

Introduction to Medical Imaging (5XSA0)

Visual feature extraction Color and texture analysis

Sveta Zinger

(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?

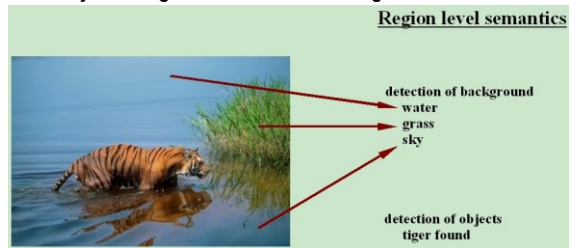
To analyse the content, index and extract it, but the input data is large and redundant .

* 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

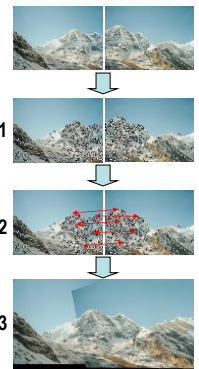


http://www-list.cea.fr/fr/programmes/systemes_interactifs/labo_lic2m/piria/w3/pirinet.php

Introduction – (5)

* Image stitching

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



Outline

7

- * **Low-level feature extraction**
 - Edge detection (Prewitt, Sobel, Canny)
 - Interest point operators (Harris, Moravec, DoG)
- * **High-level feature extraction**
 - Shape matching, image morphology, distance transform
- * **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

8

* Example



Edge detection – (1)

9

- * **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

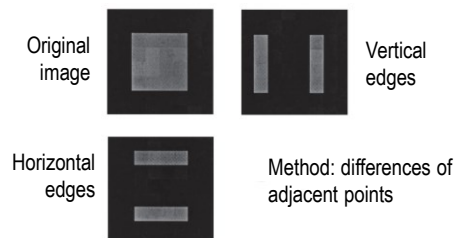
$$E_h(x, y) = |I(x, y) - I(x, y+1)| \quad \forall x \in 1, N; y \in 1, N-1$$
 - Vertical and horizontal edges

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

Edge detection – (2)

10

- Vertical edges and horizontal edges: example



Edge detection – (3)

11

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

2	-1
-1	0

- * **Improved first-order differences M_x and M_y : spacing the differenced points by one pixel**

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

M_x

1
0
-1

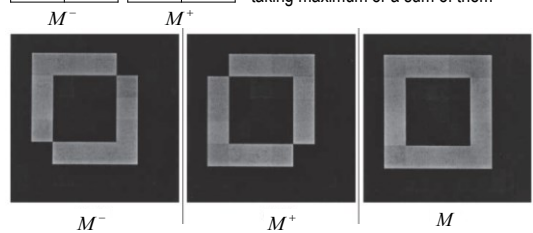
M_y

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

12

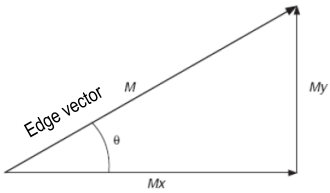
+1	0	0	+1
0	-1	-1	0

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




13

Edge detection / vectorial format



Edge magnitude is the length of the vector


Edge direction is the vector's orientation

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

14

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

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

15

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

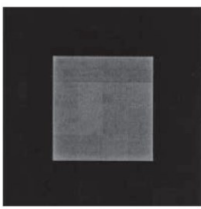
My

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

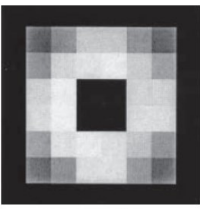
16


Edge detection / Prewitt operator – (2)

Original image



Edge magnitude



TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

17


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.

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

18

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

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

Edge detection / Sobel operator – (3)

19

Original image

Sobel edge
magnitude

Thresholded
magnitude



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

20

* 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)

21

* 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)

22

* 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)

23

* Non-maximum suppression

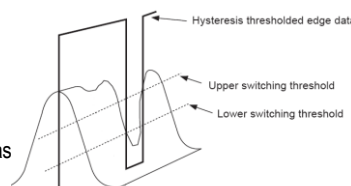
- locates the highest points in the edge magnitude data
- given a 3x3 region, a point is at a maximum if the gradient at either side of it is less than the gradient at that point

Edge detection / Canny operator – (5)

