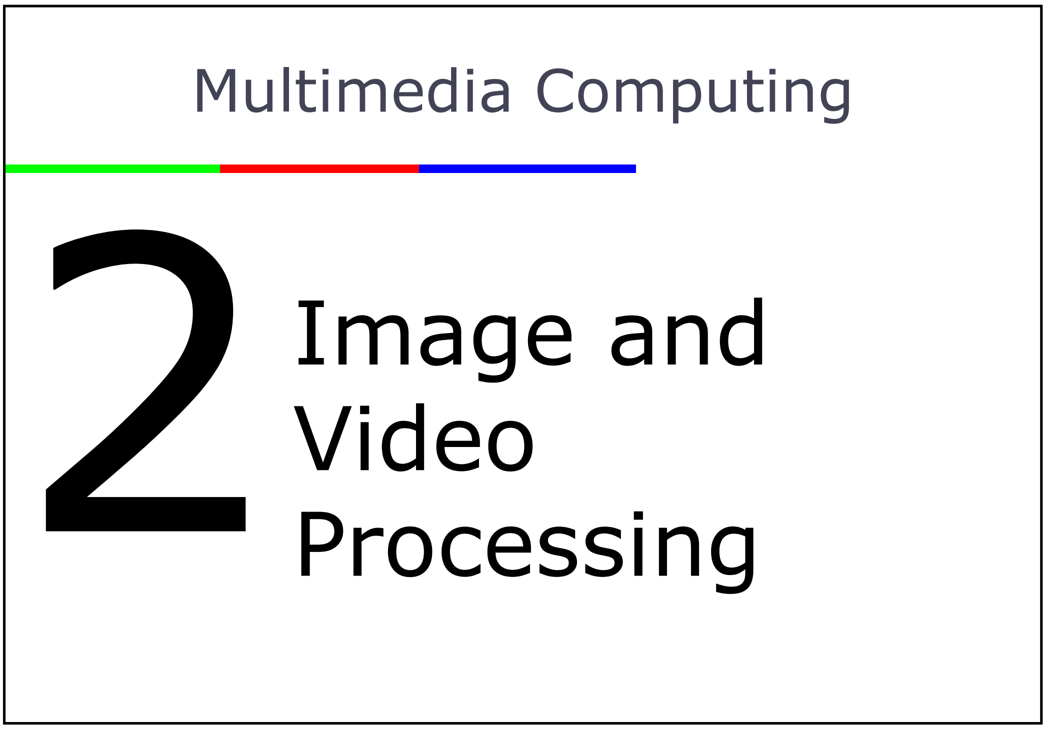
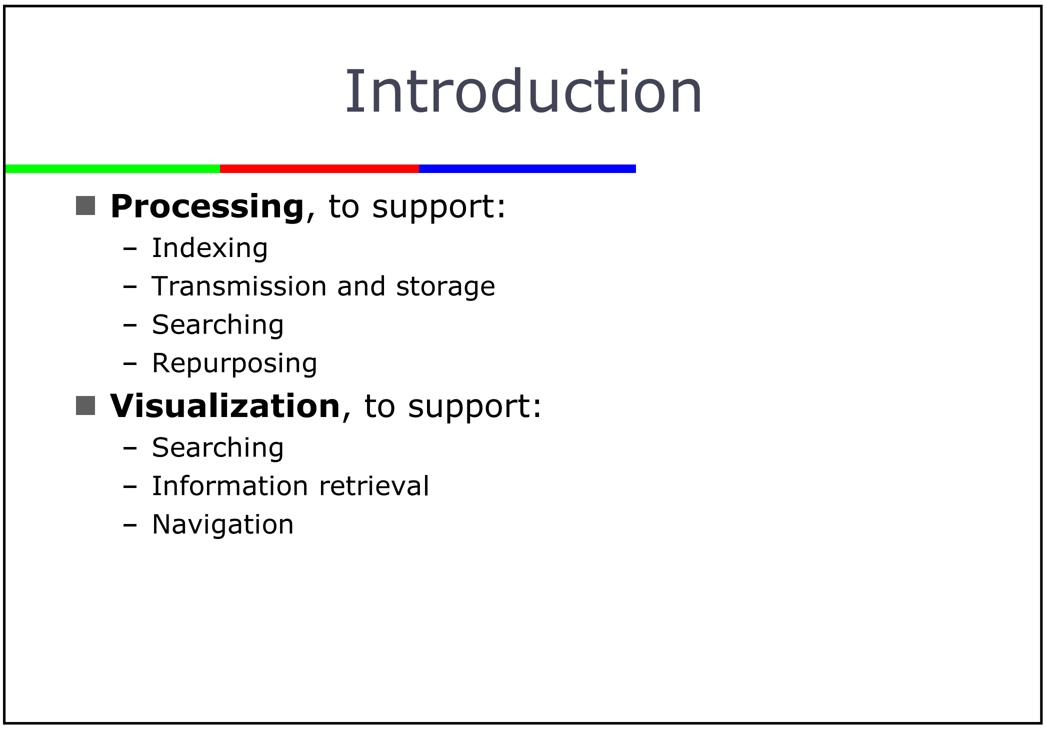


