Computer Vision for HCI

Interest Points

1

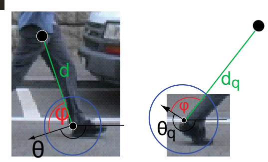
Topics of This Lecture

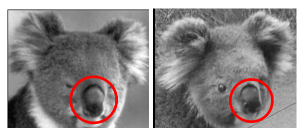
- Local Invariant Features
  - Motivation
  - Requirements, Invariances
- Keypoint Localization
  - Features from Accelerated Segment Test (FAST)
  - Harris
  - Shi-Tomasi
- Scale Invariant Region Selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussians detector
- Local Descriptors
  - Orientation normalization
  - SIFT

Slides adapted from Brown, Efros, Frolova, Grauman, Jacobs, Lazebnik, Leibe, Lowe, Mikolajczyk, Seitz, Simakov, Szeliski, and Tuytelaars

2

Global vs. Local Matching

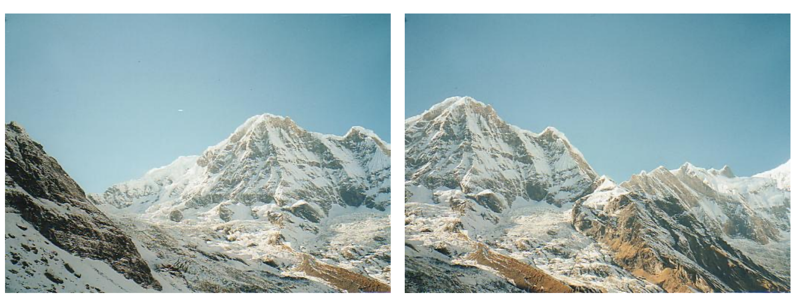
- Global image representations can be easily corrupted (e.g., by even small occlusion)
- Instead, describe and match using multiple local regions
- Increased robustness to
  - Occlusions
  
  - Articulation
  

  - Within-category variations
  

3

Application:

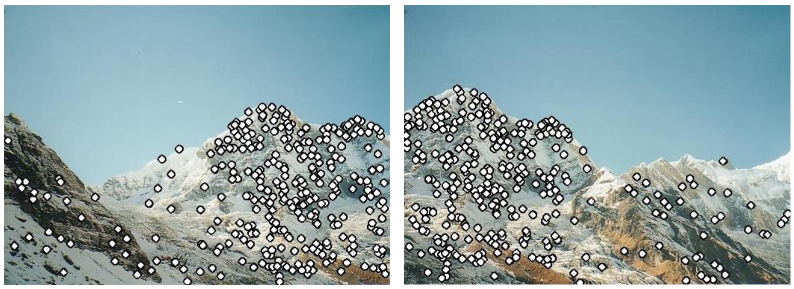
Image Stitching w/ Local Features



4

Application:
Image Stitching w/ Local Features

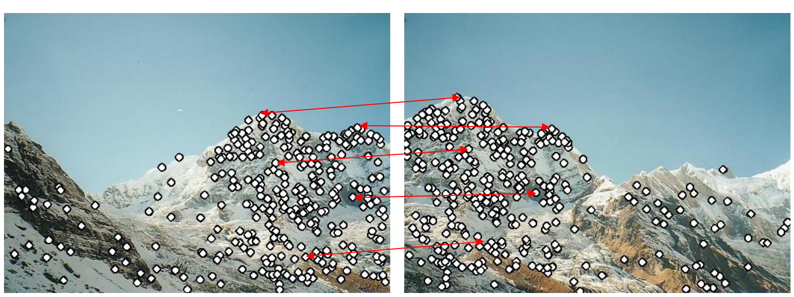
Find good locations



5

Application:
Image Stitching w/ Local Features

Match/Correspond them



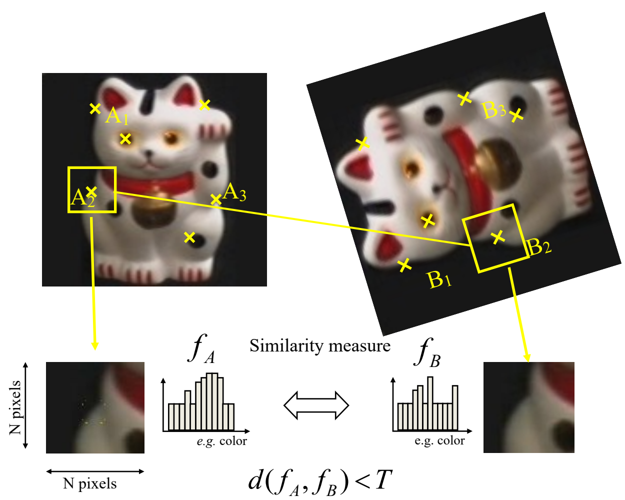
6

Application:
Image Stitching w/ Local Features

Transform images so points align

7

General Approach

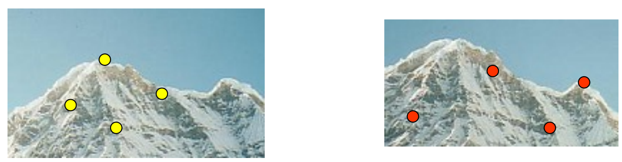


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and *normalize* the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

8

Common Requirements

- Problem 1:
  - Detect the <u>same</u> points *independently* in both images



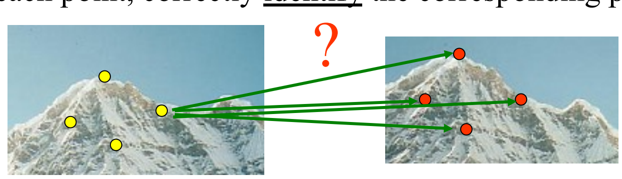
No chance to match!

We need a <u>repeatable</u> detector!

9

Common Requirements

- Problem 1:
  - Detect the same points *independently* in both images
- Problem 2:
  - For each point, correctly <u>identify</u> the corresponding point



We need a reliable and distinctive descriptor!

10

Invariance: Geometric Transformations

11

Invariance: Photometric Transformations

12

Requirements

- **Repeatable**: region extraction needs to be
  - Invariant to translation, rotation, scale changes, and other simple (affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- **Local**: to be more robust to occlusion and clutter
- **Large Quantity**: need a sufficient number of regions to cover the object
- **Highly Distinctive**: regions should contain "interesting" structure
- **Efficient**: close to real-time performance

13

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…

Detectors are a basic building block for many applications in Computer Vision

14

Topics of This Lecture

- Local Invariant Features
  - Motivation
  - Requirements, Invariances
- Keypoint Localization
  - Features from Accelerated Segment Test (FAST)
  - Harris
  - Shi-Tomasi
- Scale Invariant Region Selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussians detector
- Local Descriptors
  - Orientation normalization
  - SIFT

