

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SOFTWARE FOR BIOMETRIC RECOGNITION OF A HUMAN EYE IRIS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LUKÁŠ MARUNIAK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SOFTWARE PRO BIOMETRICKÉ ROZPOZNÁVÁNÍ DUHOVKY LIDSKÉHO OKA

SOFTWARE FOR BIOMETRIC RECOGNITION OF A HUMAN EYE IRIS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LUKÁŠ MARUNIAK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MARTIN DRAHANSKÝ, Ph.D.

BRNO 2015

Abstrakt

V mojej práci sa zaoberám úlohou rozpoznania ľudskej dúhovky zo snímku za pomoci použitia evolučných algoritmov. Práca sa v úvode zaoberá otázkou biometrie, jej významom a základnými pojmami, s ktorými sa v texte neskor stretávam. Následne popisujem proces rozpoznania dúhovky ako aj teóriu evolučných algoritmov. V implementačnej časti popisujem návrh a implementáciu riešenia detekcie dúhovky za pomoci evolučných algoritmov, pričom kladiem doraz na detekciu okrajov zreničky a dúhovky.

Abstract

In my thesis, I focus on the task of recognizing human iris from an image. In the beginning, the work deals with a question of biometrics, its importance and basic concepts, which are necessary for use in following text. Subsequently process of human Iris detection is described together with theory of evolution algorithms. In the implementation part, is described the design of implemented solution, which uses evolution algorithms, where is emphasis on correct pupil and iris boundary detection.

Klíčová slova

biometria, rozpoznávanie dúhovky, extrakcia rysov, daugman, integro-diferenciálny operátor, model hrubého zarovnania, normalizácia, segmentácia, hammingova vzdialenosť, demodulácia, gáborov filter, vlnková transformácia, evolučné algoritmy, evolučné stratégie, genetické algoritmy

Keywords

biometry, iris recognition, feature extraction, daugman, integro-differential operator, Rubber sheet model, normalization, segmentation, Hamming distance, demodulation, Gabor wavelet, wavelet transformation, evolution algorithms, evolution strategies, genetic algorithms

Citace

Lukáš Maruniak: Software for Biometric Recognition of a Human Eye Iris, diplomová práce, Brno, FIT VUT v Brně, 2015

Software for Biometric Recognition of a Human Eye Iris

Prohlášení

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením p. Martina Drahanského

.....
Lukáš Maruniak
May 27, 2015

© Lukáš Maruniak, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Contents

1	Introduction	3
2	Theoretical basis	5
2.1	Biometrics	5
2.1.1	Identification and Verification	6
2.1.2	Biometric systems	6
2.1.3	Performance metrics	9
2.2	Human Iris	10
3	Segmentation	14
3.1	Overview	14
3.2	Review of available alghorithms	14
3.2.1	Hough transformation	14
3.2.2	Integro-differential operator	15
3.2.3	Discrete Circular Active Contours	16
3.2.4	Other Segmentation Methods	16
4	Normalisation	18
4.1	Overview	18
4.2	Review of available alghorithms	18
4.2.1	Daugman's Rubber Sheet Model	18
4.2.2	Non-linear Normalization Model	19
4.2.3	Other Normalization Methods	20
5	Feature Encoding	21
5.1	Overview	21
5.2	Wavelet encoding	21
5.3	Gabor Filters	21
5.4	Other methods	23
6	Matching	24
6.1	Hamming distance	24
6.2	Other matching alghorithms	25
7	Evolution algorithms	26
7.1	Evolution algorithms	26
7.1.1	Introduction into evolution algorithms	26
7.1.2	Darwin evolution	26

7.1.3	Evolution process simulation	27
7.1.4	Components of evolution algorithm	28
7.1.5	Application of evolution algorithm	28
7.1.6	Robustness of evolution algorithm	29
7.2	Genetic algorithms	30
7.2.1	Introduction	30
7.2.2	Canonical scheme of genetic algorithm activity	31
7.2.3	Selection	32
7.2.4	Crossover	35
7.2.5	Mutation	36
7.2.6	Population renewal	36
7.2.7	Schema theorem	37
7.2.8	Deceptive problems	37
7.3	Evolution strategies, ES	38
7.3.1	Introduction	38
7.3.2	Main characteristics in Evolution strategies	38
7.3.3	Mutation in evolution strategies	39
7.3.4	Crossover	40
7.3.5	Population renewal	40
7.3.6	Basic strategy 1+1	41
7.3.7	Autoevolution of control parameters	41
7.4	Multimodal functions	41
7.4.1	Techniques for multimodal functions solving	42
8	Implementation	44
8.1	Graphical user interface (GUI)	44
8.2	Convention solution	47
8.2.1	Pupil detection	47
8.2.2	Iris detection	49
8.3	Solution with the use of evolution algorithms	49
8.3.1	Pupil detection	50
8.3.2	Iris detection	51
9	Testing	53
9.1	Data	53
9.2	Results	53
9.2.1	Parameters setup	53
9.2.2	Results	54
10	Conclusion	56
10.1	Summary	56
A	Obsah CD	59

Chapter 1

Introduction

In my thesis I covered the topic of detection of a human iris. I focused on identification of human eye pupil and iris boundaries using evolution algorithms.

The detection of human iris is very powerful and reliable person identification method. It can be used in many commercial sectors to protect malicious access, identify right persons, or other not expected use, like it was done in the year 2002, when Afghan girl was identified by her Iris Patterns by John Daugmans Iris recognition algorithm. ¹.

Currently there exists lot of convention solutions of human iris detection and recognition. These algorithms detects pupil and iris boundary with relative high accuracy (approx. 99%) and in reasonable amount of time. [23]

However, when we focus on area of evolution algorithms, there are not so many reliable methods as it is in area of convention solutions. Therefore my main motivation and aim of this thesis was to cover this topic and try to answer the question, whether soft computing and especially evolution algorithms, can acquire similar results in acceptable amount of time as it is by using convention solutions.

In the beginning of my thesis, in chapter **Theoretical basis** I cover the necessary theoretical background, which is core for understanding the whole process of iris recognition.

In section **Biometrics** I specify the meaning of biometrics in today's world, the field of interest in biometrics, and main components used in other part of thesis.

In section **Human Iris** I briefly describe the main parts of human Iris, its meaning and unique patterns which are used in detection.

In next chapters I focus on individual parts of the human iris recognition process. These are **Segmentation**, **Normalisation**, **Feature Encoding** and **Matching**.

Every of these chapters contains explanation of this process, review of available literature, algorithms and it's description.

In chapter **Evolution algorithms** I describe in detail the area of evolution algorithms. I focused mainly on two sectors of evolution algorithms, namely **Genetic algorithms** and **Evo-**

¹<http://www.cl.cam.ac.uk/jgd1000/afghan.html>

lution strategies, ES which are used in evolution algorithm design used in eye iris segmentation. Also in section **Multimodal functions** i'm describing the methods of solving multimodal functions, which are causing the evolution algorithm stuck in local minimum.

In chapter **Implementation** I'm describing the process of solution implementation and its three main parts, namely **Graphical user interface (GUI)**, **Convention solution** and **Solution with the use of evolution algorithms**

In the chapter **Testing** I'm describing the test results and how the tests were performed. Also I'm briefly describing the database of which tested images were acquired.

In the chapter **Conclusion** I'm trying to summarize the recognition results and make conclusion with emphasis on possible future work.

Chapter 2

Theoretical basis

2.1 Biometrics

Biometrics is the science and technology of measuring and analysing biological data. In information technology, biometrics refers to technologies that measure and analyse human body characteristics, such as DNA, fingerprints, eye retinas and irises, voice patterns, facial patterns, hand measurements and other features for authentication purposes.[3]

The advantages of biometrics are:

- highly resistant to fraud
- increases safety
- can't be easily removed or transferred
- can't be lost
- increases comfort

Disadvantages are:

- output is matching score
- can't be revoked
- biometric system itself could be countervail-able
- doesn't keep privacy

The reason why is biometrics so difficult, is in processing of biological information. This process has to be resistant to interclass and intraclass variability, segmentation problems, performance issues, the uniqueness of processed biological feature, fusion of several biometric features etc.

2.1.1 Identification and Verification

People automatically recognize faces, body shape, voice, signature etc. This recognition is done by subliminal process in our brain, which is developed through our childhood.

The recognition of other person is based on unique identity of a human being.

Identity is unique characteristic of every human. However it's necessary to distinguish between physical and digital identity. For one individual, there exists only one physical identity. On the other hand, one individual can have many digital identities (mail accounts etc.).

Identity is related to two processes - Identification and Verification.

Identification detects the identity of a given object, in our case some human. In situation, when person gives his/her own biometric information, but not his/her own Identity to system, system compares the input with all records from database and in output returns whether the given input matches some record in database, or not.

This process is very time-consuming, especially when it's used in large systems, which contain many registered persons. In this case, the database could be divided into subcategories. For example the database of fingerprints could be divided into several subclasses and the search algorithm could run only against specific subclass for a given input. Identification is also called 1:MANY comparison. The example of the system for Identification could be dactyloscopic system(AFIS), database of refugees etc.

Different approach is **Verification**. In this case the user provides to the system his/her own digital identity and then is his/her own biometric data searched across the database. If the record doesn't exist, the access is refused.

If the record is found, then is compared. If both records match to some degree, specified by threshold value, then system consider both identities as equal.

Verification is also called 1:1 comparison, because there are compared single data from input with single data from a database. The example of such system could be access restriction system in banks, or e-mail services.

Sometimes we can encounter the term authentication. This term is used in access control systems. By authentication the system confirms the authenticity (credibility) of given person. The decision about person credibility is running mostly against some threshold, which is computed by the system.

From a certain point of view, biometrics offers best level of security and comfort, i.e. who we are. The idea is, that we are on our own, the holders of identification key.

Despite all, there is still a chance that our own identity will be misused. For example, we leave our fingerprints all over the places, and also our photography can be used to overcome 2D face-recognition system. The misuse of identity is called **Identity theft** and it's common problem all over the world.

2.1.2 Biometric systems

The biometric system consists of several parts which are shown in Figure 2.1.

In the enrolment sub-process, when the user use biometric system for first time, biomet-

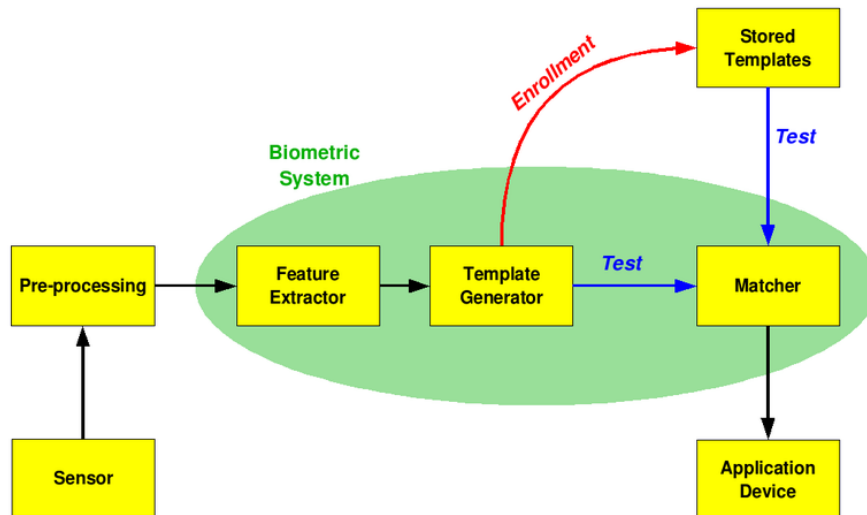


Figure 2.1: Biometric system diagram [1]

ric information from an individual is captured and stored. In subsequent uses, biometric information is detected and compared with the information stored at the time of enrollment. Note that it is crucial that storage and retrieval of such systems themselves are secure if the biometric system is robust.

Sensor is an interface between the real world and the system. It has to acquire all the necessary data. Most of the time it is an image acquisition system, but it vary according to the characteristics desired.

In the pre-processing have to be removed artefacts from the sensor, to enhance the input (e.g. removing background noise, reflection from camera etc.), to use some of many ways of normalization, etc. In the feature extraction process, there has to be implemented a proper way to extract only necessary information from input.

This is an important step, because the correct features need to be extracted in the optimal way. A vector of numbers or an image with particular properties is used to create a template. A template is a synthesis of the relevant characteristics extracted from the source. Elements of the biometric measurement that are not used in the comparison algorithm are discarded in the template to reduce the filesize and to protect the identity of the enrollee.

If enrollment is being performed, the template is simply stored in some place (on a card or within a database or both). If a matching phase is being performed, the obtained template is passed to a matcher that compares it with other existing templates, estimating the distance between them using any algorithm (e.g. Hamming distance).

The matching program will analyze the template with the input. This will be then output for any specified use or purpose (e.g. entrance in a restricted area). Selection of biometrics in any practical application depending upon the characteristic measurements and user requirements. [1]

Every biometric system can have its weak spot.

As it can be seen on Figure 2.2, right on the input (type 1) could be the sensor manipulated by incorrect biometric feature(e.g. artificial fingertip). The communication between sensor and feature extractor (type 2) can attacked using old data replication method. Also, feature

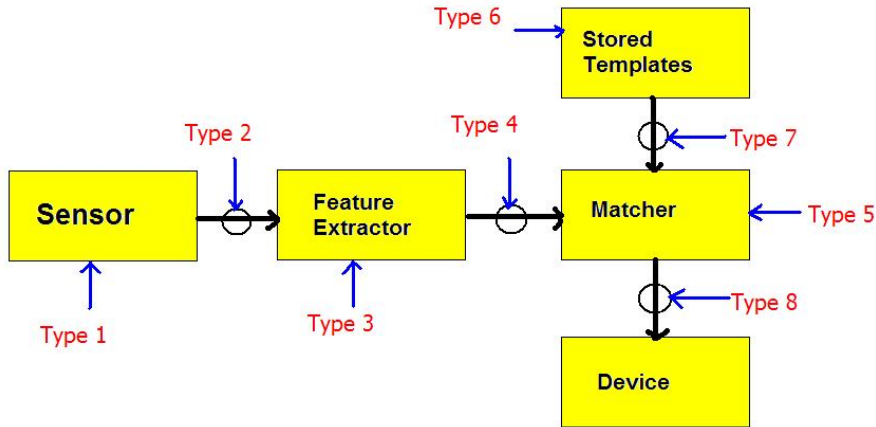


Figure 2.2: Biometric attacks[2]

extractor (type 3) can be attacked by injecting Trojan horse program into its code. Real features can be replaced by synthetic features and modified in database of stored templates (type 4). Also Matchers (Type 5) program can be modified. The communication between database(type 6) can be blocked, or the record in database replaced by false one. And last but not least, the results, which are transferred into application could be overridden (type 7).

