

Filters in Image Processing

Analysing Images through Visual Descriptors

David Svoboda and Tomáš Majtner

email: svoboda@fi.muni.cz
Centre for Biomedical Image Analysis
Faculty of Informatics, Masaryk University, Brno, CZ



May 14, 2018

Outline

- 1 Motivation
- 2 Basic idea for image descriptors
- 3 Image classification
- 4 Most common image descriptors
 - Haralick features
 - Local binary patterns (LBP)
 - MPEG-7 descriptors
 - Scale-invariant feature transform (SIFT)
 - Zernike features
 - Moment invariants

Motivation



- Unknown image
- No meta information
- How to determine, what is in the image?

Motivation

- Results of a Google search for keyword 'obama' (from Nov. 2011)



Motivation

- Results of searching for visually similar images of the official photo of president Obama (from Nov. 2011)



Basic idea for image descriptors

What are image descriptors?

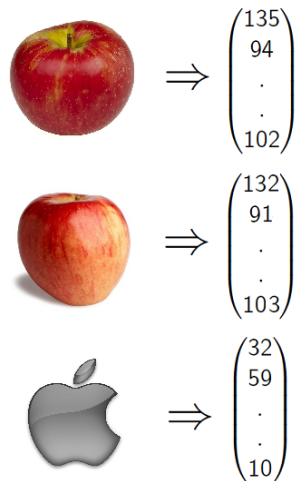
- a smaller (a shorter) form of an image, which encodes some important image characteristics
- this image form is used in image recognition tasks including
 - comparing images
 - finding similar images
 - distinguish images

Desired properties

- fast computation (real-time tasks)
- invariance to scale, rotation, and distortion changes

Basic idea for image descriptors

Feature extraction
(via image descriptors)



Similarity evaluation
(image classification)

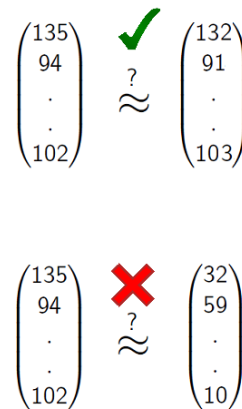


Image classification

Image classification

- includes a broad range of approaches to the identification of images.
- analyses the numerical properties of various image features and organizes data into categories – **image classes (clusters)**.
- compares the feature vectors using a chosen metric \Rightarrow close objects in feature space are considered visually similar and form clusters.

Image classes may be

- specified a priori by an analyst – **supervised classification**
- clustered automatically – **unsupervised classification**

Classification algorithms typically employ two phases

- training phase* – a unique description of each classification category (training class) is created
- testing phase* – feature-space partitions are used to classify image features

Image classification

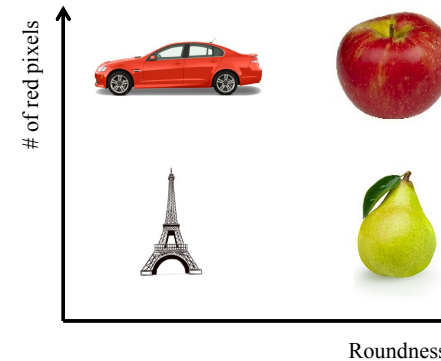
Most common classification methods

- *Cluster Analysis* – unsupervised method *k-means clustering*
- *Decision Trees* – non-parametric supervised method
- *Neural Networks* – statistical learning algorithms for supervised classification
- *Support Vector Machine (SVM)* – supervised classification, very popular
- *k-Nearest Neighbours algorithm (k-NN)* – simple, non-parametric, supervised method
- *Convolutional Neural Networks (CNN)* – learning based method

Image classification

Simple example: feature vector has 2 components

- ① Roundness – x-axis
- ② # of red pixels – y-axis



- What would be the feature vector of this query image?



