



HAL
open science

Segmentation of irises acquired in degraded conditions

Thierry Lefèvre

► **To cite this version:**

Thierry Lefèvre. Segmentation of irises acquired in degraded conditions. Other. Institut National des Télécommunications, 2013. English. NNT : 2013TELE0005 . tel-01726957

HAL Id: tel-01726957

<https://theses.hal.science/tel-01726957v1>

Submitted on 8 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THALES



**DOCTORAT EN CO-ACCREDITATION
TELECOM SUDPARIS ET L'UNIVERSITE EVRY VAL D'ESSONNE**

**Spécialité :
Informatique**

Ecole doctorale : Sciences et Ingénierie

Présentée par

Thierry Lefèvre

**Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS**

Segmentation d'Iris Acquis En Conditions Dégradées

Soutenue le 30 Janvier 2013

devant le jury composé de :

**Professeur Bernadette Dorizzi,
Professeur Sonia Garcia Salicetti
Madame Nadege Lemperiere
Professor Natalia A. Schmid
Professeur Florence Rossant
Professeur Jean-Luc Dugelay
Professeur Hugo Proença**

**Directrice de thèse
Encadrante de thèse
Encadrante de thèse
Rapporteur
Rapporteur
Examineur
Examineur**

Thèse n°2013TELE0005

CONFIDENTIAL

CONFIDENTIAL

Introduction and Context of the Thesis

Biometrics aims at identifying people based on their intrinsic morphological or behavioral characteristics. These characteristics can be face shape, fingerprint, hand shape, gait, signature. The key idea of biometrics is to guarantee the identity of people based on what they are instead of what they possess (an ID card) or what they know (a password). In this thesis we have studied the identification of people based on the texture of their irises, namely, iris recognition.

Iris recognition is a relatively young biometric modality : the first automatic recognition system able to identify people based on their iris texture was proposed by John Daugman in [16] in 1993. Despite its youth, iris recognition was very quickly seen as a very interesting biometric modality for critical applications. This modality has several interesting properties :

- Iris texture is epigenetic, meaning that it is not entirely determined by genotype. As a consequence, the two irises of a single person differ, and so do twins' irises.
- It has been assessed that iris texture is rich enough to uniquely characterize each individual in a very large population.
- Iris is an internal organ accessible without contact, making it suitable for user friendly application.
- It is hard to fake someone's identity using a synthetic lens. Liveliness detection is easily performed in real applications because the pupil always has high frequency contractions/dilations that can be detected and extracted from a video stream.
- Iris texture is known to be stable over time and very well protected from external degradations (though, this last claim has been challenged in [20]).

The recognition rate is especially critical when working on very large databases because the number of errors produced by the system is proportional to the size of the database. To this end, industrial companies have been constantly working on trying improving their algorithms to decrease the error rate. Accordingly, it is important for industrial companies to be able to transfer new results proposed by the research community into robust, accurate and scalable industrial systems.

The proposed thesis has been conducted as part of a partnership between the industrial company *Thales* and *Telecom Sud-Paris*. The purpose of this partnership was to transfer the academic knowledge of *Telecom Sud-Paris* on iris recognition to *Thales* in order to develop an industrial product. The first step of this collaboration was to study the limitations of traditional iris recognition systems and to propose some possible innovations to solve them. A classical iris recognition system usually follows the main steps proposed by Daugman's in [16].

Image Acquisition : Image acquisition is done under Near Infra Red (NIR) illumination, having wavelengths between 700 and 900 nm. At these wavelengths even dark brown irises show a very rich texture which is suitable for recognition. In a standard controlled acquisition scenario, the subject is asked to stand still and look straight at the camera from a short distance. However, recent works tend to relax acquisition conditions. For example an image can be acquired at a distance [4] or using a visible wavelength [60]. Subjects are also less constrained : they do not have to look straight at the camera [10] or they may move during the acquisition [43].

Image Segmentation : Given the acquired eye image, the first algorithmic task is the segmentation of the iris, aiming at isolating iris texture from other elements of the image such as eyelids, shadows or glasses. Segmentation is challenging as the more the acquisition conditions are relaxed, the more degradations have to be handled at this stage.

- Texture Normalization* : Texture is mapped into a dimensionless coordinate system to handle variability in the eye image, such as pupil dilation. The most common choice for normalization is the rubber sheet model introduced by Daugman in [16]. Iris borders are modeled by two non-concentric circles and texture is unwrapped with respect to these circles. Precision is a critical issue at this stage as small errors in the estimation of the circles' parameters can dramatically decrease performance of the overall system as outlined by Proenca in [58].
- Feature Extraction* : The discriminative iris texture's features are extracted. These features are the basis of the comparison of iris images in order to perform the comparison. The most used features are based on a quantization of Gabor filters' phase. This quantization generates a binary code characterizing the iris.
- Pattern Matching* : Comparison of the input features to a reference in order to decide if the two images come from the same iris. In systems based on Daugman [16] the comparison is made by computing the Hamming distance of the two binary codes, which are characteristic of the two irises. The Hamming distance between two binary codes is low when they come from the same iris and high when they come from different irises. Therefore, the key aspect at this stage is to fix a threshold in order to decide whether or not the two binary codes come from the same iris.

In his early article [16], Daugman assumes that the iris has been acquired under a constrained acquisition scenario. This means that the subject is active and fully cooperative during the acquisition. However, a recent trend in iris recognition systems is to reduce the acquisition constraints in order to make the systems more user friendly. This could allow to decrease check-in time to get in airplanes for instance. Relaxing the acquisition conditions generates degradations in the image, such as blur or illumination inhomogeneities. As such degradations appear at the first stage of the system, they tend to be increased at each subsequent stage. Hence, there is an important degradation of the system's recognition performance when not designed to handle such data.

Two main approaches have been considered in the literature to adapt iris recognition systems to degraded images. The first approach is to extract more information from the normalized image. This allows the extracted features to remain discriminative even when the amount of the biometric information is reduced. The second approach is to consider new models for segmentation that can more robustly identify the iris region and borders in the image.

In this context, the company *Thales* dedicated two theses to these two subjects. The first one, by Sandra Cremer [13], focuses on the improvement of the *Feature Extraction* and the *Pattern Matching* stages of the system. The second one, here presented, focuses on the *Image Segmentation* and *Texture Normalization* stages. The two theses have been conducted simultaneously and have complementary results. Indeed, the first thesis focuses on an effective exploitation of the biometric information whereas the second one aims at formatting this biometric information as well as possible. The final goal of these two theses is to develop a complete software working with images acquired in relaxed acquisition conditions for fast border crossing. However, no acquisition systems were available during the theses. Therefore, we decided to define the ICE2005 and the ND-Iris databases as references for our experiments because they contain most of the degradation we expect to have on the target system such as blur or off-angle images.

This thesis focuses on the *Image Segmentation* and *Texture Normalization* stages. Our purpose is to generate an accurate normalized image with an indication of the texture's areas hidden by occlusions. Two distinct actions are required to produce such information :

- The identification of pixels belonging to the iris in the eye image (generating a so called segmentation mask)

- The estimation of a parametric description of the inner and outer iris boundaries in the image, but also under possible occlusions

These two piece of information are critical in order to carry out recognition. The parametric contours are used to unwrap the iris texture to produce the normalized image which is then used for *Feature Extraction*. The segmentation mask is used to remove artifacts from the normalized image at the *Pattern Matching* stage.

During this thesis we developed a complete segmentation system that can work with degraded data. We present this complete system in Annexe A, but in the core document focus on 4 critical issues for which we develop some novel solutions :

- Rough localization of the pupil in the eye image in order to initialize the complete segmentation process.
- Accurate detection of the pixels belonging to the iris texture in the image, i.e. iris region segmentation.
- Estimation of the iris borders under possible occlusions in order to generate the normalized image.
- Evaluation of the quality of segmentation results in order to identify failures of the segmentation process.

The pupil is considered as the most stable element in an eye image acquired in near infrared illumination. Indeed, the pupil always appears as a very dark region inside the iris. Accordingly, most of the algorithms in the literature first roughly localize the pupil area in the image in order to initialize some more complex algorithms. The difficulty of this localization is to be able to distinguish the pupil from other dark elements in the eye such as eyebrows, packed eyelashes, hair or the arms of the glasses. To solve this issue, we propose firstly to identify the regions of eyebrows and eyelashes and then remove them from the pupil localization process.

