IMAGE INDEXING AND RETRIEVAL

I. IMAGE DESCRIPTION

Ewa Kijak
Linkmedia team - IRISA / INRIA

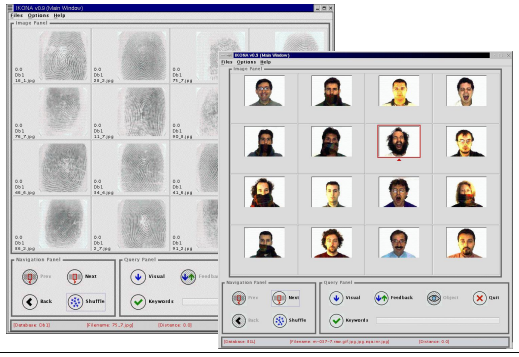
Outline

I. Introduction
1. What are the data and what we want to do?
2. Image comparison
II. Global descriptors
III. Local descriptors

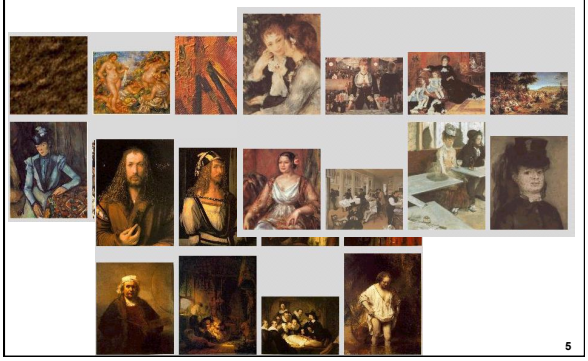
1. What are the data ?

- Let's stick to digital still images
  - Taken with your camera
  - Professional pictures: photo agencies, journals
  - Security databases: fingerprints, portraits
  - Professional collections: products, plants, portraits, scanned books
  - Medical images, satellite aerial images

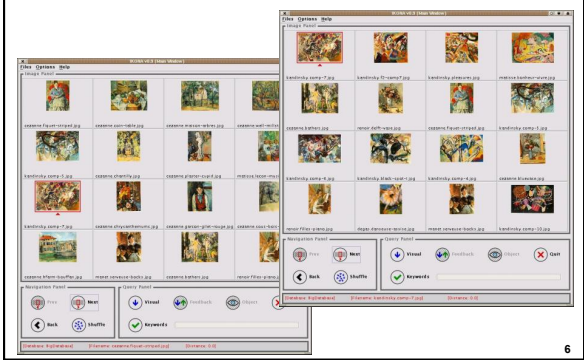
Biometry



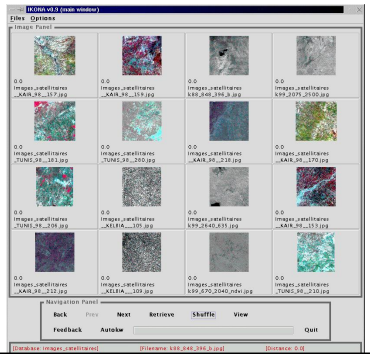
Culture



Art



Satellital



7

How many images?

- Home collection: 100 000 in 30 years
- Professional collection: 1 to billions
- Flick'r: more than two billions
  - Around 2,000 pictures uploaded per minute

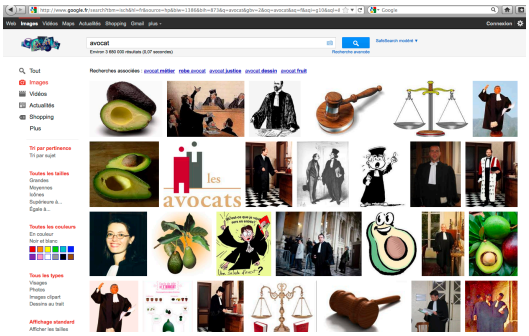
8

What for?