Haralick features



- introduced in 1973 by Professor Haralick (see photo) from City University of New York
- popular approach for texture analysis
- Haralick features are still used in research
- based on so called *co-occurrence matrix*

Haralick features

Co-occurrence matrix

Co-occurrence matrix

- is the distribution of co-occurring values at a given offset
- mathematically, the co-occurrence matrix C is defined as

$$C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{if } I(p, q) = i \wedge I(p + \Delta x, q + \Delta y) = j \\ & \text{or } I(p, q) = i \wedge I(p - \Delta x, q - \Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$

- i and j are the image intensity values of the image
- p and q are the spatial positions in the $n \times m$ image I
- the offset $(\Delta x, \Delta y)$ depends on the used direction θ and the distance d at which the matrix is computed

Haralick features

Co-occurrence matrix

- $(\Delta x, \Delta y)$ represents the [separation vector](#)
- 4 orientations are usually considered
 - horizontal – separation vector $(1, 0)$ for distance 1
 - vertical – separation vector $(0, 1)$ for distance 1
 - main diagonal – separation vector $(1, 1)$ for distance 1
 - minor diagonal – separation vector $(1, -1)$ for distance 1

0	3	3
0	0	1
2	2	1

Original image I

#(0, 0)	#(0, 1)	#(0, 2)	#(0, 3)
#(1, 0)	#(1, 1)	#(1, 2)	#(1, 3)
#(2, 0)	#(2, 1)	#(2, 2)	#(2, 3)
#(3, 0)	#(3, 1)	#(3, 2)	#(3, 3)

General form of co-occurrence matrix for image I

Haralick features

Co-occurrence matrix

0	3	3
0	0	1
2	2	1

Original image I

$$C_{1,0} = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$$

$$C_{1,1} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$C_{0,1} = \begin{bmatrix} 2 & 0 & 2 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$C_{1,-1} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

Haralick features

Co-occurrence matrix

- because simple 8-bit images could have 256 intensity values, corresponding co-occurrence matrices will be very large
 - solution is to use [quantization](#) prior to the extraction process
- co-occurrence matrices are in the end normalized and averaged to form the final co-occurrence matrix C
- **Note:** All co-occurrence matrices are symmetric (why?)

Haralick features

[Haralick suggested 14 features](#) that could be derived from the matrix and form the feature vector of Haralick features

- entropy:
$$-\sum_{i=1}^q \sum_{j=1}^q C(i, j) \log C(i, j)$$
- texture correlation:
$$\sum_{i=1}^q \sum_{j=1}^q |i - j| C(i, j)$$
- texture homogeneity:
$$\sum_{i=1}^q \sum_{j=1}^q \frac{C(i, j)}{1 + |i - j|}$$
- and the others ... (q is the maximal intensity present in the image)

Haralick features

Bibliography

- R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Trans. on Systems, Man and Cyber.*, SMC-3(6):610–621, 1973.
- L. Tesař, D. Smutek, A. Shimizu, and H. Kobatake. 3D Extension of Haralick Texture Features for Medical Image Analysis. In *Proceedings of the Fourth IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, SPPRA '07, pages 350–355. ACTA Press, 2007.

Local binary patterns (LBP)



- introduced in 1994 by Ojala (upper photo) and Pietikäinen (lower photo) from University of Oulu, Finland
- descriptor became famous after generalization in 2002
- originally proposed for face recognition
- currently used also in (bio)medical image analysis, motion analysis, eye localization, fingerprint recognition, and many others

Local binary patterns (LBP)

Original approach (1994)

Idea: Texture can be described by the **pattern** and its **strength**

LBP pattern

- 1 each pixel is compared with its 8 neighbours
- 2 if the intensity value of neighbouring pixel is greater than or equal to the value of examined pixel's intensity, write 1 (otherwise, write 0)

Example			Threshold		
7	1	9	1	0	1
2	5	5	0	1	1
6	0	3	1	0	0

Local binary patterns (LBP)

Original approach (1994)

