Cross entropy clustering approach to iris segmentation for biometrics purpose

2015.02.16-21

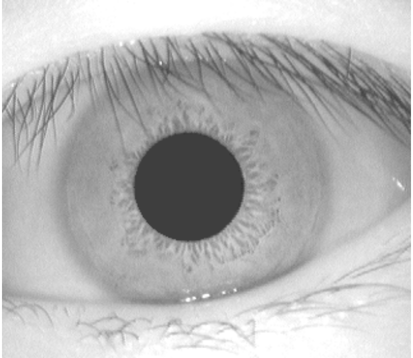
Krzysztof Misztal

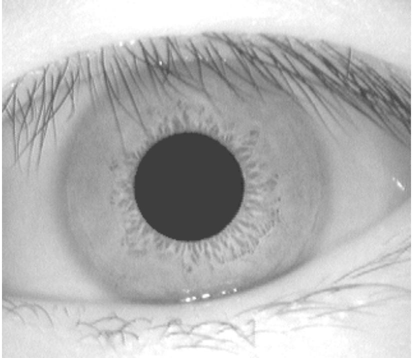
misztal.edu.pl

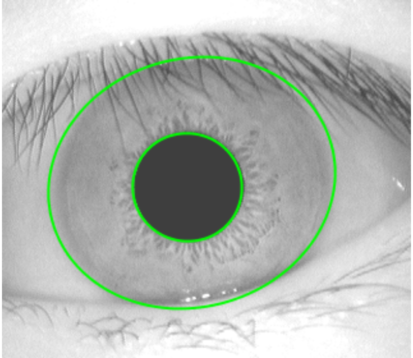
Outline

Outline

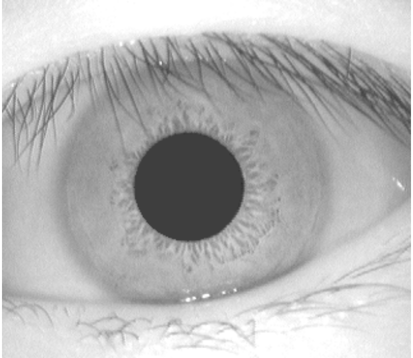
Iris pattern recognition







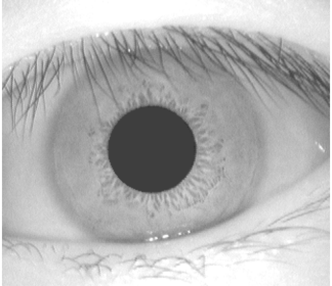
Iris image segmentation.



Iris image segmentation.

Sometimes it is difficult.

Iris image segmentation – quick look



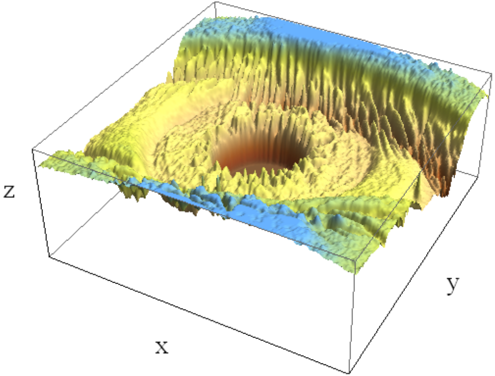


Image interpretation
and its consequences.

Iris image segmentation – quick look

The interesting regions of the image:

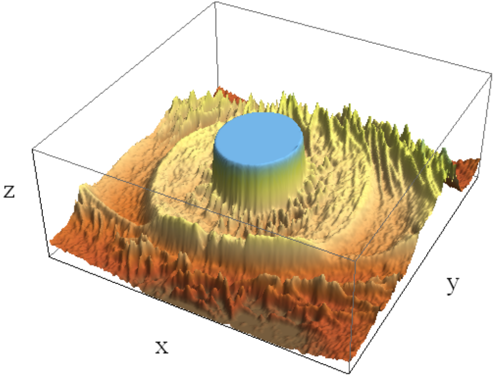


Image interpretation
and its consequences.