The next task is the identification of the pixels belonging to the iris texture in the image. The iris texture can vary largely from one person to another and is dependent on acquisition factors, such as blur or illumination. Therefore, the algorithm must be very robust to handle this variability in the image. Moreover, it must also be very precise as inaccuracies will provide wrong information to the *Pattern Matching* algorithms, leading to erroneous comparison results. Accordingly, we have decided to study a classical family of segmentation algorithms that have become popular for iris segmentation [17, 76, 66, 62] : Active Contours. These kinds of segmentation algorithms model the expected solution as being the minimum of a given functional. This minimum is found by iteratively morphing a contour until it reaches the minimum of the energy. However, despite this growth in popularity, there is no comparative evaluation of such algorithms in the literature. Therefore, we implemented several Active Contour algorithms and discuss their advantages and limitations

The final step of iris segmentation is the estimation of the iris shape under possible occlusions. This task is critical because this estimation of the iris borders is used to produce the normalized image that is used for the *Feature Extraction* and *Pattern Matching* stages. Small inaccuracies in the estimation of these borders result in a large corruption of the normalized image, leading to a significant decrease of the system's recognition performance. However, this estimation of the iris border can be difficult when acquisition conditions are relaxed. The iris is more likely to be occluded by eyelids and the contours may have a non-circular shape due to gaze deviation. Therefore, we propose a new robust and effective way to fit ellipses on the iris border in order to generate the normalized image.

This thesis is conducted from an industrial perspective. In such context, it is mandatory to detect erroneous results of automatic algorithms, like segmentation failures. Unlike the pupil and iris segmentation issues that are widely studied in the literature, estimating the quality of an iris segmentation have not been widely studied so far. Indeed, few works in the literature focus on this specific subject [38, 29, 81, 30, 80].

The outline of this thesis is divided into two main parts. In the first part (Chapter 1 to 3), we describe some general considerations about iris recognition systems and in the second part (Chapter 4 to 7), we describe more deeply the specificity of this thesis.

In Chapter 1, we present in detail a typical iris recognition system based on the ideas of Daugman [16]. We describe each module and we explain why the degradation of the input data quality has such a negative impact on this kind of system.

In Chapter 2, we explain how to evaluate the recognition performance of a biometric system, with a focus on iris recognition systems. Then, we describe how we can use the system's recognition performance to evaluate the accuracy of an iris segmentation algorithm. Finally, we describe the databases we used during this thesis to evaluate our algorithms.

In Chapter 3, we make a survey of classical iris segmentation methods available in the literature. We propose a distinction between two main families : *Contours First* strategies and *Texture First* strategies. *Contours First* strategies first search for the parametric contours of the iris borders, and then estimate the pixels belonging to the iris texture by removing the occlusions. *Texture First* strategies first identify accurately the iris region in the image and then use this information to estimate the parametric contours of the iris under possible occlusions. These two strategies have their advantages and shortcomings which we analyze. In this thesis, we decided to apply a *Texture First* strategy to segment the iris texture ; at the end of this chapter, we explain why we made this choice and the impacts it had on the design of our system.

In Chapter 4, we describe the algorithms we implemented to roughly localize the pupil area in the eye image. The main difficulty at this stage is to distinguish the pupil from other dark elements in the eye image. Therefore we propose two algorithms for identifying eyelashes and eyebrows in order to remove them from the localization process. Once these areas have been removed, it is possible to identify the pupil area with simple algorithms such as thresholding. We evaluate the impact of our two eyelash and eyebrow detectors on two databases with different kind of degradations.

In Chapter 5 we describe the algorithms we implemented to segment the iris region in the image. We have decided to focus on Active Contours for iris segmentation. Basically, these algorithms evolve by morphing a contour onto the object to be segmented. Therefore, there are two elements to define in order to apply such an algorithm : the curve's representation and the object's model. In this chapter, we propose to study the influence of these two elements on the system's recognition performance.

In Chapter 6, we describe the contour fitting algorithm we designed to estimate the borders of the iris under possible occlusions. Our algorithm results from the formalism of Active Contours presented in Chapter 5 for iris segmentation. However, the purpose of our algorithm is to estimate the location of the iris border under possible occlusions instead of identifying pixels that belong to the iris in the image. Our algorithm iteratively evolves an elliptic contour until it best fits the iris borders' location.

In Chapter 7, we describe the algorithm we designed to estimate the quality of a given segmented image. We first make a brief survey on the evaluation of iris quality. Then, we present a new set of segmentation quality measures specifically designed for iris segmentation. This set of individual measures is used to predict the intrinsic tendency of the segmented image to produce erroneous comparisons.

In Chapter 8, we summarize all the contributions of this thesis and give some perspectives for further works.

Remerciements

Je tiens à remercier tous ceux qui m'ont permis de mener à bien cette thèse.

Je tiens en premier lieu à remercier ma directrice de thèse Bernadette Dorizzi, pour m'avoir encadré et soutenu au cours de ces trois années. J'ai trouvé les échanges que nous avons pu avoir extrêmement enrichissants et ils m'ont souvent amené à pousser plus loin mes analyses et mes recherches, ce qui a énormément contribué à la réussite de cette thèse.

Je tiens également à remercier Sonia Garcia pour son encadrement. Son esprit d'analyse et son sens du détail m'ont beaucoup apportés, en particulier en ce qui concerne la rédaction du présent manuscrit que ses remarques ont grandement contribué à améliorer.

Je remercie également toute l'équipe de Thales pour m'avoir accueilli au cours de ces trois ans, avec une pensée particulière pour mes trois encadrants Valérie Letournel, Nadège Lempérière et Stéphane Bélardi. Valérie pour avoir initié cette thèse et permis son démarrage dans les meilleures conditions. Stéphane pour son investissement sincère et constant au bon déroulement de la thèse. Nadège, en premier lieu pour ses conseils et remarques techniques, mais surtout pour son investissement et ses encouragements qui m'ont soutenu tout au long de ma thèse.

Je remercie également tous les autres membres de l'équipe : Philippe Robin, Cyrille Bichon, Christophe Dieupart, Florian Hadjadj, Sandra Marti, Sandra Cremer, Nicolas Vervelle, Yves Legier, François Ribière, Michel Dieumegard, Mathieu Seillier, Aline Heuze, Vincent Monier, Nathalie Dumas, Silvia Gandy et Victor Leyva Sifuentes. J'ai vraiment pris un grand plaisir à travailler au sein de cette équipe dont j'ai pu apprécier la gentillesse et la convivialité et qui ont su rendre agréable chaque jour de travail. J'ajoute un remerciement particulier pour Silvia pour avoir relu mon manuscrit de thèse.

Je souhaite encore remercier les stagiaires qui ont travaillé dans le cadre de la reconnaissance d'iris. En premier lieu Boris Tchorbadjian qui a développé le premier prototype de démonstrateur iris. Ensuite Christophe Renoir pour s'être chargé de la fastidieuse tâche d'annoter manuellement une base de données. Enfin, Jean-Charles Bricolas qui a travaillé sous ma direction dans le but d'améliorer les algorithmes de segmentation d'iris. J'ai particulièrement apprécié sa curiosité et sa motivation qui l'ont amené à travailler bien au-delà du cadre initial de son stage. Ainsi, son travail m'a permis de mettre en lumière certains des résultats discutés dans cette thèse.

Je remercie également mes collègues de Telecom Sud Paris, en particulier Nesma Houmani et Guillaume Sutra. Nesma pour ses conseils et ses avis, Guillaume pour les échanges constructifs que nous eut sur la segmentation de l'iris. Je remercie également Phan Viêt Anh pour le développement du logiciel OSIRIS qui m'a servi de base de travail au début de ma thèse.

Je remercie Natalia Schmid et Florence Rossant pour avoir accepté d'être rapporteurs pour ma thèse, ainsi que Jean-Luc Dugelay et Hugo Proença pour avoir accepté de faire partie de mon jury.

Pour finir, je tiens à remercier mes proches qui m'ont apporté un soutien sans faille au cours de ces trois ans. Mes amis pour m'avoir permis de prendre de la distance vis-à-vis de mon travail et de me changer les idées dans les moments difficiles. Ma famille pour avoir tout fait pour m'aider au cours de cette thèse, mais également tout au long de ma scolarité. En effet, c'est à eux que je dois d'avoir pu m'investir si totalement et sereinement dans mes études, et cette réussite est pour beaucoup la leur.

Contents

| | |
|---|-----------|
| Introduction and Context of the Thesis | 2 |
| I General Considerations About Iris Recognition | 9 |
| 1 A Classical Iris Recognition System | 12 |
| 1.1 Image Acquisition | 14 |
| 1.2 Image Segmentation | 14 |
| 1.3 Texture Normalization | 15 |
| 1.4 Feature Extraction | 16 |
| 1.5 Pattern Matching | 16 |
| 1.6 Limitations of Classical Iris Recognition Systems | 18 |
| 1.7 Conclusion | 21 |
| 2 Evaluation of Iris Recognition Systems | 22 |
| 2.1 General Evaluation of Biometric Systems | 22 |
| 2.2 Evaluation of Segmentation Quality | 25 |
| 2.3 Databases | 28 |
| 2.4 Conclusion | 30 |
| 3 Segmentation Techniques for Iris Recognition | 32 |
| 3.1 Early Segmentation Approaches | 33 |
| 3.2 Contour First Strategy | 34 |
| 3.3 Texture First Strategy | 37 |
| 3.4 Focus of this Thesis | 39 |
| II Contributions of this Thesis | 41 |
| 4 Automatic Localization of the Pupil Area in the Eye | 44 |
| 4.1 Segmentation of eyelashes and eyebrows | 44 |
| 4.2 Rough Pupil Segmentation | 50 |
| 4.3 Evaluation | 52 |
| 4.4 Conclusion | 53 |
| 5 Iris Region Segmentation | 54 |
| 5.1 Deformable Models | 54 |
| 5.2 <i>Parametric Contours</i> | 58 |
| 5.3 <i>Geometric Contours</i> | 65 |
| 5.4 Texture Based Active Contours | 73 |
| 5.5 Evaluation | 80 |
| 5.6 Conclusion | 86 |