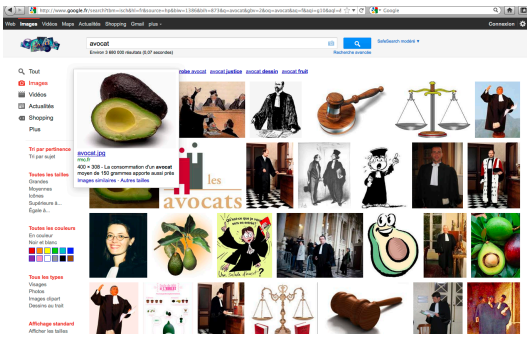
- Compare two images
- Compare an image to a collection: duplicate, extract, transformed copy
- Recognize an object, a person
- Detect an object
- Describe a collection
- Define keywords associated to an image

  - With no text associated to the images

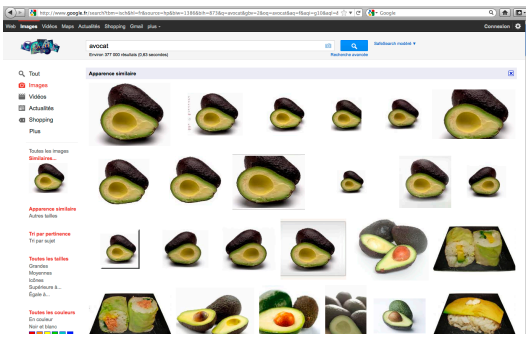
9



10



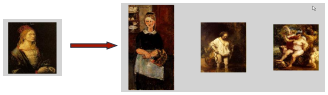
11



12

What for?

- Paradigm: query by example
  - Similar image search/classification



  - Copy detection

Applications

Applications

- Query by humming

Applications

The basic operation: image comparison

- How to compare the content of two images?
  - No keywords, no text…
- How to compare images directly?
- Descriptors: an indirect comparison

2. Comparing images

- Objective distance
  - Mean square error

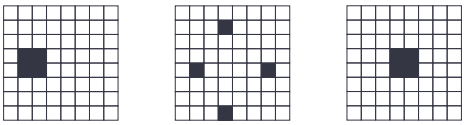
$$MSE = \frac{1}{Nr.Nc}\sum_{i=1}^{Nr}\sum_{j=1}^{Nc}[I_1(i,j)-I_2(i,j)]^2$$

  - Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10\log_{10}\left(\frac{255^2}{MQE}\right)$$

- Perceptual distance
  - Still to be found

An example



- Which are the two closest images?

$$MQE_{1,2} = \frac{6 \times 255^2}{64} = 3048$$

$$MQE_{1,3} = \frac{8 \times 255^2}{64} = 4064$$

- And perceptually?

From images to descriptors

- Image data are numerous, voluminous...
  And not very informative
- One should try to reduce, condense the data:
  -> image descriptors

- Let's translate image similarity
  in term of descriptor similarity

Algorithm

1. One or several descriptors are associated to each image
2. These descriptors are stored and organized in a database
3. The descriptors of the query image are computed
4. These descriptors are compared to those of the database
5. The results are translated in image comparison terms

4 important aspects

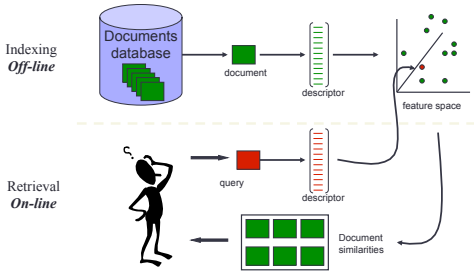
1. Descriptors may be invariant to transformations:
   - Translating or rotating the image will not change the descriptor
   - Same for illumination changes, crops...
2. A distance should be defined for each kind of descriptors: Euclidean, L_1, EMD...
3. Descriptors are high dimensional vectors
   - From 1 to 900 dim!

4 important aspects

4. A search strategy:
   - Retrieving the k nearest descriptors
     - Given a desc., find the k nearest for the distance
     - Problem: how to choose k?
     - Nearest does not mean near.
   - Retrieving the descriptors closer than ε
     - Given a desc., find the desc. Closer than ε
     - Problem: how to choose ε?
     - Should ε be uniform in the desc. space?
     - Can provide 0 or 2 zillions desc.