Idea: Texture can be described by the **pattern** and its **strength**

LBP pattern

- 3 take the digits from top-left corner in clockwise order and interpret them as decimal number
- 4 this decimal number represents the pattern

Example			Threshold		
7	1	9	1	0	1
2	5	5	0	1	1
6	0	3	1	0	0

Binary: 10110010
LBP = 178

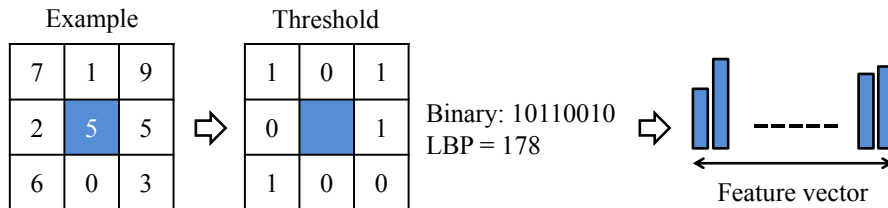
Local binary patterns (LBP)

Original approach (1994)

Idea: Texture can be described by the **pattern** and its **strength**

Strength of the pattern

- 5 decimals from entire image are used to form histogram (256 bins – why?)
- 6 concatenation of the normalized histogram values gives us the feature vector

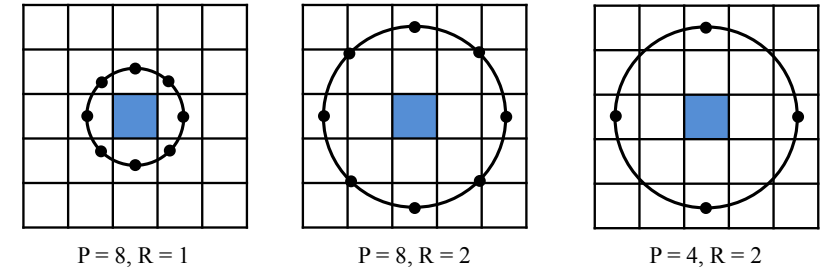


Local binary patterns (LBP)

Generalization of LBP (2002)

Idea: No limitation to the size of the neighbourhood and the number of sampling points

- parameter P - number of sampling points
- parameter R - size of the neighbourhood

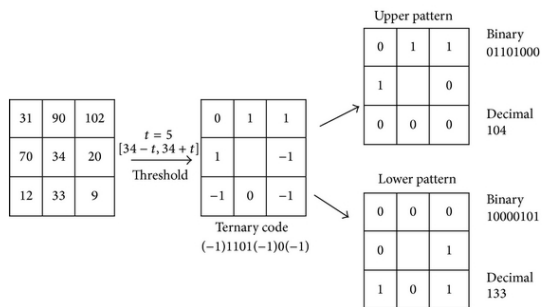


- when the sampling point is not in the centre of the pixel, **bilinear interpolation** is used

Local binary patterns (LBP)

LBP descriptor has many variants and modifications

- *Median binary patterns* – thresholding against the median within the neighbourhood
- *Local ternary patterns* – solving problem of nearly constant areas



- and the others ...

Local binary patterns (LBP)

Bibliography

- T. Ojala, M. Pietikäinen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *12th IAPR Intern. Conf. on Patt. Recog. Vol. 1 - Conf. A: Computer Vision and Image Processing*, pages 582–585, Oct. 1994.
- T. Ojala, M. Pietikäinen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.
- M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Computational imaging and vision. Springer Verlag, London, 2011.

