Identification Systems

Lecture 8

Piotr Syga

2021-05-04

Image representation

- We will interpret an image as a function $f(x, y) = i(x, y)r(x, y)$, with
  - $0 < i(\cdot, \cdot) < \infty$, depicting the intensity of the light
  - $0 < r(\cdot, \cdot) < 1$, depicting the reflectiveness of an object

Image representation

- We are assuming a discrete space of a fixed size $N \times M$, hence we work on a matrix of elements called pixels:

$$F(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, M-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1, 0) & f(N-1, 1) & \dots & f(N-1, M-1) \end{bmatrix}$$

- $f(x, y) \in \{0, \dots, L-1\}^c$ $(c \in \mathbb{N}^+)$, with $L = 2^K$, $k \in \mathbb{N}^+$ and $c \in \mathbb{N}^+$

Image representation

- Basic mapping process may be performed as follows:

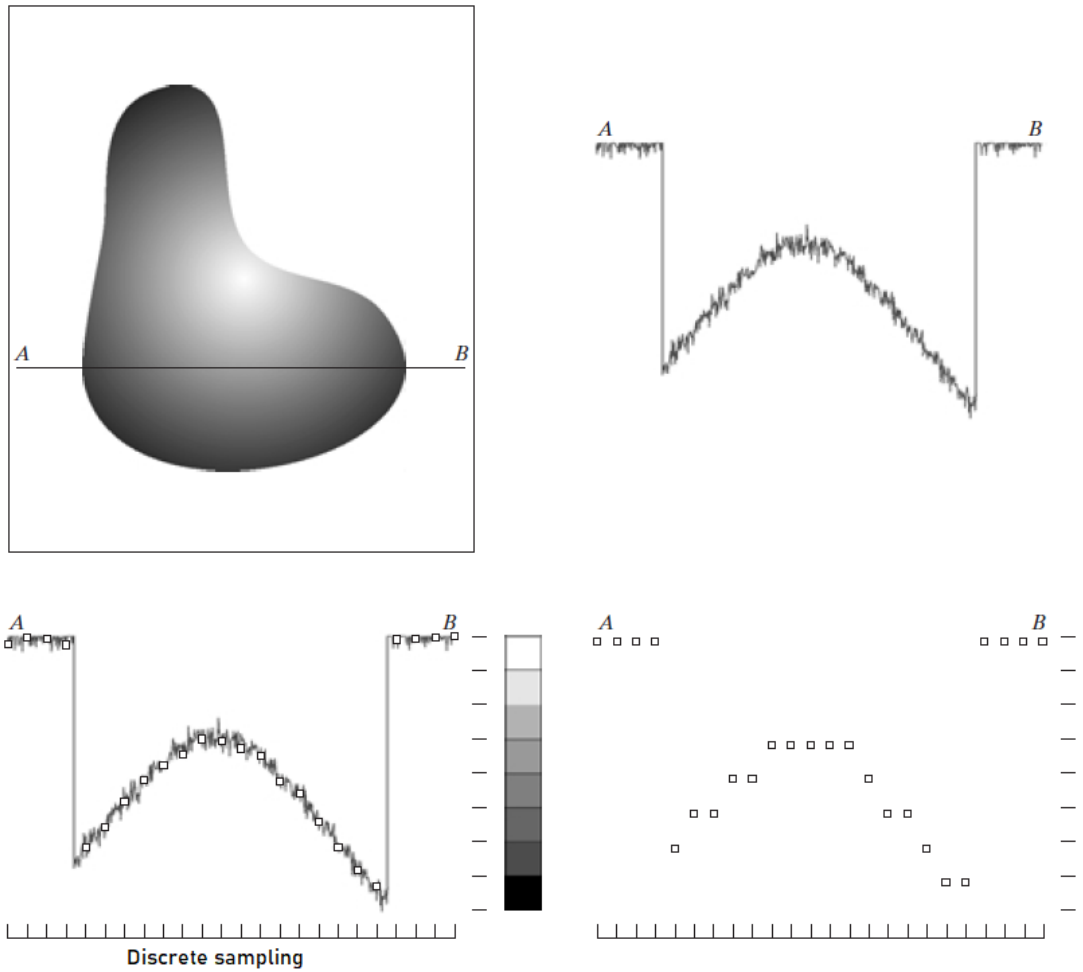


Image analysis – basic definitions

Neighborhood

For each pixel p at (x, y) we can determine a set called neighborhood:

- 4-neighborhood
  $N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$,
- diagonal neighborhood $N_D(p) =$
  $\{(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)\}$,
- 8-neighborhood $N_8(p) = N_4(p) \cup N_D(p)$.

For pixels at the edges we may either skip a value or use padding for a constant size neighborhoods.

Image analysis – basic definitions

Adjacent pixels

We call two pixels p_0, p_1 x-adjacent, if $p_1 \in N_x(p_0)$.

Path

A path (w.r.t. N_x) from a pixel p_0 to p_n is a sequence $p_0, p_1, \ldots, p_n$,such that each pair p_i, p_{i+1} is adjacent.

Connection

Let V be a set of values. We say that two pixels p_0, p_n with intensities from V are connected if there is a path from p_0 to p_n and all the pixels on the path have intensities from V.

Image analysis – basic definitions

Connected component

For a fixed pixel p a set of connected pixels $C(p)$ is called a connected component.

Contour

Let C be a connected component, a set $B(C) = \left\{p \in C : \exists_{q \in N_8(p)} q \notin C\right\}$ is called a contour.
Note: For a neighborhood with skipped pixels, the border pixels do belong to the contour.