15

FAST Detector

16

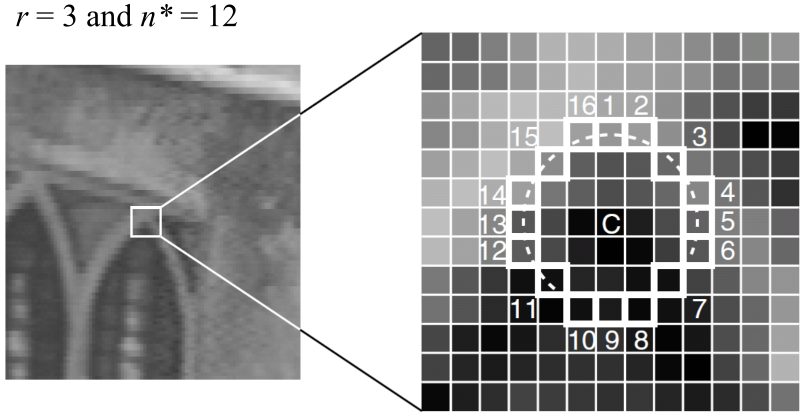
Features from Accelerated Segment Test (**FAST**)

- Compare local neighborhood around each pixel to determine if the pixel is a good feature point
- For each pixel x
  - Look at the pixels on the border of a circle of radius r around x
  - Let n be the number of **contiguous pixels** (check for wrap-around!!) whose intensities are either
    1) **all** $> (I(x) + T)$ ***or*** 2) **all** $< (I(x) - T)$
    - $I(x)$ is the intensity at pixel x and T is a threshold
  - If $n \geq n^*$ then the pixel is considered a feature
    - Original paper (Rosten 2006) suggests using $r = 3$ (yields 16 pixels on border) and $n^* = 12$
    - Later results suggest using $r = 3$ (yields 16 pixels on border) and $n^* = 9$ provides best detector

17

FAST Example

- $r = 3$ and $n^* = 12$



- Techniques can be employed to decrease computation time

18

Example

- Use FAST corner detector to determine if pixel X with intensity value of 100 is a corner point
  - Intensity threshold T = 20
    - Thus “Above” > 120 and “Below” < 80 (else “Within”)
  - Contiguous pixel count $n \geq n^* = 9$

Values of the pixels on the border of the circle (of radius 3) centered at X:

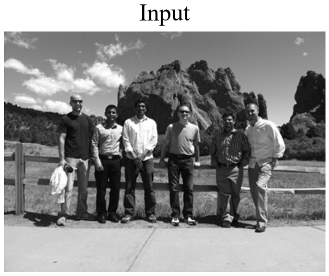
70	125	130	125	140	135	75	70	65	140	145	140	100	105	65	60
B	A	A	A	A	A	B	B	B	A	A	A	W	W	B	B
			5				3			3				3	

5 < 9 so **NO** (the pixel X is <u>not</u> a corner point)!

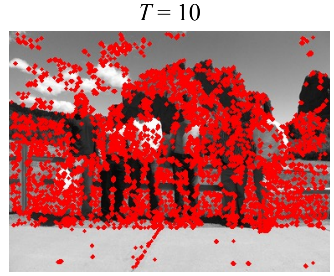
19

FAST Example ($r = 3$, $n^* = 12$)

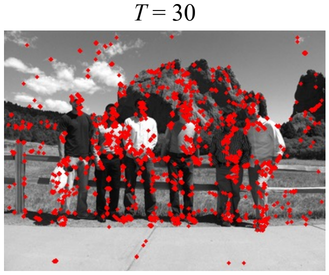
Input



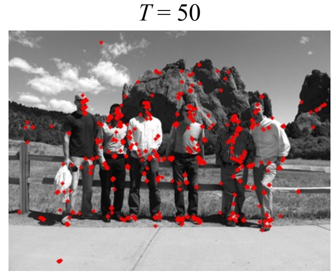
$T = 10$



$T = 30$



$T = 50$

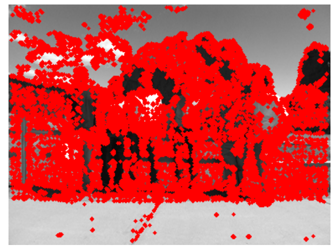


20

FAST Example ($r = 3$, $n^* = 9$)

Input

$T = 10$



$T = 30$

$T = 50$

21

Harris Detector

22

Harris Detector

- Based on the matrix M, which is a 2x2 correlation matrix from image derivatives around a pixel (recall from Optic Flow and KLT!):

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

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

Window/Weighting function

Gradients with respect to x and y

To evaluate the "cornerness" of a particular pixel location from its local region

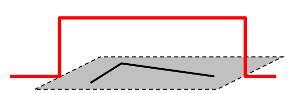
23

Window/Weighting 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

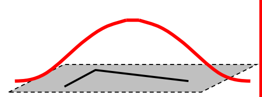
$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$



1 in window, 0 outside

- Option 2: Gaussian window
  - Gaussian performs weighted sum
  - Also is rotation invariant

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



Gaussian

24

Gradients

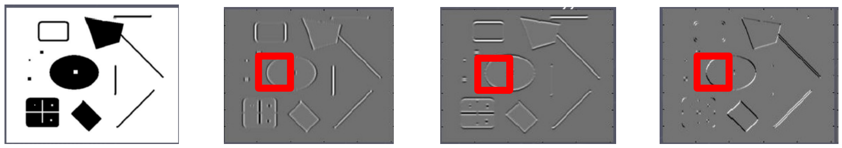
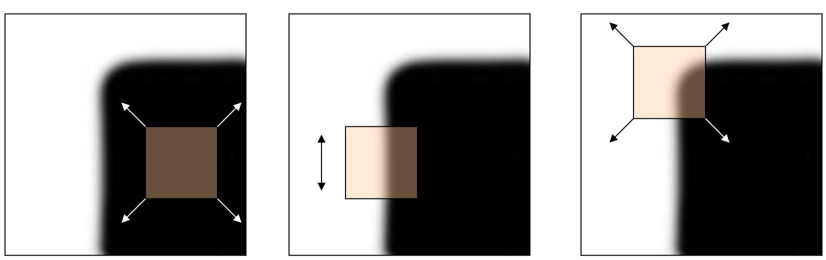


Image I | I_x | I_y | I_xI_y

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

25

Flat vs. Edge vs. Corner



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

26

Eigenvalues of M

Let $\lambda_1 \geq \lambda_2$ be the Eigenvalues of M

$\lambda_1 < e$: intensity is nearly constant over patch

$\lambda_1 \gg \lambda_2$: edge was found

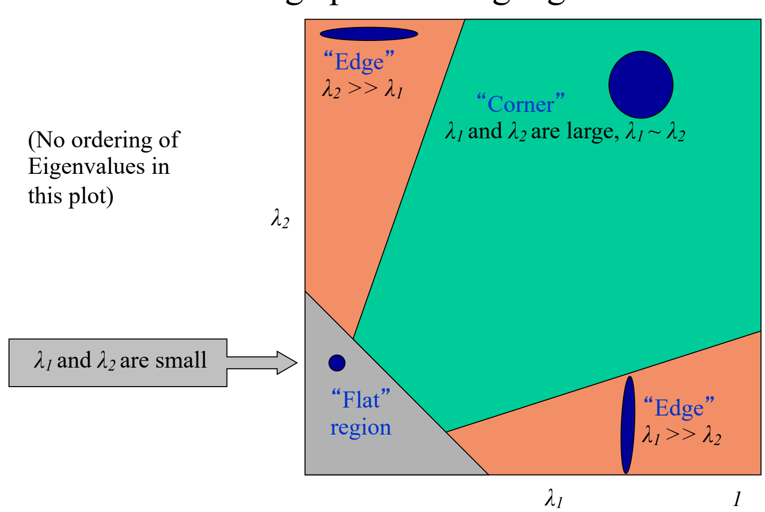
$\lambda_2 > T$: corner or textured patch found

(Recall KLT "Good Features" to track!)

