Content-based Image and Video Retrieval

Fall 2012/2013

Visual Descriptors – cont. & Image Segmentation

09.10.2012

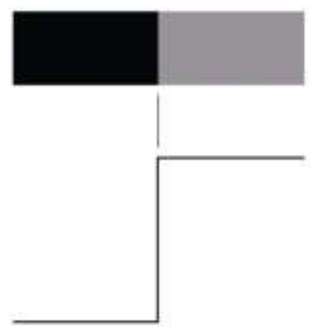
cv:hci Computer Vision for Human-Computer Interaction Research Group
Universität Karlsruhe (TH) Research University · founded 1825
Fakultät für Informatik

- Last week
  - Color descriptors
  - Texture descriptors
- This week
  - Local descriptors
  - Segmentation

- Basics: Edge Detection

Edge Detection

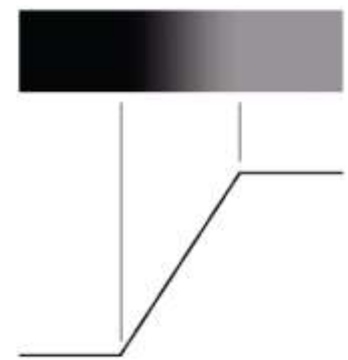
- Edge:
  - An edge is a set of connected pixels that lie on the boundary between two regions
- An ideal edge:
  - A set of connected pixels, each of which is located at an orthogonal step transition in gray level



Edge Detection

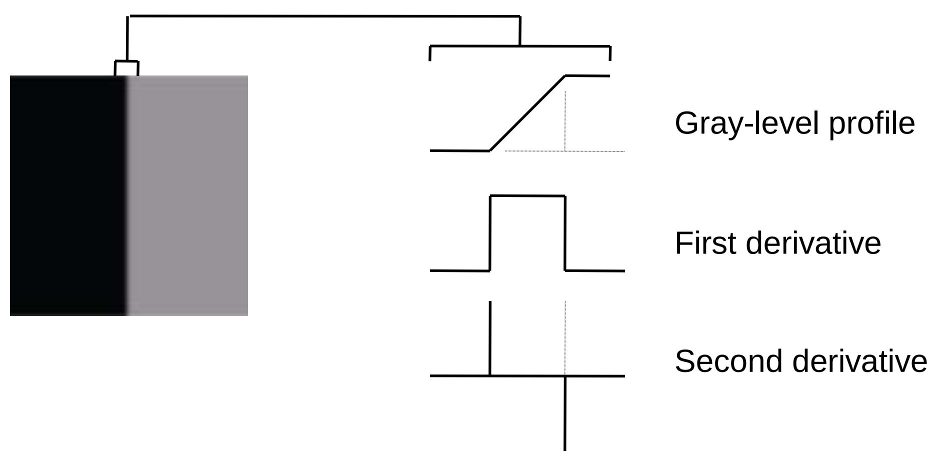
- In practice, edges are blurred due to optics, sampling, and other image acquisition imperfections
  - "Ramp"-like profile
  - Degree of blurring is determined by:
    - Image acquisition systems
    - Sampling rate
    - Illumination conditions
  - An edge point is any point contained in the ramp, an edge is a set of such connected points

Model of a ramp digital edge



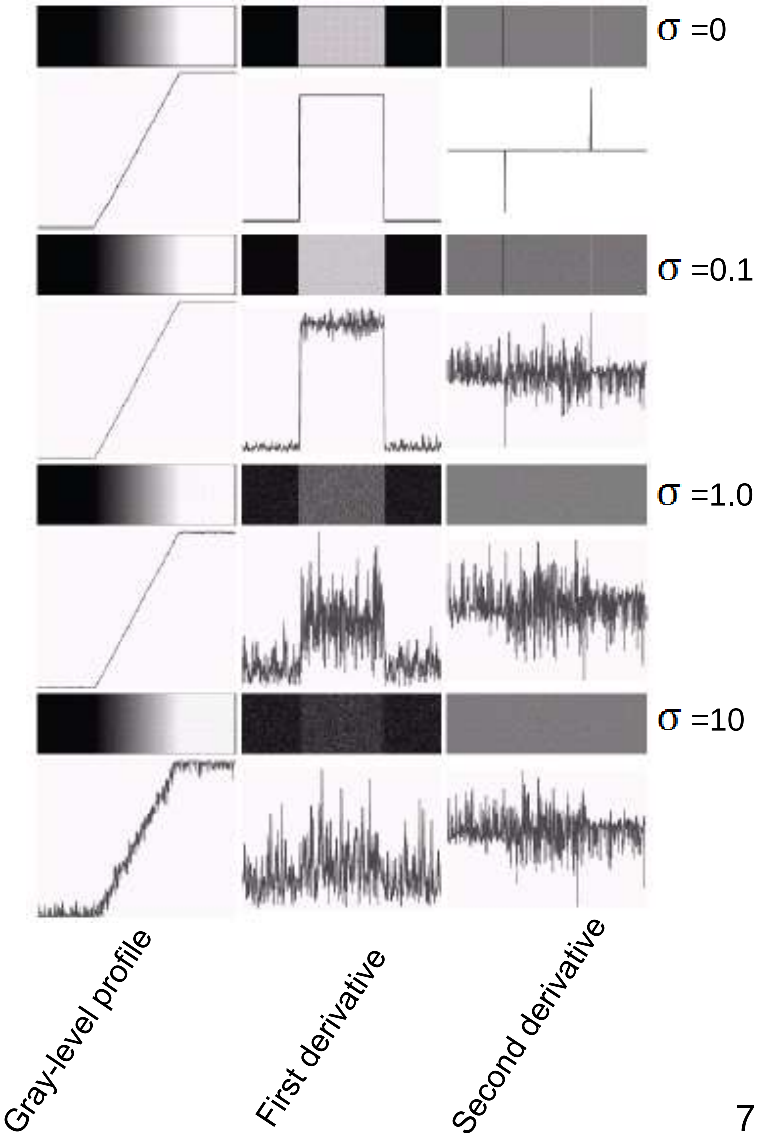
Edge Detection

- The magnitude of the first derivative : detect the presence of an edge
- The sign of the second derivative : determine on which side of an edge
- The “zero crossing property” of the second derivative



Edge Detection

- A ramp edge corrupted with increasing additive Gaussian noise, $\mu = 0$, $\sigma = 0, 0.1, 1.0$ and 10.0
- Derivative is very sensitive to noise => image smoothing



Edge Detection

- Edge point
  - A point in an image is defined to be an *edge point* if its two-dimensional first-order derivative is larger than a specified threshold
- Edge segment
  - A set of connected edge points according to a predefined criterion of connectedness
- Edge
  - Assemble short edge segments into longer edges

Gradient Operators

- First-order derivatives of an image are based on various approximations of the 2-D gradient

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- The magnitude of the gradient vector is often referred as the gradient

$$\nabla f = mag(\nabla f) = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

- Usually, magnitude is approximated by absolute values

$$\nabla f \approx |G_x| + |G_y|$$

Gradient Operators

- The direction of the gradient vector

$$\alpha(x,y) = tan^{-1}\left(\frac{G_y}{G_x}\right)$$

- The direction of an edge at (x,y) is perpendicular to the direction of the gradient vector at the point

The Gradient

- Robert cross-gradient operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$G_x = (z_9 - z_5) \text{ and } G_y = (z_8 - z_6)$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

-1	0
0	1

0	-1
1	0

- Sobel Operator

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

-1	0	1
-2	0	2
-1	0	1

G_x

-1	-2	-1
0	0	0
1	2	1

G_y

The Gradient

- Prewitt Operator

$$\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_2 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

-1	0	1
-1	0	1
-1	0	1

G_x

-1	-1	-1
0	0	0
1	1	1

G_y

Examples

Original image

$|G_y|$

$|G_x|$

$|G_x| + |G_y|$

Laplacian

- The Laplacian is a second-order derivative

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Digital approximation of the Laplacian

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$$

0	-1	0
-1	4	-1
0	-1	0

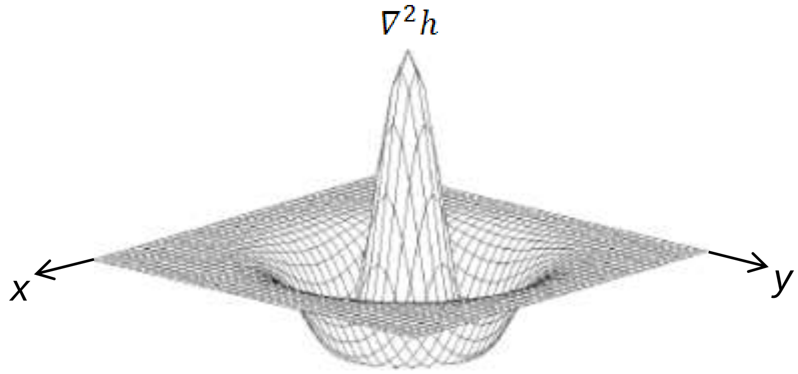
-1	-1	-1
-1	8	-1
-1	-1	-1

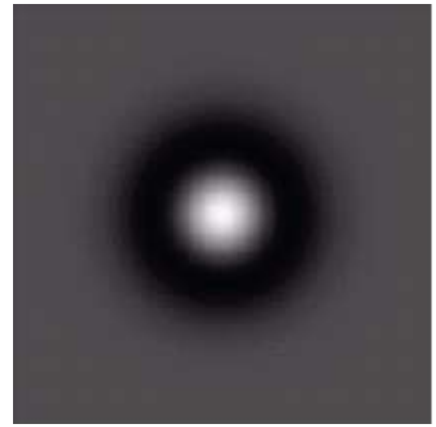
Laplacian of Gaussian

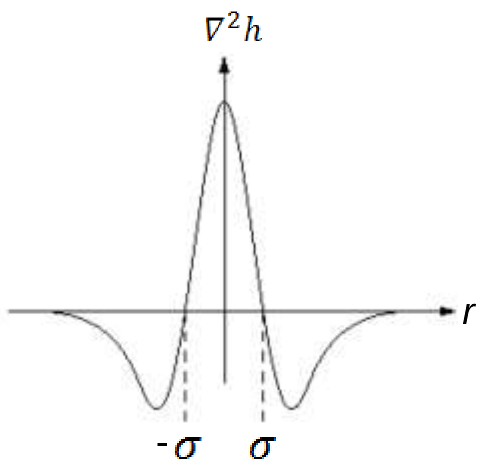
- The Laplacian can not be used in its original form for edge detection due to:
  - Sensitive to noise
  - The magnitude of the Laplacian produces double edges
  - It is unable to detect edge direction
- Laplacian of Gaussian(LoG): combining the Laplacian with Gaussian smoothing
  - Gaussian function
    - Lowpass filter, noise reduction
  - LoG :
    - Highpass filter, abrupt change (edge) detection