Convolution and filters

Convolution

A convolution of two matrices A and B is calculated as

$$C(j,\, k) = A * B = \sum_p \sum_q A(p,\, q) B(j - p, k - q)\ .$$

For two signals $x,\, y \in \mathbb{C}^n$ there holds $x * y = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{x\right\}\mathcal{F}\left\{y\right\}\right\}$, where $\mathcal{F}$ denotes Fourier transform of a signal.

Convolution is widely used in image processing useful filters include:

- boxfiltering

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

- Gauss filter

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

- edge detection
  - vertical (Prewitt)

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

  - vertical with augmentation (Sobel)

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

  - horizontal

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

  - horizontal with augmentation

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

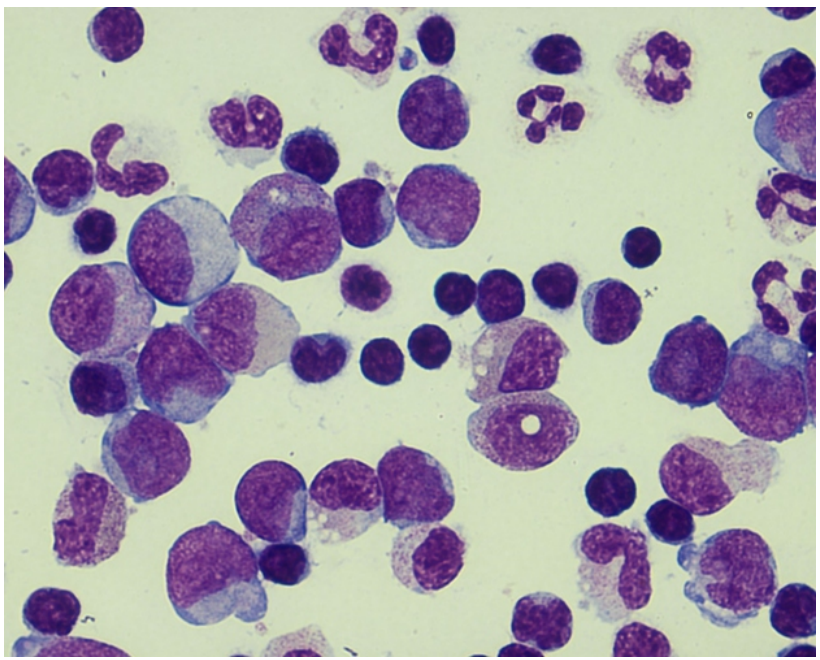
  - image sharpening

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Image segmentation

- A sequence of operations that allows us to find a mask/contour of given objects in the picture is called segmentation.
- The methods depend on prior knowledge, type and number of the objects, the number of channels, required accuracy and time efficiency and the quality of the image.
- Main uses include finding anomalies, detection of specific shapes and objects, motion detection and tracking, autonomous driving, object classification or user identification.

Image segmentation



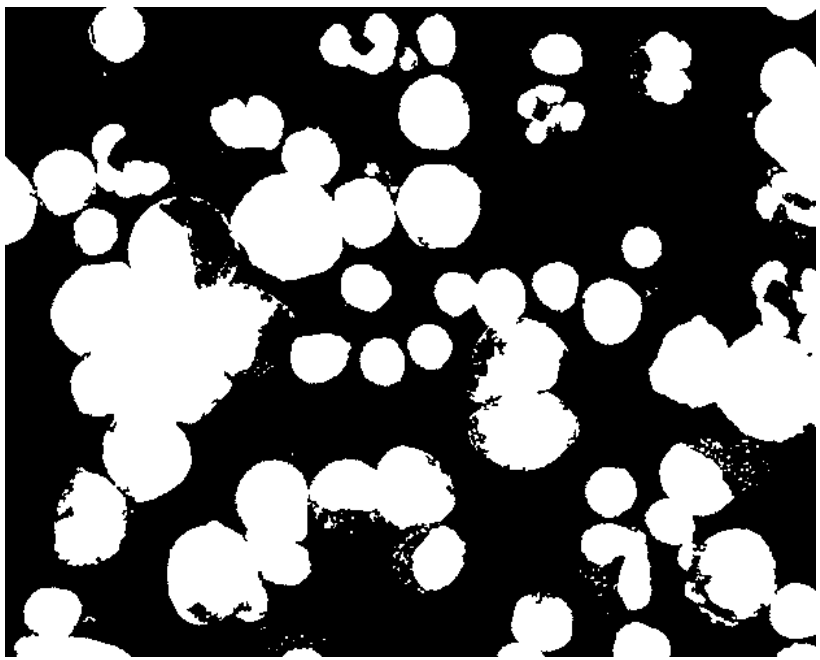
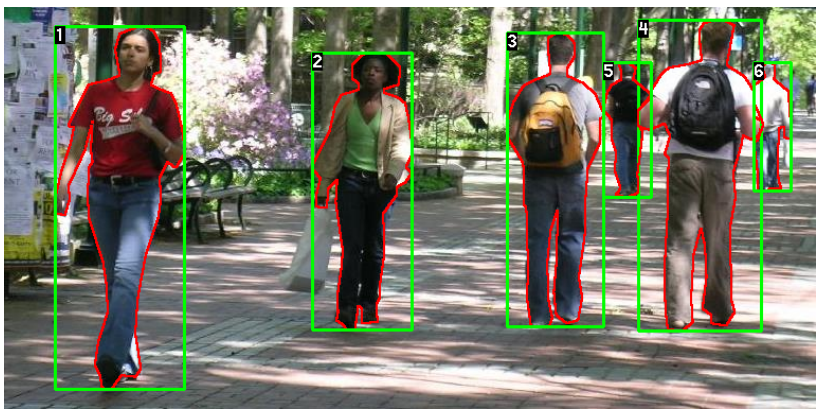


