Advanced Video Content Analysis and Video Compression (5LSH0), Module 4

Visual feature extraction
Part I: Color and texture analysis

Sveta Zinger

Video Coding and Architectures Research group, TU/e
(s.zinger@tue.nl)

Introduction – (1)

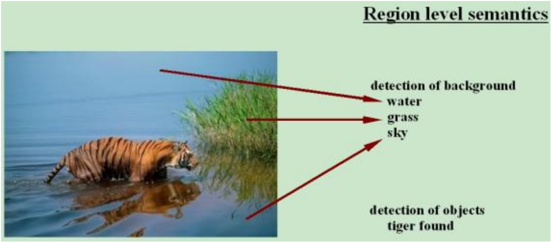
* **Features**
  – What are features?
    Feature – a piece of information which is relevant to solving a problem
    Features can refer to
    • result of a neighbourhood operation applied to the image
    • specific structures in image (points, edges, shapes)

Introduction – (2)

* **Features**
  – Why do we need them?
    To analyse the content, index and extract it, but the input data is large and redundant .
* **Applications**
  – Object recognition / detection / categorization
  – Content Based Image Retrieval (CIBR)
  – Image stitching (e.g. panorama)

Introduction – (3)

* **Object recognition / detection / categorization**



http://chr.millet.googlepages.com/

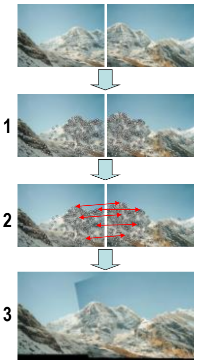
Introduction – (4)

* **Content-based image retrieval**
  Query is a word (“cup”) or an image 
  Retrieve images with corresponding content

http://www-list.cea.fr/fr/programmes/systemes_interactifs/labo_lic2m/piria/w3/pirianet.php

Introduction – (5)

* **Image stitching**
  1. Find feature points in both images
  2. Find corresponding pairs
  3. Find a parametric transformation



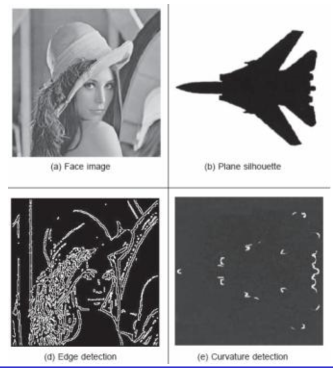
Outline

* **Low-level feature extraction**
  - Edge detection (Prewitt, Sobel, Canny)
  - Interest point operators (Harris, Moravec, DoG)
  - Scale Invariant Feature Transform (SIFT)
* **High-level feature extraction**
  - Shape matching, shape extraction, object description
* **Texture**

Most of the slides are based on the book “Feature Extraction and Image Processing”, M. Nixon and A. Aguado: http://users.ecs.soton.ac.uk/msn/book/

Low-level features

* **Example**



TU/e PdW-SZ-EB / 2016 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 4 Features VCA

Edge detection – (1)

* **What do we detect?**
  - Contrast – difference in intensity
* **First order differences of adjacent points:**
  - Vertical edges

    $E_v(x,y) = |I(x,y) - I(x+1,y)| \quad \forall x \in 1, N-1;\ y \in 1, N$
  - Horizontal edges

    $E_h(x,y) = |I(x,y) - I(x,y+1)| \quad \forall x \in 1, N;\ y \in 1, N-1$
  - Vertical and horizontal edges

$$E_{vh}(x,y) = E_v + E_h = |2 \times I(x,y) - I(x+1,y) - I(x,y+1)| \quad \forall x,y \in 1, N-1$$

TU/e PdW-SZ-EB / 2016 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 4 Features VCA

Edge detection – (2)

- Vertical edges and horizontal edges: example

Original image

Vertical edges

Horizontal edges

Method: differences of adjacent points

TU/e PdW-SZ-EB / 2016 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 4 Features VCA