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad r^2 = x^2 + y^2$$

Laplacian of Gaussian





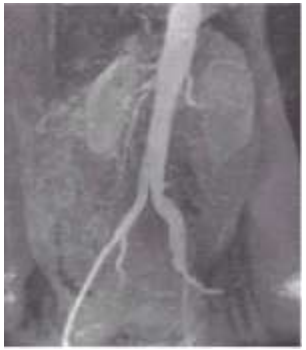


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

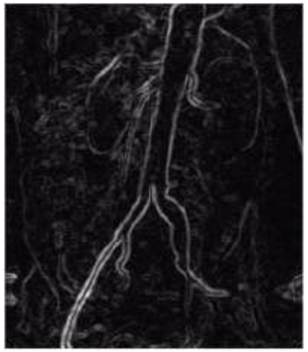
5x5 approximation mask

Laplacian of Gaussian

- Comparison of LoG and gradient operator (Sobel)
  - The edges in the zero-crossing image are thinner than the gradient edges
  - The edges determined by zero crossings form numerous closed loops (spaghetti effect)
  - The computation of zero-crossings presents a challenge



Original image



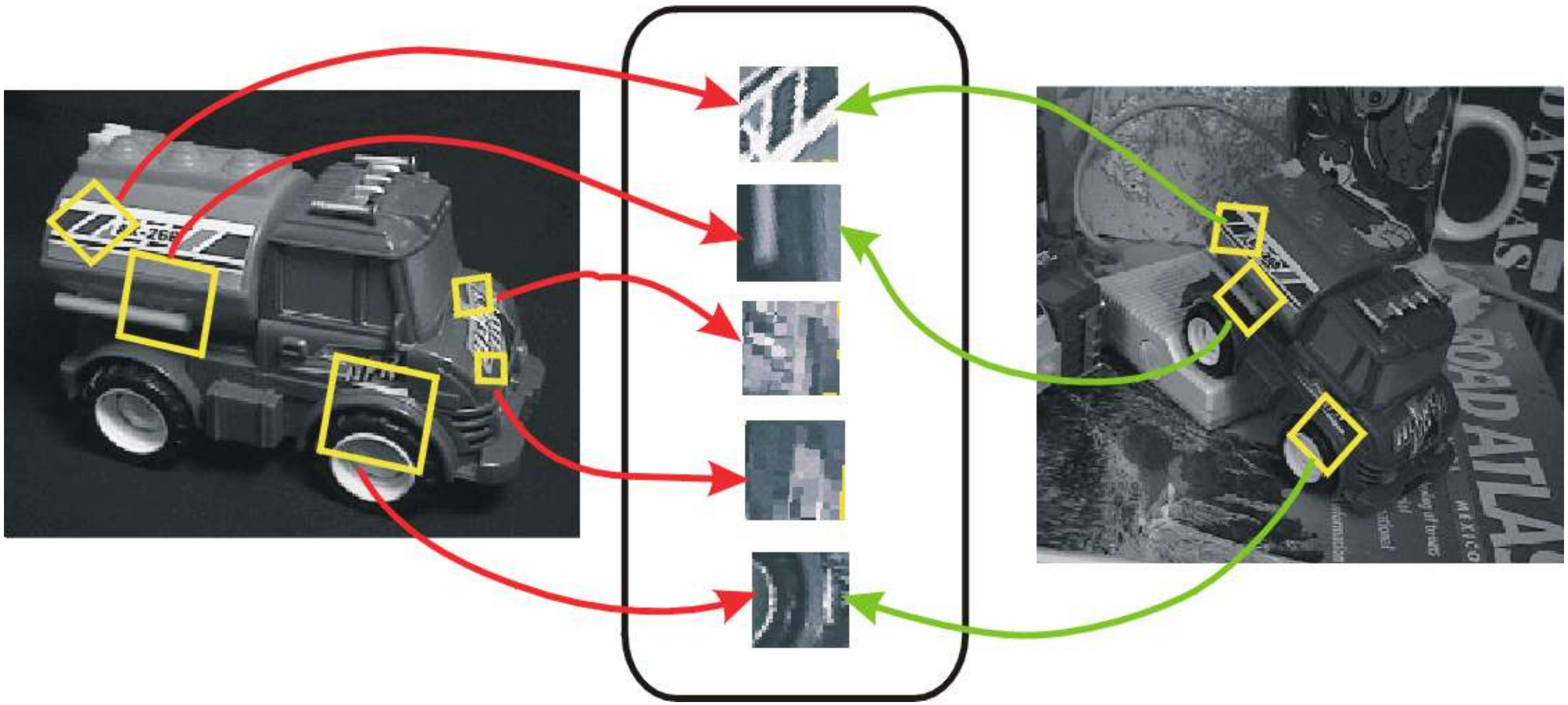
Sobel gradient

Zero-crossings of LoG

- Local Descriptors

Invariant Local Feature

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

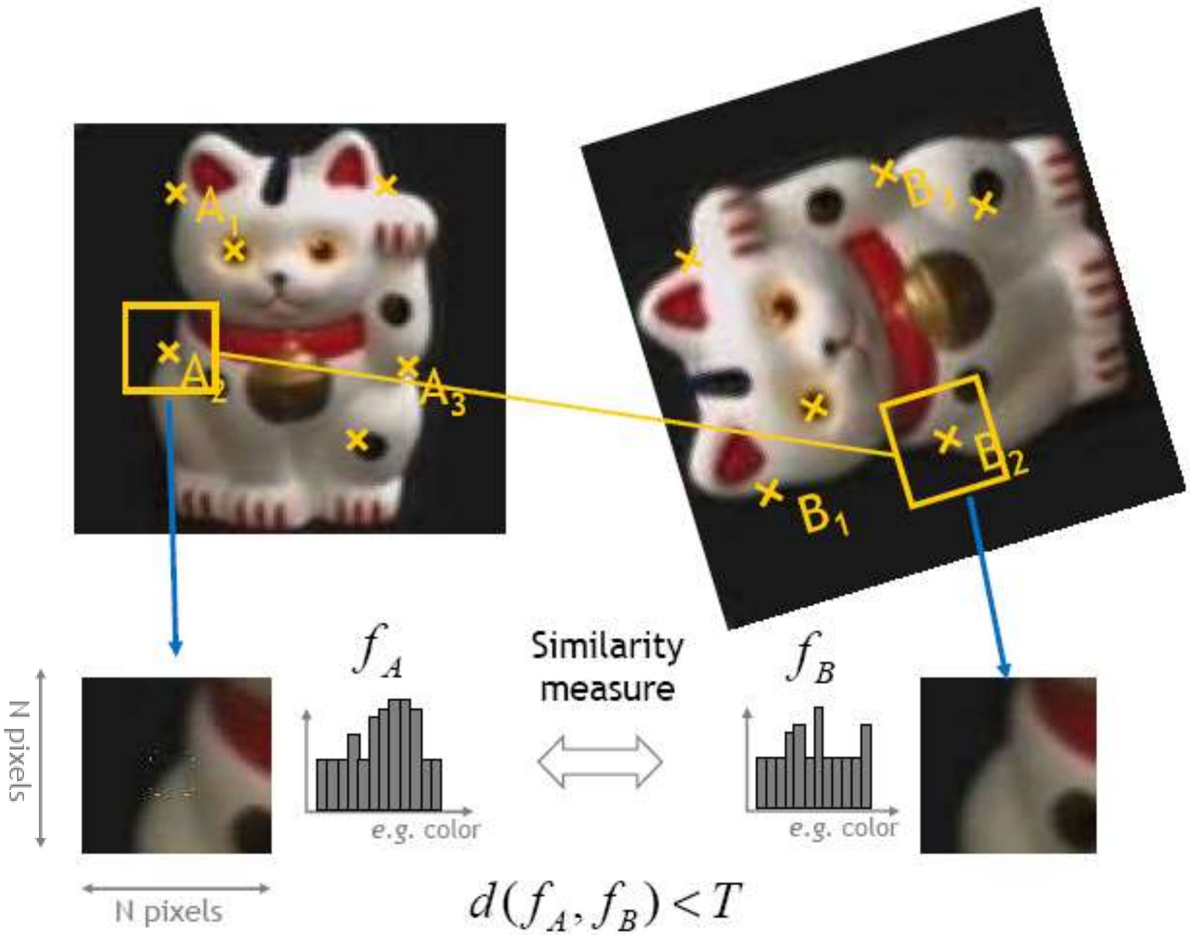


local feature patches

Components of local feature

- Key or interest points
  - Specify repeatable points
  - x-,y-position and scale
  - e.g. corners, blobs
- Local (key point) descriptors
  - Define the feature representation around an interest point
  - e.g raw pixels or a histogram of gradient in the neighborhood of a key point

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

Keypoint Detectors

- 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]
  - Dense Sampling
  - Others…
- Reference site:
  - http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