Iris image segmentation – quick look

The interesting regions of the image:
the highest one which corresponds to the pupil

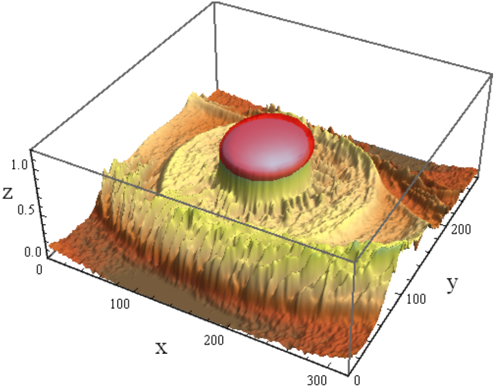


Image interpretation
and its consequences.

Iris image segmentation – quick look

The interesting regions of the image:
the highest one which corresponds to the pupil and **the region below and around the pupil** – the iris region.
If we find them then the iris is localized.

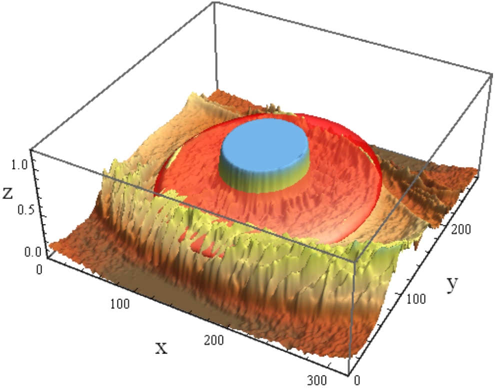


Image interpretation
and its consequences.

Outline

Cross Entropy Clustering (CEC)

The cross entropy clustering (CEC) is a clustering method, which was recently developed with the use of information theory. The main advantage of CEC is that it automatically reduces unnecessary clusters while combining the speed and simplicity of k-means with the ability to use various Gaussian mixture models.

Cross Entropy Clustering (CEC)

The cross entropy clustering (CEC) is a clustering method, which was recently developed with the use of information theory. The main advantage of CEC is that it automatically reduces unnecessary clusters while combining the speed and simplicity of k-means with the ability to use various Gaussian mixture models.

Implementation

- in Java: https://github.com/kmisztal/CEC,
- in Project R: package CEC,
- in Project R: package GMUM.r: http://gmum.ii.uj.edu.pl.

CEC, part I

The general idea of CEC relies on:

- finding the splitting of a set $U \subset \mathbb{R}^N$ into pairwise disjoint sets $U_1, \ldots, U_k$,
- for each U_i we want to choose the best describing it density f_i,
- the f_i is selected as the standard Gaussian density in $\mathbb{R}^d$ which is defined by

$$N(m, \Sigma) : x \to \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}} \exp(-\frac{1}{2}\|x - m\|_\Sigma^2),$$

where $\|x - m\|_\Sigma$ – Mahalanobis norm.

CEC, part II

- In fact we compare two densities:
  - empirical – uniform density on U_i (denoted by $\mathcal{U}_i$),
  - theoretical – density f_i.
- The comparison of two probabilities is done by the cross entropy according to

$$H^{\times}(\mathcal{U}_i \| f_i) = -\int_S \mathcal{U}_i(x) \ln f_i(x) dx.$$

  the first argument ($\mathcal{U}_i$) is treated as the "target" probability distribution, and the second (f_i) as the estimated one for which an evaluation is attemped how well it "fits" the target.
- Moreover, the cross entropy corresponds to the theoretical code-length of compression, in our case, of $\mathcal{U}_i$-randomly chosen element of $\mathbb{R}^N$ with the code optimized for density f_i.

CEC, part III

- In general case we would specify just the density subfamilies $\mathcal{F}_i$ and tray to find the optimal density $f_i \in \mathcal{F}_i$.
- Thus, the mean code-length for splitting $U_1, \dots, U_n$ described by $\mathcal{F}_1, \dots, \mathcal{F}_n$ equals

$$E_\mu(U_1, \mathcal{F}_1; \dots; U_n, \mathcal{F}_n) := \sum_{i=1}^{n} \mu(U_i) \cdot (-\ln(\mu(U_i)) + H^\times(\mathcal{U}_i \| \mathcal{F}_i)),$$

  where

$$H^\times(\mathcal{U}_i \| \mathcal{F}_i) = \inf_{f \in \mathcal{F}_i} H^\times(\mathcal{U}_i \| f).$$

- $-\ln(\mu(U_i))$ in the above formula corresponds to the memory needed for identify algorithm which is used for coding the element $x \in U_i$.