Image segmentation



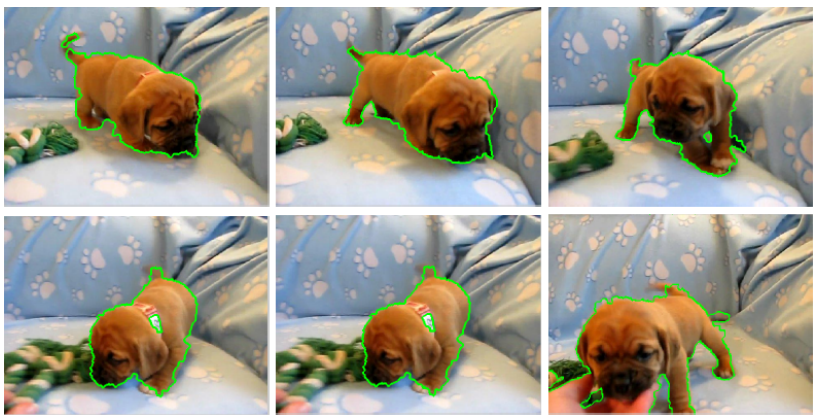


Image segmentation

- Simplest examples include threshold binarization, Otsu (multi)thresholding, morphological contouring and edge detection (e.g. Canny edges).
- Region growing: Do{
  $S' \leftarrow S$
  $\forall_{x \in N(S')} \text{if} \frac{1}{\delta} \text{mean}(S') < x < \delta \text{mean}(S') : \ S' \leftarrow S' \cup \{x\};$
  }While$(\|S - S'\| > \Delta)$,
- Circle Hough Transform
- GLCM

$$M_{\Delta x, \Delta y}(k, o) = \sum_{x,y} \begin{cases} 1 & I(x,y) = k \wedge I(x + \Delta x, y + \Delta y) = o \\ 0 & \text{oth.} \end{cases}$$

  - contrast $\sum_{k,o} |k - o|^2 p(k, o)$
  - correlation $\sum_{k,o} \frac{(k - \mu k)(o - \mu o) p(k,o)}{\sigma_k \sigma_o}$
  - energy $\sum_{k,o} p(k, o)^2$
  - homogeneity $\sum_{k,o} \frac{p(k,o)}{1 + |k - o|}$

Image segmentation

- Gabor wavelets – a series of wavelets that minimize the variance in both frequency and space domains (they do not have to be orthogonal)

$$f(x) = e^{-\frac{(x-x_0)^2}{a^2}} e^{-iA(x-x_0)}$$

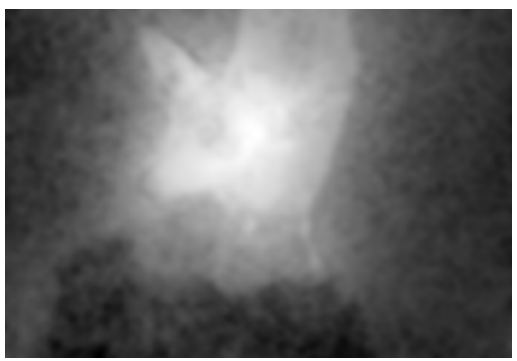




Image segmentation

- Active contour
  - Extension of 1D signal analysis – for a set of samples we find 'sufficiently good estimation' with a curve with a fixed properties (e.g., smoothness, curvature).
  - We define smoothness by balancing internal and external energies

    $E = aE_1 + bE_2$, s.t. $a + b = 1$ for $E_1 = \int \left(\frac{dM(x)}{dx} \right)^2 dx$,

    $E_2 = \int \left(\frac{d^2 M(x)}{d^2 x} \right)^2 dx$
  - We minimize: $\mathrm{dist} + \lambda E$, where dist is usually L_2 norm
  - We can interpret dist as a sum of outer energies

    $E_{\mathrm{dist}} = \lambda_{\mathrm{torsion}} e_{\mathrm{torsion}} + \lambda_{\mathrm{edge}} e_{\mathrm{edge}} + \lambda_{\mathrm{length}} e_{\mathrm{length}}$
  - For efficiency we relax the model so that inner energy includes only 1 order, and outer energy is reduced to the edge factor only

    $E_{\mathrm{dist}} = \lambda_k e_k = -\lambda_k \int \left\| (G_\sigma * \nabla^2 I) M(s) \right\|^2 ds$

