IRIS segmentation

Eduard Bakštein

edurard.bakstein@fel.cvut.cz

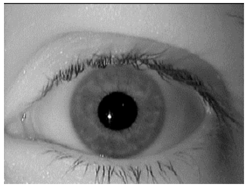
30.11.2023

acknowledgement: **Andrzej Drygajlo, EPFL Switzerland**

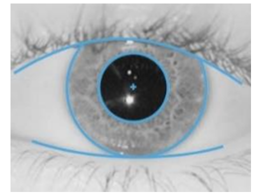
Iris recognition process

- Input: image of the eye
- Iris Segmentation
- Projection
- Feature extraction
- Encoding
- Comparison / matching

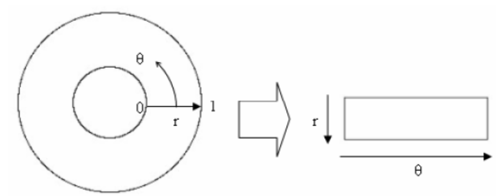
Iris recognition process



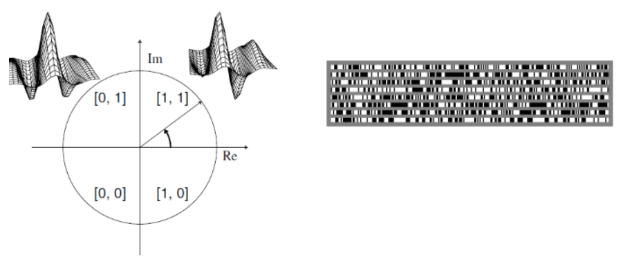
iris image



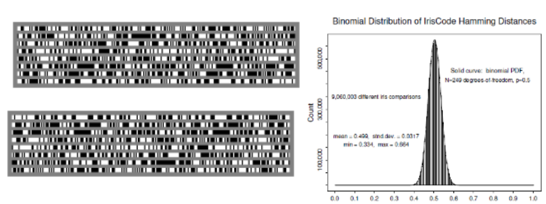
iris region segmentation



unwrapping



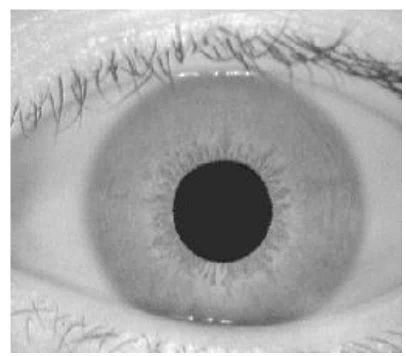
feature extraction & encoding

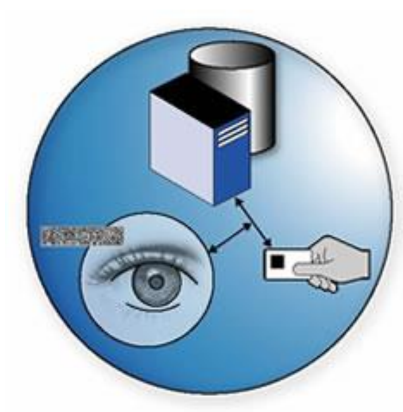


iris code comparison (database)

Result

Acquiring IRIS image





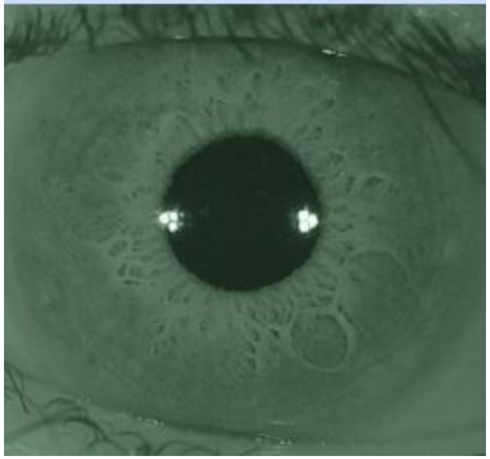
Visible or Infrared

Visible light

- Layers visible
- Less texture information
- Melanin absorbs visible light

(Near) Infrared light

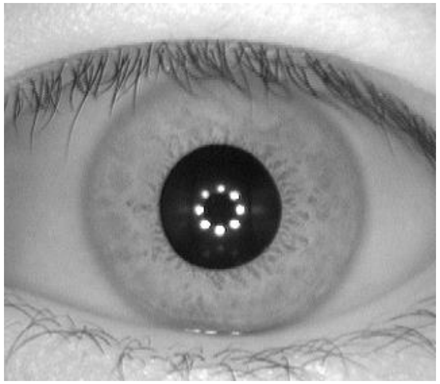
- (NIR)
- Melanin reflects most infrared light
- More texture is visible
- Specular reflections suppressed
- Preferred for iris recognition systems



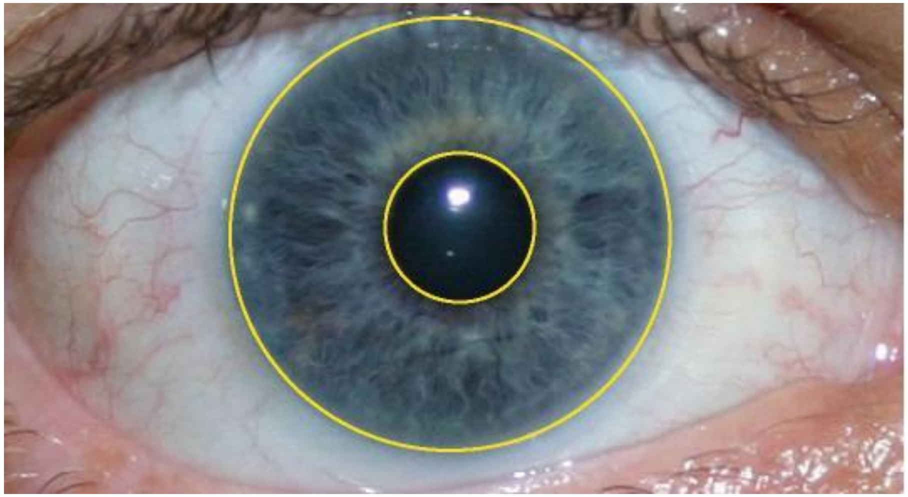
Iris image acquisition: requirements

- At least 70 pixels per iris radius (typically 100-140px)
- Monochrome CCD camera 640x480 px with NIR filter usually sufficient

- Getting the detailed view of the iris:
  1. Another wider-angle “face” camera used to steer the Iris camera to the direct spot
  2. User asked to move to desired position

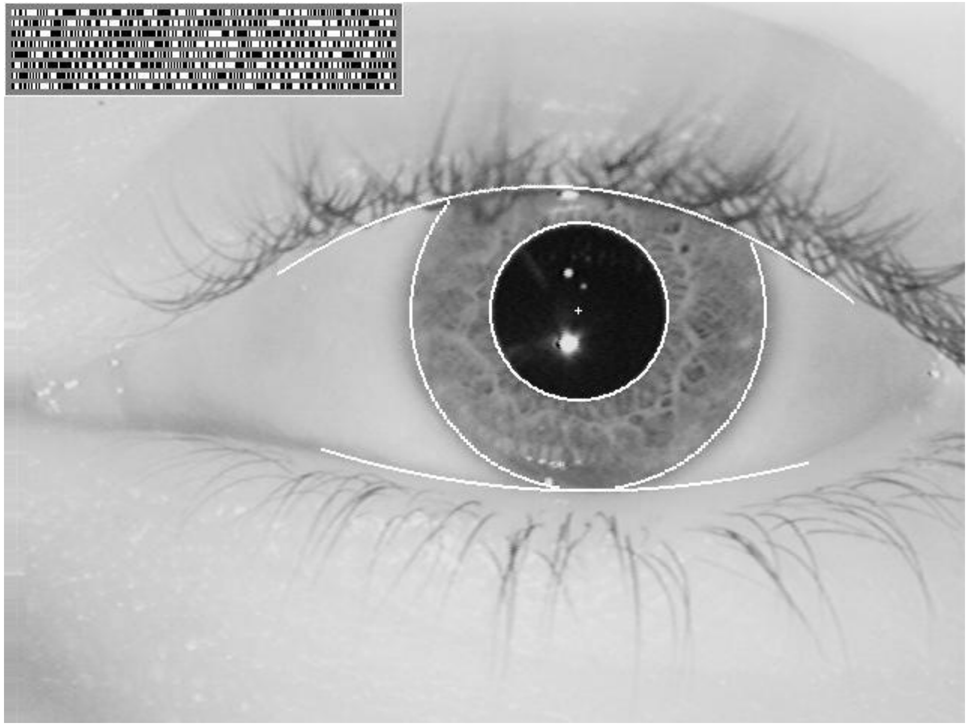


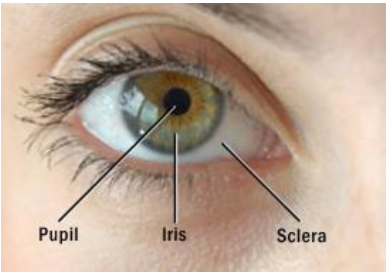
Segmentation



Aim: find the region of clean iris image

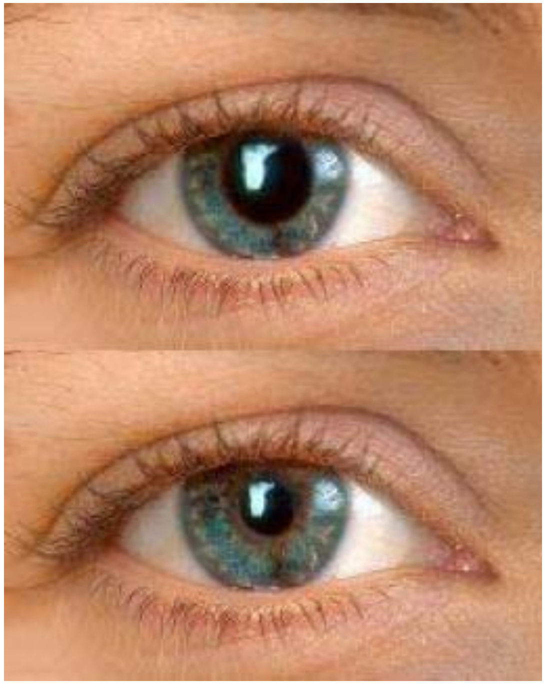
- Annular area between pupil and sclera
- Occlusions by eyelids and eyelashes need to be eliminated
- Easiest modelled by 2 circles





Intra-class variations

The segmenting algorithm has to address following problems:

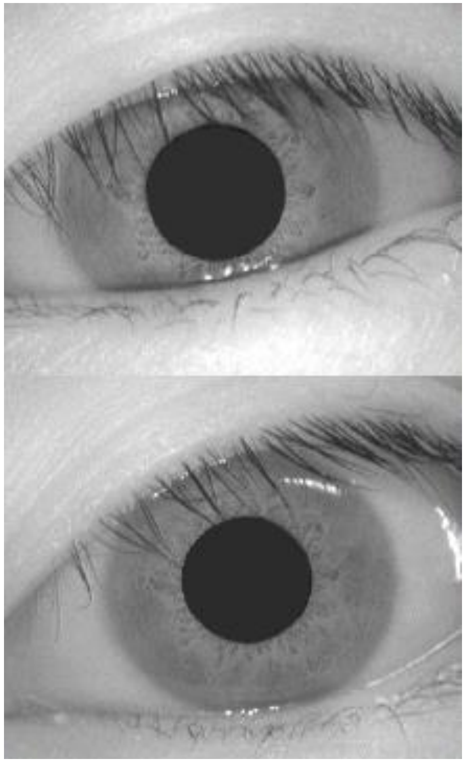


pupil dilation

(lighting changes)

inconsistent iris size

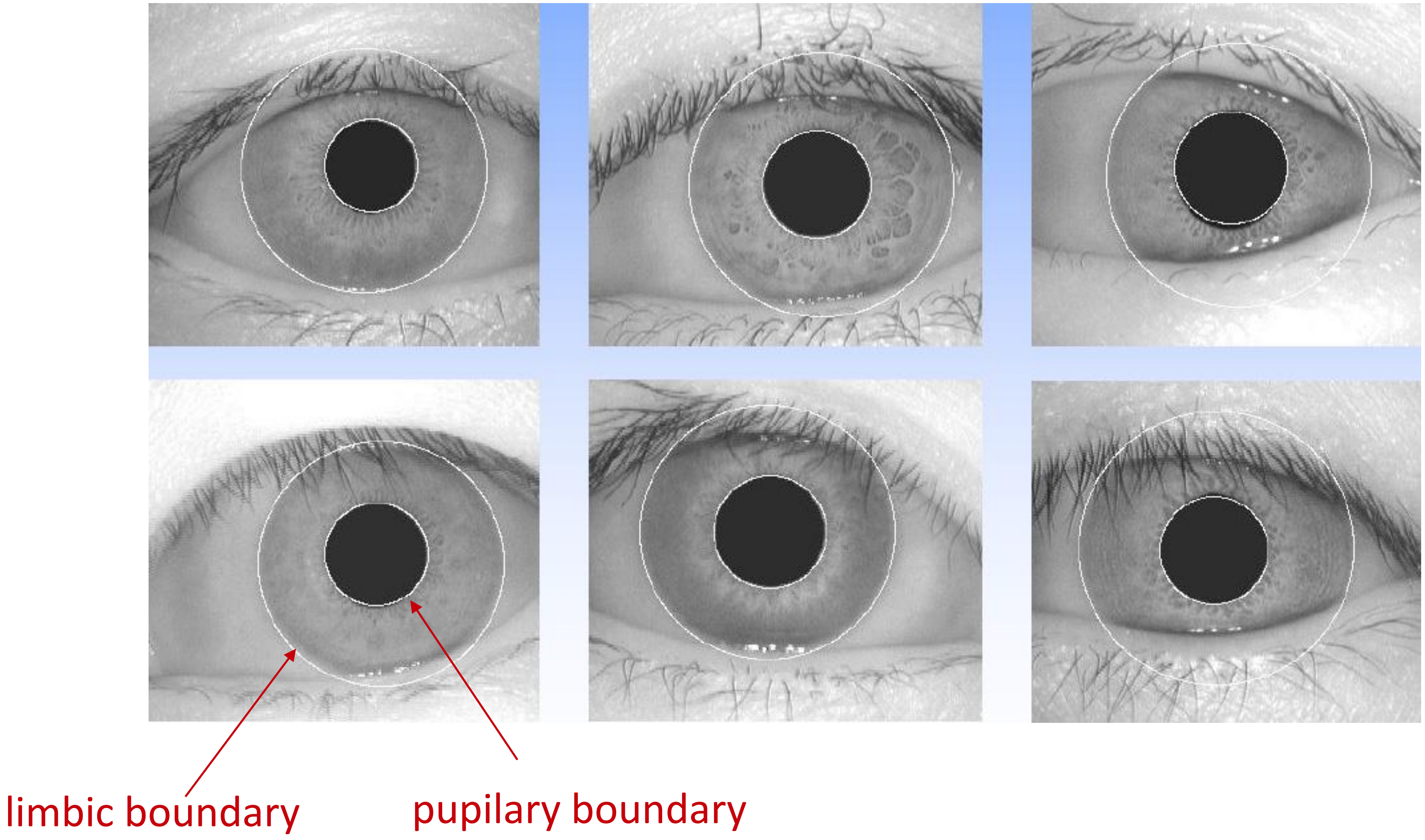
(distance from the camera)



eye rotation

(head tilt)

Detected Curvilinear boundaries



Curvilinear detector

Assumption: both the pupilary and limbic boundary can be approximated by (non-concentric) circles

(problem: off-axis gaze and specific cases)

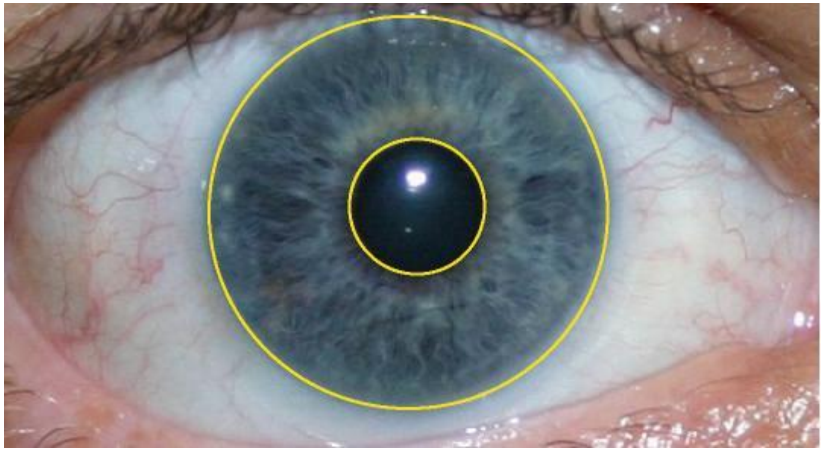
Daugman's approach

$$max_{(r,\,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,\,x_0,\,y_0} \frac{I(x,y)}{2\pi r}\, ds \right|$$

- searching circle parameters (x_0,y_0,r) that maximize blurred integro-differential function of the iris image. This maximum is gained when the circle parameters meet either the pupil or limbic properties.

Other possibility

- Hough transform
- RANSAC
- Active contours…

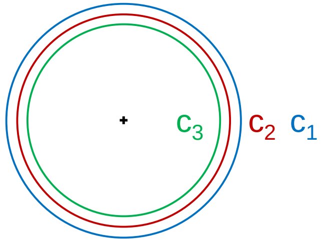


#1 Daugman's circular detector

$$max_{(r,\,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,\,x_0,\,y_0} \frac{I(x,y)}{2\pi r}\, ds \right|$$

I(x,y) – imag, G(r) – 1D Gaussian smoothing,

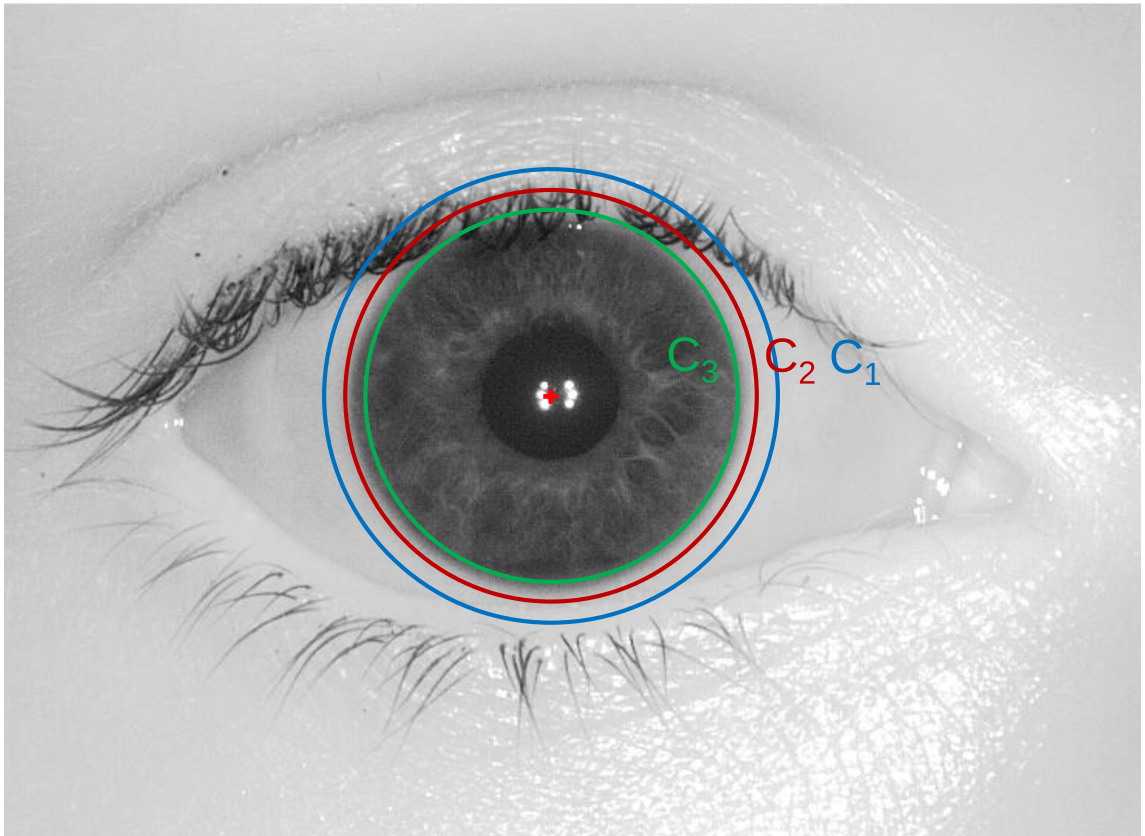
x_0,y_0,r – circle center coordinates + radius

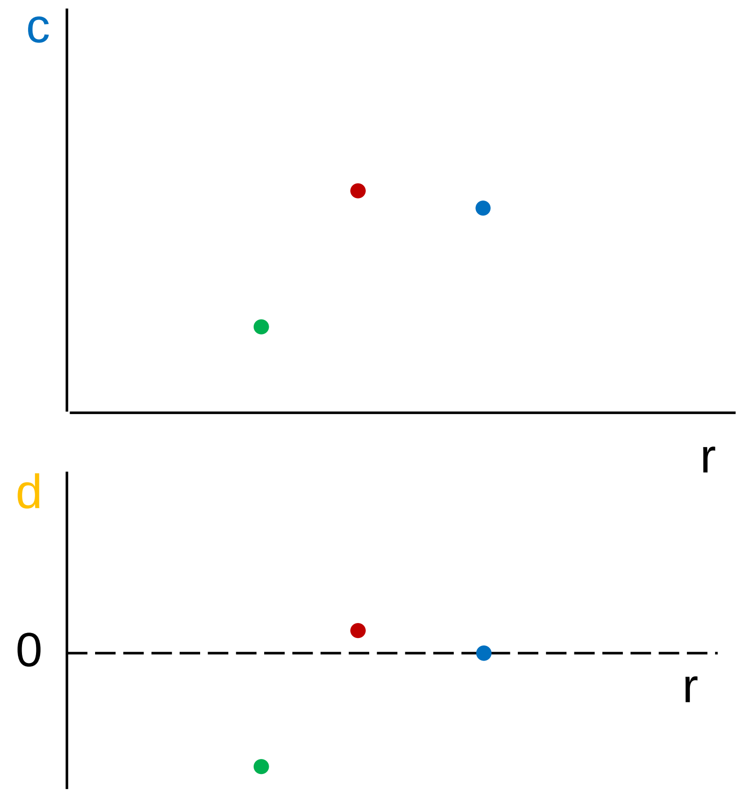


Idea: for given center x_0,y_0 and defined range of radius values $<r_{min}, r_{max}>$

1. c_1 = mean of image values over a circular path (x_0,y_0,r_{min})
2. change radius by 1px (until r_{max}), compute and store c_i using 1.
3. Compute difference $\bar{d} = diff(\bar{c})$

#1 Daugman‘s circular detector





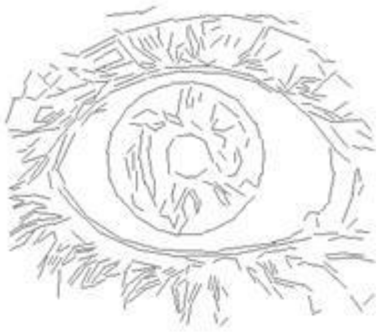
$$\max_{(r,x_0,y_0)} \left| \underbrace{\boxed{G_\sigma(r)}}_{\text{Gaussian smoothing}} * \underbrace{\boxed{\frac{\partial}{\partial r} \underbrace{\boxed{\oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r}\, ds}}_{c_i}}}_{d_i} \right|$$