CEC, part IV

- The goal of CEC is to give spliting of set U, such that

$$E_\mu(U_1, \mathcal{F}_1; \ldots; U_n, \mathcal{F}_n)$$

  is minimal.
- Namely, for given density families $\mathcal{F}_1, \ldots, \mathcal{F}_n$ we are looking for proper splitting $U_1, \ldots, U_n$ of the given set U.
- As a result we get following estimation

$$U \sim \max(p_1 f_1, \ldots, p_k f_k),$$

  where f_i belong to given density families $\mathcal{F}_i$.

CEC – optimal number of clusters

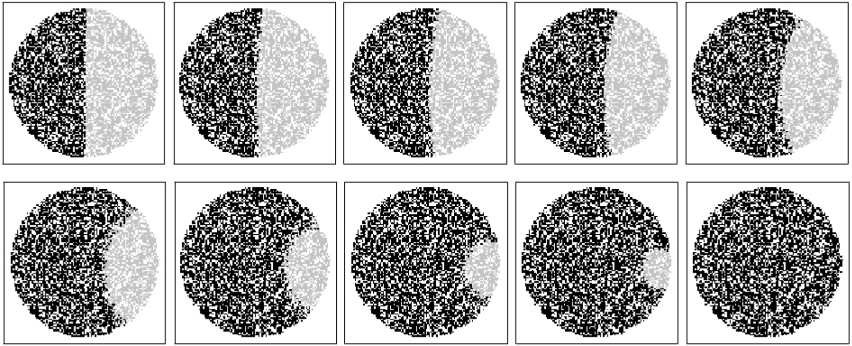
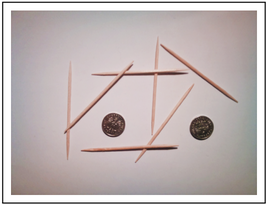
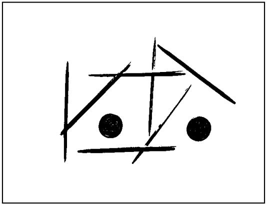


Figure: The step-by-step view of clusters reduction in the case of a disc-like set for the Spherical CEC – the data was divided initially into two almost equal parts.

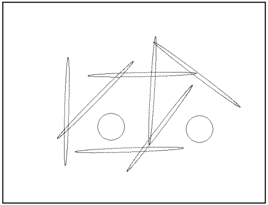
CEC – detection and recognition



(a) matches and coins



(b) binaryzation



(c) detected objects

Various patterns of the image can be distinguished, for example multiple types of objects can be detected simultaneously, e.g. the search for matches (Gaussian with specified covariance matrix) and coins (spherical Gaussian with fixed radius) is possible at the same time – compare with [Tabor, Misztal].

CEC – issues

In general, we have to solve two issues:

- **calculation:** how to calculate

$$H^{\times}(\mathcal{U}_i \| \mathcal{F}_i)$$

  for given density family $\mathcal{F}_i$,

CEC – issues

In general, we have to solve two issues:

- **calculation:** how to calculate

$$H^{\times}(\mathcal{U}_i \| \mathcal{F}_i)$$

  for given density family $\mathcal{F}_i$,
- **design:** how to chose the correct family $\mathcal{F}_i$ which meets our expectations.