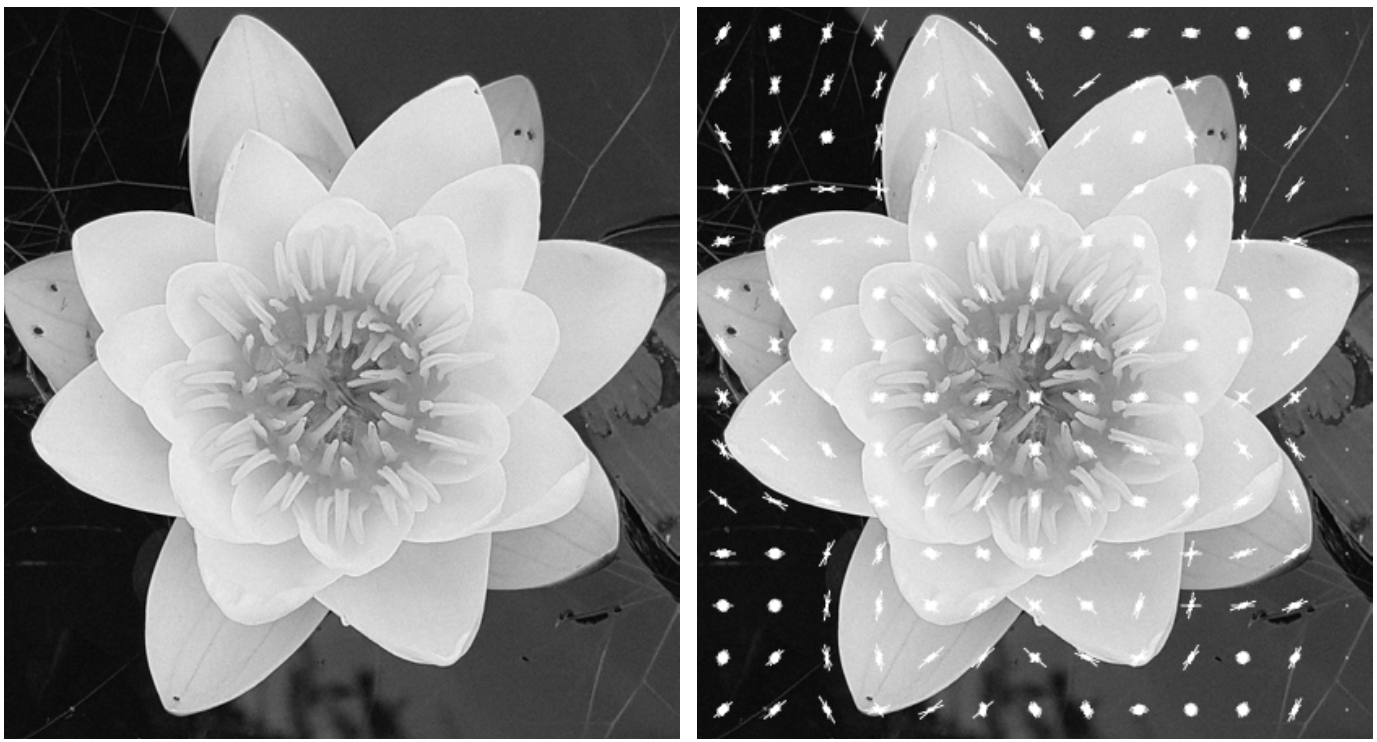
Image segmentation

- Active contour cont.
  - We obtain optimization problem $\min_{f;\alpha,\lambda}\left(\alpha\int\left\|\dot{f}(s)\right\|^2 ds-\lambda\int\|\nabla I(f(s))\|^2 ds\right)$ that is solved using dynamic programming.
  - Since the gradient is maximized by the edges the evolution of the curve stops, yet there are Active contours for images with low local contrast, e.g., T.F. Chan, L.A. Vese: *Active Contours Without Edges*
    We use :$F(C)=F_1(C)+F_2(C)=$
    $\int \text{inner}C|I(x,y)-c_1|^2\,dxdy+\int \text{outer}C|I(x,y)-c_2|^2\,dxdy$, for constats c_1, c_2 that are mean intensities on both sides of the curve$C$.
    We minimize $\inf_C m\cdot\text{len}(C)+F(C)$

Histogram of oriented gradients

- A simple and quick local descriptor for shape estimation.
- We calculate vertical and horizontal gradients $\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$, $\begin{bmatrix}-1\\0\\1\end{bmatrix}$ – obtaining matrices describing gradient magnitude and direction.
- We divide the image into patches (e.g., 8×8) and create a histogram of magnitudes into bins depicting the angle (Note: we often use the same bin for α and $\alpha + 180$).
- If the angle is not equal to the limit value of the bin, we divide the magnitude proportionally into 2 bins.
- We normalize the histogram (by concatenation) over a block of 2×2 patches.

Histogram of oriented gradients



Retinex

J. Hérault, B. Durette: *Modeling Visual Perception for Image Processing*, J.-M. Morel, A. B. Petro, C. Sbert: *A PDE formalization of retinex theory*

- Uneven lighting may hinder every segmentation method.
- Simple solution like histogram equalization (or CLAHE) may not be sufficient, hence we imitate the image processing done by human eye.
- We represent each image in a dual way – *parvocellular* for static details, *magnocellular* for dynamism (motion, contrast, energy changes, lighting changes) so that we can estimate the reflection function $r(x, y)$.

