HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Biometric authentication systems

Chapter 3: Feature extraction and Matching (2)

1

Content

- Edges:
  - Detection
  - Linking
- Feature extraction
  - Global features
  - Local features
  - Matching
- Applications

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

2

Local features vs Global features

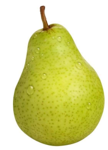
- Two types of features are extracted from the image:
  - local and global features (descriptors)
- **Global features**
  - Describe the image as a whole to the generalize the entire object
  - Include contour representations, shape descriptors, and texture features
    - Examples: Invariant Moments (Hu, Zernike), Histogram Oriented Gradients (HOG), PHOG, and Co-HOG,...
- **Local feature**:
  - the local features describe the image patches (key points in the image) of an object
  - represents the texture/color in an image patch
    - Examples: SIFT, SURF, LBP, BRISK, MSER and FREAK, …

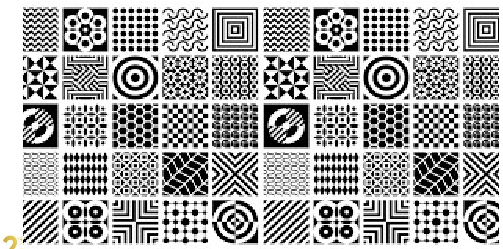
Feature extraction

- **Global features**
  - **Color / Shape / Texture**
- Local features

Global features?

How to distinguish these objects?





Types of features

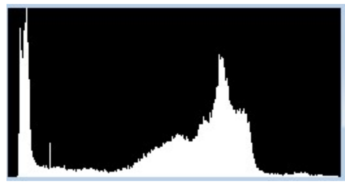
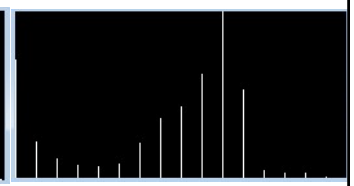
- Contour representation, Shape features
- Color descriptors
- Texture features

Color features

- Histogram

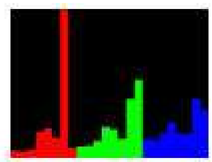
256 bins intensity histogram

16 bins intensity histogram


Image


Histogramme


Histobin(8,8,8)

Distance / Similarity

- L1 ou L2 (euclidian) distances are often used

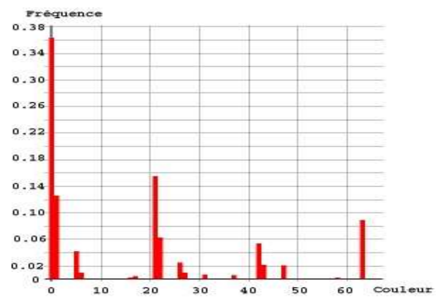
$$\mathrm{d}_{\mathrm{L1}}(\mathrm{H},\mathrm{G}) = \sum_{i=1}^{N} \left| h_i - g_i \right|$$

- Histogram intersection

$$\cap\ (\mathrm{H},\mathrm{G}) \ = \frac{\sum_i \min(h_i, g_i)}{\sum_i g_i}$$

Advantages of histogram

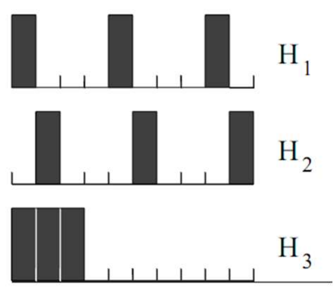
- Invariant to basic geometric transformations:
  - Rotation
  - Translation
  - Zoom (Scaling)



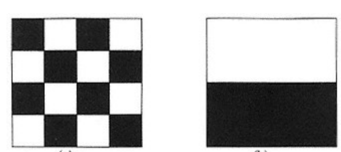
9

Some inconveniences

- The similarity between colors in adjacent colors (bin) is not taken into account



- The spatial distribution of pixel values is not considered: 2 different images may have the same histogram

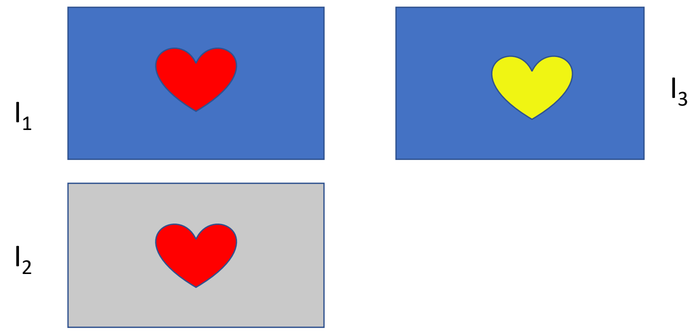


(a) (b)

10

Some inconveniences

- Background effect: ***d(I1,I2) ? d(I1, I3)***



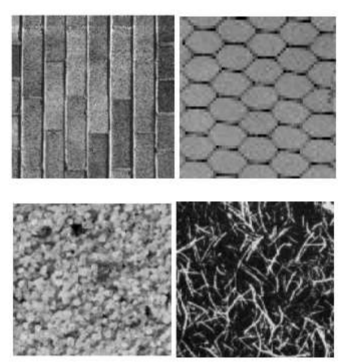
- Color representation dependency (color space), device dependency, …

Texture features

- A **texture** can be defined as
  - a region with variation of intensity
  - as a spatial organization of pixels

A texture can be **periodic** (a pattern that repeats itself) …

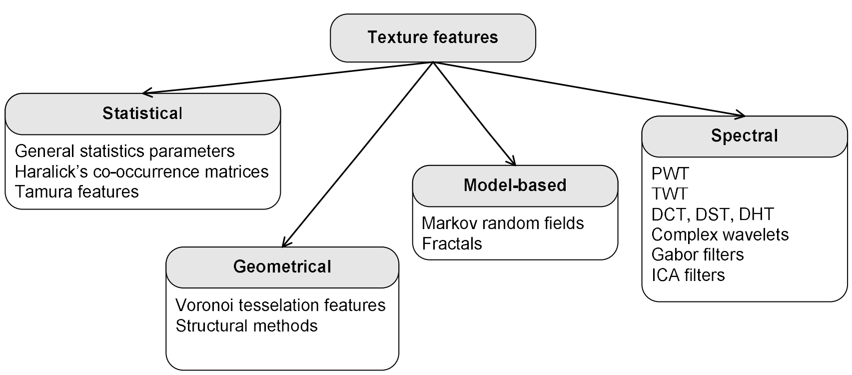
…or non-periodic (no pattern, more disorganized)



Texture features

- There are several methods for analyzing textures:
  - First order statistics
    - Statistics on histogram
  - Co-occurrence matrices
    - Searching patterns
  - Frequential analysis
    - Gabor filter
  - …
- The most difficult is to find a good representation (good parameters) for each texture

Texture features



First order statistics

- Histogram-based: mean, variance, skewness, kurtosis, energy, entropy, ...

$$\mu = \frac{1}{n}\sum_{i=0}^{H_g} h(i)$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=0}^{H_g} (h(i)-\mu)^2$$

$$s = \frac{1}{n\sigma^3}\sum_{i=0}^{H_g} (h(i)-\mu)^3$$

$$k = \frac{1}{n\sigma^4}\sum_{i=0}^{H_g} (h(i)-\mu)^4$$

$$E = \sum_{i=0}^{H_g} (P(i))^2$$

$$H = -\sum_{i=0}^{H_g} P(i)\log(P(i))$$

$$P(i) = \frac{h(i)}{n}$$

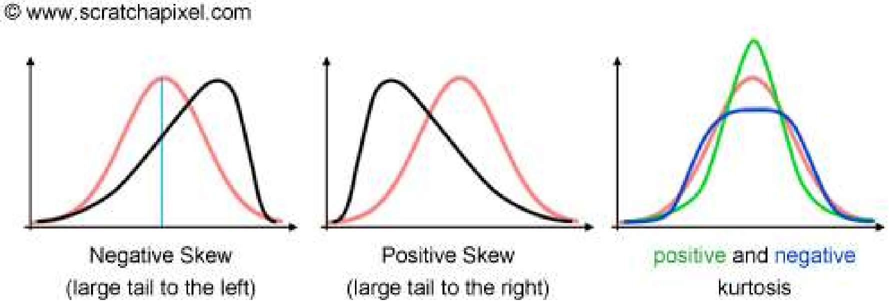
h(i) : số điểm ảnh ở mức xám i

15

First order statistics

- Skewness vs. kurtosis (red: normal distribution with mean = 5, variance = 4)

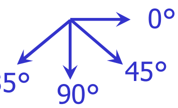
© www.scratchapixel.com



16

GLCM (Grey Level Co-occurence Matrices)

- The idea here is to identify **gray level that repeat themselves** given a **distance** and a **direction**
  - *Co-occurence matrices (Haralick)*
- **Matrix of size *Ng x Ng***
  - *Ng* is the number of gray level in the image (256x256)
  - We often reduce that number to 8x8, 16x16 or 32x32
- Many matrices, one for each distance and direction
  - **Distance** : 1, 2, 3 (,4, …)
  - **Direction** : 0°, 45°, 90°, 135° (, …)



- Processing time can be very long

GLCM

$$CM_{d,\beta}(c_i,c_j)=\frac{card(\{p_1,p_2 \mid I(p_1)=c_i, I(p_2)=c_j, N_{d,\beta}(p_1,p_2)=true\})}{card(\{p1,p2 \mid N_{d,\beta}(p_1,p_2)=true\})}$$

$N_{d,\beta}(p_1,p_2)=true$ p_2 is a neigbor of p_1 at a distance ***d*** direction β

GLCM

- Example on how to compute these matrices:

1	4	4	3
4	2	3	2
1	2	1	4
1	2	2	3

Image

	1	2	3	4
1	?	?	?	?
2	?	?	?	?
3	?	?	?	?
4	?	?	?	?

Matrix for distance=1 and direction=0°

We loop over the image and for each pair of pixels following the given distance and orientation, we increment the co-occurence matrix

GLCM

- Example on how to compute these matrices:

1	**4**	4	3
4	2	3	2
1	2	1	4
1	2	2	3

Image

	1	2	3	4
1	0	0	0	**1**
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Matrix for distance=1 and direction=0°

Pair of neighbor pixels (1,4)

GLCM

- Example on how to compute these matrices:

1	**4**	**4**	3
4	2	3	2
1	2	1	4
1	2	2	3

Image

	1	2	3	4
1	0	0	0	1
2	0	0	0	0
3	0	0	0	0
4	0	0	0	**1**

Matrix for distance=1 and direction=0°

Pair of neighbor pixels (4,4)

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

21

GLCM

- Example on how to compute these matrices:

1	**4**	**4**	**3**
4	**2**	**3**	**2**
1	2	1	4
1	2	2	3

Image

	1	2	3	4
1	0	0	0	**1**
2	0	0	**1**	0
3	0	**1**	0	0
4	0	**1**	**1**	**1**

Matrix for distance=1 and direction=0°

...and so on (after 2 lines)

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

22

GLCM

- Example on how to compute these matrices (final):

1	4	4	3
4	2	3	2
1	2	1	4
1	2	2	3

Image

	1	**2**	**3**	**4**
1	0	2	0	2
2	1	1	2	0
3	0	1	0	0
4	0	1	1	1

Matrix for distance=1 and direction=0°

	1	**2**	**3**	**4**
1	0	2	1	0
2	1	1	0	0
3	0	0	0	1
4	0	2	1	0

Matrix for distance=1 and direction=45°

...and so on for each matrix (several matrices at the end)