27

Interpreting the Eigenvalues

- Classification of image points using eigenvalues of M



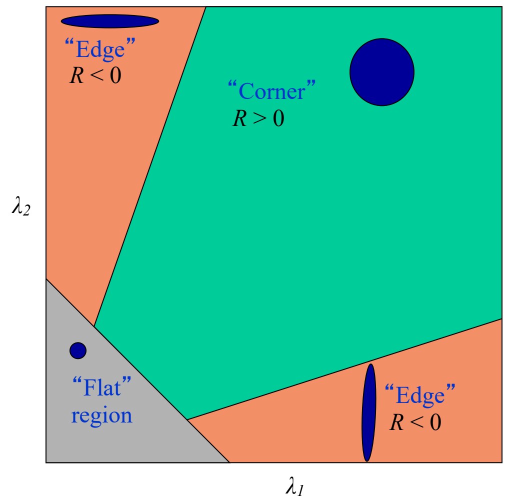
28

"Harris" Corner Response Function

$$R = \det(M) - \alpha\,\mathrm{trace}(M)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2$$

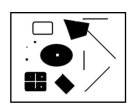
Know the det() and trace() relationship to eigenvalues!

- Fast approximation
  - Avoids computing Eigenvalues
  - α: constant (0.04 to 0.06)



29

Summary: Harris Detector

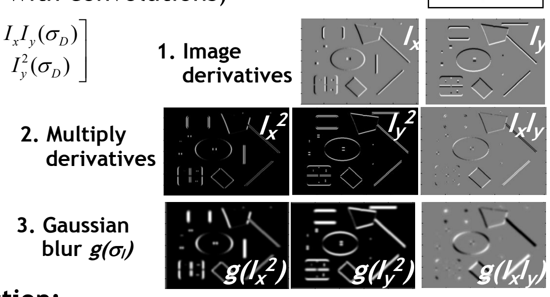


Compute *M* (can easily do with convolutions)

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

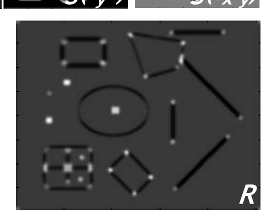
Gaussian **window** sigma

Gaussian **derivative** sigma



Compute Cornerness function:

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



Remove small values in *R*, and then perform non-maximum suppression (keeping only peaks)

30

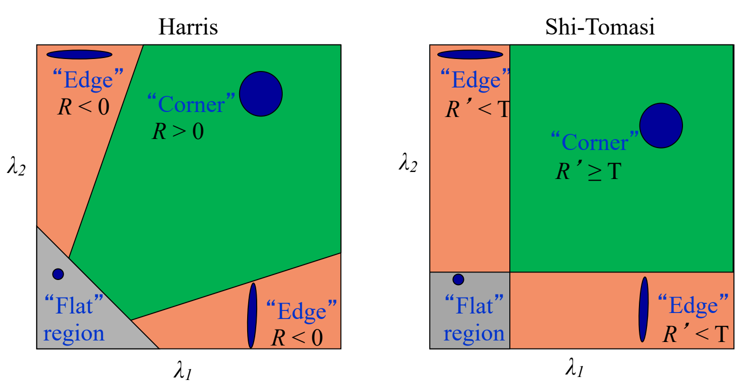
Harris Detector – Responses

Shi-Tomasi Corner Detector

- Similar to Harris corner points

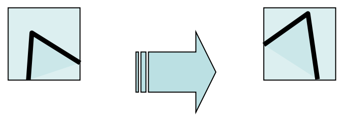
Harris $R = \det(M) - \alpha\, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$

Shi-Tomasi $R' = \min(\lambda_1, \lambda_2)$ (recall KLT good features!)



Harris and Shi-Tomasi Detector: Properties

- Rotation invariance?



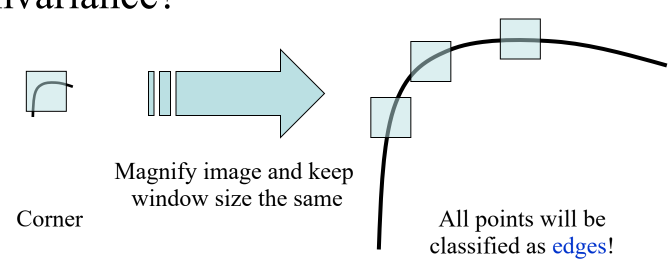
Eigenvalues remains the same

Corner response R is invariant to image rotation

33

Harris and Shi-Tomasi Detector: Properties

- Rotation invariance
- Scale invariance?



Not invariant to image scale!

34

Topics of This Lecture

- Local Invariant Features
  - Motivation
  - Requirements, Invariances
- Keypoint Localization
  - Features from Accelerated Segment Test (FAST)
  - Harris
  - Shi-Tomasi
- Scale Invariant Region Selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussians detector
- Local Descriptors
  - Orientation normalization
  - SIFT

35

From Points to Regions…

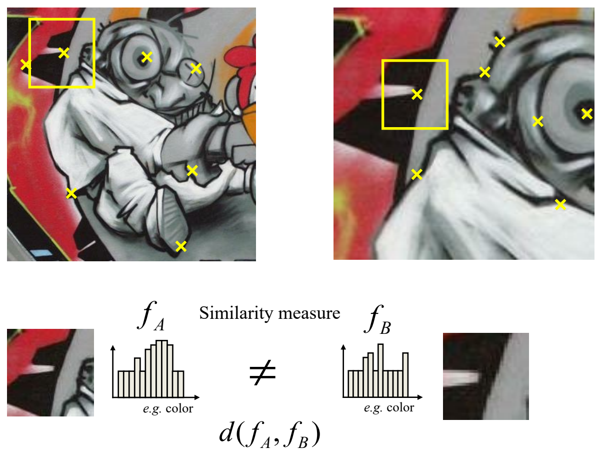
- Harris and Shi-Tomasi operators define interest points

- How can we detect scale invariant "interest *regions*"?
  - In order to **compare** points (e.g., for image matching or stitching), we need to compute a descriptor over a region
  - How can we define such a region in a **scale invariant** manner?