| | | |
|----------|---|------------|
| 6 | Contour Fitting for Normalization | 88 |
| 6.1 | Parametric Deformable Template Optimization | 90 |
| 6.2 | Model Initialization | 95 |
| 6.3 | Evaluation | 96 |
| 6.4 | Conclusion | 101 |
| 7 | Image Quality and Automatic Detection of Segmentation Failures | 102 |
| 7.1 | What Is Iris Data Quality? | 104 |
| 7.2 | Rough Region Segmentation | 107 |
| 7.3 | Quality Metrics | 111 |
| 7.4 | Failure Detection | 125 |
| 7.5 | Evaluating the Impact of the Fused Quality Score | 129 |
| 7.6 | Conclusion | 134 |
| 8 | Conclusion | 136 |
| | Bibliography | 140 |
| | III Appendix | 145 |
| A | Modular System for Iris Segmentation | 148 |
| A.1 | System Overview | 148 |
| A.2 | Preprocessing | 150 |
| A.3 | Pupil Segmentation | 152 |
| A.4 | Iris Segmentation | 153 |
| A.5 | Segmentation Quality | 155 |
| B | Variational Contour Optimization | 158 |
| B.1 | General elements | 158 |
| B.2 | Edge energy E_{edge} | 158 |
| B.3 | Region energy E_r | 160 |
| B.4 | Regularization Energy E_s | 163 |
| B.5 | Circular model | 164 |

Part I

General Considerations About Iris Recognition

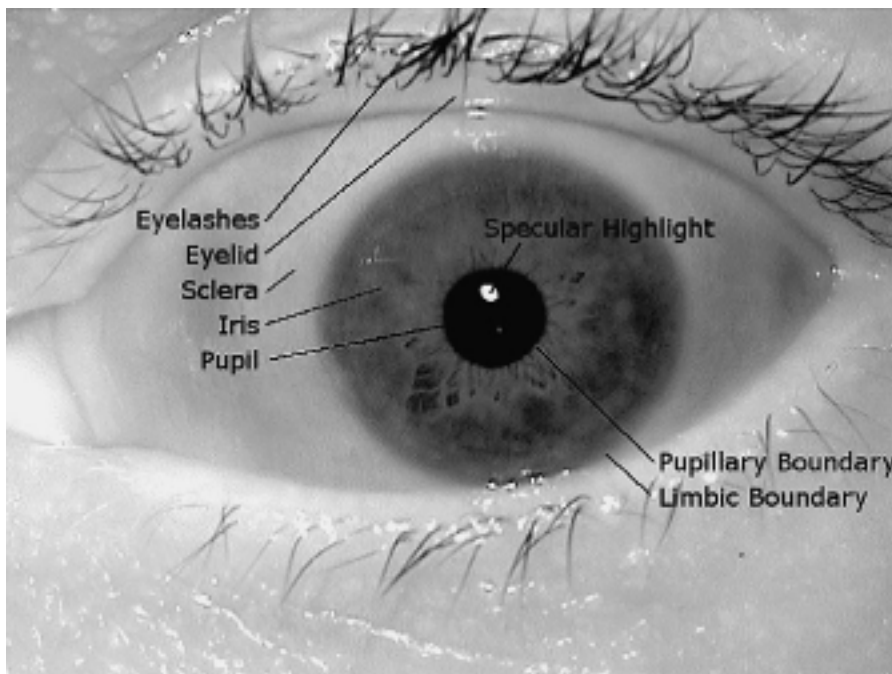
CONFIDENTIAL

CONFIDENTIAL

Chapter 1

A Classical Iris Recognition System

The iris is the annular colored part between the pupil and the limbus (also called sclera). It acts as a diaphragm regulating the amount of light entering the eye. The possibility to identify human beings using their irises was proposed by an ophthalmologist, Frank Burch, in 1936. However, the first patent describing a conceptual iris-based recognition system was released almost 50 years later by Flom and Safir in 1987 [23]. It took another decade to get the first automatic system able to identify people based on the texture of their irises. This pioneer work was proposed by John Daugman in 1993 [16] and was followed by Wildes' system in 1997 [77], based on a different recognition algorithm.



Most actual iris recognition systems are based on the earliest works of Daugman [16] and are usually divided in five main parts as illustrated in Figure 1 :

- Image Acquisition* : Acquisition of a high resolution eye image containing the iris texture.
- Image Segmentation* : Detection of the iris region in the image and estimation of iris borders' shape under possible occlusions.
- Texture Normalization* : Unwrapping of the iris texture to generate a "dimensionless" image in polar coordinates.
- Feature Extraction* : Extraction of the features characterizing the iris texture.
- Pattern Matching* : Comparison of the input features to reference ones in order to decide if the two images come from the same iris.

In this chapter, we present this classical iris recognition system. From Section 1.1 to 1.5, we briefly describe each of the above modules. In Section 1.6, we present some limitations of these traditional iris recognition systems.

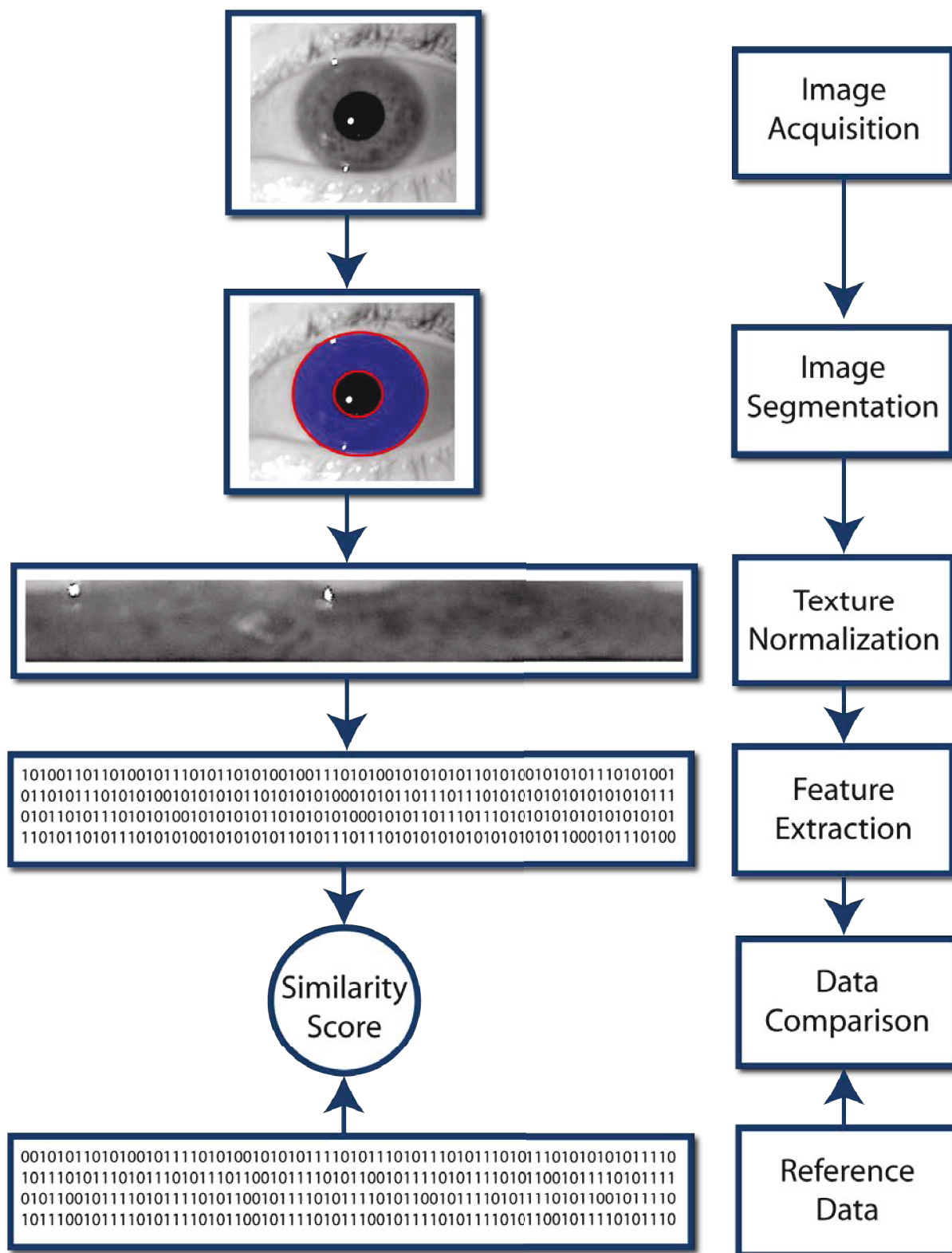


Figure 1: A standard iris recognition system. The image is first acquired in Near infrared (NIR) illumination. Iris texture is then segmented and normalized (unwrapped). Characteristic features of the texture are extracted and represented using a binary code (called iris code in the literature). The comparison of two irises is done by computing the Hamming distance between these two codes.