Keypoint Localization

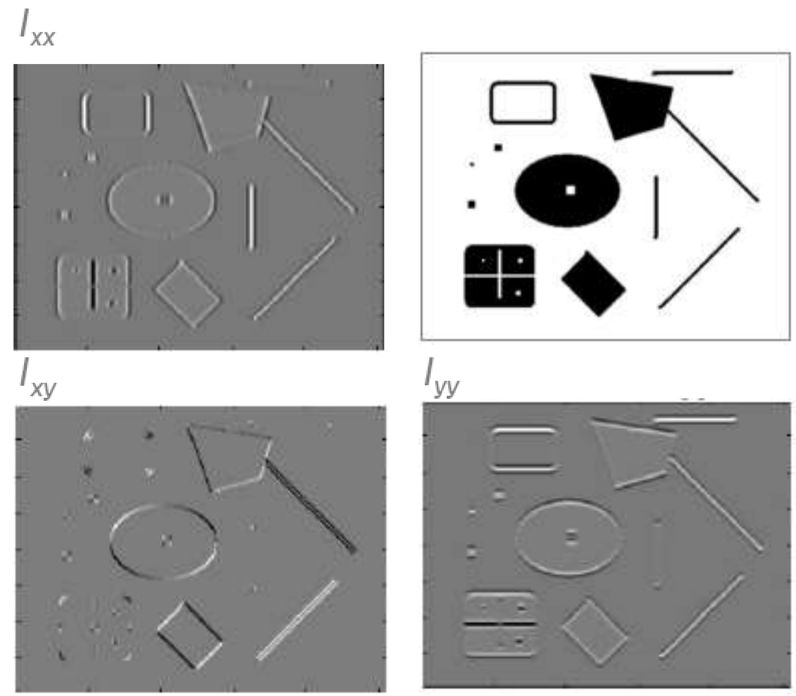
- Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

  => *Look for two-dimensional signal changes*

Hessian Detector [Beaudet78]

- Hessian determinant

$$Hessian\,(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

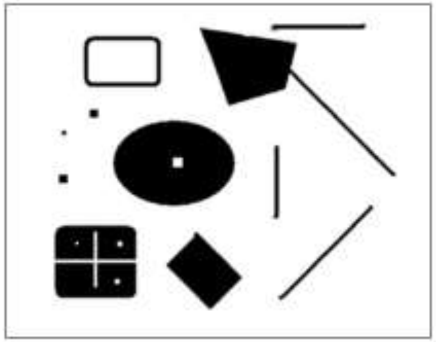


Intuition: Search for strong derivatives in two orthogonal directions

Harris Detector [Harris88]

- Second moment matrix(autocorrelation matrix)

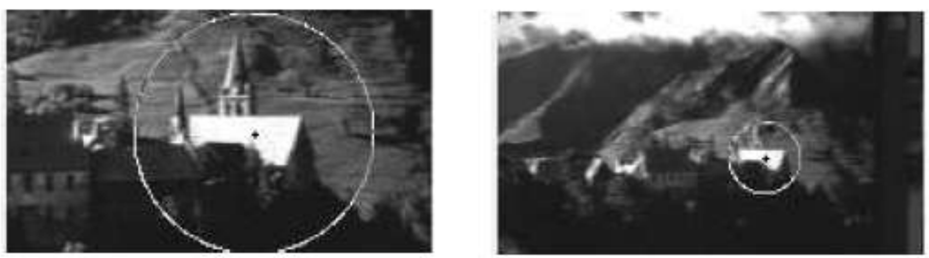
$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$



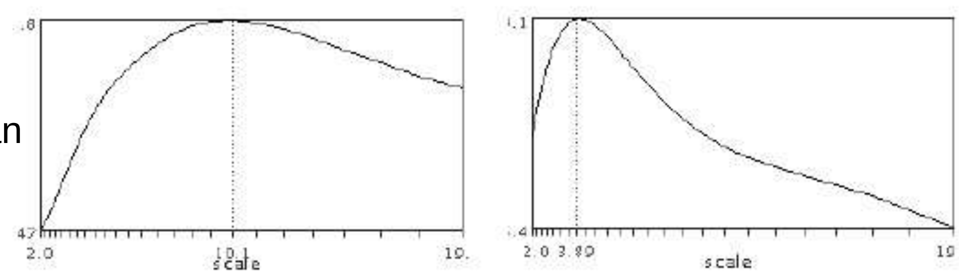
Intuition: Search for local neighborhoods where the image content has two main directions (eigenvectors)

Scale Selection

- Scale selection principle (T. Lindeberg ’94)
  - In the absence of other evidence, assume that a scale level, at which (possibly non-linear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data.
- Selection of points at characteristic scale in scale space



Laplacian

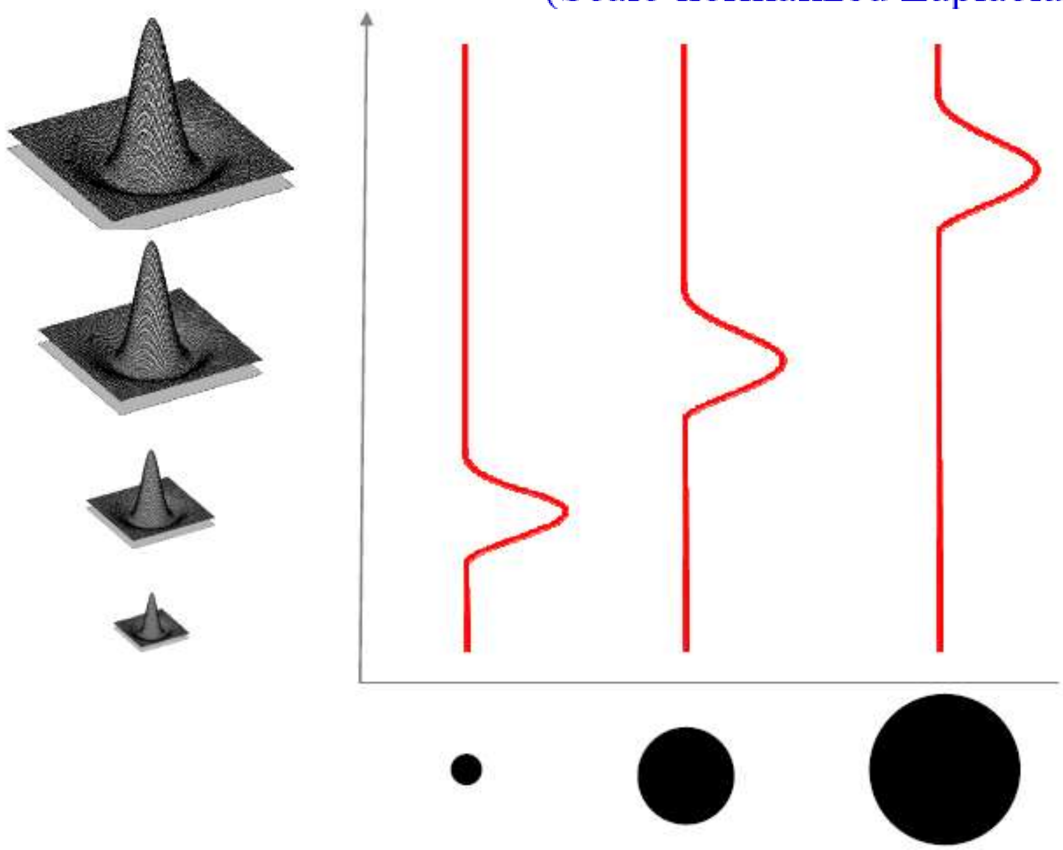


Chacteristic scale :
- maximum in scale space
- scale invariant

Scale Invariant Detection

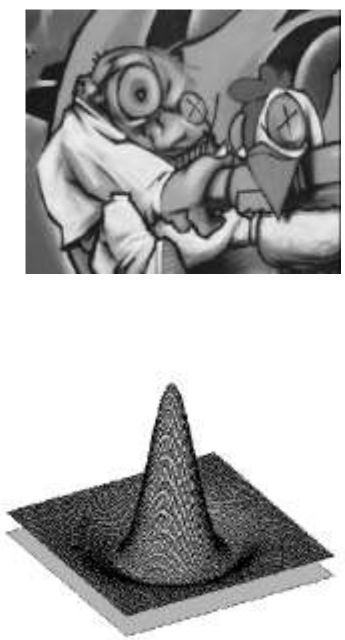
- Kernels for determining scale
  - Laplacian-of-Gaussian $L = \sigma^2 \left(G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma) \right)$

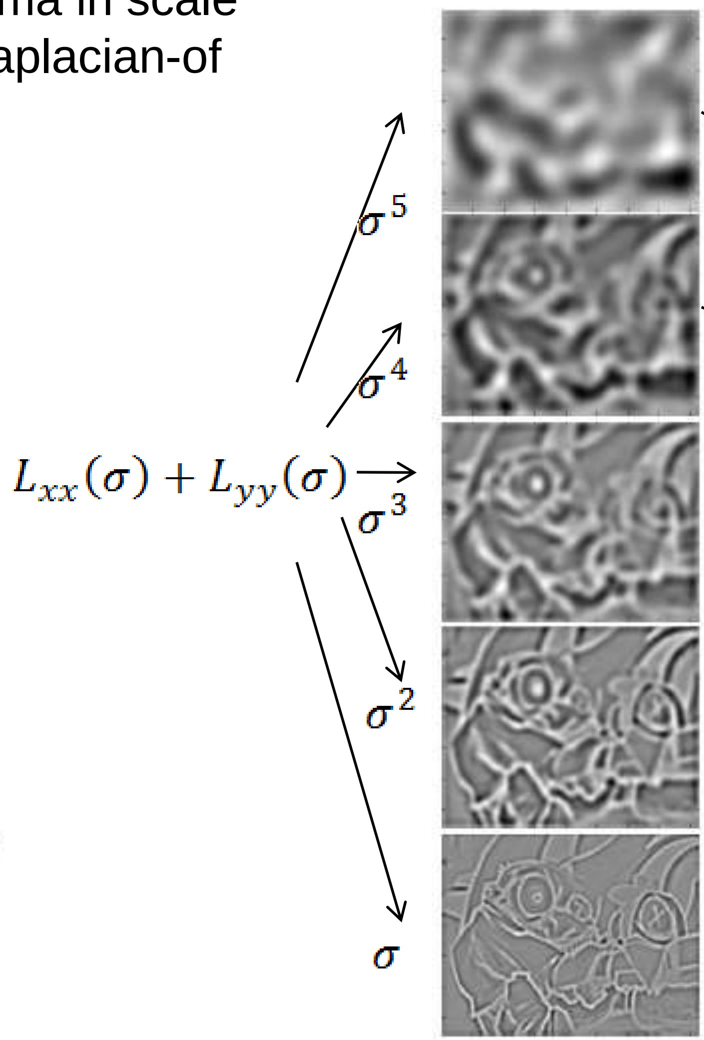
(Scale-normalized Laplacian)