GLCM

- Most important/populer parameters computed from GLCM:

$$Energy = \sum_i \sum_j CM_d{}^2(i,j)$$

minimal when all elements are equal

$$entropy = -\sum_i \sum_j CM_d(i,j)\log(CM_d(i,j))$$

a measure of chaos, maximal when all elements are equal

$$contrast = \sum_i \sum_j (i-j)^2 CM_d(i,j)$$

small values when big elements are near the main diagonal

$$\mathrm{idm} = \sum_i \sum_j \frac{1}{1+(i-j)^2} CM_d(i,j)$$

idm (inverse differential moment) has small values when big elements are far from the main diagonal

GLCM

- Haralick features:
  - For each GLCM, we can compute up to 14 (13) parameters characterizing the texture, of which the most important : mean, variance, energy, inertia, entropy, inverse differential moment
  - Ref: http://haralick.org/journals/TexturalFeatures.pdf

Invariances

- Rotation?
  - Average on all directions
- Scaling?
  - Multi-resolutions

Texture features comparision

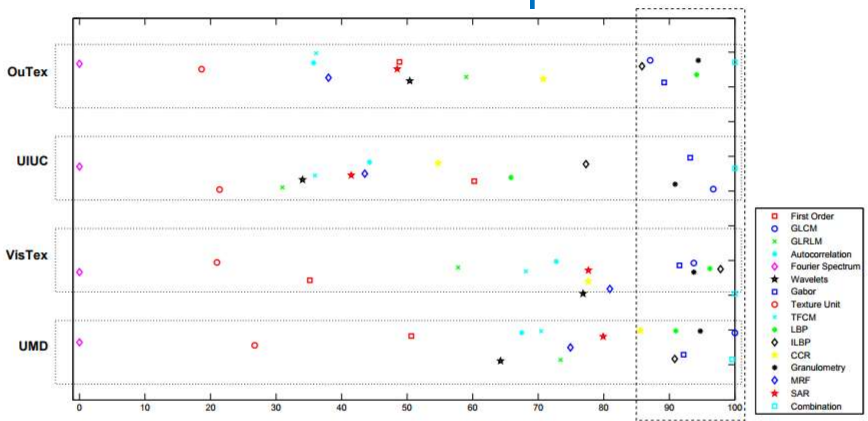


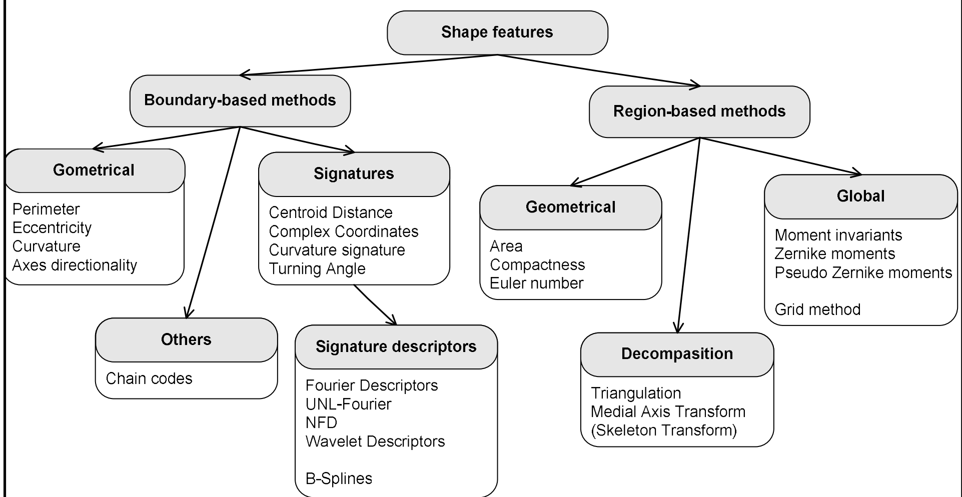
Figure 10. Comparison of results for all considered features on each data set. Each horizontal dotted box displays results for one data set. The x-axis shows normalized results (between 0 and 100) according to the minimum and maximum classification rates achieved for each data set. The vertical slashed box on the right-hand side selects the feature extraction methods that achieved the best results throughout all data sets.

Source : William Robson Schwartz et al. Evaluation of Feature Descriptors for Texture Classification – 2012 JEI

Shape features

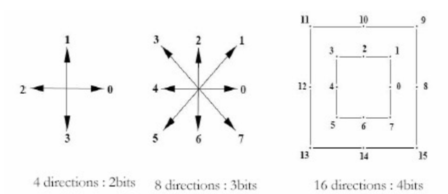
- Contour-based features
  - Chain coding, polygone approximation, geometric parameters, angular profile, surface, perimeter, …
- Region based:
  - Invariant moments, …

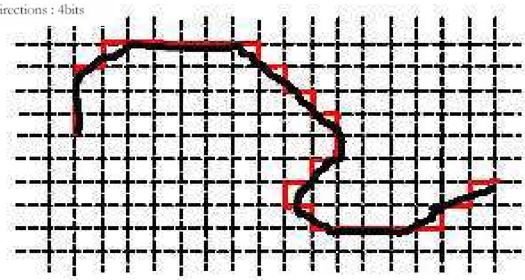
Shape features



29

Examples: Freeman chain coding

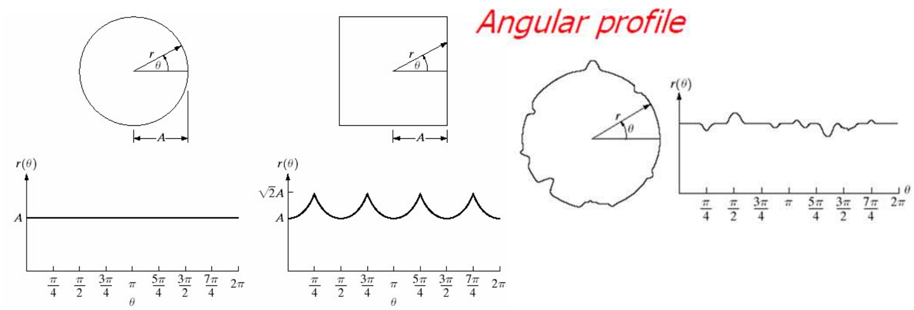


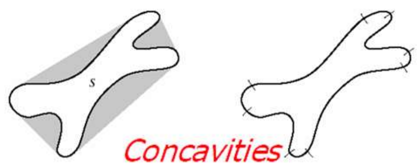


11101000000303030332323030000001010 34x2 = 68bits
221100007777654670000101 24x3 = 72bits
1098815156546788101 16x4 = 64bits

30

Examples: angular profile, …





31

Examples : Image moments

- Moment
$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$
$$M_{0,0} = area\ \ of\ the\ region\ D$$
$$(M_{0,1}, M_{1,0}) = centroid\ of\ D$$
- Central moments:
$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$
$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

Invariant to translation

32

Invariant moments (Hu's moments)

invariant to translation, scale, and rotation, and reflection

$$
\begin{aligned}
I_1 &= \eta_{20} + \eta_{02} \\
I_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
I_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
I_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
I_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} \\
&\quad - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
I_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2
\end{aligned}
$$

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}}$$

Change for image reflection

$$
\begin{aligned}
I_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} \\
&\quad + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} \\
&\quad + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

Examples : Hu's moments

6 images and their Hu Moments

id	Image	H[0]	H[1]	H[2]	H[3]	H[4]	H[5]	H[6]
K0	K	2.78871	6.50638	9.44249	9.84018	-19.593	-13.1205	19.6797
S0	S	2.67431	5.77446	9.90311	11.0016	-21.4722	-14.1102	22.0012
S1	S	2.67431	5.77446	9.90311	11.0016	-21.4722	-14.1102	22.0012
S2	S	2.65884	5.7358	9.66822	10.7427	-20.9914	-13.8694	21.3202
S3	S	2.66083	5.745	9.80616	10.8859	-21.2468	-13.9653	21.8214
S4	S	2.66083	5.745	9.80616	10.8859	-21.2468	-13.9653	-21.8214

https://www.learnopencv.com/wp-content/uploads/2018/12/HuMoments-Shape-Matching.png

Feature extraction

- Global features
- **Local features**
  - **Interest point detector**
  - Local descriptor

Why local features?

- Image matching: a challenging problem

Source: CS131 - Juan Carlos Niebles and Ranjay Krishna

Image matching

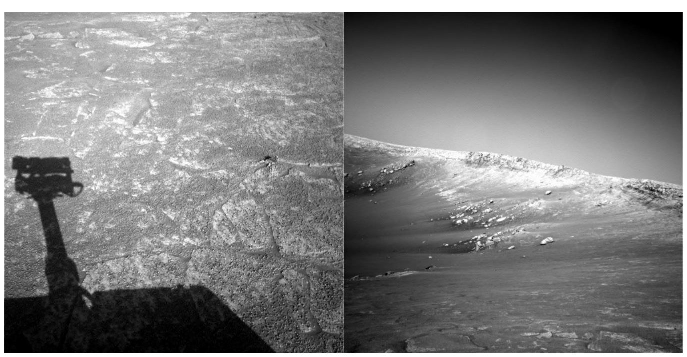
by swashford

by Diva Sian

by scgbt

Slide credit: Steve Seitz

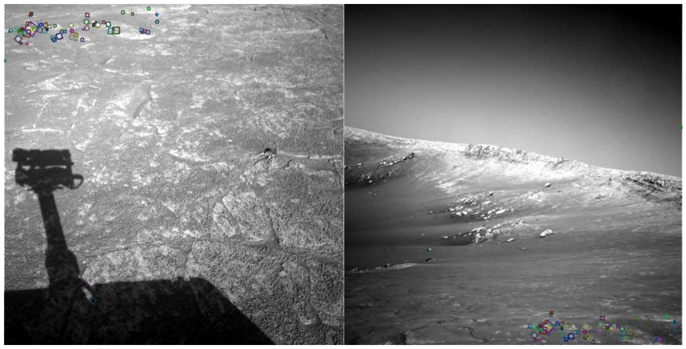
Harder Still?



NASA Mars Rover images

Slide credit: Steve Seitz

Answer Below (Look for tiny colored squares)



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

Slide credit: Steve Seitz

41

- Recognition of specific objects/scenes

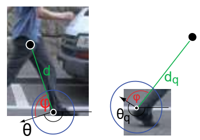
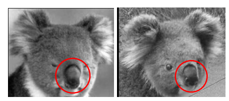
Sivic and Zisserman, 2003

D. Lowe 2002

42

Motivation for using local features

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

  - Articulation
    
  - Intra-category variations
    

Source: CS131 - Juan Carlos Niebles and Ranjay Krishna

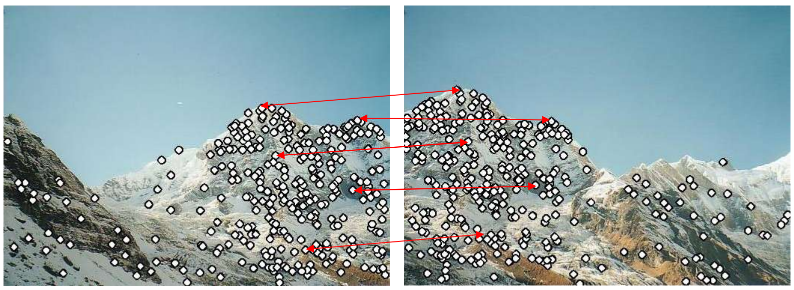
Local features and alignment

- We need to match (align) images
- Global methods sensitive to occlusion, lighting, parallax effects. So look for local features that match well.
- How would you do it by eye?