1.1 Image Acquisition

Acquisition of high quality iris image is a challenging task because the iris represent a very small portion of the whole face. The image acquisition uses Near Infra Red (NIR) illumination, having wavelengths between 700 and 900 nm. At those wavelengths even dark brown irises show very rich texture suitable for recognition. The cornea is also permeable to these wavelengths which avoid illumination reflections in the image. In a standard controlled acquisition scenario, the subject is asked to stand still and look straight at the camera from a short distance ($\approx 20\text{-}50\text{cm}$). Figure 2 shows some examples of such acquisition devices.



Figure 2: Examples of iris acquisition devices. Figure (a) and (b) : the stop and stare sensor LG2200 from LG with an acquisition range between 8cm - 25cm. Figure (c) : the hand held sensor OKI IRISPASS-h from OKI.

The sensors showed in Figure 2 can acquire image of very good quality, but they lack of user-friendliness. Therefore, the design of new sensor allowing a relaxation of the acquisition conditions has been an active way of research during the last years. For example, images can be acquired at a distance [4] or using visible wavelength [60]. The subject is also less constrained : it may not look straight at the camera [10] or move during the acquisition [43].

1.2 Image Segmentation

Once the eye image has been acquired, the iris region in the image shall be identified in the image in order to limit the *Feature Extraction* and the *Pattern Matching* to this region. It is also necessary to get an accurate parametric model of the iris borders under possible occlusions in order to apply the *Texture Normalization* that we describe in the next section. These two actions (identification of the iris region and estimation of the iris borders) are called iris segmentation. The binary mask indicating which pixels of the image belong to the iris texture is called the segmentation mask. Figure 3 illustrates the result of the segmentation process on a good quality image.

The first methods of Daugman and Wildes model the iris using two non-concentric circles. Accordingly, they reduce the segmentation stage to a circle detection problem. However, these simple methods have shown a high sensitivity to image degradations. It is also commonly accepted that circles are only an approximation of the iris true shape and that more complex models should be used to describe the iris borders. We discuss this topic in detail in Chapter 3

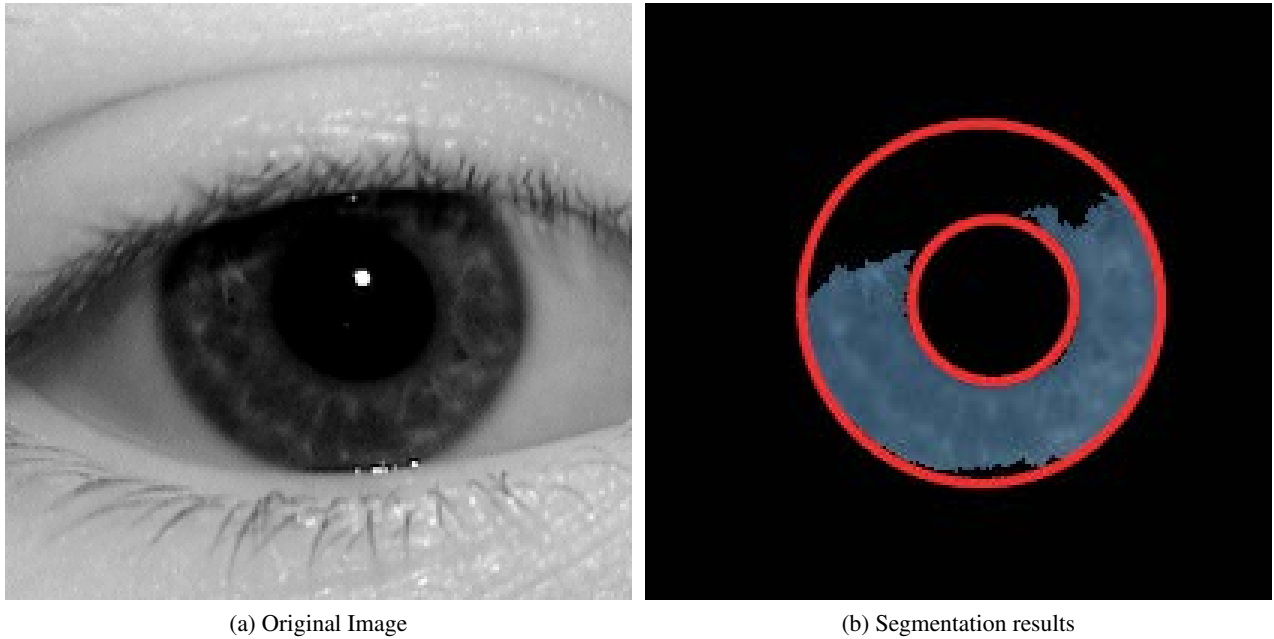


Figure 3: Illustration of the segmentation process results. The segmentation results contains both the region of the iris texture in the image and a parametric description of the iris borders under possible occlusions.

1.3 Texture Normalization

One of the main problems appearing when trying to characterize the iris texture is the deformations due to contraction and dilation of the pupil. In [16], Daugman propose to use a "rubber sheet" model to map the iris texture into a dimensionless coordinate system. Such mapping can be seen as an unwrapping of the image region contained between the inner and outer contour of the iris, as illustrated in Figure 4. The main advantage of such a mapping is that the texture displayed in the normalized image is independent of the pupil dilation.



Figure 4: Illustration of the unwrapping process of Daugman's normalization. On the left the segmented image, on the upper right the unwrapped texture of the iris and on the lower right the segmentation mask of the iris region

In the early work of Daugman [16], the pupil and iris borders are modeled using two non-concentric circles $\mathcal{C}_p(x_{cp}, y_{cp}, r_p)$, the pupil circle and $\mathcal{C}_i(x_{ci}, y_{ci}, r_i)$, the iris circle with parametrization :

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix}(\theta) = \begin{pmatrix} x_{cp} \\ y_{cp} \end{pmatrix} + r_p \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix}(\theta) = \begin{pmatrix} x_{ci} \\ y_{ci} \end{pmatrix} + r_i \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$$

with $\theta \in [0, 2\pi[$. The formula to unwrap the annular part between the two circles is then :

$$\begin{pmatrix} x \\ y \end{pmatrix} (r, \theta) = (1 - r) \begin{pmatrix} x_p \\ y_p \end{pmatrix} (\theta) + r \begin{pmatrix} x_i \\ y_i \end{pmatrix} (\theta) \quad (2)$$

with $r \in [0, 1[$ and $\theta \in [0, 2\pi[$. This model can handle changes in pupil size, however modeling the iris border with circles is now known to be too restrictive, so new representations using more general contours have been introduced in the literature. The classical formulation (2) is therefore extended to :

$$R(r, \theta) = (1 - r)P(\theta) + rL(\theta), \quad (3)$$

where P stands for the parametric description of the pupil/iris border and L stands for the parametric description of the iris/limbus border. These parametric descriptions could be geometric shapes like ellipses, or more general models such as Fourier series expansions or splines.

Let us note that *Feature Extraction* and the *Pattern Matching* are applied to the normalized image. Therefore, precision of the unwrapping is a critical aspect of iris recognition systems. In [58], Proenca points out that small errors in the normalization process dramatically decrease the performance of the overall system.

1.4 Feature Extraction

The *Feature Extraction* stage aims at extracting the characteristics of a given iris. These characteristics results in a template that will be used to perform the recognition in the *Pattern Matching* stage.

In Daugman's based system [16], the template for the iris is a binary code encoding frequency information at several scales. This binary code (the so-called iris-code) is generated by applying a set of Gabor filters with different scales and orientations at predetermined points of the normalized iris image. The filter with maximal response characterizes the iris texture at a considered point. Formally a Gabor filter can be written as :

$$G_{\{Re, Im\}} = sgn_{\{Re, Im\}} \int_{\rho} \int_{\phi} I(\rho, \phi) e^{i\omega(\theta_0 - \phi)} \cdot e^{-\frac{(r_0 - \rho)^2}{\alpha^2}} e^{-\frac{(\theta_0 - \phi)^2}{\beta^2}} \rho d\phi d\rho, \quad (4)$$

where $I(\rho, \phi)$ is the image intensity of the normalized iris image at pixel (ρ, ϕ) (normalized image can be seen as a polar image, hence the notations), α and β are the scale parameters, ω is the wavelet frequency, and (r_0, θ_0) are the coordinates of the application point in the normalized image. Gabor filters have a complex formulation, therefore they provide both phase and module information. In [16], Daugman assesses that the phase of the Gabor filters is a relevant biometric information to characterize an iris. In order to compress this phase information, Daugman proposes to consider only the quadrant of the phase, so that this information can be coded using only two bits. The unwrapped segmentation mask is also binarized in order to indicates significant bits in the iris code.