CEC – design model

Theorem ([Tabor and Spurek, 2014])

Let μ be a discrete or continuous probability measure in $\mathbb{R}^N$ with well-defined mean and covariance matrix given by

$$\mathrm{m}(\mu) := \int x d\mu(x), \qquad \Sigma(\mu) := \int (x - \mathrm{m}(\mu))(x - \mathrm{m}(\mu))^T d\mu(x).$$

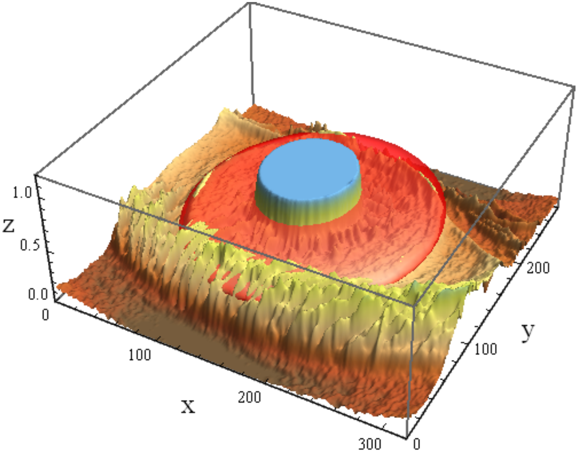
Let a fixed positive-definite symmetric matrix Σ be given. Then

$$H^{\times}(\mu \| \mathcal{G}_{\Sigma}) = H^{\times}\left(\mu_{\mathcal{G}} \| \mathcal{N}(\mathrm{m}(\mu), \Sigma(\mu))\right),$$

where $\mu_{\mathcal{G}}$ denotes the probability measure with Gaussian density of the same mean and covariance as μ. Consequently,

$$H^{\times}(\mu \| \mathcal{G}_{\Sigma}) = \frac{N}{2} \ln(2\pi) + \frac{1}{2}\mathrm{tr}(\Sigma^{-1}\Sigma_{\mu}) + \frac{1}{2} \ln \det(\Sigma). \qquad (1)$$

CEC – design model



Intuitively the covariance matrix which realized infinitum of cross entropy in such case is given by

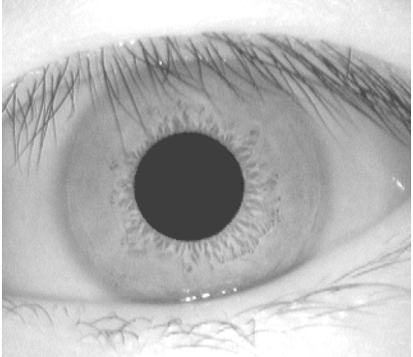
$$\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & \varepsilon \end{pmatrix}$$

Outline

Algorithm

1. Gaussian correction
2. Regression correction
3. CEC clustering
4. Result enhancement

Gaussian correction



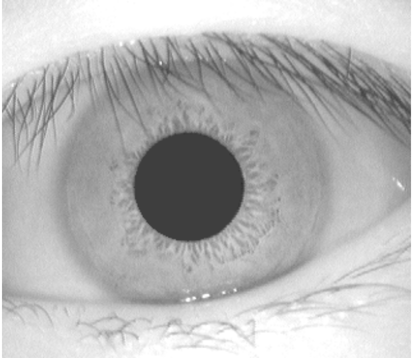
Original

Purpose:

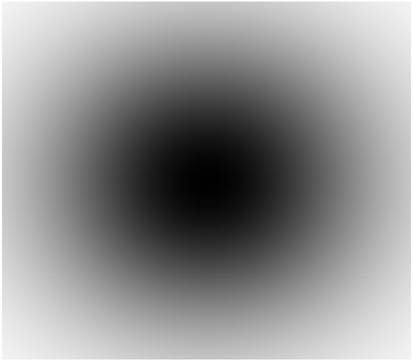
Reduce data size.

We can notice that the same regions of the skin are white or have a color very close to white – we want to increase those regions.

Gaussian correction

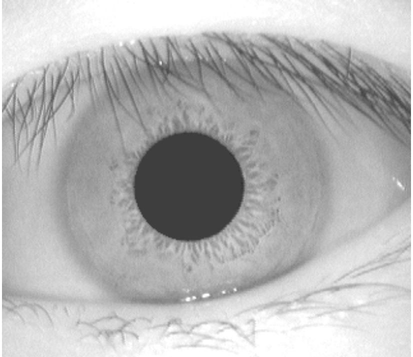


Original

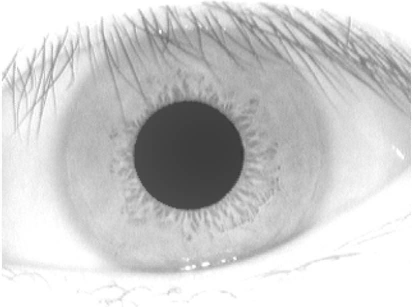


Mask
– the optimal Gaussian distribution for this image.