CBIR

- CBIRS: Content-Based Image Retrieval sytem



CBIR

- Across 2 scientific domains

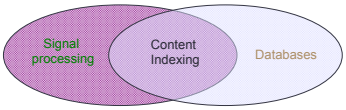


Image descriptors

- Global descriptors

  -> Approximate description of the whole image

  - Basic
    - Color
    - Texture
    - Shape
  - Advanced (Ad-hoc)
    - Faces, fingerprints, iris...

- Local descriptors

  -> Precise description of a part of the image

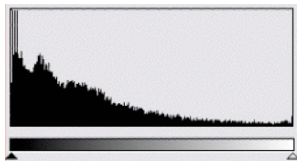
  - Provide very good results!

Outline

I. Introduction
II. Global descriptors
   1. Color
   2. Texture
   3. Shape
III. Local descriptors

Histograms

- Greylevel
  - Each bin: number of pixels of a given greylevel: $h=(h_i)_i$
  - Dimension: 256



  - Normalized histogram: $k=(k_i)_i$ with $k_i = \frac{h_i}{\sum_j h_j}$

How to describe color?

- What is the definition of a histogram for a color image?
- What is then the histogram dimension?
- How many pixels in a image of size 1920×1080 ?
- Conclusion?

How to describe color?

- Image color histogram
  = number of pixel of each color

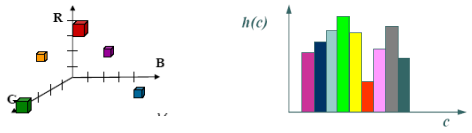
$$(r, g, b) \mapsto nb\{pixel = (r, g, b)\}$$

- 256^3 = 16,777,216 bins
  for 512x512 = 262 144 pixels!
- Solutions:
  - 3 monocolor histograms: 3x256
    - Correlation between components is lost

How to describe color?

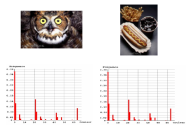
- Solutions:
  - Vectorial quantization of the color space
    - 16 bins per channel: $16^3 = 4096$

$$\forall c \in C, \quad h(c) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \delta[f(i,j) - c] \qquad \sum_{c \in C} h(c) = 1$$



How to describe color?

- +++
  - Fast and easy to compute
  - Integration over the image = robustness
- ---
  - No geometry
  - No invariance to point of view (translation, rotation), (scaling without normalization), illumination modifications
  - Which distance?



Contenu visuel différent

mais histogrammes similaires

Distances between histograms

- 2 normalized histograms $h = (h_i)_{i \in 1,n}, \; l = (l_i)_{i \in 1,n}$
  - Euclidean distance (L_2)

$$d_{L2}(h,l) = \sqrt{\sum_{i=1}^{n} (h_i - l_i)^2} = EQM(h,l)$$

  - Manhattan distance (L_1)

$$d_{L1}(h,l) = \sum_{i=1}^{n} |h_i - l_i|$$

  - Max distance (L_∞)

$$d_{\max}(h,l) = \max_i |h_i - l_i|$$

  - Histogram intersection

$$d_{inter}(h,l) = 1 - \sum_{i=1}^{n} \min(h_i, l_i)$$

Distances between histograms

- Example 1 :

I_1 : 10 x 10 black pixels

I_2 : 5 x 5 black pixels

  - What is the distance L_1 between h_1 and h_2 ?
- Exemple 2 :

  - What is the distance L_1 between h_1, h_2 and h_3 ?

Problem?

- Which are the two closest ones?

  $D(H_1, H_2) = d(H_1, H_3)$!!!