24

* 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

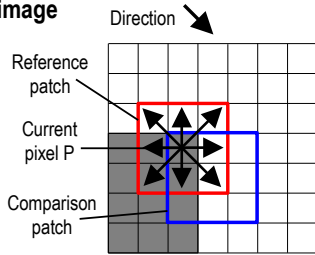
Interest points / Moravec operator – (2)

* 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)

$$V_{u,v}(x,y) = \sum_{\forall a,b \text{ in window}} (I(x+u+a, y+v+b) - I(x+a, y+b))^2$$

- 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)

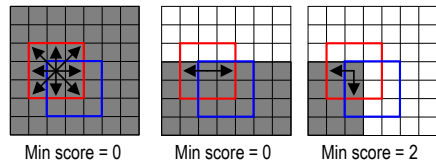
- Construct cornerness measure for each pixel

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

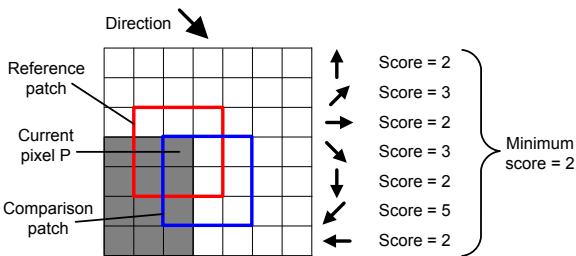
- Score = minimum of all C(x,y)
- Only high minimum score when all shifts have high difference (on corners)
- Apply threshold to C(x,y) to get points with desired 'level of cornerness'

Interest points / Moravec operator – (6)

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



Interest points / Moravec operator – (7)



Interest points / Moravec operator – (8)

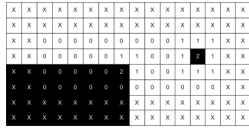
* Example scores – large view of an image indicating the results of the operator

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



<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

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

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

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

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

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

Interest points / Harris operator – (5)

- The curvature is measured by

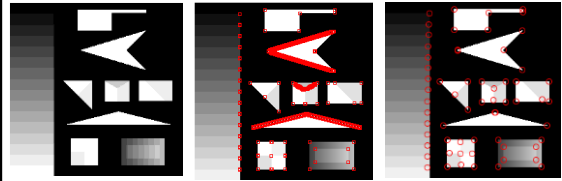
$$\kappa_k(x, y) = A(x, y)B(x, y) - C(x, y)^2 - k(A(x, y) + B(x, y))^2$$

- 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)

* Results

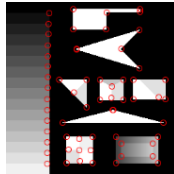


Input image Moravec Harris

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

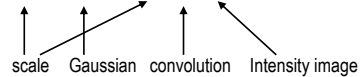
Interest points

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

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



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))

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ = L(x, y, k\sigma) - L(x, y, \sigma).$$



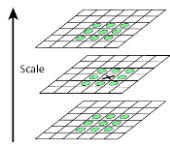
- Build pyramid by subsampling the previous level with factor 1.5 factor

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

High-level feature extraction

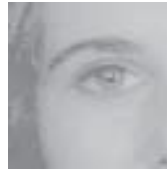
49

- * **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)

50

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



Eye image

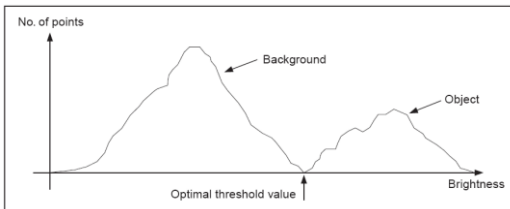


Thresholding the eye image

Shape matching Thresholding – (2)

51

Histogram: optimal thresholding



Shape matching Subtraction

52

Subtract an image from a known background before thresholding



Image of walking subject



After background subtraction



After thresholding

Shape matching Thresholding and subtraction

53

- * **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)

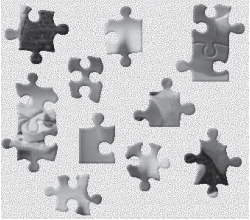
54


- * **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

55

Shape matching Template matching – (2)

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




TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

56

Shape matching Template matching – (3)