#2 Hough transform

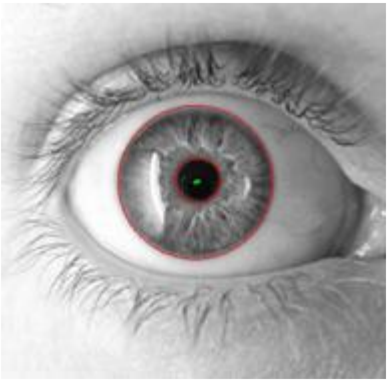
We search for most likely values of the circle parameters: (x_0,y_0,r)

The Hough procedure:

1. Edges are found in the image using edge detector
    - Threshold on local gradient in smoothed image



2. Projection to parametric space
3. Repeated for different circle sizes
4. Search the parameter space for maxima (the circle center and radius)

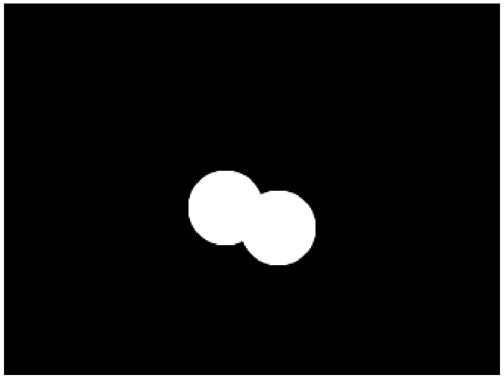


tutorial: http://www.aishack.in/tutorials/circle-hough-transform/

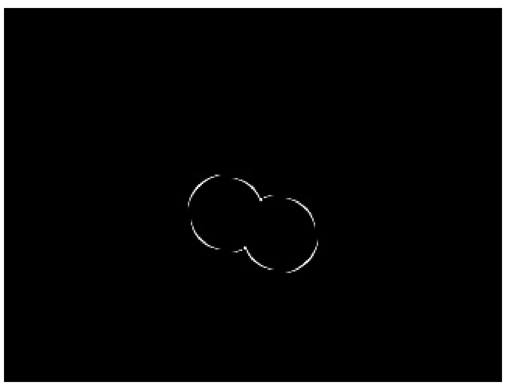
Hough transform 2: known radius

example

original image

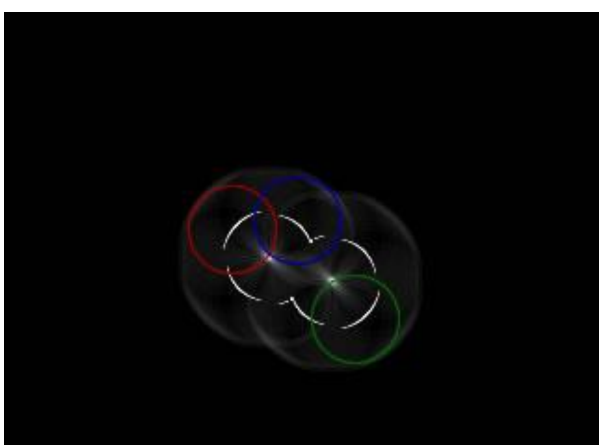


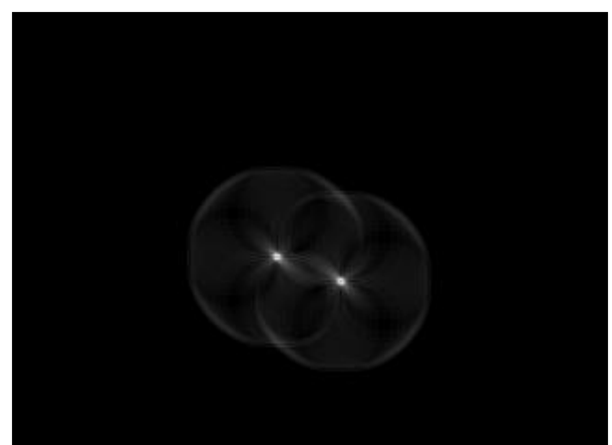
detected edges



- A circle of given radius is drawn around each edge point in the parameter space.
- Intersecting circles sum up.
- The most probable center for given radius is where most circles in the parameter space intersect = maximum value

“drawing” circles in the parameter space

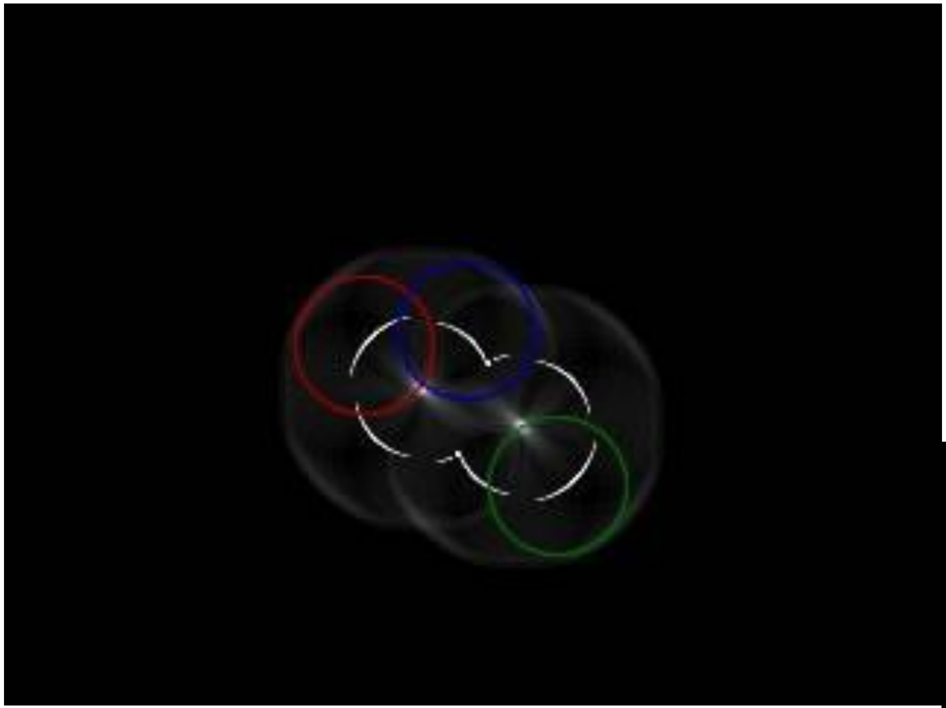




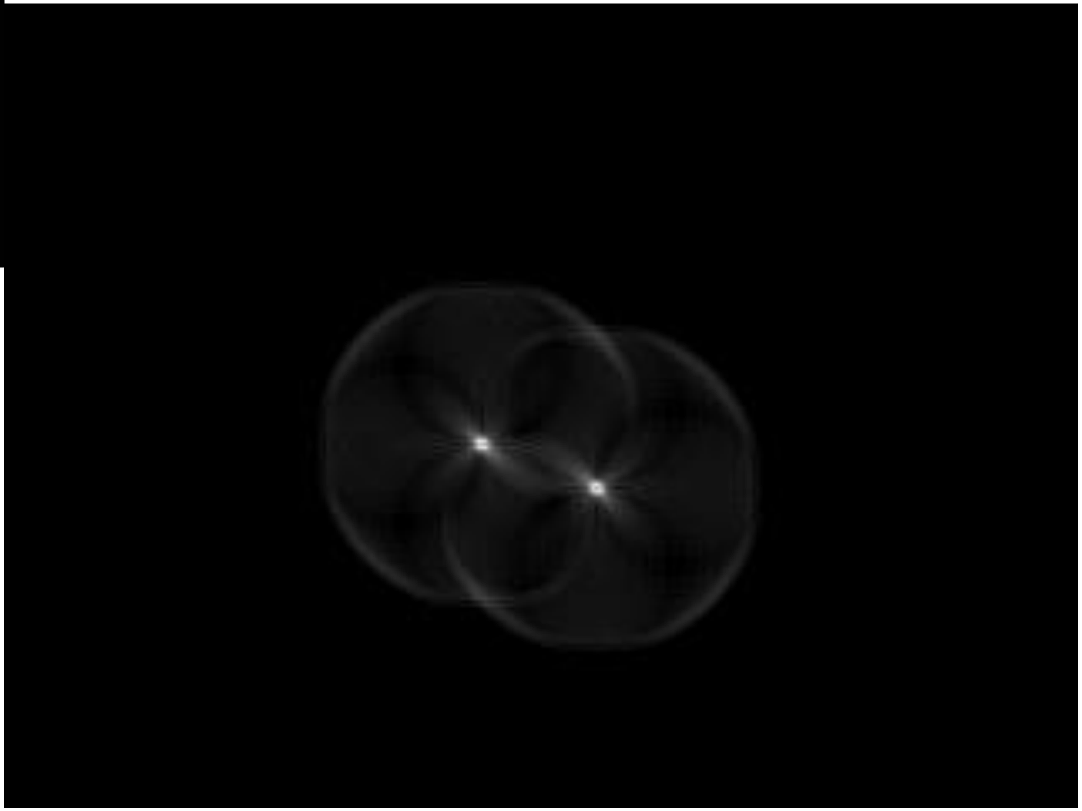
resulting parameter space

Source: http://www.aishack.in/tutorials/circle-hough-transform/

Hough transform 3: known radius



Parameter space
Here: intensity ~ value
(brighter = higher number)



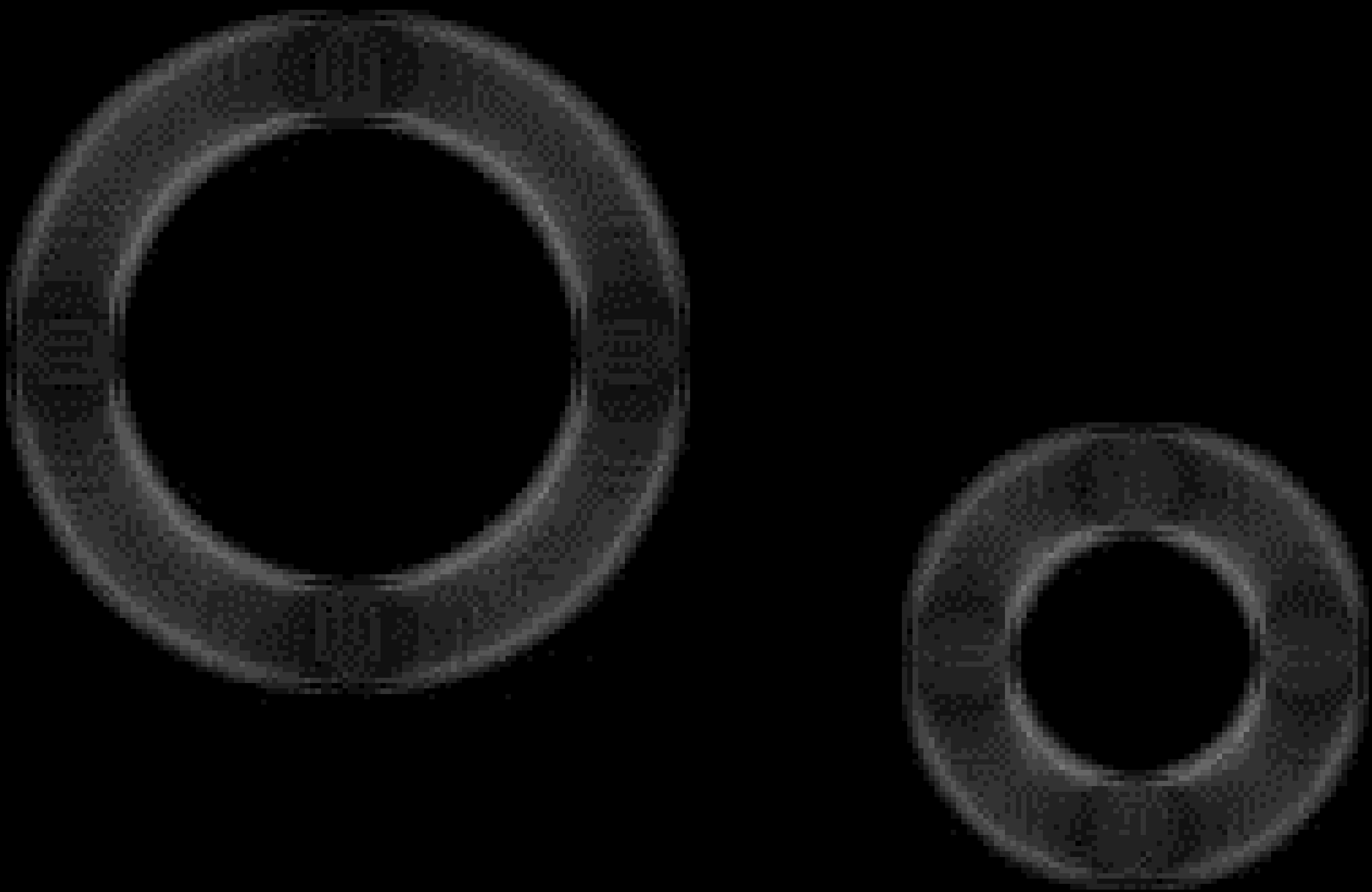
Source: http://www.aishack.in/tutorials/circle-hough-transform/