- These distances do not take bin distances into account

Advanced distances

- Quadratic distance

$$d(l,h) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (h_i - l_i) Q_{ij} (h_j - l_j)}$$

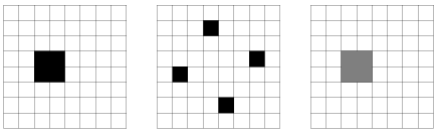
Q: color similarity matrix

- EMD: Earth Movers' distance

$$d_{emd}(h,l) = \min_{i,j} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot c_{ij} \right)$$

= minimal effort to transform (h_i) into (l_i) -> f_{ij}
  - Distances between bins are taken into account -> c_{ij}
  - Requires an optimization process

Problems?



- Which images are the two closest ones?
  - According to histograms?
  - According to you?

Advanced histograms

- Weighed histograms:
  - the vote of each pixel is weighed by its neighborhood: Laplacian, local entropy, gradient...

$$\forall c \in C, \quad h'(c) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} w(i,j)\ \delta[f(i,j) - c]$$

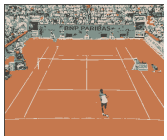
  - Greatly improves the results
  - Combines different type of information while not changing the dimension of the histogram
  - Most expensive to compute
  - Which distance?

Dominant colors

- Vector of K dominant colors:
  - Obtained by K-means
- For each dominant color c_i:
  - Color coordinates
  - Percentage p_i
  - Spatial Coherence C_i
  (Defined on a neighorhood of each point)
- Distance:

$$d(F_1, F_2) = \sum_{i=1}^{K} ({p_{1i}}^2 + {p_{2i}}^2) - \sum_{i=1}^{K} \sum_{j=1}^{K} 2 a_{ij} p_{1i} p_{2j}$$

Original image



4 dominant colors image

2. Texture

- How to characterize?

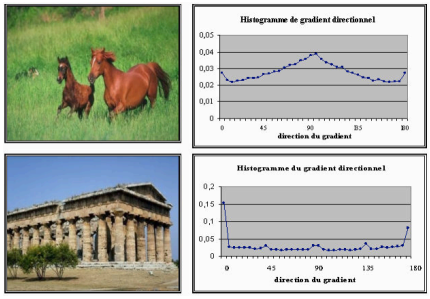
- Texture descriptors:
  Fractals, wavelets...
  - Which scale? which distance?
  - Which descriptor for which texture?

3. How to extract a shape?

- Edge extraction

How to describe a shape?

- Orientation histogram



Exercise : histograms

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	0
2	3	4	5	6	7	0	1
3	4	5	6	7	0	1	2
4	5	6	7	0	1	2	3
5	6	7	0	1	2	3	4
6	7	0	1	2	3	4	5
7	0	1	2	3	4	5	6

I1

0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
4	4	4	4	6	6	6	6
4	4	4	4	7	7	7	7
5	5	5	5	7	7	7	7
5	5	5	5	7	7	7	7
1	1	1	1	3	3	3	3
1	1	1	1	3	3	3	3

I3

1	1	1	3	5	6	7	7
1	1	3	3	5	7	7	7
1	1	3	5	7	7	7	5
1	3	5	7	7	7	5	2
6	5	7	7	7	5	2	2
6	7	7	7	5	2	2	2
7	7	7	5	2	2	0	0
7	7	6	2	2	0	0	0

I5

0	2	4	6
2	4	6	0
4	6	0	2
6	0	2	4

I2

1. Compute the normalized histograms of these images.
2. Compare these images with:
   - L1 distance
   - Histogram intersection distance
   - EMD distance

Exercise: color histogram

- Let I be an indexed color image, which colors are indexed in the following table:

	R	G	B
1	74	104	137
2	144	223	30
3	123	29	57
4	84	50	159
5	151	143	194
6	127	150	12
7	111	174	171
8	68	227	91
9	161	128	99
10	103	89	211
11	234	98	149
12	11	235	123
13	122	241	56
14	67	87	214
15	105	95	30
16	71	187	72
17	32	66	219
18	198	127	216
19	54	67	134
20	28	133	253

1	12	1	17	2	7	7	8
10	1	13	13	2	2	7	7
10	10	1	13	13	2	9	7
14	14	10	3	13	13	20	20
14	14	14	3	6	6	20	20
15	5	5	18	18	6	6	20
15	11	5	19	18	18	6	20
15	11	5	19	19	19	16	16

1. Build the color histogram of dimension 8.
2. What should be modified to obtain an histogram of size 64?

Histogram of Oriented Gradient (HOG)

- counts occurrences of gradient orientation in localized portions of an image
  - image divided into small connected regions called **cells**
  - a histogram of gradient directions is computed for the pixels within each cell
- the descriptor is the concatenation of these histograms.
- the local histograms are normalized using a a measure of the intensity across a larger region of the image, called a **block**.