The biometric features can be divided into two categories: physical and behavioral. Physical features are: fingerprint, face, eye iris, eye retina, hand geometry, palm etc. Behavioral features are: voice, face gestures, signature, keystroke dynamic, walk. By physical features, there is one stable characteristic of biometric feature(e.g. fingerprint). This feature is always available and it's not easily influenced by different circumstances. The method of physical features analysis is called static method. Behavioral features are connected with some user action. The method of behavioral features analysis is called dynamic method.

In context of biometric features, we distinguish unimodal and multimodal biometric system. Unimodal biometric system use only one biometric characteristic. Unimodal systems are highly used in commercial environment. Multimodal biometric system, on the other hand, uses more characteristics of one biometric feature (e.g. static and dynamic attributes of signature) or uses more biometric features (e.g. facial recognition with fingerprint recognition). These systems have increased reliability, but also increased acquisition costs.

Attributes of biometric features belong to very important criteria when we are deciding which specific system we should use. Base attributes are:

- Universality - means that every person using a system should possess the trait
- Uniqueness - means the trait should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another
- Permanence - relates to the manner in which a trait varies over time. More specifically,

a trait with 'good' permanence will be reasonably invariant over time with respect to the specific matching algorithm

- Measurability - (collectability) relates to the ease of acquisition or measurement of the trait. In addition, acquired data should be in a form that permits subsequent processing and extraction of the relevant feature sets
- Performance - relates to the accuracy, speed, and robustness of the used technology
- Acceptability - relates to how well individuals in the relevant population accept the technology such that they are willing to have their biometric trait captured and assessed
- Circumvention - relates to the ease with which a trait might be imitated using an artifact or substitute

2.1.3 Performance metrics

Overall performance of the biometric system is given by properties like universality, precision, speed and memory requirements. Biometric systems aren't perfect, they can accept attacker (false accept), or reject valid user (false reject).

By data processing (e.g. filtration) we are passing through extraction process, when significant features are extracted from input data. This extracted sample is then compared with template and the result is **matching score**, i.e. degree of consensus.

The comparison inside biometric system is based on a threshold, when for given input and threshold value, systems accept user, when match score is above threshold, or reject user otherwise. Therefore biometric system may come to the following error states:

- Two samples from two different persons are classified as equal, what is called **False Match**.
- Two samples from same person are classified as mismatched, what is called **False Non-Match**

For proper performance measurement, there are derived following performance metrics for biometric systems:

False accept rate or false match rate (FAR or FMR)

The probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted. In case of similarity scale, if the person is imposter in real, but the matching score is higher than the threshold, then he is treated as genuine that increase the FAR and hence performance also depends upon the selection of threshold value.

False reject rate or false non-match rate (FRR or FNMR)

The probability that the system fails to detect a match between the input pattern and a

matching template in the database. It measures the percent of valid inputs which are incorrectly rejected.

Receiver operating characteristic or relative operating characteristic (ROC)

The ROC plot is a visual characterization of the trade-off between the FAR and the FRR. In general, the matching algorithm performs a decision based on a threshold which determines how close to a template the input needs to be for it to be considered a match. If the threshold is reduced, there will be fewer false non-matches but more false accepts. Correspondingly, a higher threshold will reduce the FAR but increase the FRR.

Equal error rate or crossover error rate (EER or CER)

The rate at which both accept and reject errors are equal. The value of the EER can be easily obtained from the ROC curve. The EER is a quick way to compare the accuracy of devices with different ROC curves. In general, the device with the lowest EER is most accurate.

Failure to enroll rate (FTE or FER)

The rate at which attempts to create a template from an input is unsuccessful. This is most commonly caused by low quality inputs.

Failure to capture(acquire) rate (FTC or FTA)

Within automatic systems, the probability that the system fails to detect a biometric input when presented correctly.

2.2 Human Iris

The iris is a protected internal organ of the eye, located behind the cornea and the aqueous humour, but in front of the lens. A visible property of the iris and the fingerprint is the random morphogenesis of their minutiae. The phenotypic expression even of two irises with the same genetic genotype (as in identical twins, or the pair possessed by one individual) have uncorrelated minutiae. The iris texture has no genetic penetrance in the expression and is chaotic. In these respects the uniqueness of every iris parallels the uniqueness of every Fingerprint, common genotype or not.

The most important function of the iris is controlling the size of the pupil. Illumination, which enters the pupil and falls on the retina of the eye, is controlled by muscles in the iris. They regulate the size of the pupil and this is what permits the iris to control the amount of light entering the pupil. The change in the size results from involuntary reflexes and is not under conscious control. The tissue of the iris is soft and loosely woven and it is called stroma.

A section through the human iris is shown in Figure 2.4. In this figure, we can see the layers of the human iris. The layers of the iris have both ectodermal and mesodermal embryological origin. The visible one is the anterior layer, which bears the gaily-coloured relief and it is very lightly pigmented due to genetically determined density of melanin pigment granules. The invisible one is the posterior layer, which is very darkly pigmented, contrary to the anterior layer. The surface of this layer is finely radiantly and concentrically furrowed with dark brown colour. Muscles and the vascularized stroma are found between these lay-

ers from back to front. The surface of the anterior layer is shown in Figure 2.5. Pigment frill is the boundary between the pupil and the human iris. It is a visible section of the posterior layer and looks like a curling edge of the pupil. The whole anterior layer consists of the pupillary area and the ciliary area and their boundary is called collarette. The ciliary area is divided into the inner area, which is relatively smooth and bears radial furrows, the middle area, heavily furrowed in all directions and with pigment piles on the ridges, and the outer marginal area bearing numerous periphery crypts. [18]

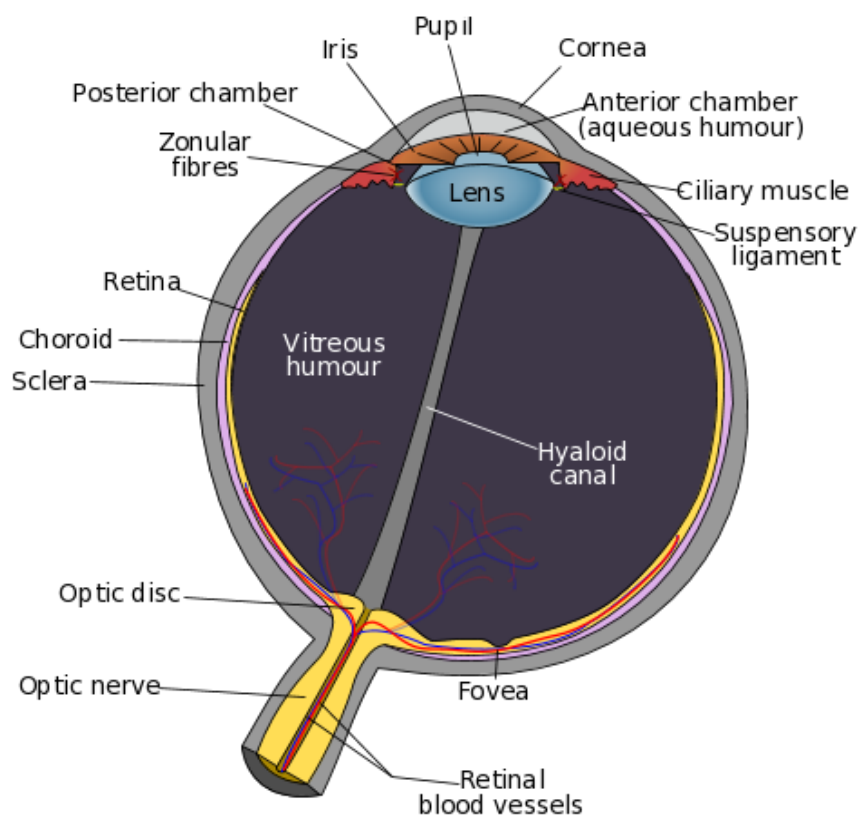


Figure 2.3: Eye structure[4]

Among the pigment related features belong the crypts and the pigment spots. The crypts, in the are the areas in which the iris is relatively thin. They have very dark color due to dark color of the posterior layer. They appear near the collarette, or on the periphery of the iris. They look like sharply demarcated excavations. The pigment spots, are random concentrations of pigment cells in the visible surface of the iris and generally appear in the ciliary area. They are known as moles and freckles with nearly black color. Features controlling the size of the pupil are radial and concentric furrows. They are called contraction furrows and control the size of the pupil. Extending radially in relation to the center of the pupil are radial furrows. The typical radial furrows may begin near the pupil and extend through the collarette. The radial furrows are creased in the anterior layer of the

iris, from which loose tissue may bulge outward and this is what permits the iris to change the size of the pupil. The concentric furrows, are generally circular and concentric with the pupil. They typically appear in the ciliary area, near the periphery of the iris and permit to bulge the loose tissue outward in different direction than the radial furrows. Collarette, mentioned briefly above, is the boundary between the ciliary area and the pupillary area. It is a sinuous line, which forms an elevated ridge running parallel with the margin of the pupil. The collarette is the thickest part of the human iris. The human iris may have some of the rare anomalous visible features. Due to aging or trauma, atrophic areas may appear on the iris, resulting in a „moth-eaten“ texture.

Finally, the most striking visible feature of the eye is typically the pupil. The pupil may not be exactly circular in shape and its deviation from the circle is a visible characteristics. Centers of the iris and the pupil are different and they can differ from each other of about 20 %.

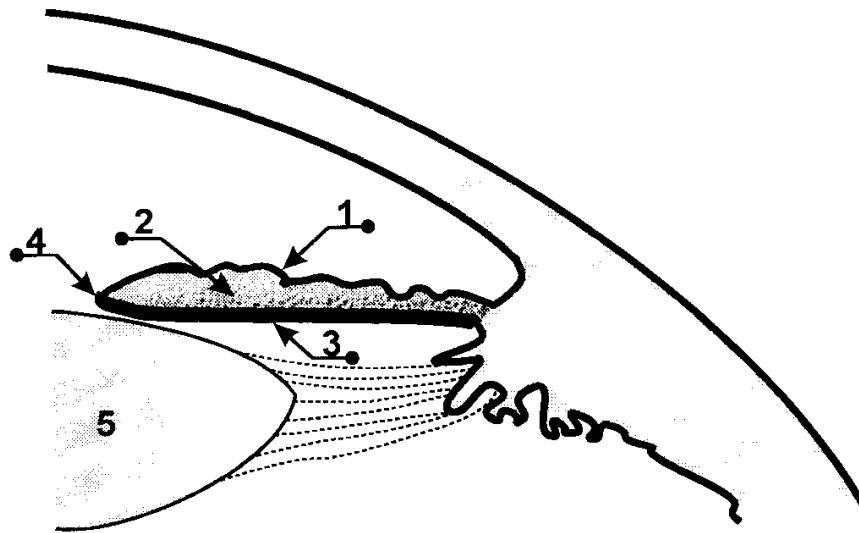


Figure 2.4: Section through the human iris (1-anterior layer, 2-stroma, 3-posterior layer, 4-pigment frill, 5-lens)[18]

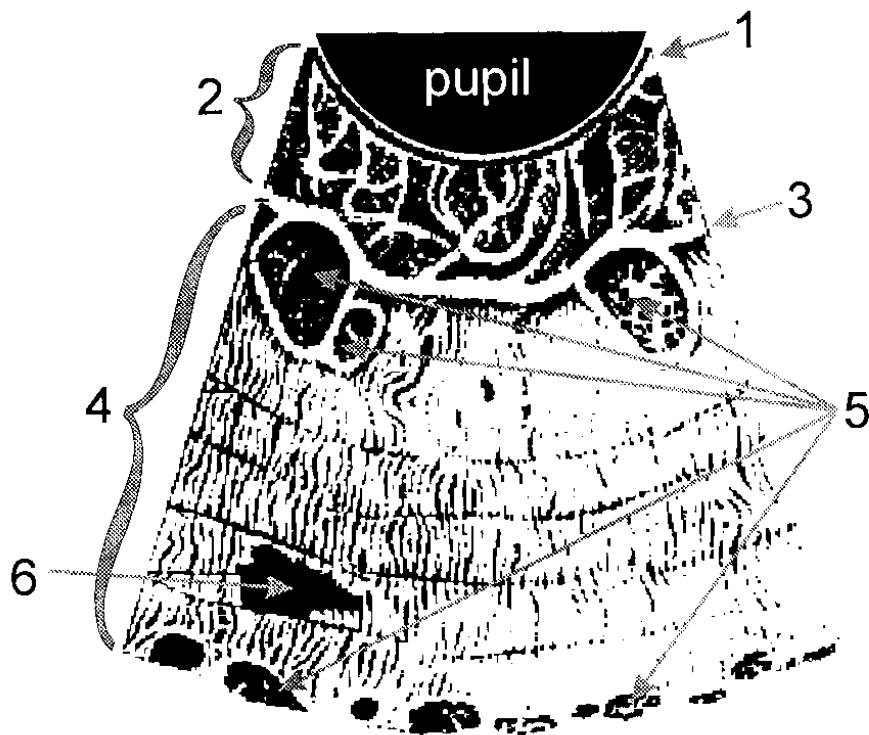


Figure 2.5: Detailed view of anterior surface of the human iris (1-pigment frill, 2-pupillary area, 3-collarette, 4-ciliary area, 5-crypts, 6-pigment spot)[18]

Chapter 3

Segmentation

3.1 Overview

The first subprocess in iris recognition process is segmentation.