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

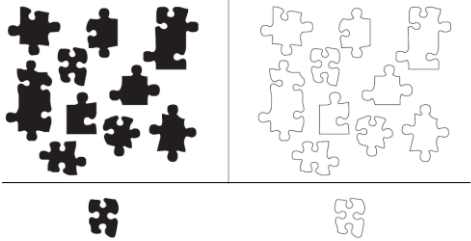
$$\min e = \sum_{\forall a,b \text{ in window}} (I(x+a, y+b) - T(a,b))^2$$
 - minimize the squared error (differences between template and image points)


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

57

Shape matching Template matching – (4)

Particular implementation: binary




TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 


58

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


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

59

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


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

60

Image morphology

- * **Mathematical morphology**
 - Tool for extracting image components such as boundaries, skeletons, etc.
 - Transition from image processing (inputs and outputs are images) to image analysis methods (inputs are images, outputs describe the contents of the image)

(The slides are based on "Digital Image Processing Using Matlab", R. C. Gonzalez, R. E. Woods, S. L. Eddins)

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

61

Basic concepts from set theory – (1)

FIGURE 9.1
 (a) Two sets A and B . (b) The union of A and B . (c) The intersection of A and B . (d) The complement of A and B . (e) The difference between A and B .

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

62

Basic concepts from set theory – (2)

FIGURE 9.2
 (a) Translation of A by z . (b) Reflection of B . The sets A and B are from Fig. 9.1.

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

63

Dilation and erosion

TABLE 9.1 Using logical expressions in MATLAB to perform set operations on binary images.

Set Operation	MATLAB Expression for Binary Images	Name
$A \cap B$	$A \& B$	AND
$A \cup B$	$A B$	OR
A^c	$\sim A$	NOT
$A - B$	$A \& \sim B$	DIFFERENCE

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

64

Dilation and erosion: definitions

- * **Dilation**
 - The dilation of A by B is the set of all displacements z such that \hat{B} and A overlap by at least one element
- * **Erosion**
 - The erosion of A by B is the set of all points z such that B , translated by z , is contained in A

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

65

Dilation

The structuring element translated to these locations does not overlap any 1-valued pixels in the original image.

When the origin is translated to the "•" locations, the structuring element overlaps 1-valued pixels in the original image.

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

66

Dilation: example

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

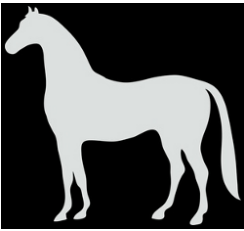
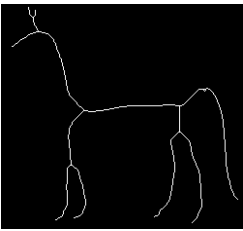
Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Structuring element $se = [0 \ 1 \ 0; \ 1 \ 1 \ 1; \ 0 \ 1 \ 0];$


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features VCA

73

Skeletons: example

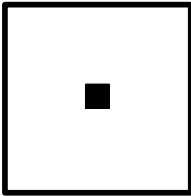
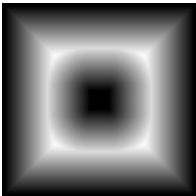
Source: <https://opencv-code.com/quick-tips/getting-the-skeleton-of-a-binary-image-in-python/>

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 


74

Distance transform

The value of each pixel is replaced by its distance to the nearest background pixel

More examples at <http://reference.wolfram.com/language/ref/DistanceTransform.html>
 Distance metrics: <http://nl.mathworks.com/help/images/distance-transform.html>


TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

75

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; . . .

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 


76

Texture – (2)


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

Wood Fabric




Nature

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

77

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

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

78

Texture description / Structural approach


- * Find and normalize Fourier transform

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

- * Entropy, energy and inertia

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

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

TU/e PdW-SZ / 2015 Fac. EE SPS-VCA Introduction Med. Imaging / 5XSA0 / Module 04 Features 

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

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

Texture-based image retrieval

Query



Retrieved images



Example

(source: http://www-list.cea.fr/fr/programmes/systemes_interactifs/labo_lic2m/pirial/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 shapes?

- difficult problem, success for particular conditions
 - background is known (thresholding, subtraction)
 - shape is fixed and noise is low (template matching, image morphology)

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
- Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using Matlab", Pearson Education, 2004
 - Chapter 9
- Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 3rd edition, Prentice Hall, 2008
 - Chapter 9