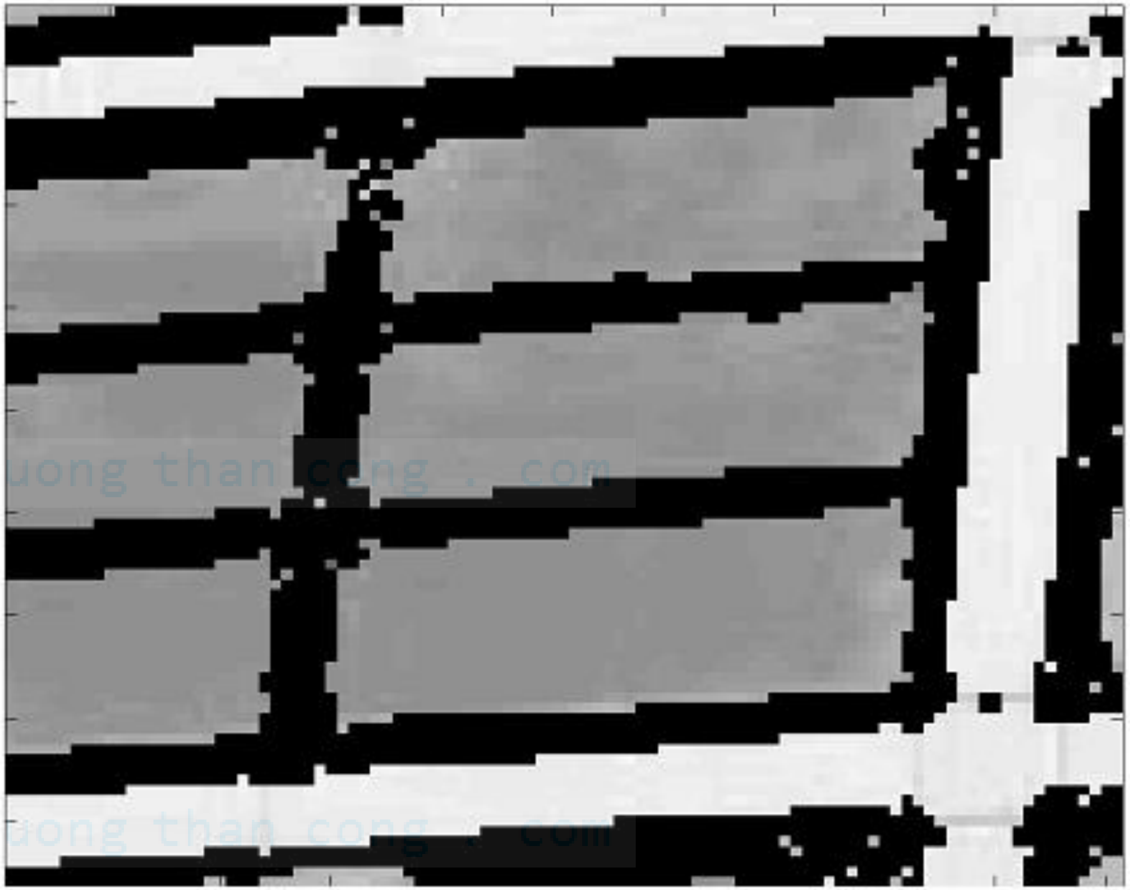
Corner response: An example



- Threshold $R > 10000$ (corners)

Corner response: An example





- $-10000 < R < 10000$ (neither edges nor corners)

Harris Corner Algorithm

1. Computer x and y derivatives of image

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Computer products of derivatives at every pixel

$$I_{x2} = I_x I_x \qquad I_{y2} = I_y I_y \qquad I_{xy} = I_x I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} I_{x2} \qquad S_{y2} = G_{\sigma'} I_{y2} \qquad S_{xy} = G_{\sigma'} I_{xy}$$

4. Define at each pixel (x, y) the matrix $M(x, y) = \begin{bmatrix} S_{x2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{x2}(x, y) \end{bmatrix}$

5. Compute the response of the detector at each pixel

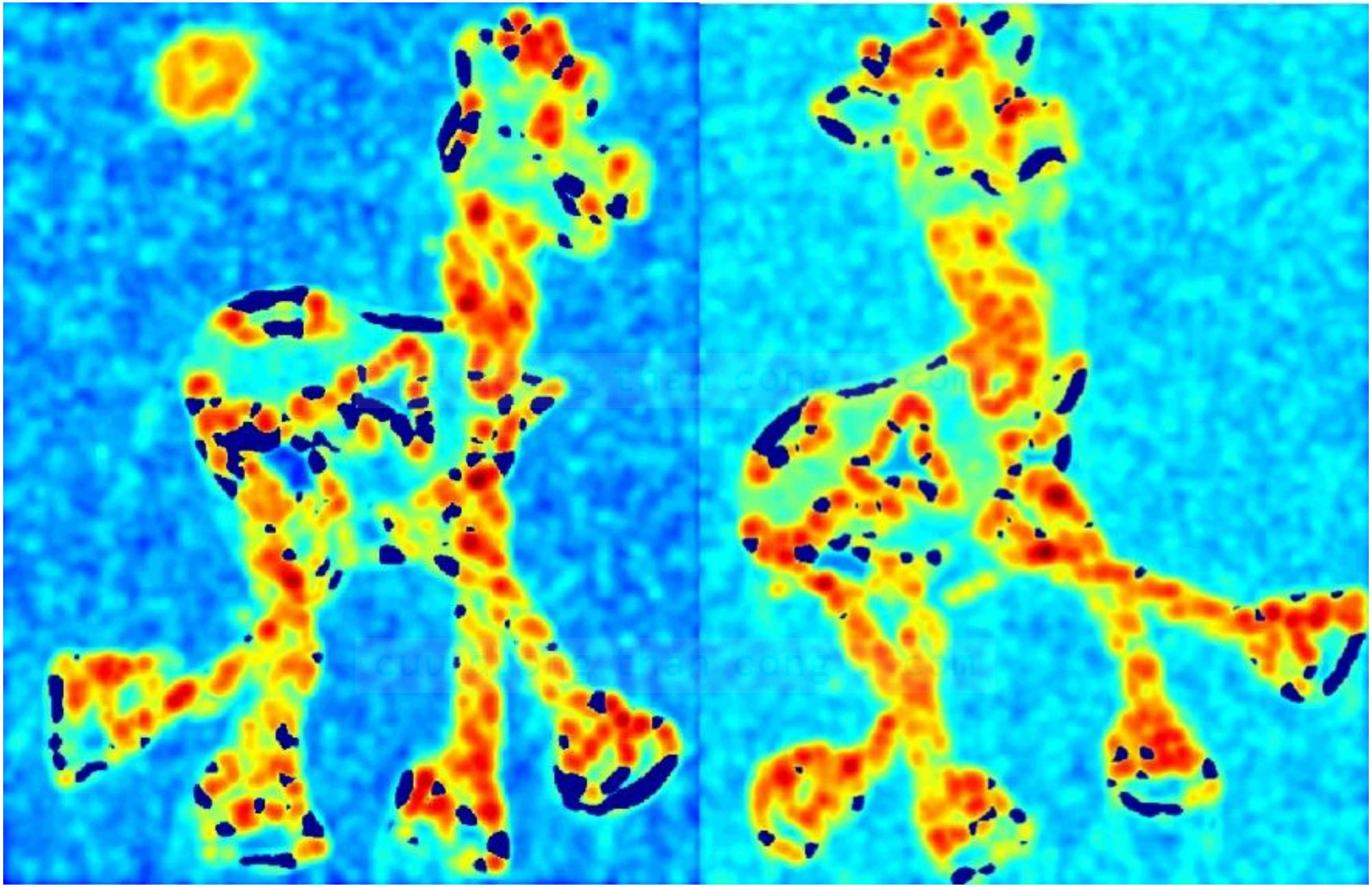
$$R = \det M - k(trace(M))^2$$

6. Threshold on value of R. Compute non-maximum suppresion

Harris detector: An example

f value (red high, blue low)

$$f = \frac{\det M}{\operatorname{trace} M}$$



Threshold (f > value)

$$f = \frac{\det M}{\operatorname{trace} M}$$

Find Local Maxima of f

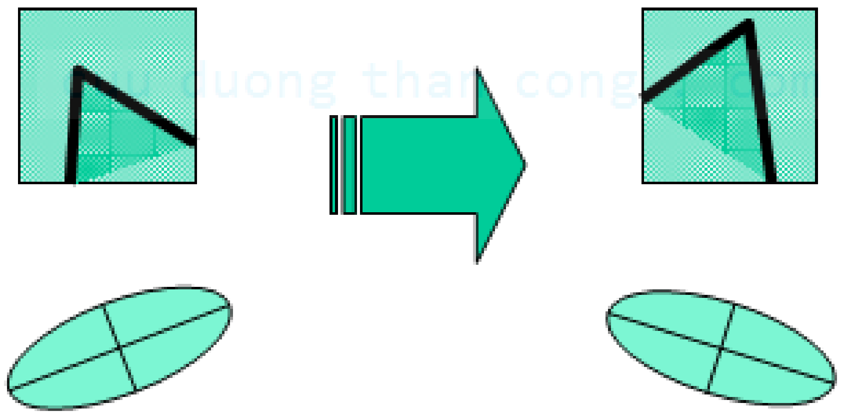
$$f = \frac{\det M}{\operatorname{trace} M}$$

Harris features (in red)

The tops of the horns are detected in both images

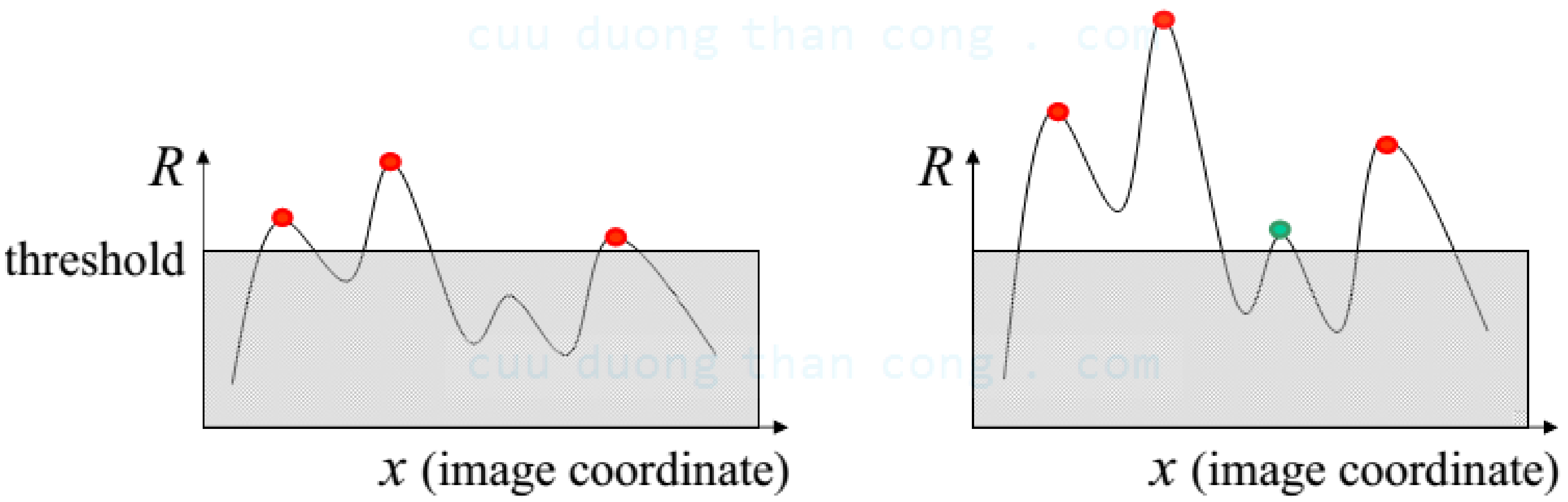
Properties of Harris Detector

- Rotation invariance
  - Ellipse rotates but its shape (i.e. eigenvalues) remains the same
  - Corner response R is invariant to image rotation



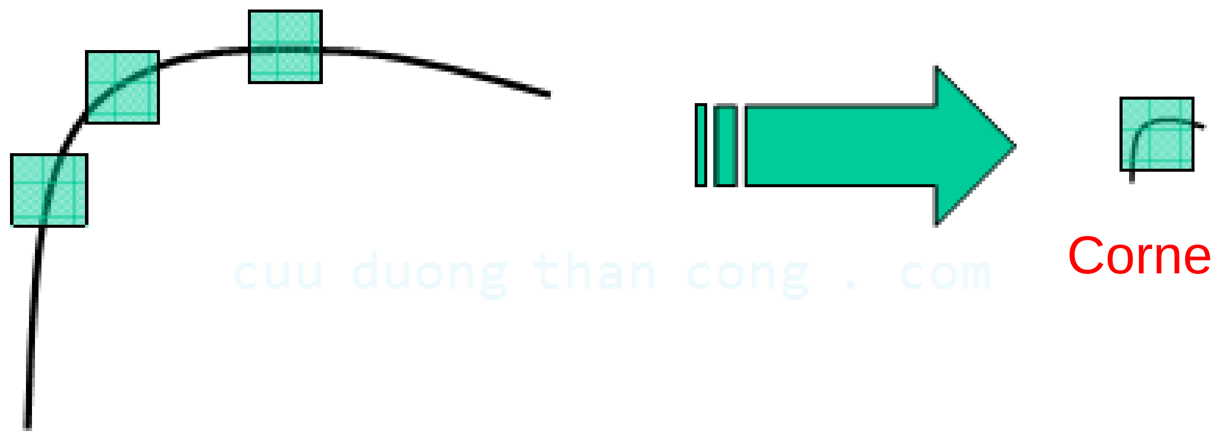
Properties of Harris Detector

- Partial invariance to additive and multiplicative intensity changes
  - Using derivatives only $\Rightarrow$ invariance to intensity shift $I \rightarrow I + b$
  - Intensity scale: $I \rightarrow aI$



Properties of Harris Detector

- Not invariant to scaling

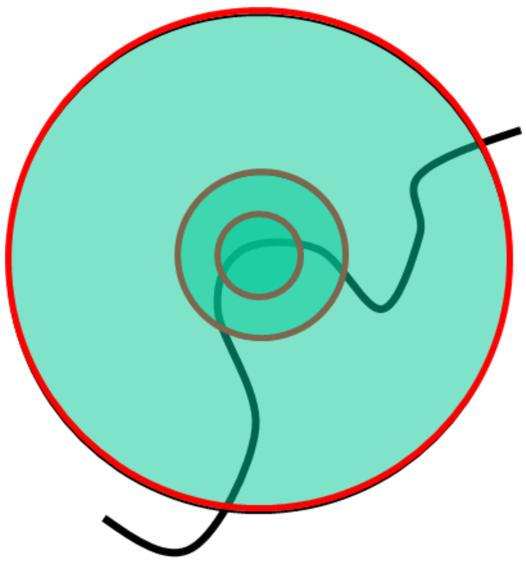


Corner!