HOG - Algorithm implementation

1. Gradient computation
2. Orientation binning
3. Block normalization

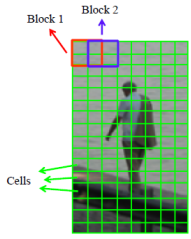
HOG - Orientation binning

Goal : creating the cells (bins) of the orientation-based histograms

- The cells can either be rectangular or radial in shape\
- The histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees (depending on whether the gradient is "unsigned" or "signed")
- Each pixel within the cell casts a weighted vote for a histogram channel (bin)
  - Pixel contribution can be the gradient magnitude (so stronger gradients have a bigger impact on the histogram)
  - The contribution is splitted between the two closest bins. For example, if a gradient vector has an angle of 85 degrees then 1/4th of its magnitude is added to the bin centered at 70 degrees, and 3/4ths of its magnitude to the bin centered at 90.

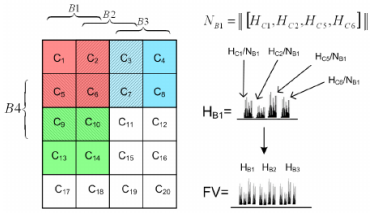
HOG - Block normalization

- To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected **blocks**.



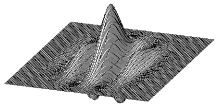
HOG - Block normalization

- The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions.
- These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor.

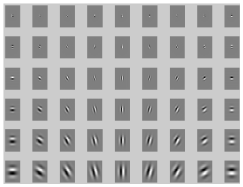


GIST

- What's Gabor filters ?
  - A Gabor filter is a linear filter used for edge detection
  - Its impulse response is defined by a sinusoidal wave multiplied by a Gaussian function



- A Gabor filters bank is a collection of Gabor filters at different scales and orientations



GIST

- Given an input image, a GIST descriptor is computed by :
  1. Convolving the image with N Gabor filters at n_1 scales and n_2 orientations ($n_1 \times n_2 = N$), producing N feature maps of the same size of the input image.
  2. Dividing each feature map into 16 regions (by a 4 x 4 grid), and then averaging the feature values within each region.
  3. Concatenating the 16 averaged values of all N feature maps, resulting in a 16xN-dim GIST descriptor.
- Intuitively, GIST summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description of the scene.

Outline

I. Introduction
II. Global descriptors
III. Local descriptors
  1. Definition and properties
  2. Support region extraction
  3. Support region description
  4. Applications

1. Local Description

- Extract features in the images
  - Characteristic points (and an area around)
  - Edges or lines
  - Regions

- Associate a descriptor to each of them
- Compare database desc. with query desc.
- Recognition = accumulation of local evidences
- Conclusion: features are as important as descriptors

Local Description

Local Descriptors

- 3 possible usages
  - Image retrieval from a database
  - Object recognition
  - Object class recognition
- 3 steps
  - Extraction of robust features: interest points
  - Computation of descriptors: SIFT
  - Comparaison of the descriptors: ε-range or k-NN

55

Invariance

- What is it?
  - The descriptor does not change when the image is transformed
- Is it needed?
  - To recognize objects in various conditions
  - To recognize copied and transformed images