MPEG in general

- **Motion Picture Experts Group (MPEG)** – developed digital audiovisual compression standards (in 1988)
- **MPEG-1** (1993) – the first standard for audio and video **MP3**
- **MPEG-2** (1995) – generic coding of moving pictures and associated audio information
- **MPEG-4** (1998) – coding of audio-visual objects
- **MPEG-7** (2002) – multimedia content description interface (including Visual descriptors)

MPEG-7 descriptors



- part of MPEG-7 visual standard
- standardized low-level descriptors for different domains
- many contributors, joining editor B. S. Manjunath (see photo)
- first public release in 2002

MPEG-7 descriptors

Division

MPEG-7 visual descriptor are divided to 4 groups

- **Colour descriptors** – robust to viewing angle, translation, and rotation of the regions of interest (ROI), 6 features are included here
- **Texture descriptors** – contain important structural information of intensity variations and their relationship to the surrounding environment, 3 features are included here
- **Shape descriptors** – techniques for describing and matching shape features of 2D and 3D, 3 features are included here
- **Motion descriptors** – description of motion features in video sequences, 4 features are included here

MPEG-7 descriptors

Texture descriptors

MPEG-7 texture descriptors consist of three feature extractors

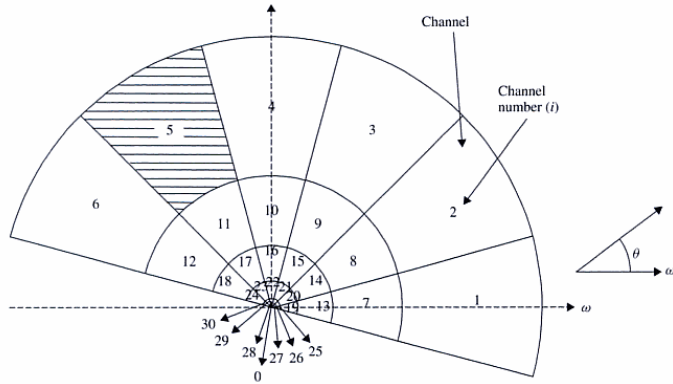
- **Homogeneous Texture Descriptor (HTD)** – characterizes the region texture using the mean energy and the energy deviation from the set of frequency channels
- **Texture Browsing Descriptor (TBD)** – specifies the perceptual characterization of the texture, which is similar to human perception
- **Edge Histogram Descriptor (EHD)** – spatial distribution of edges in the image

Notice: We will briefly describe **HTD** and **EHD**.

MPEG-7 descriptors

Homogeneous Texture Descriptor (HTD)

2D frequency plane is partitioned into 30 channels



- partitioning uniform along the angular direction and not uniform along the radial direction (in octave scale)

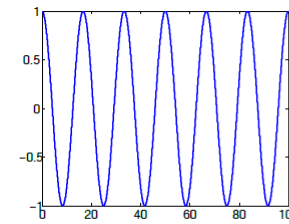
MPEG-7 descriptors

Homogeneous Texture Descriptor (HTD) – Gabor filters

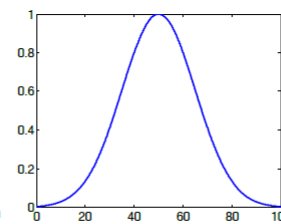
The individual channels are convolved using [Gabor filters](#)



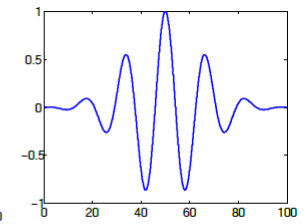
- introduced in 1946 by Dennis Gabor (see photo) for 1D signal
- the filter is obtained by modulating a sinusoid with a Gaussian function
- it responds to some frequency in a localized part of the signal



(a) Sinusoid



(b) Gaussian function

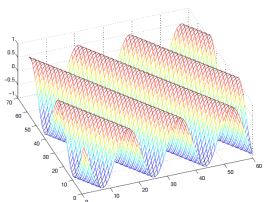


(c) Gabor filter

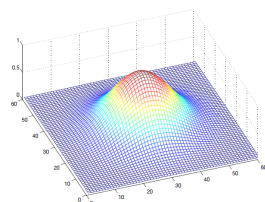
MPEG-7 descriptors

Homogeneous Texture Descriptor (HTD) – Gabor filters

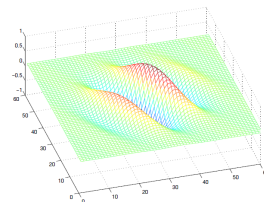
Extension of [Gabor filters](#) to 2D



(a)