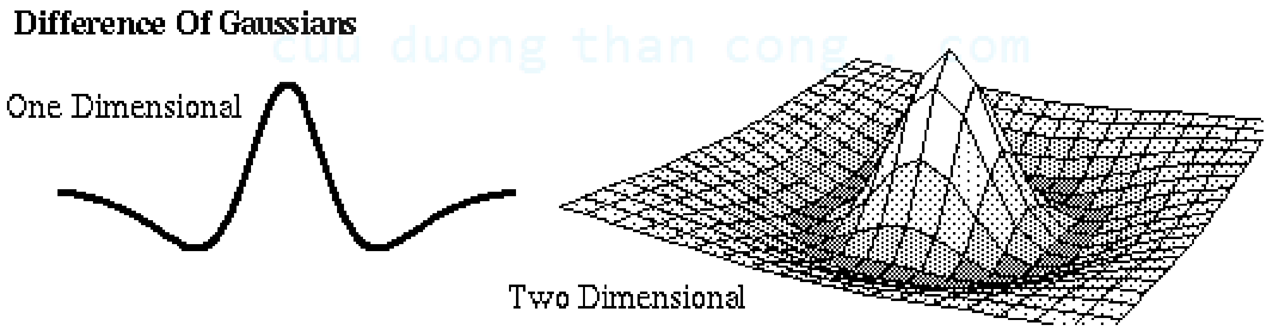
All points will be
classified as edges

Properties of Harris Detector

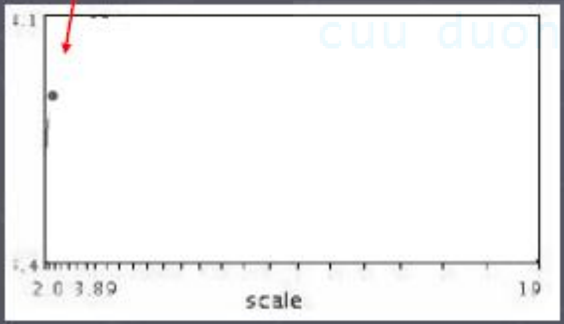
- Suppose you're looking for corners



- Key idea: find scale that gives local maximum of f
  - f is a local maximum in both position and scale
  - Common definition of f: Laplacian (or difference between two Gaussian filtered images with different sigmas σ)

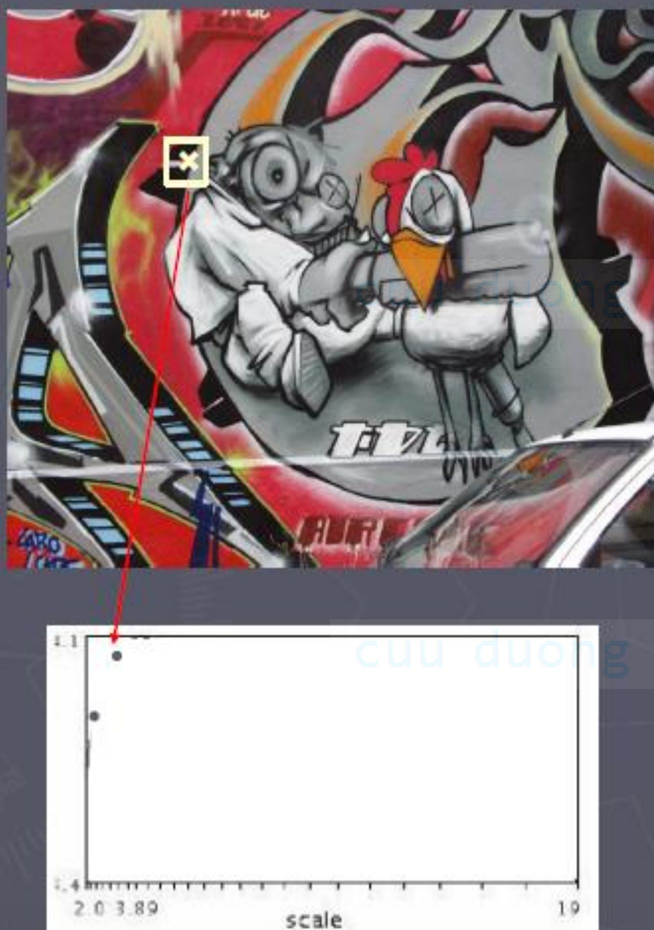


Automatic scale selection



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

Automatic scale selection



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

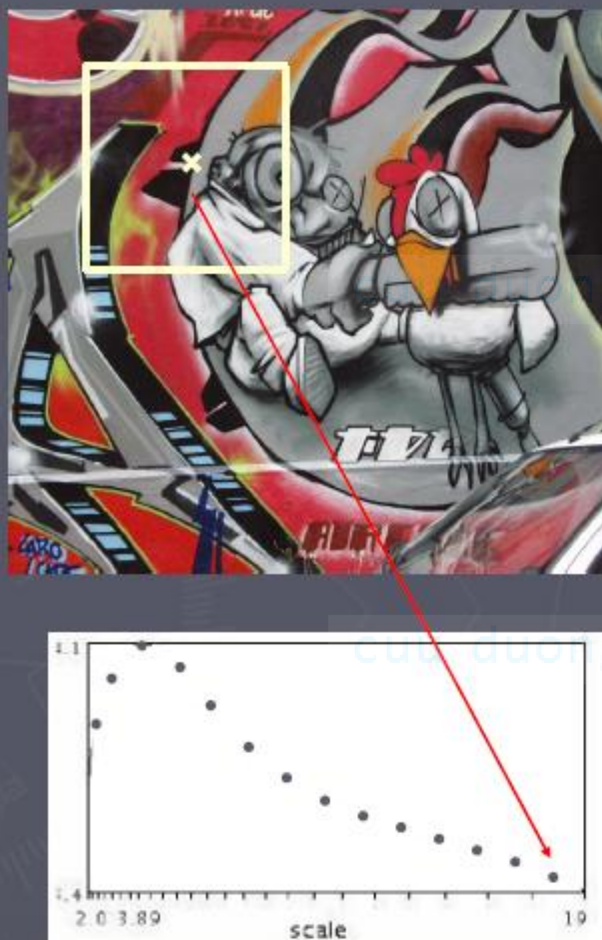
Automatic scale selection

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

Automatic scale selection

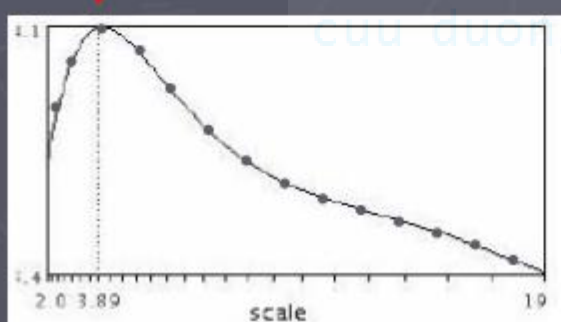
$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection



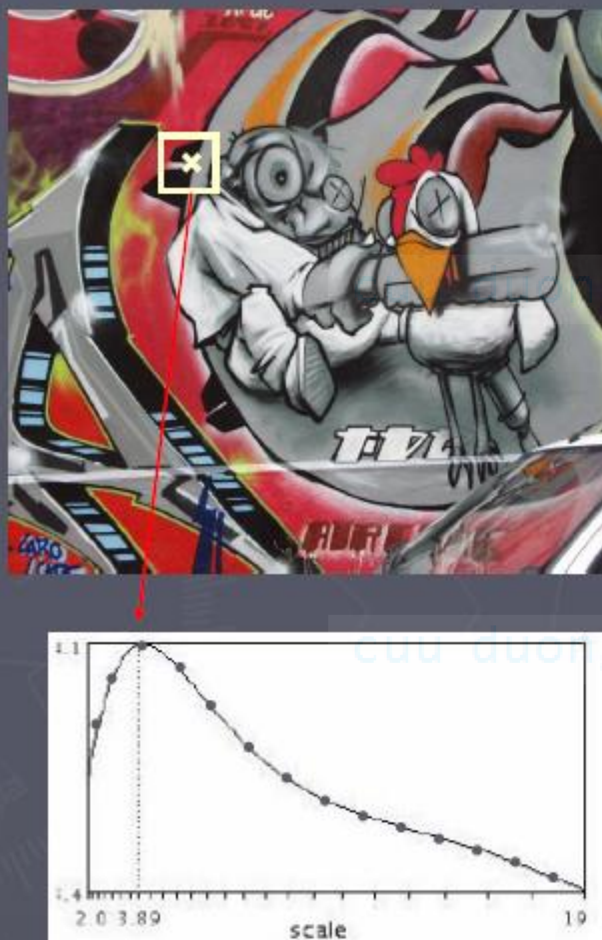
$$f(I_{i_1 \dots i_m}(x,\sigma))$$

Automatic scale selection

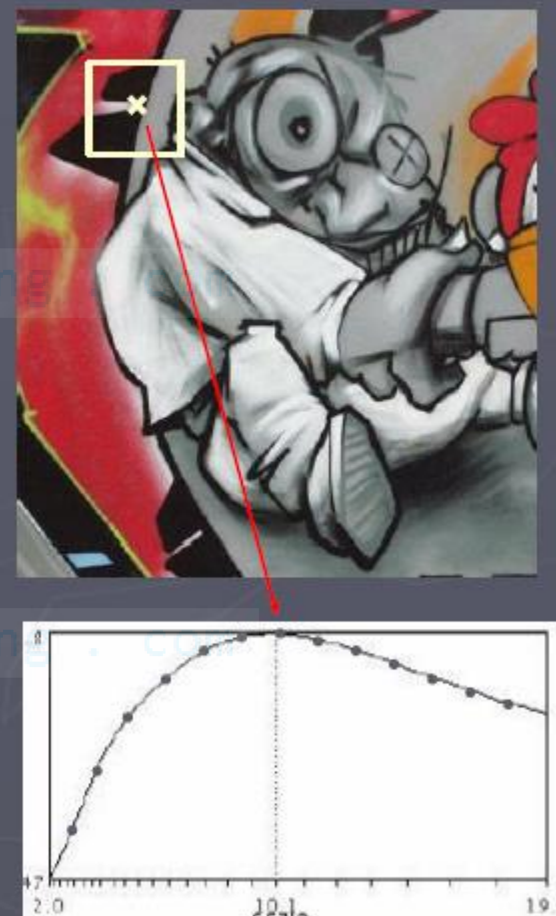


$$f(I_{i_1 \ldots i_m}(x,\sigma))$$

Automatic scale selection



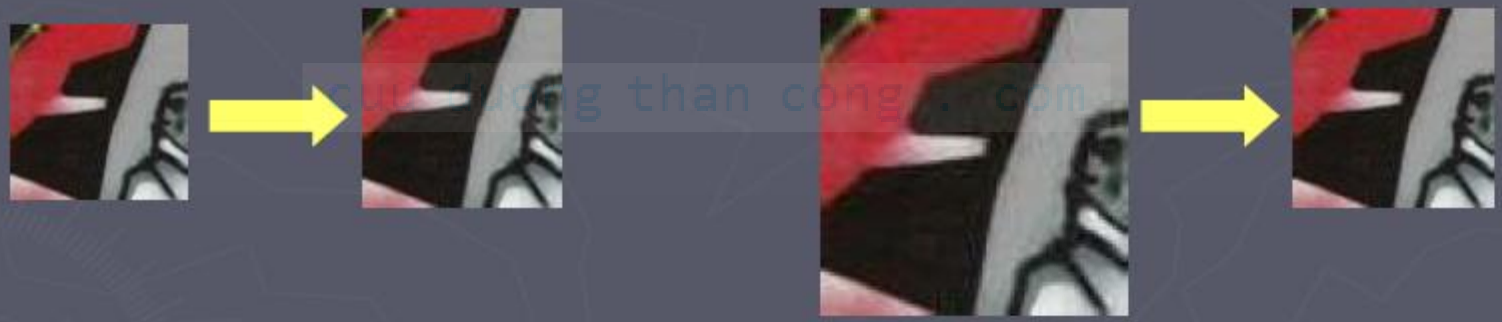
$f(I_{i_1 \ldots i_m}(x, \sigma))$



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

Automatic scale selection

Normalize: rescale to fixed size

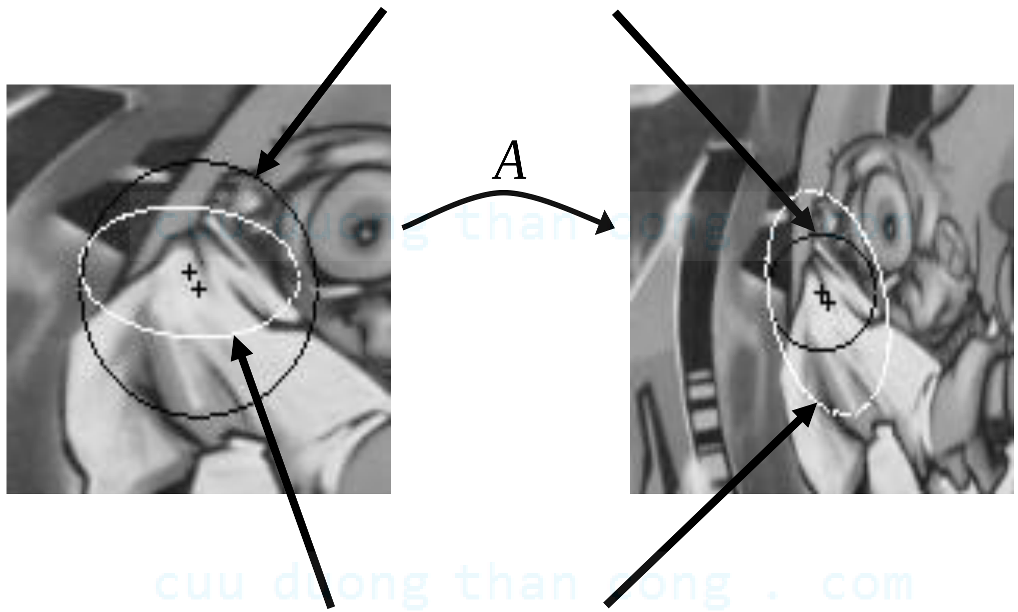


Typically, regions are normalized to circular regions of uniform diameter of 41 pixels.

Affine invariant regions

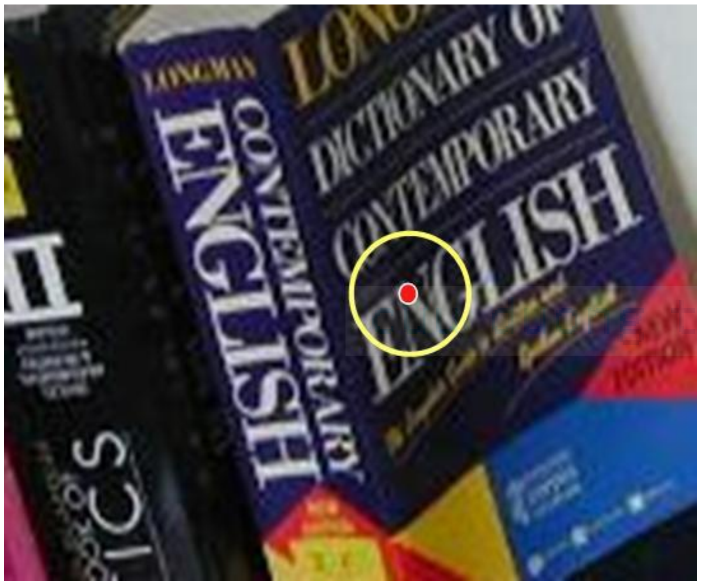
- Scale invariance is not sufficient for large baseline changes

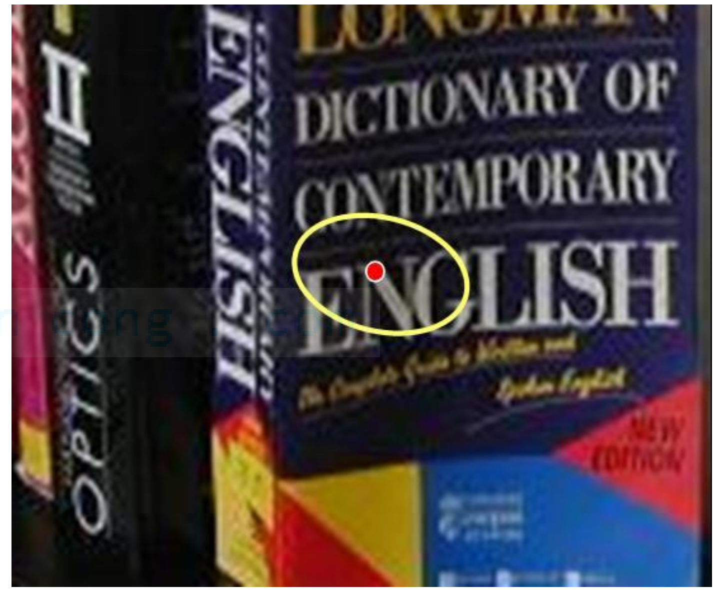
detected scale invariant region



projected regions, viewpoint changes can locally be approximated by an affine transformation A

Affine invariant regions: Examples





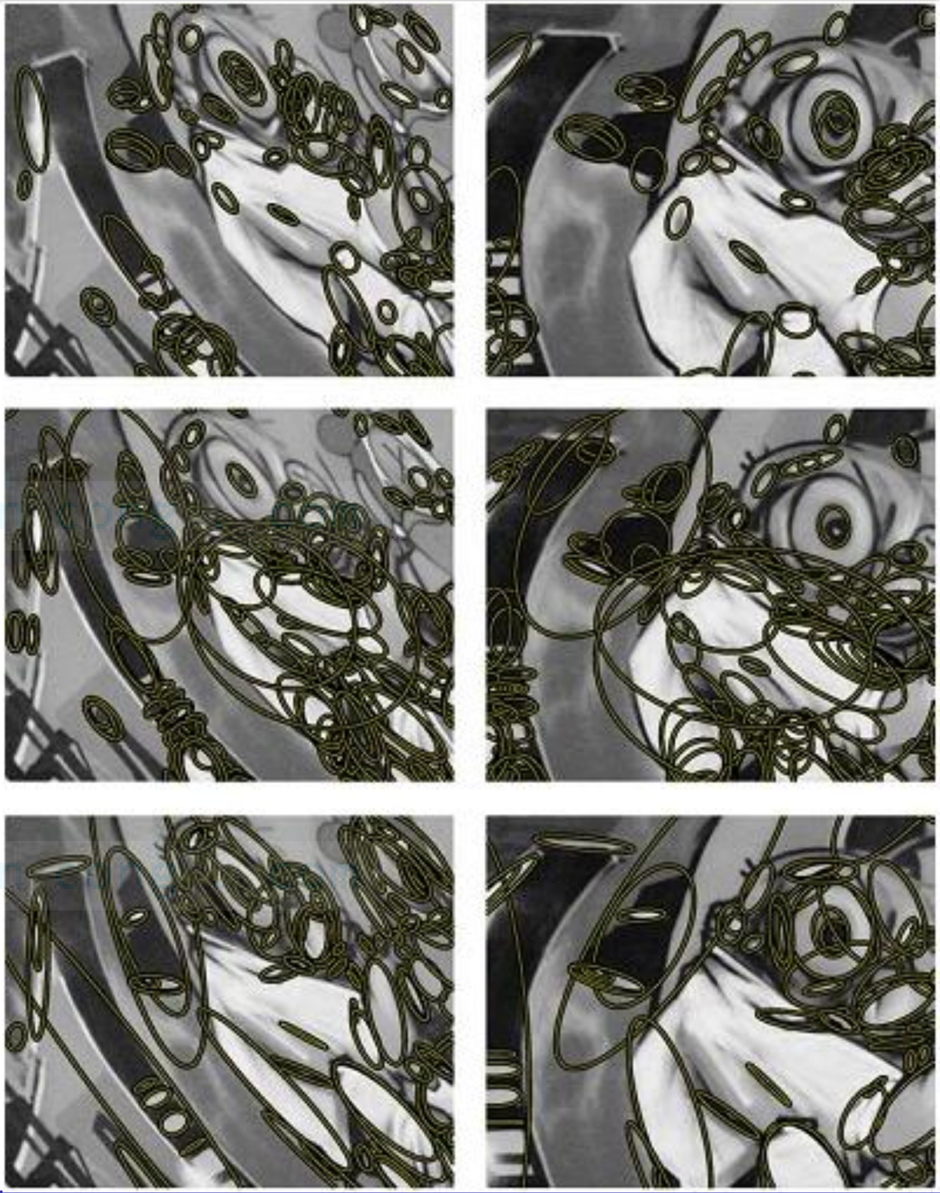
Affine invariant regions: Examples

Harris-Affine Detector

- Initialize with scale-invariant Harris points
- Estimation of the affine neighbourhood with the second moment matrix [Lindeberg'94]
- Apply affine neighbourhood estimation to the scale-invariant interest points [Mikolajczyk & Schmid'02, Schaffalitzky & Zisserman'02]
- Excellent results in a recent comparison