56

Invariance

- Scaling

57

Invariance

- Illumination variation

58

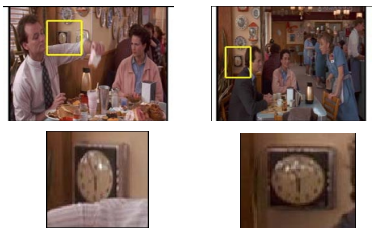
Invariance

- Viewpoint variation

59

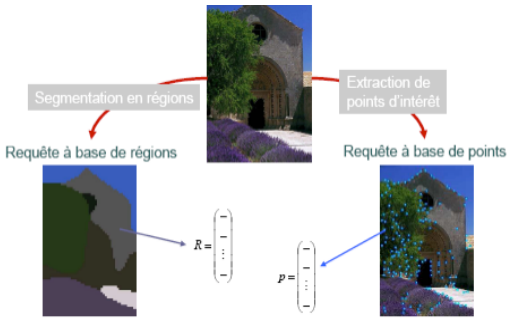
Invariance

- Occlusions



60

2. Support region extraction



Regions

- Regular grid

Partial query Search result

Regions

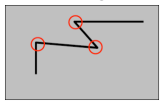
- Color segmentation

Original Images

Segmented Images

Point detection

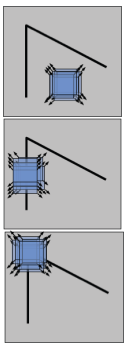
- Interest point:
  - a point different from its neighborhood (Harris-Stephen)



  - based on self-correlation idea

Point detection

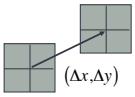
- Homogeneous region
  - No changes in any direction
- Edge
  - No changes along the edge direction
- Corner :
  - Significative changes in all directions



Point detection

Signal self-correlation around point (x,y) and a displacement $(\Delta x,\Delta y)$

$$A(x,y) = \sum_{(x_k,y_k)\in W(x,y)} \left[I(x_k,y_k) - I(x_k+\Delta x, y_k+\Delta y) \right]^2$$



$A(x,y)$
- Small in all directions → uniform region
- Large in one direction → contour
- Large in all directions → interest point

Point detection

- Signal self-correlation around point (x, y)

$$S(x,y) = \sum_{u,v \in \mathcal{W}} w_{u,v} \left[I(x+u, y+v) - I(x,y) \right]^2$$

- First order approximation

$$I(x+u, y+v) = I(x,y) + uI_x + vI_y + O(u^2, v^2)$$

$$S(x,y) = \sum_{u,v \in \mathcal{W}} w_{u,v} \left[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2 \right]^2$$

- Factorization

$$S(x,y) = \sum_{u,v \in \mathcal{W}} w_{u,v} \quad (u,v) \quad M \quad (u,v)^T$$

with

$$M = \begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$$

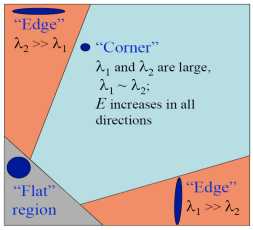
Point detection

$$M = \begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$$

- Against noise, Gaussian filtering
  - of the derivatives σ $\quad I_x \mapsto G(\sigma) \otimes I_x$
  - of the matrix **M** itself $\bar{\sigma}$ $\quad M \mapsto G(\bar{\sigma}) \otimes M$
  - and should be adapted to point scale ($\sigma = \bar{\sigma}$)
- Maximizing **M** = large eigenvalues λ_1 and λ_2
  - Karhunen Loeve Transform extractor: maximizing λ_1 and λ_2 requires to solve a second degree equation
  - Harris extractor: maximizing $\lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)$ = det **M** - k tr **M**

Harris point detection

- Harris criteria is computed on each point
- This criteria:
  - depend only on **M** eigenvalues
  - is large for corners
  - is negative for edges
  - is small for homogeneous regions
- Only local maxima are kept



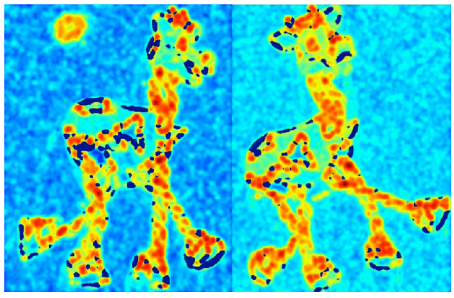
Thresholding

- Keep only local maxima
- Harris score must be thresholded
  - Score > threshold
  - N largest values
  - Score > max(score) * 0.01
- Adaptative methods
  - In each window of size X, Y
  - Apply on of the previous criteria
  - Move the window of n pixels