(b)



(c)

MPEG-7 descriptors

Homogeneous Texture Descriptor (HTD) – Gabor filters

The (s, r) -th channel, where s is [radial index](#) and r is [angular index](#), is modelled in frequency domain as

$$G_{s,r}(\omega, \theta) = \exp \left[\frac{-(\omega - \omega_s)^2}{2\sigma_s^2} \right] \cdot \exp \left[\frac{-(\theta - \theta_r)^2}{2\tau_r^2} \right]$$

- σ_s and τ_r are standard deviation of the Gaussian in radial and angular direction, respectively
- $\theta_r = 30^\circ \times r$, where $r \in \{0, 1, 2, 3, 4, 5\}$
- $\omega_s = \omega_0 \times 2^{-s}$, where $s \in \{0, 1, 2, 3, 4\}$ and ω_0 is the highest frequency

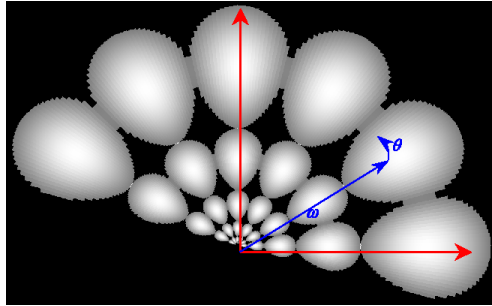
MPEG-7 descriptors

Homogeneous Texture Descriptor (HTD)

The syntax of the **HTD** is as follows:

$$\text{HTD} = [f_{DC}, f_{SD}, e_1, e_2, \dots, e_{30}, d_1, d_2, \dots, d_{30}]$$

- f_{DC} is the mean of the image
- f_{SD} is the standard deviation of the image
- e_i and d_i are non-linearly scaled and quantized mean and standard deviation of the i^{th} channel ($i \in \{1, 2, \dots, 30\}$)

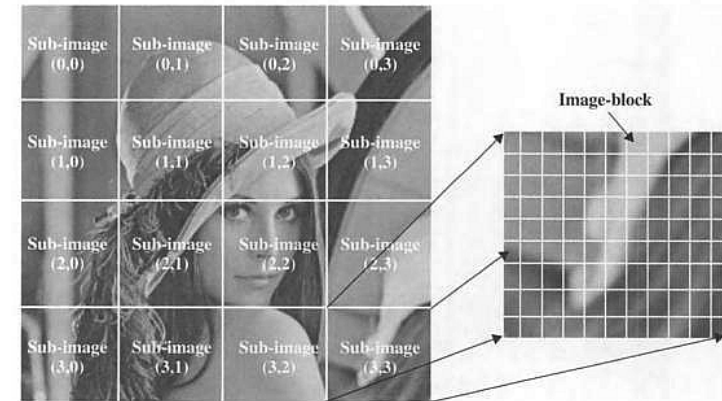


MPEG-7 descriptors

Edge Histogram Descriptor (EHD)

EHD represents the local edge distribution in the image

- divide image space in 4×4 sub-images
- each sub-image divided into non-overlapping squared image blocks (1100 image blocks)

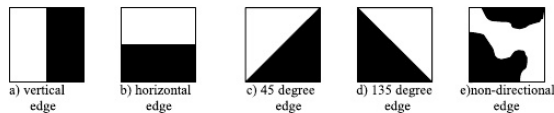


MPEG-7 descriptors

Edge Histogram Descriptor (EHD)

EHD represents the local edge distribution in the image

- each image block is classified into one of the 5 edge categories or as non-edge block