Hough transform: unknown radius

- Similar procedure
- Slice of parameter space created for each radius
- Searching global maximum
- Computationally intensive

video: http://www.aishack.in/static/img/tut/hough_circle.flv

(slices of the parameter space for different value of diameter r are shown)



R=9

Segmentation: Other options

RANSAC for circles (RANdom SAmple Consensus)

Operates on edge points (i.e. Canny detector)

1. Randomly pick subset of all original edge points,
2. Fit candidate circle to the subset (e.g. least squares: Gauss-Newton)
3. Throw away outliers – the points “far” from current candidate circle
4. Compute the number of inliers, if max. so far, name current model the optimal solution
5. Repeat 1-4 N times (or until sufficient fit achieved)

Active contours (“snakes”)

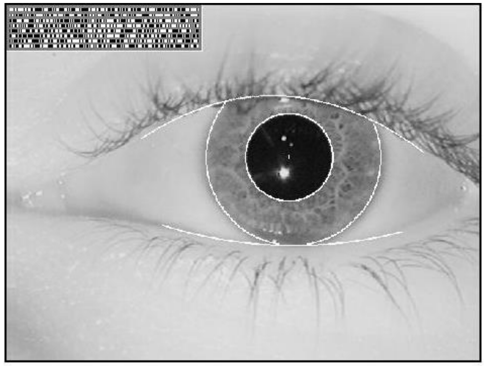
Can be used to improve non-circular iris segmentation from initial circular solution

CAREFUL PARAMETER SETTING CRITICAL FOR ALL ALGORITHMS!

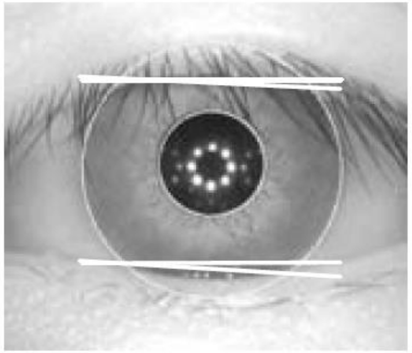
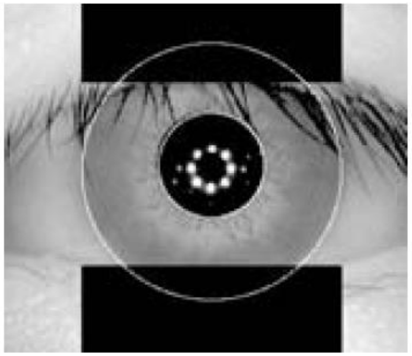
Eyelid boundaries

Similar procedures to annular iris region detection can be used. Many methods exist, e.g.:

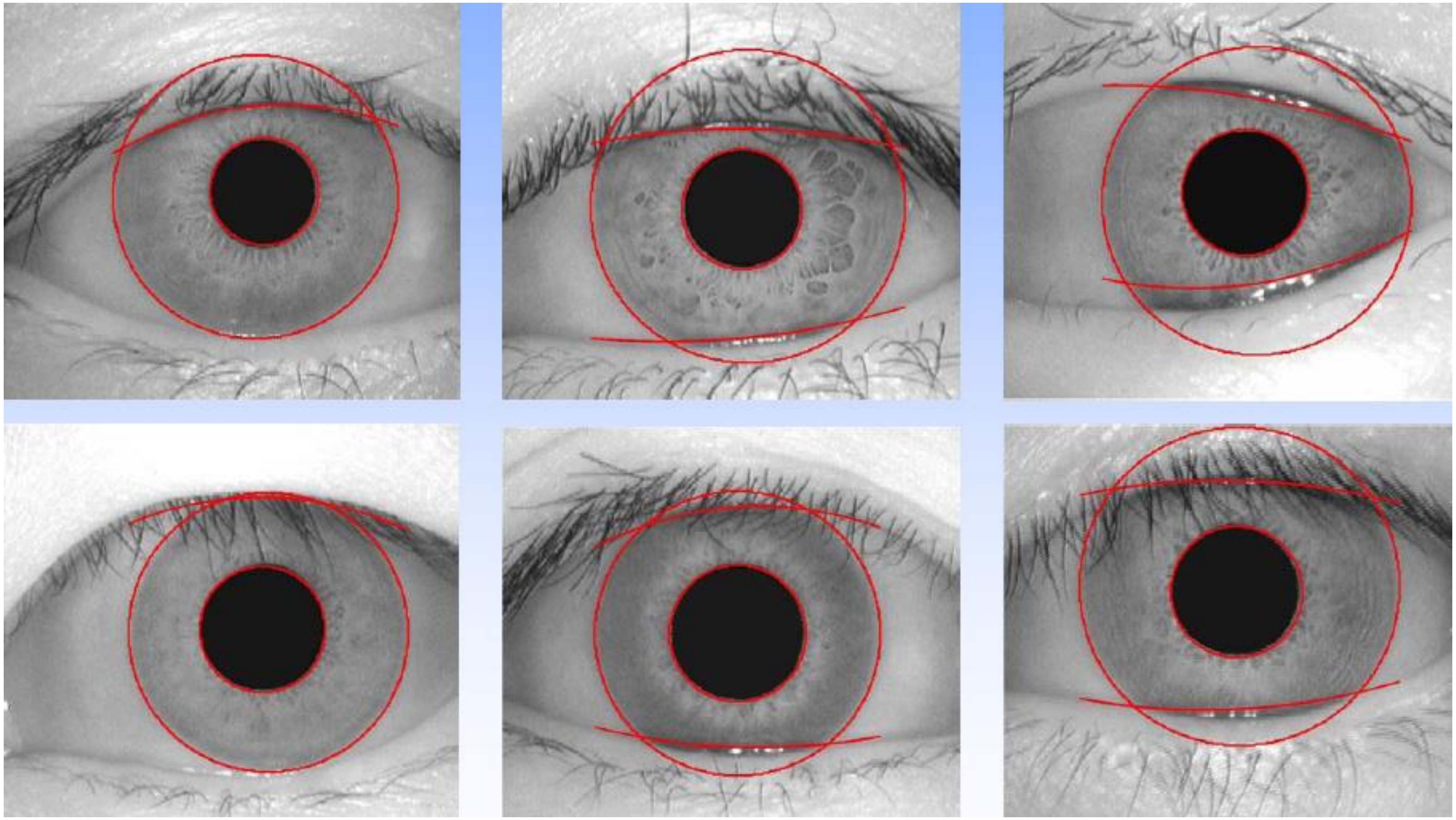
- Typical: Daugman's integro-differential operator with splines in place of circles



- Simplest: Hough transform with lines

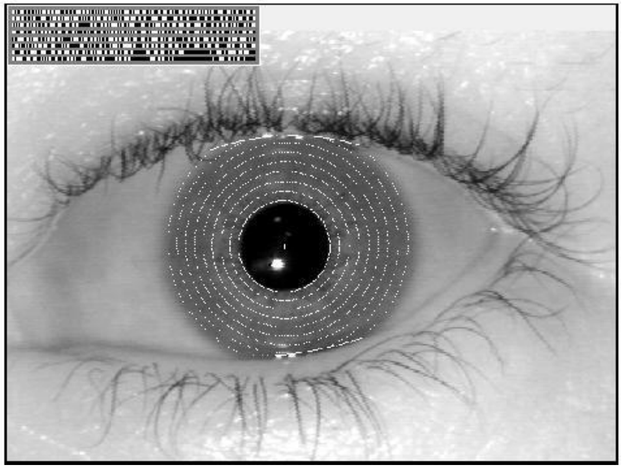
Detected eyelid boundaries



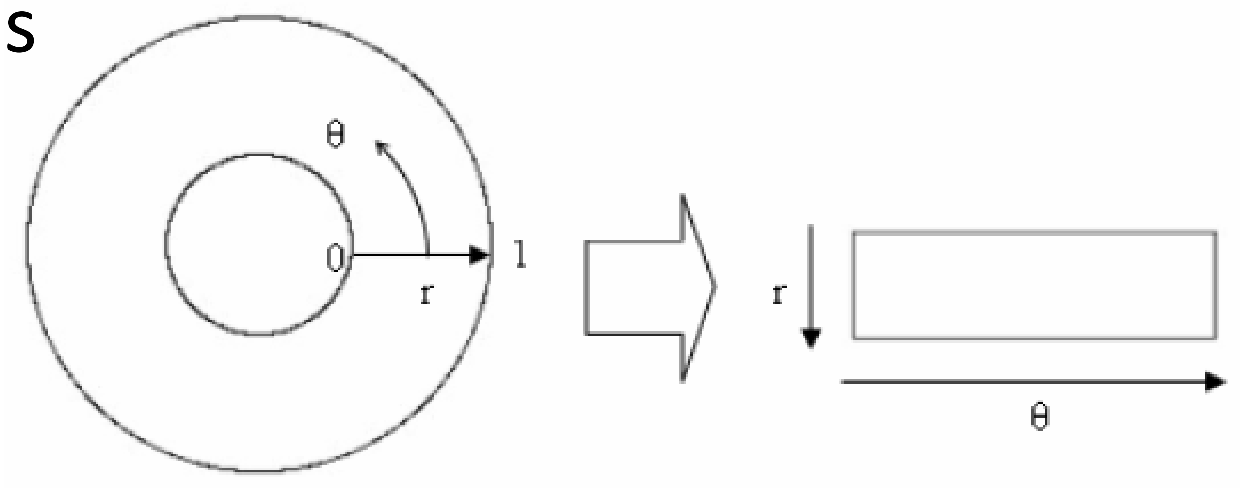
- Similar algorithm is used to detect eyelid boundaries