Edge detection – (3)

* **Template for first-order differences (impulse response of a 2D filter)**

2	-1
-1	0

* **Improved first-order differences *Mx* and *My*: spacing the differenced points by one pixel**

1	0	-1

Mx

1
0
-1

My

TU/e PdW-SZ-EB / 2016 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 4 Features VCA

Edge detection / Roberts (1965) cross operator – (1)

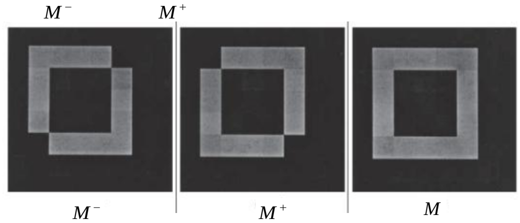
+1	0
0	-1

M^-

0	+1
-1	0

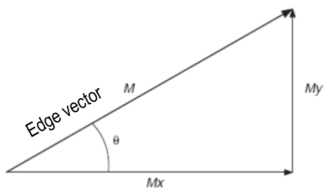
M^+

M^+ and M^- are combined by taking maximum or a sum of them



TU/e PdW-SZ-EB / 2016 Fac. EE SPS-VCA Adv. Topics MMedia Vid. Cod. / 5LSH0 / Module 4 Features VCA

Edge detection / vectorial format



Edge magnitude is the length of the vector

Edge direction is the vector's orientation

Edge detection / Roberts (1965) cross operator – (2)

* **What is the drawback of the Roberts operator?**
  – Its sensitivity to noise
* **How can we improve edge detection (try to avoid noise)?**
  – Extend templates: introduce averaging in the edge detection

Edge detection / Prewitt (1966) operator – (1)

To avoid detecting noise instead of edges, incorporate averaging within the edge detection process: extend vertical template along three rows, horizontal – along three columns

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

Mx

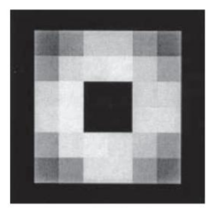
1	1	1
0	0	0
-1	-1	-1

My

Edge detection / Prewitt operator – (2)

Original image

Edge magnitude



Edge detection / Sobel (1970) operator – (1)

Double the weight at the central pixels for both Prewitt templates: obtain Sobel operator

Theoretical basis: consider the optimal forms for smoothing (Gaussian) and for differencing

Sobel operator combines optimal smoothing along one axis with optimal differencing along the other

The larger the edge detection template, the more smoothing there is to reduce noise, but edge blurring becomes a problem.

Edge detection / Sobel operator – (2)

3x3 templates for Sobel operator:

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

Mx

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

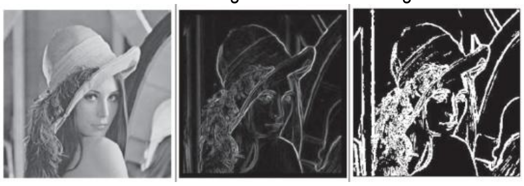
My

Window size	Smoothing coefficients
2	1 1
3	1 2 1
4	1 3 3 1
5	1 4 6 4 1

Window size	Smoothing coefficients
2	1 -1
3	1 0 -1
4	1 1 -1 -1
5	1 2 0 -2 -1

Edge detection / Sobel operator – (3)

Original image | Sobel edge magnitude | Thresholded magnitude



Edge detection / Canny (1986) operator – (1)

* **Very popular technique; three main objectives:**
  - optimal detection with no false responses (reduce the response to noise)
  - good localisation with minimal distance between detected and true edge position (edges to be detected in the right place)
  - single response to eliminate multiple responses to a single edge.