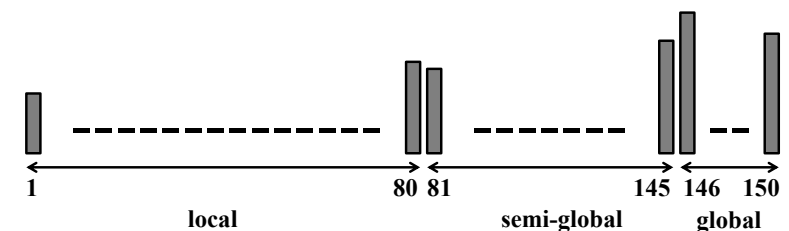
- classification is done by applying corresponding edge detector and thresholding

MPEG-7 descriptors

Edge Histogram Descriptor (EHD)

Feature vector of **EHD** consists of three types of bins

- **local** – 4×4 sub-images \times 5 types of edges
- **semi-global** – grouping of sub-images in predefined way (horizontal, vertical, ...)
- **global** – 1 bin for every type of edges



MPEG-7 descriptors

Bibliography

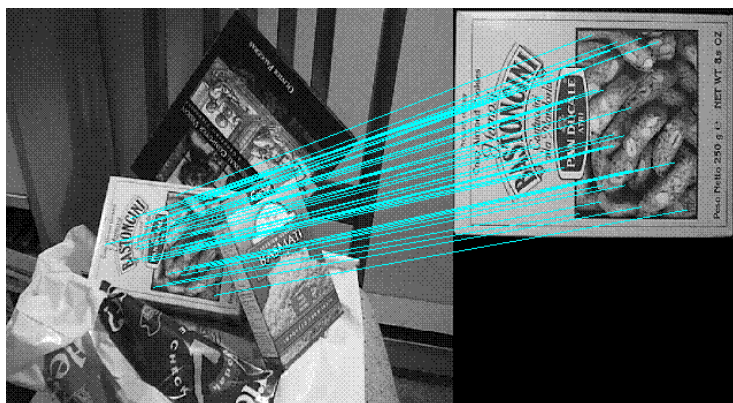
- B. S. Manjunath, P. Salembier, and T. Sikora, editors. Introduction to MPEG-7: Multimedia Content Description Interface. *Wiley & Sons, Inc.*, New York, USA, Apr. 2002.

Scale-invariant feature transform (SIFT)



- presented in 2004 (first article in 1999) by David Lowe (see photo) from University of British Columbia (UCB), Canada
- patented by UCB for commercial purposes
- local feature extraction (robust to occlusion)
- similar to human visual system
- extracting distinctive invariant features

Scale-invariant feature transform (SIFT)



- demonstration of SIFT descriptor
- finding corresponding parts of the image
- query image (in the right) is identified as a part of the image in the left

Scale-invariant feature transform (SIFT)

SIFT consists of **key point detection** and **key point descriptor**

Key point detection

- location of the peaks in scale space
- key point localization
- orientation assignment

Key point descriptor

- describing the key point as a vector
- could be used with other key point detections

Scale-invariant feature transform (SIFT)

Key point detection

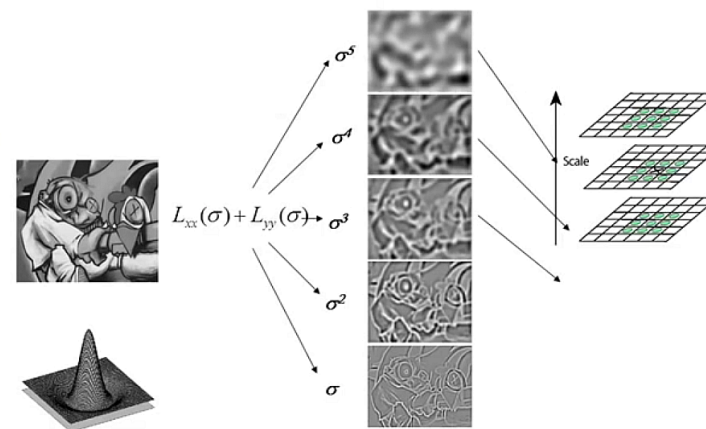
Key points are derived as local extreme point in scale space of Laplacian-of-Gaussian (LoG)