36

Naïve Approach: Exhaustive Search

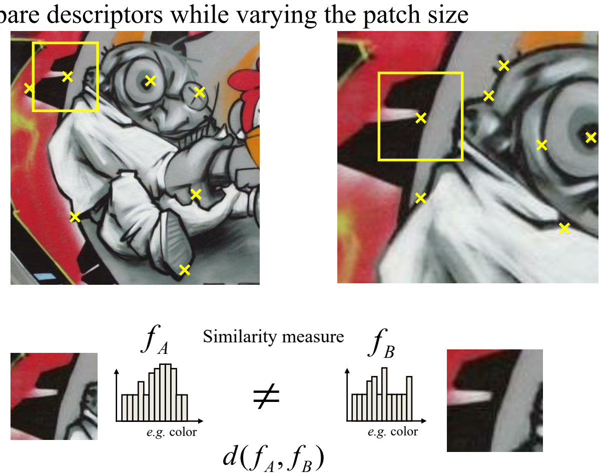
- Multi-scale procedure
  - Compare descriptors while varying the patch size in the other image



37

Naïve Approach: Exhaustive Search

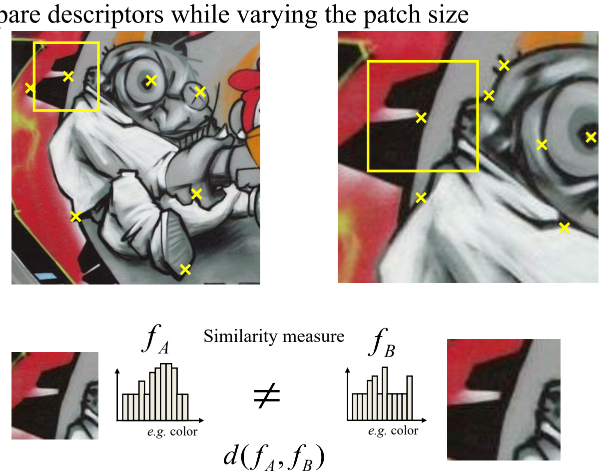
- Multi-scale procedure
  - Compare descriptors while varying the patch size



38

Naïve Approach: Exhaustive Search

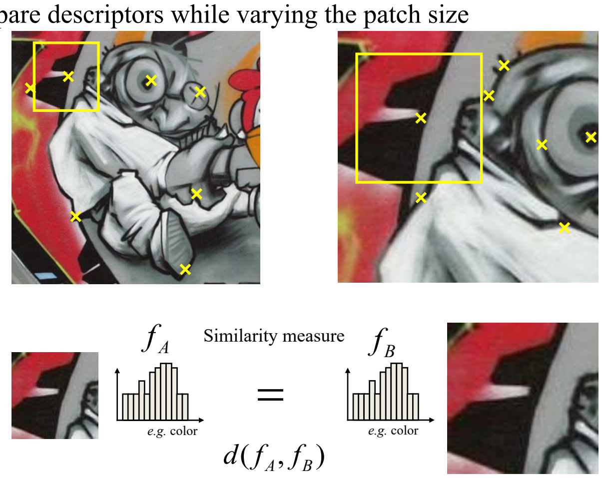
- Multi-scale procedure
  - Compare descriptors while varying the patch size



39

Naïve Approach: Exhaustive Search

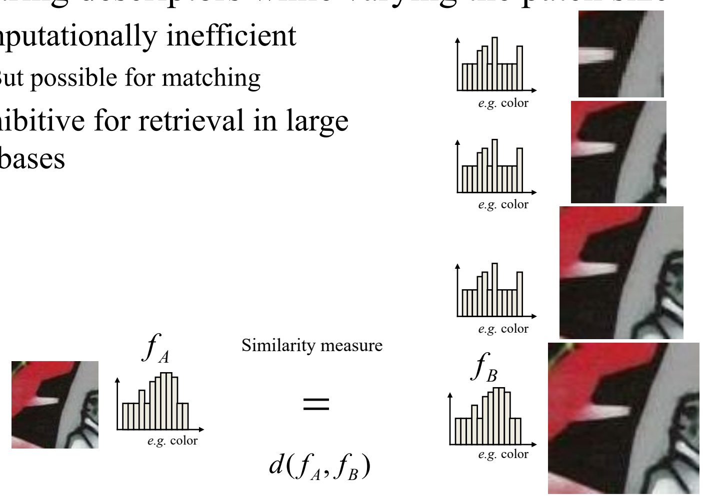
- Multi-scale procedure
  - Compare descriptors while varying the patch size



40

Naïve Approach: Exhaustive Search

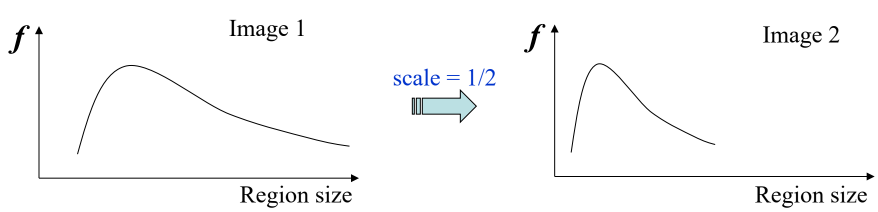
- Comparing descriptors while varying the patch size
  - Computationally inefficient
    - But possible for matching
  - Prohibitive for retrieval in large databases



41

Automatic Scale Selection

- Solution:
  - Design a "magic" function ***f*** that is maximal at the correct region size for a given location (indicates the best region size)

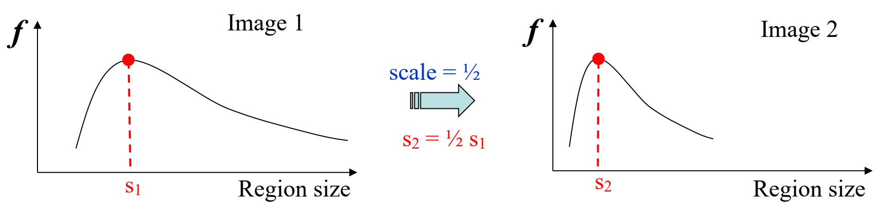


42

Automatic Scale Selection

- Choose the region size that is "best"

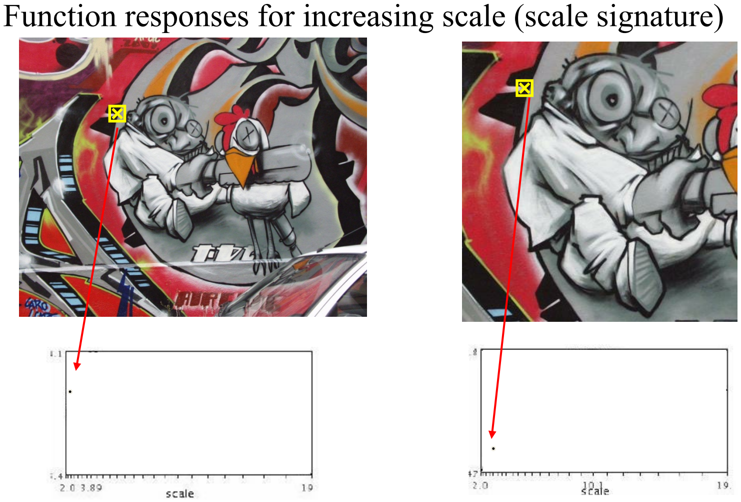
Important: this scale invariant region size (for a point) is found in each image **independently**!