[Darya Frolova and Denis Simakov]

Local features and alignment

- Detect feature points in both images
- Find corresponding pairs



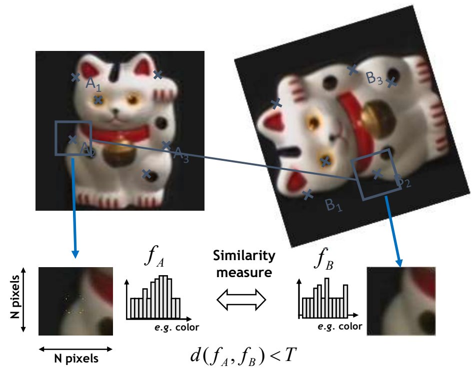
[Darya Frolova and Denis Simakov]

Local features and alignment

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

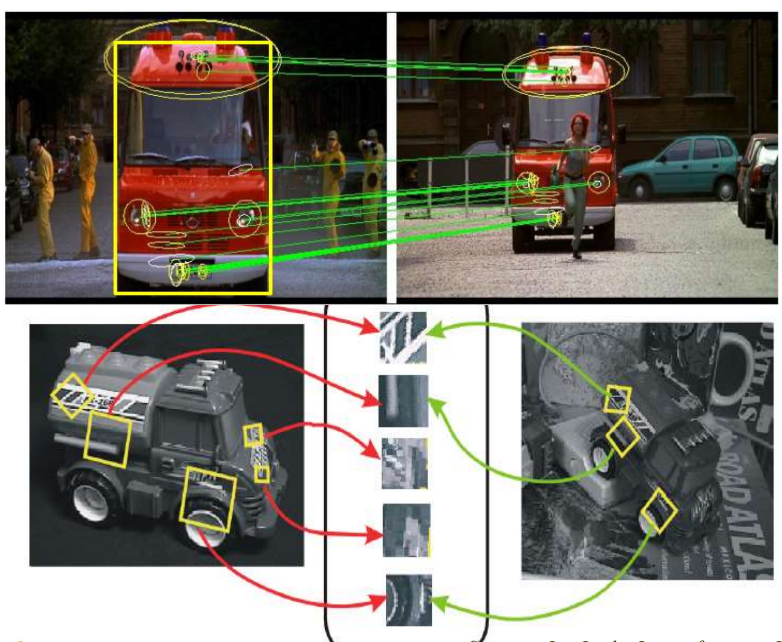
[Darya Frolova and Denis Simakov]

Local features for image matching



1. Find a set of distinctive key- points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a localdescriptor from the normalized region
5. Match local descriptors

Slide credit: Bastian Leibe



Source : Jim Little, Lowe: features, UBC.

Local features

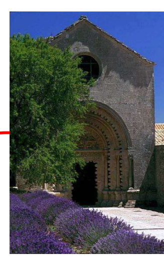
- Objectifs:
  - Look for similar objects/regions
  - Partial query

Look for pictures that contain sunflowers

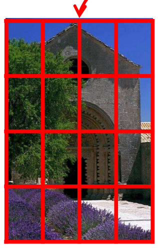
- Solution:
  - Describing **local regions**
  - Adding **spatial constraints** if need

Local feature extraction

- Local features: how to determine image patches / local regions



Dividing into patches with regular grid



Without knowledge about image content

Image segmentation

Keypoint detection

Based on the content of image

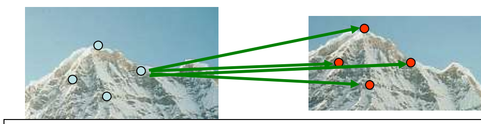
Common Requirements

- Problem 1:
  - Detect the same point *independently* in both images

No chance to match!

We need a repeatable detector!

- Problem 2:
  - For each point correctly recognize the corresponding one



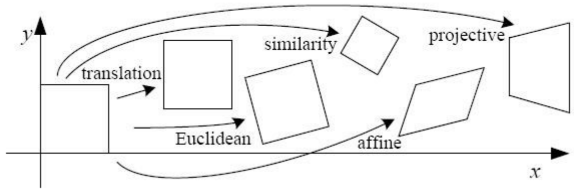
We need a reliable and distinctive descriptor!

Slide credit: Darya Frolova, Denis Simakov

Invariance: Geometric Transformations

Slide credit: Steve Seitz

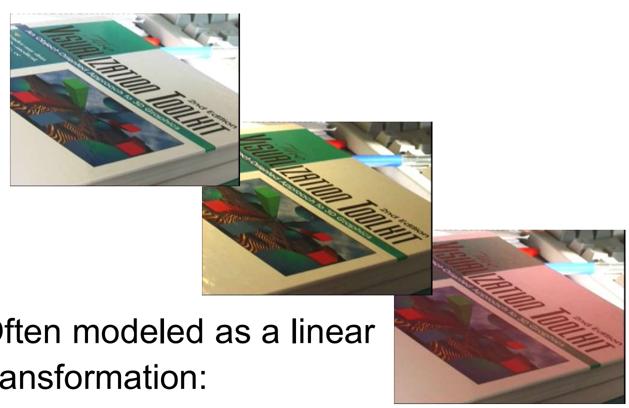
Invariance: Geometric Transformations



Levels of Geometric Invariance

Slide credit: Steve Seitz

Invariance: Photometric Transformations



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

Slide credit: Tinne Tuytelaars

Requirements

- Region extraction needs to be repeatable and accurate
  - Invariant to translation, rotation, scale changes
  - Robust or covariant to out-of-plane (≈affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- Locality: Features are local, therefore robust to occlusion and clutter
- Quantity: We need a sufficient number of regions to cover the object
- Distinctiveness: The regions should contain “interesting” structure
- Efficiency: Close to real-time performance

Main questions

- **Where** will the interest points come from?
  - What are salient features that we'll *detect* in multiple views?
- How **to *describe*** a local region?
- How to **establish *correspondences***, i.e., compute matches?

Feature extraction

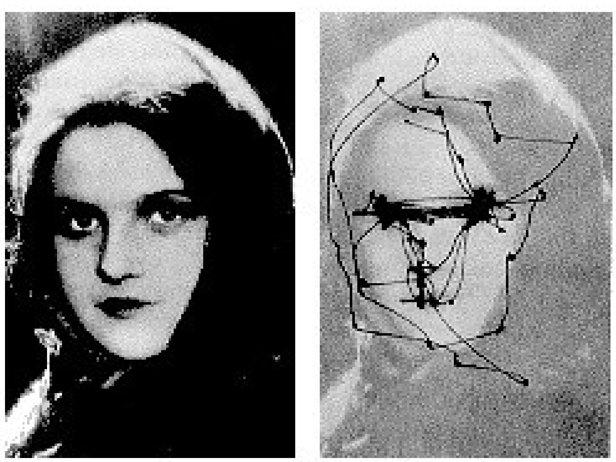
- Global features
- **Local features**
  - **Interest point detector**
  - Local descriptor
  - Matching

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

57

57

Interest points: why and where?



Yarbus eye tracking

Source : Derek Hoiem, Computer Vision, University of Illinois.

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

58

58

Interest points: why and where?

- Where will the interest points come from?

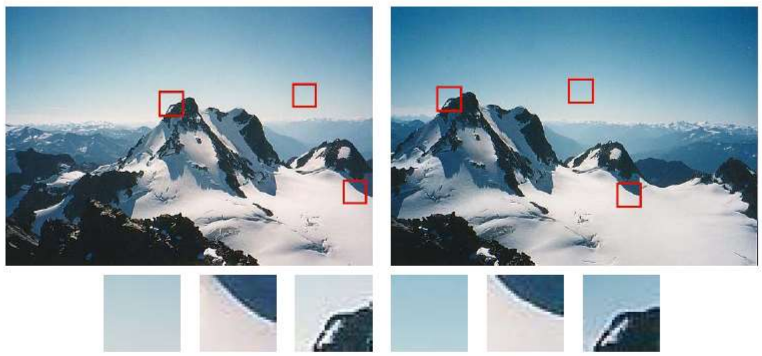


Figure 4.3: *Image pairs with extracted patches below. Notice how some patches can be localized or matched with higher accuracy than others.*

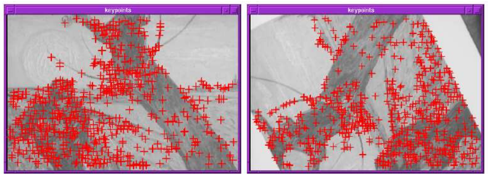
Keypoint Localization

- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

  ⇒ *Look for two-dimensional signal changes*

Slide credit: Bastian Leibe

Finding Corners



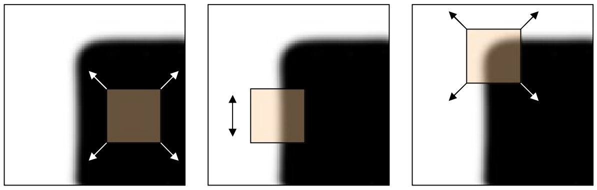
- Key property:
  - In the region around a corner, image gradient has two or more dominant directions
- Corners are *repeatable* and *distinctive*

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*, 1988.

Slide credit: Svetlana Lazebnik

Corners as distinctive interest points

- Design criteria
  - We should easily recognize the point by looking through a small window (*locality*)
  - Shifting the window in *any direction* should give *a large change* in intensity (*good localization*)



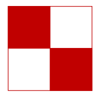
"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Slide credit: Alyosha Efros

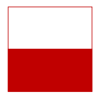
Corners versus edges



$\sum I_x^2 \longrightarrow$ Large

$\sum I_y^2 \longrightarrow$ Large

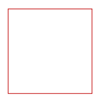
Corner



$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Large

Edge



$\sum I_x^2 \longrightarrow$ Small

$\sum I_y^2 \longrightarrow$ Small

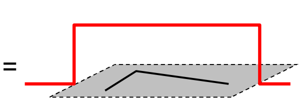
Nothing

Harris detector formulation

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

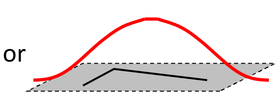
$$E(u,v)=\sum_{x,y}\underbrace{w(x,y)}_{\text{Window function}}\Big[\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}}-\underbrace{I(x,y)}_{\text{Intensity}}\Big]^2$$

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

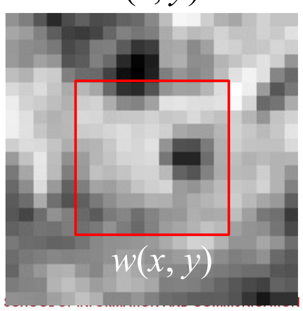
Source: R. Szeliski

Corner Detection by Auto-correlation

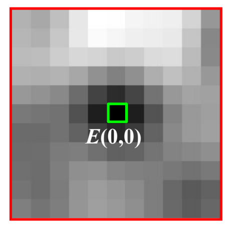
Change in appearance of window *w*(*x*,*y*) for shift [*u,v*]:

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

$I(x, y)$



$E(u, v)$



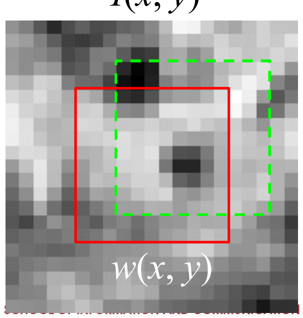
66

Corner Detection by Auto-correlation

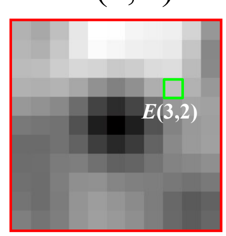
Change in appearance of window *w*(*x*,*y*) for shift [*u,v*]:

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

$I(x, y)$



$E(u, v)$



67

Corner Detection by Auto-correlation

Change in appearance of window *w*(*x*,*y*) for shift [*u*,*v*]:

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

We want to discover how E behaves for small shifts

But this is **very slow** to compute naively.
O(window_width^2 * shift_range^2 * image_width^2)

O(11^2 * 11^2 * 600^2) = 5.2 billion of these

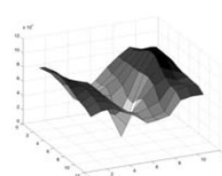
Corner Detection by Auto-correlation

Change in appearance of window *w*(*x*,*y*) for shift [*u*,*v*]:

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

We want to discover how E behaves for small shifts

But we know the response in *E* that we are looking for – strong peak. ➔ Approximation



Harris detector formulation

- This measure of change can be approximated by:

$$E(u,v) \approx [u \ \ v] \ M \begin{bmatrix} u \\ v \end{bmatrix}$$

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

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

Gradient with respect to *x*, times gradient with respect to *y*

Sum over image region – the area we are checking for corner

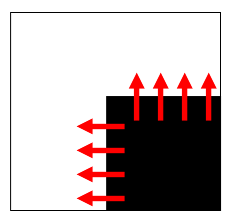
$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y]$$

Slide credit: Rick Szeliski

What does this matrix reveal?