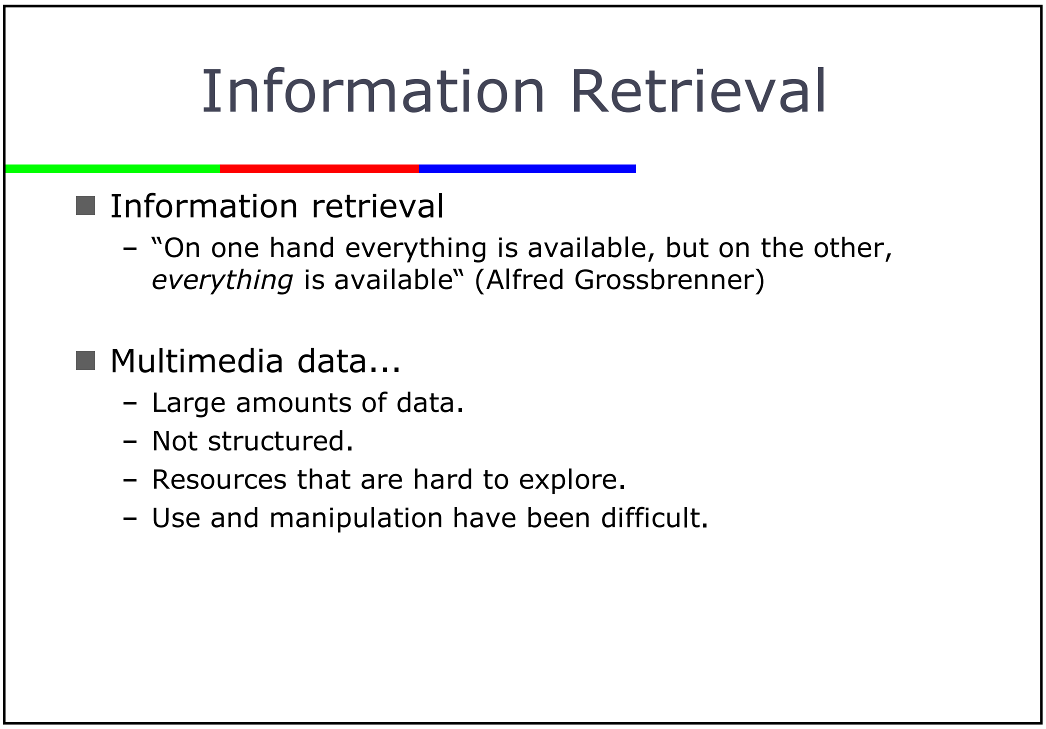
Multimedia Computing

2 Image and Video Processing



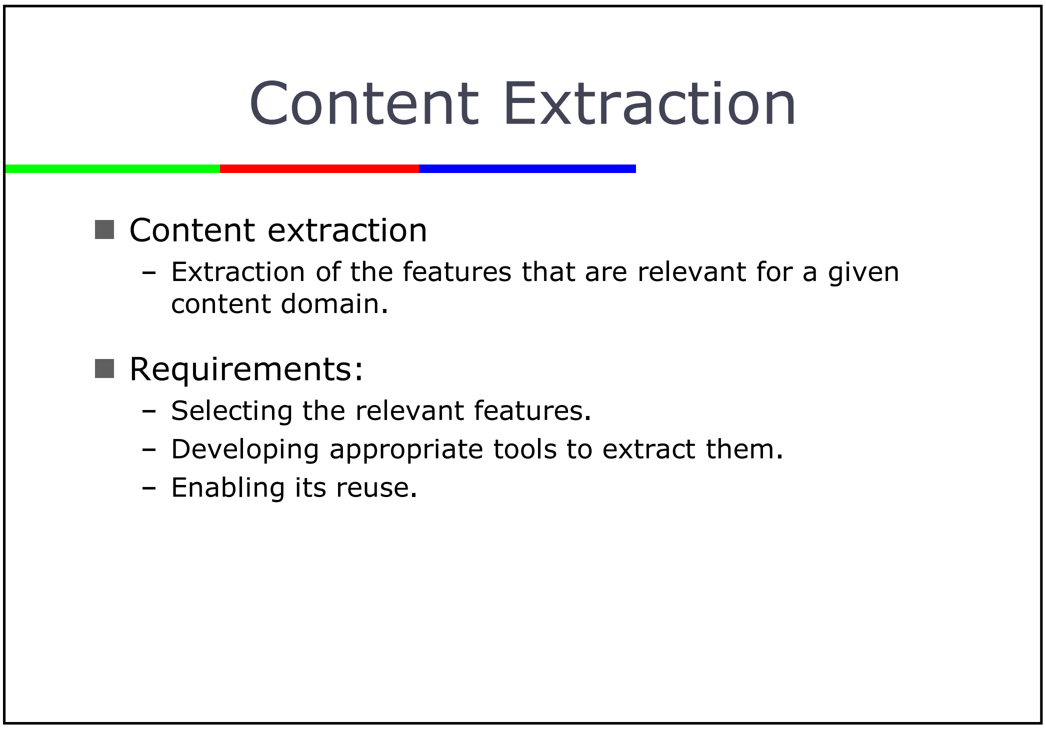
Introduction

- **Processing**, to support:
  - Indexing
  - Transmission and storage
  - Searching
  - Repurposing
- **Visualization**, to support:
  - Searching
  - Information retrieval
  - Navigation



Information Retrieval

- Information retrieval
  - “On one hand everything is available, but on the other, *everything* is available‟ (Alfred Grossbrenner)
- Multimedia data...
  - Large amounts of data.
  - Not structured.
  - Resources that are hard to explore.
  - Use and manipulation have been difficult.



Content Extraction

- Content extraction
  - Extraction of the features that are relevant for a given content domain.
- Requirements:
  - Selecting the relevant features.
  - Developing appropriate tools to extract them.
  - Enabling its reuse.

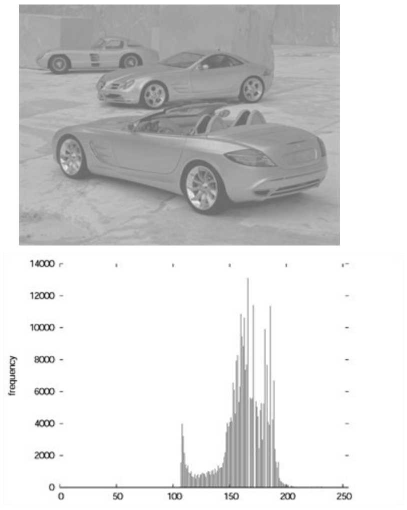
Content Extraction

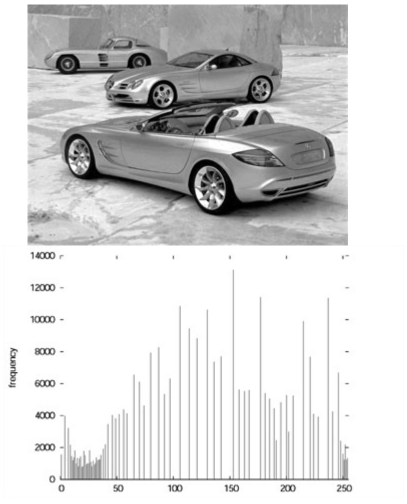
- Content extraction techniques
  - Image analysis and processing.
  - Motion analysis and temporal segmentation.
  - Audio analysis and processing.
- Applications and related systems
  - Support to processing algorithms
  - Content based retrieval
  - Content based classification and analysis

Image Processing

- Histograms
- Operations on histograms
- Filters
- Segmentation
- Features and retrieval

Histograms





Histograms

- Obtained by counting the number of colors in one image.
- The discrete histogram is obtained as follows:
  - H(k) = #pixels with color k
- Normalized histogram
  - $H_{norm}(k)$ = H(k)/N, where N is the number of pixels in the image
- It can be done in RGB, grayscale or other representations.

Histogram Operations

- Linear transformations
  - Brightness change

    $y = x + b$ ($b > 0$ more brightness, $b < 0$ less brightness)
  - Change in the dynamic range

    $y = mx + b$ (with $m <> 1$)
- Histogram equalization
  - Each possible value will have the same number of pixels.
  - Usually it is not possible to completely obtain the intended result.
  - The result depends of the application.

Contrast

- The contrast of an image in a point represents the difference between the relative intensity at that point and the intensity of neighboring points:

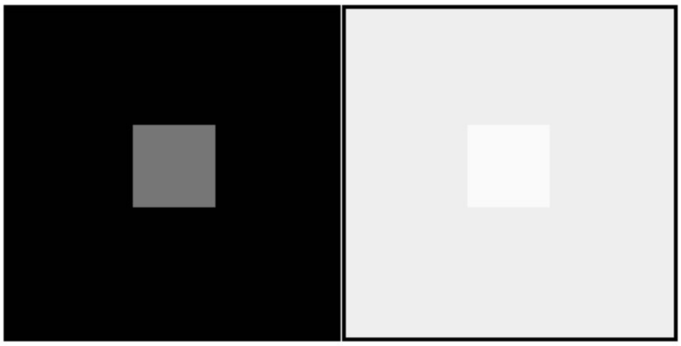
$$C = |I_p - I_n|/|I_n|$$

Example: $C_1 = |0.3 - 0.1|/|0.1| = 2$

$C_2 = |0.7 - 0.5|/|0.5| = 0.4$

The intensity is greater for C1 but the contrast is smaller.

Contrast



$$C=\left|\frac{75-25}{25}\right|=2 \qquad C=\left|\frac{178-128}{128}\right|=0.4$$

Filters

- Low pass filter: Considering a neighborhood of each pixel of 2M + 1 by 2M + 1

$$g(x,y)=(1/P)\sum_{i=-M}^{M}\sum_{j=-M}^{M}f(x+i,y+j),0\le x,y\le N-1$$

Where g(x,y) is the filtered image (NxN), f(x,y) is the original image and P = $(2M+1)^2$. In a more generic way, with different weights for each pixel:

$$g(x,y)=(1/P)\sum_{i=-M}^{M}\sum_{j=-M}^{M}h(i,j)f(x+i,y+j),0\le x,y\le N-1$$

$$P=\sum_{i=-M}^{M}\sum_{j=-M}^{M}h(i,j)$$

Filters

- Sharpen

$$h(x)=\begin{bmatrix}-1 & -1 & -1\\ -1 & 9 & -1\\ -1 & -1 & -1\end{bmatrix}\quad h(x)=\begin{bmatrix}0 & -1 & 0\\ -1 & 5 & -1\\ 0 & -1 & 0\end{bmatrix}$$

- Laplacian: Zero sum, edge detection

$$h(x)=\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{bmatrix}\quad h(x)=\begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}$$

- Other edge detection filters:

Roberts

$$h1(x,y)=\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\quad h2(x,y)=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$$

Prewitt

$$h1(x)=\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix}\quad h2(x)=\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{bmatrix}$$

Sobel

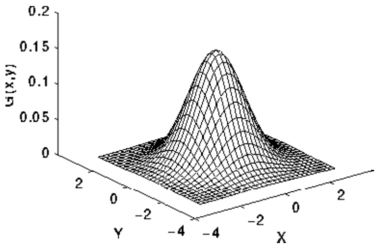
$$h1(x)=\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}\quad h2(x)=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

Gaussian Blur

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

$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Segmentation

- Segmentation: Splits an image in regions or sets of pixels.
- *Thresholding*
  - Global: Global threshold to separate objects from the background
  - Local: The image is divided in regions. The thresholding operation is applied independently to each region.
  - Adaptive: Each pixel is thresholded accordingly to is neighborhood.
- Edge based segmentation: Uses edge detection filters to detect the edges/boundaries. What is inside belongs to the object.

Morphological Operations

- Used in many situations, e.g., pre-processing and post-processing in more complex systems
- Add or remove components from the image
- Based on set theory
- Fundamental operations:
  - *Dilation*
  - *Erosion*
  - *Opening*
  - *Closing*

Morphological Operations

- Dilation

$$D(A,B) = \bigcup_{\beta \in B} (A + \beta)$$

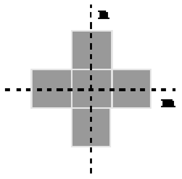
- Erosion

$$E(A,B) = \bigcap_{\beta \in B} (A - \beta)$$

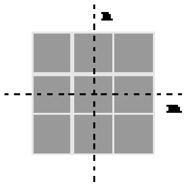
- Dilation of the objects is equivalent to erosion of the background

Morphological Operations

- B is usually one of the following elements:



C=4

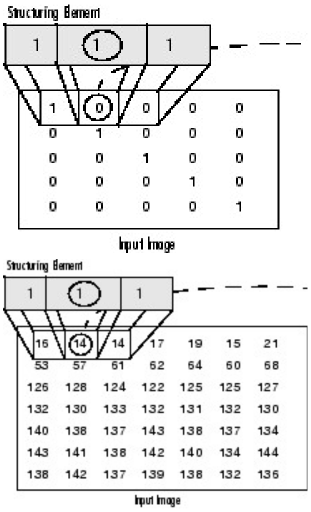


C=8

Dilation

- In general object size increases
  - OpenCV: *The effect of dilation is to fill up holes and to thicken boundaries of objects on a dark background (that is, objects whose pixel values are greater than those of the background).*
- Algorithm:
  - Consider each pixel object (= 255) and set the background pixels (= 0) that are connected (*C-connected*) to 255.
  - In general, set to the maximum value.