Figure 5 summarizes the process described above to generate the iris code. Gabor filters are applied at specific points of the normalized image. The phase of the filter with strongest response is compressed into two bits. The process is repeated until all the analysis points have been covered.

1.5 Pattern Matching

The final step of iris recognition systems is to determine if two iris templates belong to the same iris. Comparison of two iris templates is done by computing the hamming distance of the two binary codes. Figure 6 illustrates such a computation for a genuine (iris codes generated from the same iris) and an impostor comparisons (iris codes generated from different irises).

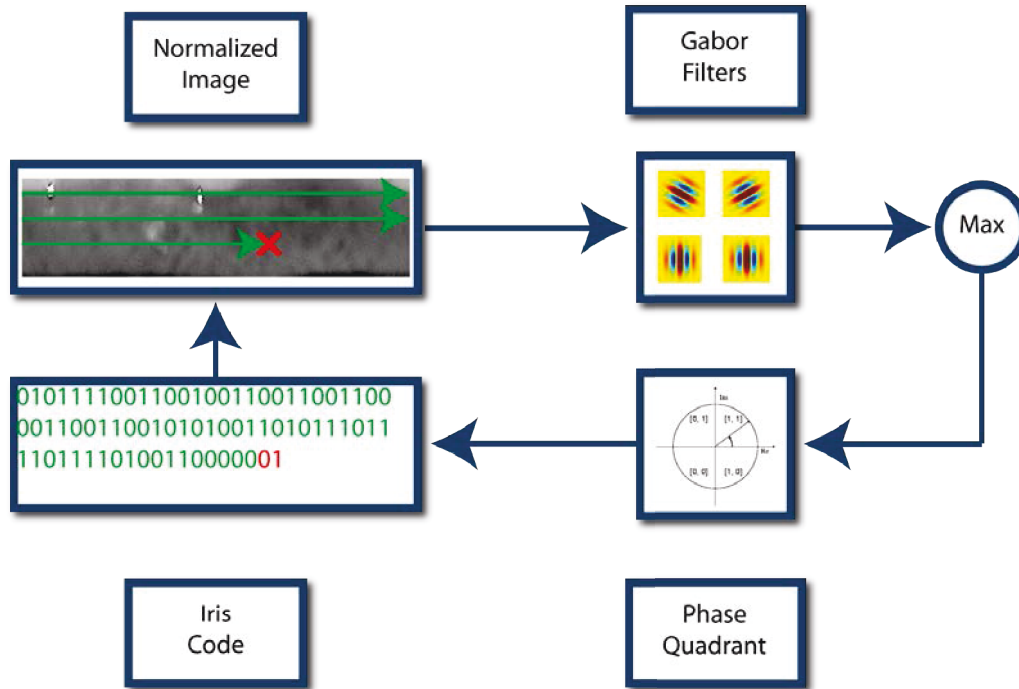


Figure 5: Feature Extraction and Iris Code Generation. The process is divided in four parts. First, an application point is selected in the normalized image. Then, a set of Gabor filters is applied at this application point. The filter giving the larger answer characterizes the texture of the iris at this point. The phase of the Gabor Filter's answer is quantized using two bits. These two bits are stored in the iris code and the process starts again at the next application point.

Formally, the hamming distance between two iris codes $\{codeA, codeB\}$ with associated segmentation masks $\{maskA, maskB\}$ is :

$$HD = \frac{\|(codeA \otimes codeB) \cap maskA \cap maskB\|}{\|maskA \cap maskB\|}, \quad (5)$$

where \otimes is the XOR operator counting the bits that disagree in $codeA$ and $codeB$, \cap is the intersection operator insuring that the information is computed only in valid region of $codeA$ and $codeB$ and $\| \|$ counts the non-zero bits in the iris code.

For impostor comparisons, the two binary codes are uncorrelated and the resulting hamming distance is equivalent to the mean of a random binary drawing of 1 and 0, i.e. 0.5. For genuine comparisons, it should result in a 0 Hamming distance between the binary codes. However, in real scenarios the iris texture is affected by the acquisition conditions, and so is the resulting iris code. It can only be stated that the comparison of two iris code should be below 0.5.

Let us note that Equation (5) assumes that the two binary codes $\{codeA, codeB\}$ are aligned when performing the comparison. This alignment is determined by the arbitrary cut done when unwrapping the texture at the *Texture Normalization* stage. However, tilting of the head produces a rotation of the iris texture in the original image that results in a shift of the texture in the normalized image. Therefore, the corresponding binary codes are not aligned and the comparison using Equation (5) is meaningless. To handle this limitation, Daugman [16] proposes to apply a binary shift to one of the iris code ($codeA$ for instance) and to perform comparisons with the shifted iris code using Equation (5). The final comparison score is the minimum of all the comparisons. The simple statistical drawing of 0 and 1 is then transformed in a "best of n" comparison test of agreement. In [16], Daugman shows that it induces a shift from 0.5 to 0.45 of the theoretical mean for an

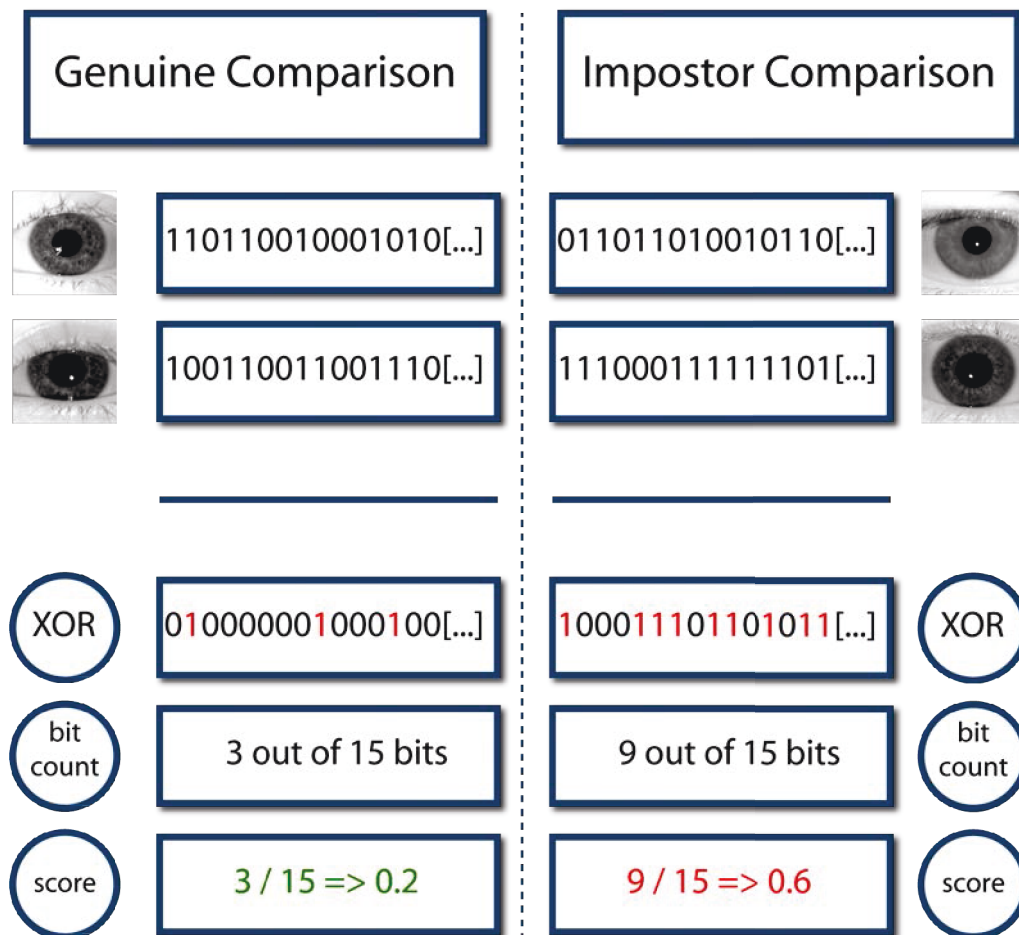


Figure 6: Illustration of the comparisons between two iris codes (not considering segmentation masks and texture rotations). First, the two iris codes are compared using a XOR operation. The sum of the 1 in the resulting binary code represents the distance between the two input iris codes. For iris codes generated from the same iris (genuine comparison) the score is expected to be below 0.5. For iris codes generated from different irises it is expected to have a score around 0.5.

impostor comparison (assuming 7 rotations).

In the end, using a threshold value τ , it is possible to state if the two iris codes $\{codeA, codeB\}$ belong to the same iris or not. If the comparison gives a value below τ the system states that the two codes have been generated from the same iris, and if it gives a value above τ the system states that the two codes have been generated from the different irises. Determination of the threshold value τ is a critical aspect of biometric systems. We discuss this issue in Chapter 2.