Retinex

- We start by a random walk from N pixels. Each source starts J paths of length K_j.
- We calculate

$$r(p) = \frac{1}{J} \sum_{j=1}^{J} \sum_{k=1}^{K_j} \lambda(I(p_k^{(j)}) - I(p_{k+1}^{(j)})) \ ,$$

  for $\lambda = \mathbb{1}_{|x|>\tau}|x|$ and a fixed threshold τ.
- We solve a differential equation

$$\begin{cases} \frac{\partial r}{\partial N} = 0 & \text{for border pixels } p \\ -\Delta r(p) = F(p) & \text{oth.} \end{cases}$$

  $F(p) = \lambda(I(p) - I(p_{(+1,0)})) + \lambda(I(p) - I(p_{(-1,0)})) + \lambda(I(p) - I(p_{(0,+1)})) + \lambda(I(p) - I(p_{(0,-1)}))$.
- For efficiency reasons we transform into frequency domain

$$I = \mathcal{F}^{-1}\left\{ \frac{\mathcal{F}\{F(p)\}}{2\left(\cos\left(2\pi\frac{r}{H}\right)+\cos\left(2\pi\frac{s}{W}\right)-2\right)} \right\}$$

Retinex

Further processing:

High-pass filtering – to limit $\frac{1}{f}$ spectra

Non-linear frequency filtering – simulation of a reverse-pyramid representation – processing from coarse to detail

Multiplexing – color shifting (by wavelength multiplication)

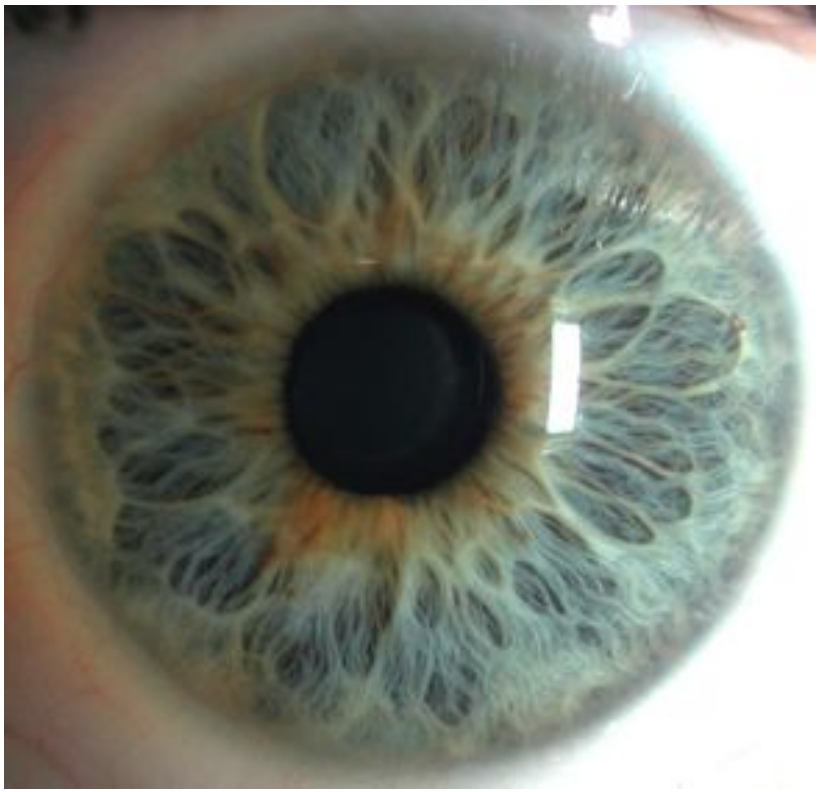
Nonuniform sampling – antialiasing

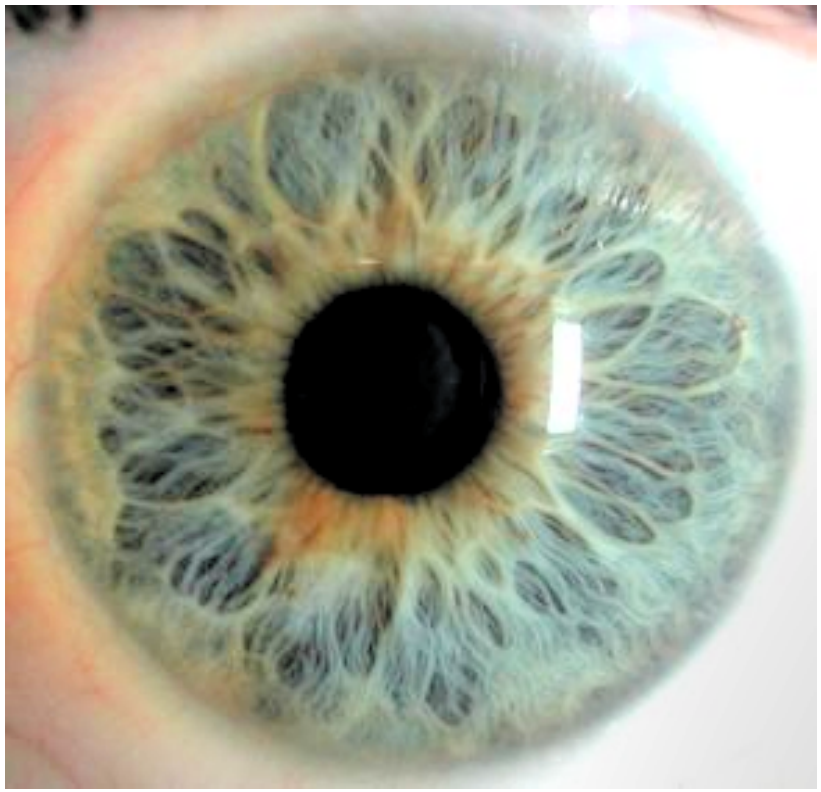
Positional sampling – simulates the localization of human receptors – contrast localization, gradient estimation, zooming