Dilation Example



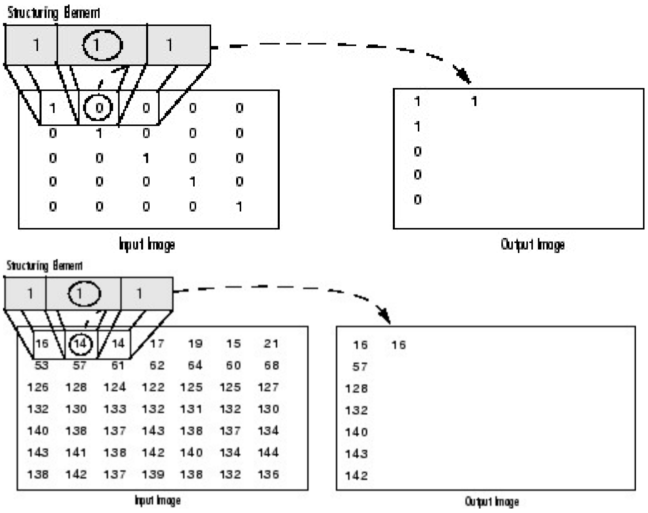
1	1	1

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

1	1	1

16	14	14	17	19	15	21
53	57	61	62	64	60	68
126	128	124	122	125	125	127
132	130	133	132	131	132	130
140	138	137	143	138	137	134
143	141	138	142	140	134	144
138	142	137	139	138	132	136

Dilation Example



Structuring Element

1	1	1

Input Image

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Output Image

1	1
1	
0	
0	
0	

Structuring Element

1	1	1

Input Image

16	14	14	17	19	15	21
53	57	61	62	64	60	68
126	128	124	122	125	125	127
132	130	133	132	131	132	130
140	138	137	143	138	137	134
143	141	138	142	140	134	144
138	142	137	139	138	132	136

Output Image

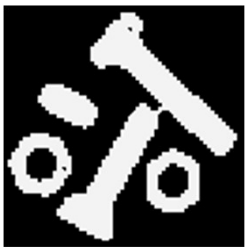
16	16
57	
128	
132	
140	
143	
142	

Erosion

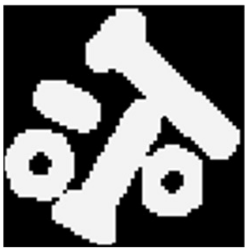
- In general object size decreases
  - OpenCV: *The effect of erosion is to remove spurious pixels (such as noise) and to thin boundaries of objects on a dark background (that is, objects whose pixel values are greater than those of the background).*
- Algorithm:
  - Consider each background object (= 0) and set the foreground pixels (= 255) that are connected (*C-connected*) to 0.
  - In general, set to the minimum value.

Examples

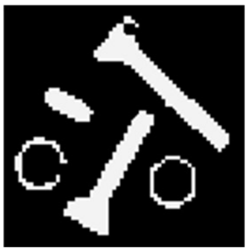
Image



Dilation

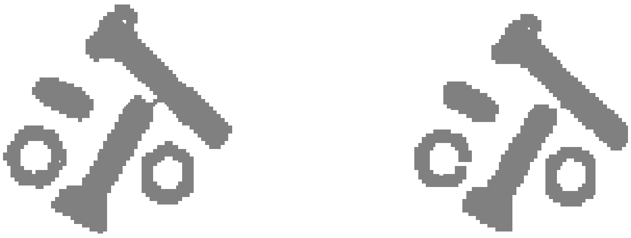


Erosion



Opening

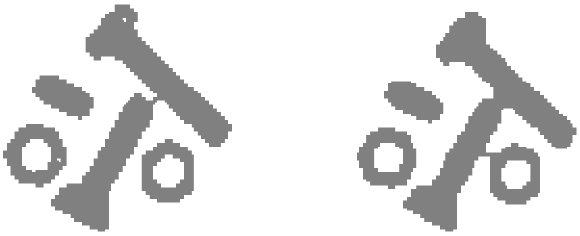
- *Erosion* seguida de *Dilation*
- *OpenCV: The process of opening has the effect of eliminating small and thin objects, breaking objects at thin points, and generally smoothing the boundaries of larger objects without significantly changing their area.*



$$O(A, B) = D(E(A, B), B)$$

Closing

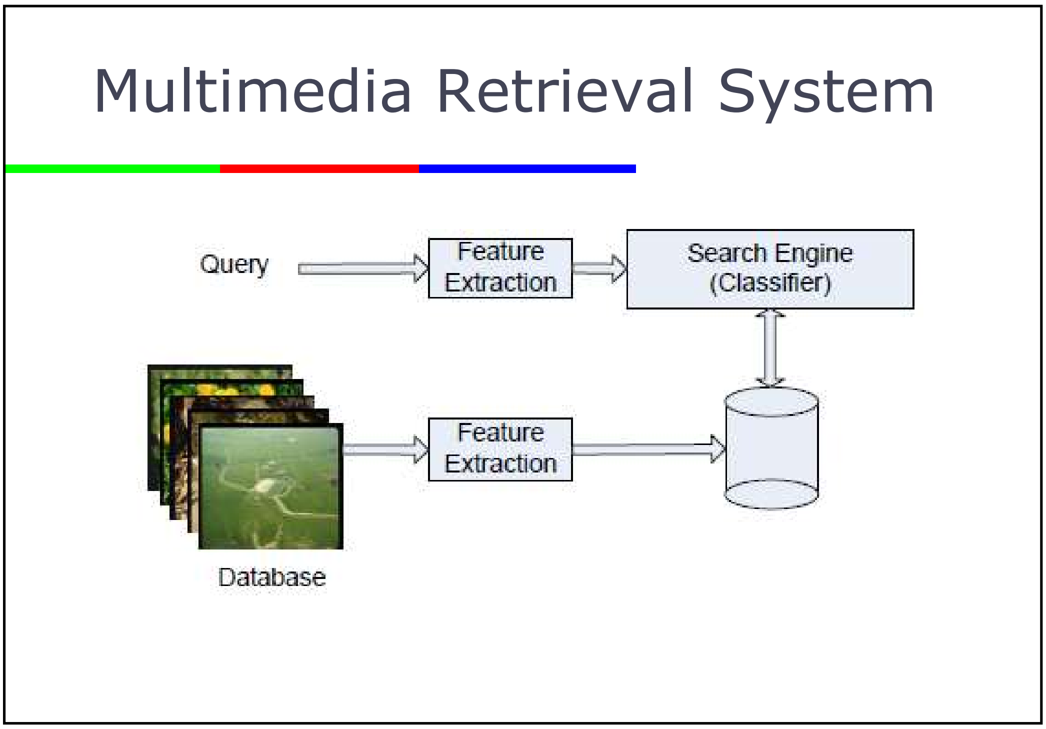
- *Dilation* followed by *Erosion*
- *OpenCV: The process of closing has the effect of filling small and thin holes in objects, connecting nearby objects, and generally smoothing the boundaries of objects without significantly changing their area.*



$$C(A,B) = E(D(A,B),B)$$

Matching and Searching

- Features: main characteristics extracted from images, audio, video…
- Distances: between features (e.g., between query and samples) to evaluate similarity
- Queries can be based on a sketch or an example
- Visual information can be extracted from images to compute the similarity between queries and the stored data (e.g., images)