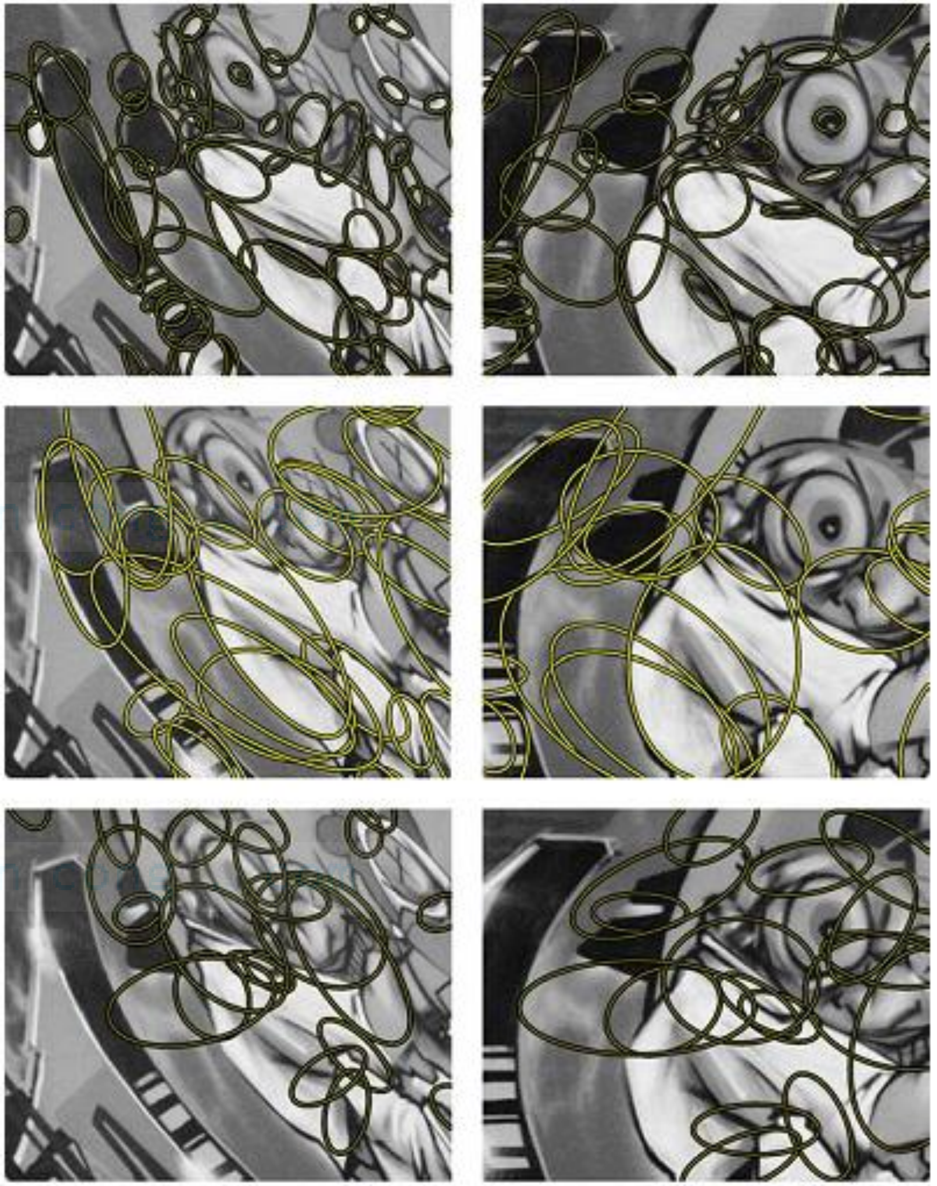
Affine-Covariant detectors

- Harris-Affine:
- Hessian-Affine:
- Maximally Stable Extremal Region (MSER)



Affine-Covariant detectors

- Intensity Extrema-Based Region (IBR)
- Edge-Based Region (EBR)
- Salient Region



Affine-Covariant detectors: Open sources

www.robots.ox.ac.uk/~vgg/research/affine/index.html

Affine Covariant Features

KATHOLIEKE UNIVERSITEIT LEUVEN

INRIA RHÔNE ALPES

Collaborative work between: the Visual Geometry Group, Katholieke Universiteit Leuven, Inria Rhone-Alpes and the Center for Machine Perception.

Overview

This page is focused on the problem of detecting affine invariant features in arbitrary images and on the performance evaluation of region detectors

Affine Covariant Regions

Image 1

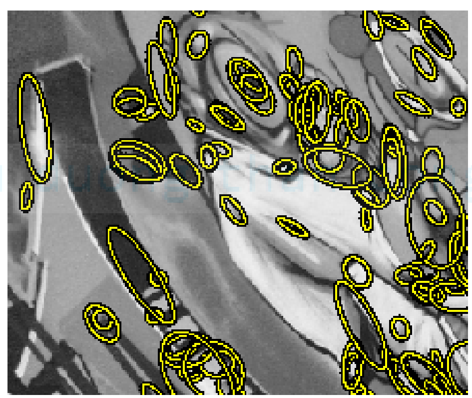
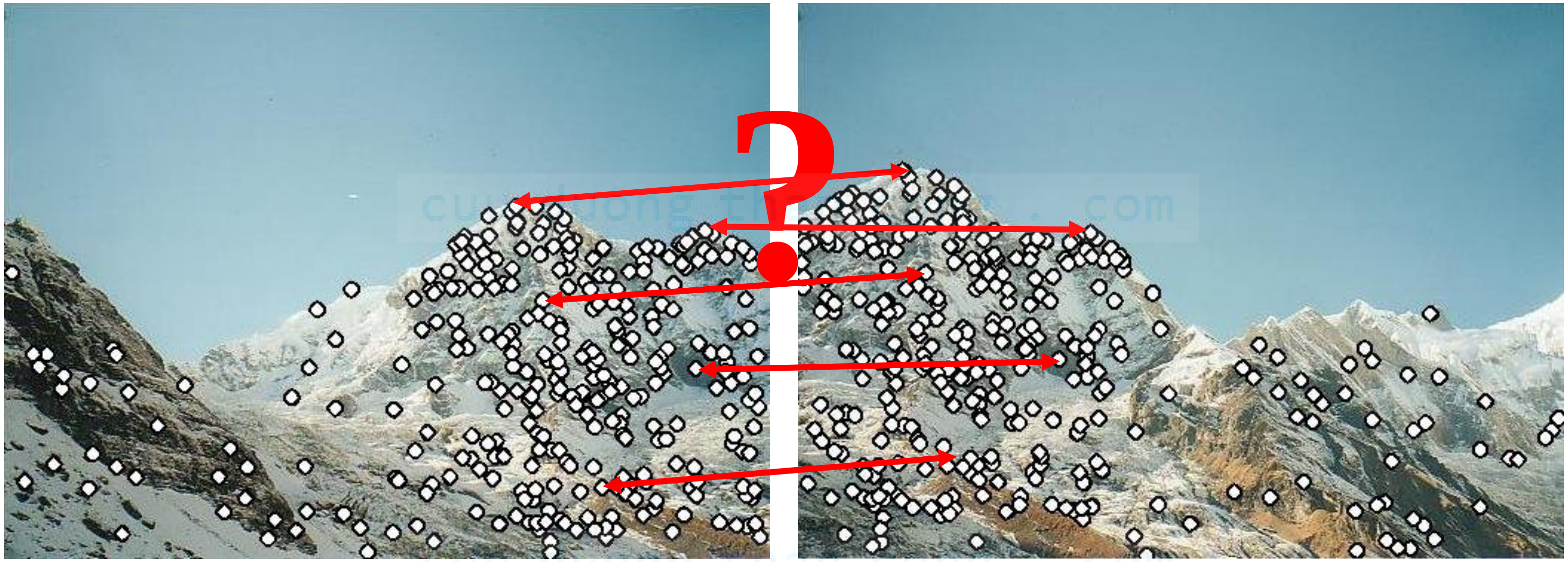


Image 2

Feature descriptors

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

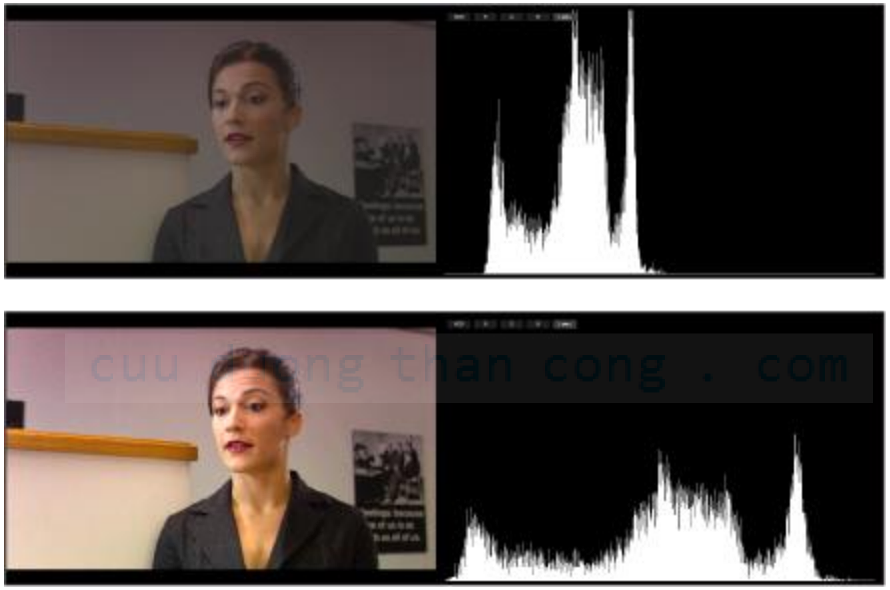


Section 8.3

REGION DESCRIPTORS

Motivation

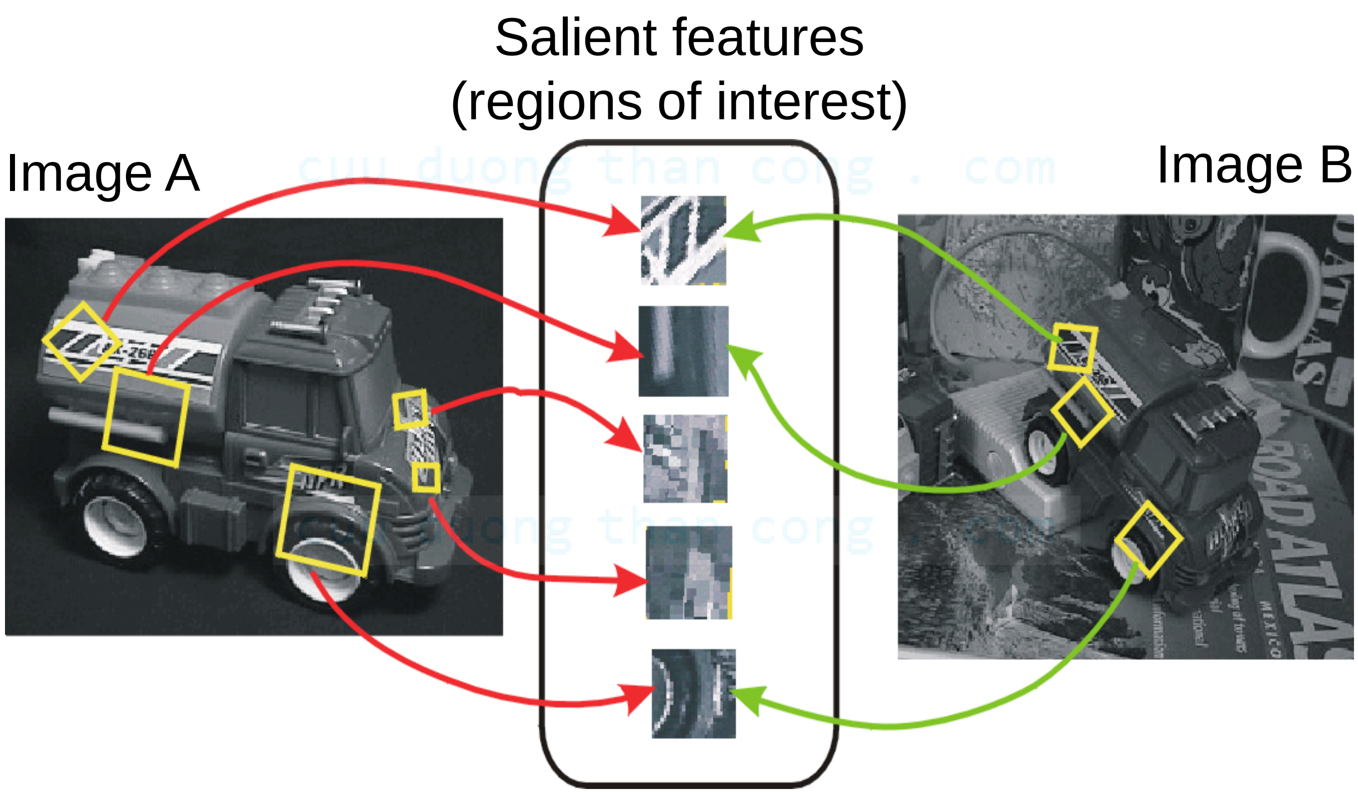
- Global feature from the whole image is often not desirable



- Instead match local regions which are prominent to the object or scene in the image

Local features

- **Local features** encode the image structure in spatial neighbourhoods at a set of feature points chosen at selected scales or orientations

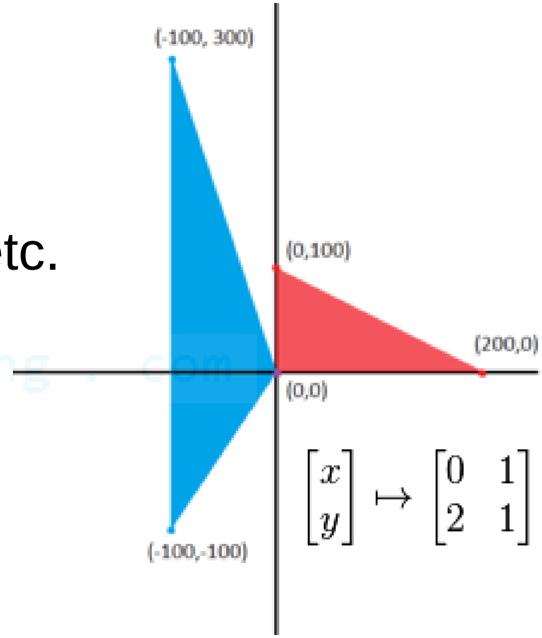


Requiments of a local feature

- Locality: robust to occlusion, clutter
- Distinctiveness: the feature descriptors should permit a high detection rate and low false positive rate
- Quantity: there should be enough points to represent the image
- Repetitive: detect the same points independently in each image
- Efficiency: real-time performance achievable
- Generality: exploit different types of features in different situations

Requiments of a local feature

- Invariant to geometric (i.e. affine) transformations
  - Translation
  - Scaling
  - Rotation
  - Sheer mapping, squeeze mapping, etc.