1.6 Limitations of Classical Iris Recognition Systems

The early articles of Daugman [16] and Wildes [77] assume that iris has been acquired under constraint acquisition scenario. This means that the subject is active and fully cooperative during the acquisition. However, a recent trend in iris recognition systems is to reduce the acquisition constraints in order to make the systems more user friendly. Relaxation of these acquisition conditions leads to a broad type of degradations in the image such as the ones displayed in Figure 7 :

- Strong occlusions by eyelid, eyelashes, shadows or glasses (Figures 7a, 7b, 7d, 7g, 7i).
- Motion or focus blur (Figure 7f).
- Illumination inhomogeneity or low contrast along the iris borders (Figures 7e, Figures 7g, Figures 7h).
- Gaze deviation (Figures 7c, Figures 7i).

The *Feature Extraction* and the *Pattern Matching* stages are highly affected by the degradation of the biometric information. Global degradations such as blur or illumination inhomogeneity tend to corrupt the available biometric information. It leads to errors in the templates generation, and therefore, to errors in the comparisons.

The *Image Segmentation* and the *Texture Normalization* stages are even more affected by degradations of the iris region in the acquired image. Global degradations such as blur, illumination inhomogeneity or low contrast tend to make the identification of the iris region harder. Therefore, simple segmentation schemes such as the ones proposed by Daugman [16] and Wildes [77] fail when such degradations occur in the image. Moreover, in degraded images, the iris may not appear circular because of occlusions or gaze deviation leading to failures of systems which assume a circular shape for the iris borders.

Accordingly, two main approaches have been considered in the literature to allow iris recognition systems working with degraded images. First approach is to extract more information from the image in order to remain discriminant when the amount quality of the biometric information is reduced. The second approach is to consider new models for segmentation that can robustly identify the iris region in the image and accurately fit a contour model on the iris borders. In this thesis, we have chosen to study improvements for the *Image Segmentation* and the *Texture Normalization* stages. Therefore, this thesis we will propose robust and accurate methods to segment the iris, assuming that the *Feature Extraction* and the *Pattern Matching* are fixed.

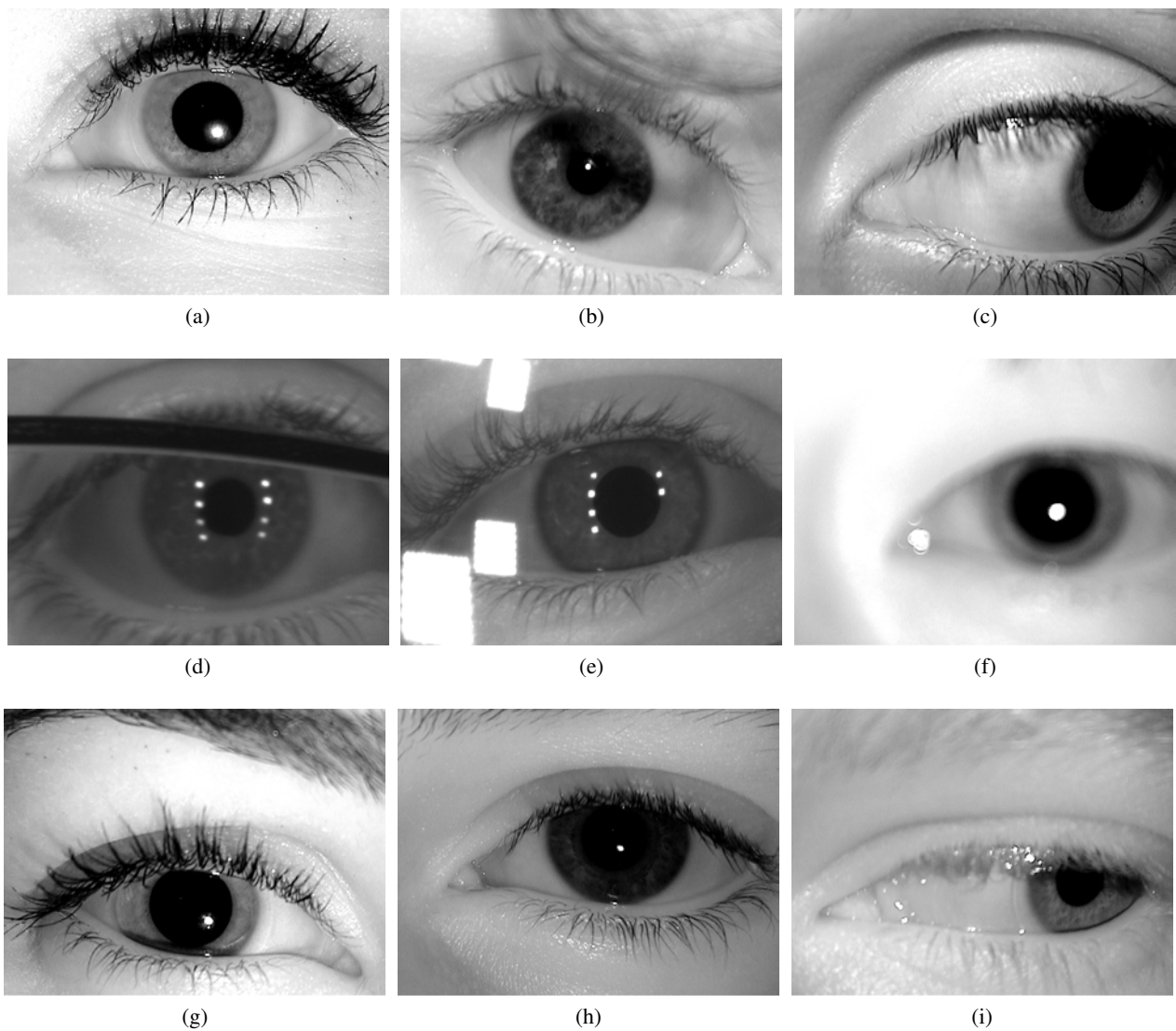


Figure 7: Examples of difficult images for segmentation. (a) : Very packed and dark eyelashes. (b) : Shadows on the pupil and the iris. (c) : Off angle image. (d) : branch of glasses. (e) : Strong reflexions due to glasses. (f) : Out of focus image. (g) : Low contrast between the iris and the sclera. (h) : Low contrast between the pupil and the iris. (i) : Off angle image with heavy occlusion.

1.7 Conclusion

In this chapter, we have presented a classical iris recognition system based on the works of Daugman [16]. Such system has 5 stages : *Image Acquisition*, *Image Segmentation*, *Texture Normalization*, *Feature Extraction*, *Pattern Matching*. They have shown very good recognition performance in constrained acquisition conditions, namely when the subject is active and fully cooperative during image acquisition. However, early iris recognition systems are sensitive to the degradation of the input data quality.

The thesis is focused on the segmentation and the normalization steps of a complete iris recognition system. This particular context for tackling the iris segmentation problem is of a main importance in this thesis, since it leads to a specific methodology for evaluating the segmentation algorithms here proposed. For this reason, in next chapter, we describe the evaluation framework proposed in this thesis for assessing iris segmentation algorithms.

Chapter 2

Evaluation of Iris Recognition Systems

Biometric systems can be used on huge databases. National databases can store the information of millions of individuals (or even a billion for the UIDAI program in India [74]). Such biometric systems are expected to achieve a very high recognition performance. Therefore, it is important to have an estimation of the expected accuracy of the system on very large databases.

In Section 2.1, we explain how we evaluate the accuracy of a general biometric system, with a particular focus on the case of iris. Then, in Section 2.2, we explain how to evaluate the quality of an iris segmentation algorithm using the whole recognition system. Finally, in Section 2.3, we describe the databases used in this thesis to validate our results.

2.1 General Evaluation of Biometric Systems

Large scale evaluation of biometric systems is done offline on reference databases : each biometric sample of each individual is compared to every other element in the database. When the samples belong to the same person, the comparison is labeled as "genuine comparison". When the samples belonged to different persons, the comparison is labeled as "impostor comparison".

To evaluate the accuracy of a biometric system, score distributions for genuine and impostor comparisons are generated for the whole database. The precision of a biometric system is characterized by its ability to separate between the two distributions. It defines a threshold setting the boundary between the two distributions (if the score of a comparison is below the threshold, it is considered as genuine, otherwise it is considered as impostor).

If the two distributions do not overlap, the system can perfectly distinguish between the genuine class and the impostor class (See Figure 8a). However, in operational systems the two distributions do overlap, so no threshold can perfectly separate the two classes (See Figure 8b). For any given threshold, some of the genuine comparisons will be taken as being impostors (producing a False Rejection; FR when this happens) and some impostor comparisons will be taken as being genuine (generating a False Acceptance ; FA).