43

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

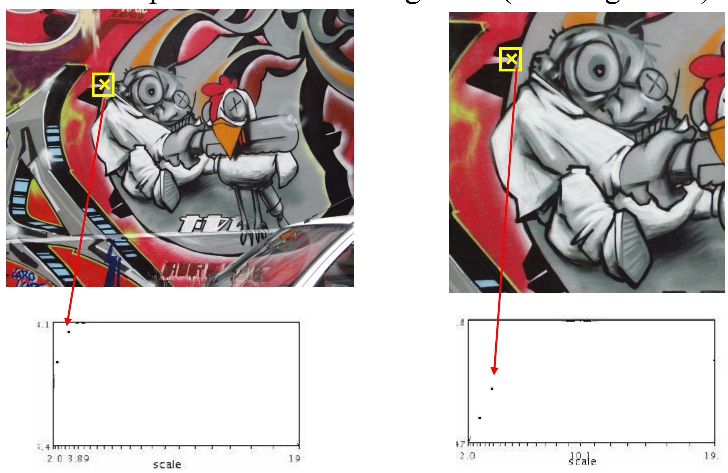


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

44

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

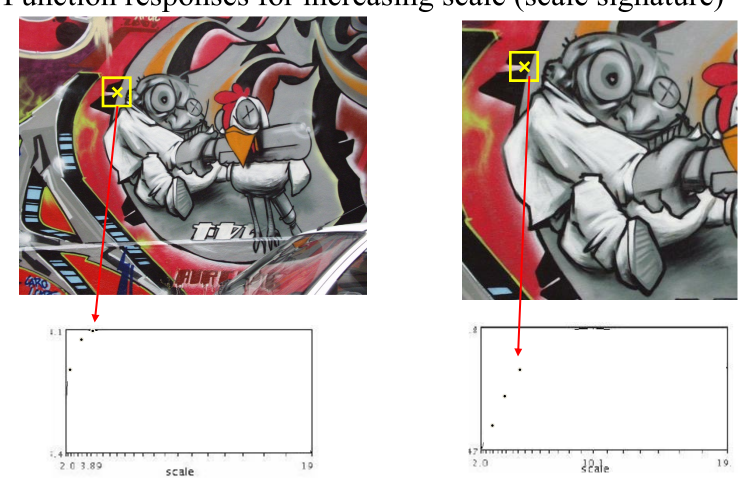


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

45

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

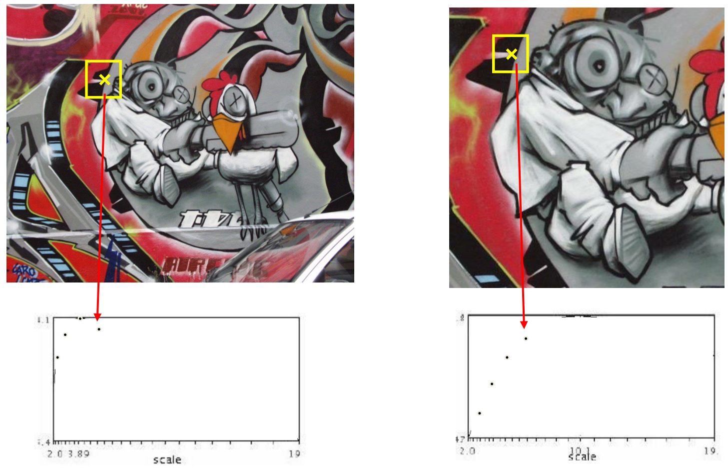


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

46

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

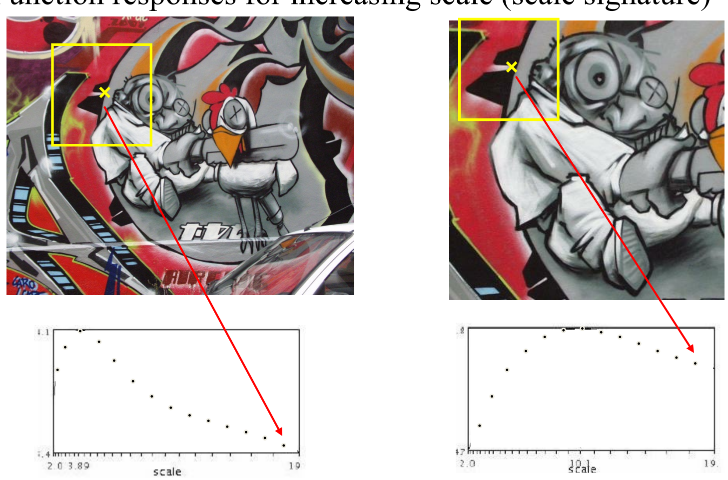


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

47

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

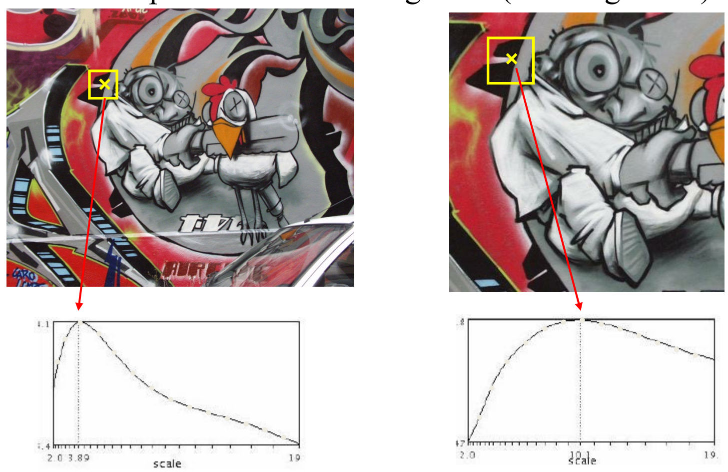


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

48

Automatic Scale Selection

- Function responses for increasing scale (scale signature)

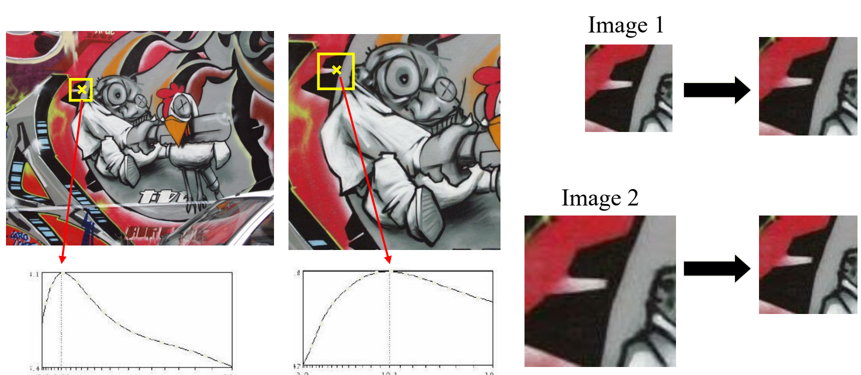


$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma'))$

49

Automatic Scale Selection

- Normalize: Rescale best region size to fixed size



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

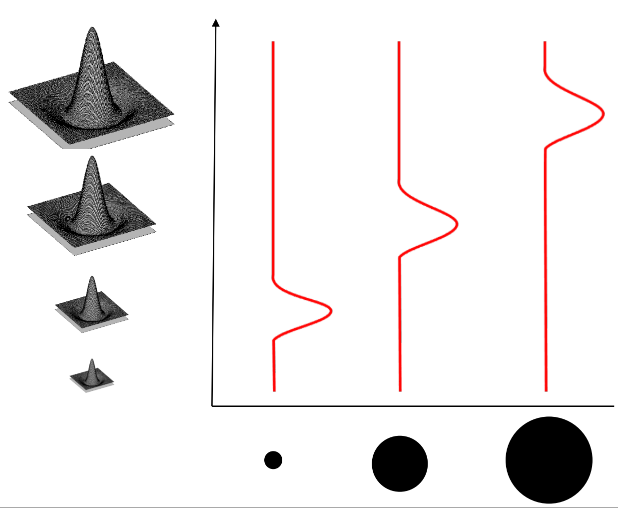
50

So what could be a possible "magic function" for ***f*** ???