- derive LoG with various σ values
- for each point, compare it in $3 \times 3 \times 3$ neighbourhood (3D image from the scale spaces)
- if central point is an extreme point (maxima or minima), consider it as a **key point**

Scale-invariant feature transform (SIFT)

Key point detection

Key points are derived as local extreme points in scale space of Laplacian-of-Gaussian (LoG)

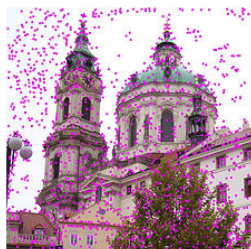


Scale-invariant feature transform (SIFT)

Key point detection

Key point localization consists of

- eliminating outliers (poorly localized along the edges)
- searching for best scales for all extreme points
- comparing to some threshold

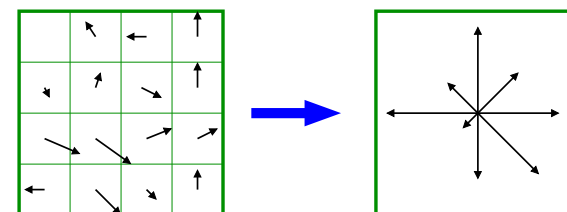


Scale-invariant feature transform (SIFT)

Key point detection

Orientation assignment to key points

- to achieve rotation invariance
- at each point compute central difference (magnitude and direction)
- for each key point, build the weighted histogram of directions (36 bins \implies per 10°), weights are gradient magnitudes
- select the peak as the direction of the key point (could be more, within 80% of max peak)
- any further calculations are done relative to this orientation

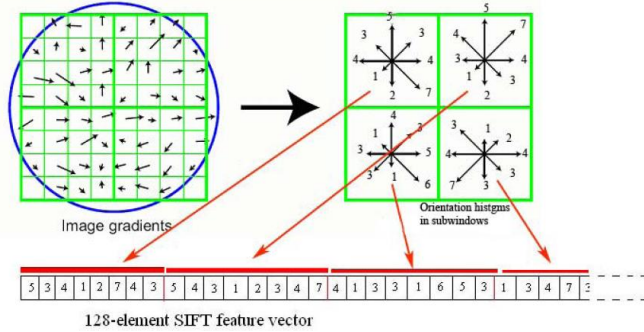


Scale-invariant feature transform (SIFT)

Key point descriptor

Extracting of **local image descriptors** at key points

- compute **relative orientation!** and magnitude in 16×16 (depicted only 8×8) neighbourhood at key point
- form weighted histogram (8 bins) for 4×4 regions
- concatenate 16 histograms in one vector of 128 dimensions which represents the **SIFT feature vector**



Scale-invariant feature transform (SIFT)

Bibliography

- D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.
- Lecture on YouTube [▶ Link](#)



Zernike Features

Zernike polynomials in 2D

$$V_{nl}(x, y) = \sum_{m=0}^{\frac{n-l}{2}} (-1)^m \frac{(n-m)!}{m! \left(\frac{n-2m+l}{2}\right)! \left(\frac{n-2m-l}{2}\right)!} (x^2 + y^2)^{\frac{n-m}{2}} e^{im\theta},$$