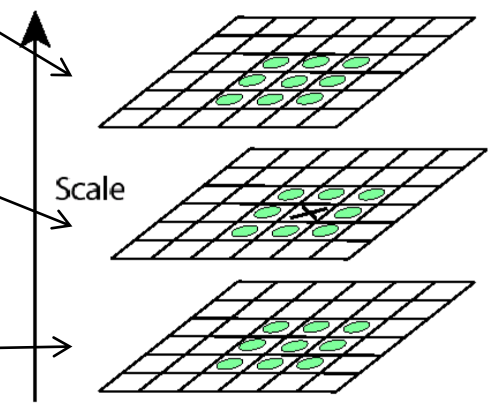


Laplacian-of-Gaussian(LoG)

- Local maxima in scale space of Laplacian-of Gaussian





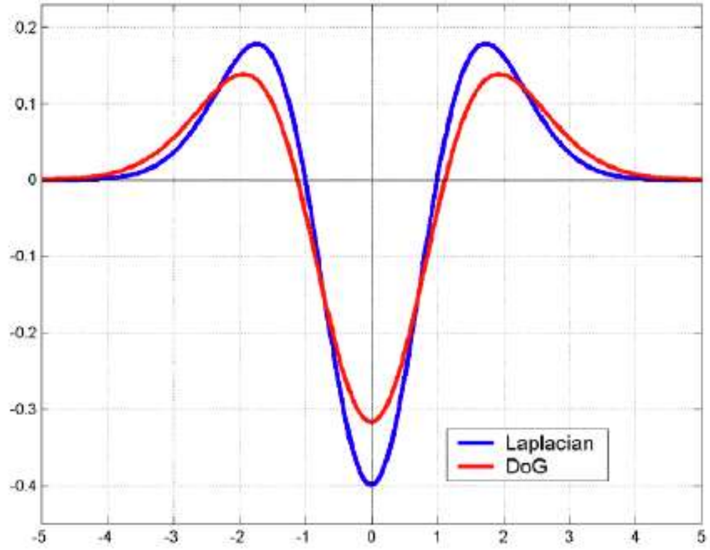


=> List of (x,y,s)

Difference-of-Gaussian (DoG)

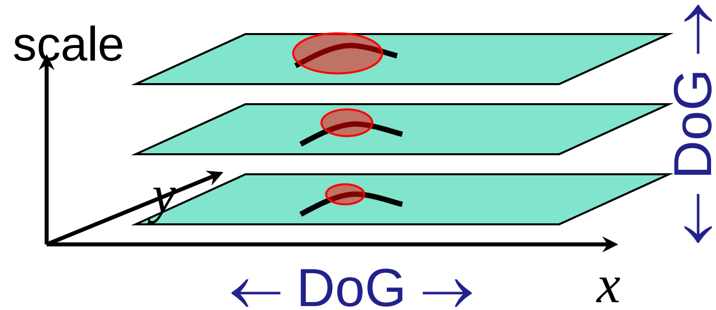
- LoG is expensive, approximate with Difference-of-Gaussian (DoG)

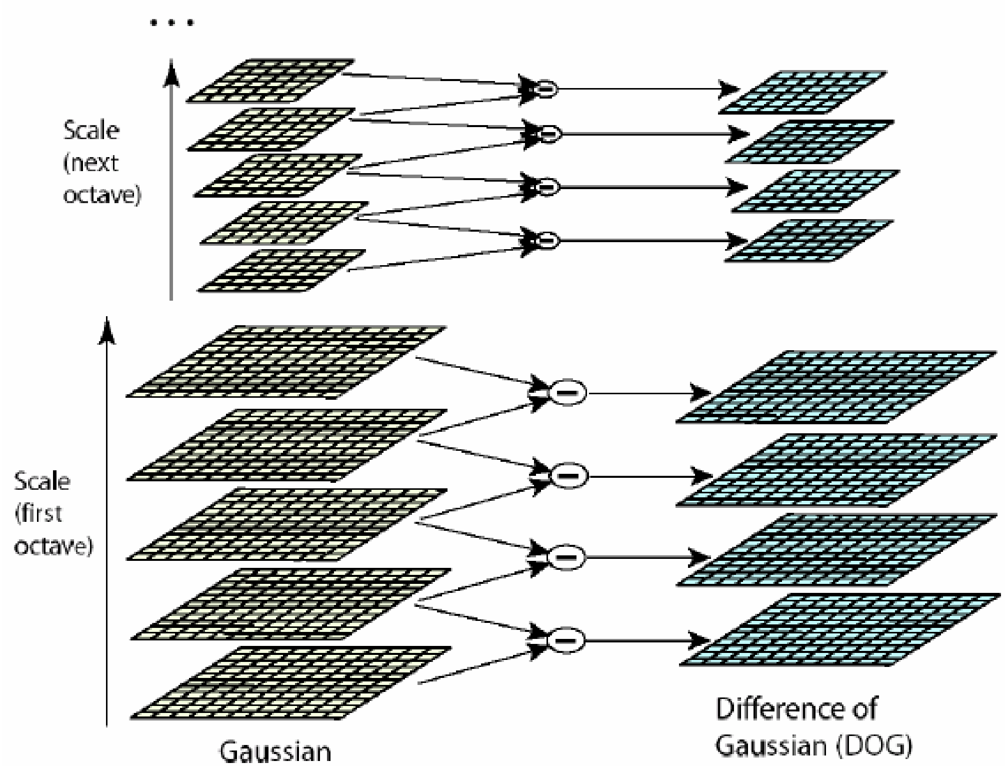
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$



SIFT: Scale-Invariant Feature Transform

- Key-point detection:
  - Find local extrema of Difference-of-Gaussians in space and scale





Scale-space octaves

Local Descriptor : SIFT

- The area around the keypoint is divided into 4 x 4 subregions
- Build an orientation histogram with 8 bins for each subregion; gradient values are weighted by a Gaussian window
- This results in a vector with 128 dimensions (4 x 4 x 8)
- Normalize this vector to unit length (grants invariance to multiplicative changes in lighting)

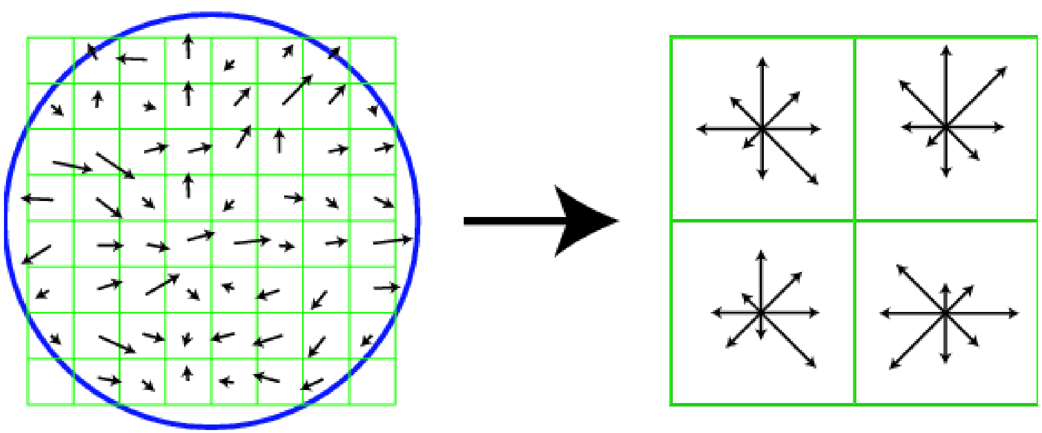


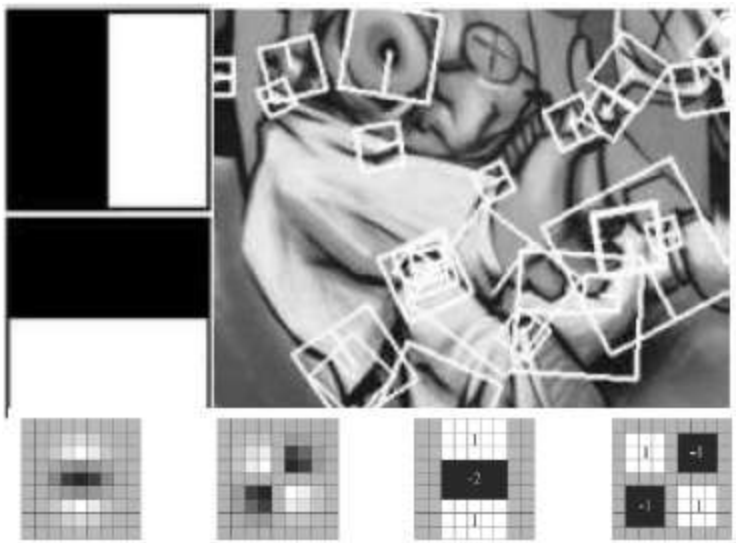
Image gradients Keypoint descriptor

Illustration shows 2x2 subregions

SIFT-Features: Properties

- Scale-invariant
- Rotation-invariant
- Robust to illumination change
- Robust to noise
- Robust to minor changes in view-point

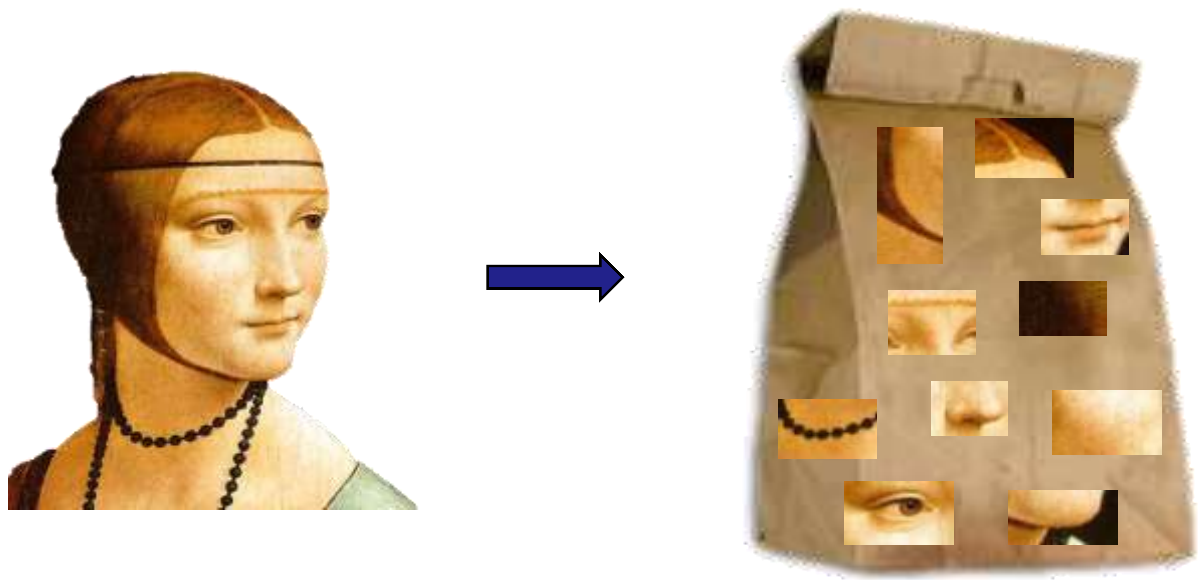
Local Descriptor : SURF



- Fast approximation of SIFT idea
  - Efficient computation by 2D box filters & integral images ⇒ 6 times faster than SIFT
- Equivalent quality for object identification
- GPU implementation available
  - Feature extraction @ 100Hz (detector + descriptor, 640×480 img) http://www.vision.ee.ethz.ch/~surf