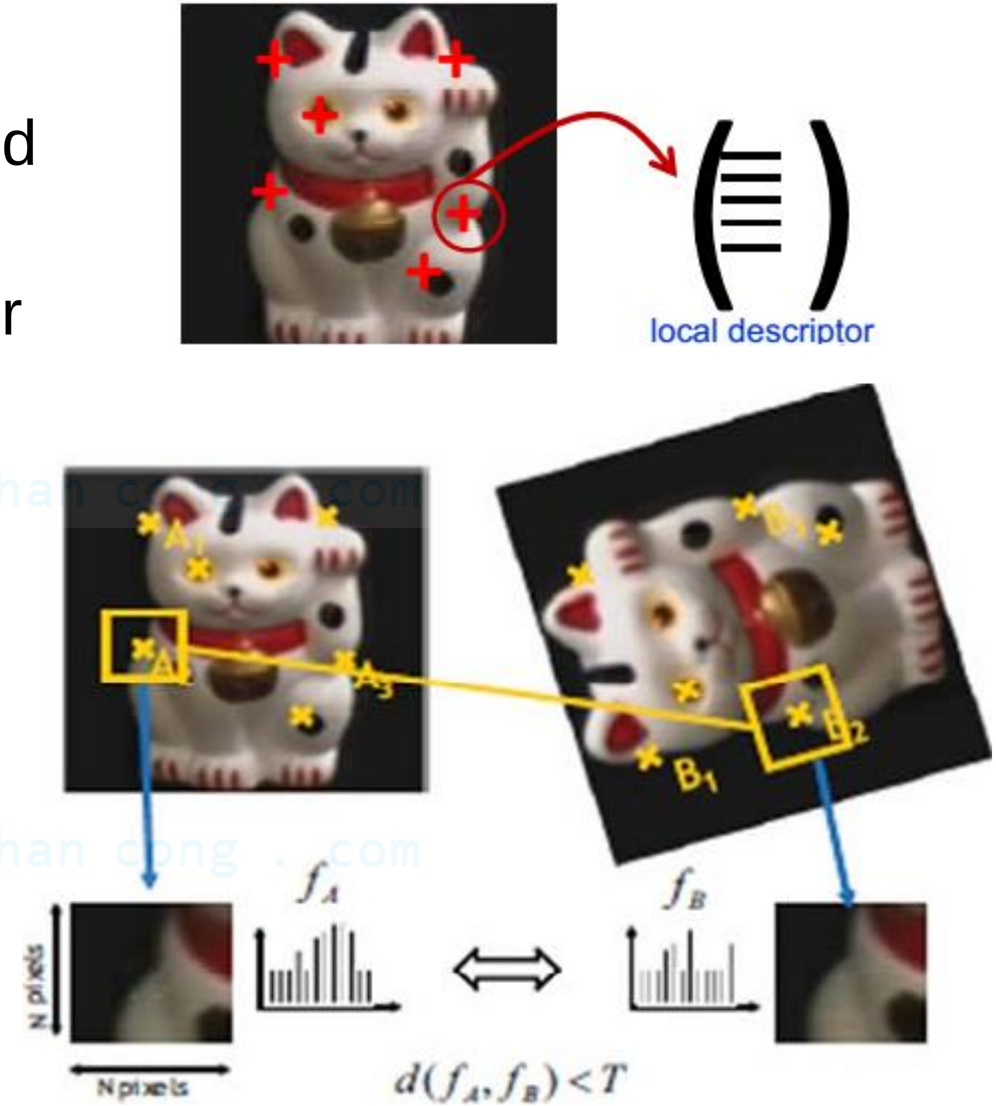


$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -100 \\ -100 \end{bmatrix}$$

- Invariant to photometric (illumination, exposure) changes
- Less affected by noise or blur

General approach

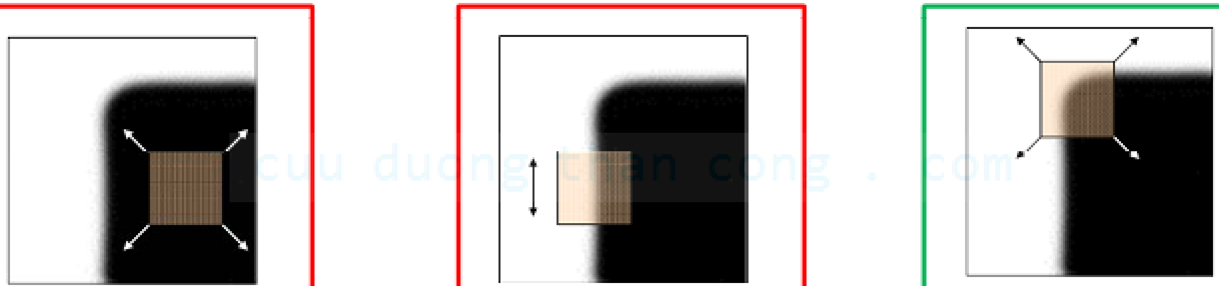
1. Find the interest points
2. Consider the region around each keypoint
3. Compute a local descriptor from the region and normalize the feature
4. Match local descriptor



Some popular detector

- Harris/Hessian corner detector
- Harris/Hessian Laplacian/Affine detector
- Laplacian of Gaussian / Difference of Gaussian detector
- Maximally Stable Extremal Regions (MSER)
- Many others ….

Looks for change in image gradient in two direction - CORNERS



No change in any direction

Change in one direction only

Change in both the directions

Harris Corner Detector [Forstner and Gulch, 1987]

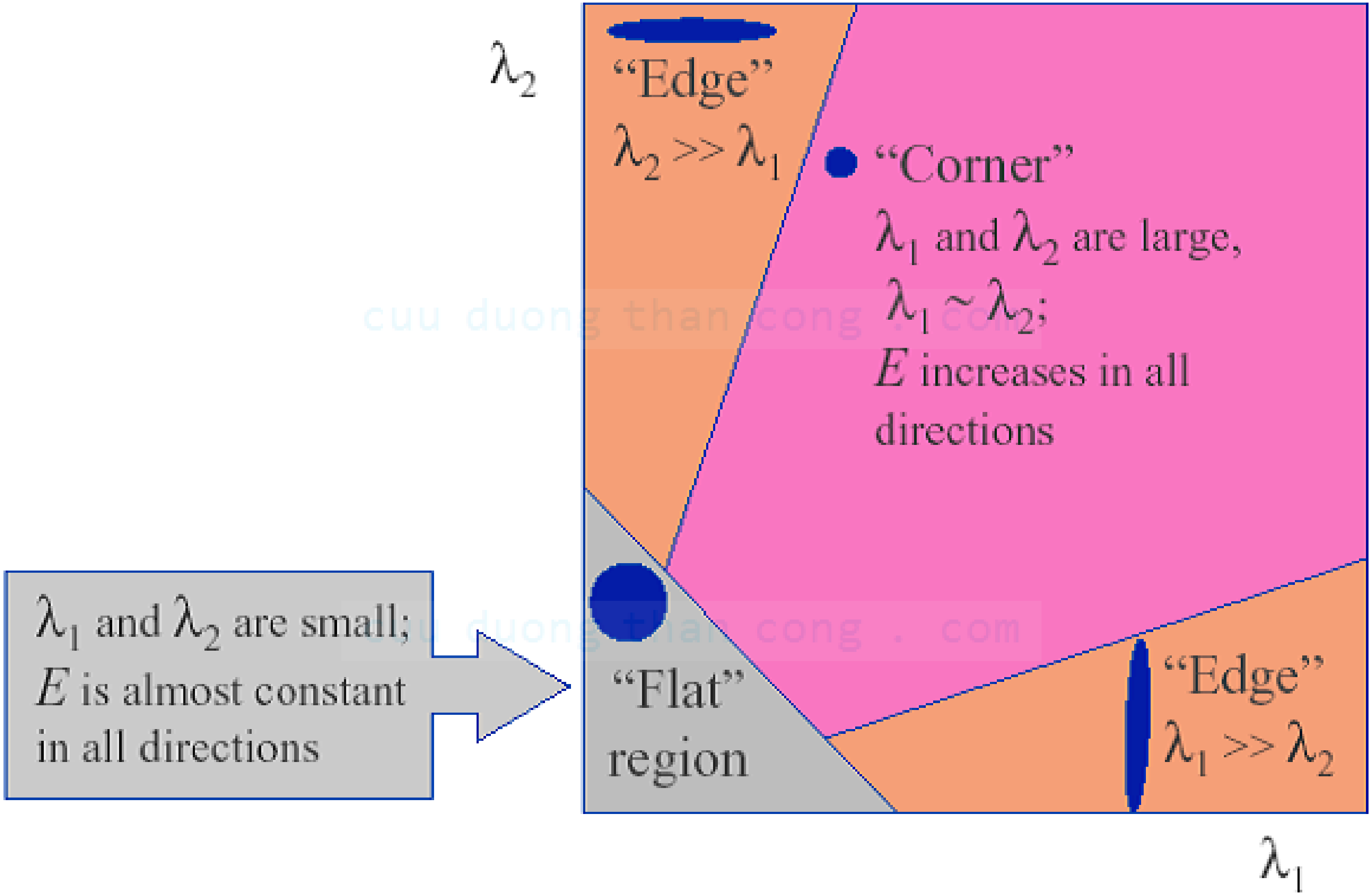
- Search for local neighborhoods where the image content has two main directions (eigenvectors)
- Consider 2nd moment matrix

$$M = \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_y I_x & {I_y}^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

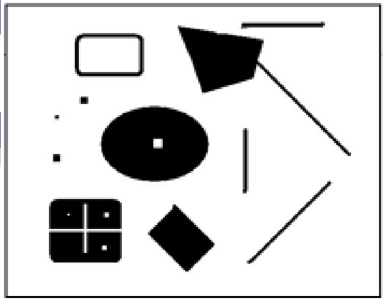
  - If either λ is close to 0, then this is not a corner, so look for locations where both are large.

Harris Corner Detector

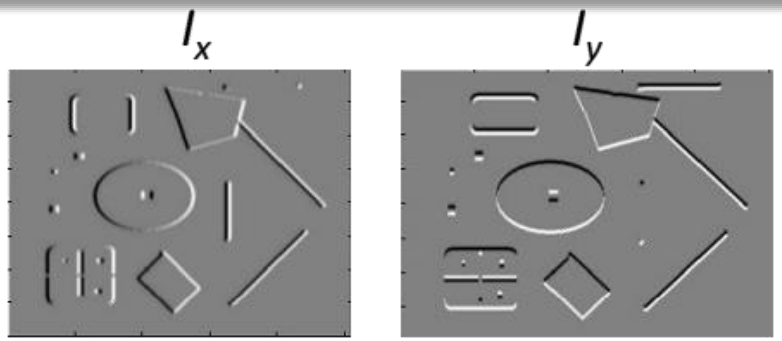
- Eigen decomposition: visualization



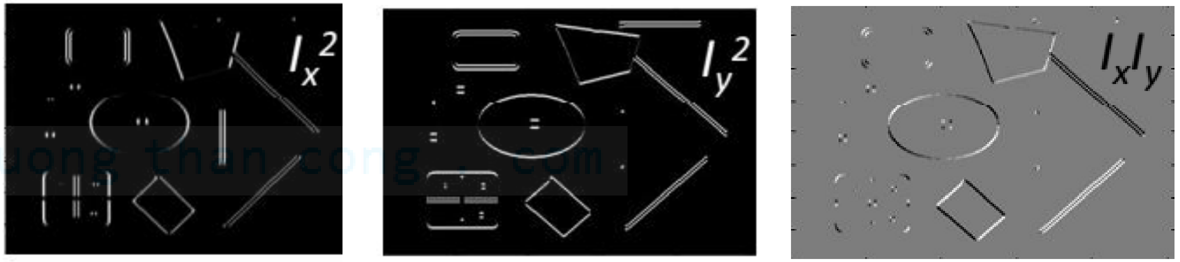
Harris Corner Detector: Exampl



- Image derivatives



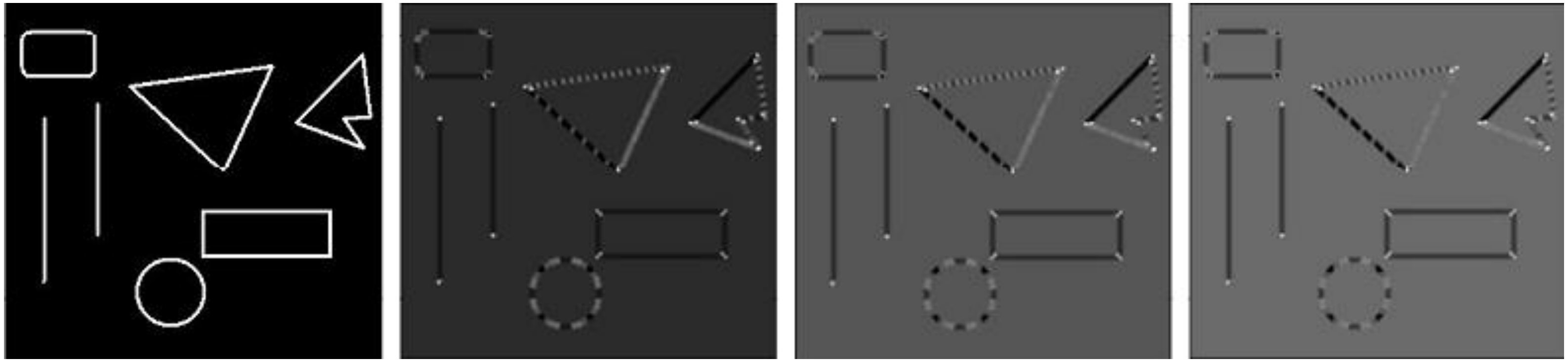
- Square of derivatives



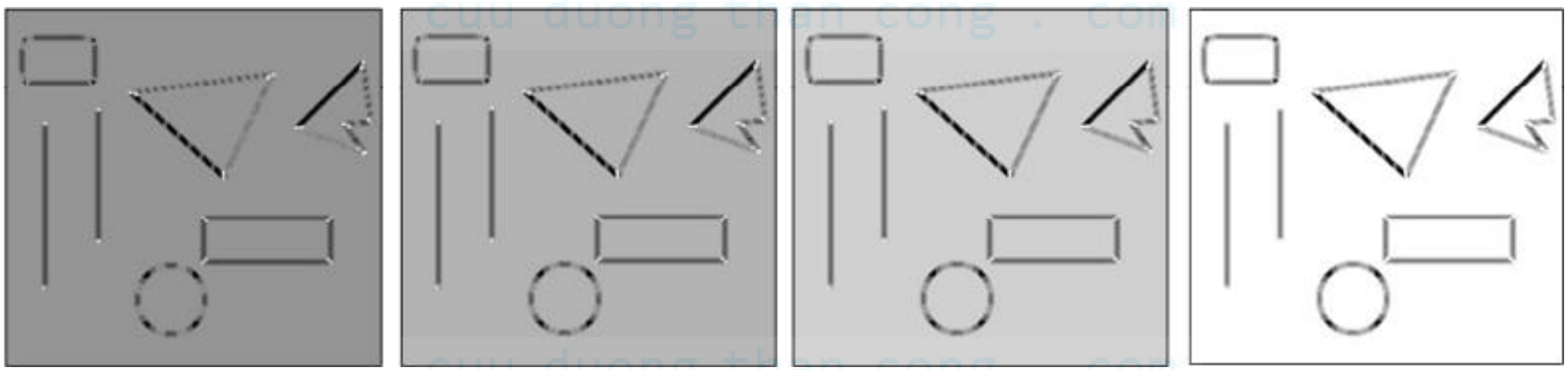
- Gaussian filter

- Cornerness function – both eigenvalues are strong

Harris Corner Detector: Examples



$k = 0.04$ $k = 0.08$ $k = 0.1$



$k = 0.14$ $k = 0.17$ $k = 0.2$ $k = 0.25$

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

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

Harris Corner Detector: Examples

Effect: A very precise corner detector

Hessian Corner Detector [Beaudet, 1978]

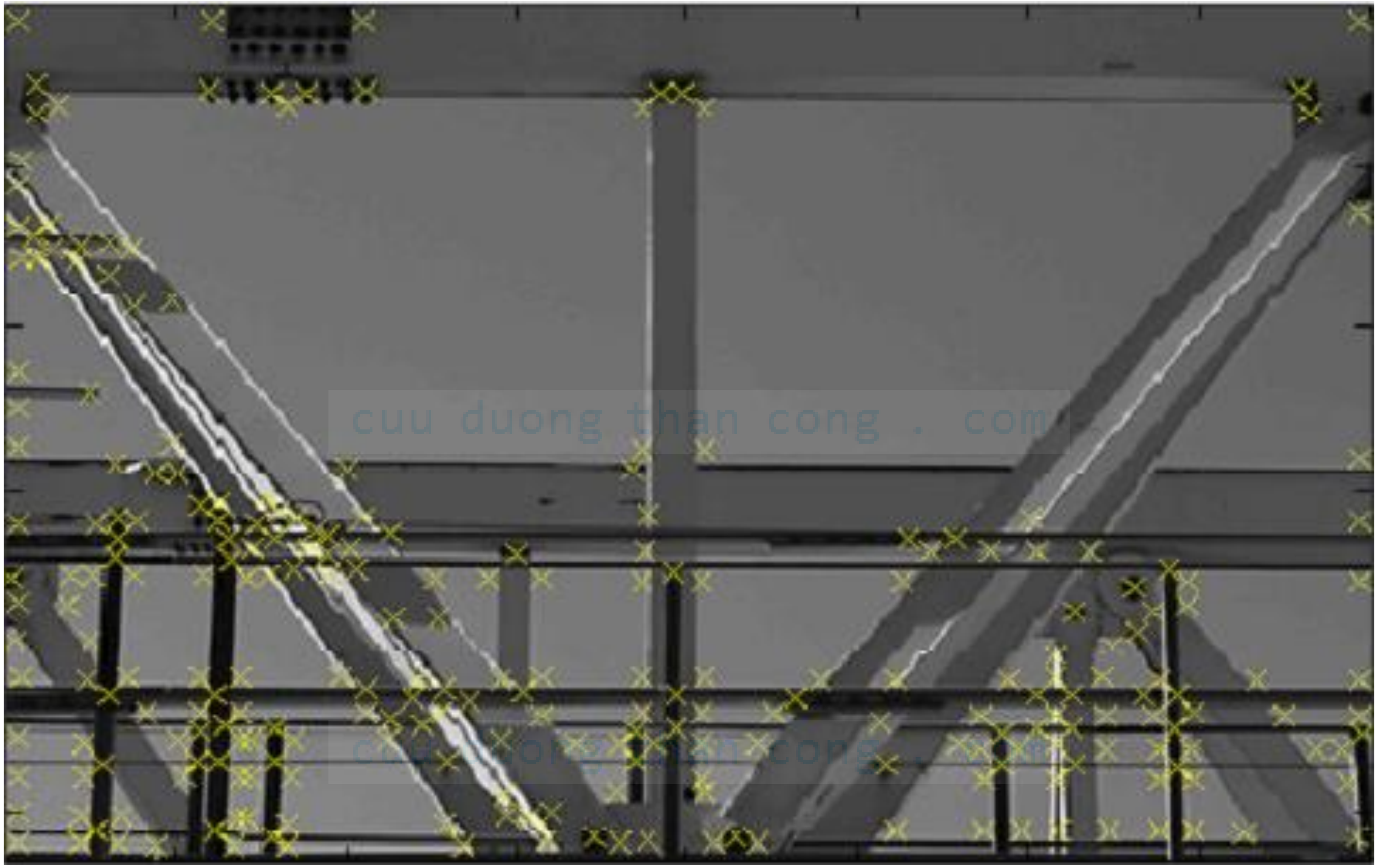
- Searches for image locations which have strong change in gradient along both the orthogonal direction