Projection

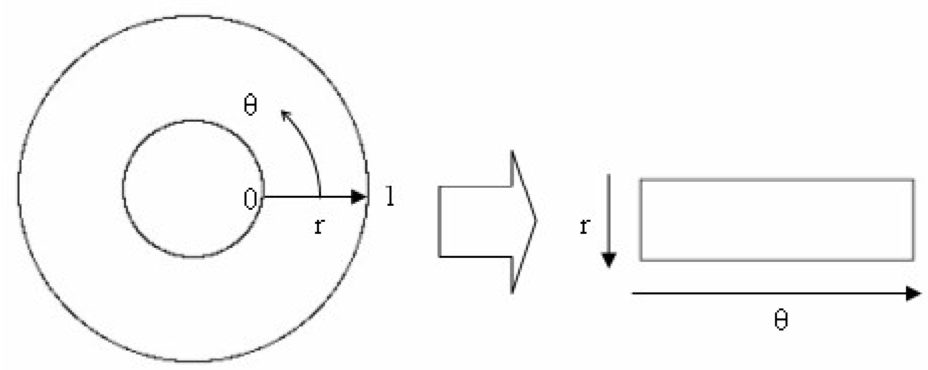
- The model has to be invariant to iris size (distance from camera), pupil size (amount of light)
- Invariance to rotation (head tilt) is addressed later in the recognition process

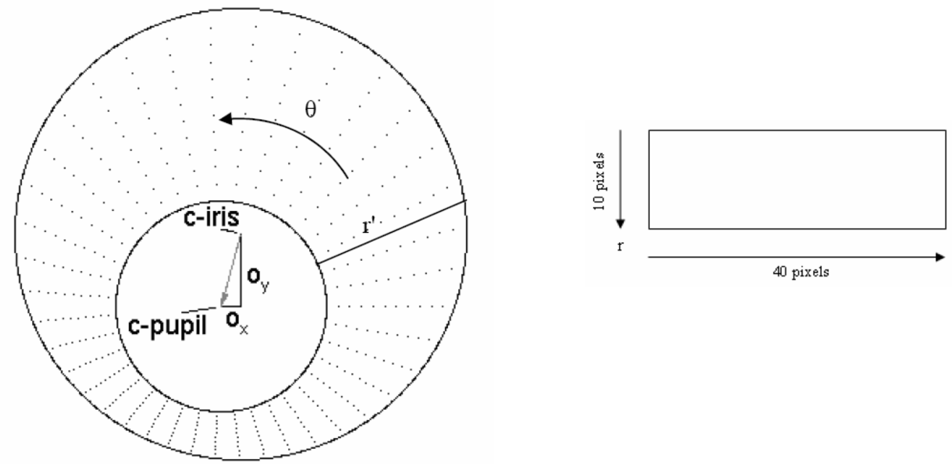


Solution: transformation to (pseudo)radial coordinates



Radial coordinates



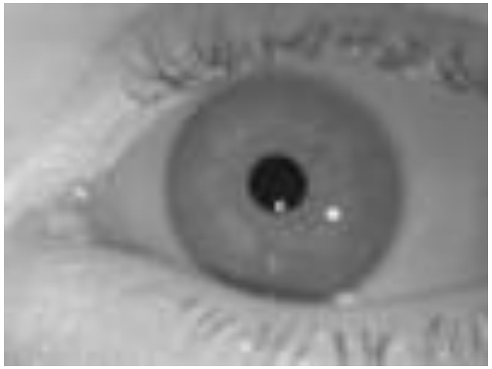
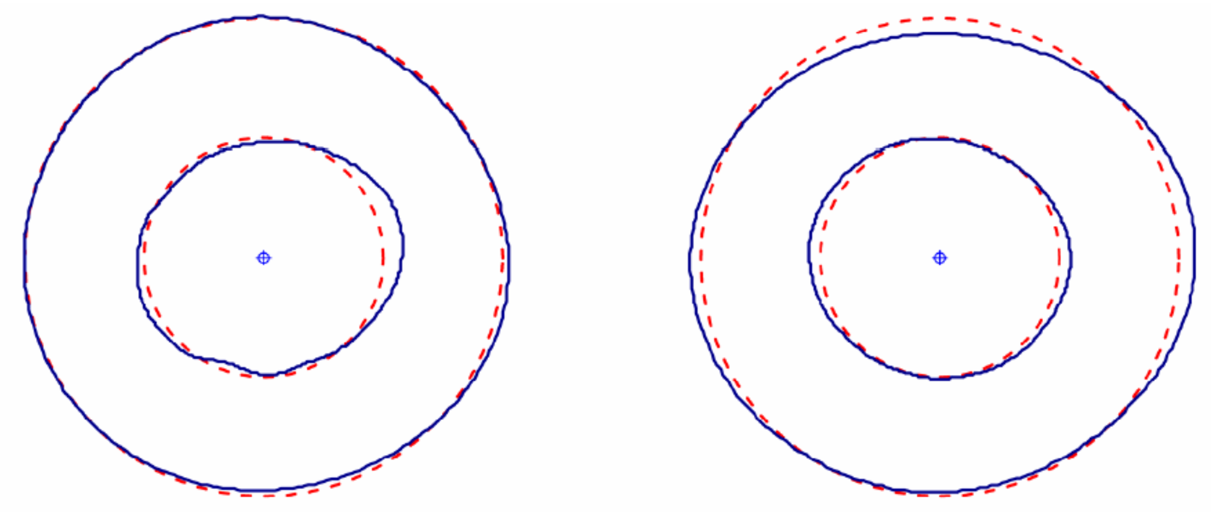


- Each point remapped to a pair of polar coordinates (ρ,θ), where ρ∈(0,1), θ∈(0, 2π)
- The model compensates pupil dilation and size inconsistencies in size and translation invariant coordinate system
- Rotational inconsistencies not compensated

Anomalous eye shape

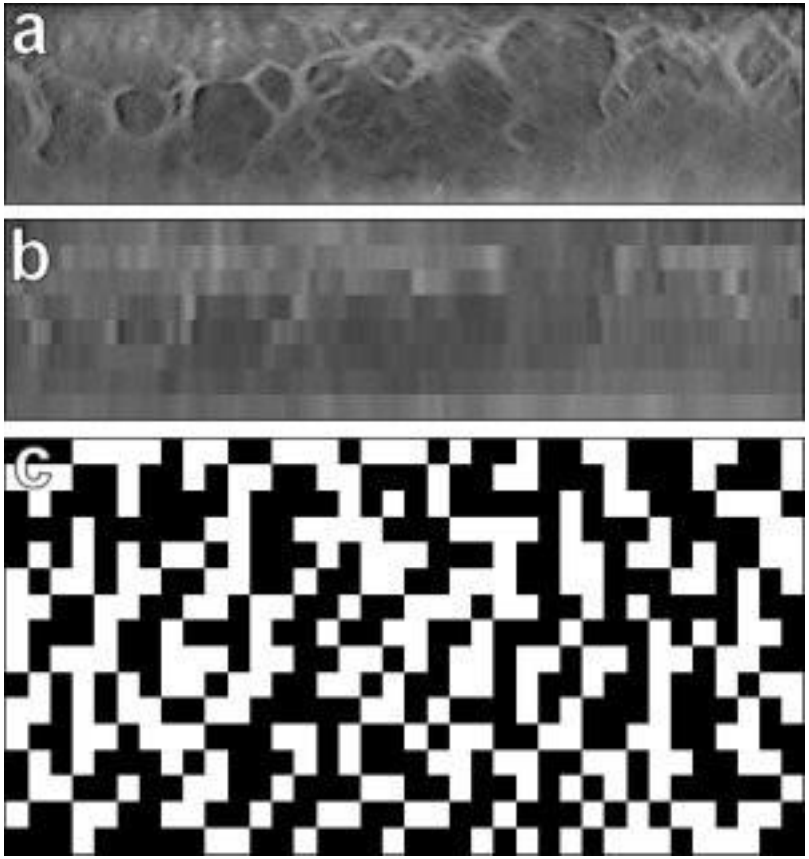
- The polar transform assumes circular iris boundary
- This may not be true especially for off-axis gaze
- Individual deviations can also play role



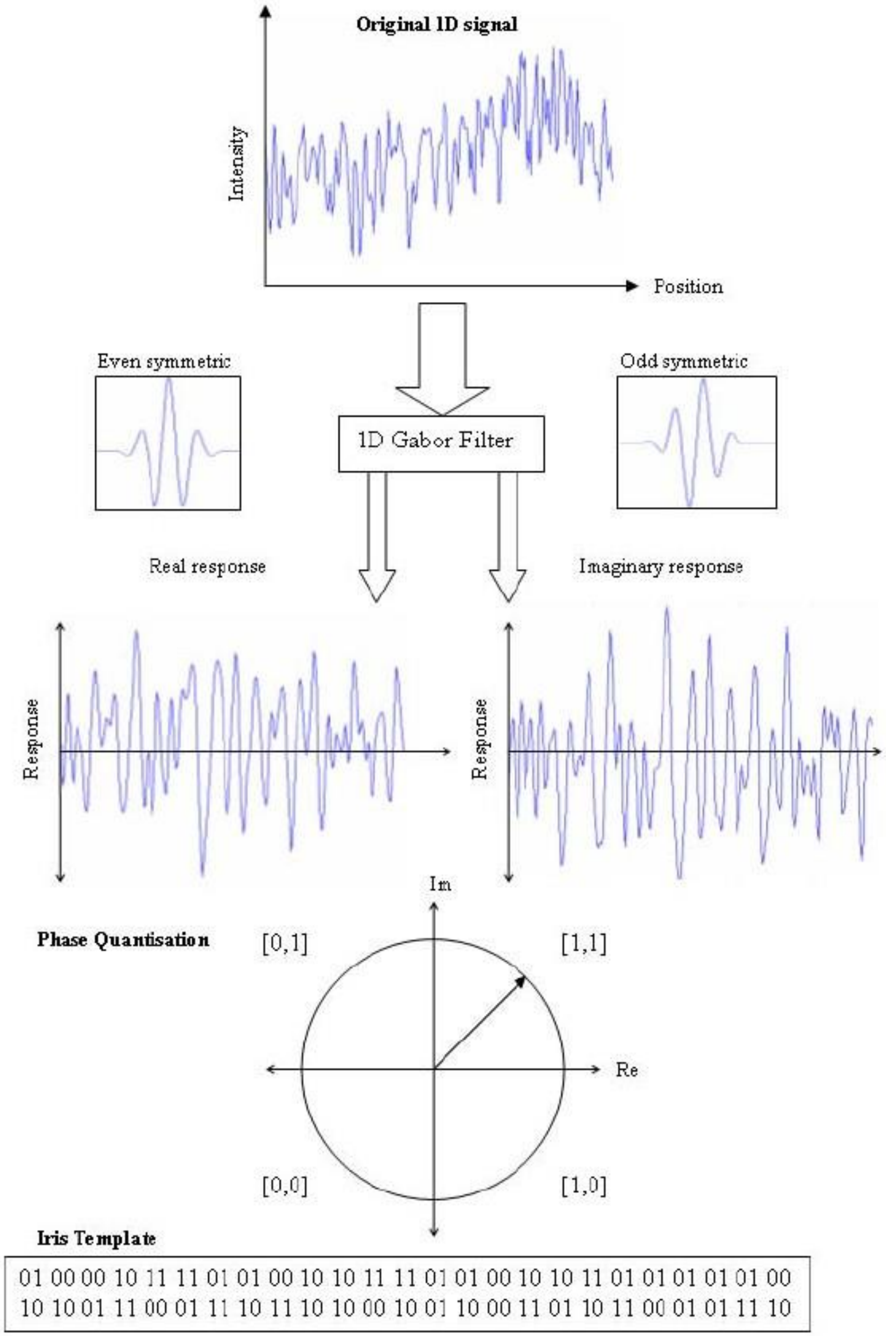
Feature extraction

- Processing the unwrapped image to extract information
- 2D Gabor wavelet filtering
- Phase quantization
- 2048-bit iris code