Edge detection / Canny operator – (2)

* **In theory**
  - Detect the zero-crossings of the second directional derivative of the smoothed image in the direction of the gradient where the gradient magnitude of the smoothed image is greater than a threshold
  - i.e. find first-directional derivative's maxima and minima in the direction of the gradient

$$d^2(G*I)/dn^2 = d([dG/dn]*I)/dn$$

where G – Gaussian function, I – image, n – direction of the gradient of the smoothed image

Edge detection / Canny operator – (3)

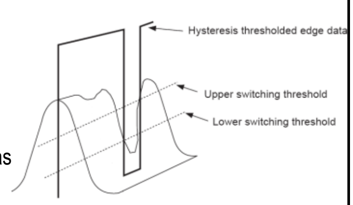
* **In practice**
  - smooth the image using a Gaussian function
  - use the Sobel operator
  - use non-maximal suppression
  - threshold with hysteresis to connect edge points

Edge detection / Canny operator – (4)

* **Non-maximum suppression**
  - locates the highest points in the edge magnitude data
  - given a 3x3 region, a point is at a maximum if the gradient at either side of it is less than the gradient at that point

Edge detection / Canny operator – (5)

* **Hysteresis method**
  - uses two thresholds
  - the high threshold – finds "seeds" for strong edges
  - edges grow as long as the edge strength does not fall below the low threshold



Edge detection / Canny operator – (6)

- **Hysteresis compared to uniform thresholding**

Hysteresis thresholding, upper level = 40, lower level = 10 | Uniform thresholding, level = 40 | Uniform thresholding, level = 10

Edge detection / Canny operator – (7)

- **Adjustable parameters of the algorithm**
  - the size of the Gaussian filter
    - smaller filter => less blurring => detection of small sharp lines
    - larger filter => more blurring => detection of larger smoother edges (like edge of rainbow)
  - thresholds for hysteresis
    - too high => miss important information
    - too low => false edges (noise)

Edge detection / Canny operator – (8)

- **Comparing Canny and Sobel**

Original image | Canny | Sobel

Interest point operators

- **What are interesting points?**
- **Depends on the application**
  - Detect corners → corner detectors
  - Detect motion → discriminative trackable points
  - Detect wheels → circular image regions
- **Commonly, detect points in the image with large intensity variation in some directions**

Java applet with Harris, Moravec detectors:
http://www.cim.mcgill.ca/~dparks/CornerDetector/mainApplet.htm

Interest points / Moravec (1977) operator – (1)

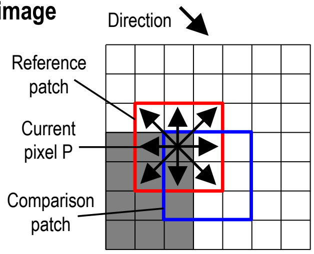
- **One of the earliest 'corner' detectors**
- **Application**
  - Mobile robot navigation → detect obstacles
  - Find similar points in two consecutive video frames
- **Defined interest point as a point with low self-similarity**

Interest points / Moravec operator – (2)

- **For each image pixel P**
  - Consider image region around P (3x3, 5x5 or 7x7 pixels)
  - Compare region with shifted regions (left, right, up, down, diagonals)

Interest points / Moravec operator – (3)

* Example: binary image
* 3x3 window



Interest points / Moravec operator – (4)

* For each image pixel P
  – For each shift, calculate intensity variation V between the two patches over all shifts (u,v)

$$V_{u,v}(x,y) = \sum_{\forall a,b\_in\_window} \left(I(x+u+a, y+v+b) - I(x+a, y+b) \right)^2$$

  – Where shifts (u,v) are (0,1), (1,1), (1,0), (1,-1), (0,-1), (-1,-1), (-1,0) and (-1,1)

Interest points / Moravec operator – (5)

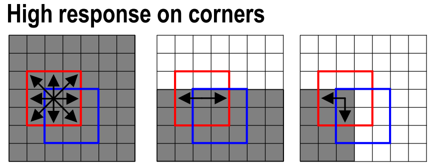
– Construct cornerness measure for each pixel

$$C(x,y) = \min\{V_{u,v}(x,y)\}$$

– Score = minimum of all C(x,y)
– Only high minimum score when all shifts have high difference (on corners)
– Apply threshold to C(x,y) to get points with desired 'level of cornerness'