Segmentation isolates the actual iris region in a digital eye image.

The iris region can be distinguished easily, by human eye, because the transition between pupil and iris and transition between iris and sclera, contains relatively high contrast.

In computer vision we need to extract our region of interest with more complex image analysis. It has to deal with eyebrows and small noise in the picture. Also it has to deal with fact, that circles, which represents pupil, or iris aren't cocentric. This whole process could be done with morphological methods in hand with working algorithms, which I describe in following sections.

The success of segmentation and therefore whole recognition process, highly depends on image quality. Images used for this thesis were taken in high quality, but contained minor reflections from camera in pupil area.

3.2 Review of available algorithms

3.2.1 Hough transformation

The Hough transformation is a standard computer vision algorithm that can be used to determine the parameters of simple geometric objects, such as lines and circles, present in an image. The circular Hough transformation can be employed to deduce the radius and centre coordinates of the pupil and iris regions.

Firstly, an edge map is generated by calculating the first derivatives of intensity values in an eye image and then thresholding the result. From the edge map, votes are cast in Hough space for the parameters of circles passing through each edge point. These parameters are the centre coordinates x_c and y_c , and the radius r , which are able to define any circle according to the equation:

$$x_c^2 + y_c^2 - r = 0 \tag{3.1}$$

A maximum point in the Hough space will correspond to the radius and centre coordinates of the circle best defined by the edge points.

There are a number of problems with the Hough transformation method. First of all, it requires threshold values to be chosen for edge detection, and this may result in critical edge points being removed, resulting in failure to detect circles/arcs. Secondly, the Hough transform is computationally intensive due to its 'brute-force' approach, and thus may not be suitable for real time applications. [17]

3.2.2 Integro-differential operator

Currently the most used algorithm for Iris segmentation is Integro-differential operator implemented by John Daugman.[19]

This operator is defined as:

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial x} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right| \quad (3.2)$$

Integro-differential operator

Where $I(x; y)$ is an image containing an eye. The operator searches over the image domain $(x; y)$ for the maximum in the blurred partial derivative with respect to increasing radius r , of the normalized contour integral of $I(x, y)$ along a circular arc ds of radius r and center coordinates x_0, y_0 .

The symbol $*$ denotes convolution and $G_\sigma(r)$ is a smoothing function such as a Gaussian of scale σ . The complete operator behaves in effect as a circular edge detector, blurred at a scale set by σ , which searches iteratively for a maximum contour integral derivative with increasing radius at successively finer scales of analysis through the three parameter space of center coordinates and radius $(x_0; y_0; r_0)$ defining a path of contour integration.

The operator in equation 3.2 serves to find both the pupillary boundary and the outer (limbus) boundary of the iris, although the initial search for the limbus also incorporates evidence of an interior pupil to improve its robustness since the limbic boundary itself usually has extremely soft contrast when long wavelength NIR illumination is used.

Once the coarse-to-fine iterative searches for both these boundaries have reached single pixel precision, then a similar approach to detecting curvilinear edges is used to localize both the upper and lower eyelid boundaries.

The path of contour integration in 3.2 is changed from circular to arcuate, with spline parameters fitted by standard statistical estimation methods to describe optimally the available evidence for each eyelid boundary.

The result of all these localization operations is the isolation of iris tissue from other image regions. [19]

The integro-differential can be seen as a variation of the Hough transform, since it too makes use of first derivatives of the image and performs a search to find geometric parameters. Since it works with raw derivative information, it does not suffer from the thresholding problems of the Hough transform. However, the algorithm can fail where there is noise in the eye image, such as from reflections, since it works only on a local scale. [17]

3.2.3 Discrete Circular Active Contours

The model detects pupil and limbus by activating and controlling the active contour using two defined forces: internal and external forces. The internal forces are responsible to expand the contour into a perfect polygon with a radius C_r larger than the contour average radius.

The internal forces are designed to expand the contour and keep it circular. The force model assumes that pupil and limbus are globally circular, rather than locally, to minimize the undesired deformations due to specular reflections and dark patches near the pupil boundary.

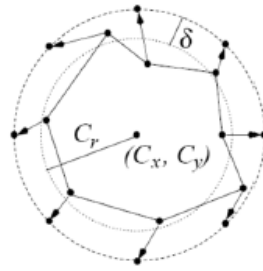


Figure 3.1: The internal forces of the Discrete Circular Active Contour [20]

The contour detection process of the model is based on the equilibrium of the defined internal forces with the external forces. The external forces are obtained from the grey level intensity values of the image and are designed to push the vertices inward. The final equilibrium is achieved when the average radius and center of the contour becomes the same as the one in m iterations ago.

3.2.4 Other Segmentation Methods

Other researchers use methods similar to the described segmentation methods. For instance, the iris localization proposed by Tisse et al. [21] is a combination of the Integro-differential and the Hough transform. The Hough transform is used for a quick guess of the pupil center and then the Integro-differential is used to accurately locate pupil and limbus using a smaller search space.

Lim et al. [22] localize pupil and limbus by providing an edge map of the intensity values of the image. The center of pupil is then chosen using a bisection method that passes perpendicular lines from every two points on the edge map. The center point is then obtained by voting the point that has the largest number of line crossovers. The pupil and limbus boundaries are then selected by increasing the radius of a virtual circle with the selected center point and choosing the two radii that have the maximum number of edge crosses by the virtual circle as the pupil and limbus radii . [20]

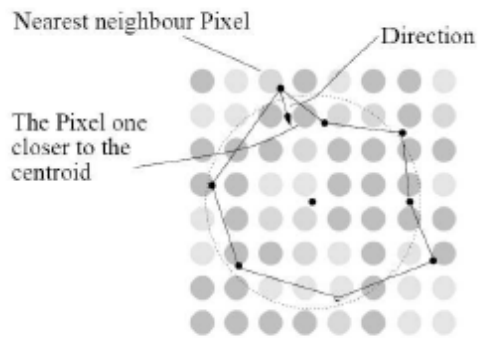


Figure 3.2: The external forces of the Discrete Circular Active Contour[20]

Chapter 4

Normalisation

4.1 Overview

Once the iris region is successfully segmented from an eye image, the next stage is to transform the iris region so that it has fixed dimensions in order to allow comparisons.

The dimensional inconsistencies among eye images are mainly due to the stretching of the iris caused by pupil dilation from varying levels of illumination.

Other sources of inconsistency include, varying imaging distance, rotation of the camera, head tilt, and rotation of the eye within the eye socket.

The normalisation process will produce iris regions, which have the same constant dimensions, so that two photographs of the same iris under different conditions will have characteristic features at the same spatial location. [17]

We must take in note, that pupil region is not always concentric within the iris region, and is usually slightly nasal. This must be taken into account if trying to normalise the 'doughnut' shaped iris region to have constant radius.

4.2 Review of available algorithms

4.2.1 Daugman's Rubber Sheet Model

Most used model for normalisation is Daugman's rubber sheet model. [19] This model remaps each pixel within iris region to a pair of polar coordinates (r, θ) where r is on interval $[0,1]$ and θ is angle $\langle 0, 2\pi \rangle$.

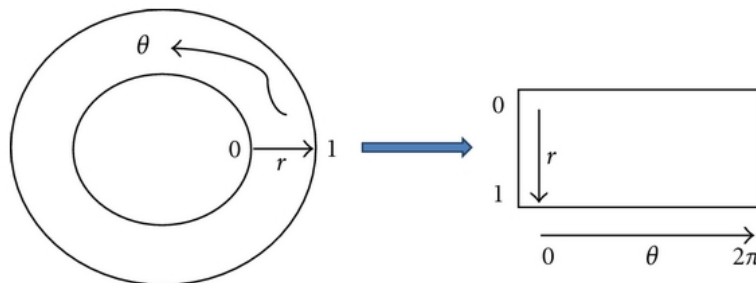


Figure 4.1: Daugman's rubber sheet model [19]

The remapping of the iris region from (x, y) Cartesian coordinates to the normalised non-concentric polar representation is modelled as $I(x(r, \theta), y(r, \theta)) \longrightarrow I(r, \theta)$ with

$$\begin{aligned}x(r, \theta) &= (1 - r)x_p(\theta) + rx_l(\theta) \\y(r, \theta) &= (1 - r)y_p(\theta) + ry_l(\theta)\end{aligned}\tag{4.1}$$

where $I(x, y)$ is the original image within segmented space, (x, y) are the original Cartesian coordinates, (r, θ) are the corresponding normalised polar coordinates, x_p, y_p and x_l, y_l are the coordinates of the pupil and iris boundaries along the θ direction.

The rubber sheet model takes into account pupil dilation and size inconsistencies in order to produce a normalised representation with constant dimensions. In this way the iris region is modelled as a flexible rubber sheet anchored at the iris boundary with the pupil centre as the reference point. Even though the homogenous rubber sheet model accounts for pupil dilation, imaging distance and non-concentric pupil displacement, it does not compensate for rotational inconsistencies. In the Daugman system, rotation is accounted for during matching by shifting the iris templates in the θ direction until two iris templates are aligned.

4.2.2 Non-linear Normalization Model

The unwrapping method proposed by Daugman assumes that iris patterns are linearly distributed in the radial direction, which allows the mapping procedure into the interval $[0, 1]$. The technique relies on two main factors:

- The image acquisition process adjusts the pupil size to a proper radius range by adjusting the illumination.
- The feature extraction process is locally applied to many different positions of the iris texture, which would compensate the local nonlinear variations.

The proposed non-linear normalization method, considers a nonlinear behavior of iris patterns due to changes of pupil size. In order to unwrap an iris region properly, a non-linear model and a linear normalization model are combined. The non-linear method, which is first applied to an iris image, is based on three assumptions:

- The pupil margin and iris root (which correspond to the inner and outer boundaries of the iris) are concentric circles.
- The margin of the pupil does not rotate significantly during pupil size changes.
- The pupil shape does not change and remain circular when pupil size changes.

The non-linear model is defined by virtual arcs, which are named „fibers“ following Wyatts work, that connect a point on the pupil border to a point on the limbus. The polar angle traversed by the arcs between these two points is $\pi/2$. The virtual arcs are defined

based on normalized pupil sizes to a fixed value using a predefined λ_{ref} , which is obtained by the mean of all λ values defined as $\lambda = r/R$ in the iris database. The r and R represent the radius of pupil and limbus respectively. The reference annular zone with ref is then linearly mapped into a fixed-size rectangle zone of $m \times n$ by equally sampling m points in each virtual concentric sampling circle with a fixed radial resolution.

It is concluded by the authors of the presented approach that the non-linear model still simplifies the real physiological mechanism of iris deformation and some more assumptions and approximations are required to support the model. The model is also believed to explicitly show the non-linear behavior of iris textures due to the improvements obtained in the experiments.[20]

4.2.3 Other Normalization Methods

Also exists a method very similar to the pseudo polar transform of Daugman. [19] In this method, after finding the center of pupil and the inner and outer boundaries of iris, the texture is transformed into polar coordinates with a fixed resolution. In the radial direction, the texture is normalized from the inner boundary to the outer boundary into 60 pixels which is fixed throughout all iris images.

The angular resolution is also fixed to a 0.8 degree over the 360 degree which produces 450 pixels in the angular direction.

Another normalization technique is also similar to Daugman's method with the difference that it is performed at the time of matching. The method is based on the diameter of the two matching irises. The ratio of the diameters are calculated and the diameter of irises are adjusted to have the same diameters. The number of samples is also fixed and it is set to a power-of-two integer in order to be suitable for the dyadic wavelet transform.

In addition, there has been some research on the pseudo polar transform in order to optimize its performance.

Chapter 5

Feature Encoding

5.1 Overview

In order to provide accurate recognition of individuals, the most discriminating information present in an iris pattern must be extracted. Only the significant features of the iris must be encoded so that comparisons between templates can be made. Most iris recognition systems make use of a band pass decomposition of the iris image to create a biometric template.

5.2 Wavelet encoding

Wavelets can be used to decompose the data in the iris region into components that appear at different resolutions. Wavelets have the advantage over traditional Fourier transform in that the frequency data is localised, allowing features which occur at the same position and resolution to be matched up. A number of wavelet filters, also called a bank of wavelets, is applied to the 2D iris region, one for each resolution with each wavelet a scaled version of some basis function. The output of applying the wavelets is then encoded in order to provide a compact and discriminating representation of the iris pattern. [17]

5.3 Gabor Filters

Gabor filters are able to provide optimum conjoint representation of a signal in space and spatial frequency. A Gabor filter is constructed by modulating a sine/cosine wave with a Gaussian. This is able to provide the optimum conjoint localisation in both space and frequency, since a sine wave is perfectly localised in frequency, but not localised in space. Modulation of the sine with a Gaussian provides localisation in space, though with loss of localisation in frequency. Decomposition of a signal is accomplished using a quadrature pair of Gabor filters, with a real part specified by a cosine modulated by a Gaussian, and an imaginary part specified by a sine modulated by a Gaussian. The real and imaginary filters are also known as the even symmetric and odd symmetric components respectively.[17]

The centre frequency of the filter is specified by the frequency of the sine/cosine wave, and the bandwidth of the filter is specified by the width of the Gaussian.