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

$$\det H = I_{xx} I_{yy} - I_{xy}^2$$

- Perform a non-maximum suppression using a 3×3 window
- Consider points having higher value than its 8 neighbors
- Select points where $\det H > \theta$

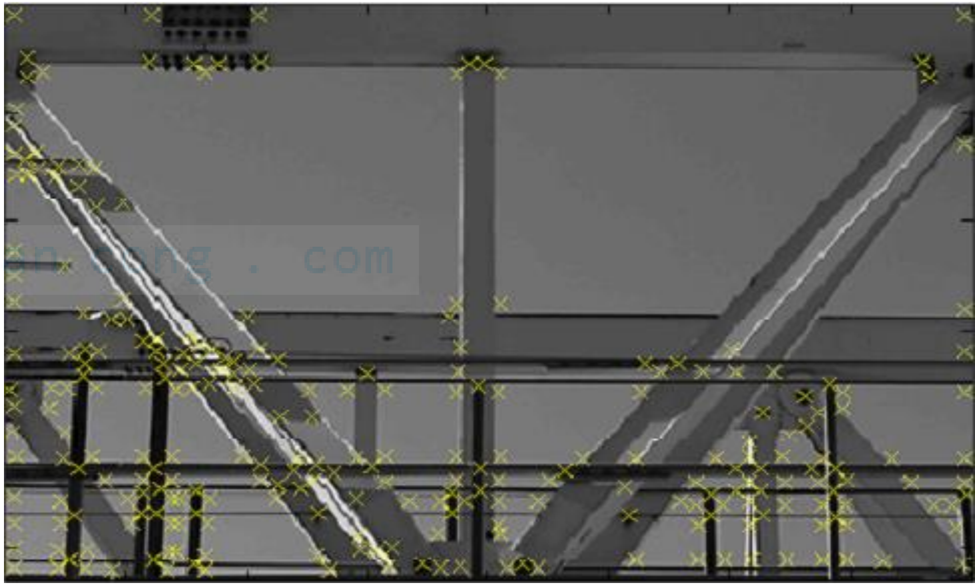
Hessian Corner Detector [Beaudet, 1978]



Effect: Responses mainly on corners and strongly textured areas

Harris Corner vs. Hessian Corner

Harris Corner Detector

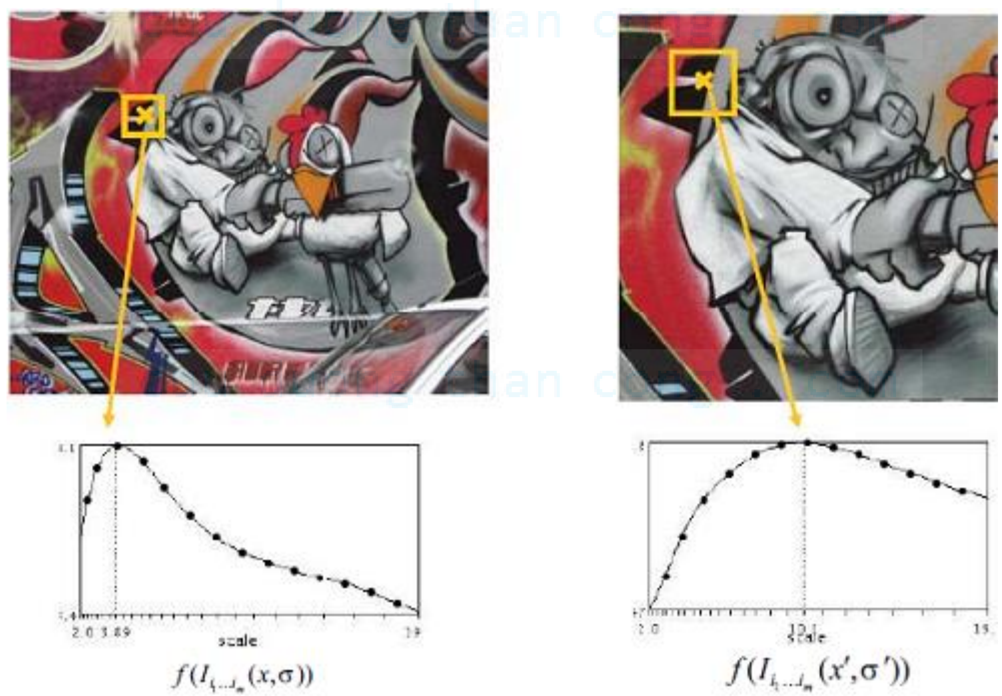


Hessian Corner Detector

Scale invariant region detection

- Both Harris and Hessian corner detectors are not scale invariant by nature
- Solution: use the concept of Scale Space

$$\left|LoG(x,\sigma_n)\right| = \sigma_n^2 \left|L_{xx}(x,\sigma_n) + L_{yy}(x,\sigma_n)\right|$$

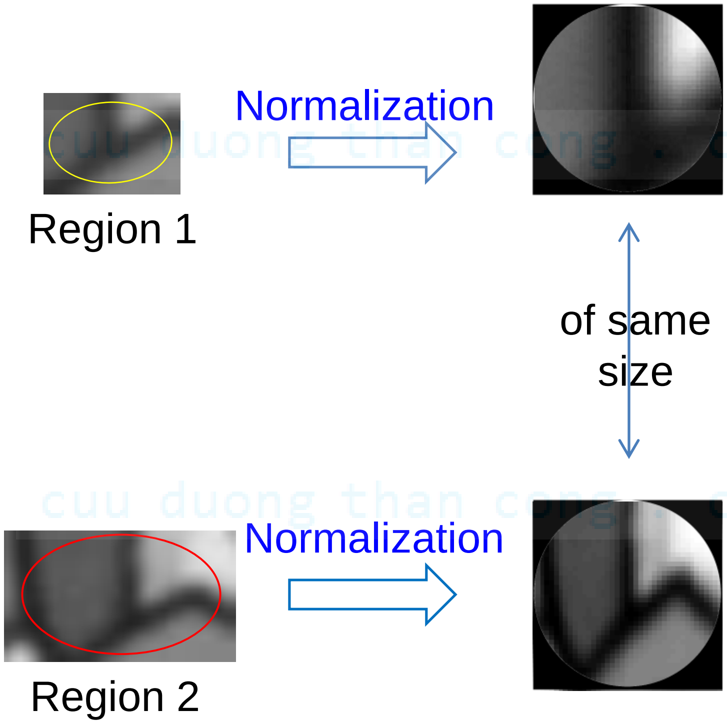


Affine invariant region detection

- Initialize with scale-invariant Harris points
- Estimation of the affine neighbourhood with the second moment matrix [Lindeberg'94]
- Apply affine neighbourhood estimation to the scale-invariant interest points [Mikolajczyk & Schmid'02, Schaffalitzky & Zisserman'02]

Normalization to a fixed scale

- Typically, regions are normalized to circular regions of uniform diameter of 41 pixels



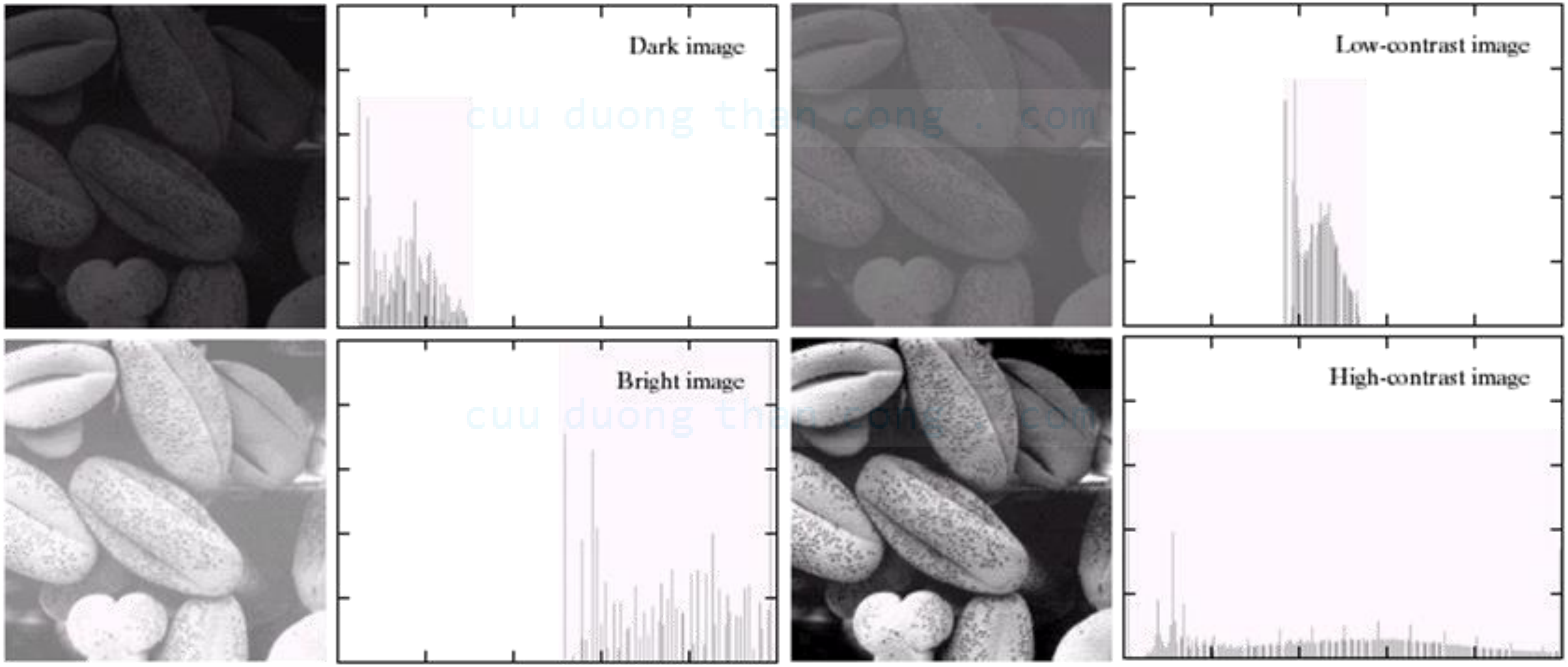
Can color be a good feature?

- No. Color is sensitive to noise and illumination changes

- Some color spaces (e.g. CIE-Lab, CIE-Luv) are more discriminative than the others (e.g. RGB) yet there is no formal proof for that

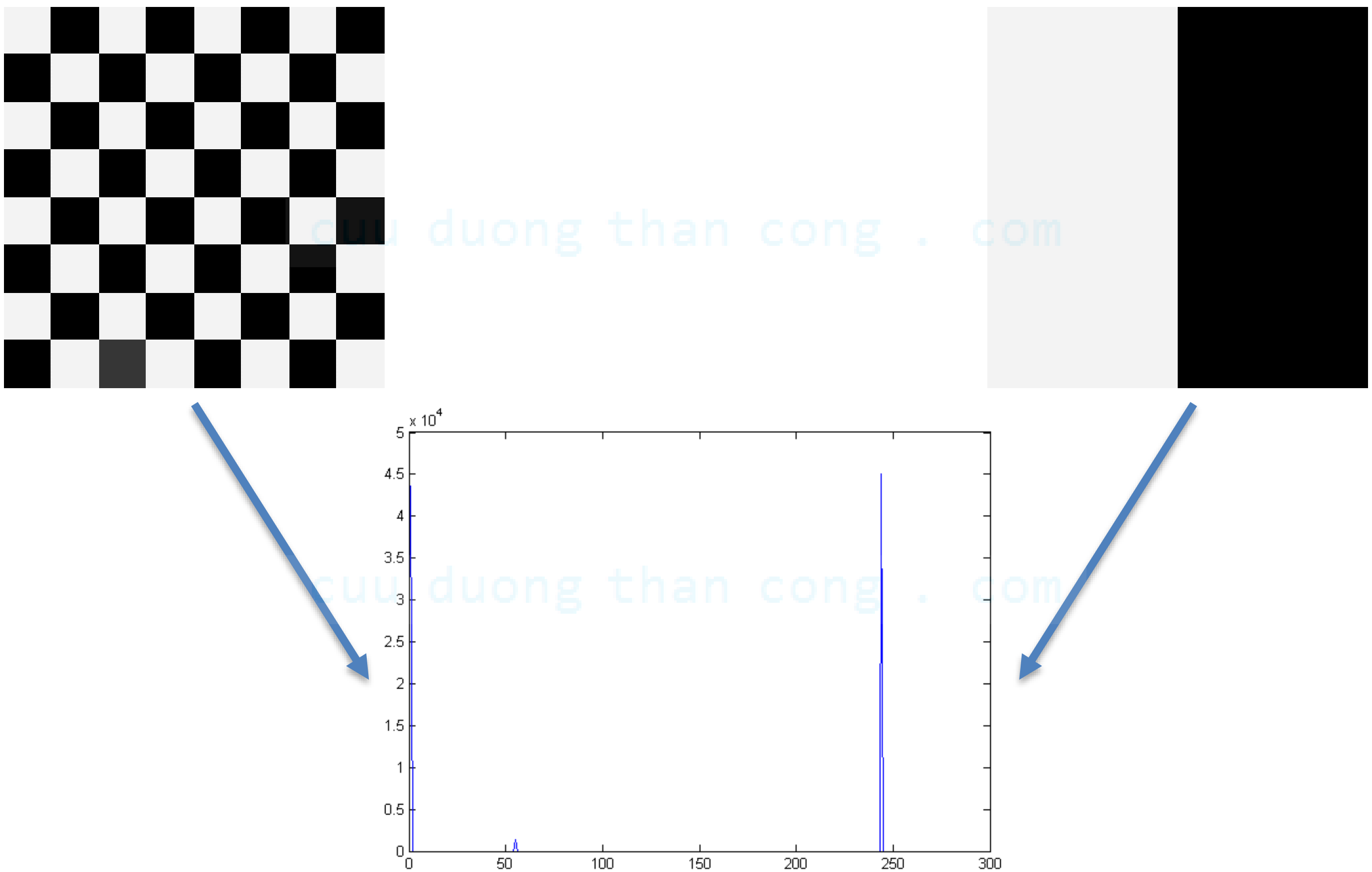
Can intensity be a good feature?

- Intensity itself is also sensitive to noise and illumination changes
- The relation between intensities may reduce the effect of illumination changes yet the effect of noise still remains



Can histogram be a good feature?

- Histogram, as well as some other statistical values, is a weak feature since it is too general



Types of local features

- Gradient-based local features

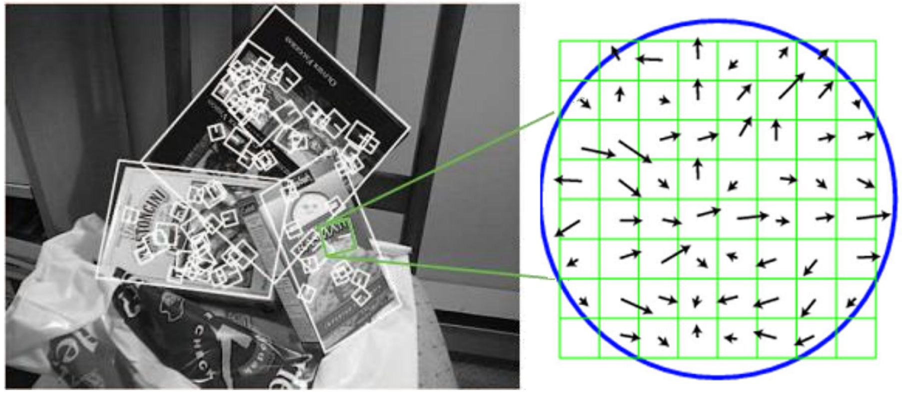
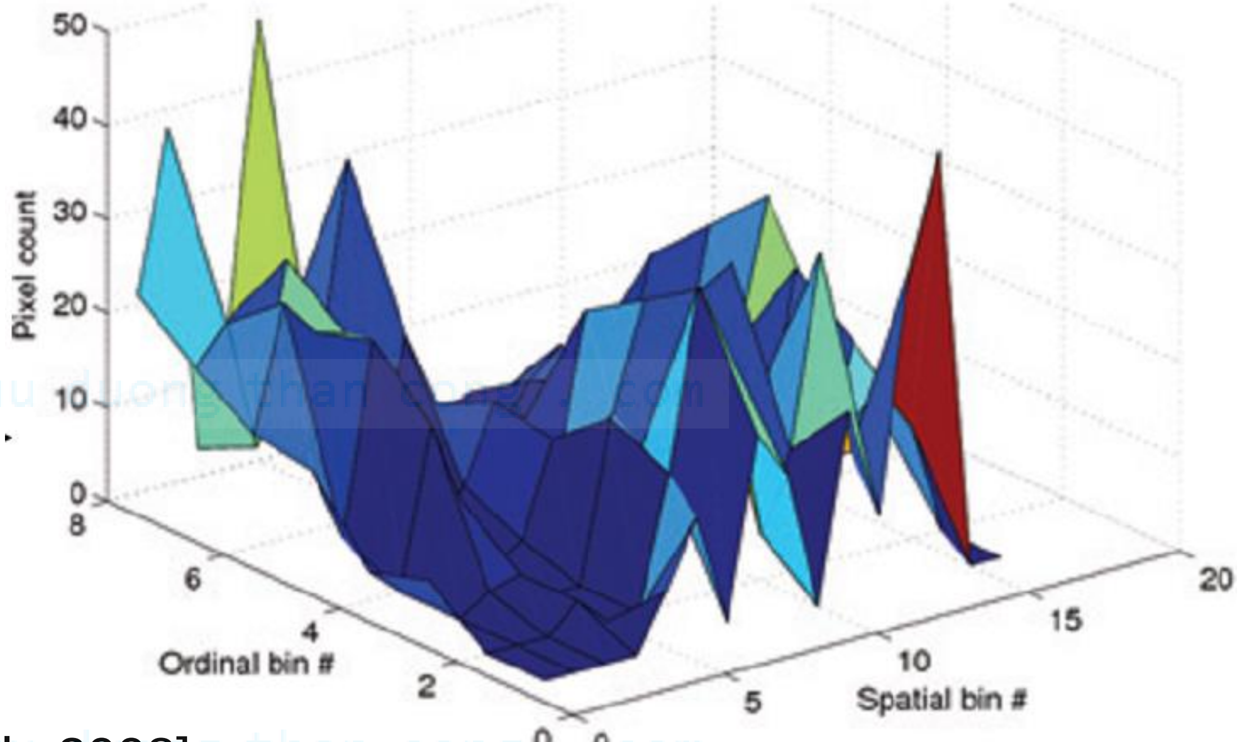