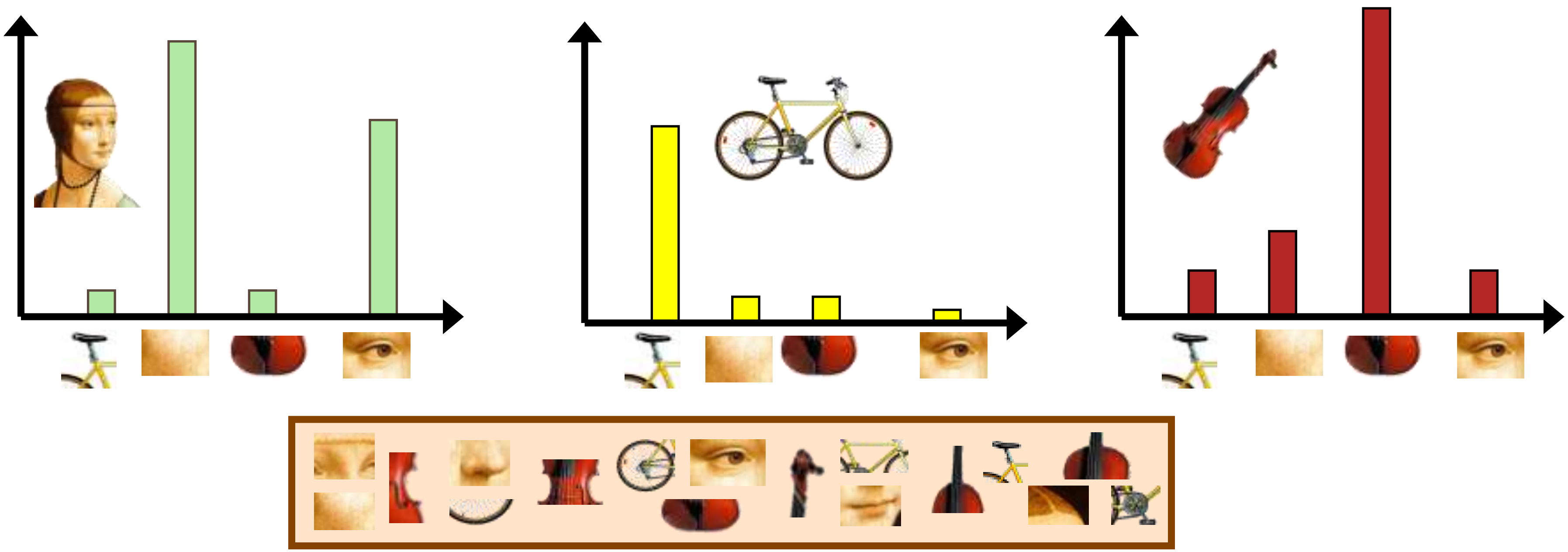
[Bay, ECCV’06], [Cornelis, CVGPU’08]

Bag-of-words



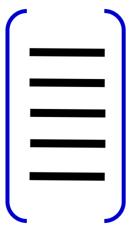
Bag-of-Words

- Analogy to text documents
- Definition
  - Independent features
  - histogram representation



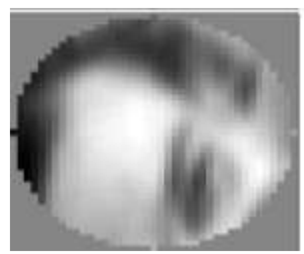
Build Visual-word Vocabulary-1

- Detect feature : Regular grid or Interest point
- Represent with local descriptor, e.g. SIFT

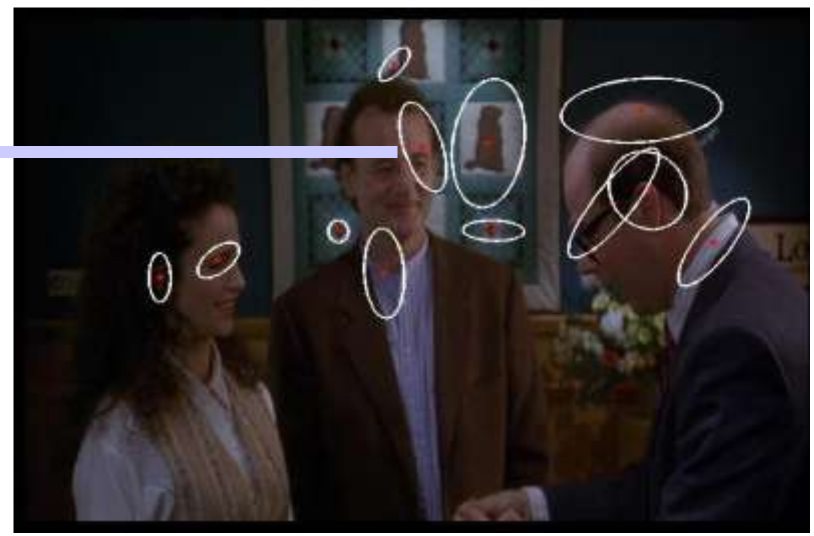


Compute SIFT descriptor

[Lowe'99]



Normalize patch



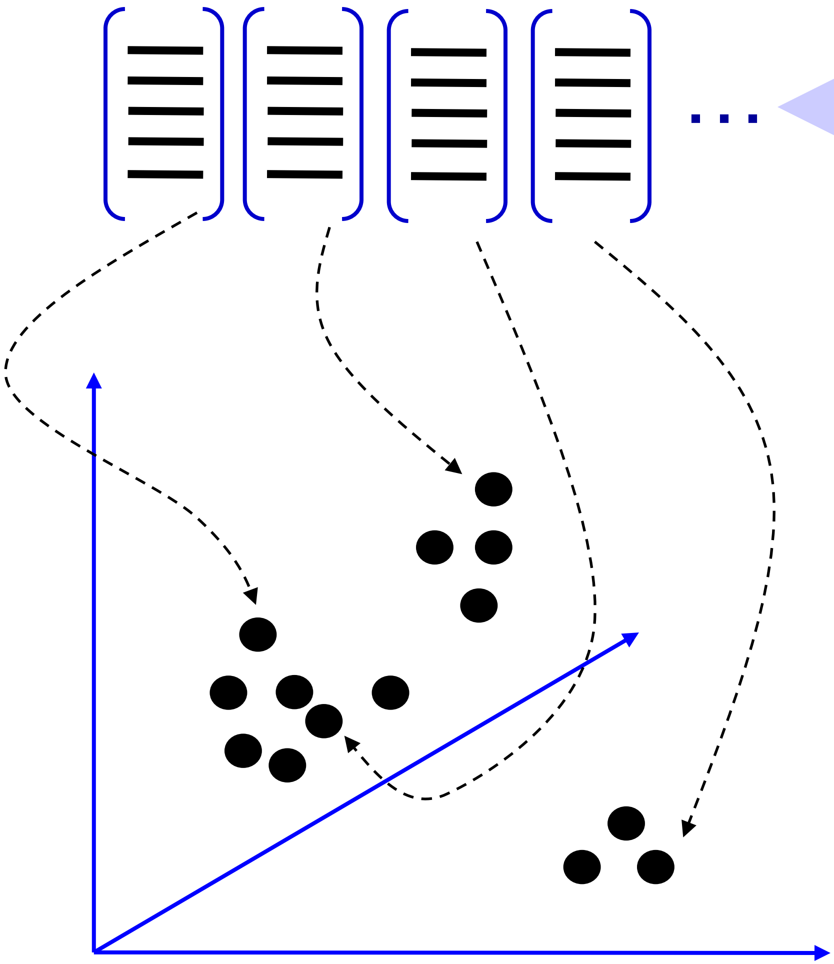
Detect patches

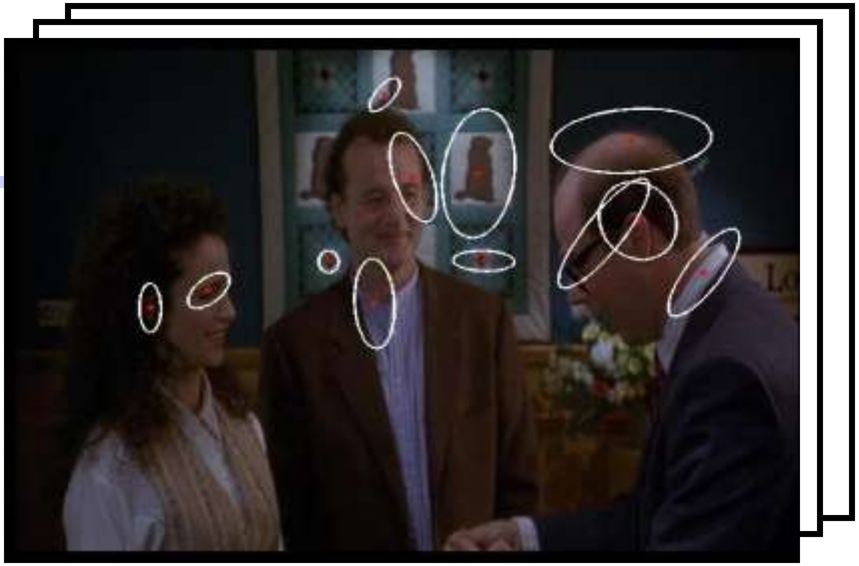
[Mikojaczyk and Schmid '02]