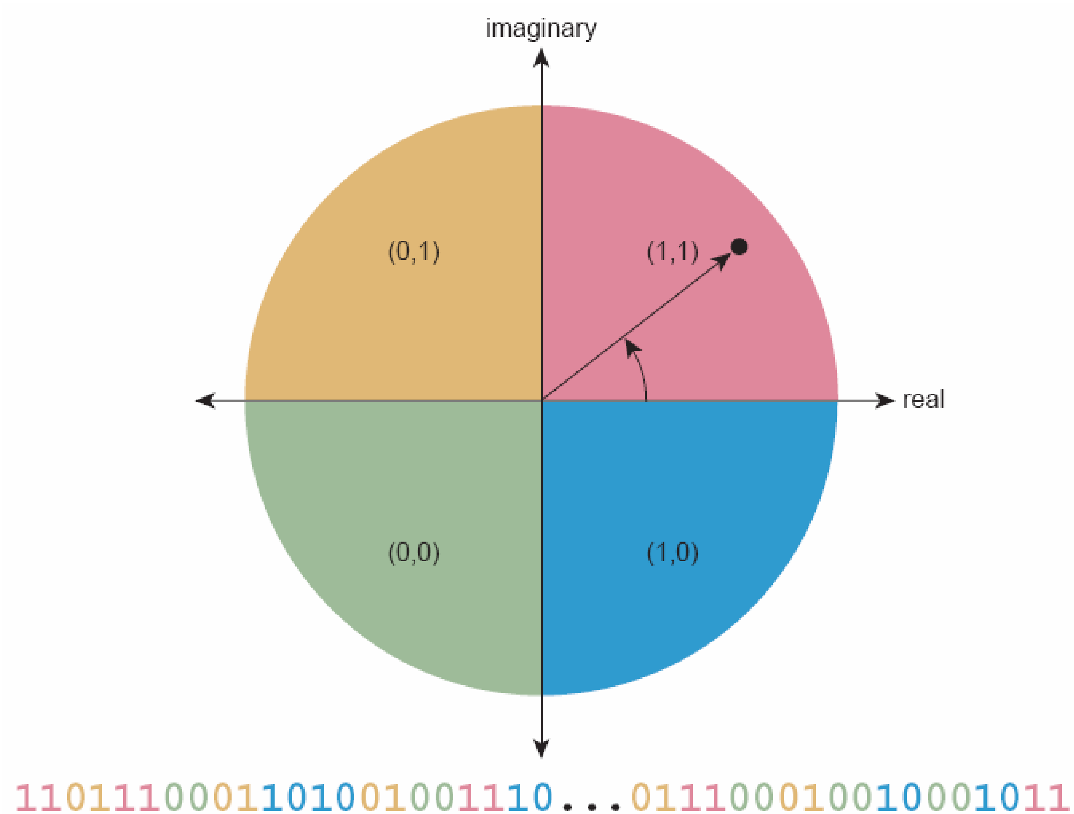
Gabor wavelet filtering

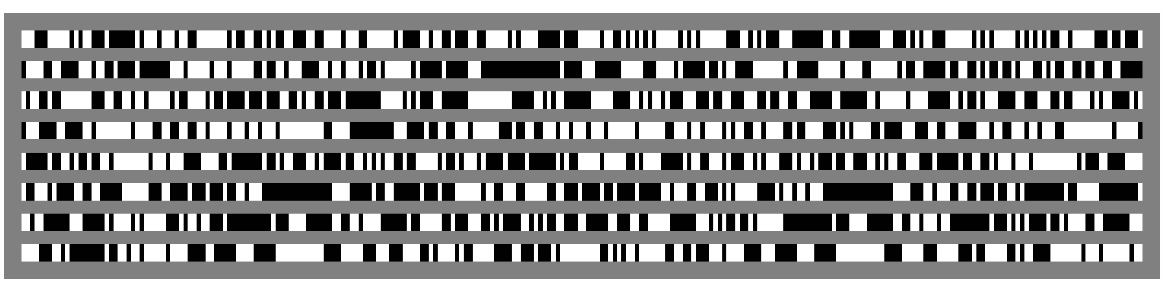
1D equivalent illustration



- The unwrapped iris image is filtered using two 2D Gabor wavelet filters using multiple parameter settings.
- The demodulating wavelets are parameterized with four degrees-of-freedom: size, orientation, and two positional coordinates. They span several octaves in size, in order to extract iris structure at many different scales of analysis

Encoding: Phase quantization

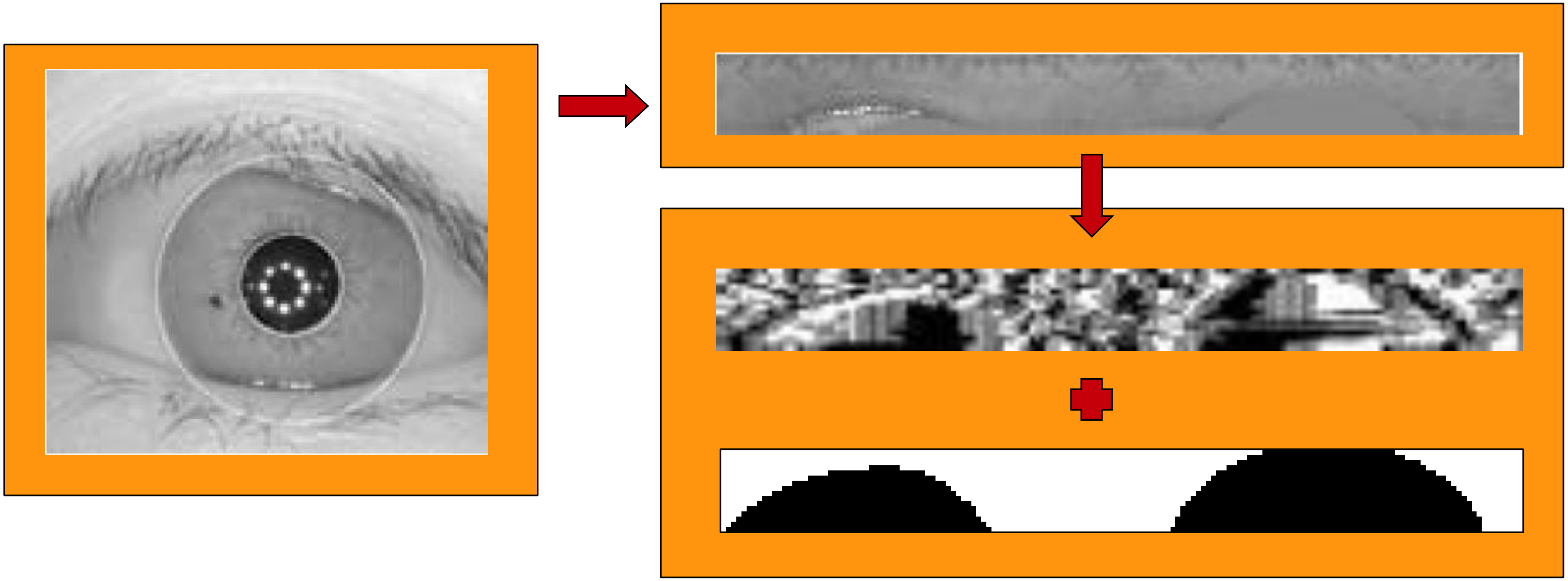




- The phase of resulting complex numbers is observed and coded into 2 bits according to the figure
- Phase quantization - continuous phase to 2 bits
- 2048 such phase bits (256 bytes) are computed for each iris.

Masking

- Areas with noise (eyelids, eyelashes...) need to be excluded
- A binary mask of the same size as the iris code is calculated. 1 in the areas of useful signal, 0 elsewhere



Iris code

Projection: doubly-dimensionless polar coordinate system

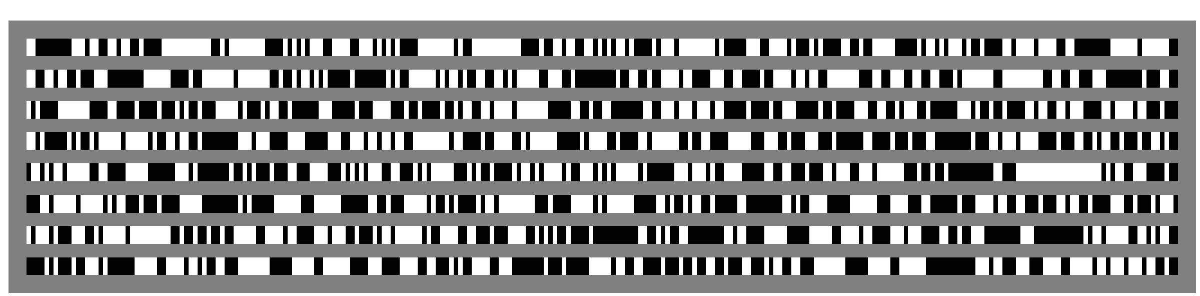
- invariant to the **size of the iris** (imaging distance and the optical magnification factor) and **pupil dilation** (lighting)

Filtering: only phase information used

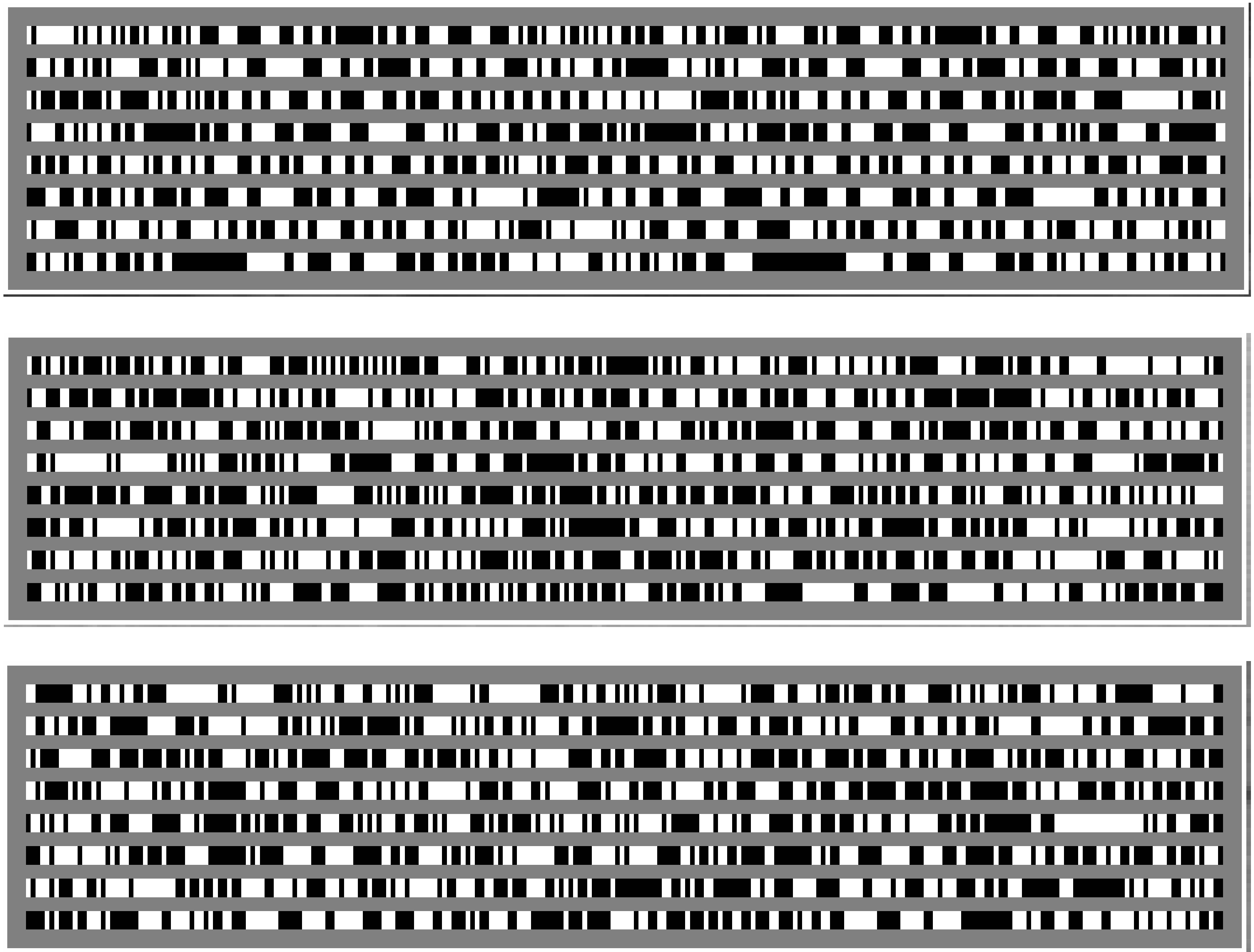
- invariant to contrast, absolute image function value (camera gain), and illumination level (unlike correlation methods)