Image gradients

  - Scale-Invariant Feature Transform (SIFT) [Lowe, 2004]
    - PCA-SIFT [Ke and Sukthankar 2004]
    - Gradient location-orientation histogram (GLOH) [Mikolajczyk et al. 2005]
  - Histogram of Oriented Gradient (HOG) [Dalal and Triggs, 2005]
  - Pyramidal Histogram Of visual Words (PHOW) [Lazebniz et al., 2006]
  - Speed-Up Robust Feature (SURF) [Herbert Bay et al., 2006]
  - DAISY [Tola et al., 2010]
  - Others: Shape context, steerable filters, spin images

Types of local features

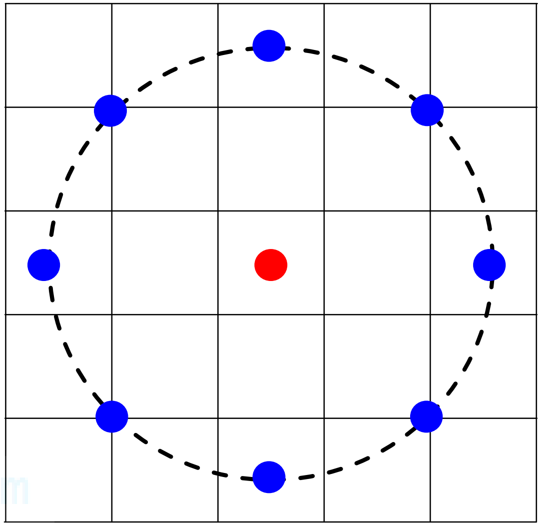
- Intensity-based local features



  - SMD [Gupta et al., 2008]
  - Ordinal spatial intensity distribution (OSID) [Tang et al., 2009]
  - Local intensity order pattern (LIOP) [Wang et al., 2011]

Types of local features

- LBP-based local features



  - Local Binary Pattern (LBP) [Ojala et al., 2002]
  - Center-Symmetric LBP (CS-LBP) [Heikkila et al., 2009]
  - Local Ternary Pattern (LTP) [Tan et al., 2010]
  - Multisupport region order-based gradient histogram (MROGH) [Bin et al., 2012]
  - And many other variants of LBP

Types of local features

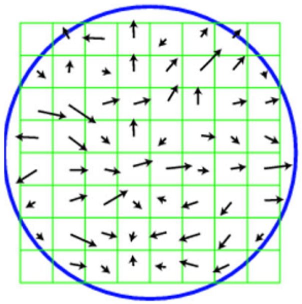
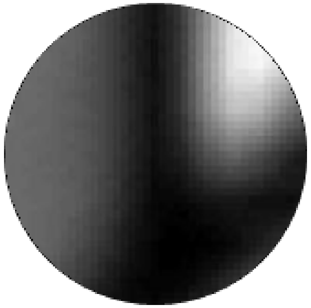
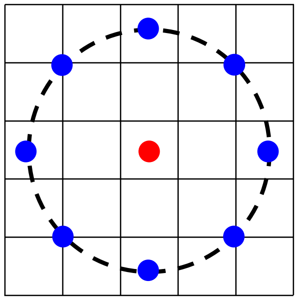


Image gradients
[Lowe04, Bin12, Tola10]



Grayscale intensity
[Wang09, Tang09, Gupta08]



Local Binary Pattern
[Wang09, Tang09, Gupta08]

- ✓ discriminative to directional changes
- ✗ computationally heavy

- ✓ invariant to illumination changes
- ✓ computationally light
- ✗ sensitive to noise

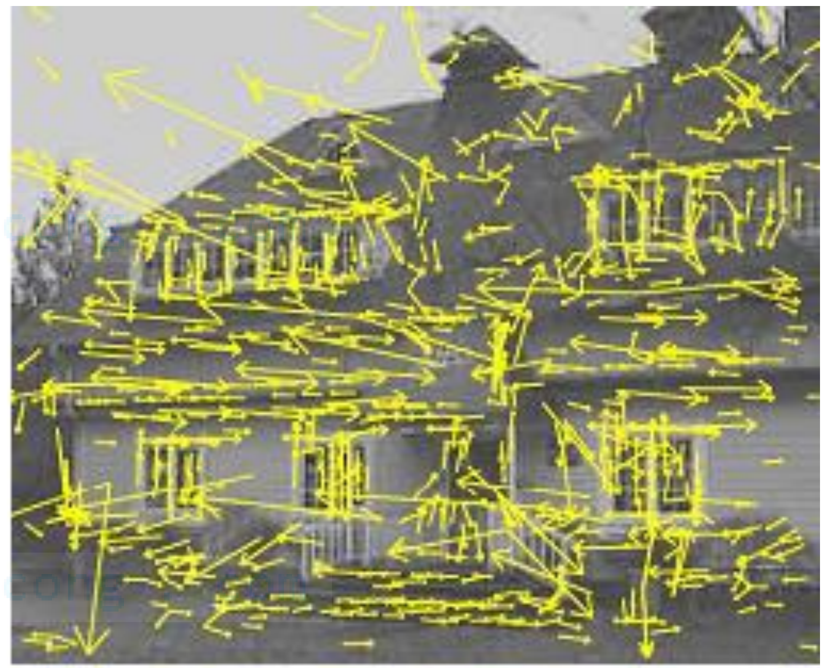
- ✓ invariant to illumination changes
- ✓ computationally light
- ✓ robust to noise
- ✗ high dimensionality

SIFT [Lowe, 2004]

- Step 1: Scale-space extrema Detection – Detect interesting points (invariant to scale and orientation) using DOG.
- Step 2: Keypoint Localization – Determine location and scale at each candidate location, and select them based on stability.
- Step 3: Orientation Estimation – Use local image gradients to assigned orientation to each localized keypoint. Preserve theta, scale and location for each feature.
- Step 4: Keypoint Descriptor – Extract local image gradients at selected scale around keypoint and form a representation invariant to local shape distortion and illumination

SIFT [Lowe, 2004]

- Step 1: Detect interesting points using Difference of Gaussians (DOG)



832 DOG extrema

SIFT [Lowe, 2004]

- Step 2: Accurate keypoint localization
  - Aim: reject low contrast points and points that lie on the edge
  - Reject low contrast points
    - Fit keypoint at $\underline{x}$ to nearby data using quadratic approximation

$$D(\underline{x}) = D + \frac{\partial D^T}{\partial \underline{x}}\underline{x} + \frac{1}{2}\underline{x}^T \frac{\partial^2 D^T}{\partial \underline{x}^2}\underline{x}$$

vwhere $D(x,\sigma) = [G(x,k\sigma) - G(x,\sigma)] * I(x)$

    - Calculate the local maxima of the fitted function $\{\underline{X} = (x,y,\sigma)\}$
    - Discard local minima (for contrast) $D(\underline{\hat{x}}) < 0.03$

$$\frac{\partial D}{\partial \underline{x}} = \frac{\partial\left[D + \frac{\partial D^T}{\partial \underline{x}}\underline{x} + \frac{1}{2}\underline{x}^T \frac{\partial^2 D^T}{\partial \underline{x}^2}\underline{x}\right]}{\partial \underline{x}} = 0 \Rightarrow \underline{\hat{x}} = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial \underline{x}}$$

SIFT [Lowe, 2004]

- Step 2: Accurate keypoint localization
  - Eliminating edge response: DOG gives strong response along edges $\Rightarrow$ Eliminate those responses
    - Solution: check "checkcornerness" of each keypoint
    - On the edge, one of principle curvatures is much bigger than another
    - High cornerness $\Leftrightarrow$ No dominant principle curvature component
    - Consider the concept of Hessian and Harris corner

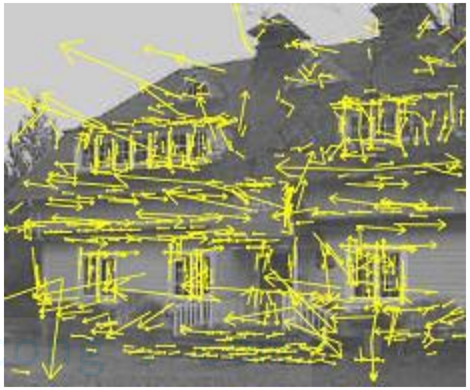
$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

$$\frac{\text{trace}\,(H)^2}{\det H} < \frac{(r+1)^2}{r}$$

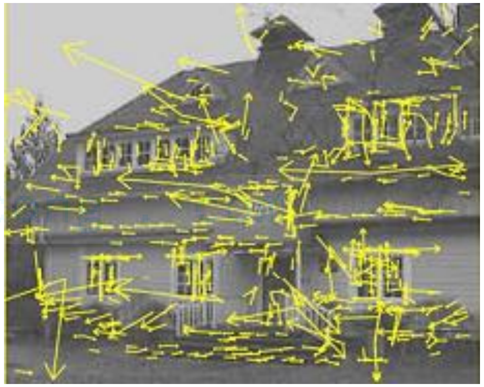
Discard points with response below threshold

SIFT [Lowe, 2004]

- Step 2: Accurate keypoint localization

  - 729 out of 832 are left after contrast thresholding

  - 536 out of 729 are left after cornerness thresholding

SIFT [Lowe, 2004]

- Step 3: Orientation assignment
  - Aim : Assign constant orientation to each keypoint based on local image property to obtain rotational invariance

To transform relative data accordingly

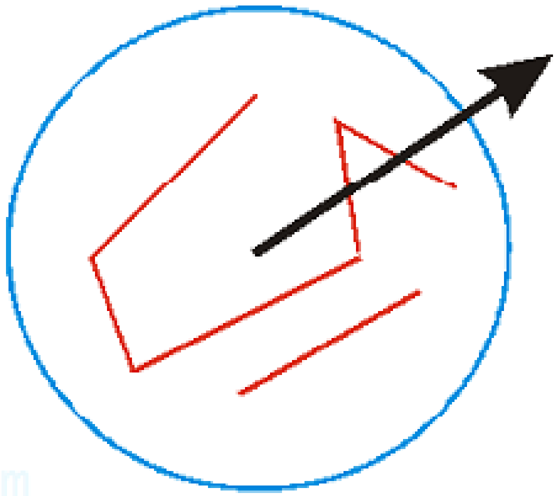
  - The magnitude and orientation of gradient of an image patch $I(x, y)$ at a particular scale is

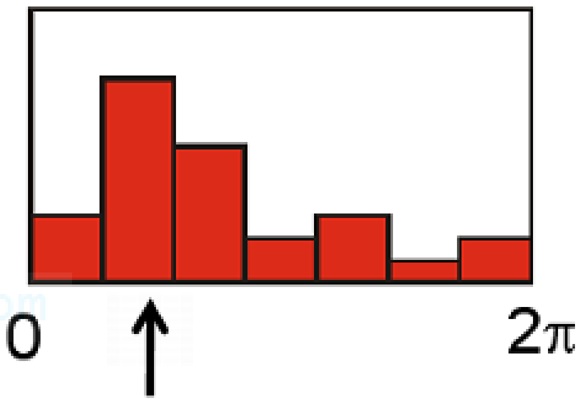
$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}$$

SIFT [Lowe, 2004]

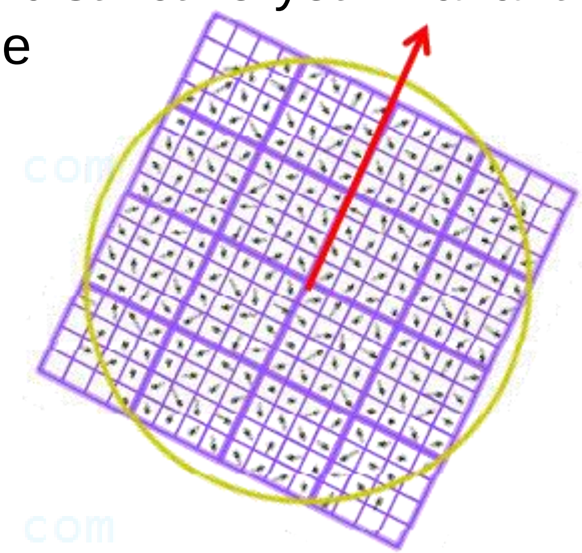
- Step 3: Orientation assignment
  - Create weighted (magnitude + Gaussian) histogram of local gradient directions computed at selected scale
  - Assign dominant orientation of the region as that of the peak of smoothed histogram
  - For multiple peaks create multiple key points





SIFT [Lowe, 2004]

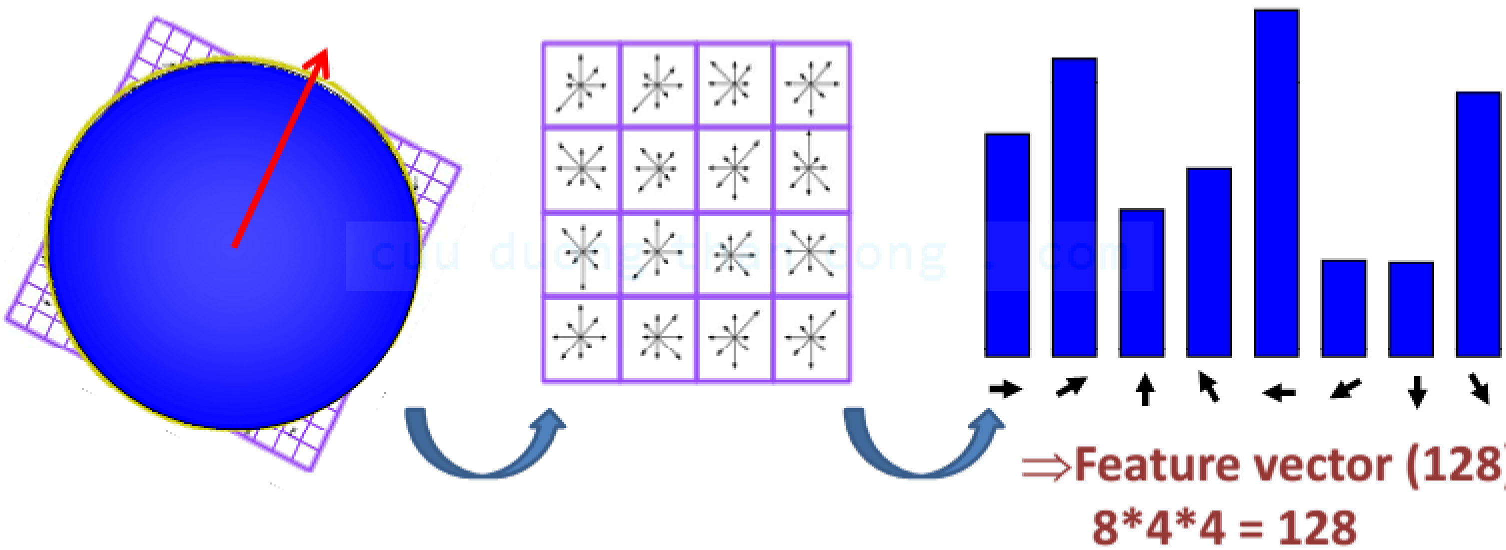
- Already obtained precise location, scale and orientation to each keypoint
- Step 4: Local image descriptor
  - Aim: Obtain local descriptor that is highly distinctive yet invariant to variation like illumination and affine change
  - Consider a rectangular grid 16×16 in the direction of the dominant orientation of the region.