where

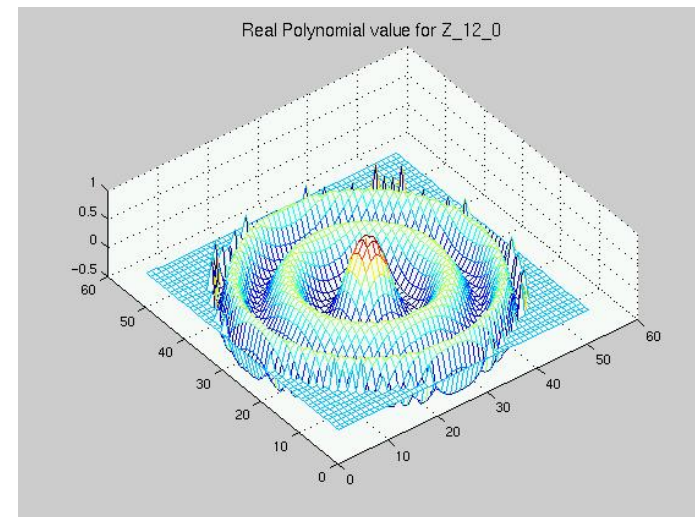
- $0 \leq l \leq n$
- $(n - l)$ is even
- $\theta = \tan^{-1} \left(\frac{y}{x} \right)$
- $x^2 + y^2 \leq 1$
- individual V_{nl} are orthogonal.



Frederik Zernike (1888-1966)

Zernike Features

Zernike polynomials in 2D – Examples



Zernike Features

Definition

Let be given an inner product

$$Z_{nl} = \frac{n+1}{\pi} \sum_x \sum_y V_{nl}^*(x, y) f(x, y),$$

where

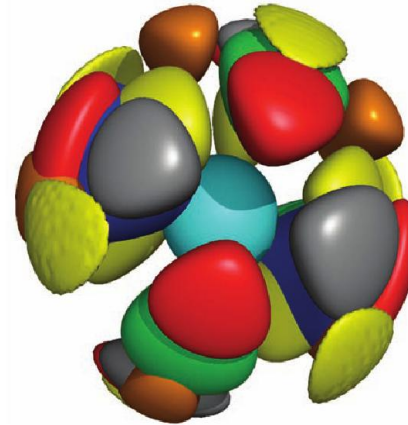
- $f(x, y)$ is an analyzed image a
- V_{nl} is a selected Zernike polynomial.

Then scalar $|Z_{nl}|$ is called a **Zernike feature/descriptor**.

Notice: $Z_{nl} \in \mathbb{C}$

Zernike Features in 3D

3D Zernike polynomial



- Novotni, M., Klein, R. Shape retrieval using 3D Zernike descriptors, Computer-Aided Design, Volume 36, Issue 11, Solid Modeling Theory and Applications, r 2004, 1047–1062
- Grandison, S., Roberts, C., Morris, R. J. The Application of 3D Zernike Moments for the Description of Model-Free Molecular Structure, Functional Motion, and Structural Reliability, Journal of Computational Biology. March 2009, 16(3): 487-500

Moment Invariants

Definition

- The 2-D moment of order $(p + q)$ of a digital image $f(k, l)$ of size $M \times N$ is defined as:

$$m_{pq} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} k^p l^q f(k, l)$$

where $p = 0, 1, 2, \dots$ and $q = 0, 1, 2, \dots$ are integers.

- The central moment of order $(p + q)$ is defined as

$$\mu_{pq} = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} (k - \bar{k})^p (l - \bar{l})^q f(k, l)$$

where

$$\bar{k} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{l} = \frac{m_{01}}{m_{00}}$$

Moment Invariants

Definition (cont.)

- The normalized central moments are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^c}$$

where

$$c = \frac{p+q}{2} + 1 \quad \text{for} \quad p+q = 2, 3, \dots$$

Now, let us define several moment invariants that are insensitive to

- translation
- scale
- change
- mirroring
- rotation

Moment Invariants

Seven invariants

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

You should know the answers . . .

- Build your own 10B descriptor for any grayscale image. Explain the meaning of individual parts of the feature vector.
- Explain the way of efficient comparison of two randomly chosen RGB color images.
- Describe the construction of so called *co-occurrence matrix*. How would you observe large scale (spanned over more than 3 pixels) texture details?
- Why do LBP feature vectors possess histograms with 256 bins?
- Which way may we compute the mean gradient direction of a selected 4×4 region?
- Propose an extension of standard Haralick features to work with 3D image data.
- How would you apply Zernike polynomial to an incoming image of any size so that you could compute the corresponding Zernike feature?