Daugman uses 2D versions of Gabor filters in order to encode iris pattern data. A 2D Gabor filter over the an image domain (x, y) is represented as

$$G(x, y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]} e^{-2\pi i[(u_0(x-x_0)+v_0(y-y_0))]} \quad (5.1)$$

where (x_0, y_0) specify position in the image, (α, β) specify the effective width and length, and (u_0, v_0) specify modulation, which has spatial frequency $\omega_0 = \sqrt{u_0^2 + v_0^2}$

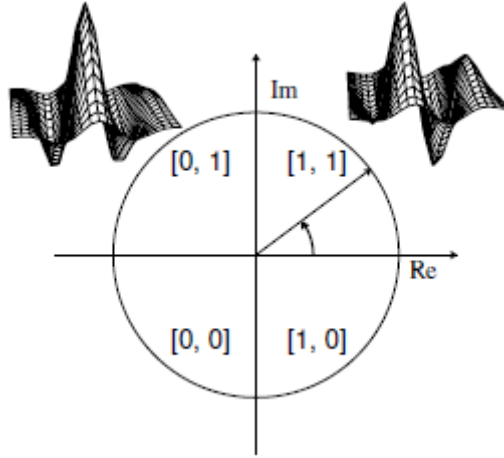


Figure 5.1: 2D Gabor wavelets[19]

In Figure 5.1 are shown 2D Gabor wavelets. Local regions of an iris are projected onto quadrature 2D Gabor wavelets, generating complex-valued coefficients whose real and imaginary parts specify the coordinates of phasor in the complex plane. The angle of each phasor is quantized to one of four quadrants, setting two bits of phase information. This process is repeated all across the iris with many wavelet sizes, frequencies, and orientations, to extract 2048 bits.

Taking only the phase will allow encoding of discriminating information in the iris, while discarding redundant information such as illumination, which is represented by the amplitude component.

These four levels are represented using two bits of data, so each pixel in the normalised iris pattern corresponds to two bits of data in the iris template. A total of 2048 bits are calculated for the template, and an equal number of masking bits are generated in order to mask out corrupted regions within the iris. This creates a compact 256-byte template, which allows for efficient storage and comparison of irises. The Daugman system takes polar coordinates for normalisation, therefore in polar form the filters are given as: [19]

$$H(r, \theta) = e^{-i\omega(\theta-\theta_0)} e^{-(r-r_0)^2/\alpha^2} e^{-i(\theta-\theta_0)^2/\beta^2} \quad (5.2)$$

where (α, β) are specify the effective width and length and (r_0, θ_0) specify the centre frequency of the filter.

The demodulation and phase quantisation process can be represented as [19]

$$h_{Re,Im} = \text{sgn}_{\{Re,Im\}} \iint_{\rho \phi} I(\rho, \phi) e^{-i\omega(\theta-\theta_0)} e^{-(r-r_0)^2/\alpha^2} e^{-i(\theta-\theta_0)^2/\beta^2} \rho d\rho d\phi \quad (5.3)$$

where $h_{Re,Im}$ can be regarded as a complex valued bit whose real and imaginary components are dependent on the sign of the 2D integral, and $I(\rho, \phi)$ is the raw iris image in a dimensionless polar coordinate system. [17]

5.4 Other methods

Currently, there exists more feature extracting methods, mostly based on wavelet filtering. These are using different wavelet filtering from Gabor wavelet filter, or some slightly modification of it.

For example I can mention Log-Gabor Filters, Haar wavelet, which uses only 87 bits instead of 2048, Laplacian of Gaussian Filters[10], Zero-crossings of the 1D wavelet etc.[10]

For detailed understanding mentioned methods, please refer to adequate literature [9]

Chapter 6

Matching

We need to have some measurement for comparing template with input. The output from feature extracting process is stored in byte code and so it's the template. Therefore we need metric, which is able to measure similarities between two byte codes.

6.1 Hamming distance

The Hamming distance [11] determines how many bits, compare to whole file are different. So when we are comparing two identical iris codes, the Hamming distance should be equal to 0.

In comparing the bit patterns X and Y , the Hamming distance, HD , is defined as the sum of disagreeing bits (sum of the XOR between X and Y) over N , the total number of bits in the bit pattern.

$$HD = \frac{1}{N} \sum_{j=1}^N X_j \oplus Y_j \quad (6.1)$$

Since an individual iris region contains features with high degrees of freedom, each iris region will produce a bit-pattern which is independent to that produced by another iris, on the other hand, two iris codes produced from the same iris will be highly correlated.

If two bits patterns are completely independent, such as iris templates generated from different irises, the Hamming distance between the two patterns should equal 0.5. This occurs because independence implies the two bit patterns will be totally random, so there is 0.5 chance of setting any bit to 1, and vice versa. Therefore, half of the bits will agree and half will disagree between the two patterns. If two patterns are derived from the same iris, the Hamming distance between them will be close to 0.0, since they are highly correlated and the bits should agree between the two iris codes.

The Hamming distance is the matching metric employed by Daugman, and calculation of the Hamming distance is taken only with bits that are generated from the actual iris region. [17]

6.2 Other matching algorithms

There exist other techniques for proper measurement similarities, or differences between two iris templates. For purpose of this thesis are these not relevant, as Hamming distance is used widely in almost every work, which is dealing with human iris recognition.

These methods are for example: Normalised Correlation, Weighted euclidean distance etc. For more detailed information please refer to adequate literature.[\[12\]](#)

Chapter 7

Evolution algorithms

7.1 Evolution algorithms

7.1.1 Introduction into evolution algorithms

Evolution algorithms is common expression for class of modern mathematic procedures, which uses models of evolution processes in nature. All of these models have common features. These mostly work with set of feasible outcomes of given task/problem. The results are being continuously improved by preferring better solutions, which came out of original solutions by applying combination and mutation, which eliminates weak/worse solutions.

7.1.2 Darwin evolution

With first evolution theory came Jean Baptiste de Lamarck. He was explaining evolution as the ability of living organism to change, specifically by continuous process of evolving and inheritance, when structures which are being used are for next generations strengthen, and those which are not are weaken. According to todays findings most of the evolution changes was not reproduced by this mechanism, however, there exists indirect possibility, that evolution was evolved by applying secondary mutation.

According to Darwin theory, which is based on natural selection, has the higher number of newborn individuals, than the environment can support, direct impact on creation of concurrency and „survival fight“.

In the population will then survive individuals with advantageous deviations acquired during the evolution process. According to this fact, only the fittest individuals will survive. Therefore given biological kind will adapt on its environment and also to its changes.

Basic signs of classic darwinism are:

- Populations are genetically variable, mutability is random in respect of environment
- Populations have infinite possibility to grow, but food and spatial resources are limited and therefore only part of population grows into individuals capable of reproduction - among the individuals has to exist „survival fight“
- Only fittest individuals are able to beget offsprings and therefore move their unique genetic dispositions at higher rate into following generations - the presence of suitable properties is continuously growing.

- Thanks to this natural selection have species adapted to environment and by longterm effect of selection is possible to explain all evolution

In other words we can say, that biologic evolution, the progressive change of genetic content of population among many generations, contains these three main components:

1. **Natural selection**, process in which high fitness individuals enters the reproduction process with higher probability than other individuals
2. **Random genetic drift**, in which random events in life of individuals influence whole population. Such events are for example random mutation of genetic material, or random death of individual with high fitness value before he/she could enter the reproduction process. Random events of genetic drift are significant especially for small population
3. **Reproduction process**, within which are offsprings created from their parents. Offsprings genetic information is being formed by mutual change of parents genetic information. Most often proceeds this process as combination of random parts of parents chromosome, from which is formed new genetic information of newborn individual(called sexual reproduction)

In biology is the „fitness“ defined as the ability to survive and reproduce in a given environment and given population. It can be understood as holistic property genotype - genetic information materially realized by individuals chromosome in population.

Missing link in Darwin’s theory added Gregor Mendel, who published his own work in the same year as Darwin’s „Origin of species“ was published. Genetic determinism (Neodarwinism) is derived from the combination of Darwinism and Mendel’s genetics. According to him, progress of evolution runs through natural selection and random genetic mutations.

7.1.3 Evolution process simulation

Evolution algorithms are based on metaphor of evolution. Solving a task is transferred to the evolution process of the population of the randomly generated solutions. Every solution is encoded into a string of symbols (parameters) and evaluated by fitness function that expresses the quality of the solution - higher value means that given solution is more perspective and therefore enters the reproduction process more often. Population of solutions is commonly called a population of individuals or chromosomes. The reproduction process is based on two driving forces:

- Various operators of crossover and mutation, which brings in population diversity
- Selection, which prioritize better individuals

The combination of variation and selection generally contributes to improving the fitness function of individuals in the newly formed population. In the process of crossing individuals, as well as in living nature, the new individuals/offsprings are obtained by crossing parental individuals. All of the components are stochastic - e.g. pairs with better fitness value enters reproduction process more often, but also weak individuals have chance to enter the reproduction process.

In the figure 7.1 is shown general schema of evolution algorithm

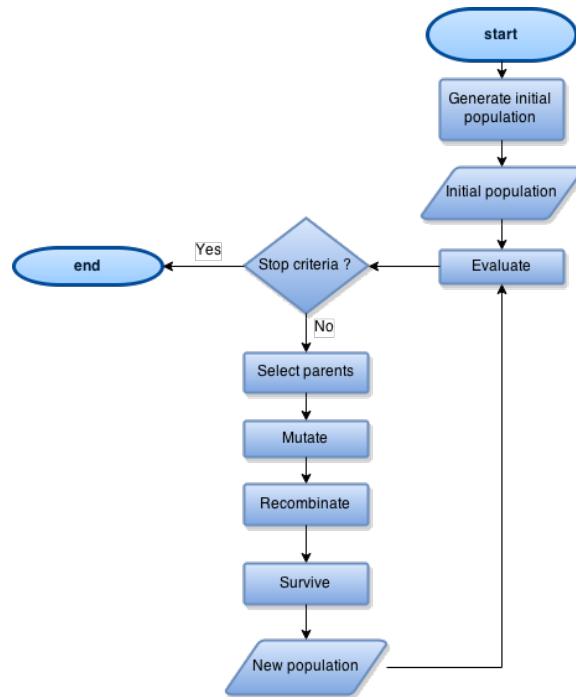


Figure 7.1: Flowchart of evolution algorithm

7.1.4 Components of evolution algorithm

Evolution algorithms consists of following components:

- Representation of solutions - encoding
- Function of evaluation the quality of solutions - fitness function
- Population
- Techniques for selecting parent individuals
- Variation of selection and mutation operators
- Renewal of the population

Specification of individual components is diverse for various types of evolution algorithms. The method of encoding is called genotype, for instance integers can be encoded in binary format. By phenotype we then mean the value of binary string, e.g., string 1010 corresponds to phenotype 12 in decadic format.

7.1.5 Application of evolution algorithm

Evolution algorithms are applicable in all those fields of computation, where classic analytical and convention methods fail, thus especially when dealing with complex nonlinear NP-complete problems. They are used in many fields of engineering design and artificial intelligence:

- Numeric, combinatorial optimization

- Modeling and identification of a model
- Management and planning
- Engineering design
- Data mining
- Machine learning and artificial intelligence

A well-known problem is finding the shortest Hamiltonian path in graph. In practice, this could be a solution to the logistical task of travelling salesman, known under acronym TSP. Encoding is mostly permutation. Also is known binary encoding, but it's rarely used.

7.1.6 Robustness of evolution algorithm

Evolution algorithms(EA) are typical by its robustness, the ability to solve difficult optimization and decision tasks, which can be characterized by properties such as multimodality, multicriteriality and various types of restrictive conditions.

Its deployment is effective in tasks, which can be characterized as follows:

- The solutions space is too wide and expert knowledge is missing, which could stretch the space of feasible solutions
- It's not possible to execute mathematical analysis of given problem
- Traditional approaches fail
- It's a task with multiple extremes, criteria and constraints conditions

Evolution algorithms are being used for numerical and combinatorial optimization, in circuits design, managing and planning the production, machine learning, creation of economic, social and ecological models etc.

It's necessary to show some certain disadvantages of evolution algorithms:

- The quality of solutions can be evaluated only relatively. It's unable to test, whether the result is global optimum
- Many tasks are very time-consuming
- For very large problems, EA provides solution, which is too far from optimum
- Completion of optimization is based on an explicit time limit or stagnation objective function

For the evolution algorithms design, there are many commercial products, which provide rapid application design. The key step is the choice of encoding a problem, which should take into consideration used genetic operators and the difficulty of fitness function computation.

Classification of individual evolution algorithms into field of softcomputing is shown on Figure 7.2. The oldest techniques are genetic algorithms and evolution strategies. Evolution programming and genetic programming were developed later.

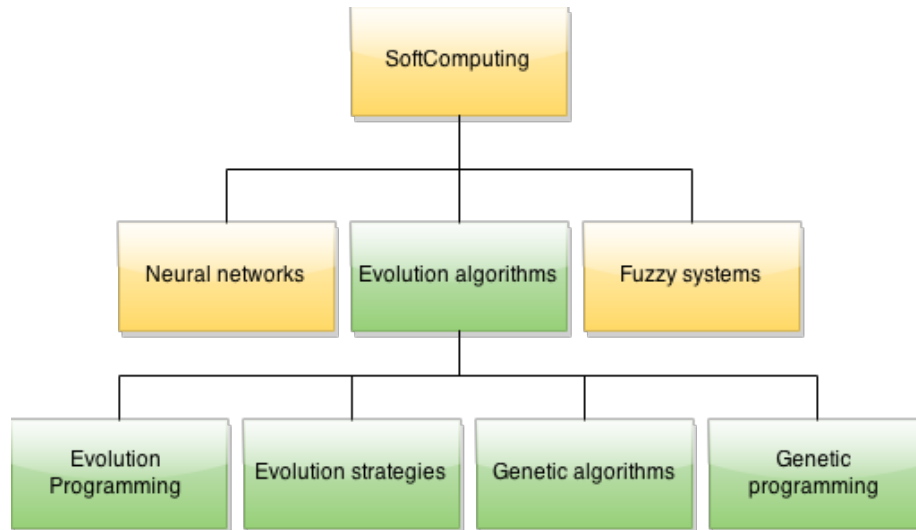


Figure 7.2: Review of evolution algorithm types

7.2 Genetic algorithms

7.2.1 Introduction

Genetic algorithms(GA) are the most widespread types of evolution algorithms. The first person, who formulated the idea of genetic algorithm was J. Holland, who used GA for study of adaptive behavior. J. Holland presented genetic algorithms firstly as search algorithms for adaptive systems of artificial intelligence. He defined crossover operator and inversion operator. Crossover operator is considered as the main distinctive factor of genetic algorithms, which uses this recombination operator primarily. In the book *Genetic Algorithms in Search, Optimization, and Machine Learning* from David Goldberg is detailed description of standard GA, including examples of implementation. This book is considered as „bible“ of genetic algorithms. After its release in 1989 there was significant growth of interest of this science field called *evolutionary computing*, which is now part of soft computing(extended to fuzzy logic, neural networks and fractals). After 1989 also grows interest of publications, which are dealing with problems solved by genetic optimization.