  - Devide the region in to 4×4 sub-regions.
  - Consider a Gaussian filter above the region which gives higher weights to pixel closer to the center of the descriptor

SIFT [Lowe, 2004]

- Step 4: Local image descriptor
  - Create a 8-bin gradient histogram for each sub-region



  - Finally normalize 128-dim vector to make it illumination invariant

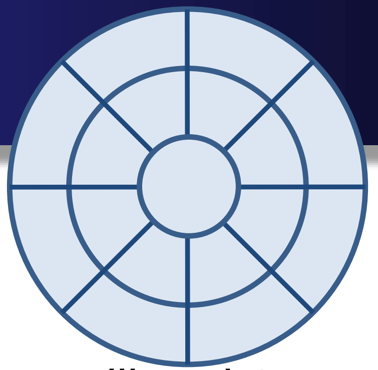
SIFT [Lowe, 2004]: Applications

- Object detection

SIFT [Lowe, 2004]: Applications

- Panorama

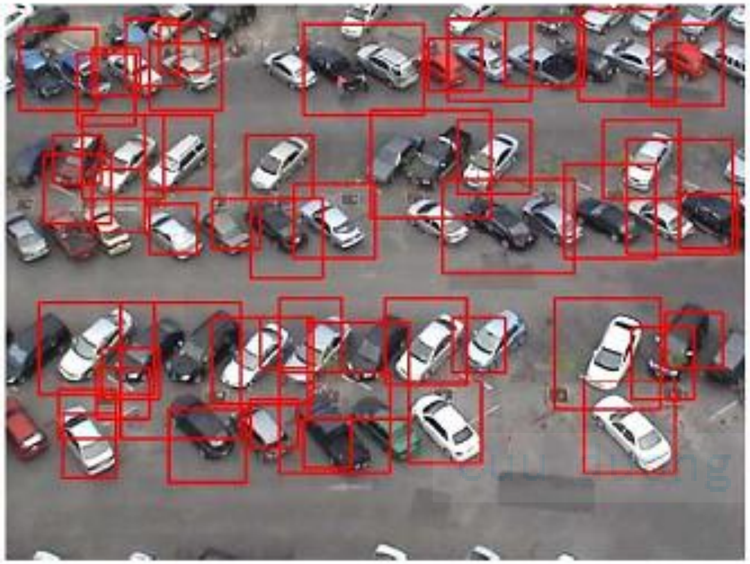
GLOH [Mikolajczyk et al., 2005]



- First 3 steps – same as SIFT
- Step 4 – Local image descriptor
  - Consider log-polar location grid with 3 different radii and 8 angular direction for two of them, in total 17 location bin
  - Form histogram of gradients having 16 bins
  - Form afeature vector of 272 dimension (17*16)
  - Perform dimensionality reduction and project the features to a 128 dimensional space

192 correct matches (yellow) and 208 false matches (blue)

Some other examples



SURF

PHOW

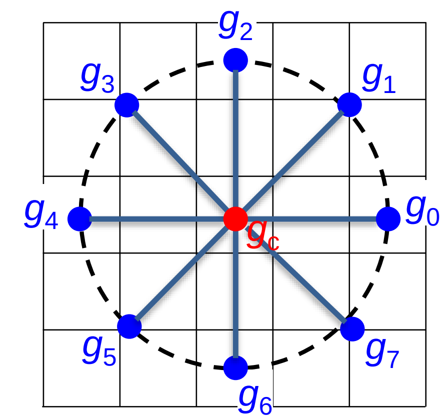
HOG

Local Binary Patterns (LBP)

- The **Local Binary Patterns** (LBP) is a texture operator that describes the local information around each pixel
- Ojala, T., Pietikainen, M., Maenpaa, T.: Multire solution gray-scale and rotation invariant texture classication with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)

Local Binary Patterns

- A texture operator that describes the local information around each pixel.

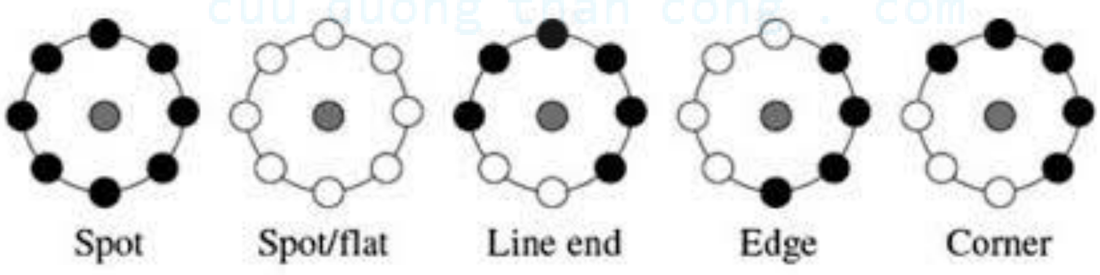


$$\mathrm{LBP}_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)\, 2^p,$$

$$s(z) = \begin{cases} 1 & z \geq 0 \\ 0 & otherwise \end{cases}$$

R: radius of the neighborhood, P: number of neighbors

g_c, g_p : the gray value of the center pixel and of p^{th}neighboring pixels



Local Binary Patterns

Input image

gray values in the
$n \times n$ neighborhood

137	140	143
144	140	139
132	135	136

Output image

Local Binary Patterns: Application

Texture analysis [Ojala02]

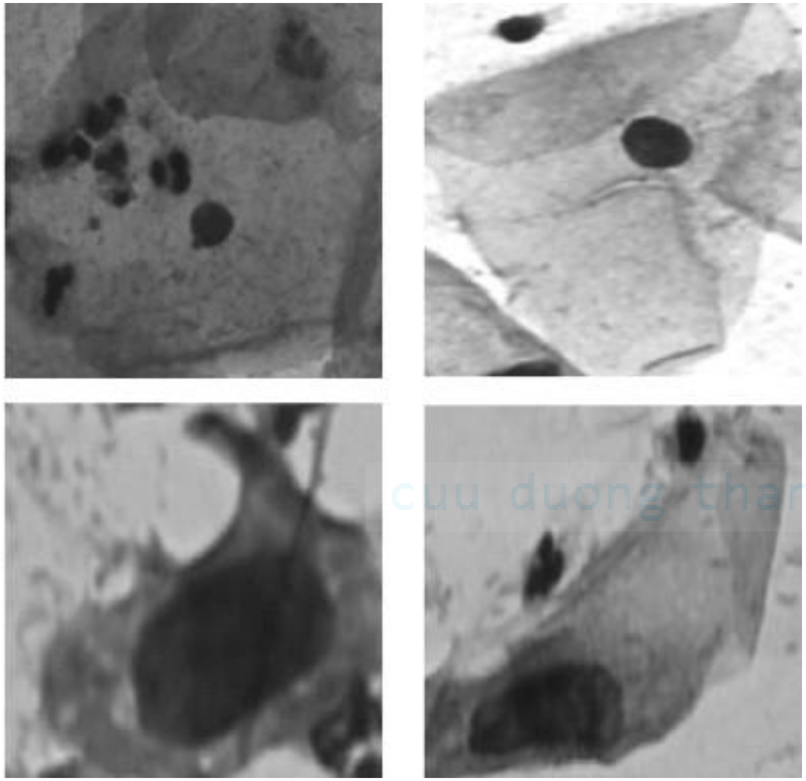
Face recognition [Tan10]

Image matching [Heikkila09]

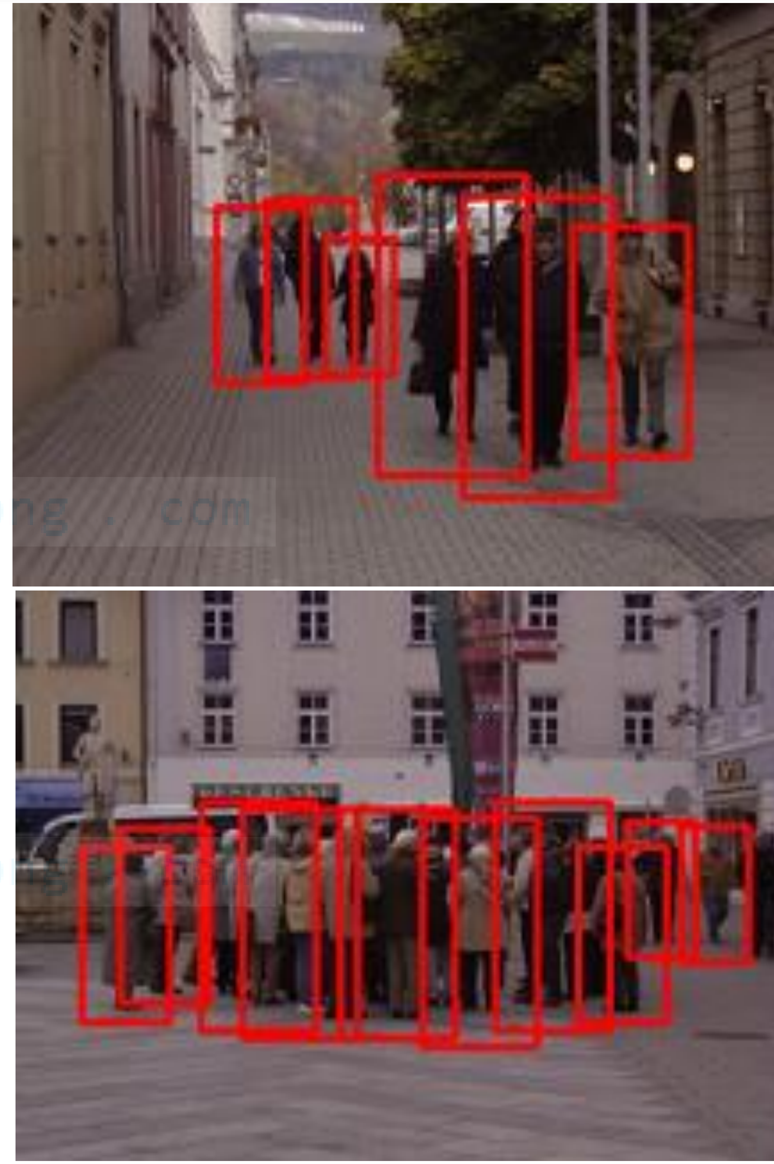
Local Binary Patterns: Application

Normal Class

Abnormal Class



Biomedical image [Nanni10]



Pedestrian detection [Wang09]

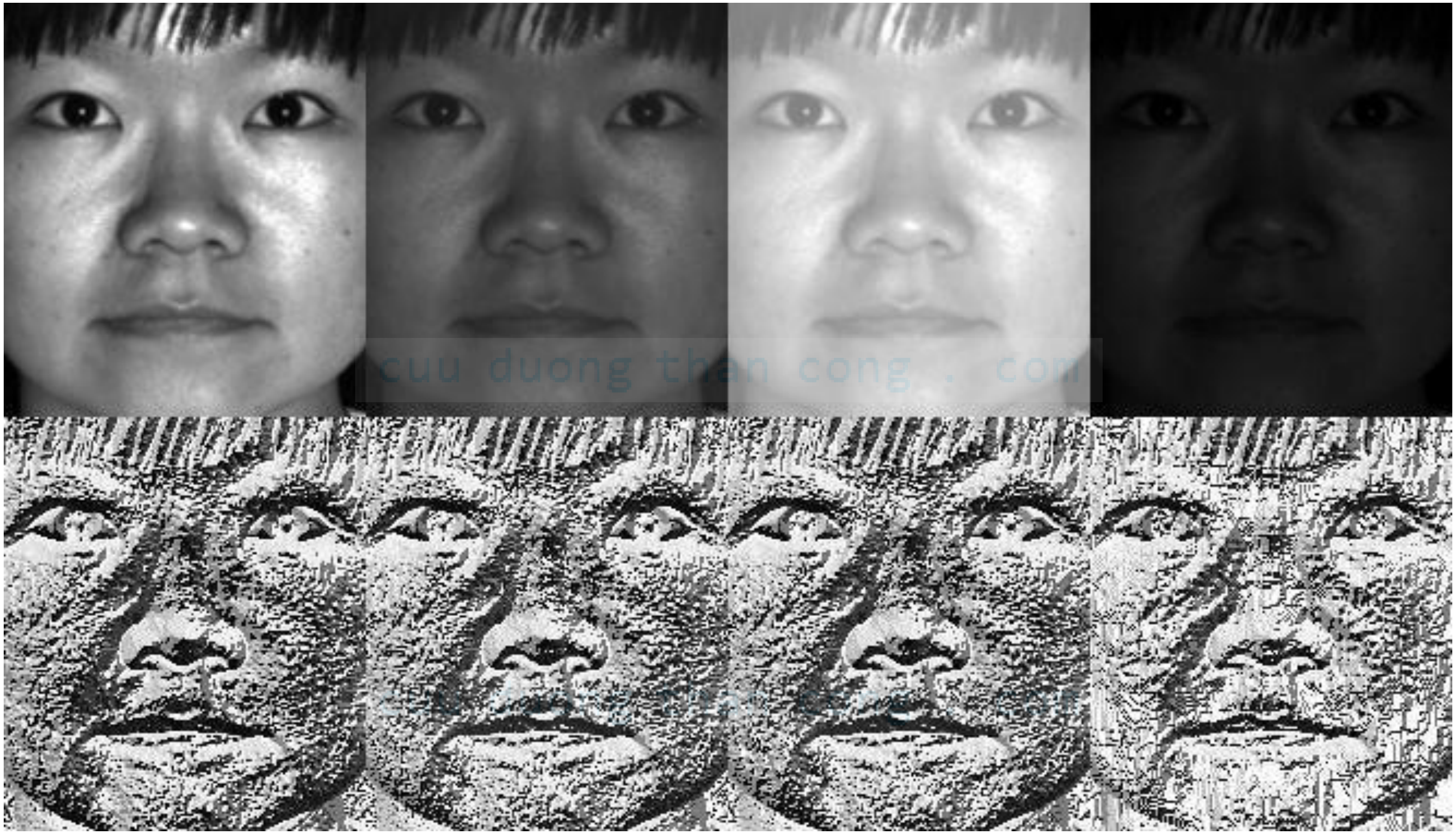
Local Binary Patterns: Application

Background subtraction [Liao09]

Properties of LBP

1. Invariant to any monotonic gray-level transformation
2. Nonparametric method
   - Require no assumptions about the underlying distribution
3. Highly discriminative against illumination changes
4. The operator is intuitive and computationally simple
5. The LBP code is quantized by its nature

Properties of LBP



Drawbacks of LBP

- Thresholding function $s(g_p - g_c)$
  - Unstable on noisy or near-uniform regions
  - Fail to deal with image details whose $g_p - g_c$ are of the same sign yet different magnitudes.

$$s(g_p - g_c) = \begin{cases} 1 & g_p - g_c \geq 0 \\ 0 & otherwise \end{cases}$$

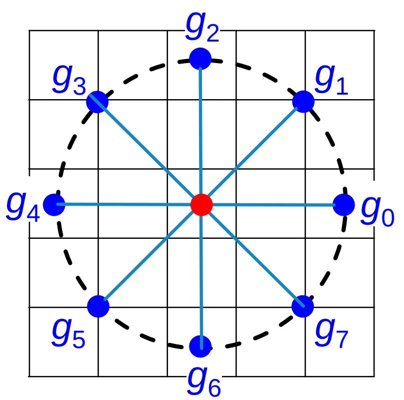
$$g_c = 29, g_p = 30 \Rightarrow s(g_c - g_p) = 0$$
$$g_c = 30, g_p = 30 \Rightarrow s(g_c - g_p) = 1$$

- The feature vectors are usually high dimensional.
  - $\text{LBP}_{8,\,R}$ has 2^8 (256) dimensions

Center-Symmetric LBP

- Center-Symmetric Local Binary Pattern (CS-LBP) compares center-symmetric pairs of pixels

$$CS - \mathrm{LBP}_{P,R}(x,y) = \sum_{p=0}^{P/2-1} s\left(g_p - g_{p+\left(\frac{P}{2}\right)}\right) 2^p,$$