Gaussian correction

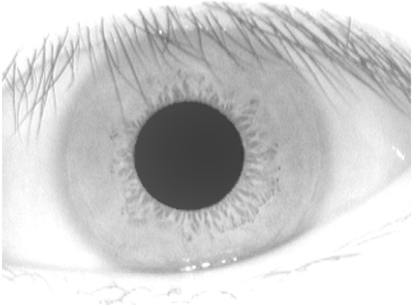


Original



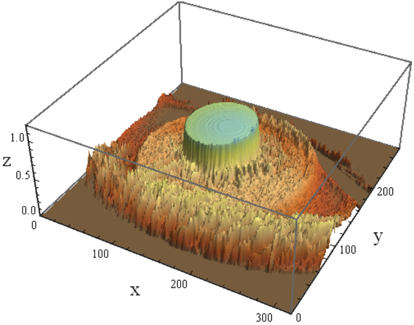
Original + Mask

Regression correction



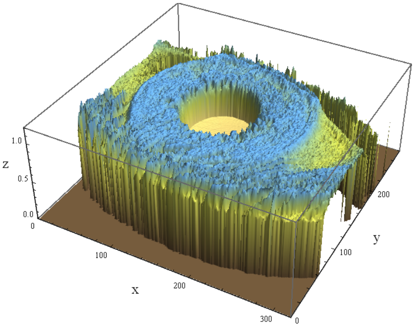
By the performing previous step we can bring the same abnormalities to image. Namely, the surface of the pupil can change, especially if its centre the pupil does not correspond to the mean of Gaussian distribution from the previous step. To fix such inconvenience we can calculate the optimal plane (using regression) and subtract it from the image.

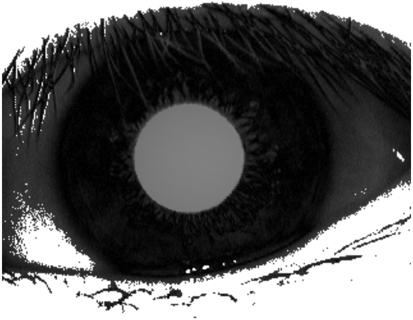
Regression correction



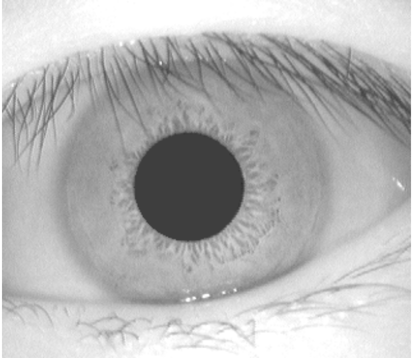
By the performing previous step we can bring the same abnormalities to image. Namely, the surface of the pupil can change, especially if its centre the pupil does not correspond to the mean of Gaussian distribution from the previous step. To fix such inconvenience we can calculate the optimal plane (using regression) and subtract it from the image.

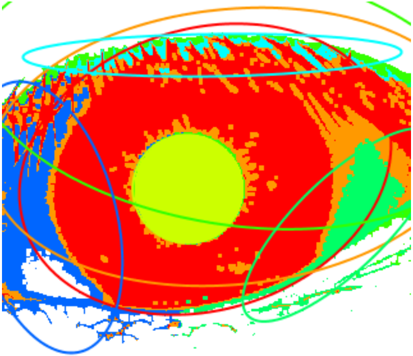
Regression correction





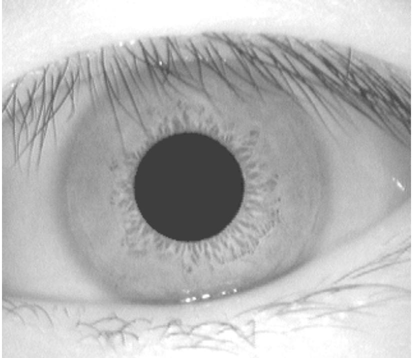
CEC clustering





The CEC was run with initial 20 clusters and end up with 7 clusters, the ε in covariance matrix was set to 10.