The main application, however, have found genetic algorithms in optimization problems, especially in the search for the global extreme of single and multi-dimensional functions. In this context we talk about classic/simple/canonical genetic algorithm (SGA), which is clearly specified in Table 7.1

Representation	Binary string
Recombination	Single-point crossover
Mutation	One bit mutation
Parents selection	Proportional selection based on fitness value
Renewal of population	Generative - with full replacement of previous population

Table 7.1: Specification of genetic operators

7.2.2 Canonical scheme of genetic algorithm activity

Firstly, we need to declare notions, which we will use in following text.

Chromosome (individual) which codes solution is represented by binary vector (string) of constant length n :

$X = (X_0, X_1, \dots, X_{n-1})$, where X_i is the i -th variable of given string

$x = (x_0, x_1, \dots, x_{n-1})$ is string of specific variable instances

$X_i = x_i, x_i \in \{0, 1\}$

$D = (X^1, X^2, \dots, X^N), X^j \in D$ is set of N strings, which specifies population D

$D \subseteq \{0, 1\}^n$

Let f be the payoff function defined upon set of binary strings of length n

$$f : \{0, 1\}^n \rightarrow R \quad (7.1)$$

which evaluates every binary vector x by real number. The goal is to find the global extreme of the function f . In the case of minimization function its about finding vector:

$$x_{opt} = \underset{x \in \{0, 1\}^n}{\operatorname{argmin}} f(x) \quad (7.2)$$

The function f is usually transformed into a function of expediency F (fitness function) in a way, that the original optimization problem was transferred to maximizing task and appropriate scale fitness was achieved. Use of this transformation helps also to change the selection pressure, which significantly affects evolution process convergence.

The activity of standard GA algorithm can be described by following pseudocode:

1. Set variable $t = 0$, generate initial population $D(0)$ randomly with mightiness N ,
2. Perform evaluation of population of individuals $D(t)$ by fitness function $F(X)$,
3. Generate population of offsprings $O(t)$ with mightiness $M \leq N$ using crossover and mutation operators
4. Create new population $D(t+1)$ by substiting part of population $D(t)$ with individuals from $O(t)$
5. Set $t \leftarrow t + 1$
6. If stop condition is not satisfied go to (2)

Phases of GA design

When dealing with genetic algorithm design is necessary to solve following phases:

- Problem representation

- Initial population
- Evaluation of individuals
- Selection operators
- Recombination operators
- Mutation operators
- Population renewal
- Size of population
- Algorithm termination

7.2.3 Selection

The selection operator creates new population $P(t+1)$ by choosing among individuals with possible repetition from old population $P(t)$. Selection can be done by several options. The most common is random selection using roulette (roulette wheel selection) [5], where the probability of individual selection $p_s(x_i)$ each individual is proportional to its fitness.

The selection process is an important part of genetic algorithms. On one hand, it must favor individuals with higher fitness value, on the other must choose a new population which is sufficiently diversified. If selection algorithm doesn't meet this expectations, it leads in first case to slow convergence of the algorithm, in second case to so called premature convergence(into local optimum). *Selection intensity* or also *selection pressure* is expressed by following equation:

$$I = \frac{\overline{M^*} - \overline{M}}{\overline{\sigma}} \quad (7.3)$$

where $\overline{M^*}$ denotes average value of fitness function in the population before selection, \overline{M} denotes average value of fitness function after selection and $\overline{\sigma}$ is the variance of fitness values before selection.

The higher the selection pressure is, the faster the algorithm converges - population after selection contains more individuals with higher fitness. Together however danger of premature convergence grows. For this situation was introduced the notion of *takeover time*, which denotes the count of generations, which are necessary for selection to fill whole population of N individuals by the best chromosome, by absence of recombination and crossover operator.

The loss of diversity is a measure of „genetic material“. During selection process, some part of chromosome is not selected. However these chromosomes also contains some information. Because of that, their loss increases the danger of premature convergence of genetic algorithm. The loss of diversity p_d is the ratio of chromosomes, which weren't selected to overall count of chromosomes in population.

There are lot of variations of selection algorithms and its modifications. The most commonly used algorithms that provide useful results are as follows:

- Proportionate selection (Roulette wheel selection)
- Residual selection (truncation)

- Linear ordering (ranking)
- Exponential ordering (ranking)
- Tournament selection

Proportionate selection

Proportionate selection was first algorithm for individuals selection. The selection probability i -th individual is given by following equation:

$$p_i = \frac{f_i}{\sum_{j=0} f_j} \quad (7.4)$$

The aim of parent-chromosome selection is to give bigger chance to those individuals, who are achieving better results. This can be done several ways. The simplest method is *Roulette wheel selection*. There can also appear negative influence of proportional selection. If there will be individual with relatively high fitness value, the population is being gradually replaced by its chromosome. To overcome this trend is suitable to adjust original fitness value (scaling process) so that the difference between quality of best and worst individual will be reduced.

The most commonly used techniques are:

1. Comprimation of fitness function (windowing):
 $f'(i) = f(i) + \beta^t$, where β is the worst fitness value in current t -generation
2. Sigma scaling:
 $f'(i) = \max(f(i) - (\langle f \rangle - c * \sigma_f), 0.0)$, where c is constant, mostly 2.0 and $\langle f \rangle$ is mean fitness value

Even though Roulette wheel selection method is based on random selection, the chance of each parent to be in next generation is directly proportional its evaluation(fitness). The name is derived from its similarity with the shape of roulette wheel. It can be also interpreted as wheel of fortune. The only difference is that the individual slices are not the same, but their size(angle) is proportional to chromosome evaluation. Take in note, that individual payoff have to be positive number. In Figure 7.3 is shown the principle of Roulette wheel selection and its scaling by constant number 10.

Linear ordering (ranking)

Linear ordering requires ranked population so that the worst individual had index 1 and best N . The selection probability value is then given by equation:

$$P_{in-rank}(i) = \frac{2-s}{N} + \frac{2i(s-1)}{N(N-1)} \quad i \in \{1, 2, \dots, N\} \quad (7.5)$$

In Table 7.2 is example of the computation of selection for selected fitness function values and two values of selection pressure [5].

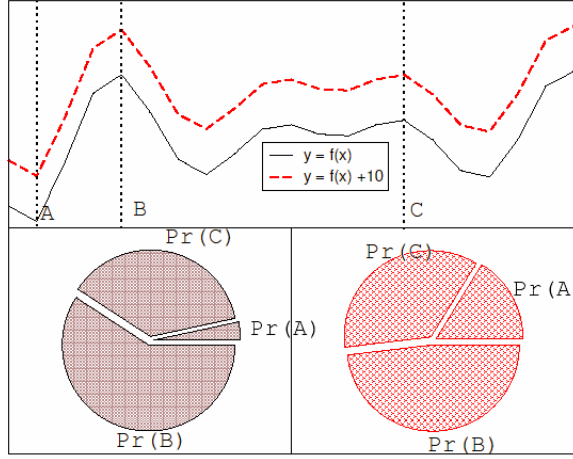


Figure 7.3: Roulette wheel selection[5]

	Fitness	Rank	P_{selFP}	$P_{selLR}(s = 2)$	$P_{selLR}(s = 1.5)$
A	1	1	0.1	0	0.167
B	5	2	0.5	0.67	0.5
C	4	2	0.4	0.33	0.33
SUM	10		1.0	1.0	1.0

Table 7.2: Linear ranking example [5]

Exponential ordering

Exponential ordering differs from linear only so, that the selection probability in population is not spread lineary, but with exponential addition. Again index N denotes the index of best individual in population. The selection probability i -th individual is given by equation:

$$P_{exp-rank}(i) = \frac{1 - \exp^{-i}}{\exp} \quad (7.6)$$

We can summarize the properties of each approach:

- Linear ordering has limited selection pressure
- Exponential ordering may allocate more than 2 copies of best individual and normalization constant c reflects the size of population

Parameter c is selected in range of $0 < c < 1$. This selection algorithm is best out of mentioned group. With the change of parameter c is possible to get suitable selection pressure together with little variability loss. The time complexity of this algorithm is $O(N \ln(N))$.

Tournament selection

This algorithm achieves results that are very similar to the previous method. His biggest advantage is absence of ranked population requirement and the simplicity of own selection. According to this is tournament selection often used. Precisely in this spirit that selection takes place - from N population individuals are t individuals selected. In next generation goes only the best one from t individuals. Whole process is being repeated as many times

as is the count of offsprings in next generation. The bigger the t parameter is, the higher the selection pressure is.

7.2.4 Crossover

The crossover operator (crossing-over) is characteristic for genetic algorithms and represents for them the basic operator of population evolution. This operator is often subject of conflicting opinions, and whether it should be even used, because it has, among some scientists, inappropriate biological basis. Proponents of genetic algorithms highlight the ability to exchange information between individuals. Opponents of GA contrariwise consider crossover operator as „breaking up the building blocks of bits“ and applies this together with mutation operator with very small probability.

Theory of building blocks explains convergence of genetic algorithms. Genetic algorithms are, according to this theory, able to identify quality gene blocks (bits) and with help of recombination operator (crossover) build up blocks with growing size. This growth is manifested externally as the convergence of algorithm to maximum fitness value. The crossover operator applies with probability p_c .

There exists whole set of variations. The basis of it is random pair selection of individuals, in which exchange of genetic information (recombination) occurs so, that from point of crossover genetic information will be exchanged. Very often is this operation performed not with 100%, but for instance with 70% probability. With this approach is reproduced only part of the population.

Single and multi point crossover

Single and multi point crossover is the simplest way of chromosome recombination. It is based on biologic analogy, when the recombination may appear in one or many points of chromosome. In single-point crossing exchange are in selected place, parts of parents interchanged. Two offsprings will be created, each of which contains part of the genetic information from both parents. In multi-point crossing exchange occurs the interchange of several parts of both parents chromosomes. Some descendants may occur with higher fitness value. These individuals are thanks to its quality selected into new population with increased frequency.

On Figure 7.5 is shown three point crossover example

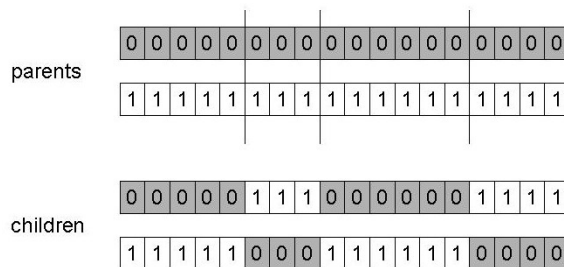


Figure 7.4: Example of three point crossover [5]

Uniform crossover

Uniform crossover represents another alternative to recombination operators. This operator scans pair 0 and 1 valued chromosome and performs interchange specific genes with probability p_u . Uniform crossover was avoided for too much „code destroying“ according to the theory of building blocks. On the other hand, the single crossbreeding can bring in the desired population diversity and the genetic algorithm is useful in solving complex multidimensional functions with many local extremes. Uniform crossover is a powerful weapon in the fight against premature algorithm convergence. Space of potential solutions is scanned more intensively than in case of single-point crossover.

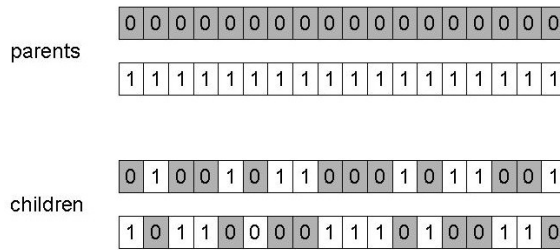


Figure 7.5: Example of uniform crossover [5]

7.2.5 Mutation

Mutation is the reproductive operator with a small frequency of occurrence but is very significant. Standard operator mutation modifies (creating mutant) genes with probability p_m . The most common is the negation bit, which is used with probability 0.0005 to 0.01. Mutations are for genetic algorithms source of new information. Effect of mutation can be completely negligible or vice versa, with fatal consequences for individuals (typically a mutation in the exponent when encoding real numbers). Too big mutation probability p_m causes instability of the population development, and vice versa too small mutation probability can not produce enough of new information for further development.

There are a number of special mutation operators for specific tasks. E.g. inversion operator. This operator inverts the order of the single elements (bits) between two randomly selected points within the chromosome.

7.2.6 Population renewal

Population recovery method determines the dynamics of state space search. There are two basic ways: a) the generative GA with complete recovery (parents extinction) when the old offspring population is completely replaced, b) partial renewal (steady state) where only one offspring replaces the weakest individual of original population. Very often are used mixed variants, when is replaced 20-50% of original population with new offsprings. There are used various techniques of substitution by partial renewal:

- According to quality: $P(t+1) = \frac{(P(t) \cdot vO(t))}{(P(t) \cdot vO(t))_{worst}}$ while preserving population size
- Tournament
- Elitism

- Factor of overcrowding. Subset of parents is randomly selected, offspring replaces individual with similar genotype

7.2.7 Schema theorem

Evolution process, specified by the operators of selection, crossover and mutation, leading to finding the optimum or suboptima was formalized by J. Holland with use of scheme theory, which was elaborated later by other authors [Goldberg 1986]. From the theory and practice of genetic algorithms, it is still known current problem of suitable problem encoding and suitable assortment of genetic operators selection, which greatly affects solution convergence. Most GA models is based on the theory of building blocks (schemes). Scheme is similarity template defined over the alphabet $\{0, 1, *\}$. Symbol $*$ denotes, that on respective place of binary string may appear arbitrary value 0 or 1. We say that a string $r \in \{0, 1\}^n$, is an example/model of scheme $t \in \{0, 1, *\}^n$, if in any scheme position with other character than $*$ is the value of the character in string same as character value in scheme. For more information on this area refer to relevant literature [5] [24]

7.2.8 Deceptive problems

Deceptive problems are characterized by the so-called nonlinear interaction parameters / genes, when best rated solution is composed of under-rated schemes of lower order. Minimal deceptive problem (MDP):

Let assume binary chromosomes of length = 2: $\{00, 01, 10, 11\}$ and fitness function $f()$.