Multimedia Retrieval System

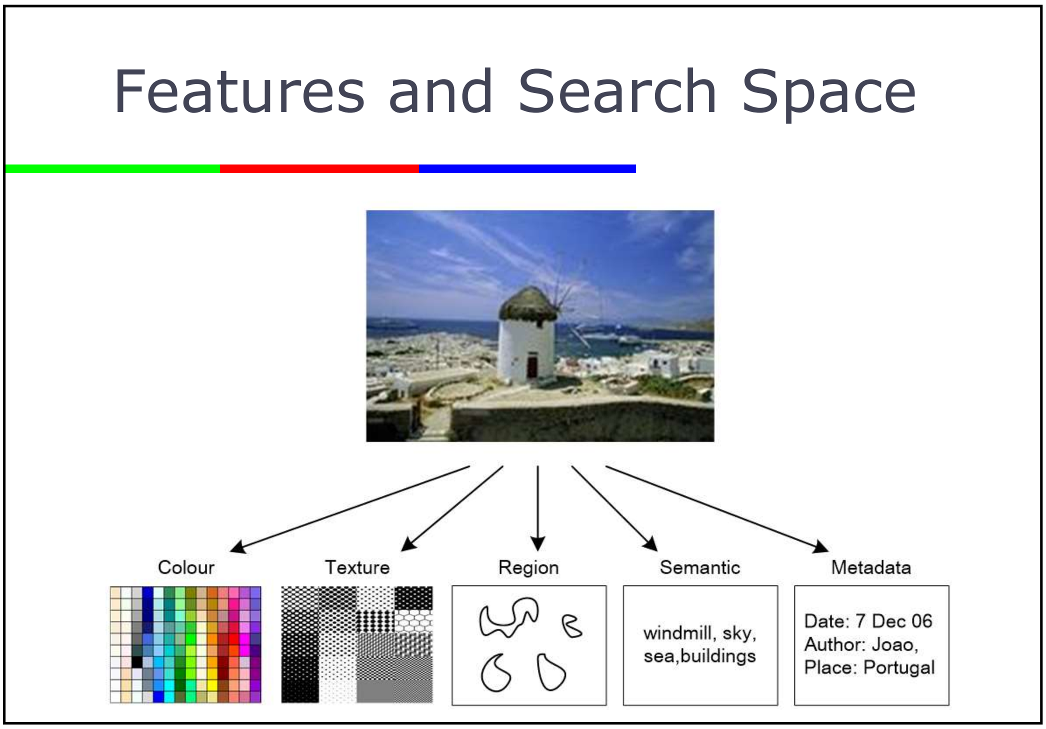
Query

Feature Extraction

Search Engine (Classifier)

Feature Extraction

Database



Features and Search Space

Colour

Texture

Region

Semantic

Metadata

windmill, sky, sea,buildings

Date: 7 Dec 06
Author: Joao,
Place: Portugal

Color

- Histogram…
- Color moments, average and variance (1st and 2nd moments)

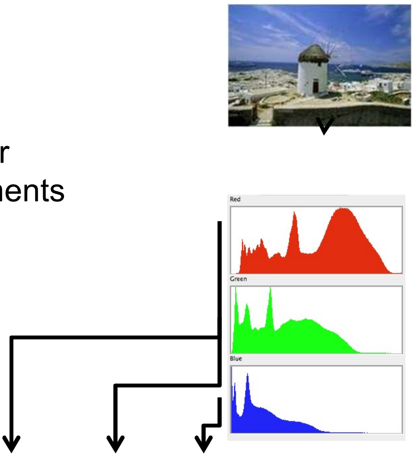
$$m_r = \sum \frac{(xi - \overline{x})^r}{N}$$

- Standard deviation, square root of variance
- Skewness and kurtosis (3rd and 4th moments)

Color Moments Example

Color moments

Marginal color histograms



$\rightarrow$ $\boxed{d_{hR} = (bin_1, bin_2, \ldots, bin_{16})}$

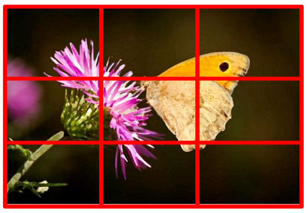
$\rightarrow$ $\boxed{d_{hG} = (bin_1, bin_2, \ldots, bin_{16})}$

$\rightarrow$ $\boxed{d_{hB} = (bin_1, bin_2, \ldots, bin_{16})}$

$$d_{cm} = (\overbrace{m_R, s_R^{\;2}}, \overbrace{m_G, s_G^{\;2}}, \overbrace{m_B, s_B^{\;2}})$$

Marginal Color Moments

- Image is divided in 9 tiles (3x3)



For each of three color channels the mean and variance of each tile are calculated

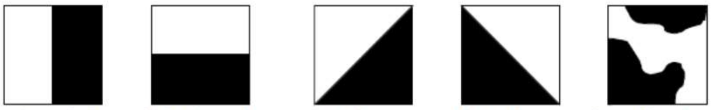
$$\mu_{t,c}=\frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}I_{t,c}(i,j) \qquad \sigma_{t,c}^{2}=\frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_{t,c}(i,j)-\mu_{t,c}\right]^{2}$$

Image is represented by the feature vector

$$x=\left[\mu_{1,1} \quad \sigma_{1,1}^{2} \quad \dots \quad \mu_{9,3} \quad \sigma_{9,3}^{2}\right]^{T}$$

Edge Histogram

- Image can be divided in parts/tiles
- Calculate edges for each sub-image and use edge count



a) vertical edge

b) horizontal edge

c) 45 degree edge

d) 135 degree edge

e)non-directional edge

$$\begin{array}{|c|c|}\hline 1 & -1 \\ \hline 1 & -1 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline \sqrt{2} & 0 \\ \hline 0 & -\sqrt{2} \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 0 & \sqrt{2} \\ \hline -\sqrt{2} & 0 \\ \hline\end{array} \quad \begin{array}{|c|c|}\hline 2 & -2 \\ \hline -2 & 2 \\ \hline\end{array}$$

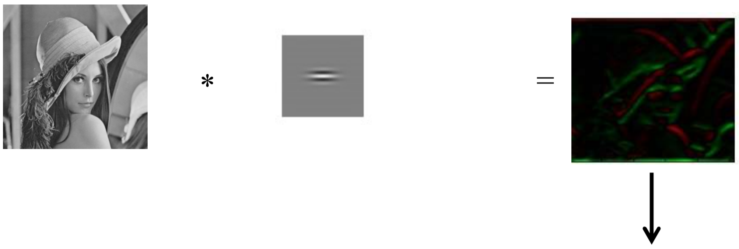
a) ver_edge_filter() b) hor_edge_filter() c) dia45_edge_filter() d) dia135_edge_filter() e) nond_edge_filter()

Texture

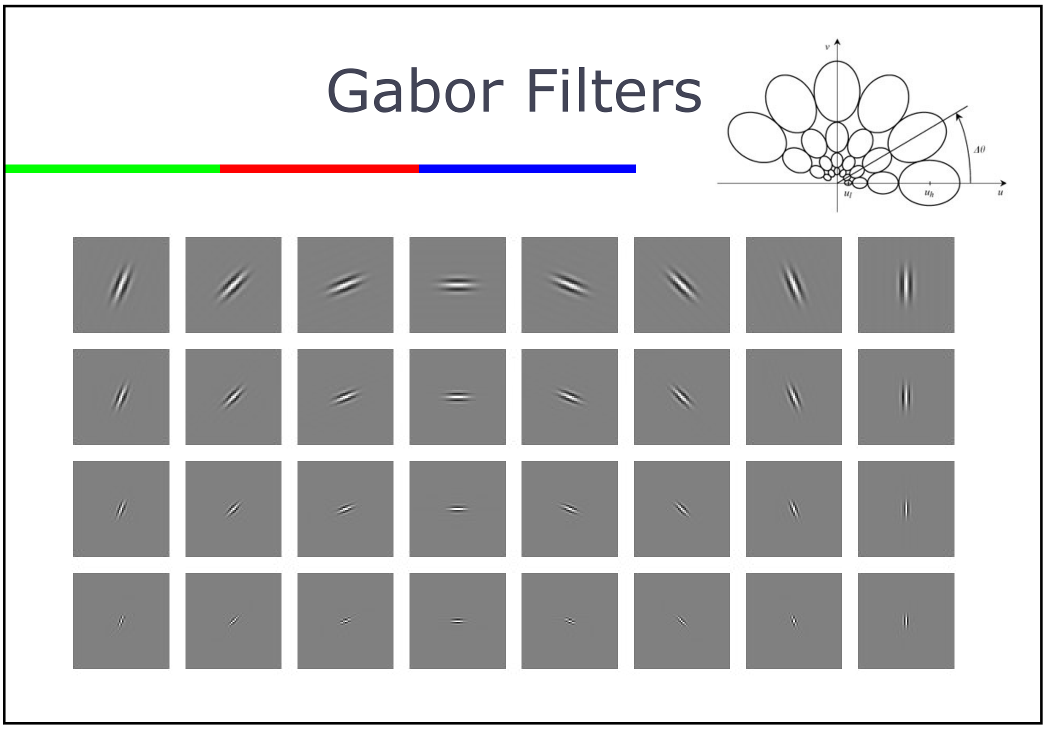
Gabor Filters

- Images are convolved (operator convolution *****) with each filter individually:

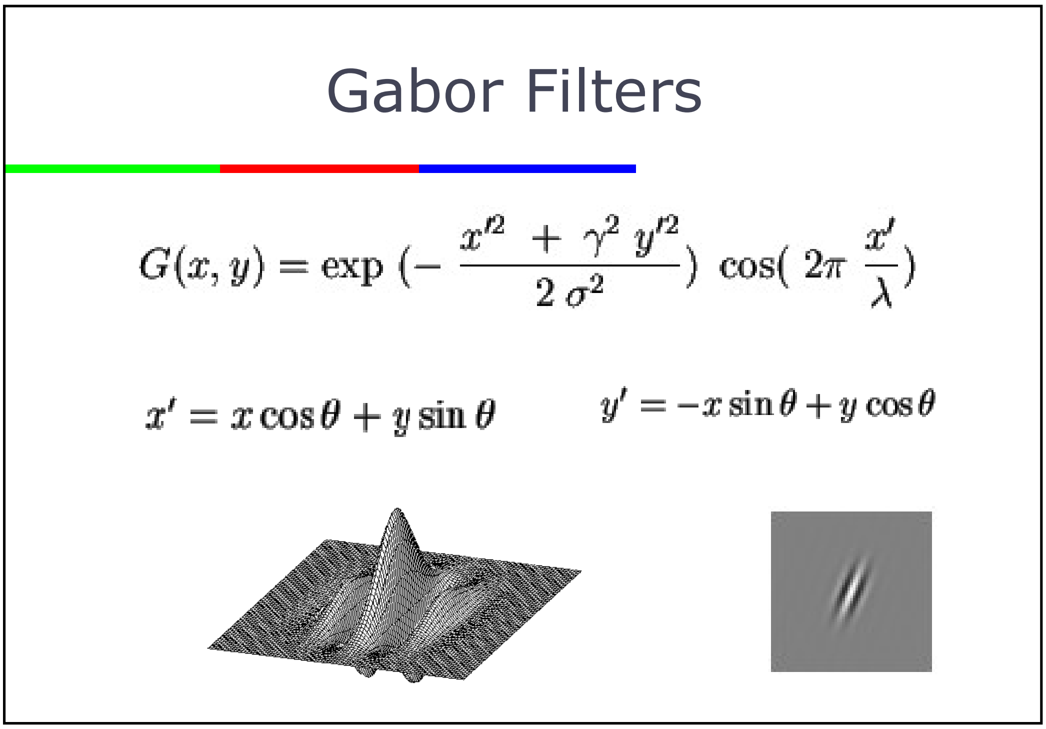
$$\int I\left(x_1,y_1\right) * g_{m\theta}\left(x-x_1,y-y_1\right) dx_1 dy_1 = W_{m\theta}\left(x,y\right)$$



The mean and variance of the output of each filter is used as a descriptor: d = [(m_1, v_1), $(m_2, v_2,)$, ..., $(m_{24,}\ v_{24}\ )$]



Gabor Filters

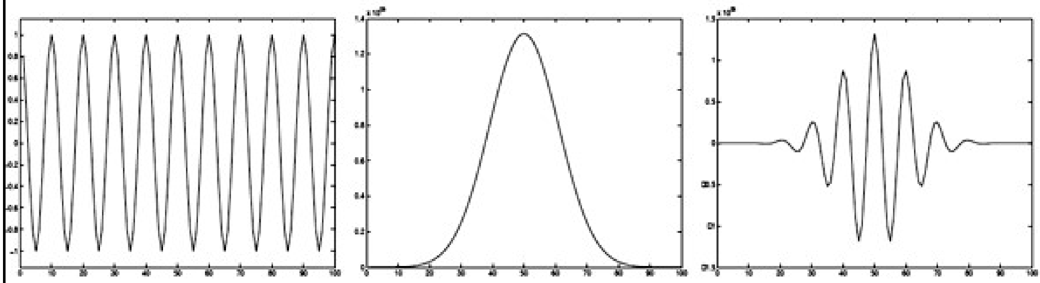


Gabor Filters

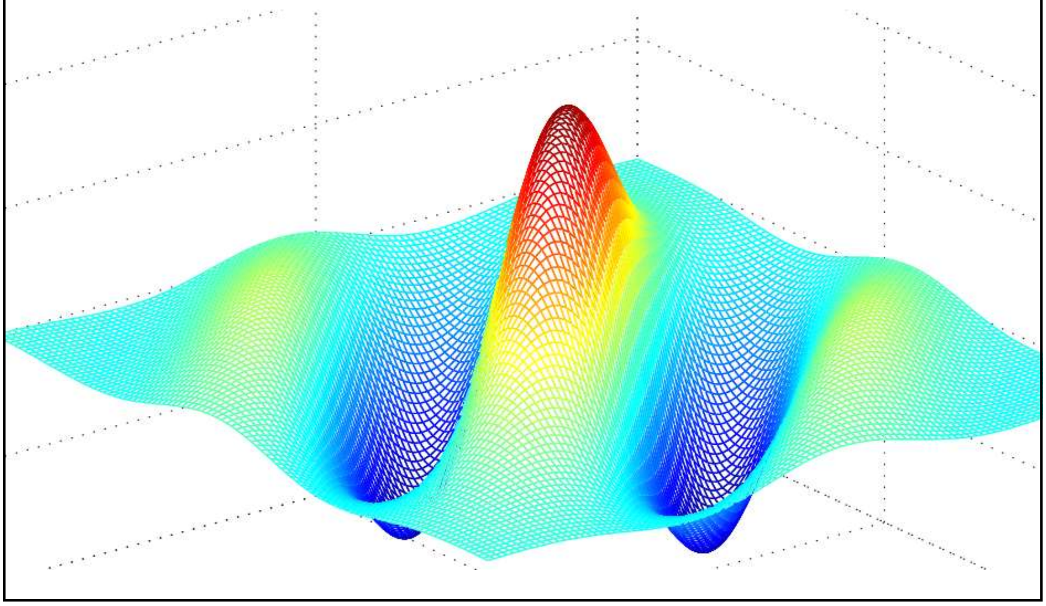
$$G(x, y) = \exp\,(-\,\frac{x'^2 \;+\; \gamma^2\, y'^2}{2\,\sigma^2})\;\cos(\,2\pi\,\frac{x'}{\lambda})$$

$$x' = x\cos\theta + y\sin\theta \qquad y' = -x\sin\theta + y\cos\theta$$

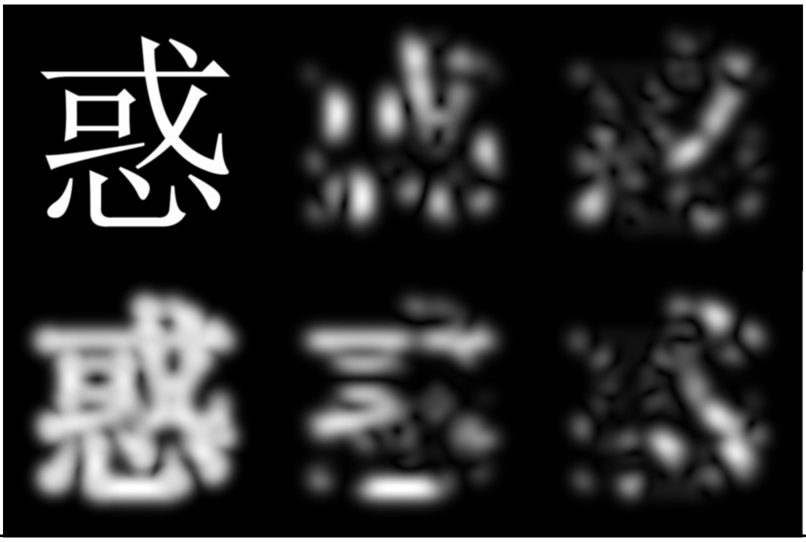
Gabor Filter



Gabor Filter



Texture



Local Features [Lowe2004]

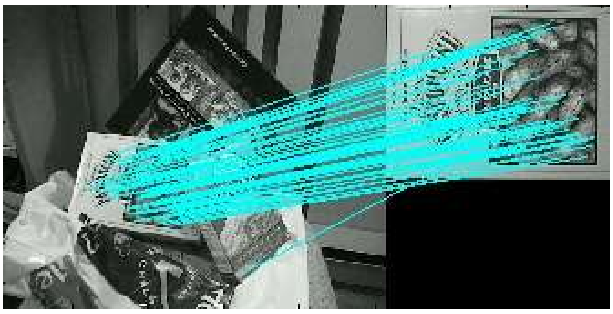
- **Locality**: features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness**: individual features can be matched to a large database of objects
- **Quantity**: many features can be generated for even small objects
- **Efficiency**: close to real-time performance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

Distinctive image features from scale-invariant keypoints.
David G. Lowe, International Journal of Computer Vision,
60, 2 (2004), pp. 91-110