Very compact

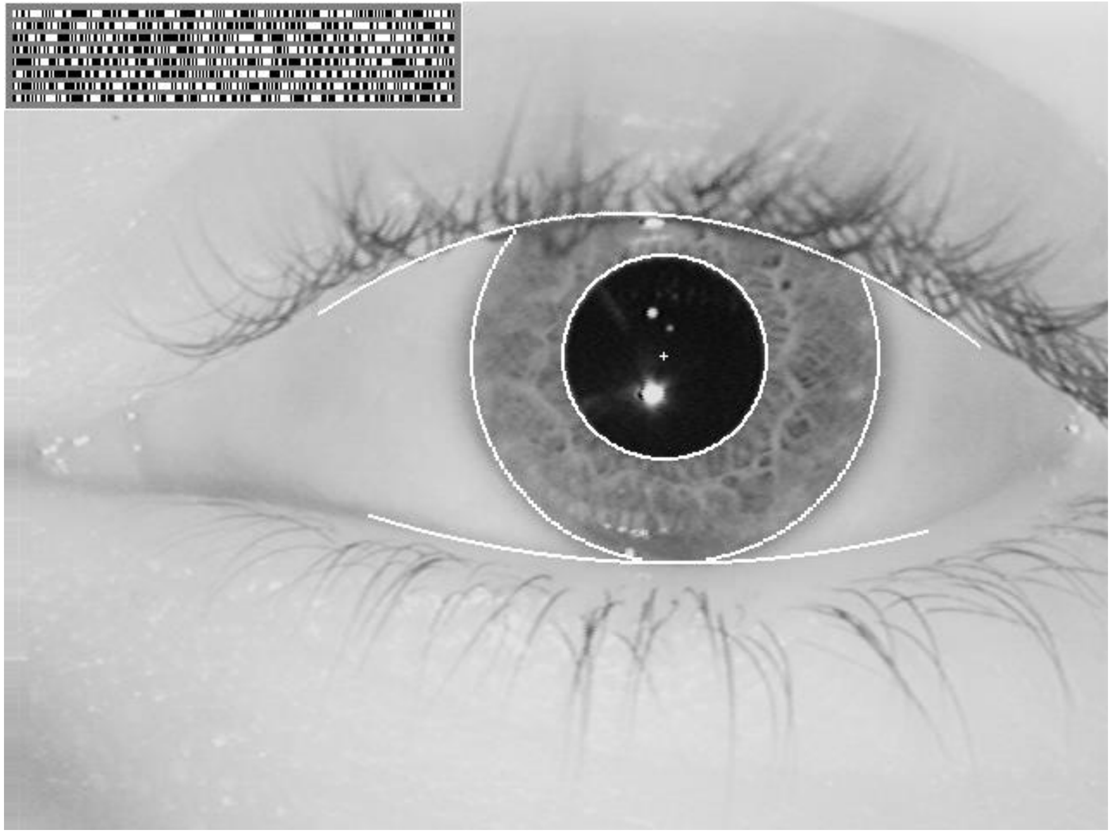
- Typically just 256 bytes + 256 bytes mask (depends on settings of the Gabor wavelet filtering) - small for storage
- Thanks to phase quantization.



Example iris codes



Iris code comparison



- Different eyes’ Iris Codes are compared by vector Exclusive-OR’ing in order to detect the fraction of their bits that disagree.

Iris code comparisons

Iris code bits are all of equal importance

Hamming distance:

- Distance between 2 binary vectors (strings)
- Number of differing bits (characters)
- “Number of substitutions required to change one string to the other”
- Sequence of XOR and *norm* operators (number of ones in XOR'ed sequences)

Examples:

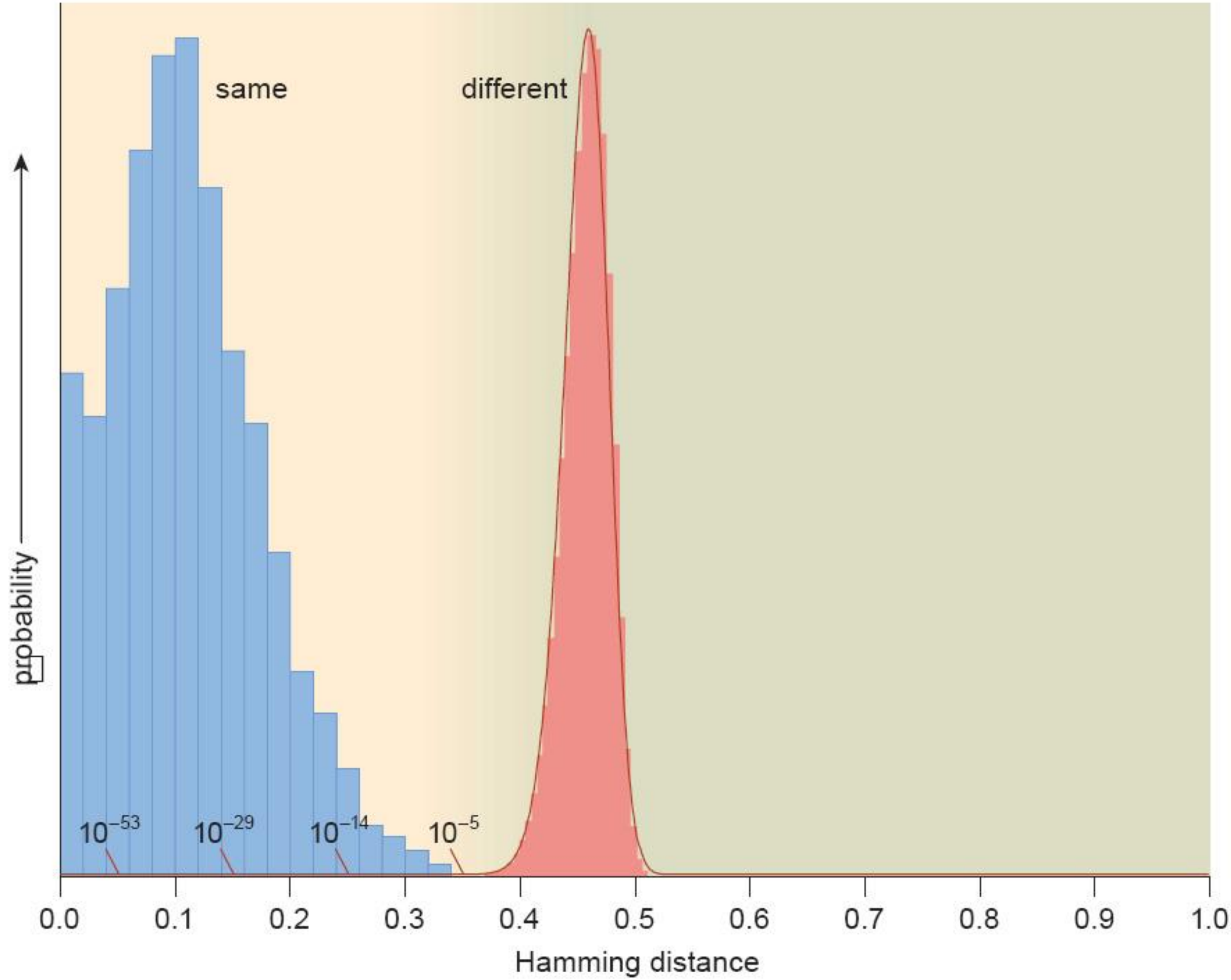
- hockey and soccer, H=3
- 1001011 and 1100011, H=2

Code comparison

$$H = \frac{\|(codeA \otimes codeB) \cap maskA \cap maskB\|}{\|maskA \cap maskB\|}.$$

- $\otimes$ - XOR operator - one for each bit that disagrees
- codeA codeB - iris codes,
- $\cap$ - AND - keep only bits unmasked by both masks
- maskA maskB - noise masking templates for respective iris codes
- || - norm operator - calculate number of bits = 1
- Normalized by the number of bits that are available in both codes (denominator)

Iris comparisons



Comparison properties

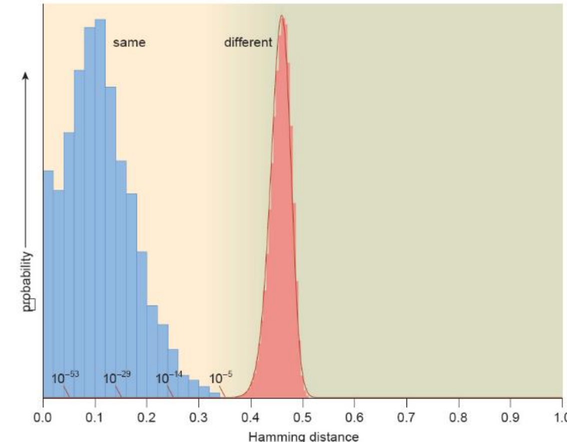
Left distribution: different images of the same eye are compared; typically about 10% of the bits may differ.

Right distribution: IrisCodes from different eyes compared, with rotations (best match - min HD). Tightly packed around 45%

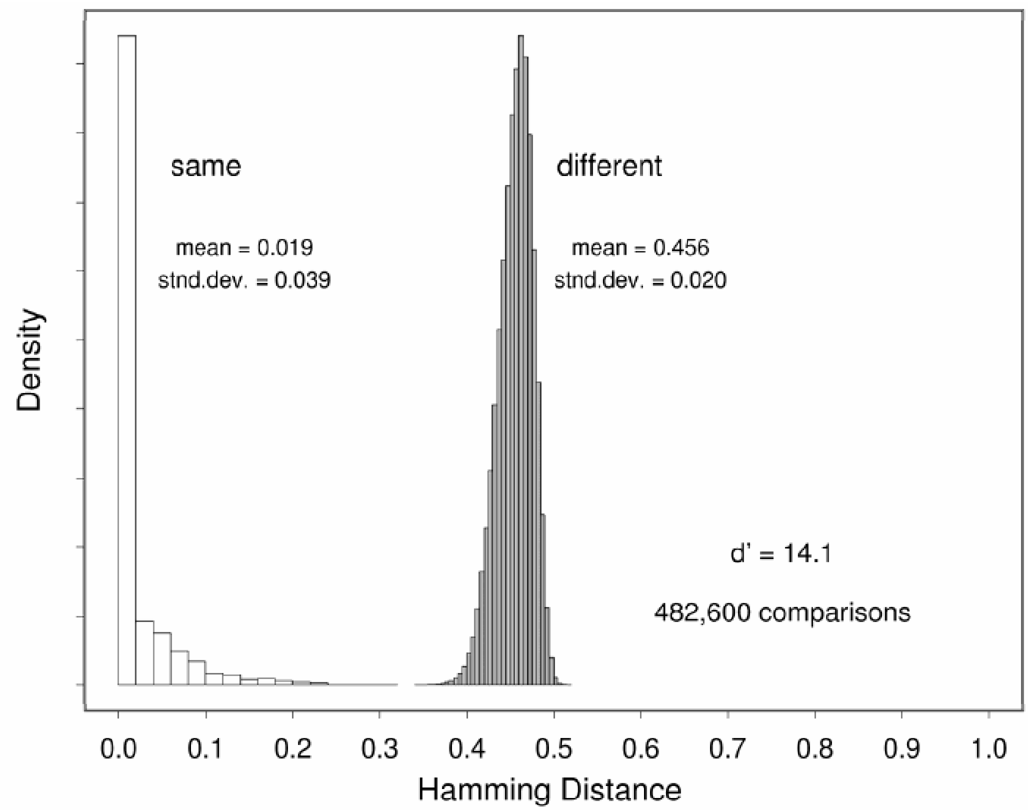
Very narrow right-hand distribution (different irises), it is possible to make identification decisions with astronomic levels of confidence.