- First, let's consider an axis-aligned corner:

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



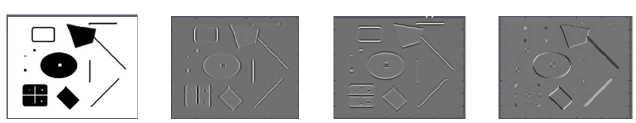


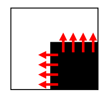
Image I I_x I_y $I_x I_y$

Slide credit: David Jacobs

What does this matrix reveal?

- First, let's consider an axis-aligned corner:

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



- This means:
  - 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.
- What if we have a corner that is not aligned with the image axes?

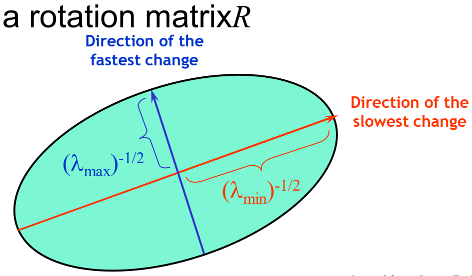
Slide credit: David Jacobs

General case

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

(Eigenvalue decomposition)

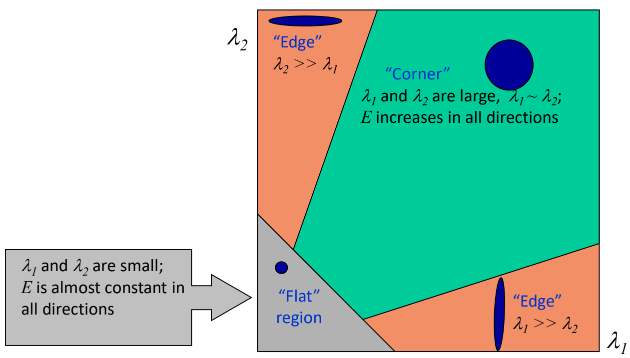
- We can visualize M as an ellipse with axis lengths determined by the eigenvalues and orientation determined by a rotation matrix R



adapted from Darya Frolova, Denis Simakov

Interpreting the eigenvalues

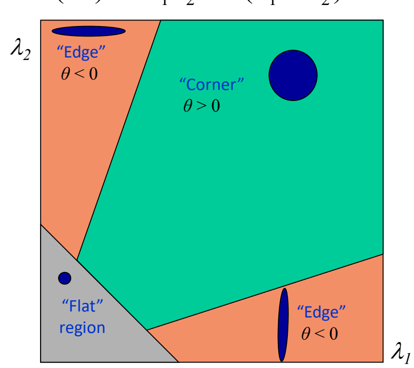
- Classification of image points using eigenvalues of M :



Slide credit: Kristen Grauman

Corner response function

$$\theta = \det(M) - \alpha \operatorname{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$



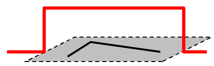
- Fast approximation
  - Avoid computing the eigenvalues
  - α: constant (0.04 to 0.06)

Slide credit: Kristen Grauman

Window Function $w(x,y)$

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

- Option 1: uniform window
  - Sum over square window

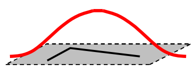


1 in window, 0 outside

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

  - Problem: not rotation invariant

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum



Gaussian

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

  - Result is rotation invariant

Slide credit: Bastian Leibe

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Summary: Harris Detector [Harris88]

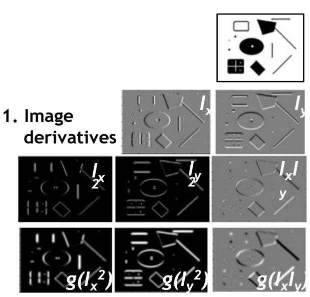
- Compute second moment matrix (autocorrelation matrix)

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

1. Image derivatives

2. Square of derivatives

3. Gaussian filter $g(\sigma_I)$



- Compute corner response

4. Cornerness function – two strong eigenvalues

$$\theta = \det[M(\sigma_I, \sigma_D)] - \alpha[\mathrm{trace}(M(\sigma_I, \sigma_D))]^2$$
$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Perform non-maximum suppression

C.Harris and M.Stephens. "A Combined Corner and Edge Detector." *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

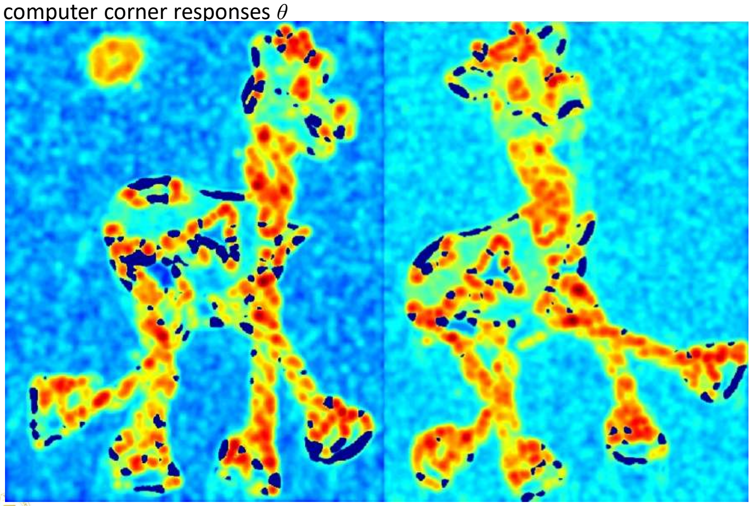
Slide credit: Krystian Mikolajczyk

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Harris Detector: Workflow

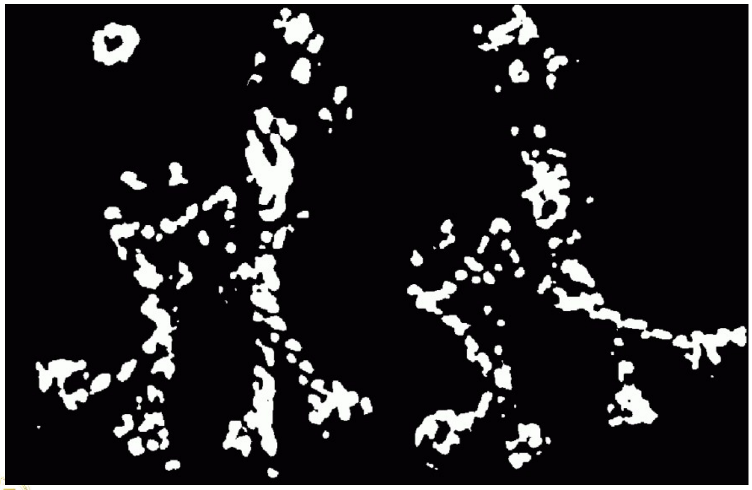
Harris Detector: Workflow

computer corner responses θ



Harris Detector: Workflow

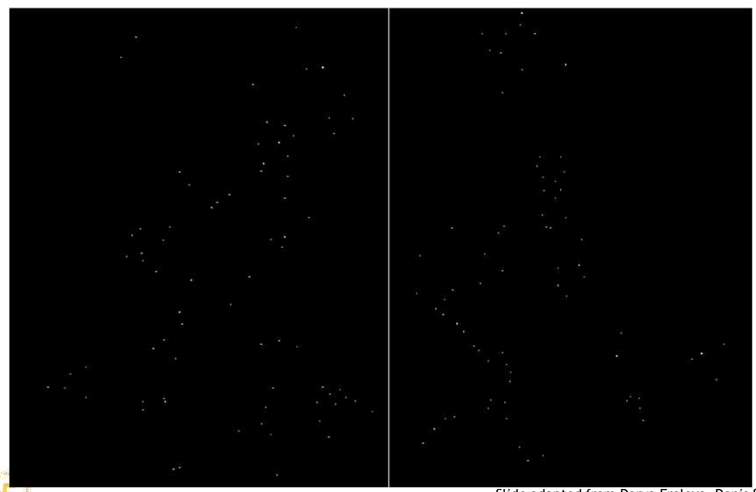
Take points where θ > threshold



85

Harris Detector: Workflow

Take only the local maxima of θ, where θ > threshold



86

Harris Detector: Workflow

Resulting Harris points

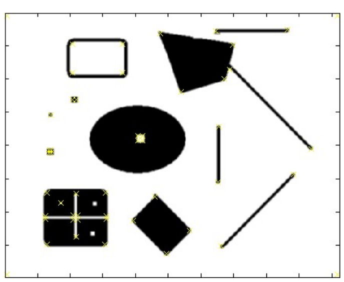
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide adapted from Darya Frolova, Denis Simakov

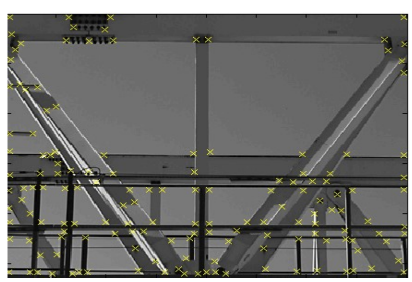
87

87

Harris Detector – Responses[Harris88]



Effect: A very precise corner detector.



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide credit: Krystian Mikolajczyk

88

88

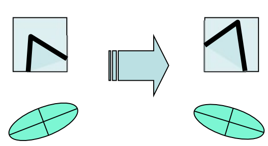
Harris Detector: Properties

- Translation invariance?

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance?



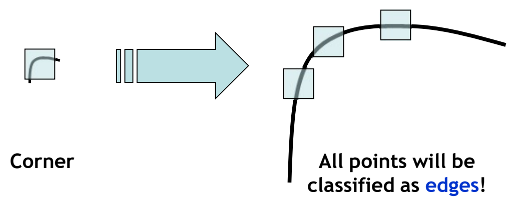
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

***Corner response θ* is invariant to image rotation**

Slide credit: Kristen Grauman

Harris Detector: Properties

- Translation invariance
- Rotation invariance
- Scale invariance?



Not invariant to image scale!

Slide credit: Kristen Grauman

91

Scale invariance: how to?