[Mata, Chum, Urban & Pajdla, '02]

[Sivic & Zisserman, '03]

Build Visual-word Vocabulary-2





Visual descriptors

Build Visual-word Vocabulary-3

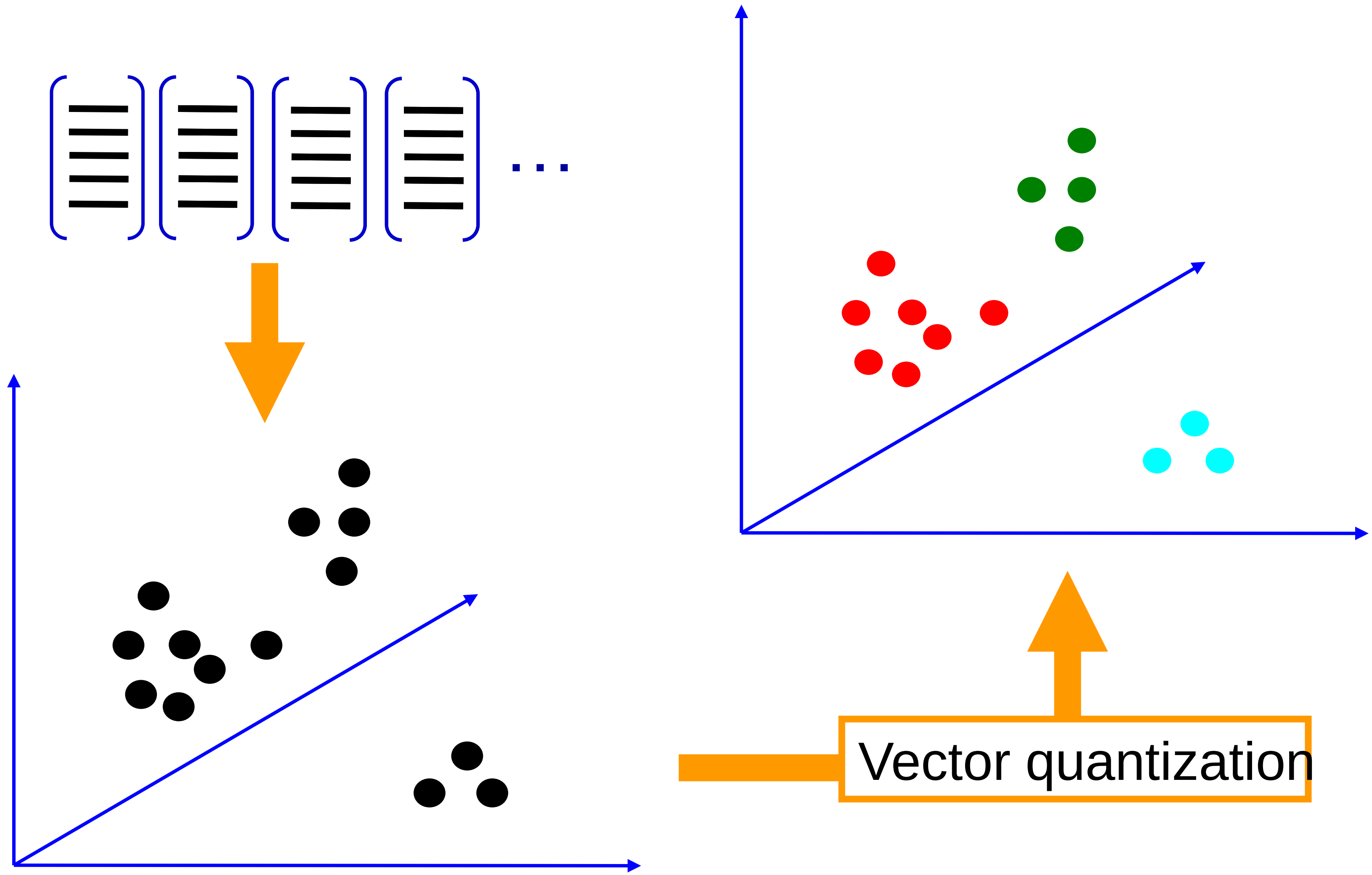


Image Representation

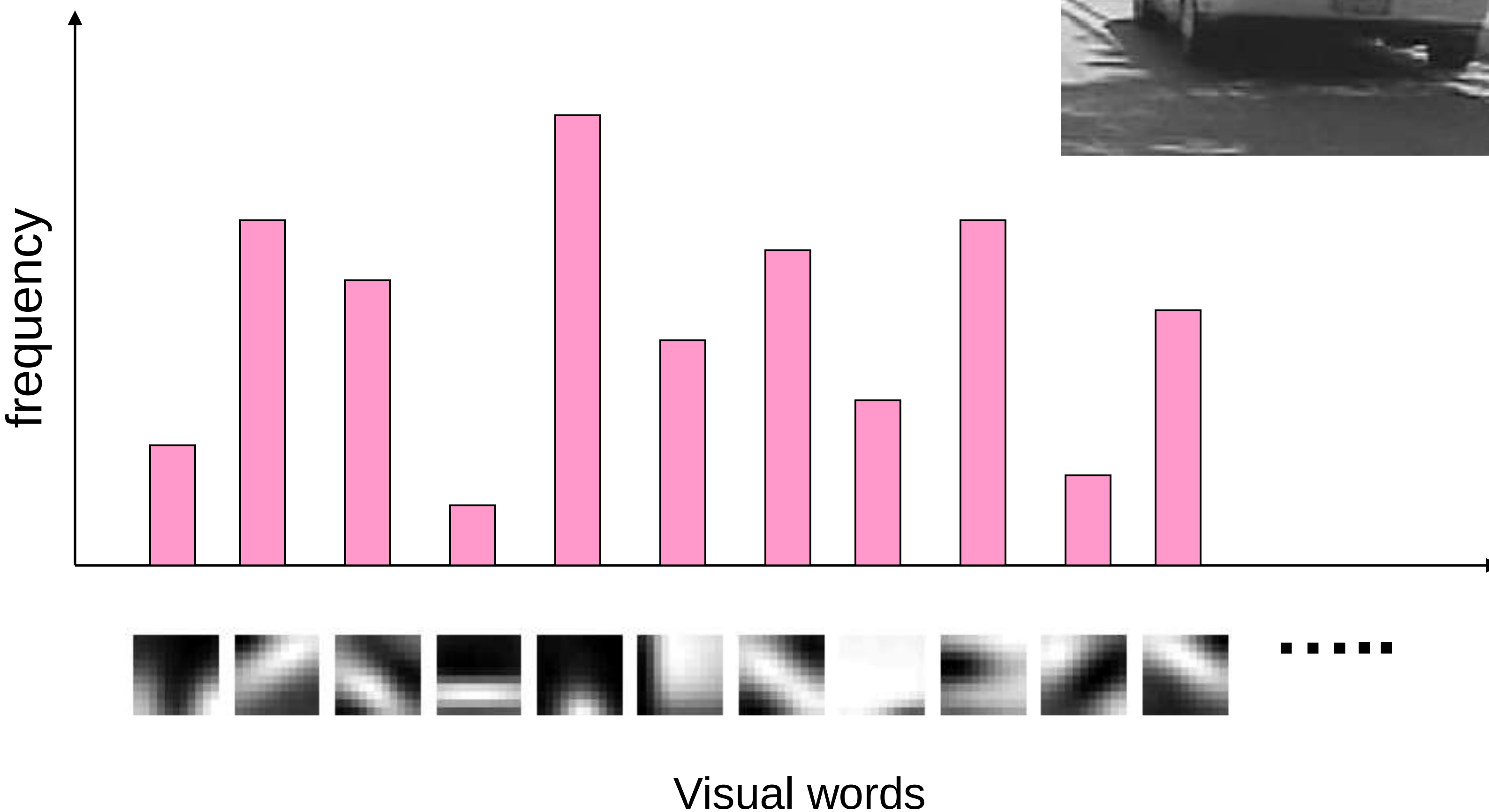


Image Segmentation

Image Segmentation

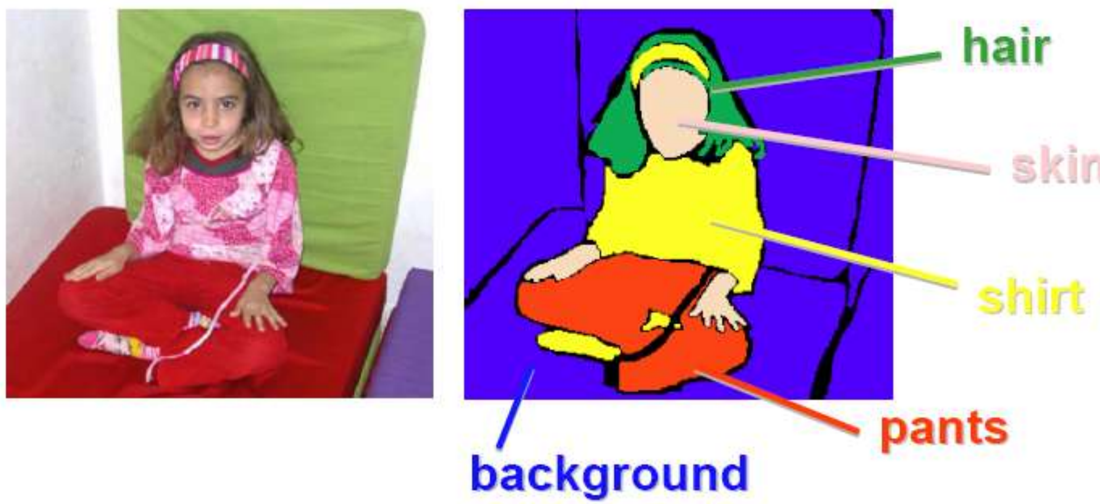
- One of the key problem in computer vision
- Identification of homogenous region in the image
- Partition an image into *meaningful* regions with respect to a particular application
- The segmentation is based on measurements taken from the image and might be *grey-level*, *colour*, *texture*, *depth* or *motion (in video)*