Result enhancement – selecting clusters



In our case we decided to pick up the two clusters with smallest empirical color variance.

Result enhancement – delete "outliers"

Theorem ([Misztal and Tabor, 2013])

Consider the uniform probability density on the ellipse $E \subset \mathbb{R}^2$ with covariance Σ_E. Then

$$E = \mathcal{B}_{\Sigma_E}(\mathrm{m}_E, 2). \tag{2}$$

$$\mathcal{B}_{\Sigma_E}(\mathrm{m}_E, r) = \{x \in \mathbb{R}^N \colon (x - \mathrm{m}_E)^T \Sigma_E^{-1} (x - \mathrm{m}_E) \leq r^2\}.$$

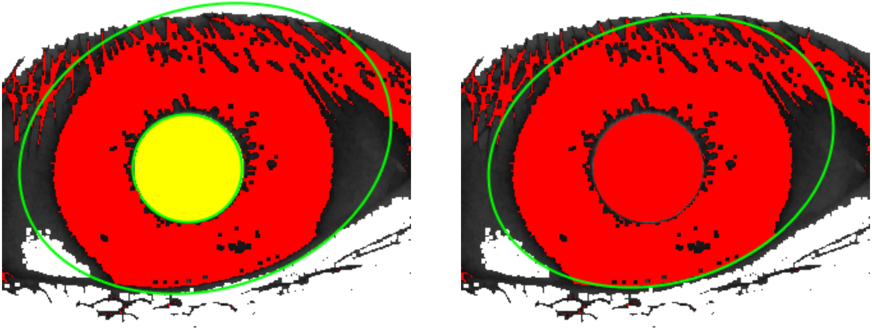
Result enhancement – Ellipse Shrinking

The Ellipse Shrinking algorithm finds iteratively the optimal ellipse describing the given set. We start with all point of the given set classified as members of optimal ellipse. Then we proceed with the following two steps:

1. compute the optimal ellipse for the current set, namely $\mathcal{B}_\Sigma(\mu, 2)$ (where Σ and μ are calculated for the current set);
2. from the optimal set delete points outside the optimal ball, namely, points which Mahalanobis distance from the mean of current set is greater than 2 (compare with Theorem 2).

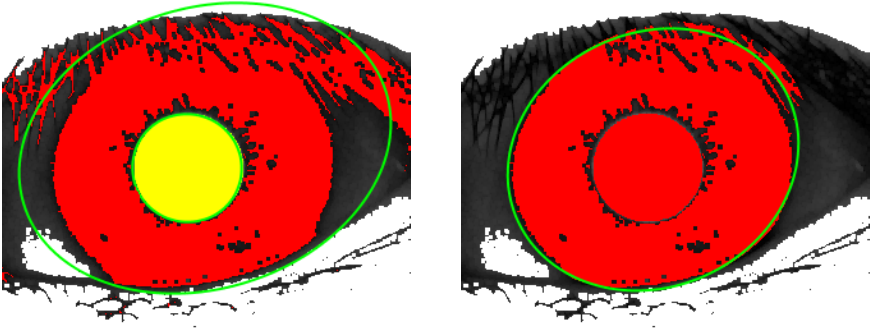
We repeat the above two steps until no points are removed in the second step.