51

A Useful Function

- "Scale-normalized" Laplacian-of-Gaussian = *blob/spot* detector

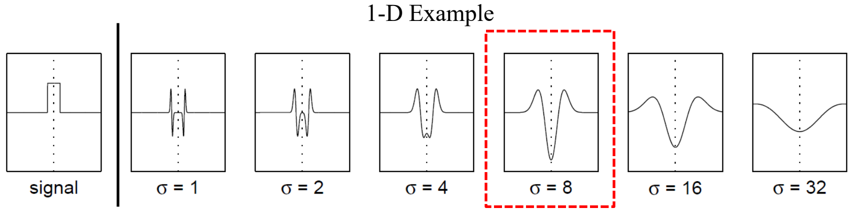


52

Scale-Normalized Laplacian-of-Gaussian

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

1-D Example



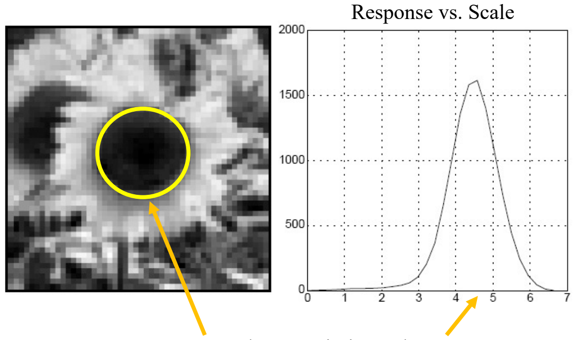
Scale-normalized Laplacian shows the strongest response at the *characteristic scale* of the original signal ($\sigma = 8$, $x = 0$)

53

Characteristic Scale

2-D Example

Response vs. Scale

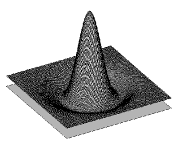


Characteristic scale

54

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local <u>maxima</u> in scale space (above/below/around) of scale-normalized Laplacian-of-Gaussian



$$\sigma^2\left(G_{xx}(\sigma)+G_{yy}(\sigma)\right)$$

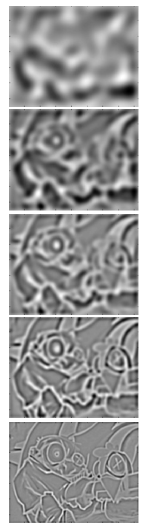
$\sigma = \sigma_0^5$

$\sigma = \sigma_0^4$

$\sigma = \sigma_0^3$

$\sigma = \sigma_0^2$

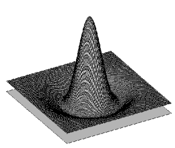
$\sigma = \sigma_0$



55

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local <u>maxima</u> in scale space (above/below/around) of scale-normalized Laplacian-of-Gaussian



$$\sigma^2\left(G_{xx}(\sigma)+G_{yy}(\sigma)\right)$$

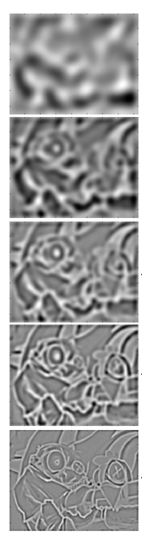
$\sigma = \sigma_0^5$

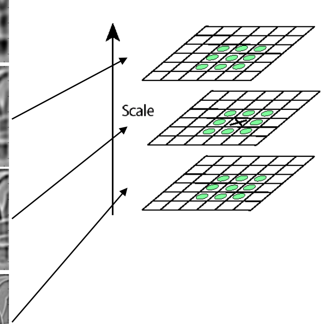
$\sigma = \sigma_0^4$

$\sigma = \sigma_0^3$

$\sigma = \sigma_0^2$

$\sigma = \sigma_0$

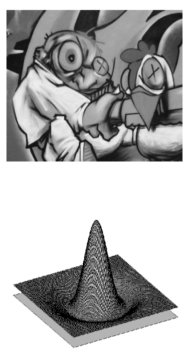


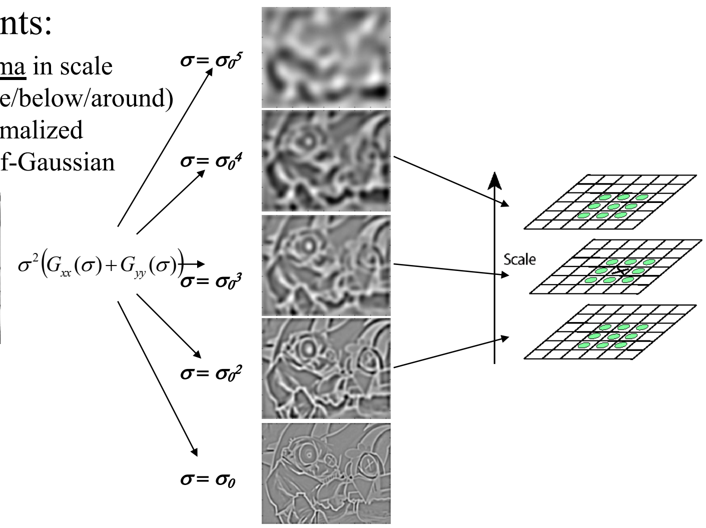


56

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local <u>maxima</u> in scale space (above/below/around) of scale-normalized Laplacian-of-Gaussian