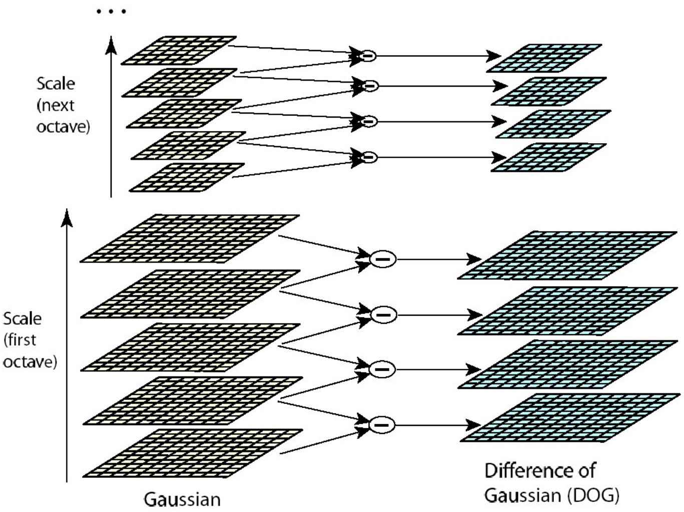
SIFT

- Scale-Invariant Feature Transform

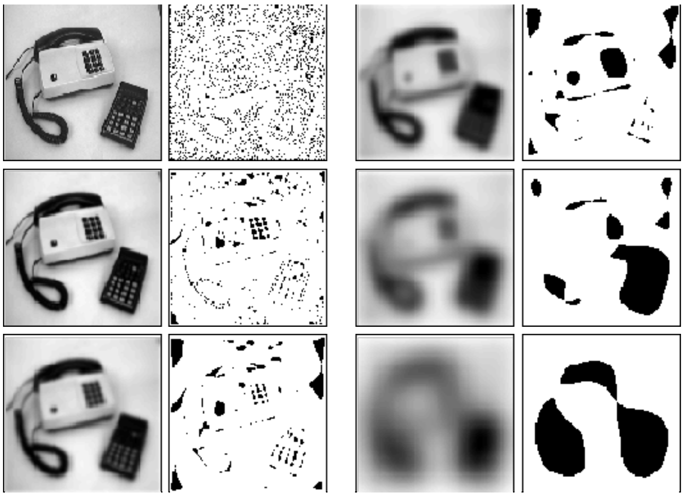
Distinctive image features from scale-invariant keypoints. David G. Lowe, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110



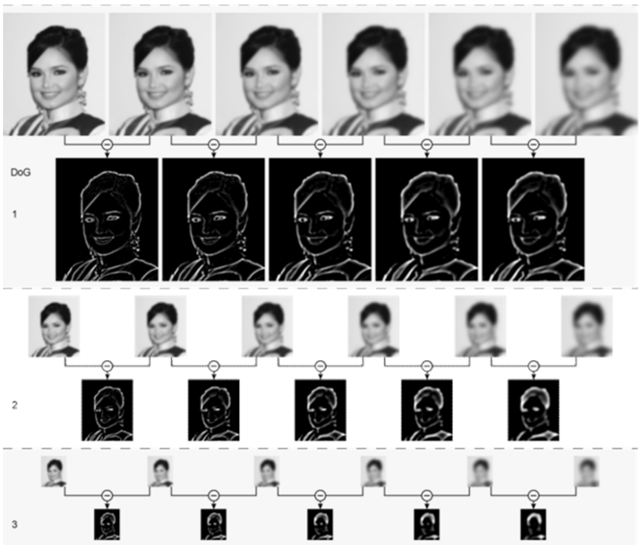
Scale Space Processing



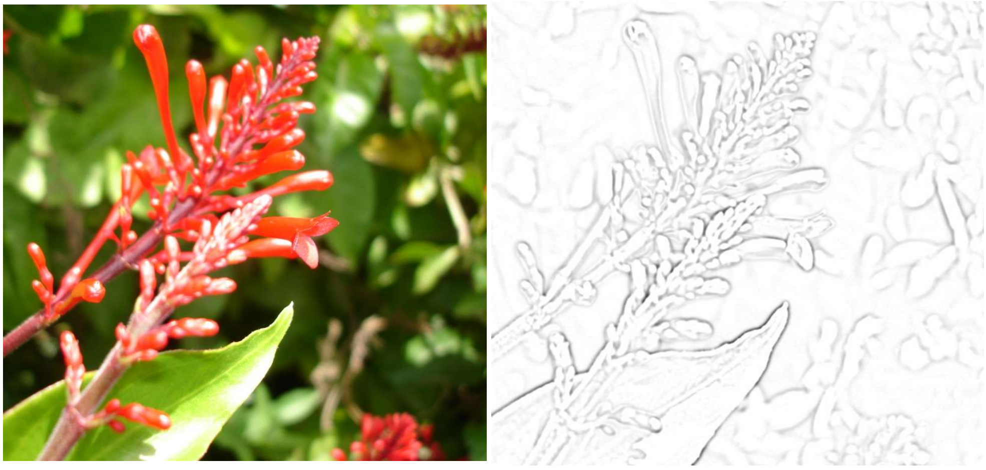
Scale Space



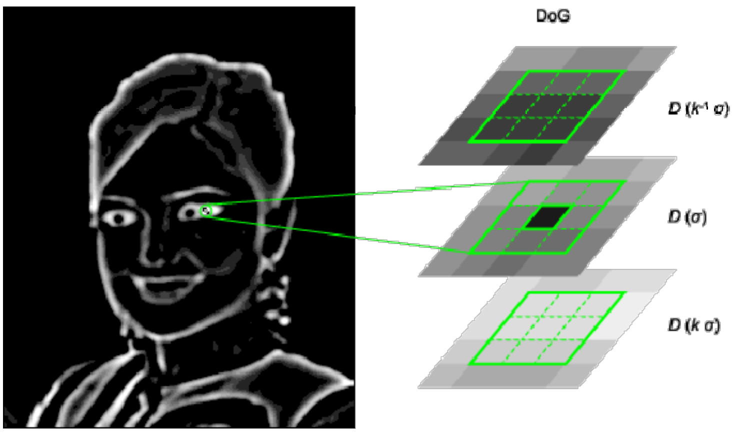
Scale Space Processing



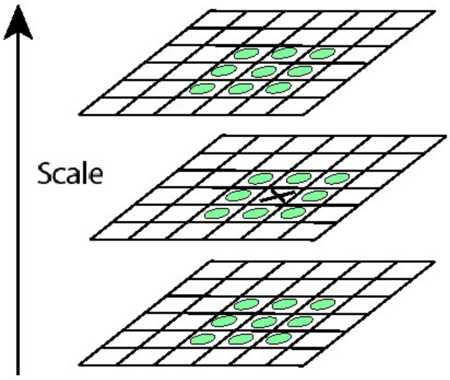
DoG – Difference of Gaussians



Key Point Localization

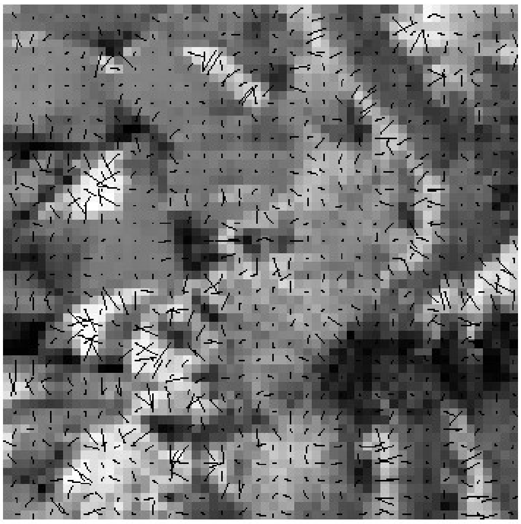


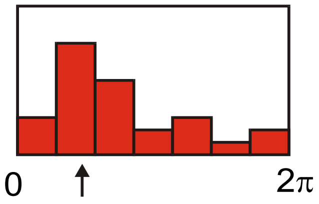
Key Point Localization



Keypoint candidates: Maxima and minima of the difference-of-Gaussian images are Detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles) [Lowe2004]