Result enhancement – Ellipse Shrinking

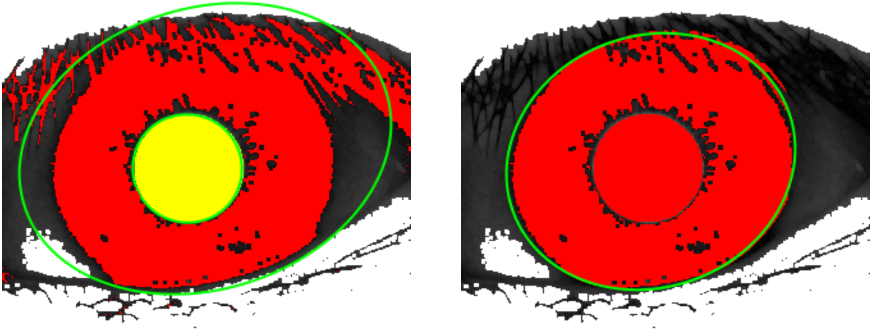


Result enhancement – Ellipse Shrinking

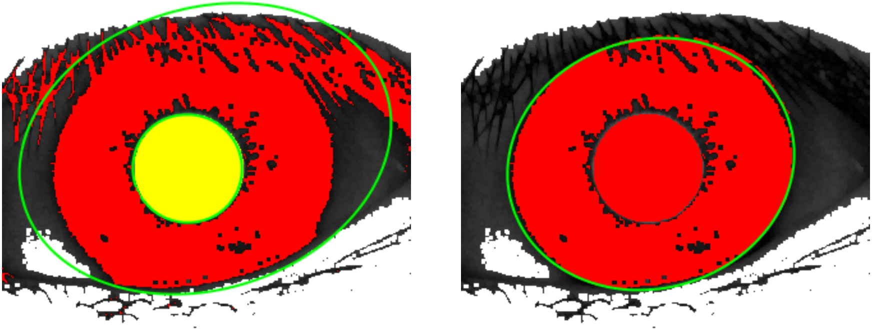
Result enhancement – Ellipse Shrinking



Result enhancement – Ellipse Shrinking

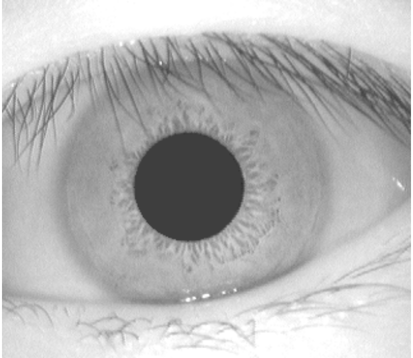


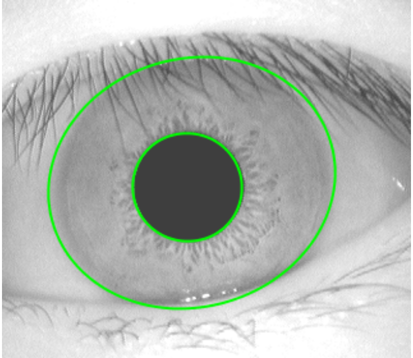
Result enhancement – Ellipse Shrinking



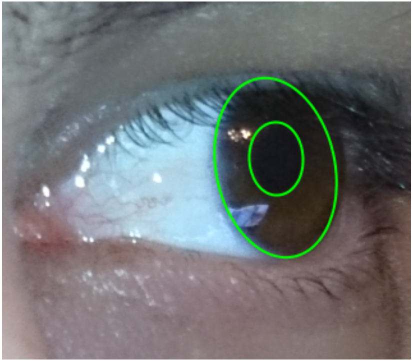
Outline

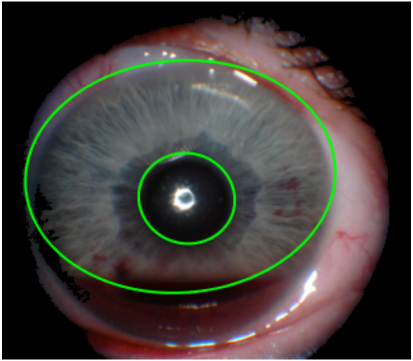
Result – ideal image





Result – non ideal images





The iris images can be affected by many factors that influence the shape, pattern or at least it may disturb the information collected from the iris, for example **off-angle or tilted** images (image on the left) or when the iris is **damaged by a disease** (image on the right).

Thank you for your kind attention.

Bibliography

Jain, A. K., Ross, A., and Prabhakar, S. (2004).
An introduction to biometric recognition.
Circuits and Systems for Video Technology, IEEE Transactions on, 14(1):4–20.

Misztal, K. and Tabor, J. (2013).
Mahalanobis distance-based algorithm for ellipse growing in iris preprocessing.
In *Computer Information Systems and Industrial Management*, volume 8104, pages 158–167. Lecture Notes in Computer Science, Springer Berlin Heidelberg.

Tabor, J. and Spurek, P. (2014).
Cross-entropy clustering.
Pattern Recognition, 47(9):3046–3059.