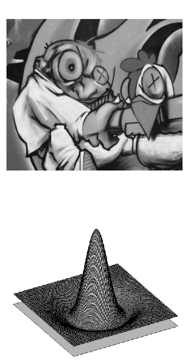


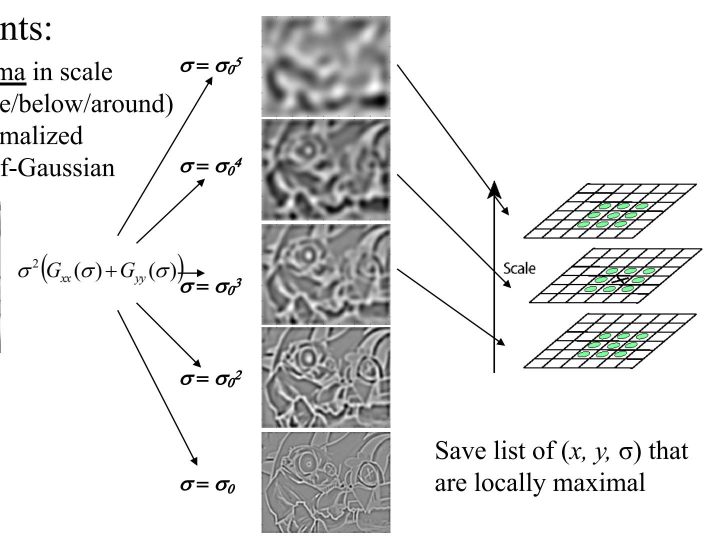


57

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local <u>maxima</u> in scale space (above/below/around) of scale-normalized Laplacian-of-Gaussian



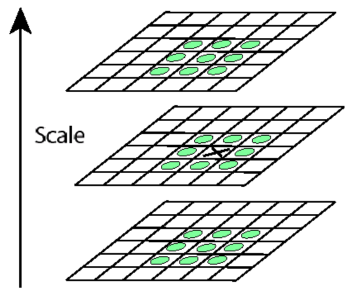


Save list of (x, y, σ) that are locally maximal

58

Keypoint Localization

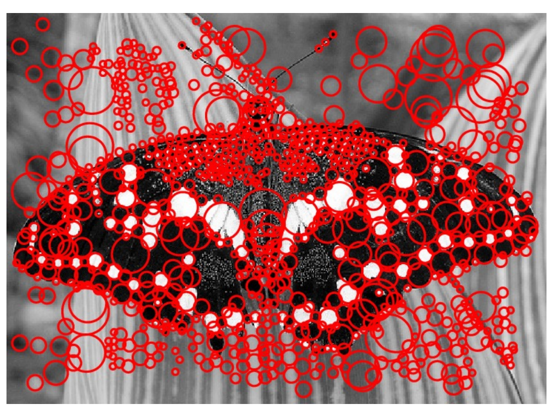
- Detect maxima in scale space
- Then reject points with low image contrast (threshold)
- Eliminate edge responses



Candidate keypoints:
list of (x,y,σ)

59

LoG Detector



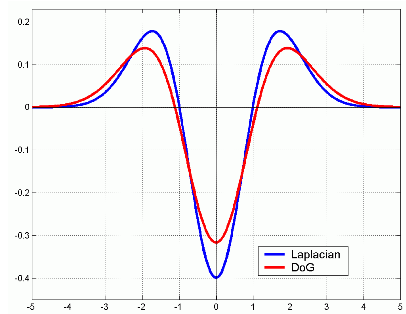
60

Technical Detail

- We can efficiently approximate the <u>scale-normalized</u> Laplacian with a “difference of Gaussians” (DoG):

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

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

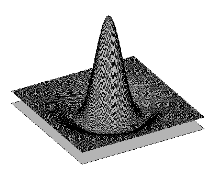


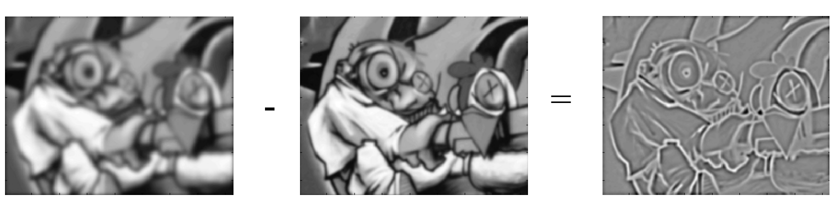
(Since it is an approximation, DoG is generally slightly inferior to LoG)

61

Difference-of-Gaussians (DoG)

- Difference-of-Gaussians as approximation of the scale-normalized LoG
- Advantage
  - No need to compute 2^{nd} derivatives





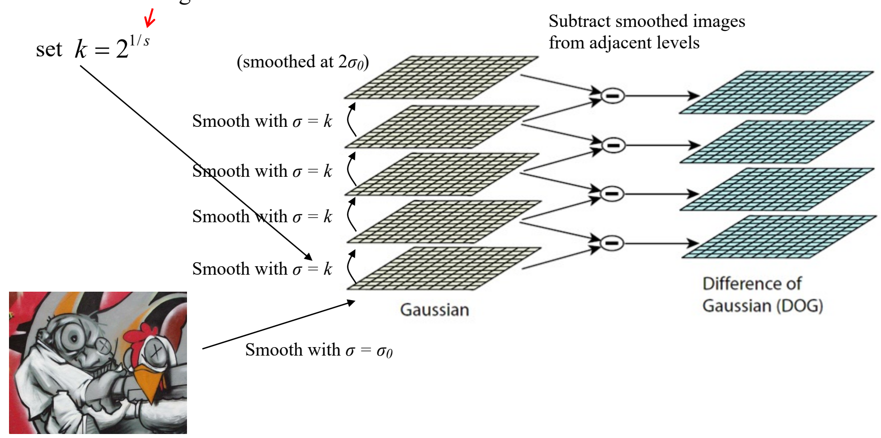
62

DoG "Octave"

Octave = "set of images for which smoothing σ is **doubled**"

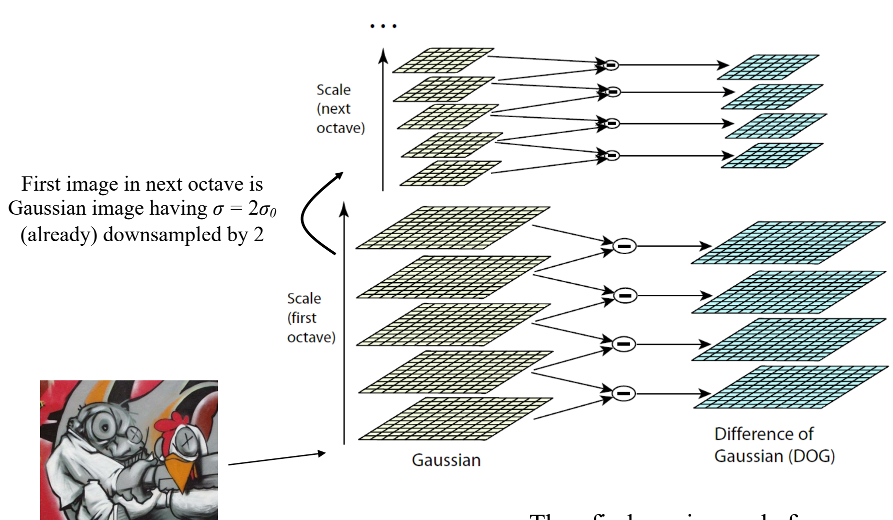
s = desired # of images in DoG "octave"