Different Examples

- Search in image collections
  - Find representations that make sense to the user and is related to picture content
- Video summarization / shot boundary detection
  - Find similar frames, represent subsequences by key frame
- Finding people
  - Specific detectors, part-based detectors
- Finding buildings
- Finding machine parts
- Background subtraction

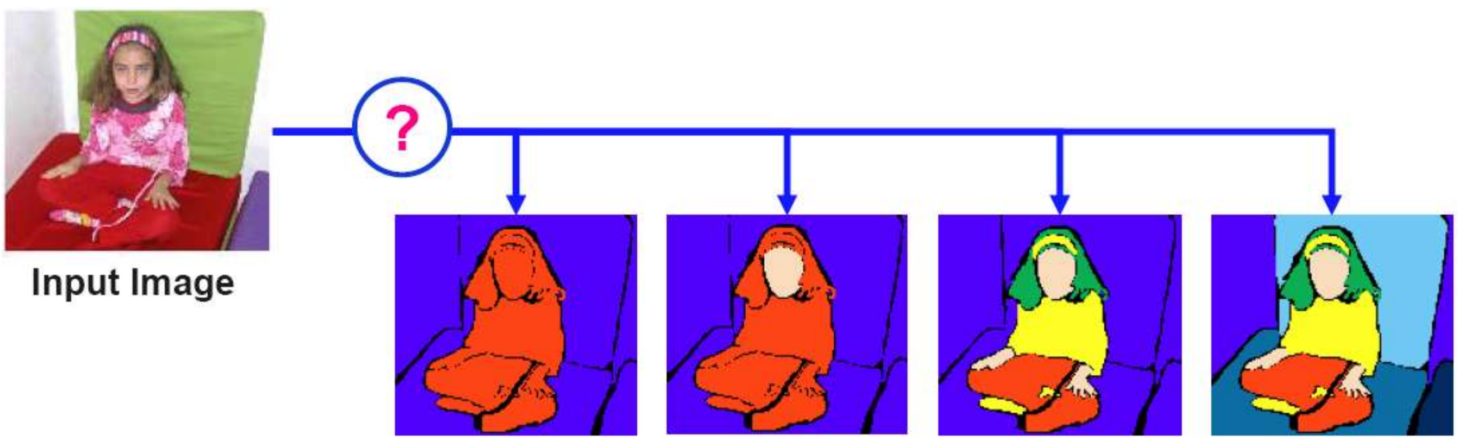
Motivation

- Before high-level reasoning on image, it can be broken down into its major structural components
- Necessary for extracting reasonable local features (color, texture, etc.)
- Simplify or change image representation into more meaningful one for ease of analysis



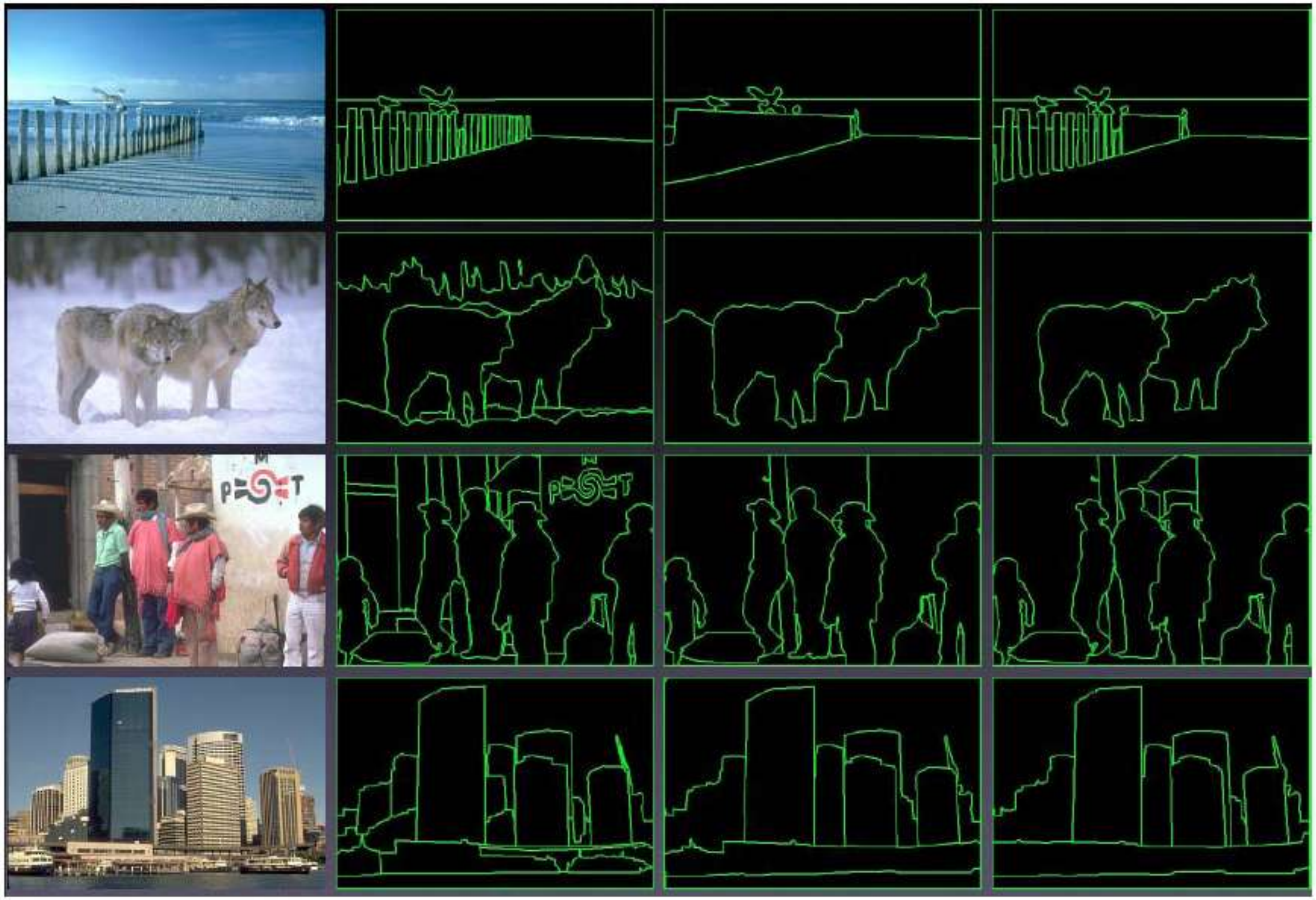
Difficulties

- What is “correct” segmentation?
  - No single correct answer
  - Interpretation depends on prior world knowledge
  - World knowledge is difficult to represent



Alternative segmentations

“Correct” Segmentation



Good Segmentation?

- Typical assumptions (inspired from human vision):
  - Intensity / color coherence
  - Texture coherence
  - Motion coherence

Image Segmentation

- Categories:
  - Pixel-based Segmentation
  - Region-based Segmentation
  - Edge-based Segmentation
  - (Graph-based Segmentation)

Pixel-based Segmentation

- Thresholding
- Clustering

Thresholding

- Determine the best threshold given a histogram of intensities
- Automatic thresholding
  - P-tile method
  - Mode method
  - Local adaptive method

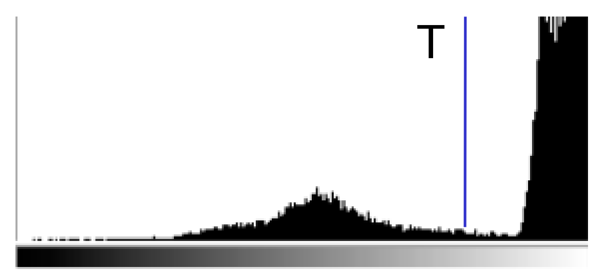
- Limitation of thresholding
  - Use *global* information
  - Ignore *spatial* relationships among pixels

Thresholding

- Determine the best threshold given a histogram of intensities
- Simplest way to segment an image: separate light and dark regions

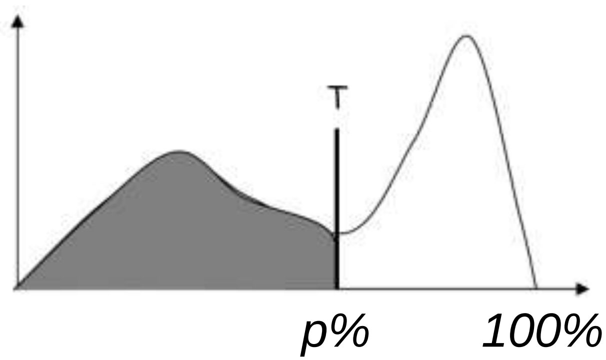
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{otherwise} \end{cases}$$





P-tile method

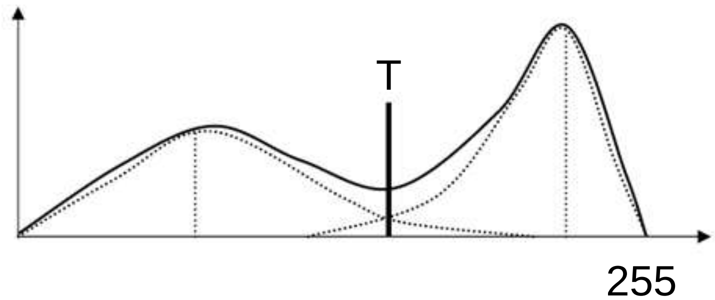
- Use the a priori knowledge about the size of the object: assume an object with size *p*
- Choose the threshold such that *p%* of the overall histogram is determined



$\Longrightarrow$ Obviously limited use

Mode-Method

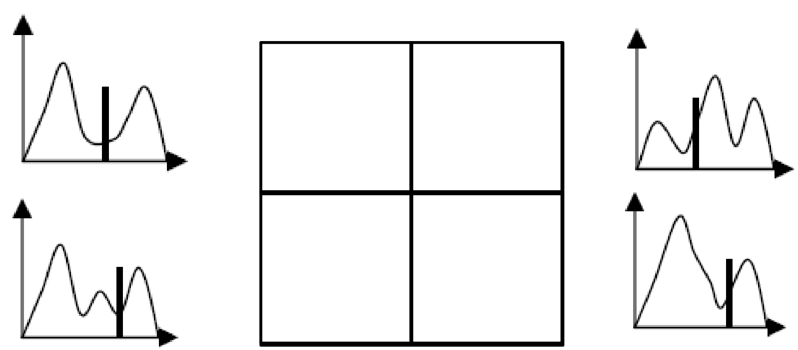
- Find the peaks (modes) of the histogram and the local minimum between them
- Set threshold to the pixel value of the local minimum



- Not trivial to find peaks and local minimum on a noisy histogram
  - Ignore local peaks
  - Maximize “peakiness”

Adaptive-Method

- One single global threshold does not work for uneven illumination
- Local adaptive method
  - Divide an image into *mxm* subimages and determine a threshold for each subimage



Clustering

- Process of partitioning a set of "patterns" into clusters – find subsets of points which are close together
- Cluster pixels based on
  - Intensity values
  - Color properties
  - Motion/optical flow properties
  - Texture measurements etc.
- Input: set of measurements $x_1, x_2, \ldots x_m$.
- Output: set of clusters and their centers

Simple Clustering Approaches

- Agglomerative Clustering (Merging)
  1. Make each point (pixel) a separate cluster
  2. Merge clusters with smallest inter-cluster distance until clustering is satisfactory

- Divisive Clustering (Splitting)
  1. Construct a single cluster using all points
  2. Split clusters with largest inter-cluster distance until clustering is satisfactory

- Difficulties:
  - Choice of inter-cluster distances
  - Stopping criterion (how many cluster are there?)