MDP has a global optimum $f(11)$, but is considered as deceptive, because some sub-optimal 1. order schemes have greater value of payoff function, than other 1. order schemes: $f(0*) > f(1*) > f(*0) > f(*1)$,

so schemes $0*$ and $*0$ are being spread, which are composed into 00 individual, while for the creation of optimal individual 11 should be schemes $1*$ and $*1$ propagated.

MDP is an abstract example (when the length of chromosome is 2), because there is no such evaluation for individuals $00, 01, 10, 11$, in order that average evaluation of 1. order schemes could satisfy described relationships.

There are two types of minimal deceptive problem: MDPI and MDPII, which are shown in Figures

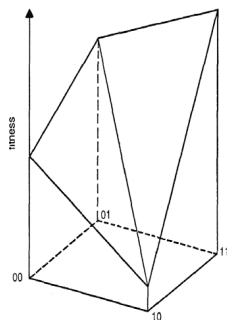


Figure 7.6: type I (MDPI)[5]

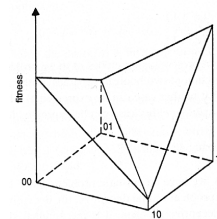


Figure 7.7: type II (MDPII)[5]

On longer chromosomes then we work with schemes of type $f(* * 0 * * * 1*)$, etc.

7.3 Evolution strategies, ES

7.3.1 Introduction

In 1963 began, in those time still students, Hans-Paul Schwefel and Ingo Rechenberg at the Technical University of Berlin, with mimicking evolution in nature to be able to optimize the shape of objects in order to decrease the turbulation in wind tunnel. They were obviously convinced, that their method mimics best continuous changes in natural processes.

Therefore called their method generally - *evolution strategies*. Over time, although had been shown, that the method is convenient only for solutions of certain type problems, mostly engineering, but the name remained. According to practical methods focus on engineering problems solving, became this method popular, especially between construction and machine engineers.

As with many other scientific approaches, also in optimization of one method and its superiority competing methods(e.g. genetic algorithms) are decisive emotion rather than blunt scientific argumentation. As genetic algorithms as evolutionary strategies or perhaps simulated annealing¹ contains enough parameters to setup little differences in procedures. If it happens then, that somebody proves, that with his/her favourite algorithm got better results, or achieved results faster, you can always say, that it's based on parameters settings, that can be always adjusted in order to get with another technique/algorithm better results. When selecting algorithms for problem solving, to a great extent depends on solver's personal choice. There exists also *evolution programming*, which also as *evolution strategies* puts more emphasis on similarities in parents and their offsprings behavior, rather than mimicking the genecic operators, as we can observe in processes at molecular level.

Although *evolution programming* is similar to *evolution strategies*, both procedures were developed independently. Evolution programming was designed near the year 1960 by Lawrence Fogel.

Fogel, with evolution programming, began with forecast of symbiotic strings generated by finite automatas from so-called Markow processes and unsteady time series, in an effort to predict events, which is basic condition of organism's artificial intelligence, who, based on forecast adjust its behavior.

From practical point of view is evolution programming in optimization used as successfully as in evolution strategies.

7.3.2 Main characteristics in Evolution strategies

Evolution strategies, as well as evolution programming and genetic algorithms(including other optimization techniques) are suitable especially for functions optimization, where is unable to find analytic solution and classic techniques like gradient methods fail. The main domain of these methods is function optimization, whether with discrete or real variables, which are „spread“ across search space (like wildly rugged mountains), and therefore have lot of local optimas and is difficult to find the best one.

Although both, evolution programming and evolution strategies exists independently on each other more than 30 years, only few years ago their proponents admitted, that these methods are in fact very similar. Principles of those methods, i.e. optimization set of individuals in population, inheritance of most individual properties while transitioning from

¹<http://katrinaeg.com/simulated-annealing.html>

one population into following, and better individuals based on their fitness value are in evolution strategies and evolution programming equal with genetic algorithms. If, however, evolution strategy and evolution programming are trying to optimize their functions of real variable, they truly operate with values of real variables (unlike genetic algorithms, which operates with strings of bits and these converts to real numbers only when it's necessary).

Representation	Real encoding (x_1, x_2, \dots, x_n) , x_i is real number
Recombination	Discrete or arithmetic
Mutation	Gaussian probability distribution
Parents selection	Random - with uniform distribution
Renewal of population	(μ, λ) or $(\mu + \lambda)$
Special procedures	Autoadaptation of mutation size

Table 7.3: Specification of evolution strategies operators

7.3.3 Mutation in evolution strategies

The important part which drives evolution process in evolution strategies is mutation, which is applied on each target parameter/element of solution vector (x_1, x_2, \dots, x_n) . It uses Gaussian probability distribution with 0 mean value and σ variance. Often it's used normalized distribution with denotation $N(0, 1)$.

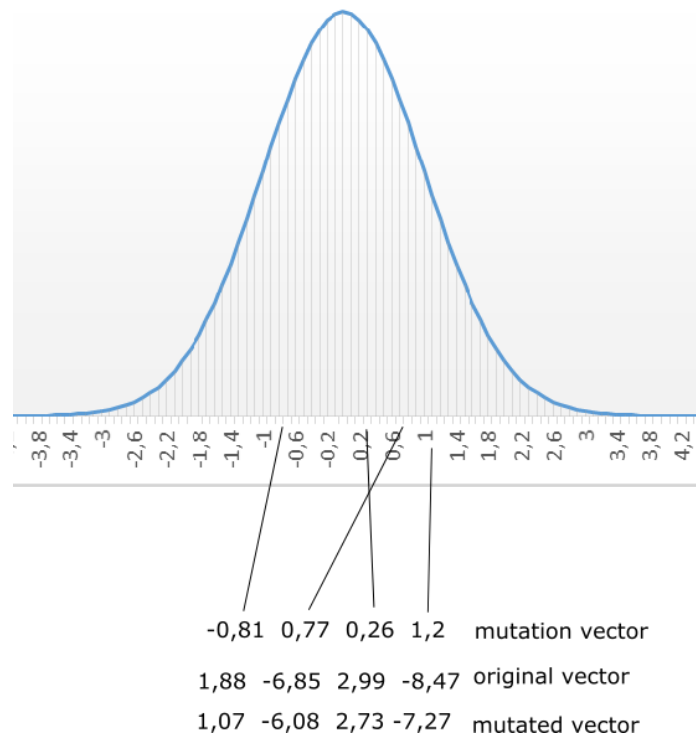


Figure 7.8: Mutation in evolution strategies

In Figure 7.8 is displayed representation of Gaussian normal distribution with mean variance value $\sigma = 1$ and 4 randomly generated values. These values after its random

mixing, form mutation vector.

These values are then added to vector values of parent and form new offspring vector of values.

7.3.4 Crossover

Evolution strategies mimics evolution of individuals in species, and therefore crossover operation began to be used also in evolution strategies.

According to, that individuals represents directly encoded problem solutions, i.e. a mixture of real numbers, integers and other parameters, crossover is specifically defined for each type of problem.

Suppose, that we have picked two individuals characterized by two vectors of real numbers. Discrete crossover creates new individual, whose vectore is formed by the values, that were taken either from one, or from another vector.

Crossover „average“ forms individual, whose elementary values of vector are always created as average value of its parents.

8,11	-4,22	5,12	6,81	parent_1
2,09	-4,22	-1,12	6,81	offspring
2,09	2,22	-1,12	-2,81	parent_2

Figure 7.9: Discrete crossover

8,11	-4,22	5,12	6,81	parent_1
2,09	2,22	-1,12	-2,81	parent_2
5,10	-1,00	2,00	2,00	offspring

Figure 7.10: „Average“ crossover

In Figure 7.9 is shown example of discrete crossover in evolution strategies. In Figure 7.10 is shown example of „average“ crossover in evolution strategies.

These are two of many variations of crossover in evolution strategies.

7.3.5 Population renewal

In the original implementation, better offspring simply displaces worse parent, who is then removed from the population. If parent is still better than offspring, then parent stays in population. We call this variant plus strategy ($\mu + \lambda$) (size of parents population + size of offsprings population, originally variant $1 + 1$).

For an instance let's say that later algorithms generate from 10 parents 100 offsprings, then ranks parents and offsprings according to theirs fitness value and pickup 10 best into next generation. It's actually an application of elitism principle.

Another strategy is strategy (μ, λ) , in which could be individuals of parent population replaced by inferior offsprings. This technique is often used for difficult optimization tasks

and allows to leave local optimum. The selection pressure is high, often is used constant $\lambda \approx 7\mu$.

7.3.6 Basic strategy 1+1

In this variation is population formed by single individual, who is represented only by vector of search parameters (x_1, x_2, \dots, x_n) . To individual parameters is applied mutation from Gaussian distribution. Parameter σ is modified in order to let approximately $\frac{1}{5}$ of mutations be successful, i.e. to let 20% of offsprings be better than parents. If there is less amount of successful offsprings, the number of mutations will decrease, if more, it will increase.

We can formalize this approach with following equations [5]:

$$\begin{aligned} \sigma &= \frac{\sigma}{c}; p_u > 1/5 \\ \sigma &= \sigma * c; p_u < 1/5 \\ \sigma &= \sigma; p_u = 1/5 \end{aligned} \tag{7.7}$$

where p_u stands for % of successful mutations, $0.8 \leq c \leq 1$

You can refer, for an algorithm, which uses these conditional expressions, to relevant literature [5].

7.3.7 Autoevolution of control parameters

Besides the basic explicit rule for modifying 1/5 of the mutation size are using other three techniques based on autoevolution of control parameters. That means, that chromosome is formed by two sections: section of target parameters and section of control parameters. Both sections are subject to evolutionary processes and changes their values during the process of evolution. For more information refer to relevant literature [5].

To summary this evolution algorithms are used mainly for optimization problems with real parameters (search for multi-value function extremes) or problems with mixed parameters (real, integer, discrete parameters).

The variations of evolution strategies with autoevolution of control parameters are more efficient than variations with adaptation of mean distribution dispersion using rule 1:5.

7.4 Multimodal functions

Traditional genetic algorithms (GA's) with elitist selection are suitable for locating the optimum of unimodal functions as they converge to a single solution of the search space. Real optimization problems, however, often lead to multimodal domains and so require the identification of multiple optima, either global or local. For this purpose, niching methods extend simple GA's by promoting the formation of stable subpopulations in the neighborhood of optimal solutions.

Niching methods have been developed to reduce the effect of genetic drift resulting from the selection operator in the standard GA. They maintain population diversity and permit the GA to investigate many peaks in parallel. On the other hand, they prevent the GA from being trapped in local optima of the search space. Niching GA's are based on the mechanics

of natural ecosystems. In nature, animals compete to survive by hunting, feeding, grazing, breeding, etc., and different species evolve to fill each role. A niche can be viewed as a subspace in the environment that can support different types of life. A species is defined as a group of individuals with similar biological features capable of interbreeding among themselves but that are unable to breed with individuals outside their group. For each niche, the physical resources are finite and must be shared among the population of that niche. By analogy, niching methods tend to achieve a natural emergence of niches and species in the environment (search space). A niche is commonly referred to as an optimum of the domain, the fitness representing the resources of that niche. Species can be defined as similar individuals in terms of similarity metrics.[15]

7.4.1 Techniques for multimodal functions solving

There are two basic explicit techniques for multimodal function solving:

- Fitness sharing
- Crowding

Fitness sharing

Fitness sharing modifies the search landscape by reducing the payoff in densely populated regions. It lowers each population element's fitness by an amount nearly equal to the number of similar individuals in the population. Typically, the shared fitness f'_i of an individual i with fitness f_i is shown in equation 7.8:

$$f'_i = \frac{f_i}{m_i} \quad (7.8)$$

where m_i is the niche count which measures the approximate number of individuals with whom the fitness f_i is shared. The niche count is calculated by summing a sharing function over all members of the population:

$$m_i = \sum_{j=1}^N sh(d_{ij}) \quad (7.9)$$

where N denotes the population size and d_{ij} represents the distance between the individual i and the individual j . Thence, the sharing function (sh) measures the similarity level between two population elements. It returns one if the elements are identical, zero if their distance is higher than a threshold of dissimilarity, and an intermediate value at intermediate level of dissimilarity. The most widely used sharing function is given as follows:

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_s)^\alpha & \text{if } d < \sigma_s \\ 0 & \text{otherwise} \end{cases} \quad (7.10)$$

where σ_s denotes the threshold of dissimilarity (also the distance cutoff or the niche radius) and α is a constant parameter which regulates the shape of the sharing function. α is commonly set to one with the resulting sharing function referred to as the triangular sharing function. The distance d_{ij} between two individuals i and j is characterized by a similarity metric based on either genotypic or phenotypic similarity. Genotypic similarity is related to bitstring representation and is generally the Hamming distance. Phenotypic

similarity is directly linked to real parameters of the search space. It can be the Euclidian distance for instance.[15]

Crowding

Crowding methods insert new elements in the population by replacing similar elements. We distinguish deterministic and multimodal crowding.