Local histogram equalization – simulates fotoreceptor's compression

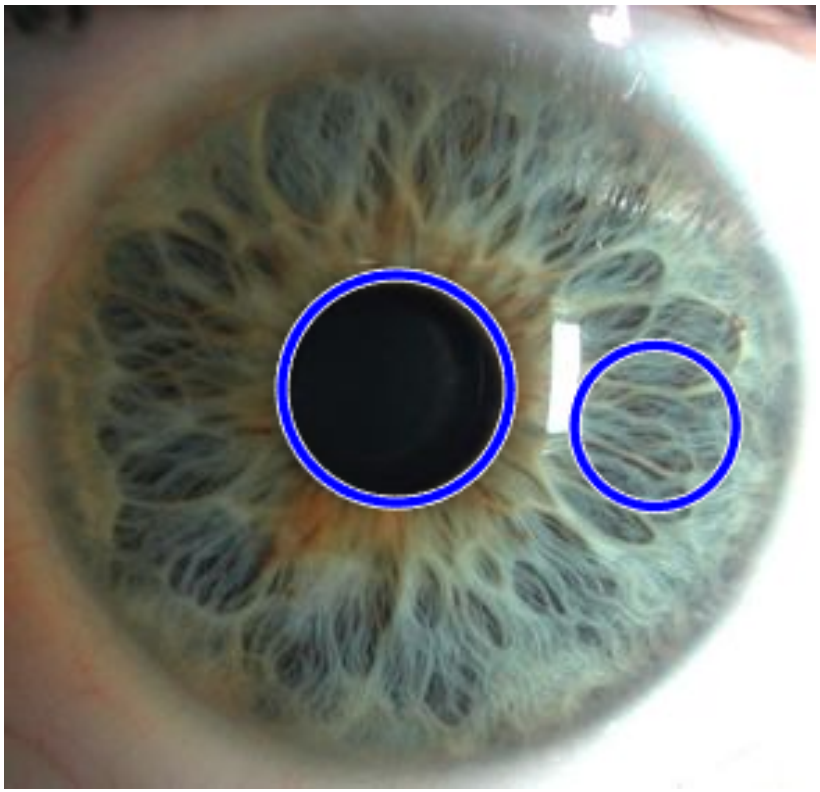
Local contrast equalization – simulates ganglion's compression