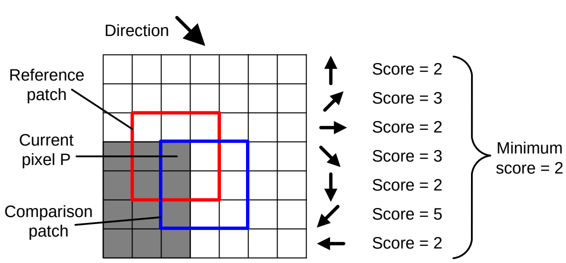
Interest points / Moravec operator – (6)

* Low response in flat areas
* Low response on edges
* High response on corners



Min score = 0 Min score = 0 Min score = 2

Interest points / Moravec operator – (7)



Interest points / Moravec operator – (8)

* Example scores – large view of an image indicating the results of the operator

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	0	0	0	0	0	0	0	0	0	1	1	1	x	x
x	x	0	0	0	0	0	1	1	0	0	1	2	1	x	x
x	x	0	0	0	0	0	2	1	0	0	1	1	1	x	x
x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Interest points / Moravec operator – (9)

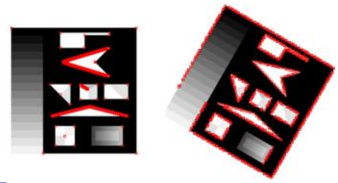
* **Results**
  – Corners are detected, however
  – Sensitive to noise

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	0	0	0	0	0	0	0	0	0	1	1	1	x	x
x	x	0	0	0	0	0	1	1	0	0	1	2	1	x	x
x	x	0	0	0	0	0	2	1	0	0	1	1	1	x	x
x	x	0	0	0	0	0	0	0	0	0	0	0	0	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

http://www.cim.mcgill.ca/~dparks/CornerDetector/mainMoravec.htm

Interest points / Moravec operator – (10)

* **Results**
  – not rotationally invariant
  – If edge not in any the 4 principle directions → detected as being corners



Interest points / Harris (1988) operator – (1)

* **Based upon Moravec operator**
* **Application:**
  – Navigation of mobile robot to detect obstacles
  – Find similar points in two consecutive video frames
* **Invariant to rotation, illumination, noise**

Chris Harris and Mike Stephens, "A Combined Corner and Edge Detector", Proc. of The Fourth Alvey Vision Conference, Manchester, pp 147-151. 1988

Interest points / Harris operator – (2)

* **Based on local autocorrelation of image**
  – Comparable to the Moravec detector
  – Approximate the increment between image points by the directional derivative

$$I(x+a+u, y+b+v) = I(x+a, y+b) + \\ + \frac{\partial I(x+a, y+b)}{\partial x}u + \frac{\partial I(x+a, y+b)}{\partial y}v$$

Interest points / Harris operator – (3)

– Then the autocorrelation function is

$$V_{u,v}(x,y) = \\ = \sum_{\forall a,b\_in\_window} \left(\frac{\partial I(x+a, y+b)}{\partial x}u + \frac{\partial I(x+a, y+b)}{\partial y}v \right)^2$$

Interest points / Harris operator – (4)

– By expansion of the squared term we obtain

$$V_{u,v}(x,y) = A(x,y)u^2 + 2C(x,y)uv + B(x,y)v^2$$

where

$$A(x,y) = \sum_{\forall a,b\_in\_window} \left(\frac{\partial I(x+a, y+b)}{\partial x} \right)^2$$

$$B(x,y) = \sum_{\forall a,b\_in\_window} \left(\frac{\partial I(x+a, y+b)}{\partial y} \right)^2$$

$$C(x,y) = \sum_{\forall a,b\_in\_window} \left(\frac{\partial I(x+a, y+b)}{\partial x} \right) \left(\frac{\partial I(x+a, y+b)}{\partial y} \right)$$

Interest points / Harris operator – (5)

– The curvature is measured by

$$\kappa_k(x,y) = A(x,y)B(x,y) - C(x,y)^2 - k\left(A(x,y) + B(x,y)\right)^2$$

– parameter k controls the sensitivity of the detector

– the higher k, the more sensitivity to changes (and to noise)

Derivation of the formula: M. Nixon and A. Aguado, "Feature Extraction and Image Processing", 2002, pp.141-143.