Probability of two different irises agreeing just by chance in more than 75% of their IrisCode bits (HD<0.25) is only 1 in 10^{14}

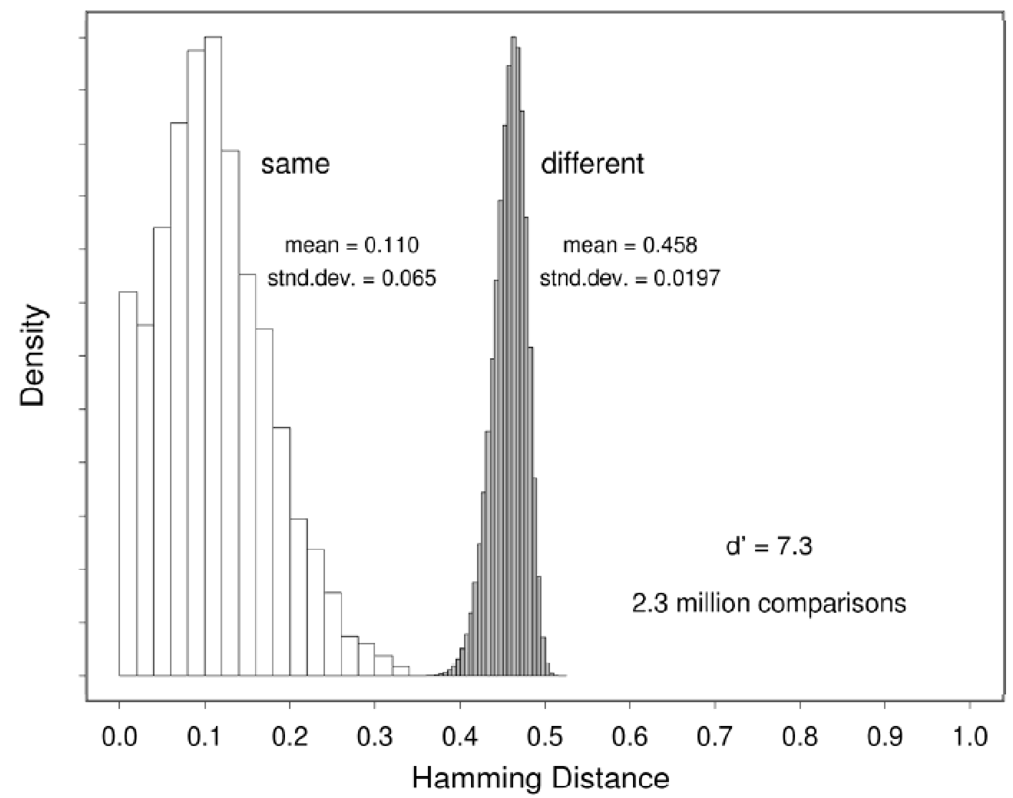
Extremely low probabilities of False Match enable the iris recognition algorithms to search through extremely large databases (10^{10}) scale despite many opportunities to make a false match



Comparisons: system quality



ideal imaging



non-ideal imaging

Comparing distributions for the same and different irises says a lot about the identification system

Comparison: false match rate

Observed False Match Rates in 200 billion comparisons

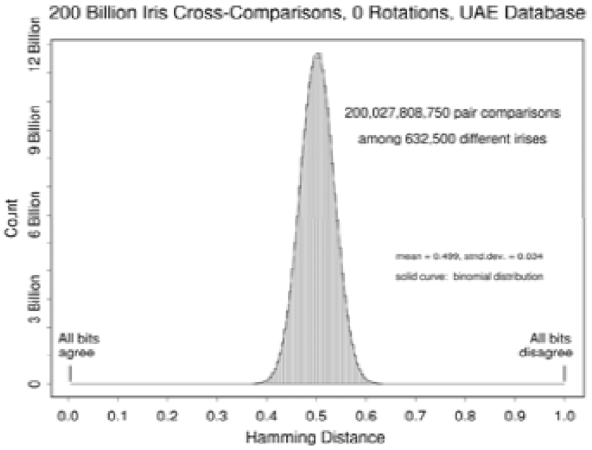
HD Criterion Policy	*Observed False Match Rate*
0.220	0 *(theor: 1 in 5 $\times 10^{15}$)*
0.225	0 *(theor: 1 in 1 $\times 10^{15}$)*
0.230	0 *(theor: 1 in 3 $\times 10^{14}$)*
0.235	0 *(theor: 1 in 9 $\times 10^{13}$)*
0.240	0 *(theor: 1 in 3 $\times 10^{13}$)*
0.245	0 *(theor: 1 in 8 $\times 10^{12}$)*
0.250	0 *(theor: 1 in 2 $\times 10^{12}$)*
0.255	0 *(theor: 1 in 7 $\times 10^{11}$)*
0.262	1 in 200 billion
0.267	1 in 50 billion
0.272	1 in 13 billion
0.277	1 in 2.7 billion
0.282	1 in 284 million
0.287	1 in 96 million
0.292	1 in 40 million
0.297	1 in 18 million
0.302	1 in 8 million
0.307	1 in 4 million
0.312	1 in 2 million
0.317	1 in 1 million

IrisCode statistics: Bernoulli trials

Jacob Bernoulli (1645-1705) analyzed *coin-tossing* and derived the binomial distribution. If the probability of "heads" is p, then the likelihood that a fraction $x = m/N$ out of N tosses will turn up "heads" is:

$$P(x) = \frac{N!}{m!(N-m)!} p^m (1-p)^{(N-m)}$$

University of Groningen



Code comparisons: masking

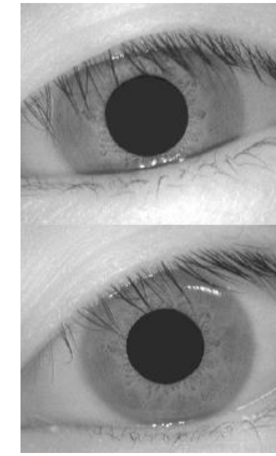
- In case of differing iris parts occluded in the two compared iris images, the number of effective bits can be very low.
- The probability of false match increases.
- Renormalization of HD by the number of available bits is necessary, as well as is the decision criterion

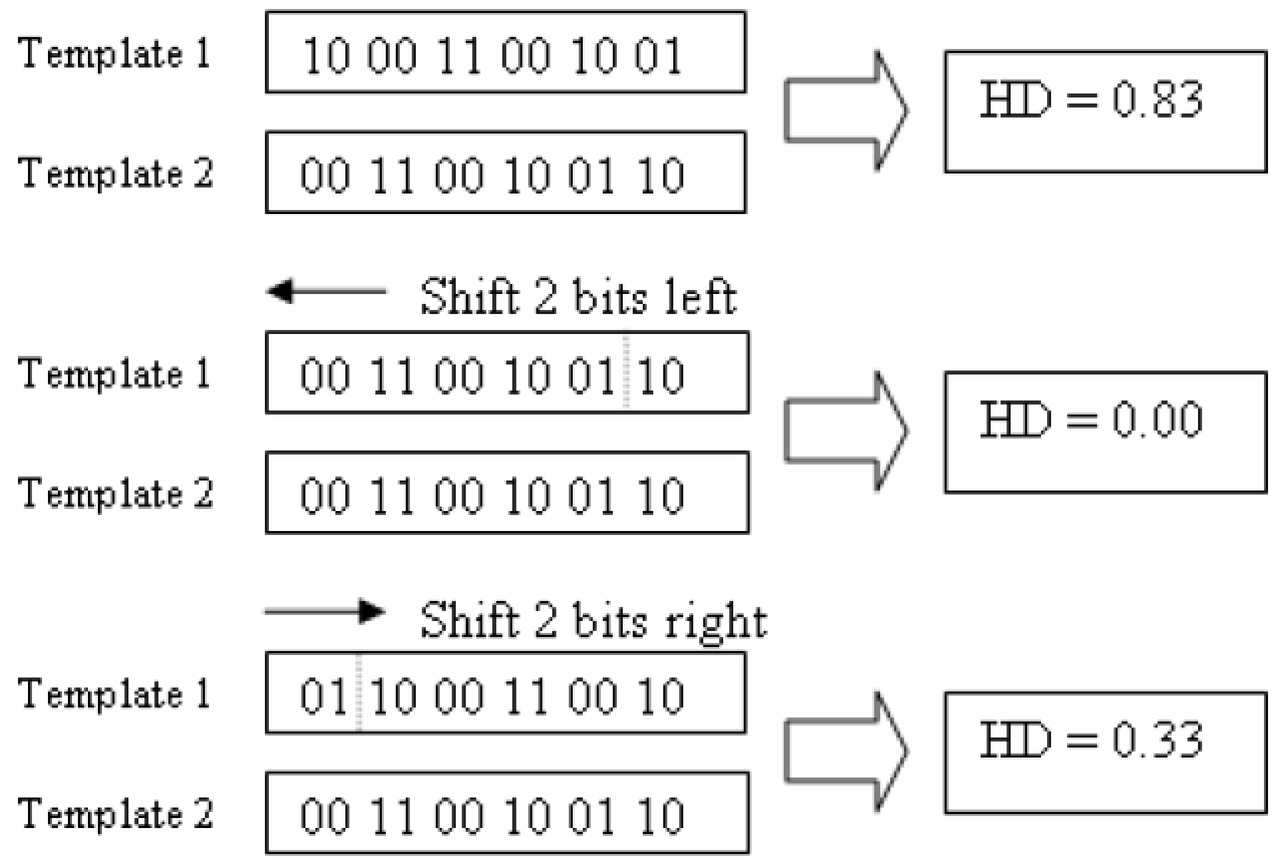
$$HD_{norm} = 0.5 - (0.5 - HD_{raw}) \cdot \sqrt{\frac{n}{N_{typical}}}$$

- $N_{typical}$ is typical number of available bits in given database
- Formula based on Bernoulli distribution

irisCode comparisons: rotation

- To account for iris rotation, the codes are shifted one against another in selected range
- Minimum HD is calculated





irisCode comparison Performance

- On a 300MHz PC (long ago):

Operation	Execution time
Assessing image focus	15 ms
Scrubbing specular reflections	56 ms
Localizing the eye and iris	90 ms
Fitting the pupillary boundary	12 ms
Detecting and fitting both the eyelids	93 ms
Removing eyelashes and contact lens artifacts	78 ms
Demodulation and IrisCode creation	102 ms
XOR comparison of any two IrisCodes	10 μs

Key messages

1. Iris region found by circular detector
2. Image unwrapped in a polar coordinate system
3. Image filtered using Gabor wavelet filters
4. Only phase information is used (phase quantization)
5. Phase quantization converts filtered image to binary code
6. Binary mask showing noise, eyelids and eyelashes stored along with the code
7. Iris codes compared using hamming distance
8. Iris recognition has extremely low false accept rate

Iris recognition summary

Strengths

- It has the potential for exceptionally high levels of accuracy
- It is capable of reliable identification as well as verification
- Believed to be the most reliable metric
- Stability of characteristic over a lifetime
- Distant cameras – less obtrusive

Weaknesses

- Acquisition of the image requires moderate training and attentiveness
- It is biased for false rejection (better for identification)
- A proprietary acquisition device is necessary for deployment - expensive
- There is some user discomfort with eye-based technology
- Sunglasses, ambient light etc

Thank you for your attention