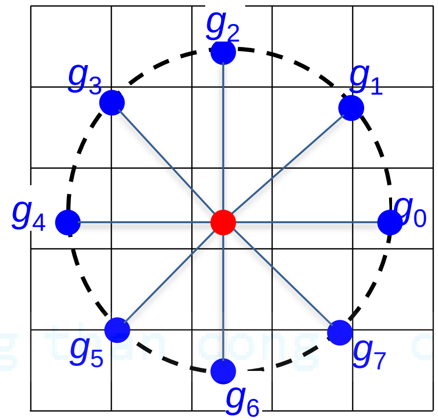
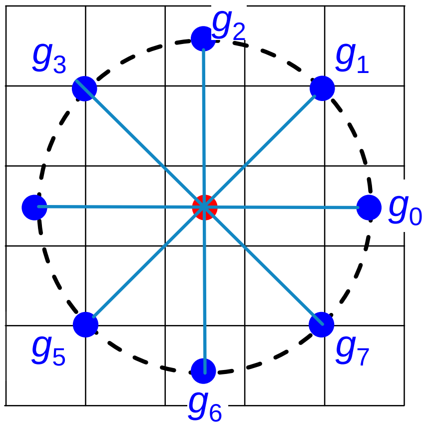


$$s(z) = \begin{cases} 1 & z > T \\ 0 & otherwise \end{cases}$$

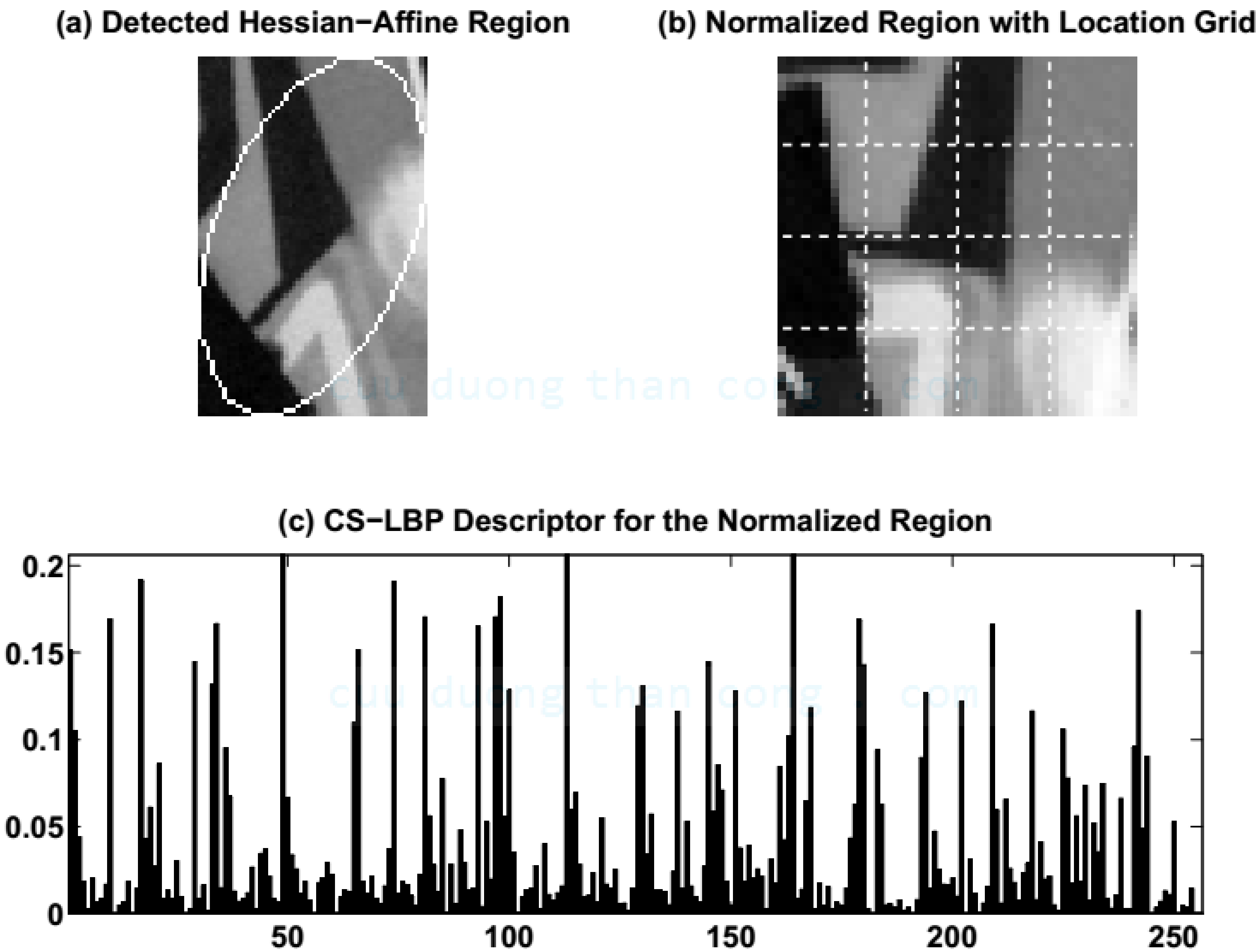
R: radius of the neighborhood, P: number of neighbors

$g_p, g_{p+\left(\frac{P}{2}\right)}$: gray values of the center-symmetric pair of pixels

LBP vs. CS-LBP

	$LBP_{8,R}$	$CS\text{-}LBP_{8,R}$
Operator design		
Number of neighbors	8	8
Number of comparisons	8	4
Number of binary encodings	$2^8 = 256$	$2^4 = 16$
Thresholding scheme	$s(z) = s(g_p - g_c) = \begin{cases} 1 & z \geq 0 \\ 0 & otherwise \end{cases}$	$s(z) = s(g_p - g_{p+P/2}) = \begin{cases} 1 & z > T \\ 0 & otherwise \end{cases}$

The CS-LBP Descriptor



LBP vs. CS-LBP vs. SIFT

- Heikkilä, Marko, Matti Pietikäinen, and Cordelia Schmid. "Description of interest regions with local binary patterns." *Pattern recognition* 42.3 (2009): 425-436.

https://pdfs.semanticscholar.org/3696/34f497852e05d5e72b12874e2a3db2d3945f.pdf

Section 8.4

DISTANCE MEASURES FOR FEATURE MATCHING

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
  3. Find the one with min distance

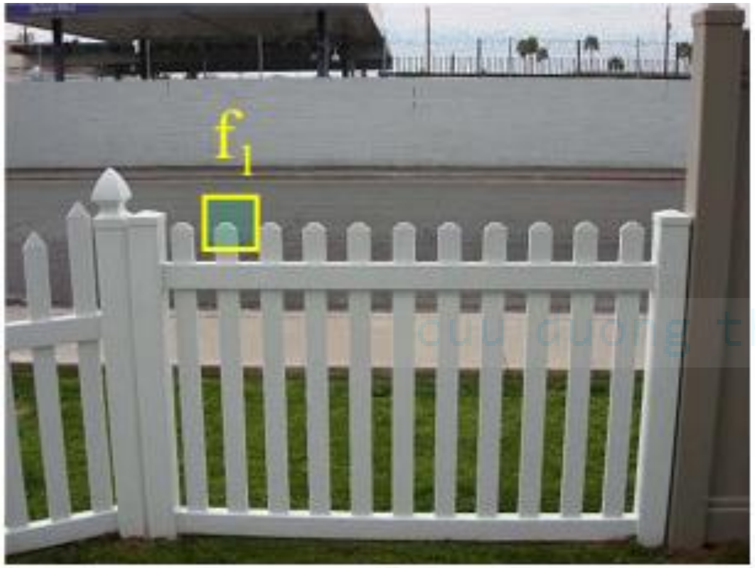
Feature distance: SSD

- How to define the similarity between 2 features, f_1 and f_2?
- Simple approach is $SSD(f_1, f_2)$
  - sum of square differences between entries of the two descriptors
  - doesn't provide a way to discard ambiguous (bad) matches

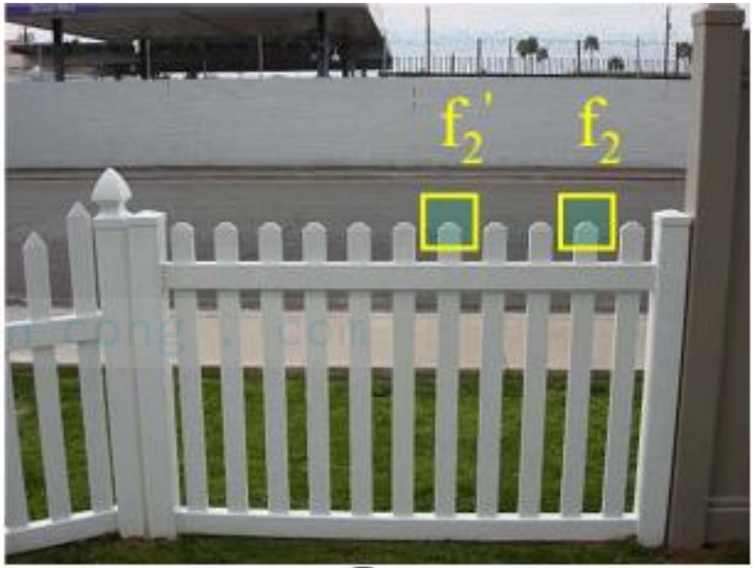
I_1 I_2

Feature distance: Ratio of SSDs

- How to define the similarity between 2 features, f_1 and f_2?
- Better approach: $\text{ratio distance} = \frac{SSD(f_1, f_2)}{SSD(f_1, f_2')}$
  - f_2 is best SSD match to f_1 in I_2, f_2' is 2nd best SSD match to f_1 in I_2
  - An ambiguous/bad match will have ratio close to 1
  - Look for unique matches which have low ratio

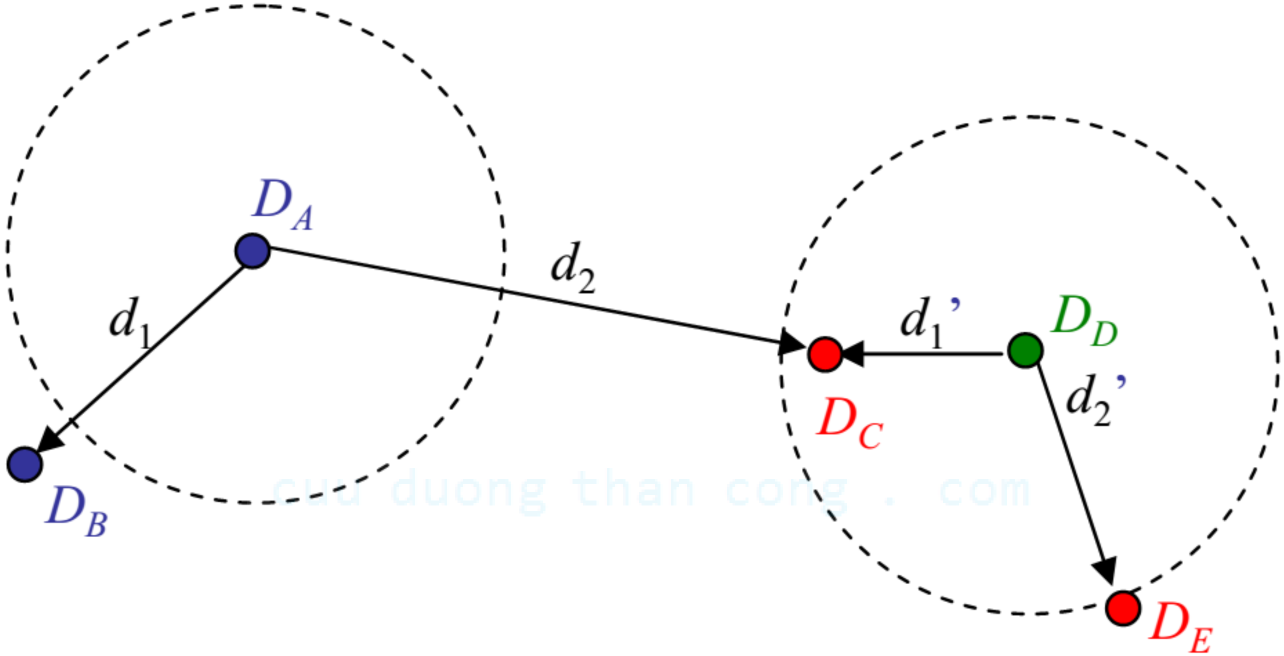


I_1



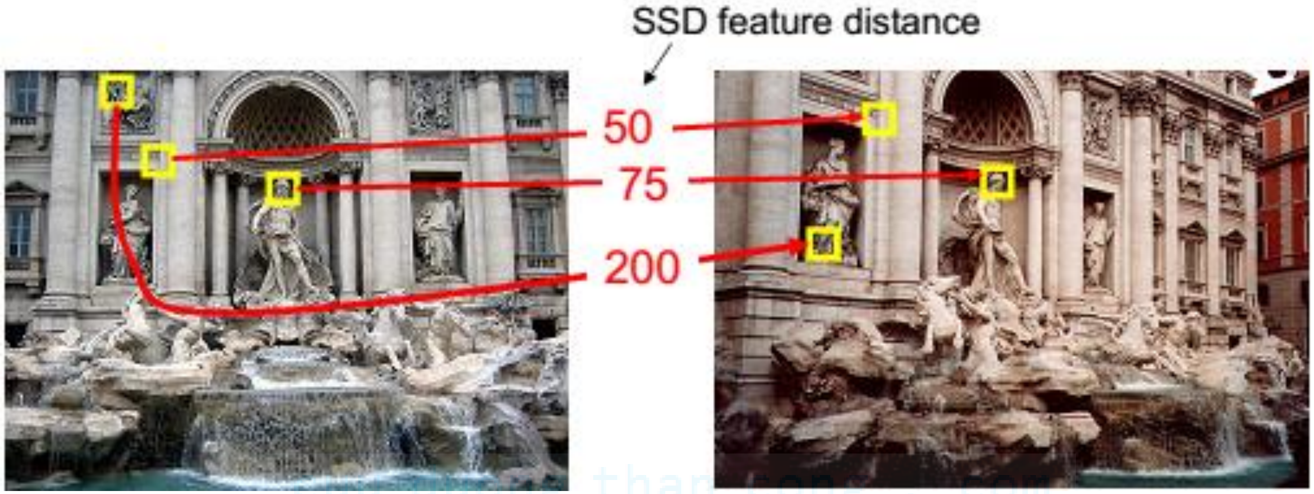
I_2

Feature distance



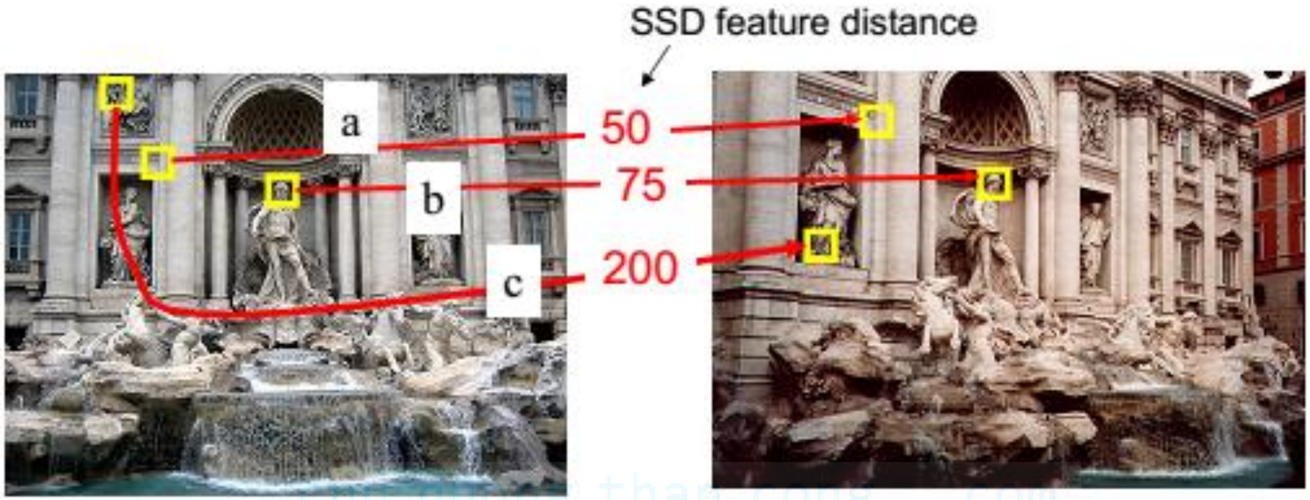
Fixed threshold, nearest neighbor, and nearest neighbor distance ratio matching. At a fixed distance threshold (dashed circles), descriptor D_A fails to match D_B and D_D incorrectly matches D_C and D_E. If we pick the nearest neighbor, D_A correctly matches $D_B\, but D_D$ incorrectly matches D_C. Use nearest neighbor distance ratio (NNDR) matching, the small NNDR d_1/d_2 correctly matches D_A with D_B, and the large NNDR d_1'/d_2' correctly rejects matches for D_D

Effect of threshold T



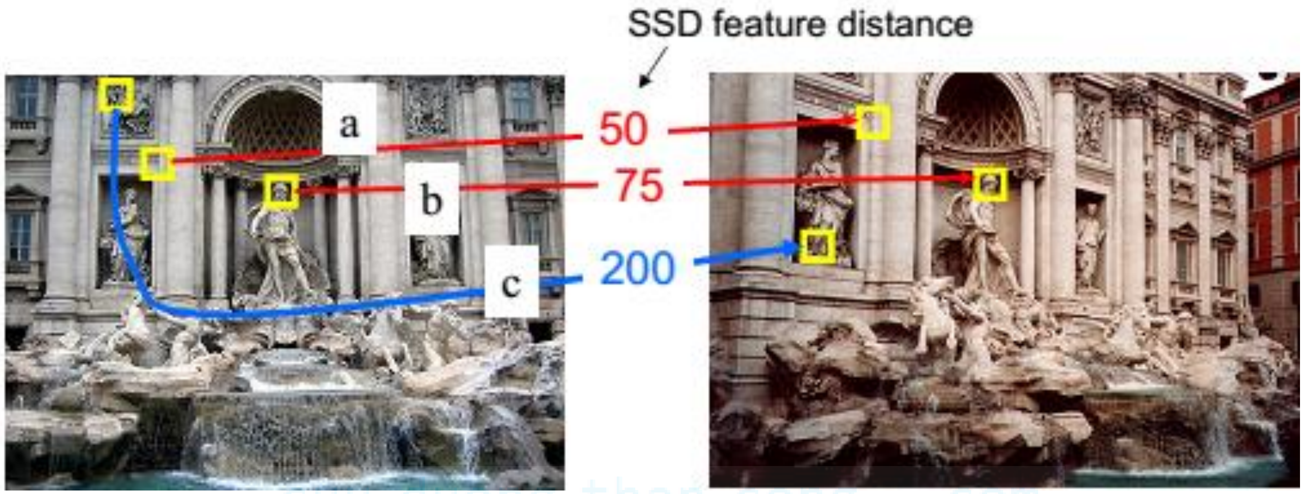
- Suppose we use SSD
- Small values are possible matches but how small?
- Decision rule: Accept match if $SSD < T$
  - where T is a threshold

Effect of threshold T



- Example: Large T
- $T = 250 \Rightarrow$ a, b, c are all accepted as matches
- a and b are true matches ("true positives")
  - they are actually matches
- c is a false match ("false positive")
  - actually not a match

Effect of threshold T



- Example: Smaller T
- T = 100 ⇒ only a and b are accepted as matches
- a and b are true matches ("true positives")
- c is no longer a "false positive" (it is a "true negative")

References

- Richard Szeliski, “Computer Vision: Algorithms and Applications”, 2011. Part 4.3.
- Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. International Journal of Computer Vision 65(1-2), 43–72 (2005)
- Lecture 06: Harris Corner Detector, CS486, Pennsylvania State University

  http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf
- Lecture Notes of Computer Vision CSE 576, Spring 2008, University of Washington

  https://courses.cs.washington.edu/courses/cse576/08sp/