Interest points / Harris operator – (6)

– The curvature can be written as

$$\kappa_k(x,y) = \alpha\beta - k(\alpha + \beta)^2$$

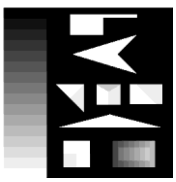
– the values of α and β are proportional to the autocorrelation function along the principal axes

– $\alpha\beta = A(x,y)B(x,y) - C(x,y)^2$ – makes the measure large when the values of α and β increase

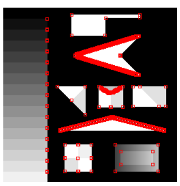
– $\alpha + \beta = A(x,y) + B(x,y)$ – decreases the values on flat borders => we want to find corners

Interest points / Harris operator – (6)

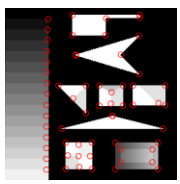
* Results



Input image



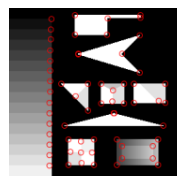
Moravec



Harris

Interest points / Harris operator – (7)

* **Results**
  – Corners are well detected
  – Rotationally invariant
  – Applied for finding points for matching pairs of images



* **What is the drawback of image matching based on Harris interest points?**
  – Sensitivity to scale: does not provide a good basis for matching images of different sizes

Interest points
DoG (Lindeberg, 1994) – Difference of Gaussians – (1)

* **How to find image locations invariant with respect to translation, scaling and rotation?**
  – Koendering (1984) & Lindeberg (1994) showed that Gaussian kernels are the only possible for scale-space analysis

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y),$$

scale Gaussian convolution Intensity image

Interest points
DoG – Difference of Gaussians – (2)

* **Detect stable points in scale-space**
  – Convolve input image with difference-of-Gaussians separated in scale by factor k (commonly sqrt(2))

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma). \end{aligned}$$

factor

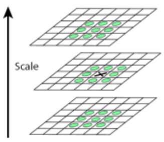
scale-space function

  – Build pyramid by subsampling the previous level with factor 1.5 factor

49

Interest points
DoG – Difference of Gaussians – (3)

* **Generate interest-points**
  - Detection of local extrema (maxima & minima)
  - Compare each point in D(x,y) with neighbors at same scale and at neighboring scales
  - Point is interest-point if smaller or larger than all other points



Img. from Lowe, IJCV 2004

50

Scale Invariant Feature Transform – (1) (Lowe, 1999)

* **Describes gradient in local region**
* **Aims at finding features invariant to**
  - image scaling
  - translation
  - rotation
  - illumination changes
  - 3D projection
  - occlusion

51

Scale Invariant Feature Transform – (2)

* **Scale Invariant Feature Transform (SIFT)**
  - Proposed by David Lowe in 1999 *
  - Extended journal version in 2004 **
* **Widely used, high performance**
  - Interest point detector → Difference-of-Gaussians
  - How to describe the interest points?