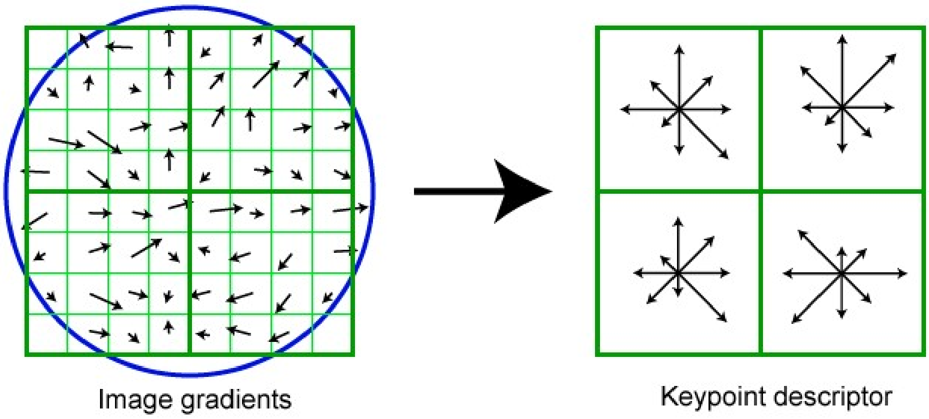
Gradient Orientation





Gradient Histogram

Keypoints and Descriptors

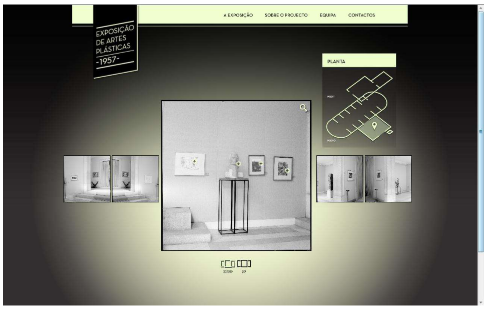


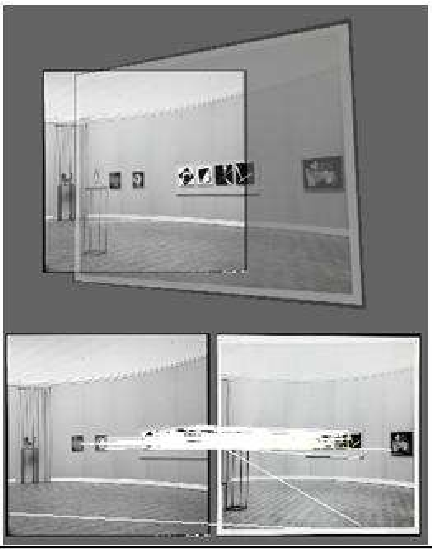
Keypoints and Descriptors

- Orientations assigned to each keypoint location based on local image gradient directions.
- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions in the original proposal
  (in the previous slide 2x2 histogram array)

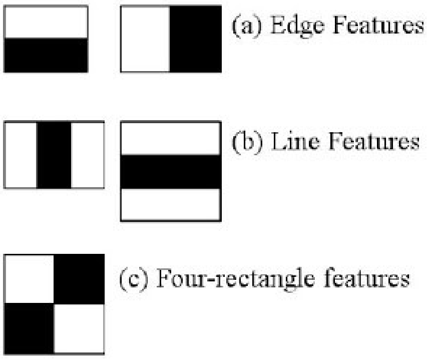
SIFT – Application Example

Image matching to reconstruct a physical space





Face Detection Features

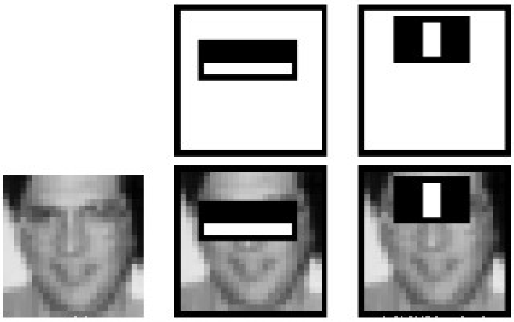


Haar Features

Similar to convolution kernels

Each feature results in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle.

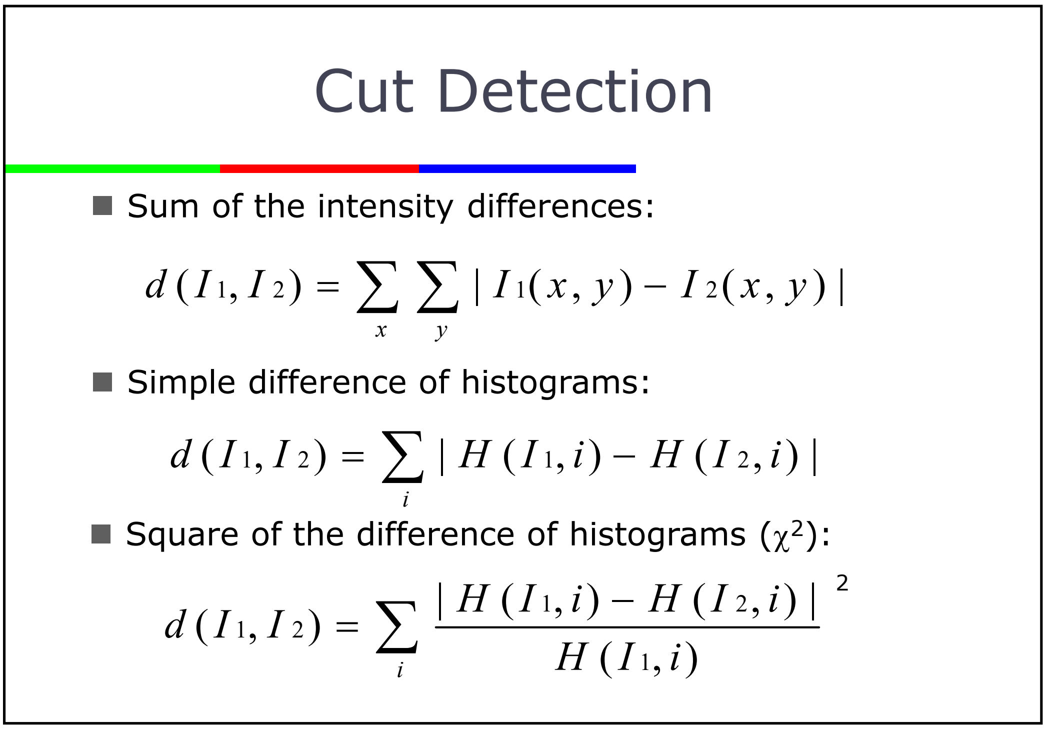
Face Detection Features



These features are then used (and selected) in a cascade of classifiers
Selected features are included if they can perform better than random guessing (detect more than half the cases)

Video - Cut Detection

- Pixels difference
- Histograms difference
- Histograms difference (χ^2)
- Gradual transitions detection
  - Twin-Comparison
  - Models



Cut Detection

- Sum of the intensity differences:

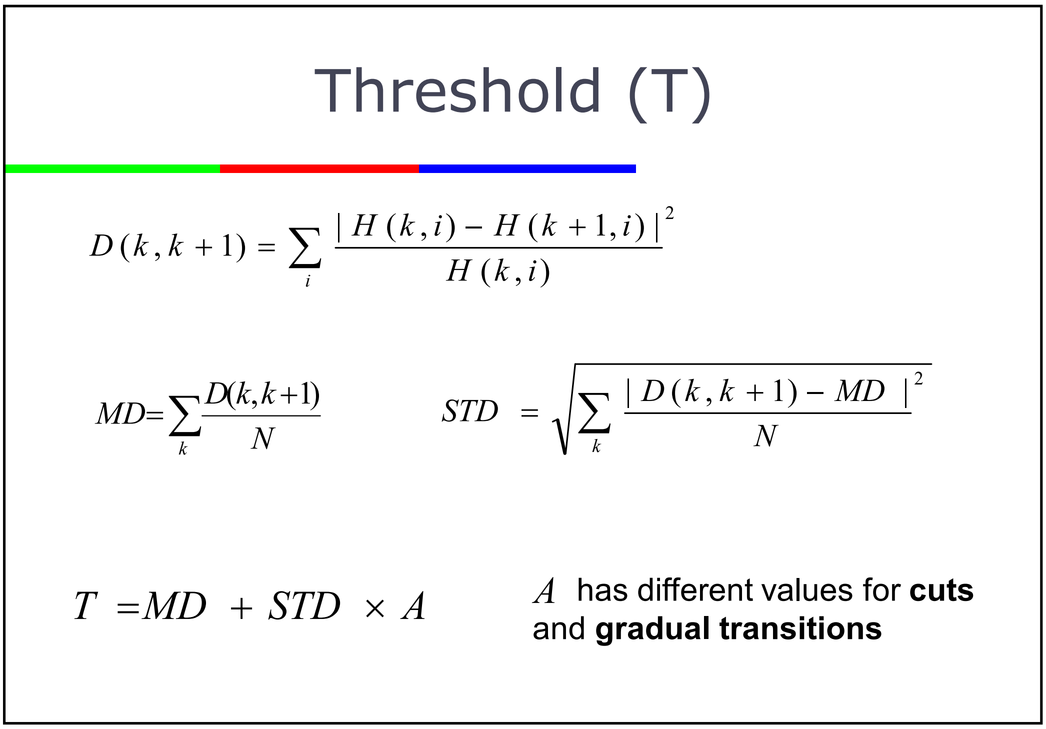
$$d(I_1, I_2) = \sum_x \sum_y | I_1(x, y) - I_2(x, y) |$$