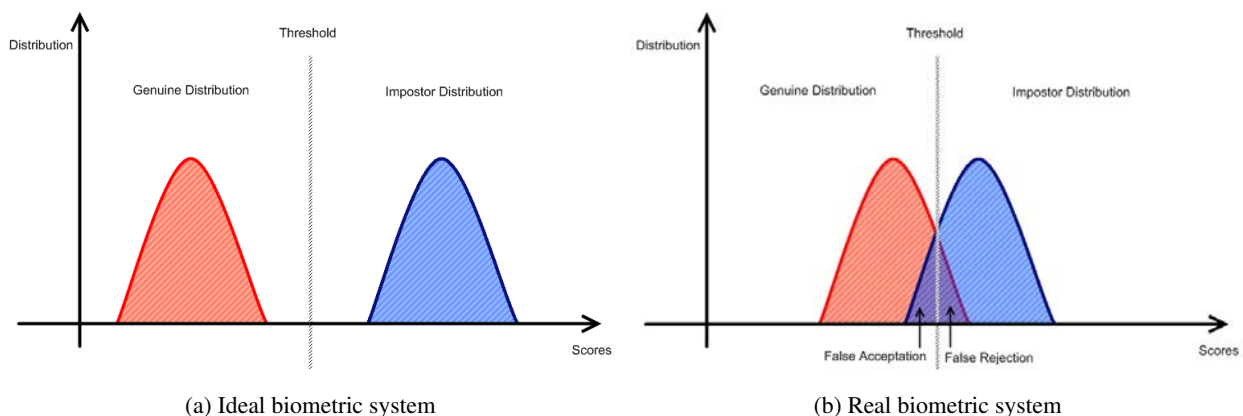


Figure 8: Genuine and impostor scores in biometric systems. Figure (a) : Ideal Biometric system where the genuine and impostor scores do not overlap. Figure (a) : Operational biometric system with an overlapping between genuine and impostor scores generating errors.

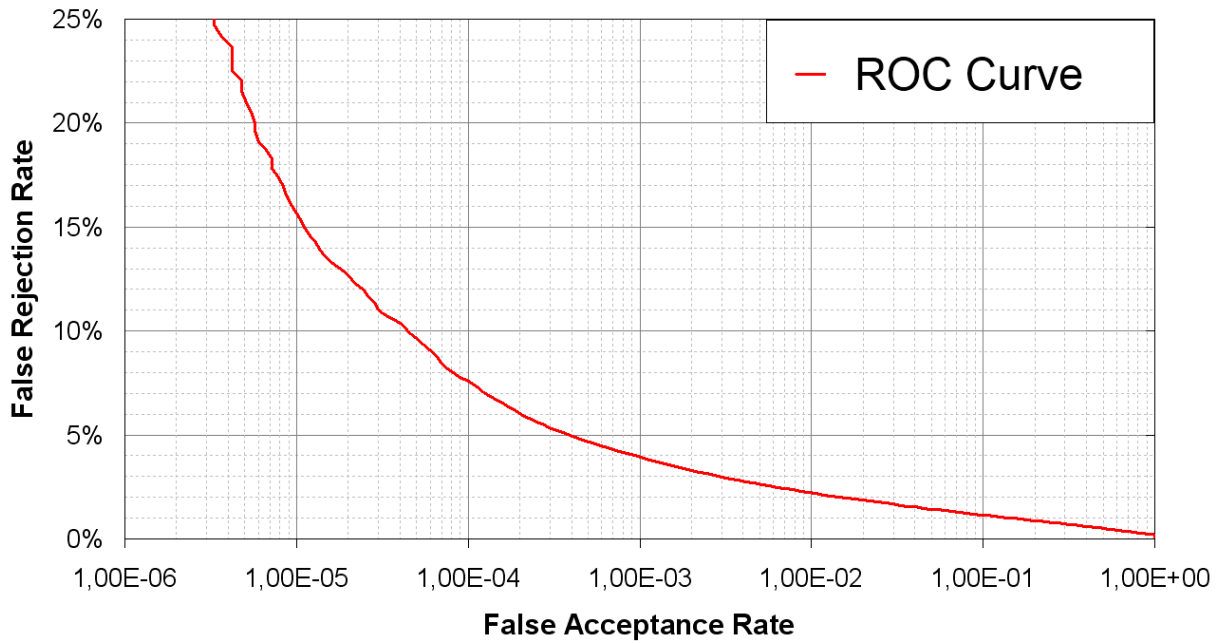


Figure 9: Example of ROC curve, setting the balance achievable between FAR and FRR for a given system.

A biometric system is characterized by the average amount of FA and FR for a given threshold namely the False Acceptance Rate (FAR) and False Rejection Rate (FRR). Definition of FAR and FRR are introduced in [28].

In a biometric system, any user has to claim an identity I . He also provides a biometric template X_Q (through iris acquisition for example). X_Q shall be similar to the biometric template stored in database for this identity $I : X_I$. Then the hypotheses are :

H_0 The input X_Q does not belong to the same person as the template X_I .

H_1 The input X_Q belongs to the same person as the template X_I .

The resulting decisions are :

D_0 The person is not who she claims to be (fraud).

D_1 The person is who she claims to be.

The decision rule is then : if the matching score $S(X_Q, X_I)$ is less than a given threshold, decide D_1 otherwise decide D_0 . The resulting errors are :

type I : False Acceptance (D_1 is decided when H_0 is true).

type II : False Rejection (D_0 is decided when H_1 is true).

FAR is the probability of *type-I* error and FRR is the probability of *type-II* error :

$$\begin{aligned} FAR &= P(D_1|H_0), \\ FRR &= P(D_0|H_1), \end{aligned} \quad (6)$$

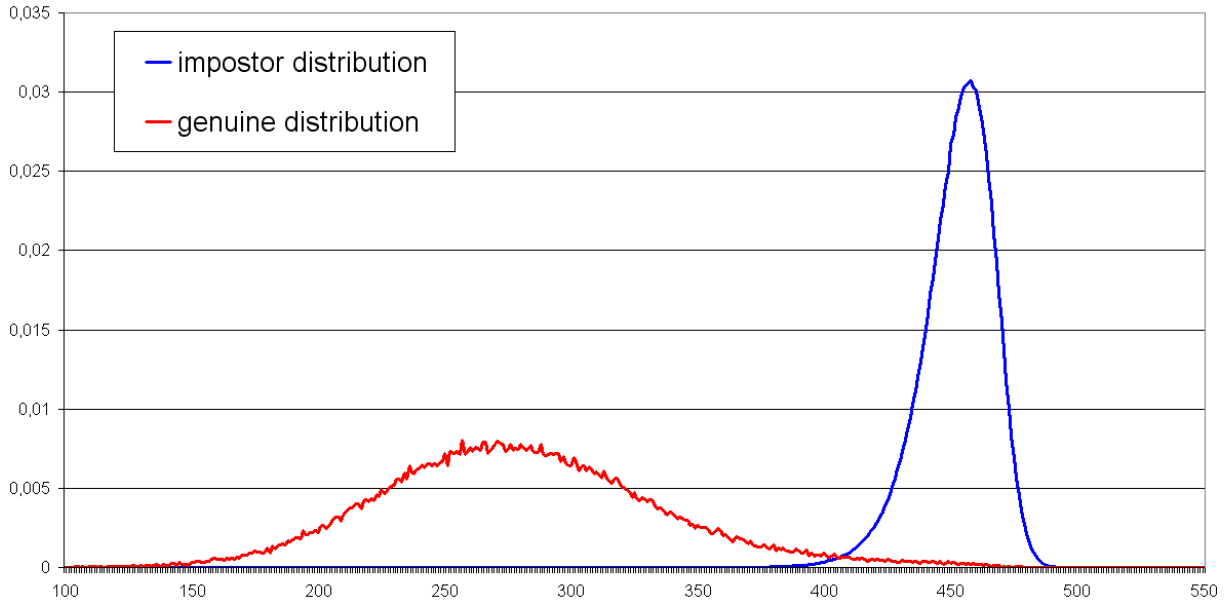


Figure 10: Distribution of the genuine and impostor scores for an iris recognition system

As explained earlier, the evaluation of a biometric system is done using the genuine and impostor distributions for a reference database. Estimation of FAR and FRR for a given threshold t (assuming that genuine comparisons have lower scores than impostor comparisons) is :

$$\begin{aligned} FAR(t) &= \int_{-\infty}^t p(S(X_Q, X_I)|H_0)dS, \\ FRR(t) &= \int_t^{\infty} p(S(X_Q, X_I)|H_1)dS, \end{aligned} \quad (7)$$

The system performance at any operating point (variation of threshold t) is represented by the Receiver Operating Characteristic (ROC) curve such as the one displayed in Figure 9. This representation allows evaluating the expected behavior of the system for a given threshold, and therefore, to determine a threshold according to this behavior. For instance, a system regulating the entrance to a restricted area should prevent impostors from getting in (low FAR), even if the authorized persons need several attempts to get in (high FRR). Given the system's ROC curve displayed in Figure 9, a possible operating point could be : $FAR = 10^{-5}$ and $FRR = 0.15$. For such an operating point, an impostor has one chance out of 100.000 to get in, but regular users are refused 15% of the time.

Figure 10 shows the typical distributions for an iris recognition system. As expected, the impostor distribution has a mean around 0.45 and the genuine distribution has a mean around 0.28. The genuine distribution has a larger variance and the right tail of the distribution tends to expand further on the right. Comparisons scores in this right tail usually involve images of very low quality.

2.2 Evaluation of Segmentation Quality

As this thesis is devoted to iris segmentation, evaluating the quality of our segmentation results is a key issue. The evaluation of segmentation quality is a classical image processing problem. However, despite a global consensus about the importance of this topic, it is still an open problem.