* D. G. Lowe, Object Recognition From Local Scale-invariant Features, 1999, *Proc. of the International Conference on Computer Vision (ICCV)*, pp 1150--1157, 1999

** D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

52

Scale Invariant Feature Transform – (3)

* **Major stages of SIFT feature extraction**
  - **Scale-space extrema detection**: difference-of-Gaussians function to identify interest points
  - **Keypoint localization**: determine location, scale and stability of keypoints
  - **Orientation assignment**: based on local image gradient directions
  - **Keypoint descriptor**: local image gradients are measured at the selected scale around each keypoint

53

Scale Invariant Feature Transform – (4)

* **Scale-space extrema detection**
  - Convolve input image with the Gaussian using $\sigma = \sqrt{2}$ to give an image A
  - Produce an image B by a further incremental smoothing with $\sigma = \sqrt{2}$
  - Difference of Gaussians is A-B
  - Generate the next level of pyramid – subsample the image B with a factor of 1,5 or 2 and repeat the process

54

Scale Invariant Feature Transform – (5)

* **Scale-space extrema detection**
  - Test whether the pixel is maximum or minimum compared to its 8 neighbors
  - If yes, repeat the test for the level above

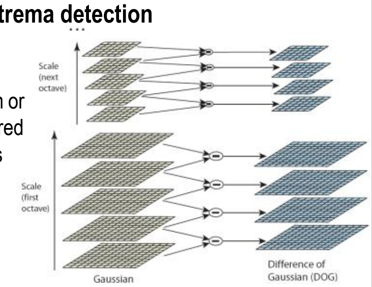


Image from D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

Scale Invariant Feature Transform – (6)

* **Keypoint localization**
  – At each key location, the smoothed image A at each level of the pyramid is processed to extract image gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

  – Apply threshold on minimum contrast
  – Eliminate edge responses

Scale Invariant Feature Transform – (7)

* **Keypoint localization: example**

Original image

Example from D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

Scale Invariant Feature Transform – (8)

* **Keypoint localization: example**

The initial 832 locations of keypoints at maxima and minima of the Difference-of-Gaussian function

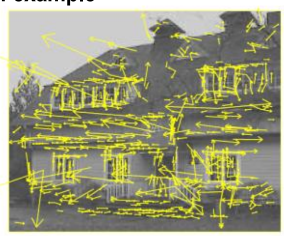
Keypoints are displayed as vectors indicating scale orientation and location

Example from D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

Scale Invariant Feature Transform – (9)

* **Keypoint localization: example**

After applying a threshold on minimum contrast, 729 points remain

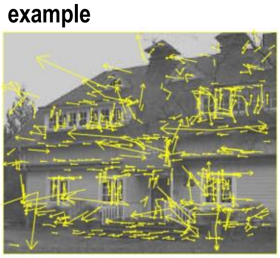


Example from D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

Scale Invariant Feature Transform – (10)

* **Keypoint localization: example**

The final 536 keypoints that remain after edge response removal



Example from D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, *International Journal of Computer Vision (IJCV2004)*, Vol. 60, Num. 2, January, 2004

Scale Invariant Feature Transform – (11)

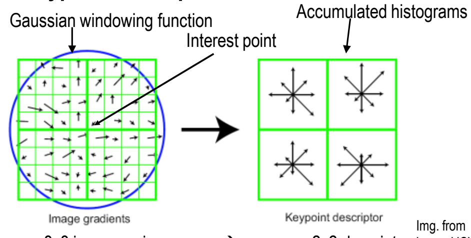
* **Orientation assignment**
  – Orientation is determined by the peak in a histogram of local image gradient orientation (see slide 62)
  – The orientation histogram is created using a Gaussian-weighted window with smoothing higher than the current smoothing scale
  – The histogram has 36 bins covering the 360 degree range of rotations and is smoothed before before peak selection

Scale Invariant Feature Transform – (12)

* **Keypoint descriptor**
  – Compute gradient magnitude and orientation around each keypoint
  – Peaks in histogram correspond to dominant directions of local gradients (see slide 62)

Scale Invariant Feature Transform – (13)

* **Keypoint descriptor**



8x8 image region → 2x2 descriptor

Img. from Lowe, IJCV 2004

Scale Invariant Feature Transform – (14)

* **Invariant to illumination, small contrast variations, rotation, scale**
  – Many applications: object recognition, indexing and retrieval, automatic panorama stitching
* **How can SIFT performance be improved?**
  – Incorporate color, texture, edge groupings in the features
  – Get features from multiple views and combine them into a single model