- Simple difference of histograms:

$$d(I_1, I_2) = \sum_i | H(I_1, i) - H(I_2, i) |$$

- Square of the difference of histograms (χ^2):

$$d(I_1, I_2) = \sum_i \frac{| H(I_1, i) - H(I_2, i) |^2}{H(I_1, i)}$$



Threshold (T)

$$D(k, k+1) = \sum_i \frac{| H(k, i) - H(k+1, i) |^2}{H(k, i)}$$

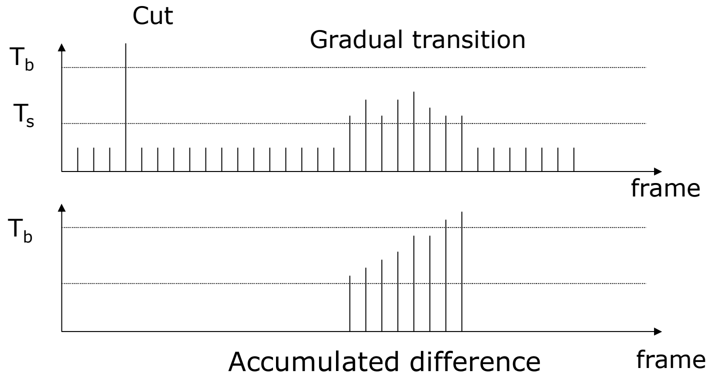
$$MD = \sum_k \frac{D(k, k+1)}{N} \qquad STD = \sqrt{\sum_k \frac{| D(k, k+1) - MD |^2}{N}}$$

$$T = MD + STD \times A$$

A has different values for **cuts** and **gradual transitions**

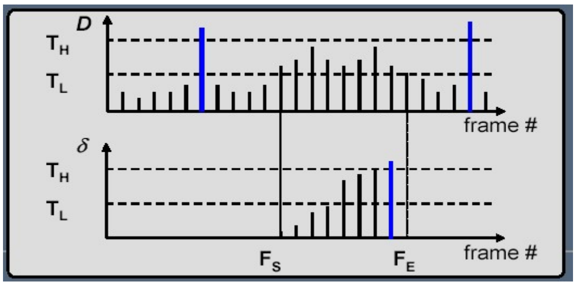
Cut Detection

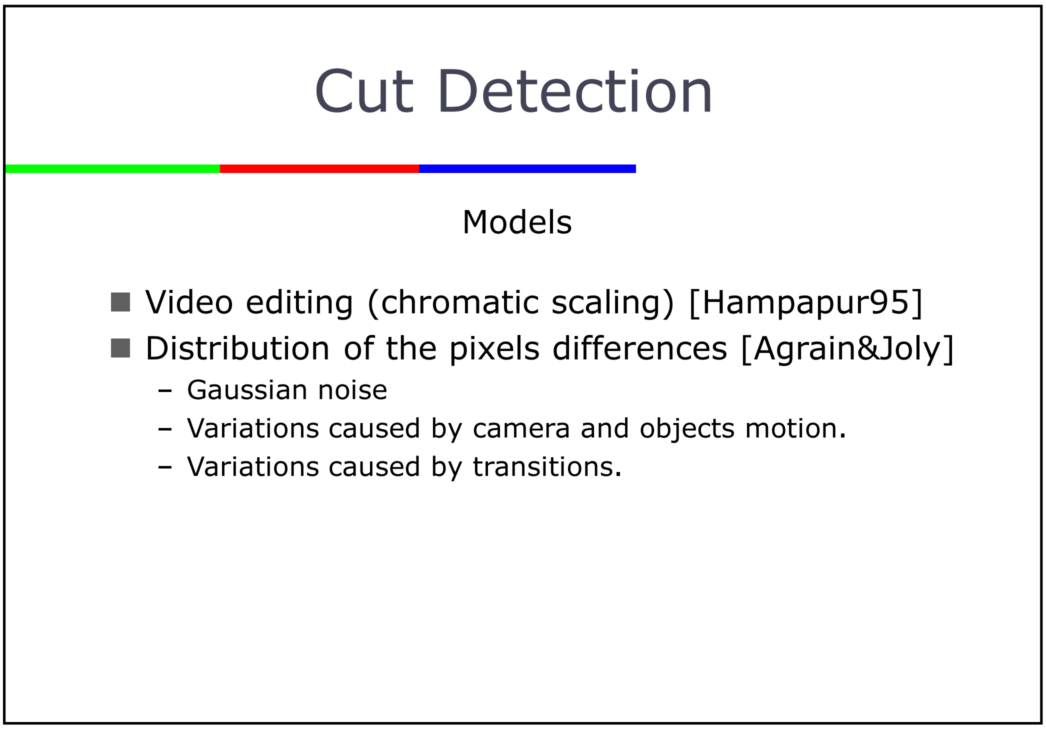
- Twin-Comparison (two levels of comparison)



Cut Detection

- Apply color-and edge histograms for segmentation
- Calculation of histogram difference measure D (e.g., a L1-norm)
- color: twin comparison method
  - **If Tb < diff shot boundary**
  - **Ts < diff < Tb accumulate differences**
  - **diff < Ts nothing**
  - **If the accumulated value (delta) is greater than Tb, a gradual change is detected.**

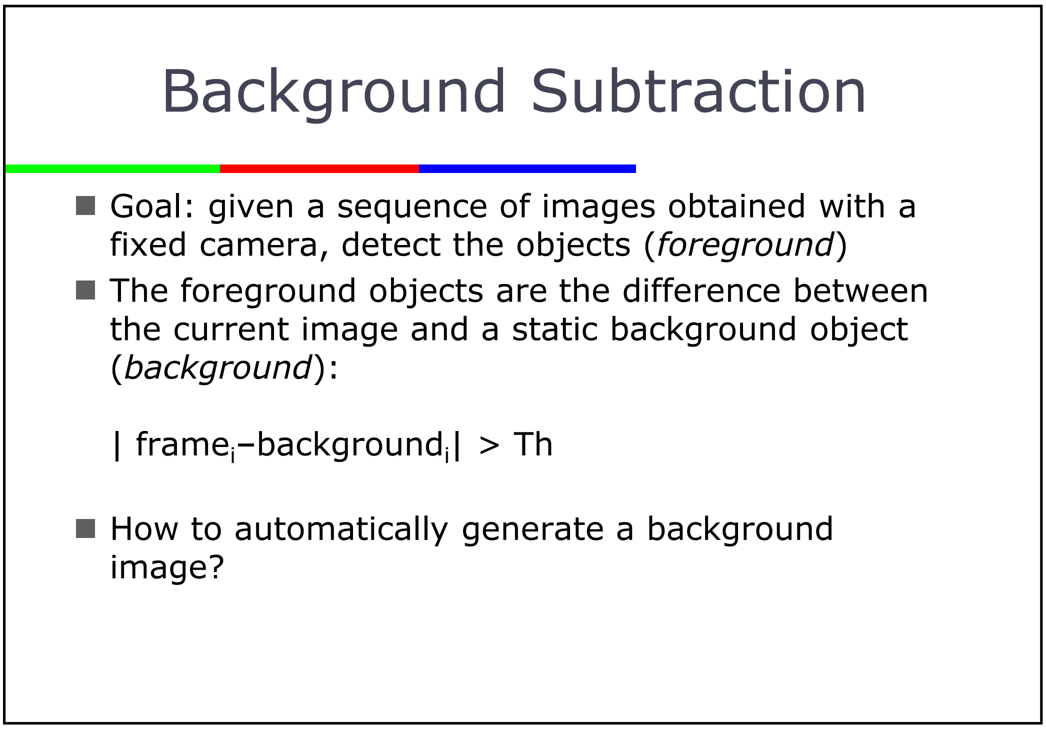




Cut Detection

Models

- Video editing (chromatic scaling) [Hampapur95]
- Distribution of the pixels differences [Agrain&Joly]
  - Gaussian noise
  - Variations caused by camera and objects motion.
  - Variations caused by transitions.



Background Subtraction

- Goal: given a sequence of images obtained with a fixed camera, detect the objects (*foreground*)
- The foreground objects are the difference between the current image and a static background object (*background*):

  $| \text{frame}_i - \text{background}_i | > \text{Th}$

- How to automatically generate a background image?

Background Subtraction

- Background obtained as the average or mode of the N previous images:
  - Fast but for the mode it requires much memory. The memory requirements are: n * size(frame)
- Background updated over time:

$$B_{i+1} = \alpha * F_i + (1 - \alpha) * B_i$$

- With a small value for α (e.g., 0,05) so that results are not much affected in each iteration
- There are no additional memory requirements

Background Subtraction

- For each image each pixel is classified as foreground or background
- What feedback from the background classification model?
  - If the pixel is classified as foreground it is ignored in the background model
  - In this way the background pixels are not changed by pixels that belong to the foreground

Background Subtraction

- Evaluated over time with selectivity:

$$B_{i+1}(x,y) = \alpha.F_t(x,y) + (1-\alpha).B_t(x,y) \text{ if } F_t(x,y) \text{ background}$$

$$B_{i+1}(x,y) = B_i(x,y) \text{ if } F_t(x,y) \text{ foreground}$$