Harris point detection

Harris point detection

- Harris score



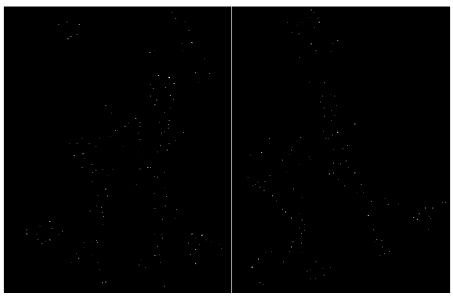
Harris point detection

- Thresholded points

73

Harris point detection

- Local maxima of criterion



74

Harris point detection

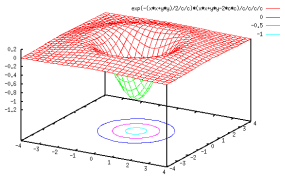
75

Harris point detection

76

Laplacian of Gaussian (LOG)

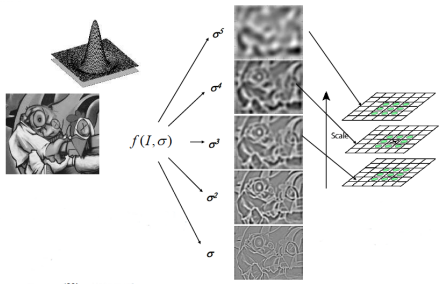
$$LOG = G_{xx}(\sigma) + G_{yy}(\sigma)$$



77

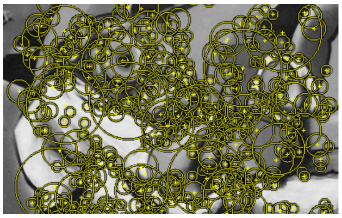
LOG detection

- Maxima and minima detection in Laplacian scale-space



78

LOG keypoint detection

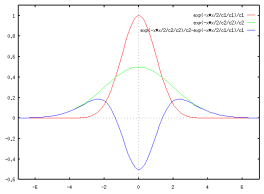


79

DOG: Difference of Gaussians

- Approximation of LOG by difference of gaussians:

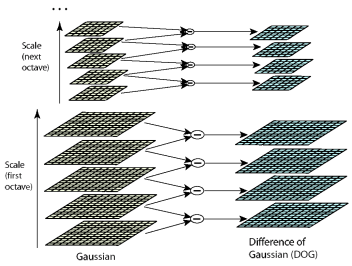
$$DOG = G(k\sigma) - G(\sigma)$$



80

DOG detection

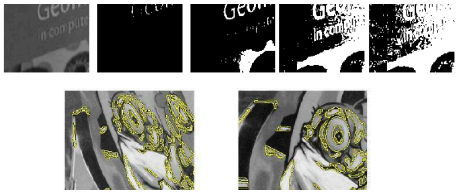
- Fast computation
- Scale-space is processed one a time



81

MSER: Maximally Stable Extremal Regions

- Based on watershed algorithm
  - Consecutive image thresholding by all thresholds
  - Maintain list of Connected Components
  - Regions = Connected Components
  with stable area (or some other property) over multiple thresholds selected



82

3. Support Region description

- We want descriptors invariant to
  - Image geometric transforms: translation, rotation, scaling
  - 3D geometric transforms: zooming, viewpoint
  - Photometric transforms: illumination

83

Descriptors - history

- Normalized cross-correlation (NCC) [~ 60s] (patches)
- Gaussian derivative-based descriptors
  - Differential invariants [Koenderink and van Doorn'87]
  - Steerable filters [Freeman and Adelson'91]
- Moment invariants [Van Gool et al.'96]
- SIFT [Lowe'99]
- Shape context [Belongie et al.'02]
- Gradient PCA [Ke and Sukthankar'04]