Scale Invariant Feature Transform – (15)

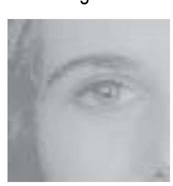
Example

High-level feature extraction

* **Concerns finding shapes in images**
* **Example: automatic face recognition**
  – extract the eyes, the ears and the nose
  – To find them, use their shape
    – white part of the eyes => ellipsoidal
    – Mouth, eyebrows => two lines

Shape matching
Thresholding – (1)

If the shape can be extracted by its brightness, simply apply thresholding

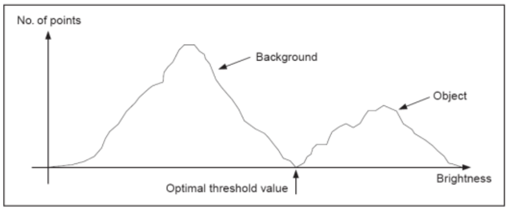


Eye image

Thesholding the eye image

Shape matching
Thresholding – (2)

Histogram: optimal thresholding



Shape matching
Subtraction

Subtract an image from a known background before thresholding

Image of walking subject | After background subtraction | After thresholding

Shape matching
Thresholding and subtraction

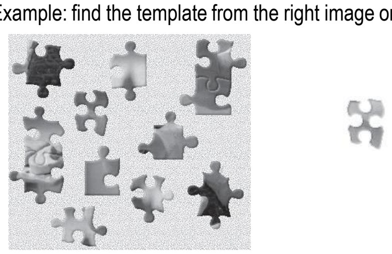
* **Summary**
  – Simple and fast techniques
  – Sensitive to noise, illumination and occlusion
* **To avoid this sensitivity, use higher-level information – how pixels are connected within the shape**

Shape matching
Template matching – (1)

* **Template – sub-image that contains the shape to find**
* **Template matching**
  – center the template on an image point
  – count how many template points match those on the image
  – repeat it for the whole image
  – the shapes lie where the count was maximal

Shape matching
Template matching – (2)

Example: find the template from the right image on the left one



Shape matching
Template matching – (3)

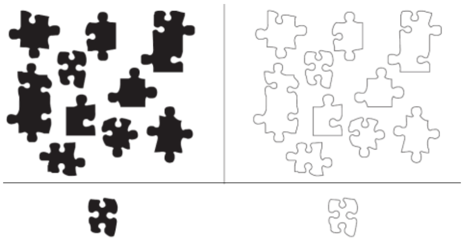
* **Possible generalization**
  – for scale and rotation invariance, try templates of different size and orientation
    – computationally expensive
* **Maximum likelihood estimation**

$$\min e = \sum_{\forall a,b\_in\_window} \left(I(x+a, y+b) - T(a,b)\right)^2$$

  – minimize the squared error (differences between template and image points)

Shape matching
Template matching – (4)

Particular implementation: binary

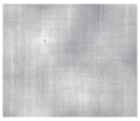


Shape matching
Template matching – (5)

Accumulator space – stores match of the template to the image at different locations (2D array with differences between template and image at different positions)

For the grey level image

For the binary image



For the edge image

Shape matching
Template matching – (6)

The best accumulator – for edges:

for grey scale and binary images some match happens even when the template is not at the best position

More degrees of freedom (rotation, scale) => more dimensions of accumulator space

Flexible shape extraction

- **For template matching the shape is fixed**
- **What if the exact shape is unknown?**
  - techniques that should evolve to the target solution
    - for example, active contours (snakes)

Flexible shape extraction / Snakes – (1)

- **When are snakes applied?**
  - natural objects
    - bananas have similar shapes but not exactly the same
  - medical imaging – objects are similar but not exactly
    - no exact representation of a vein's shape
  - video sequences – object segmentation, detection, tracking
    - some objects move, change over time

Flexible shape extraction / Snakes – (2)