- Exhaustive search
- Invariance
- Robustness

92

Exhaustive search

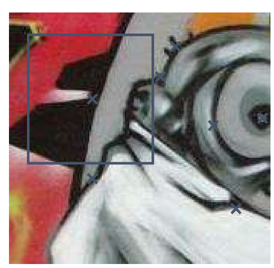
- Multi-scale approach




Slide adapted from T. Tuytelaars ECCV 2006 tutorial

Invariance

- Extract patch from each image individually

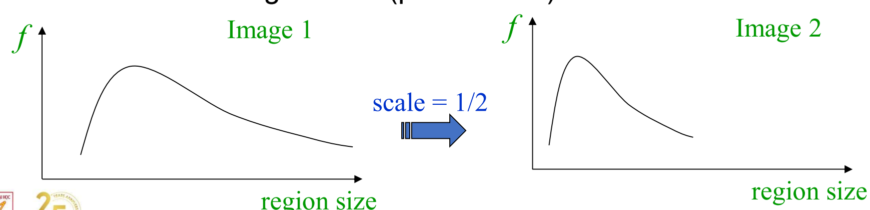



Slide adapted from T. Tuytelaars ECCV 2006 tutorial

Automatic scale selection

- Solution:
  - Design a function on the region, which is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.
  - For a point in one image, we can consider it as a function of region size (patch width)



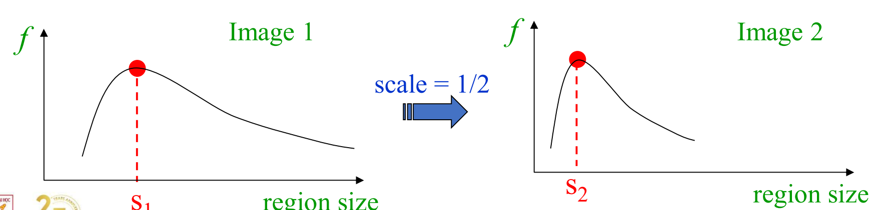
Automatic scale selection

- Common approach:

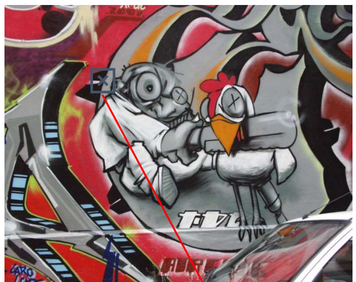
  Take a local maximum of this function

  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

Important: this scale invariant region size is found in each image independently!



Automatic Scale Selection



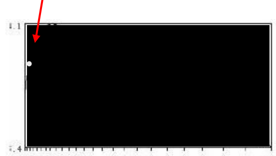
$$f(I_{i_1 \ldots i_m}(x,\sigma)) \;=\; f(I_{i_1 \ldots i_m}(x',\sigma'))$$

Same operator responses if the patch contains the same image up to scale factor.

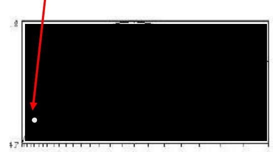
K. Grauman, B. Leibe

Example

Function responses for increasing scale (scale signature)



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

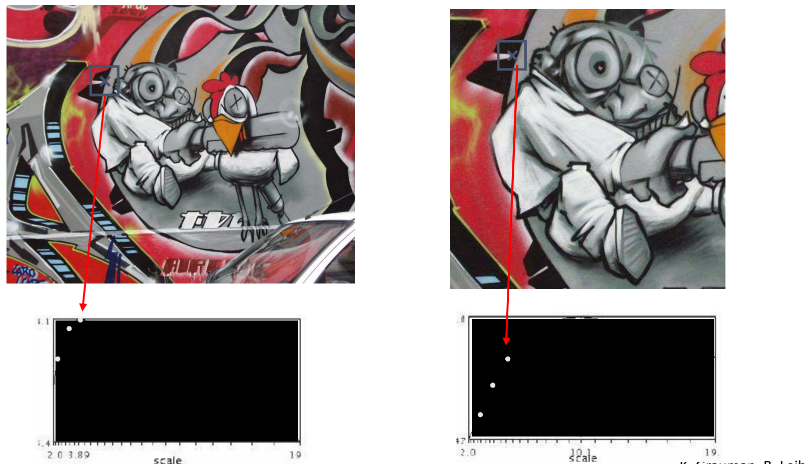


$f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

Example

Function responses for increasing scale (scale signature)



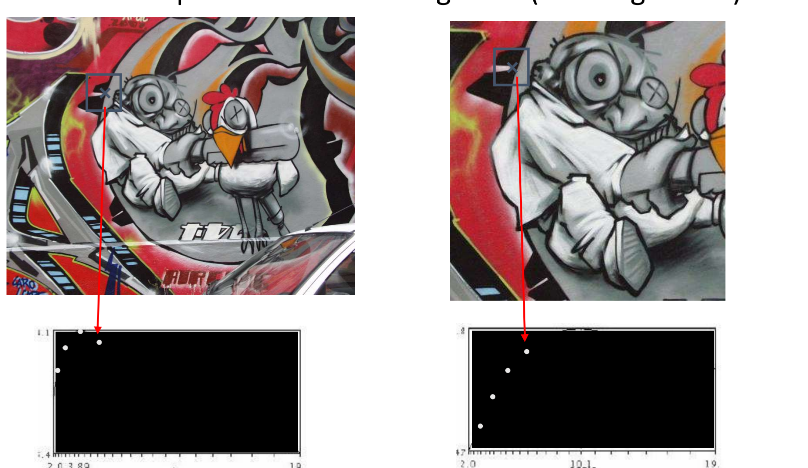
$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

99

Example

Function responses for increasing scale (scale signature)



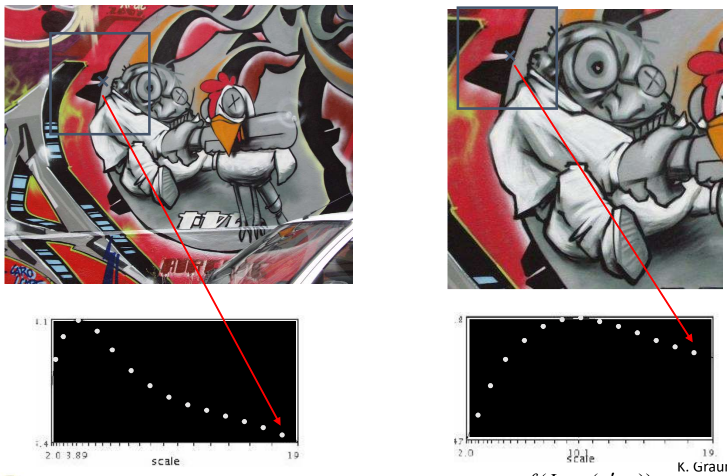
$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

K. Grauman, B. Leibe

100

Example

Function responses for increasing scale (scale signature)



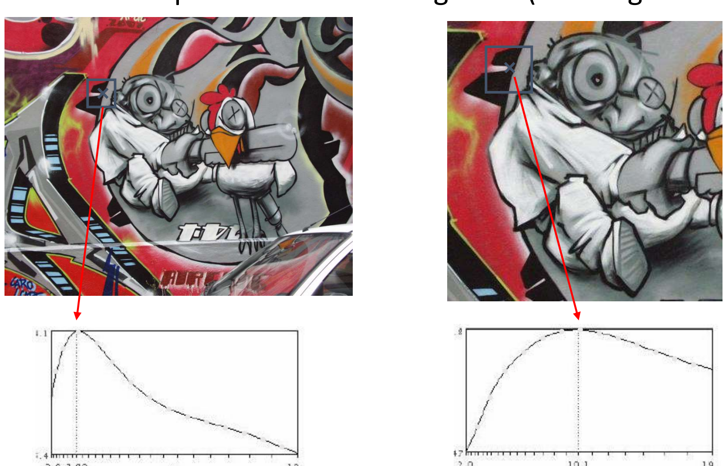
$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

K. Grauman, B. Leibe

101

Example

Function responses for increasing scale (scale signature)



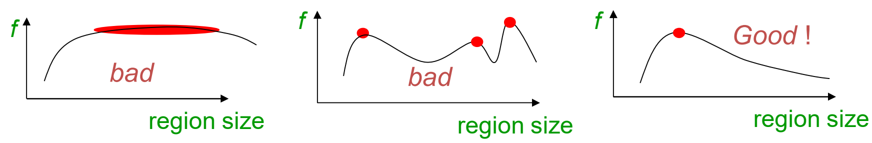
$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma'))$

K. Grauman, B. Leibe

102

Scale Invariant Detection

- A "good" function for scale detection:
  has one stable sharp peak



- For usual images: a good function would be one which responds to contrast (sharp local intensity change)

What is a useful signature function?

- Functions for determining scale $f = \text{Kernel} * \text{Image}$

Kernels:

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

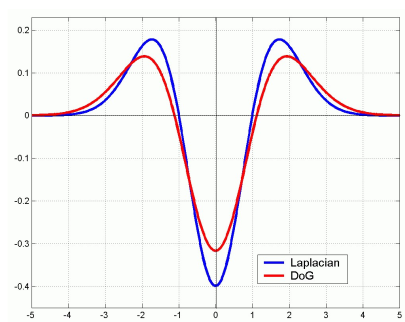
(Laplacian)

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

(Difference of Gaussians)

where Gaussian

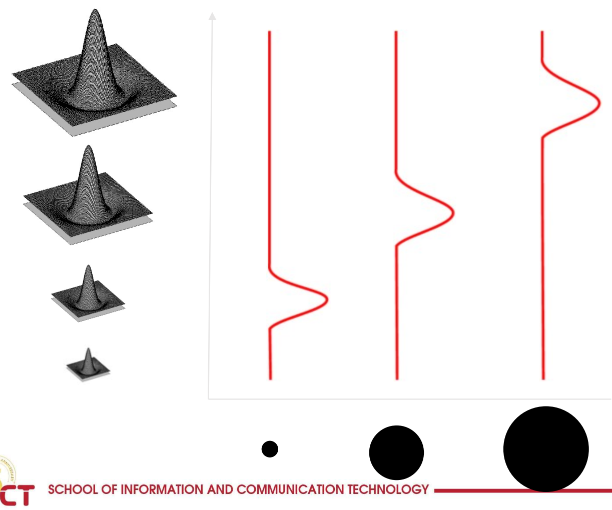
$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Note: both kernels are invariant to *scale* and *rotation*

What is a useful signature function?

- Laplacian-of-Gaussian = “blob” detector

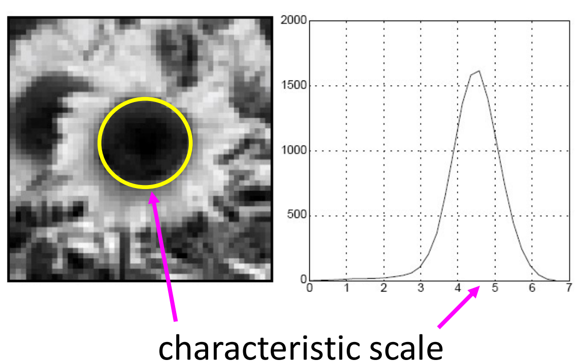


Source: K. Grauman, B. Leibe

105

Characteristic scale

- We define the ***characteristic scale*** as the scale that produces **peak of Laplacian response**



T. Lindeberg (1998). "Feature detection with automatic scale selection." *IJCV* **30** (2): pp 77--116.

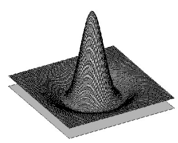
Source: Lana Lazebnik

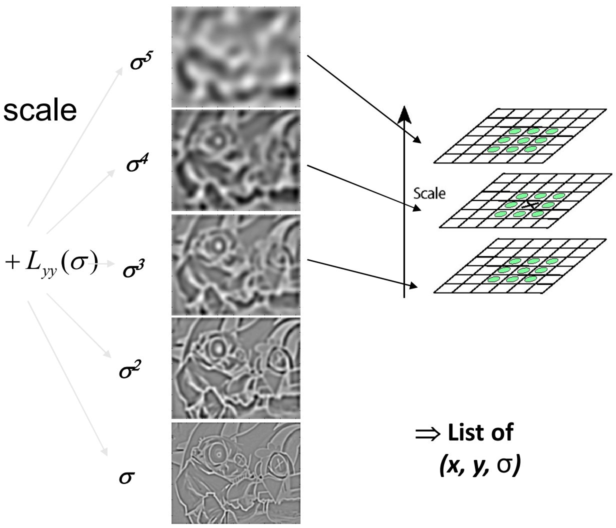
106

Laplacian-of-Gaussian (LoG)