Deterministic crowding

- Tries to distribute individuals evenly into niches
- Relies on assumption, that descendants are close (both phenotypic and genotypic) to parents
- Uses genotypic and phenotypic distance

After crossover and eventually mutation, each child replaces the nearest parent if it has a higher fitness. Thus deterministic crowding (DC) results in two sets of tournaments: (parent 1 against child 1, and parent 2 against child 2) or (parent 1 against child 2, and parent 2 against child 1). The set of tournament that yields the closest competitions is held. Similarity is computed using preferably phenotypic distance. With two distance comparisons per set of tournaments and $N/2$ sets of tournaments per generation, the resulting order of complexity of deterministic crowding is $O(N)$./citeevolutionaryComputation

Multimodal crowding

- In the selection phase, are to each individual selected k individuals, i.e. each individual will be parent
- Then as a second parent is to each individual selected the most similar one
- In the renewal phase, for each individual are S samples created, with C_f individuals in each sample. From each sample is then the most similar to given individual selected, who replaces the worst of them.

The most common parameter setup is: $S \in \langle 2, 4 \rangle$, $C_f = \langle 1, 5 \rangle$ % of population.

Chapter 8

Implementation

For the purpose of the work, there was necessary to implement solution(algorithm), which will be able to extract relevant features from human eye for further iris detection. To be exact, mentioned algorithm should focus on pupil and iris boundary detection and this should be done with usage of evolution algorithms, with the best possible accuracy and time minimization.

Also it was necessary to test mentioned algorithm, its time consumption and compare results with convention solutions.

As it was mentioned in the chapter [Introduction](#), there already exists many convention solutions which show good results[19][14][9]

Therefore arises question, why to implement algorithm, which overall already shows good results.

Main motivation of this, was to compare results of solution, which uses evolution algorithm, to convention solutions. Measure accuracy and time consumption in order to tell how different results give these different approaches and whether evolution algorithms are suitable for use in human eye iris processing.

I therefore divided implementation into three main parts:

- Graphical user interface (GUI)
- Convention solution
- Solution with the use of evolution algorithms

8.1 Graphical user interface (GUI)

GUI was implemented in JavaFX framework¹, because of its ease of use and broad range of tools.

I divided GUI into three main parts:

- Main panel
- Right panel

¹<https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>

- Console output

User can easily switch between results as images, and results as evolution data as it is shown in Figure 8.1 and Figure 8.2

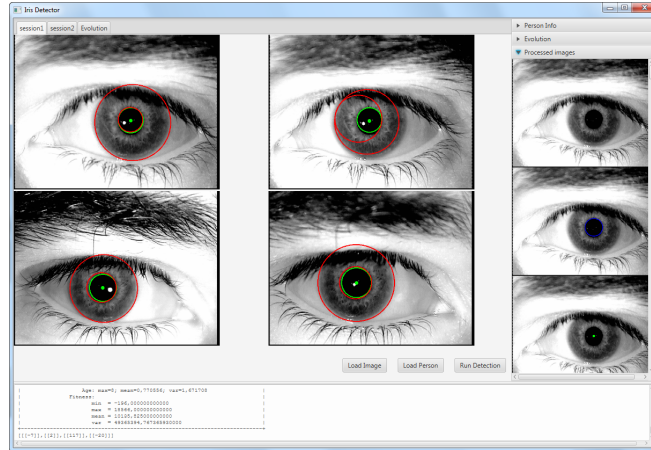


Figure 8.1: Processed images with iris and pupil circles

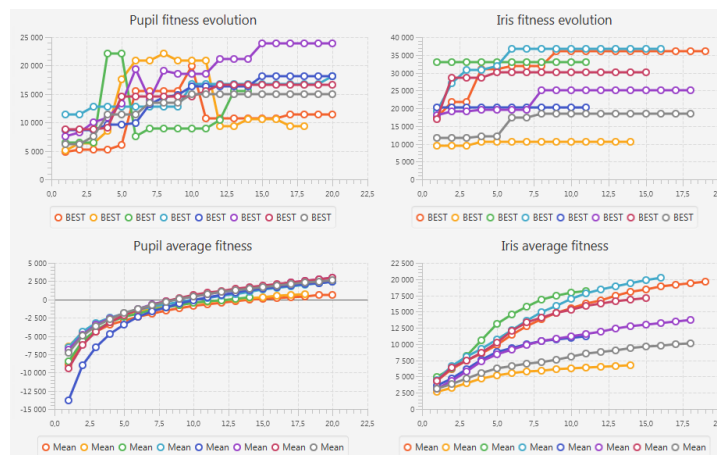


Figure 8.2: Evolution data results

Main panel Main panel contains tabbed pane, which contains three tabs. Two of them (session1 and session2) displays results from iris segmentation process (both convention and evolution approach). Third tab - displays statistics from evolution process. Statistics are displayed as charts, which contains data about progress of average and best fitness function, obtained through generations. This is done for both, iris and pupil detection process.

In bottom of *Main panel* are located three buttons with selfexplanatory labels:

- Load Image - loads image which is then used in evaluation process
- Load Person - loads all images for related person (for more info about image database visit [Data](#) section)

- Run detection - starts both types of detection - using evolution algorithms and convention solution

Right panel Right panel consists of three „accordion“ panels:

- Person info
- Evolution
- Processed images

Person info panel contains information regarding person, which is loaded (the information is loaded from description file, which is part of *BioSecure*(9.1) database).

Evolution panel contains user interface utilities for setting up evolution parameters. Currently is able to setup *Generation size* and *Population size*.

Panel *Processed images* displays images, which were processed during iris detection process. For each subprocess its output is displayed as separate image in *Processed images* pane.

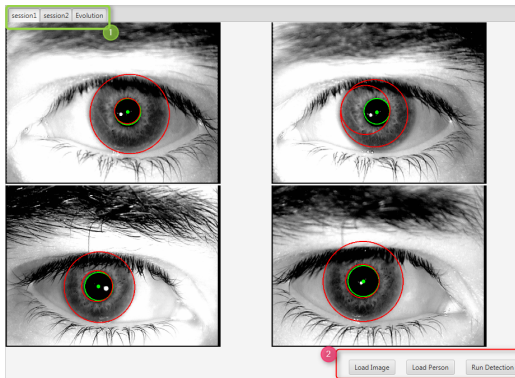


Figure 8.3: Main panel: 1)Tabs 2)buttons

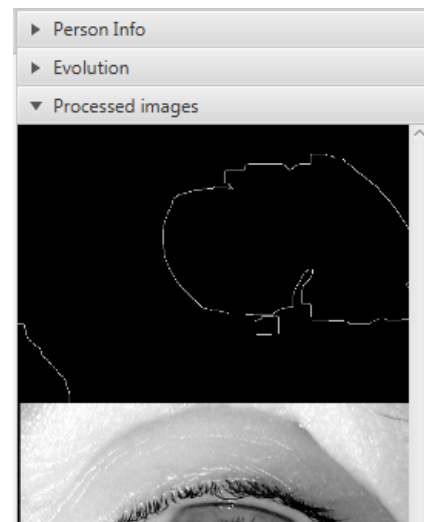


Figure 8.4: Right panel

Console output Displays statistics acquired from last generation of evolution process. The statistics contains:

- Evolution statistics: Number of generations, killed individuals and invalid individuals
- Fitness statistics: Age of oldest individual (with best fitness), minimal, mean and maximal fitness value
- Time statistics: Time consumption of selection, altering, fitness calculation and overall time of computation

8.2 Convention solution

As it was already mentioned, currently exists broad variety of convention methods for iris segmentation and detection.

For the purpose of this work I focused on proper iris and especially pupil detection using convention method. This is then used, as it is described more specific in [Solution with the use of evolution algorithms](#), in evolution solution for fitness value evaluation.

For image processing was chosen OpenCV² library, as this solution is implemented in Java SE³, and there already exists OpenCV wrapper for Java language, even though is primarily designed for C++ language.

Also is necessary to mention, that all images were taken as grayscale images, therefore, unless it was necessary to display iris, or pupil circle, no image color transformation on obtained image was necessary.

8.2.1 Pupil detection

Pupil area is relatively easy to distinguish from another part of image. It is a dark area and mostly its circular (more or less with little deviations). Therefore I considered these approaches with following results:

- Circular Hough transformation
- Distance transformation
- Contour detection and its analysis

Image preprocess Before pupil segmentation process began, it was expected to preprocess image, in order to remove reflections from image, as it is mentioned in [Data](#) section, so transformations (like Hough, or Distance) could work properly.

The whole preprocessing consisted of these parts:

- Tophat morphological operation
- Image thresholding
- Gaussian blur morphological operation
- Image dilation
- Subtraction from original image

Tophat operation is the difference between an input image and its opening - works as follows:⁴

$$dst = tophat(src, element) = src - open(src, element) \quad (8.1)$$

where *dst* is the destination image, *src* is the source image, *element* is the kernel over which tophat is computed and *open* is an another morphological operation which simply in binary image removes small objects. For more info regarding open operation refer to

²<http://opencv.org/>

³<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁴http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html#top-hat

available resources⁵.

In other words this transformation detects small reflection in pupil area. This is not enough because it still contains lot of noisy area. Therefore this need to be followed by image thresholding, which remove non-continuous areas by setting each pixel to 0, or to 1, in case when given pixel value is over or under threshold. *Gaussian blur* removes very small areas (less than 9x9 pixels - size of kernel) by setting given pixel value to mean of its neighbours. *Image dilation* enlarges then camera reflection in order to remove it when it is being subtracted from original image.

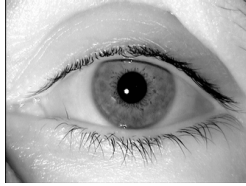


Figure 8.5: Original image before preprocessing

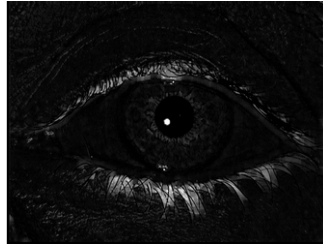


Figure 8.6: Tophat operation

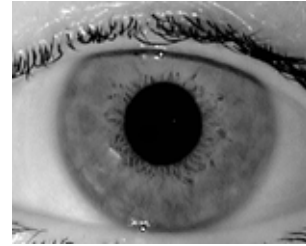


Figure 8.7: Result after preprocessing

Circular Hough transformation Circular Hough transformation (CHT)^{3.2.1} is a technique for finding a circle in 2-dimensional space. The circular Hough transform first generates two dimensional parameter space (x_c, y_c) using the gradient of grayscale. Then the radius of circle r is determined for each local maximum in the (x_c, y_c) space. That means that CHT counts pixels in image for given circle $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and for given votes/threshold (number of pixels) is then decided, whether this set of pixels can be considered as circle or not.

From above it is clear, that the crucial factors for proper Hough transformation are: the radius range, threshold value for *canny edge detection* and number of votes for candidate circle.

According to fact, that thresholded image often contained eyebrow or eyelashes contours, occasionally happened that false circles were detected and this approach should then require further image preprocessing.

Distance transformation To get more information from image and pupil area boundary, I considered to use distance transformation⁶ in order to obtain pupil center coordinates. Distance transformation calculates the distance to the closest zero pixel for each pixel of the source image. Therefore was necessary to obtain thresholded image where pupil area is highlighted. After this was distance transformation, with normalization factor 1.0, applied. Although in most of the pictures, center of the pupil was successfully founded, in pictures where preprocessing didn't remove camera reflection from pupil, it failed.

⁵http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html

⁶http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html#distancetransform

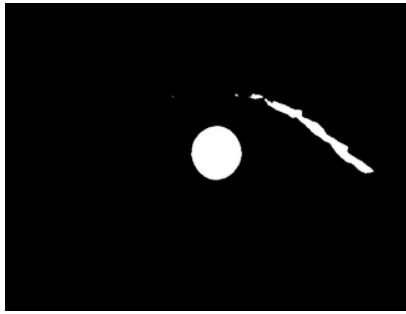


Figure 8.8: Thresholded image

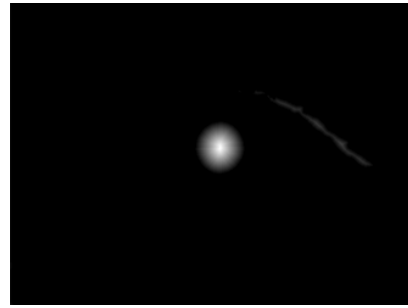


Figure 8.9: Distance transformation applied

Contour detection and its analysis As another approach was considered contour detection.

Contour detection⁷ simply detects closed areas over binary image. Since in thresholded image lot of false noisy objects are detected as contours, it needs to be analyzed, whether given contour/area is searched pupil.

This can be done by iterating through contour collection, when every area is being checked on its proportions e.g. width/height ratio, size etc.⁸

This approach with proper parameter settings showed best results.

8.2.2 Iris detection

For Iris detection using convention method, was Hough circle transformation chosen.

As this detection finds circles in edge image, it was necessary to obtain proper parameters for canny edge detector. After several attempts, for lower threshold value was set to 150 for binary image conversion.

However with this method I didn't get any satisfying results and according to fact, that this method detection wasn't used in further work, especially fitness function evaluation, I haven't continued with further investigation. This should be then addressed in future work.

8.3 Solution with the use of evolution algorithms

The solution with the use of evolution algorithms is divided into 2 main parts:

- Pupil detection
- Iris detection

This is done because of fact, that both parts are independent on each other, and that for each of them, need to be possible, to set different evolution parameters and to provide different fitness function design.

However both of them share same basis idea:

Both, iris and pupil boundary(circle) is evaluated as number of pixels, which are common with image, which contains highlighted pupil, respectively iris area. In other words the

⁷http://docs.opencv.org/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html

⁸<http://opencv-code.com/tutorials/pupil-detection-from-an-eye-image/>

candidate circle is as good as many pixels has it common with input image with which is evaluated.