- **What is a snake?**
  - based on energy minimization
  - deforms to fit local minima
  - local, not global, => initial location must be provided

Flexible shape extraction / Snakes – (3)

* **Energy to minimize is a sum of**
  – internal energy caused by stretching and bending
  – measure of the attraction of image features such as contours
  – measure of external constraints either from higher level shape information or user applied energy

Flexible shape extraction / Snakes – (4)

Example

Source: http://www.engr.uconn.edu/~cmli/research/segeff.html

Texture – (1)

* **What is texture?**
  – No mathematical definition exists
  – Oxford dictionary (1996)
    – arrangement of threads etc. in textile fabric, characteristic feel due to this; arrangement of small constituent parts, perceived structure, (of skin, rock, soil, organic tissue, literary work, etc.); representation of structure and detail of objects in art; . . .

Texture – (2)

Examples
(source: http://textures.forrest.cz/)

Fabric

Wood

Nature

Texture description

* **Find measurements that characterize a texture**
* **We consider structural (transform-based) and statistical approaches**
* **Texture description helps to**
  – classify the texture
  – segment an image according to its texture content

Texture description / Structural approach

* **Find and normalize Fourier transform**

$$FI = FFT(I); \qquad NFI_{u,v} = \frac{|FI(u,v)|}{\sqrt{\sum_{(u\neq 0)\wedge(v\neq 0)} |FI(u,v)|^2}}$$

* **Entropy, energy and inertia**

$$h = \sum_{u=1}^{N}\sum_{v=1}^{N} NFI(u,v)\log(NFI(u,v)) \qquad e = \sum_{u=1}^{N}\sum_{v=1}^{N}(NFI(u,v))^2$$

$$i = \sum_{u=1}^{N}\sum_{v=1}^{N}(u-v)^2\, NFI(u,v)$$

Texture description / Statistical approach

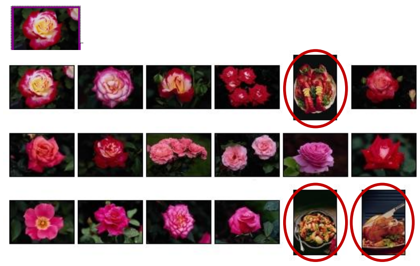
* **Co-occurrence matrix of Haralick**
  - elements – counts of the number of pixel pairs for specific brightness levels, when separated by some distance and at some relative inclination
  - for brightness levels $b1$ and $b2$ the co-occurrence matrix C is

$$C(b1,b2)=\sum_{x=1}^{N}\sum_{y=1}^{N}\left(I(x,y)=b1\right)\wedge\left(I(x',y')=b2\right)$$

Texture-based image retrieval

Query

Retrieved images



Example

(source: http://www-list.cea.fr/fr/programmes/systemes_interactifs/labo_lic2m/piria/w3/pirianet.php)

Summary and conclusions – (1)

* **How to detect edges?**
  - Sobel operator – may be good for simple cases
  - Canny operator – better results, but parameters have to be adjusted
* **How to find interest points?**
  - Harris operator – if no scale variations presumed
  - Difference-of-Gaussians – otherwise

Summary and conclusions – (2)

* **How to extract features of an object?**
  - SIFT – works well, but depends on many parameters
* **How to extract shapes?**
  - difficult problem, success for particular conditions
    - background is known (thresholding, substraction)
    - shape is fixed (template matching)
    - hypothesis about the shape and its location is used, while shape may vary (snakes)

Summary and conclusions – (3)

* **Can texture help?**
  - yes, often works well if the texture is present
    - can also be used for image segmentation
* **And if we have color images?**
  - use histograms of colors
    - can be combined with edges

References

- M. Nixon and A. Aguado, “Feature Extraction and Image Processing”, 2002
- D. Lowe, Distinctive Image Features From Scale-Invariant Keypoints, International Journal of Computer Vision (IJCV2004), Vol. 60, Num. 2, January, 2004