- Interest points:
  Local maxima in scale space of LoG

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$





⇒ **List of *(x, y, σ)***

Source: K. Grauman, B. Leibe

Example: Scale-space blob detector

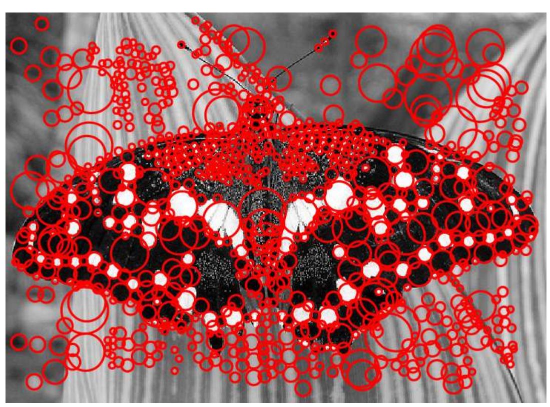
Source: Lana Lazebnik

Example: Scale-space blob detector

sigma = 11.9912

Source: Lana Lazebnik

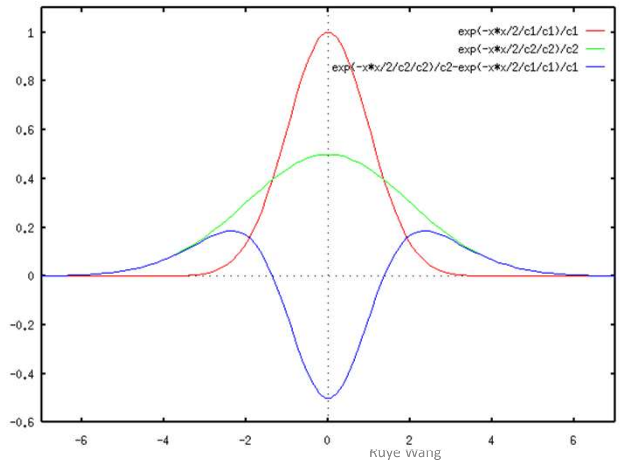
Example: Scale-space blob detector



Source: Lana Lazebnik

Alternative approach

Approximate LoG with Difference-of-Gaussian (DoG).



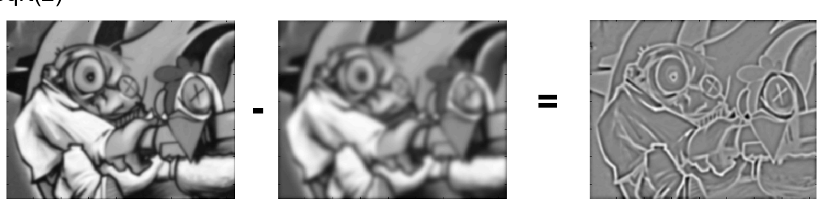
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

111

Alternative approach

- Approximate LoG with Difference-of-Gaussian (DoG):
  - 1. Blur image with σ Gaussian kernel
  - 2. Blur image with *kσ* Gaussian kernel
  - 3. Subtract 2. from 1.

Small *k* gives a closer approximation to LoG, but usually we want to build a scale space quickly out of this. *k = 1.6* gives an appropriate scale space, k = sqrt(2)

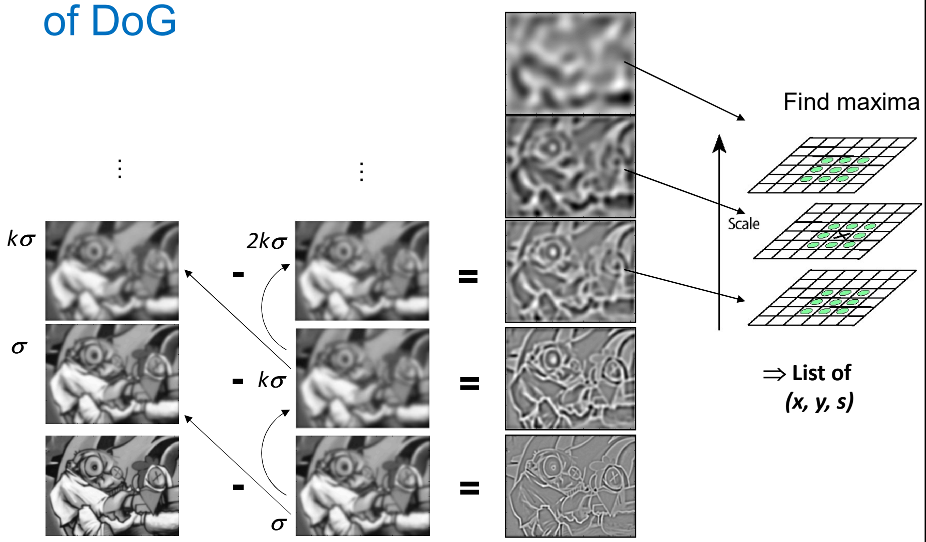


Source: K. Grauman, B. Leibe

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

112

Find local maxima in position-scale space of DoG

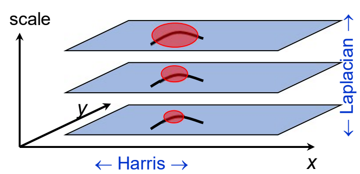


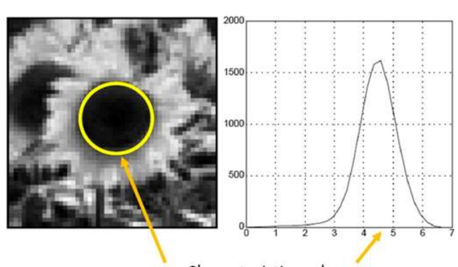
113

Harris-Laplacian

- Harris-Laplacian[1]

  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale





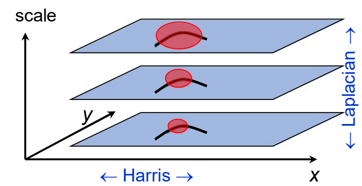
Characteristic scale

[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

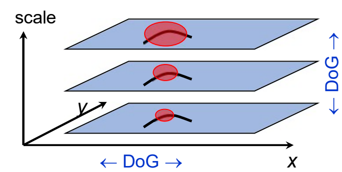
114

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



- SIFT (D.Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



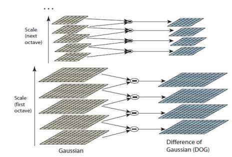
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

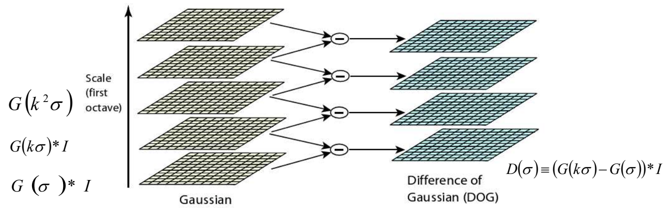
DoG (SIFT) keypoint Detector

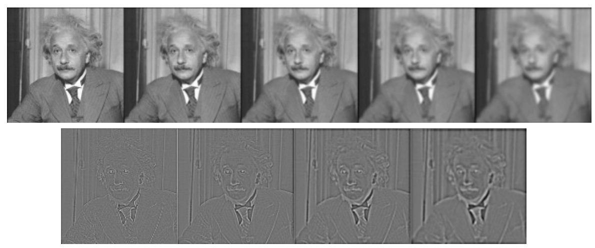
- DoG at multi-octaves
- Extrema detection in scale space
- Keypoint location
  - Interpolation
  - Removing instable points
- Orientation Assignment

DoG (SIFT) Detector

- DoG at multi-octaves



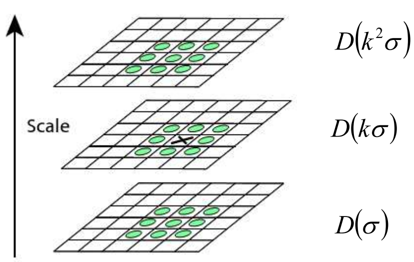




117

DoG (SIFT) Detector

- Scale-Space Extrema Choose all extrema within 3x3x3 neighborhood



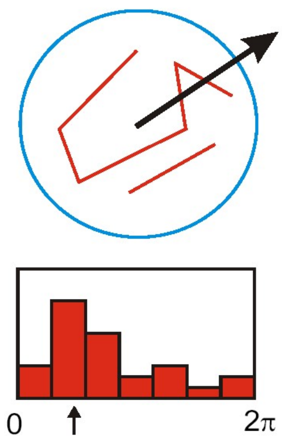
X is selected if it is larger or smaller than all 26 neighbors (its 8 neighbors in the current image and 9 neighbors each in the scales above and below)

118

DoG (SIFT) Detector

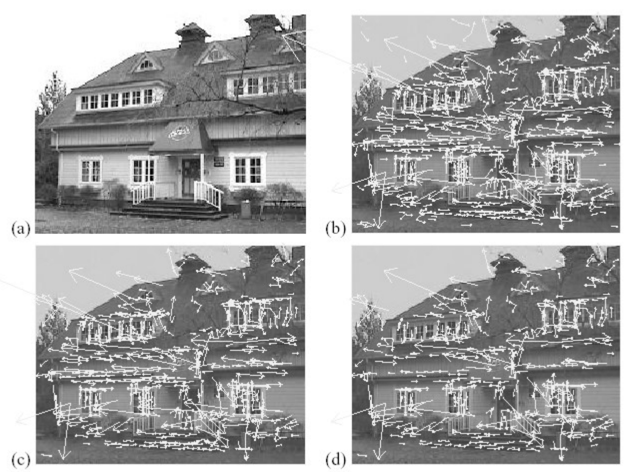
- Orientation assignment
  - Create histogram of local gradient directions at selected scale
  - Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates

  (x, y, scale,orientation)



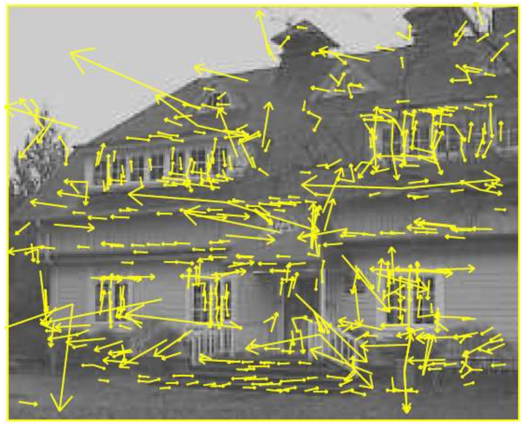
If 2 major orientations, use both.