Therefore the evolution algorithm design, to be more specific - in this case it's used evolution strategy concept, consists of these main sections, which have to be solved:

- Problem encoding
- Design of fitness function
- Evolution constraints/parameters setup

8.3.1 Pupil detection

For proper pupil detection was necessary to have an input image with pupil area highlighted. For this purpose is used thresholded image, which highlights very dark parts of image and therefore also pupil area. This image is then converted to edge map(using canny edge detector) and is further used as input for fitness function. Example of input image is shown in Figure 8.10

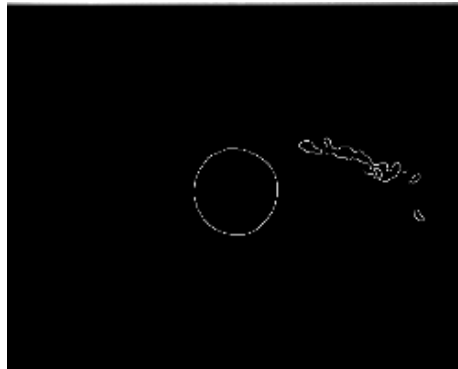


Figure 8.10: Input image for evolutionary pupil detection

Problem encoding Because fitness function evaluates individuals, which are circles over 2-dimensional space, following encoding was implemented. The whole genotype consists of these main parts, which are encoded into integer values:

- x-coordinate of the circle center
- y-coordinate of the circle center
- radius of circle
- number of pixels of which is circle shifted in x-direction
- number of pixels of which is circle shifted in y-direction
- radius adjust

Each of these values has its range set, which can be modified, in order to reduce the search space.

Design of fitness function Fitness function works in following steps:

- Obtains candidate circle parameter from genotype
- Computes common pixels with input image
- Computes fitness value

Fitness value is adjusted to extent, that also center point - obtained in distance transformation enters the fitness computation process, when circles, which are far from possible center are being penalized. The final fitness value is then computed as:

$$value = common_pixels - d(dist, center)^2 \quad (8.2)$$

where $d(dist, center)$ denotes the distance between center obtained from distance transformation and candidate circle center.

Evolution constraints/parameters setup The setup of evolution computation allows to setup these parameters:

- Population size - number of candidate circles in each generations
- Offspring fraction - how many offsprings will be selected into next generation
- Offspring selector - which type of selection mechanism will be used for offspring selection
- Survivors selector - which type of selection mechanism will be used for parents(survivors) selection
- Mutator - which type of mutation will be used
- Alterer - which type of Crossover technique will be used

8.3.2 Iris detection

For proper Iris detection was necessary to obtain image which highlights outer iris area. Also as by pupil detection, as input image was selected binary image with proper threshold setting. However, since Iris area is not as easy to be distinguished in image as pupil area, lot of information in thresholding was lost. Therefore I recommend different approach as this depends heavily on light conditions in which was input image taken. In Figure 8.11 is shown example of such an image.

Problem encoding Problem encoding is the same as by pupil detection, but with that difference, that x and y coordinates of pupil circle are computed only in area, close to candidate center, obtained from distance transformation 8.2.1.

The whole genotype consists though of these main parts, which are encoded into integer values:

- radius of circle
- number of pixels of which is circle shifted in x-direction prior to candidate center obtained from distance transformation



Figure 8.11: Input image for evolutionary iris detection

- number of pixels of which is circle shifted in y-direction prior to candidate center obtained from distance transformation
- radius adjust

Design of fitness function Design of fitness function is the same as in pupil boundary detection process. [8.3.1](#)

Evolution constraints/parameters setup Evolution parameters are the same as in pupil boundary detection process. [8.3.1](#)

Chapter 9

Testing

9.1 Data

For purpose of this work was used the BioSecure Multimodal Database (BMDB)¹. This database consists inter alia, of iris database, against which was implemented algorithm tested. The iris database corresponds to 210 users of desktop dataset. For each person and for each session, 4 grey-level images of the 2 eyes (left and right) are available. In general, the pupil and the iris of the eye are centered in the image. There are 2 sessions for each person.

In the database, some images have an interlacing problem as visible on the Figure 9.1.

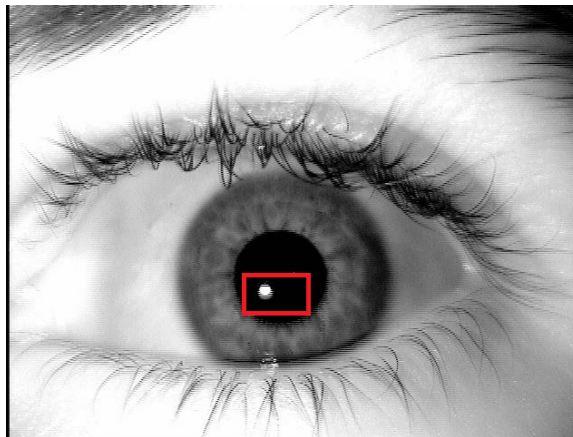


Figure 9.1: Original image presenting interlacing distortions square

9.2 Results

9.2.1 Parameters setup

The testing of the algorithm was done with following parameters setup.

Pupil detection

¹<http://biosecure.it-sudparis.eu/AB/>

- Number of generations: 20
- Population size: 500
- Offspring fraction: 0.75
- Offspring selector: Tournament selector - size 3
- Survivors selector: Tournament selector - size 3
- Mutation - Gaussian mutation - probability 0.18
- Crossover operator : Mean crossover - probability 0.5

These parameters was selected after several test attempts. During the testing was found, that in pupil detection process, lower size of tournament selector set leads to better results. This is with high probability due to fact, that with this parameter setup is search space examined more widely and doesn't stuck quickly in local maximum. Also for accessing wide range of searched values, relative high crossover and mutation probability was set.

Iris detection

- Number of generations: 20
- Population size: 200
- Offspring fraction: 0.8
- Offspring selector: Tournament selector - size 10
- Survivors selector: Tournament selector - size 10
- Mutation - Gaussian mutation - probability 0.18
- Crossover operator : Mean crossover - probability 0.5

Parameters setup for Iris detection is based on previous testing of pupil boundary detection. Therefore wide range of candidate circles are tested through evaluation process in conjunction with small generations count, as evolution converges in relatively short amount of generations.

9.2.2 Results

Overall results from test show that approximately 75% of tested images were properly selected using evolution algorithm approach. This may differ from parameters settings, as for larger population are more candidate circles checked, but on the other hand, it consumes lot of resources. For given parameters took evaluation of 1 image approx. 7 seconds and this should be improved in future work. However its satisfying that in some cases, even when there is small information, evolution algorithm can find better solution than implemented convention method. On Figure 9.2 and Figure 9.3 is shown result from one single test session. In table 9.1 are results compared to Wildes segmentation method[15]

	Proposed Method	Wildes Method
Segmentation rate	75%	93.17%
Error rate	25%	6.83%

Table 9.1: Methods result comparison [15]

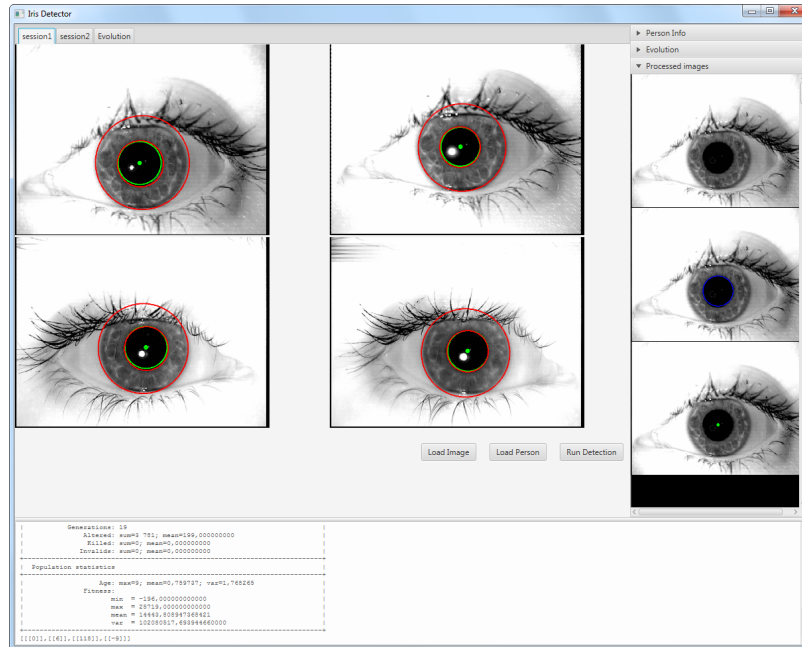


Figure 9.2: Result image - Green: convention solution, Red: evolutionary computed solution

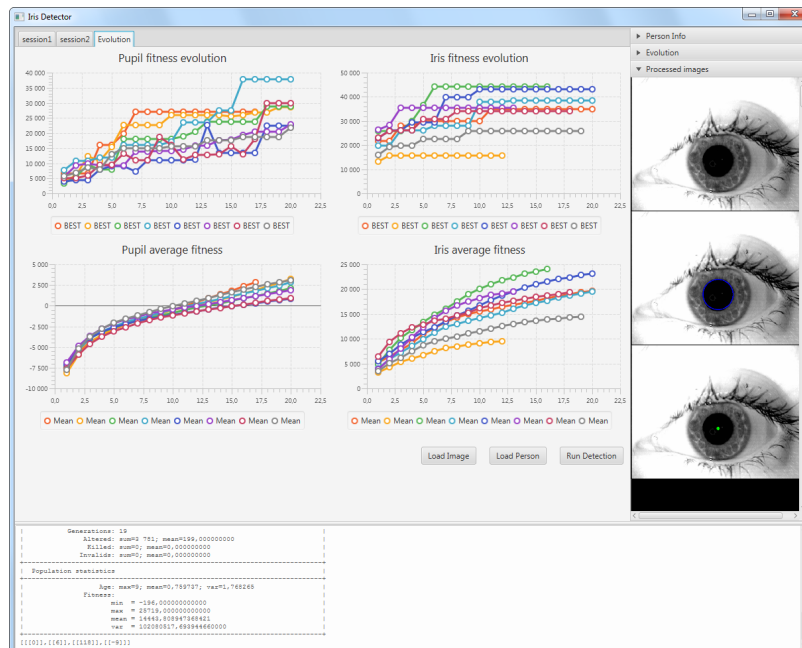


Figure 9.3: Evolution results

Chapter 10

Conclusion

10.1 Summary

The overall results from implemented solutions shows, that evolution algorithms can be used, with considerable success, in field of human iris detection. However, what need to be discussed is their use, due to their time complexity. It need to be said, according to implemented solution , that this depends solely on input image i.e. some reference obtained by convention methods, and therefor need to be reviewed, what are other options to obtain input reference for evolution process. To summary this, evolution algorithms in conjunction with convention methods are suitable for those problems solving, especially in cases, where significant part of information is missing, or is not clear.

Bibliography

- [1] “Biometrics,” 2001-. <http://en.wikipedia.org/wiki/Biometrics>.
- [2] “Biometric systems and security design principles.” http://wiki.cas.mcmaster.ca/index.php/Biometric_Systems_and_Security_Design_Principles.
- [3] “Biometrics definition.” <http://searchsecurity.techtarget.com/definition/biometricss>.
- [4] “Iris (anatomy),” 2001-. [http://en.wikipedia.org/wiki/Iris_\(anatomy\)](http://en.wikipedia.org/wiki/Iris_(anatomy)).
- [5] J. Schwarz and L. Sekanina, “Aplikované evoluční algoritmy,” 2006.
- [6] M. Drahanský, F. Orság, and kol., *Biometrie*. Computer PRESS, Brno, 2011.
- [7] M. Drahanský, *Biometrické systémy*. FIT VUT Brno, 2011.
- [8] G. Williams, *Iris Recognition Technology*. Aerospace and Electronic Systems Magazine, 2011.
- [9] M. Burge and K. Bowyer, *Handbook of iris recognition*. London: Springer London, 2013.
- [10] R. Jillela and A. A. Ross, “Methods for iris segmentation,” *Handbook of Iris Recognition*, p. 239, 2013.
- [11] S. Li and A. K. Jain, *Encyclopedia of biometrics*. New York: Springer, c2009.
- [12] J. Sarvaiya, *Advances in Computing, Control*,. Piscataway: IEEE, 2009.
- [13] M. B. P. Carneiro, A. C. P. Veiga, F. C. Castro, E. L. Flores, and G. A. Carrijo, “Application of evolutionary algorithms for iris localization,” *2009 IEEE International Conference on Multimedia and Expo*, pp. 506–509, 2009.
- [14] M. Shamsi and A. R. Kenari, “Iris boundary detection using an ellipse integro differential method,” *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, pp. 1–5, 2012.
- [15] B. Sareni and L. Krahenbuhl, “Fitness sharing and niching methods revisited,” *IEEE Transactions on Evolutionary Computation*, vol. vol. 2, no. issue 3, pp. 97–106, 2002.
- [16] J. Daugman, “How the afghan girl was identified by her iris patterns.” online, 2002-04-01.

- [17] L. Masek, "Recognition of human iris patterns for biometric identification," Master's thesis, The University of Western Australia, 2003.
- [18] A. Muroň and J. Pospíšil, "The human iris structure and its usages." online.
- [19] J. Daugman, "How iris recognition works," tech. rep., University of Cambridge, 2004.
- [20] E. M. Arvacheh, "A study of segmentation and normalization for iris recognition systems," Master's thesis, University of Waterloo, 2006.
- [21] C. loic Tisse, L. Martin, L. Torres, and M. Robert, "Person identification technique using human iris recognition," tech. rep., 2002.
- [22] S. Lim, K. Lee, O. Byeon, and T. Kim, "Efficient iris recognition through improvement of feature vector and classifier," tech. rep., 2001.
- [23] C. Rathgeb, A. Uhl, and P. Wild, "Iris segmentation methodologies," in *Iris Biometrics*, p. 49, London: Springer London, 2013.
- [24] J. Holland, "Building blocks, cohort genetic algorithms, and hyperplane-defined functions," *Evolutionary Computation*, vol. 8, pp. 373–391, Dec 2000.

Appendix A

Obsah CD

- *IrisDetectorFX/* - contains source codes and runnable distribution archive. More info in *IrisDetectorFX/README* file
- *results/* - Contains results from two test sessions
- *projekt.pdf*