set $k = 2^{1/s}$



63

DoG Octave Pyramid



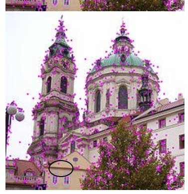
Then find maxima as before

64

Example of Keypoint Detection

DoG extrema

Points remaining after contrast threshold



Points remaining after removing edge responses

65

Topics of This Lecture

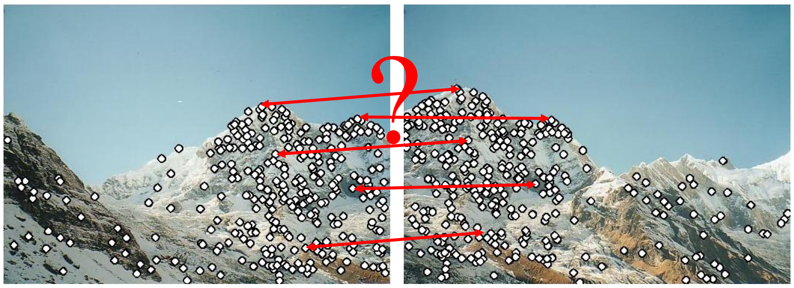
- Local Invariant Features
  - Motivation
  - Requirements, Invariances
- Keypoint Localization
  - Features from Accelerated Segment Test (FAST)
  - Harris
  - Shi-Tomasi
- Scale Invariant Region Selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussians detector
- Local Descriptors
  - Orientation normalization
  - SIFT

66

Local Descriptors

- We now know how to detect points and appropriate scales
- Next question:

How to *describe* them for matching?



Point descriptor should be:
1. Invariant
2. Distinctive

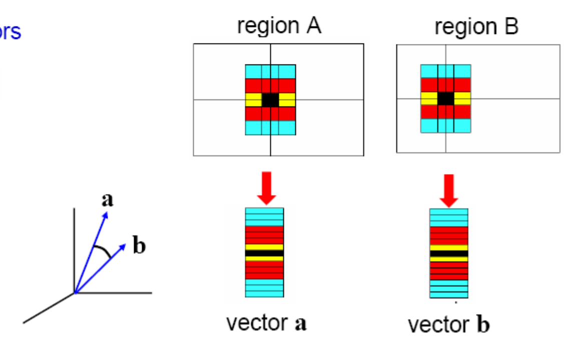
67

Local Descriptors

- Simplest descriptor: rasterized list of intensities within a patch
- To what is this going to be invariant (for normalized patch sizes)?
  - Translation and scale only (not rotation!)

Write regions as vectors

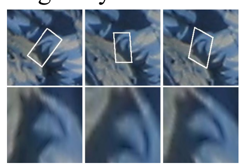
$$\mathtt{A} \rightarrow \mathbf{a}, \ \mathtt{B} \rightarrow \mathbf{b}$$



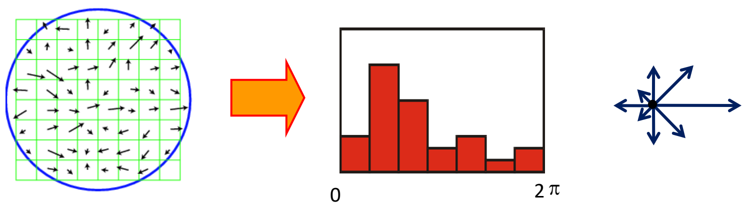
68

Feature Descriptors

- Disadvantage of patches as descriptors:
  - Small deformations can greatly affect matching score



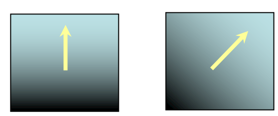
- Solution: "**histogram of gradient directions**"



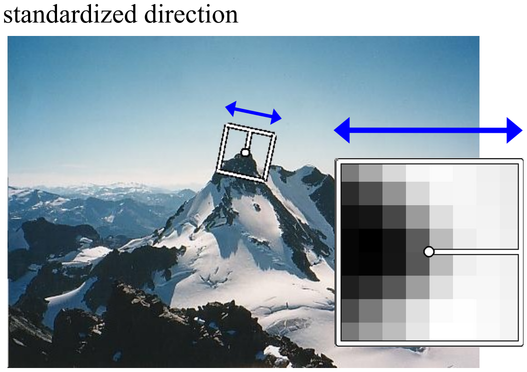
69

Rotation Invariant Descriptors

- Find local gradient directions in patch
- Find <u>dominant</u> direction of overall gradient for the patch



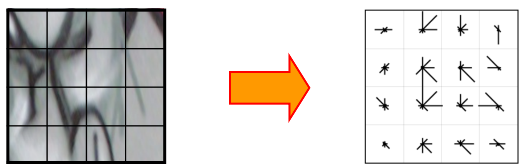
- Rotate patch according to this angle
  - This puts the patches into a standardized direction



70

Feature Descriptors: **SIFT**

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch (for keypoint at chosen scale) **after** standardized direction alignment into 4x4 sub-patches: 16 cells
  - Compute Gaussian weighted histogram of gradient directions (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions
  - Rasterize and normalize to unit vector (enhances invariance to affine changes in illumination)
    - Clip high values and renormalize vector to reduce effects of non-linear illumination



71

Working with SIFT Descriptors

- One image with *n* feature points yields:
  - *n* 128-dimensional descriptors: each one is a collection of histograms of the gradient orientations within a patch
    - [n x 128 matrix]
  - *n* scale parameters specifying the size of each patch
    - [n x 1 vector]
  - *n* orientation parameters specifying the angle of the patch
    - [n x 1 vector]
  - *n* 2-D points giving positions of the patches
    - [n x 2 matrix]

72

Matching Sift Descriptors

- Given two images…
- Match each keypoint independently (and greedily)
  - Employ 128-D descriptors
  - Best candidate match is “closest Euclidean neighbor”
  - **Can discard matches using ratio of closest (“best”) and second closest (“2nd best”) neighbors**
    - Simple measure of how <u>unique</u> is the match
- Can employ more costly (better) approaches to match complete <u>sets</u> of points (e.g., Hungarian algorithm)

73

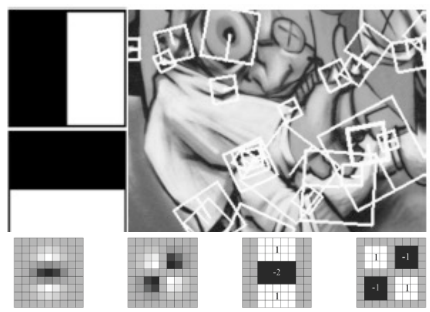
Advantages of SIFT

- Fairly robust matching technique
  - Can handle significant changes in viewpoint
  - Can handle significant changes in illumination
    - Sometimes even day vs. night
  - Fast and efficient—can run in real time
    - Lots of code available

74

Variations and Implementations

- SURF: Fast approximation of SIFT (multiple times faster)
  - Efficient computation by 2-D box filters and integral images
  - Essentially equivalent quality for object identification
- GPU implementations of SIFT and SURF exist



75

Summary

- Corners
  - Image gradient has two dominant directions
  - FAST, Harris, Shi-Tomasi
  - Corner detectors are rotation invariant but not scale invariant
- Automatic scale selection
  - Select patch size yielding maximum value of scale invariant function
  - LoG and DoG detectors
- Need descriptors to match keypoints
  - Use histograms for rotation invariance
  - SIFT, SURF

76