84

Descriptors - history

- Efficient descriptors
  - SURF descriptor [Bay et al.'08] – U-SURF
  - DAISY descriptor [Tola et al.'08, Windler et al'09]
- (recent) Extensions to SIFT
  - Color SIFT [Sande et al.'10]
  - RootSIFT [Arandjelovic et Zisserman'12]
    - Normalizing SIFT with square root transformation

Descriptors - history

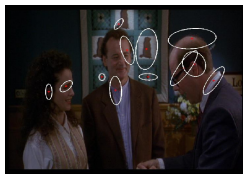
- Fast And Compact Descriptors
  - (Very) large scale applications
    - -> memory issues, computation time issues
  - Mobile phone applications
- Compact binary descriptors:
  - Comparison of pairs of intensity values
  - LBP (Local Binary Patterns) [Heikkila et al.'09]
  - BRIEF (Binary Robust Independent Elementary Features) [Calonder et al.'10]
  - ORB (Oriented FAST and Rotated BRIEF)[Rublee et al.'11]
  - BRISK (Binary Robust Invariant Scalable Keypoints) [Leutenegger et al.'11]

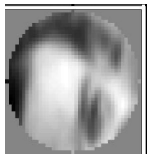
Descriptors - history

- And also:
  - FREAK (Fast Retina Keypoint)[Alahi et al. '12]
  - CARD (Compact and Realtime Descriptor)
  - LDB (Local Difference Binary) [Yang et al.'12]
- More robust:
  - LIOP (Local Intensity Order Pattern)[Wang at al.'11]

The SIFT Descriptor

- Step 1: point extraction + gradient orientation
- Step 2: normalization of the surrounding area
- Step 3: descriptor computation

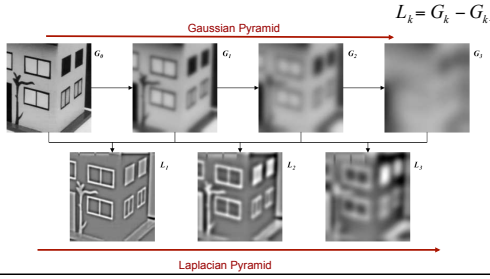




The SIFT Descriptor

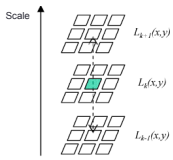
- Point detection
  - DOG filter ≈ Laplacian approximation

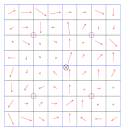
$$G_k = I \otimes G(k\sigma)$$
$$L_k = G_k - G_{k-1}$$

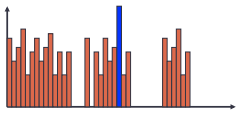


The SIFT Descriptor

- Keypoint localization
  - Detect maxima and minima of DOG in scale space
- Keypoint orientation
  - Compute gradient on the neighborhood of each point



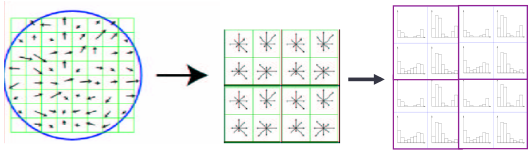




Dimension : 36 (from 0 to 360°)

The SIFT Descriptor

- Descriptor computation
  - In the **normalized** area
    - Compute the gradient at each pixel
    - Quantifie the gradient orientation in 8 values
    - Split the area in 16 parts
    - Histogram gradient orientations in the 16 sub-regions (weighted by gradient magnitude value)
  - Descriptor size = 128 = 16 histo x 8 dim



91

The SIFT Descriptor

- Invariance
  - Translation
    - Description centered on a point
  - Rotation
    - Histograms computed relatively to a principal orientation
  - Scale change
    - Keypoints detected on scale space
  - Illumination changes
    - Based on gradient orientation
    - Vector normalization

92

4. Applications

- 3 prototypical applications
  - Recognizing a particular object
  - Recognizing objects
  - Recognizing classes of objects

93