Segmentation by k-means

- Simple clustering methods use greedy approaches
- Alternative:
  - Formulate an objective function that should be optimized
  - Assuming that we know that there should be k-clusters, a good objective function would be

$$\Phi(\text{clusters}, \text{data}) = \sum_{i \in \text{clusters}} \left\{ \sum_{j \in i\text{th cluster}} (x_j - c_i)^T (x_j - c_i) \right\}$$

  - Where x_j is a point coordinate, c_j is a cluster center

- If allocation of points to clusters were known, centers could be easily computed
  - But this is not the case

k-means algorithm

- Define iterative algorithm:
  - Assume the cluster centers are known and allocate each point to closest cluster
  - Assume allocation is known and choose new set of cluster centers. Each center is the mean of the points allocated to the that cluster

- Algorithm:
  - Choose initial mean values for *k* regions
  - Classify *n* pixels by assigning them to “closest” mean
  - Recompute the means as the average of samples in their (new) classes
  - Continue till there is no change in mean values

Color Clustering Examples

- Clustering in RGB space

Original images

Segmented images

9 clusters

5 clusters

4 clusters

Region-based Segmentation

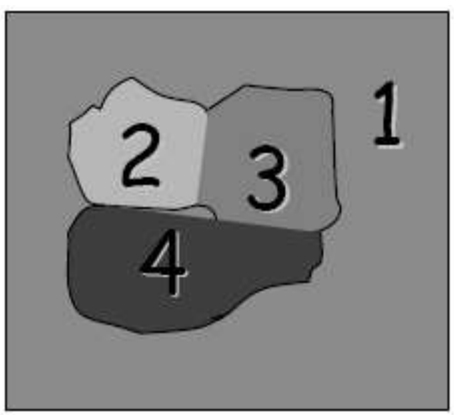
- The main idea in ***region-based segmentation*** techniques is to identify different regions in an image that have similar features (gray level, colour, texture, etc.).
- There are two main region-based image segmentation techniques:
  - Region merging
  - Region splitting

Region Merging

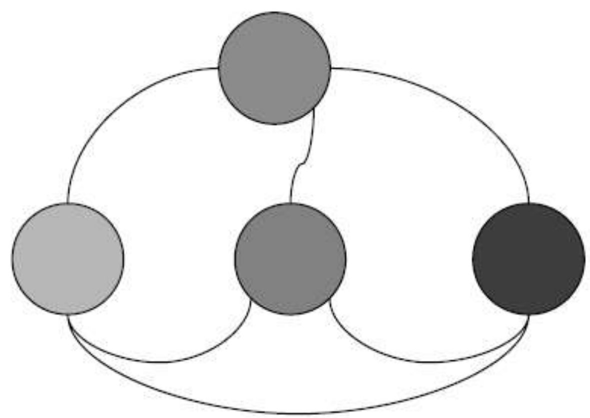
- Merge two adjacent regions if they have "similar" properties according to some criterion.
- What does "similar" mean?
  - Examples:
    - "similar" average values : $|\mu_i - \mu_j| < T$
    - "small" spread of gray values : $|g_{max} - g_{min}| < T$
      - $g_{max} = max\ \{g(x,y)|(x,y) \in R_i \cup R_j\}$
      - $g_{min} = min\ \{g(x,y)|(x,y) \in R_i \cup R_j\}$
  - Note: non-transitiv
    - A similar to B, and B similar to C does not imply that A is similar to C.

Region Merging

- Start with an initial segmentation
  - e.g. By thresholding
- Form the Region Adjacency Graph (RAG)
  - Regions are the nodes
  - Adjacency relations are the links



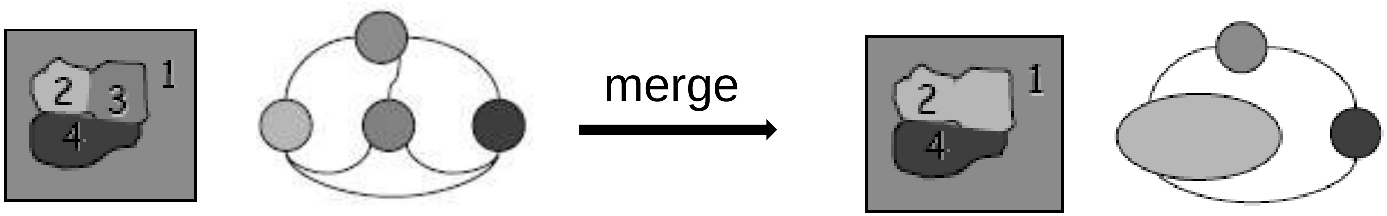
Initial segmentation



RAG

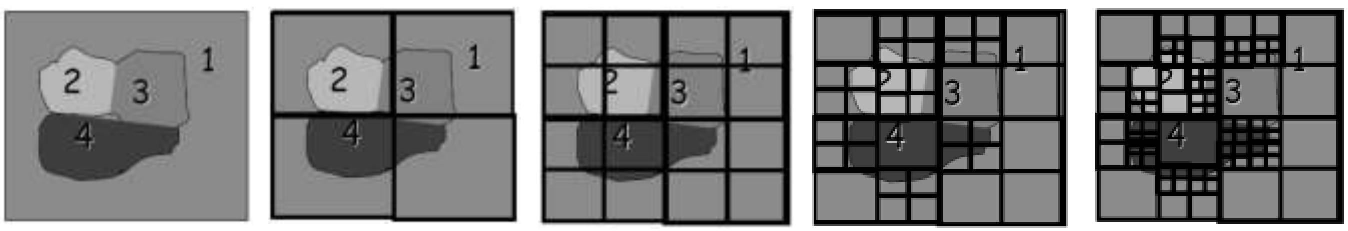
Region Merging

- For each region in the image do:
  - Consider its adjacent regions and test if they are similar
  - If they are similar, merge them and update the RAG
- Repeat the merging steps until there are no more merges.



Region Splitting

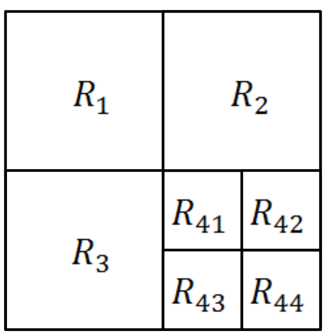
- Quad-tree decomposition:
  - Subdivide the entire image successively into smaller and smaller quadrant regions until having homogeneous regions.

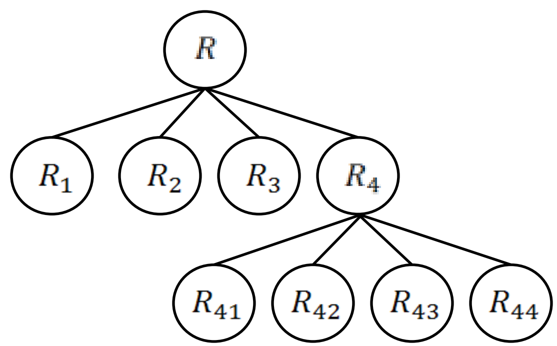


- The subdivision is represented with quad tree

The Quadtree Representation

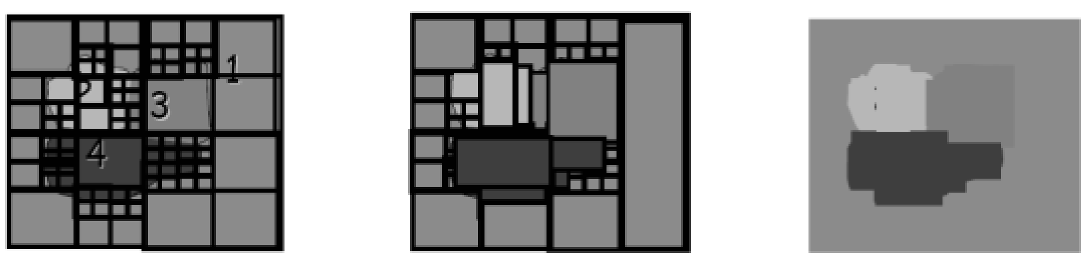
- Quadtrees:
  - Trees where nodes have 4 children
- Build quadtree:
  - Nodes represent regions
  - Every time a region is split, it‘s node give birth to 4 children
  - Leaves are nodes for uniform regions





Region Splitting & Merging

- Splitting only results in adjacent regions with identical properties
- The final result can be obtained through merging the quadtree
  - Siblings that are “similar” can be merged



Edge-based Segmentations

- Based on detection of discontinuities, and segment the image along the discontinuities
- 3 basic types of gray-level discontinuities: points, lines, edges
- Edge detection is the most common approach for detecting meaningful discontinuities

Edge Linking and Boundary Detection

- From intensity discontinuities to more general segmentation
  - For example, from edge pixels to line segments

- Local processing
  - Analysis of small neighborhood
    - Strength and direction of the gradient of edge pixels
- Global processing
  - Analysis of the whole image
    - Global relationships between pixels
  - Hough Transform