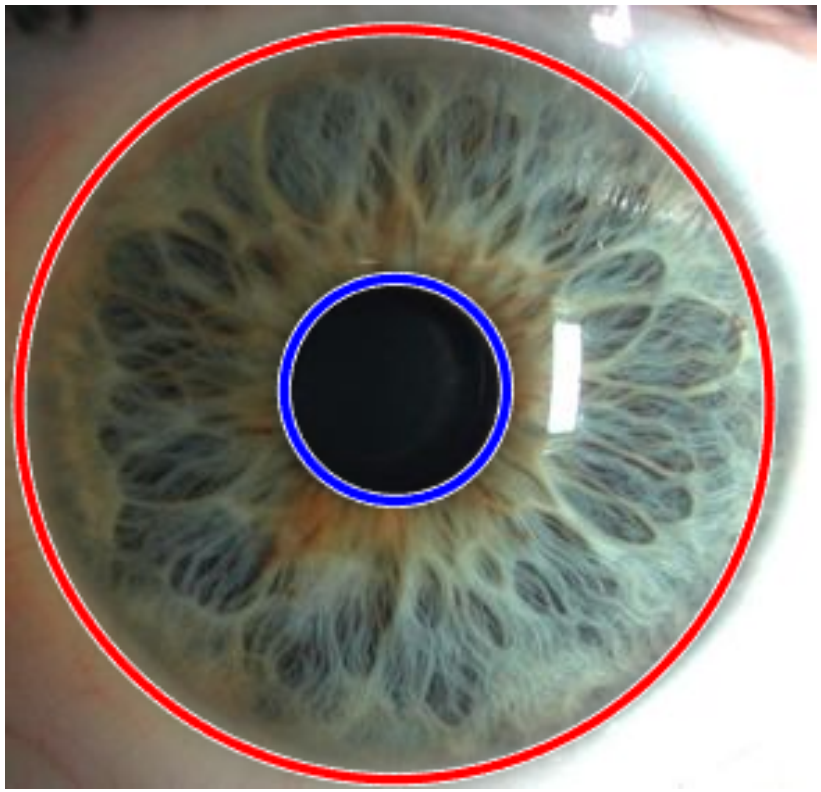
Retinex



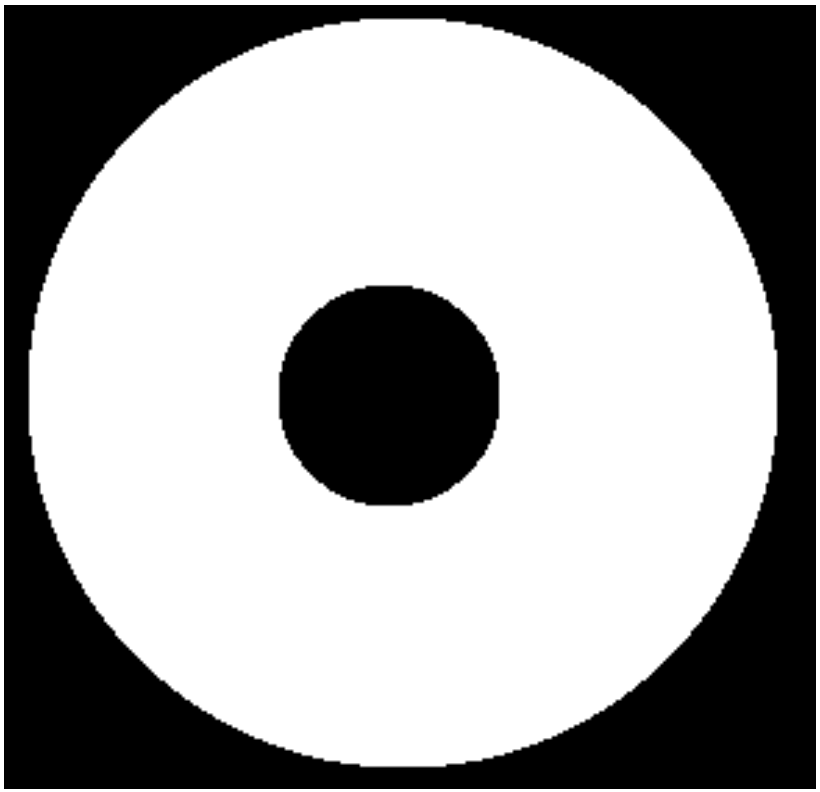


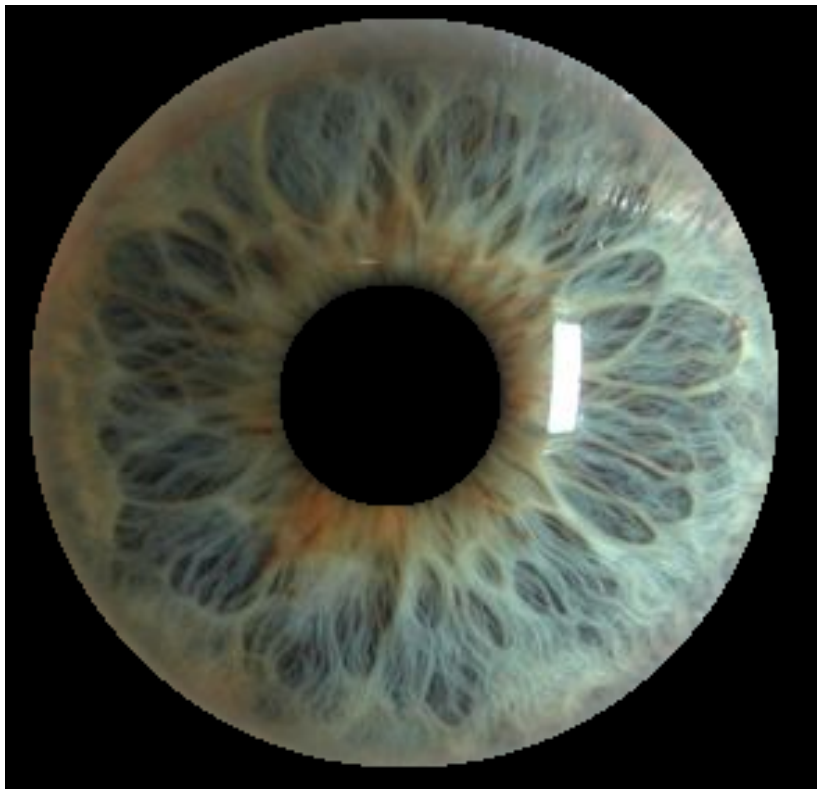
Retinex





Retinex





Motivation and problems

- Face is most common biometric mode used in person to person identification.
- Face may be occluded by facial hair, accesories related to fashion, beliefs or medical condition.
- Two person may look similar (or identical), even if they are unrelated.
- How should we represent a face as a user feature vector?
- Should we keep 2D or 3D profiles, what with tilts and rotation in each case?
- How to cope with aging or artificial face modifications (permanent or not)?
- How to keep user privacy intact? How to prevent unintended identification? How do we keep our solution robust against presentation attacks?

Main steps

Both classic algorithms (like Eigenface and Fisherface) and Neural Network-based use several common steps:

- Image preprocessing – quality enhancement
- Face detection
- Face alignment and normalization
- Feature extraction
- Classification – user identification

Requirements

- The method should be efficient – facial image may cover relatively small part of high resolution picture.
- The method should be universal.
- We aim for robustness against scale, uneven lighting, occlusions, noisy image, compression, blurring and environmental conditions.
- We want to minimize the influence of face alignment.

Main types: sliding window or region-based localization.

Haar wavelets and Haar-like features

Haar wavelets

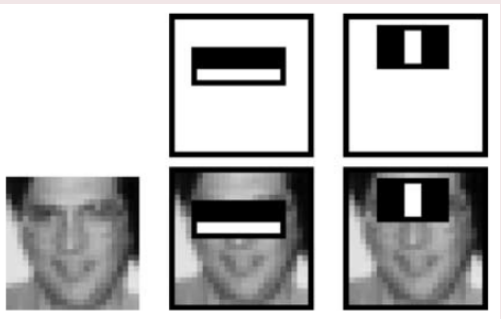
For every pair n, $k \in \mathbb{Z}$, the Haar function $\psi_{n,k}$ is defined on $\mathbb{R}$ as $\psi_{n,k}(t) = 2^{n/2}\psi(2^n t - k)$, for $t \in \mathbb{R}$ and

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & \text{oth.} \end{cases}$$

Haar wavelets and Haar-like features

Haar-like features (P. Viola, M.J. Jones: Robust Real-Time Face Detection)

Based on Haar-wavelets we can create a series of weak classifiers that can be calculated in constant time. We create rectangle filters that may be efficiently applied to an image using fast convolution. A group of such weak classifiers allows us to create a strong classifier. A cascade of strong classifiers is used.
Example of filters interposed on images:



Haar wavelets and Haar-like features

A single strong classifier $h(x)$ is composed as $h(x) = \text{sign} \sum_j w_j h_j(x)$, where w_j are weak classifier weights,

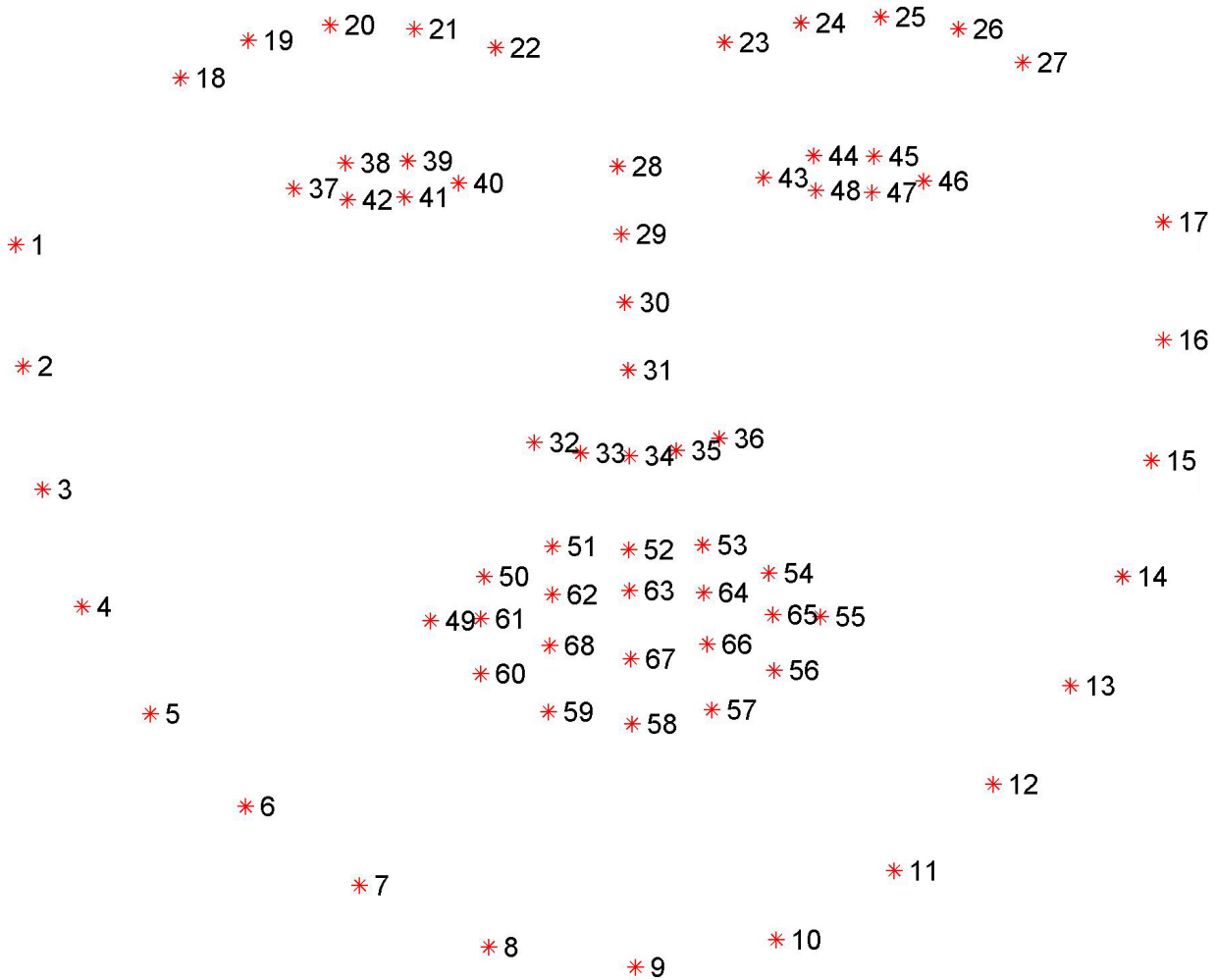
$$h_j(x) = \begin{cases} -1 & p_j f_j < p_j T_j \\ 1 & \text{oth.} \end{cases}$$

with polarity $p_j \in \{-1, 1\}$, f_j – an output of rectangular filter and T_j – threshold acceptance level.
The parameters w, p, f and T are learned using AdaBoost in the training stage.
Classification is made based on a cascade of length N that eliminates false positive matches – we assume that we start with a face and abort if a strong classifier determines otherwise.

Other methods

- A simple method for face localization utilizes HoG and a classifier – a linear SVM or CNN, e.g. using dlib.
- Face detection using colors and PCA.
- Face detection using HoG-like edge orientation.
- You only look once detection. (YOLO paper)

Landmarks – recalled



Facial alignment

- The face orientation both in cases of rotation and tilt may affect face detection and identification.
- A proper localization of the landmarks may help with both identification and face realignment.
- Decision whether to perform realignment and normalization is not an obvious one: matching may be performed after the normalization or pose synthesis.
- Due to bounding box rescaling before the measurement even small misidentification of landmark localization may have significant effect on the user identification if we use landmarks to detect faces.

Facial alignment

Examples of misidentified and corrected features, limited to detecting 5 key features by Zhang et al.: Facial Landmark Detection by Deep Multi-task Learning.