Example of keypoint detection



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures (removing edge responses)

DoG (SIFT) Detector



A SIFT keypoint : **{x, y, scale, dominant orientation}**

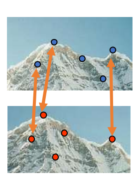
Source: Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004

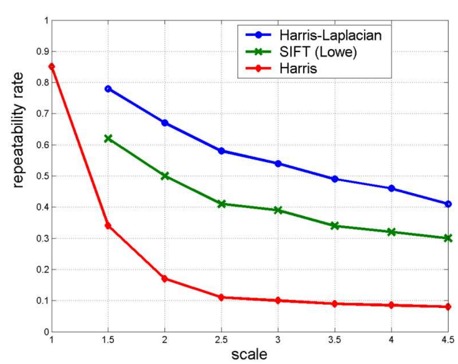
Scale Invariant Detectors

- Experimental evaluation of detectors w.r.t. scale change

Repeatability rate:

$$\frac{\#\ \text{correspondences}}{\#\ \text{possible correspondences}}$$





K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

Slide credit: CS131 -Juan Carlos Niebles and Ranjay Krishna

Many existing detectors available

- Hessian & Harris [Beaudet '78], [Harris '88]
- Laplacian, DoG [Lindeberg '98], [Lowe '99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid '01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid '04]
- EBR and IBR [Tuytelaars & Van Gool '04]
- MSER [Matas '02]
- Salient Regions [Kadir & Brady '01]
- Others…

- *Those detectors have become a basic building block for many recent applications in Computer Vision.*

Slide credit: Bastian Leibe

123

Feature extraction

- Global features
- **Local features**
  - Interest point detector
  - **Local descriptor**
  - Matching

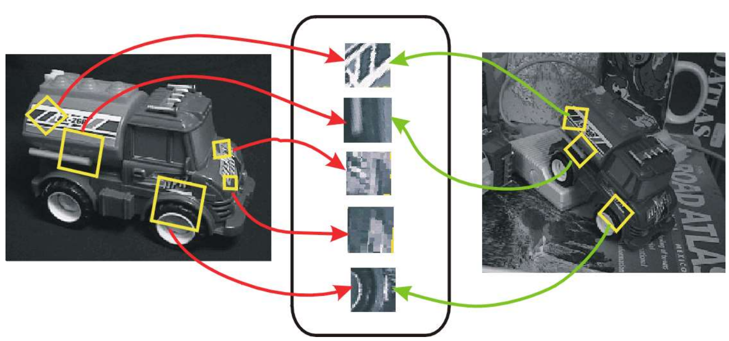
124

Local Descriptor

- Compact, good representation for local information
- Invariant
  - Geometric transformations: rotation, translation, scaling,..
  - Camera view change
  - Illiminution
- Exemples
  - SIFT, SURF(Speeded Up Robust Features), PCA-SIFT, …
  - LBP, BRISK, MSER and FREAK, …

Invariant local features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



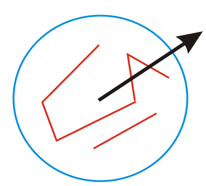
Following slides credit: CVPR 2003 Tutorial on **Recognition and Matching Based on Local Invariant Features** David Lowe

Advantages of invariant local features

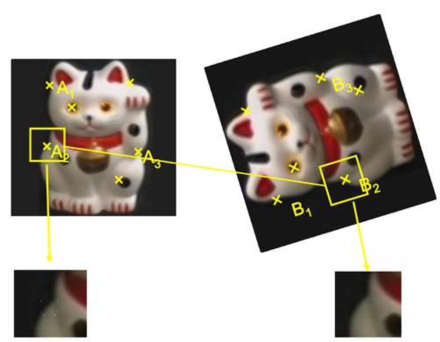
- Locality:
  - features are local, so robust to occlusion and clutter (no prior segmentation)
- Distinctiveness:
  - individual features can be matched to a large database of objects
- Quantity:
  - many features can be generated for even small objects
- Efficiency:
  - close to real-time performance
- Extensibility:
  - can easily be extended to wide range of differing feature types, with each adding robustness

Becoming rotation invariant

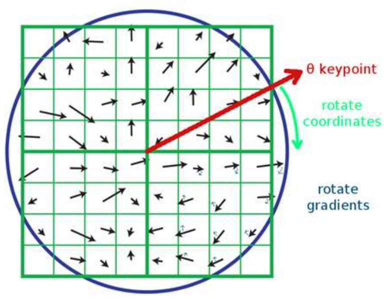
- We are given a keypoint and its scale from DoG



- We will select **a characteristic orientation** for the keypoint (based on the most prominent gradient there)
- We will describe all features **relative** to this orientation



SIFT descriptor formation

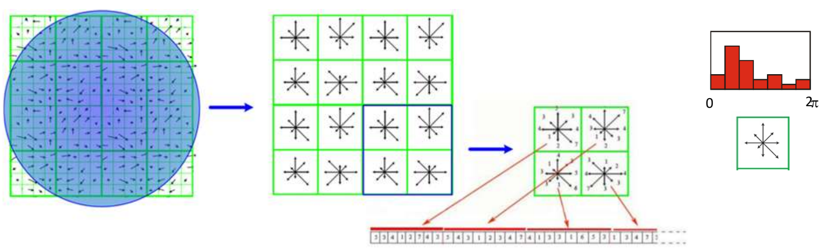


- Use the blurred image associated with the keypoint's scale
- Take image gradients over the keypoint neighborhood.
- To become rotation invariant, rotate the gradient **directions** AND **locations** by (-keypoint orientation)
  - Now we've cancelled out rotation and have gradients expressed at locations **relative** to keypoint orientation θ
  - We could also have just rotated the whole image by -θ, but that would be slower.

Source: Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004
http://campar.in.tum.de/twiki/pub/Chair/TeachingWs13TDCV/feature_descriptors.pdf

129

SIFT descriptor formation



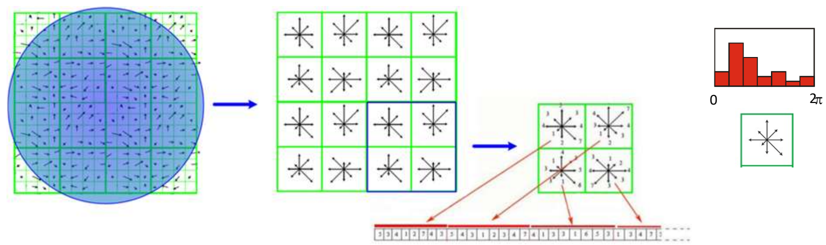
- Using precise gradient locations is fragile. We'd like to allow some "slop" in the image, and still produce a very similar descriptor
- Using Gaussian filter : to ***avoid sudden changes*** in the descriptor with small changes in the position of the window, and to give ***less emphasis to gradients that are far*** from the center of the descriptor, as these are most affected by misregistration errors
- Create array of orientation histograms (a 4x4 array is shown)

SIFT: Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004

Image: Ashish A Gupta, PhD thesis 2013

130

SIFT descriptor formation



- Put the rotated gradients into their local orientation histograms
  - A local orientation histogram has n bins (e.g 8 as shown).
  - To avoid all boundary affects in which the descriptor ***abruptly changes*** as a sample shifts smoothly from being within one histogram to another or from one orientation to another
    - ➔ interpolation is used to distribute the value of each gradient sample into adjacent histogram bins
    - Also, scale **down gradient contributions** for gradients far from the bin center: a weight of 1-d

  d: distance of the sample from the central value of the bin
- The SIFT authors found that best results were with 8 orientation bins per histogram and and a 4x4 histogram array ➔ a SIFT descriptor: vector of 128 values

Image: Ashish A Gupta, PhD thesis 2013

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

131

SIFT descriptor formation

- Adding robustness to **illumination changes**:
  - The descriptor is made of gradients (differences between pixels)
    - ➔ ***already invariant to changes in brightness*** (e.g. adding 10 to all image pixels yields the exact same descriptor)
  - A higher -contrast photo will increase the magnitude of gradients linearly
    - ➔ to correct for contrast changes, ***normalize the vector*** (scale to length 1.0)
  - Very large image gradients are usually from unreliable 3D illumination effects (glare, etc)
    - ➔ to reduce their effect, ***clamp all values*** in the vector to ***be ≤ 0.2*** (an experimentally tuned value). Then normalize the vector again.

➔ Result is a vector which is fairly invariant to illumination changes

SIFT: Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004

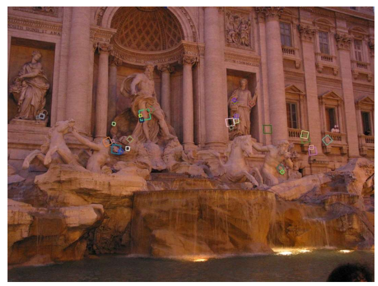
Image: Ashish A Gupta, PhD thesis 2013

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

132

SIFT

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint: up to about 60 degree out of plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time



Steve Seitz

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

133

Sensitivity to number of histogram orientations

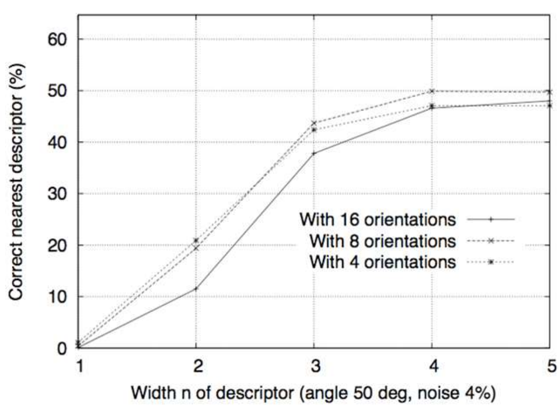


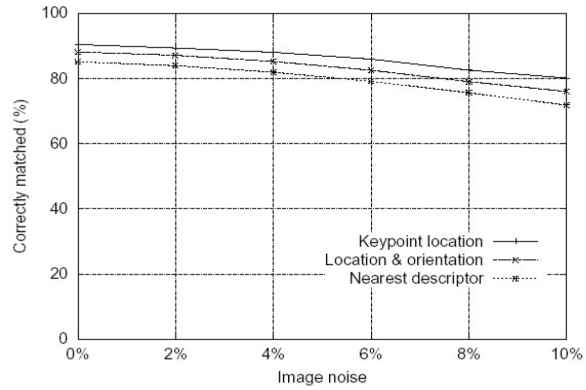
Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the $n \times n$ keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

134

Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



David G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, 60, 2 (2004), pp. 91-110

Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features

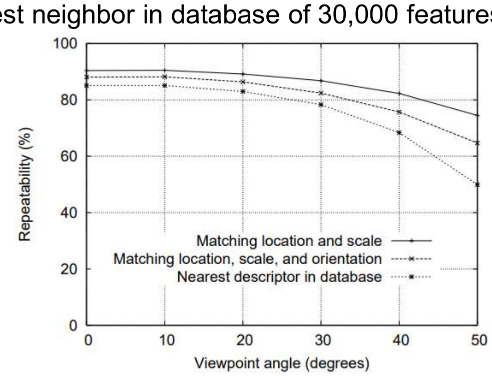
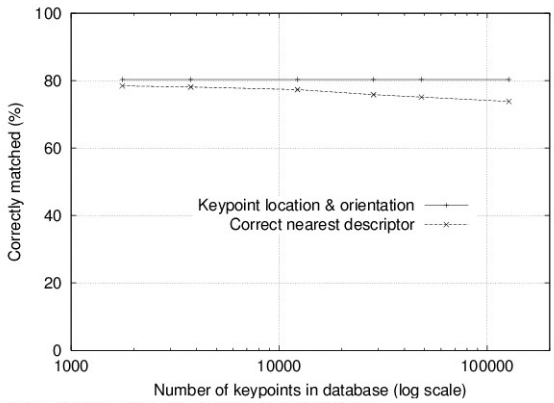


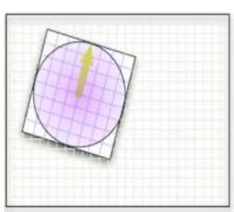
Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

Distinctiveness of features

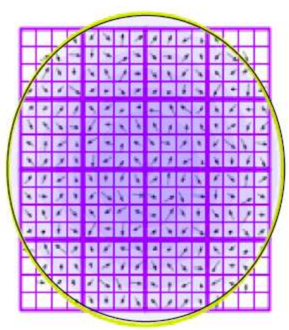
- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match



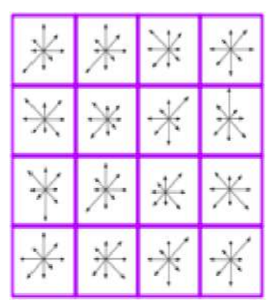
SIFT Keypoint Descriptor: summary



Blur the image using the scale of the keypoint (scale invariance)



Compute gradients in respect to the keypoint orientation(rotation invariance)



Compute orientation histogram in 8 directions over 4x4 sample regions

Source: Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004
http://campar.in.tum.de/twiki/pub/Chair/TeachingWs13TDCV/feature_descriptors.pdf

Other detectors and descriptors

Popular features: SURF, HOG, SIFT
http://campar.in.tum.de/twiki/pub/Chair/TeachingWs13TDCV/feature_descriptors.pdf

Summary some local features:
http://www.cse.iitm.ac.in/~vplab/courses/CV_DIP/PDF/Feature_Detectors_and_Descriptors.pdf

139

Feature extraction

- Global features
- **Local features**
  - Interest point detector
  - Local descriptor
  - **Matching**

140

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
   - Use L1, L2, cosine, Mahalanobis,… distance
2. Test all the features in I_2, find the one with min distance

OpenCV:
- Brute force matching
- Flann Matching: Fast Library for Approximate Nearest Neighbors

[Muja and Lowe, 2009]

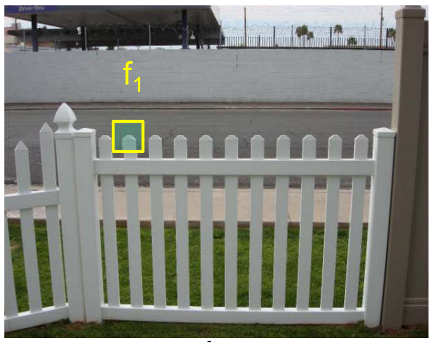
Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In VISAPP (1), pages 331–340, 2009

Feature matching

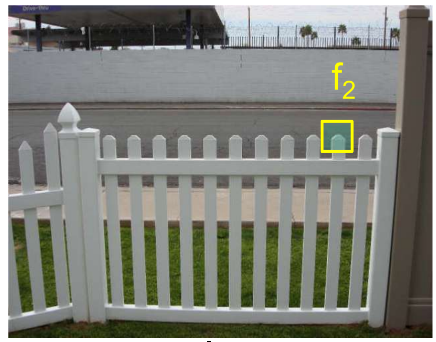
- How to define the difference between two features f_1, f_2?
  - Simple approach: use only distance value d(f1, f2)
    - ➔ can give good score to very ambiguous matches
  - Better approaches: add additional constraints
    - Radio of distance
    - Spatial constraints between neigborhood pixels
    - Fitting the transformation, then refine the matches (RANSAC)

Feature matching

- Simple approach: use distance value d(f1, f2)
    - ➔ can give good score to very ambiguous matches



I_1



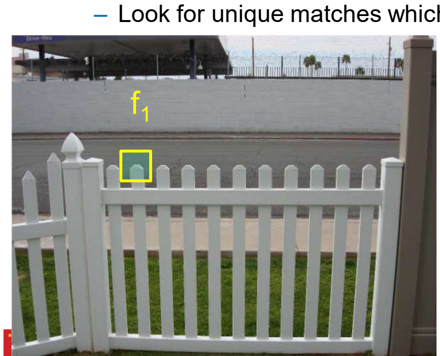
I_2

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

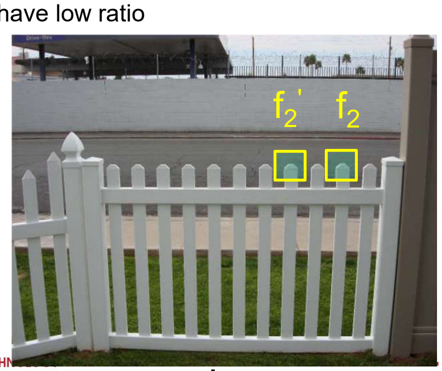
143

Feature matching

- Better approaches: radio of distance = d(f1,f2) / d(f1,f2')
    - f2 is best match to f1 in I2;
    - f2’ is 2nd best SSD match to f1 in I2
    - An ambiguous/bad match will have ratio close 1
    - Look for unique matches which have low ratio



I_1



I_2

144

Ratio of distances reliable for matching

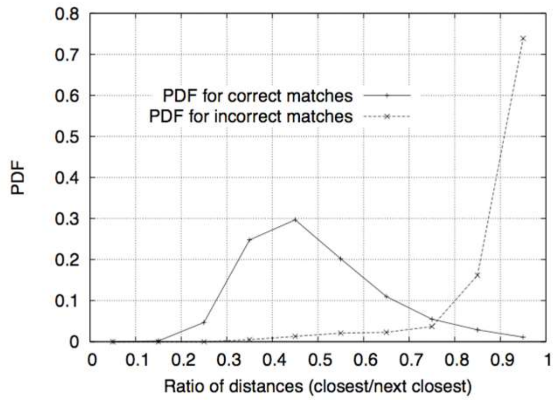


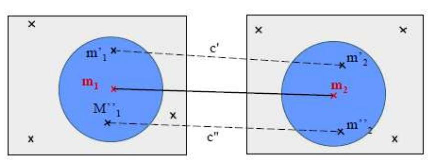
Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

David G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, 60, 2 (2004), pp. 91-110

Feature matching

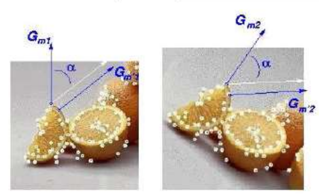
- Better approaches: Spatial constraints between neigborhood pixels

Contraintes de voisinage



Contraintes de rigidité locale :

Angle α entre 2 gradients couleur



Source: *from slides* of Valérie Gouet-Brunet

Feature matching

- Better approaches: fitting the transformation (RANSAC alg.)
  - Fitting 2D transformation matrix
    - Six variables
      - Each point give two equations
      - ➔ at least three points
    - Least squares

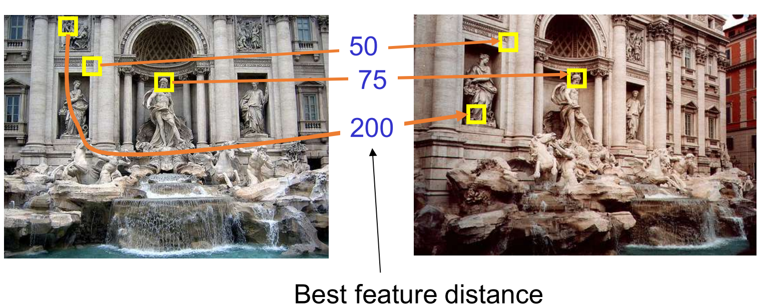
$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

  - RANSAC: refinement of matches
    - Compute error:

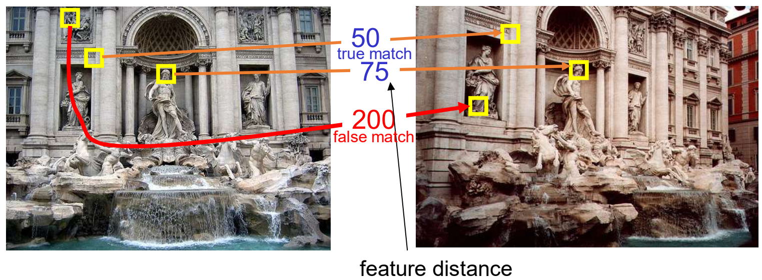
$$\left\| \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} - H \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \right\|_2$$

Evaluating the results

How can we measure the performance of a feature matcher?



True/false positives



The distance threshold affects performance

- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?

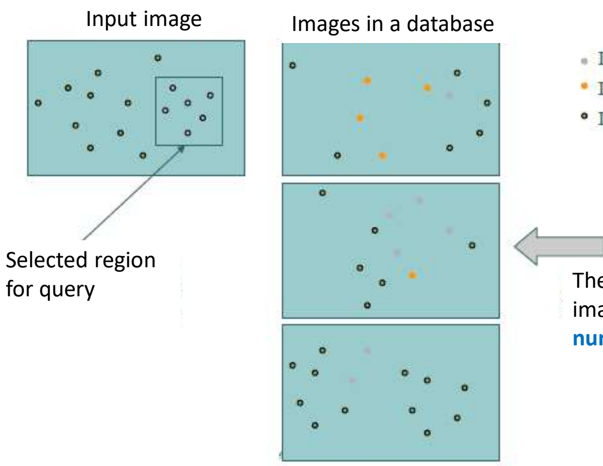
Image matching

- How to define the distance between 2 images I1, I2?
  - Using global features: easy

    d(I1, I2) = d(feature of I1, feature of I2)

  - Using local features:
    - Voting strategy
    - Solving an optimization problem (time consuming)
    - Building a "global" feature from local features: BoW (bag-of-words, bag-of-features), VLAD, ..

Voting strategy



- Distance faible ≤ ε
- Distance plus élevée ≤ ε
- Distance > ε

The similarity between 2 images is based on **the number of matches**

Source: Modified from slides of Valérie Gouet-Brunet

Optimization problem

- *Transportation problem*

I_1 : {(r_i, w_i), i=1, N} Provider

I_2 : {(r'_j, w_j), j=1, M} Consommer

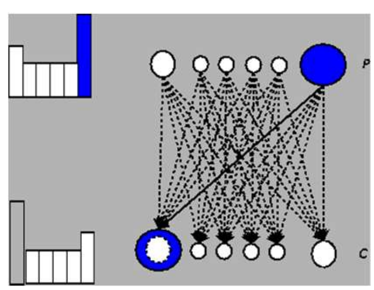
d(I_1, I_2) =???

$$d(I_1, I_2) = \min \sum_i \sum_j f_{ij} \times d(r_i, r'_j)$$

$$f_{ij} \geq 0$$

$$\sum_i f_{ij} \leq w_j, \sum_j f_{ij} \leq w_i$$

$$\sum_i \sum_j f_{ij} = \min(\sum_i w_i, \sum_j w_j)$$

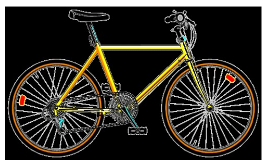


$$d_{EMD}(I_1, I_2) = \frac{\sum_i \sum_j f_{ij}^* \times d(r_i, r'_j)}{\sum_i \sum_j f_{ij}^*}$$

http://vellum.cz/~mikc/oss-projects/CarRecognition/doc/dp/node29.html

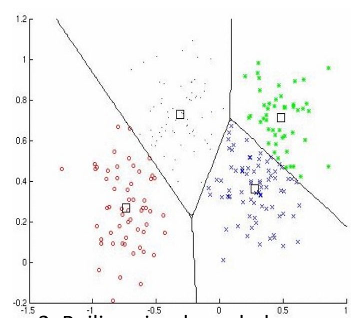
Bag-of-words

- Local feature ~~ a word
- An image ~~ a document
- Apply a technique for textual document representation: vector model




Visual Vocabulary …

1.Extracting local features from a set of images

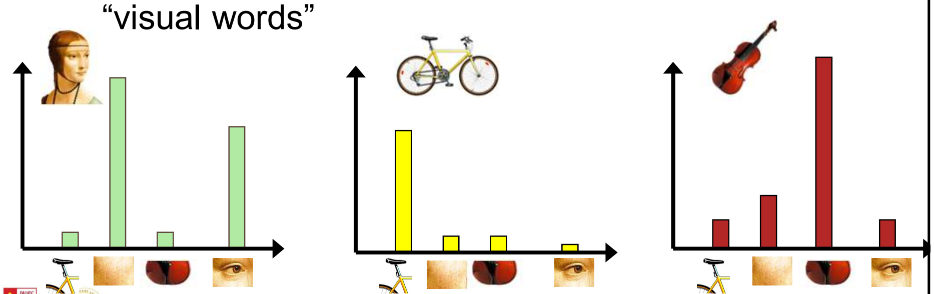


2. Builing visual vocabulary (dictionary) using a clustering method

3. An image is represented by a bag of words ➔ can be represented by tf.idf vector

Bag of words: outline

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

155

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict

156