This statement becomes even more true when applied to the iris segmentation problem. This problem refers to two different actions : the generation of the segmentation masks and the estimation of the iris borders for normalization. Our first attempt to evaluate the quality of our segmentation algorithms was to compare the results with a manually generated ground truth. Though this approach seemed pertinent for the segmentation mask, it appeared as irrelevant for the evaluation of the normalization contours. Indeed, we noticed that the recognition performance of systems using simple iris segmentation algorithms were better than the recognition performance of systems using our manual ground truth. We find out that this behavior comes from the small impressions introduced by the operator during the generation of the ground truth. Indeed, results produced by an human operator have a few pixels of uncertainty and automatic algorithms usually provide more reliable results (even simple ones). Therefore, the only effective way of evaluating the accuracy of a segmentation algorithm is by assessing its impact on the recognition performance.

In Biometrics, the quality of every module of a system is evaluated with respect to its impact on the overall recognition performance. It is intuitive that the better a segmentation algorithm performs, the better the recognition performance should be. Accordingly, the first way to evaluate the quality of segmentation is by using the ROC curves presented in the previous section.

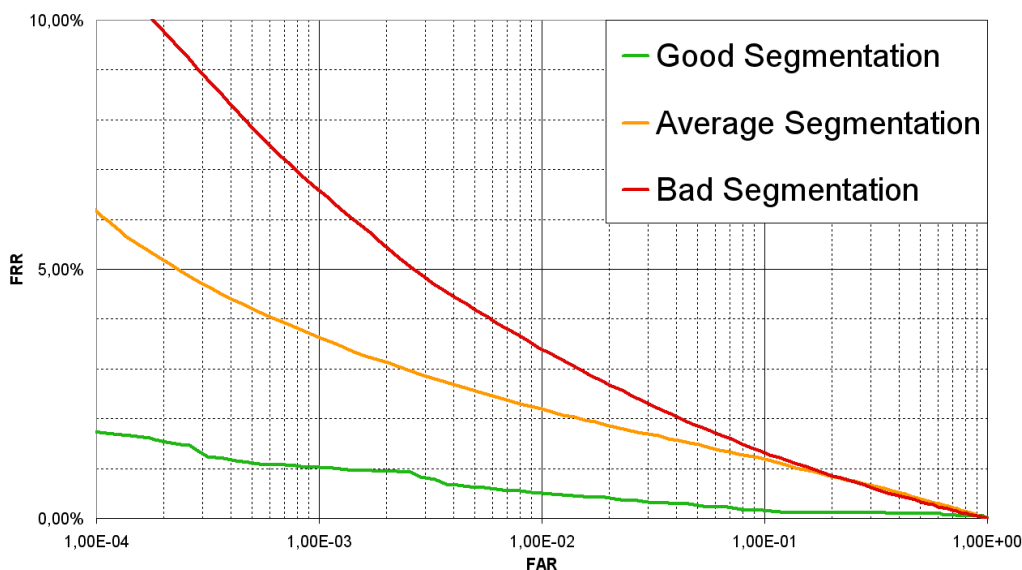


Figure 11: Ranking of different segmentations using ROC curves, using the same features extractor and the same features matching.

ROC curves enable a global evaluation of the system's performance, but they fail to assess the segmentation quality of a single image. To handle this issue, we use a concept introduced in the article of Elham Tabassi [70]. The key idea is to define image specific error rates.

- Image false match rate iFMR - the proportion of comparisons involving this image producing a false match.
- Image false non-match rate iFNMR - the proportion of comparisons involving this image producing a false non match.

Let us note the FAR and the FRR we introduced in the previous chapter are related to a system, whereas the FNM and the FNMR are related to a recognition algorithm. The FAR and the FRR consider images that have been rejected by the system as being errors (because their quality is too low for instance). On the opposite, the FMR and the FNMR only consider mistakes of the comparison algorithm.

Taking the formulation of the original article, we define s_{kl}^{ij} the comparison score of the k -th image of subject i with the l -th image of the subject j . Then, the set of impostors for the k -th image of subject i is :

$$I(i, k) = \{s_{kl}^{ij}, i \neq j, j = 1 \dots J, l = 1 \dots N_j\}, \quad (8)$$

for comparison against all N_j images of all J persons in the test set. The image false match rate is then defined for a given threshold τ as :

$$iFMR(\tau, i, k) = \frac{\sum_{s \in I(i, k)} 1 - H(s - \tau)}{\sum_{s \in I(i, k)} 1}, \quad (9)$$

with H the Heaviside function defined as :

$$H(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \quad (10)$$

The non false match rate is defined in a similar way. For a given threshold τ :

$$G(i, k) = \{s_{kl}^{ii}, k \neq l, l = 1 \dots N_i\}, \quad (11)$$

$$iFNMR(\tau, i, k) = \frac{\sum_{s \in G(i, k)} H(s - \tau)}{\sum_{s \in G(i, k)} 1}. \quad (12)$$

For a given threshold τ , the $iFMR$ and $iFNMR$ value of an image k defines its tendency to produce false match or false non match comparisons. An interesting aspect of this formulation is that images on which segmentation has failed will have a tendency to produce more False Non-Matches than correctly segmented ones. Figure 12 shows the $iFNR$ and $iFNMR$ for some images of very bad quality or poorly segmented.

Let us note that the evaluation tools we described are specific to a given recognition system. We will discuss some more image related quality metrics in Chapter 7 where we try to predict failures of the segmentation algorithms.

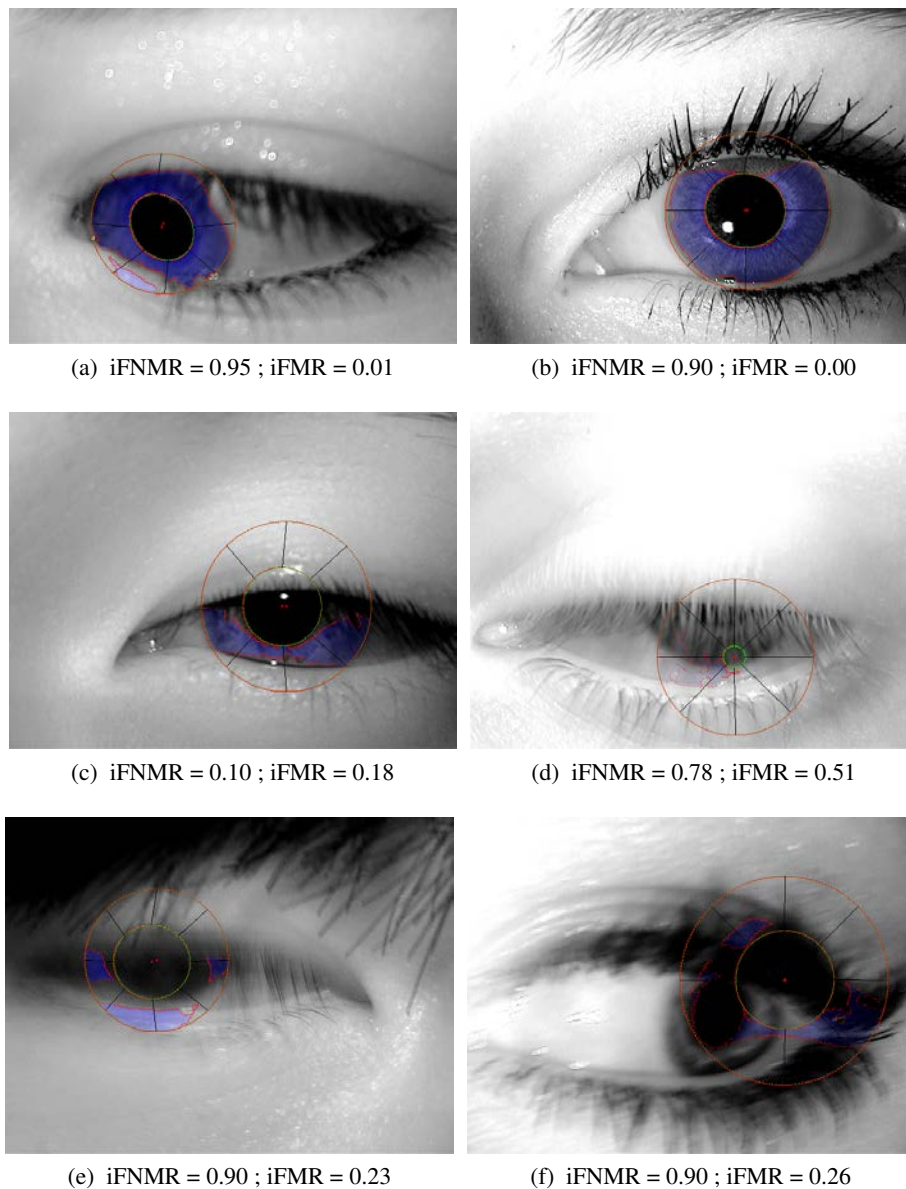


Figure 12: Examples of images identified as not suitable for recognition based on their image False Match Rate and image False Non Match Rate (threshold set at 0.36). The results of the segmentation process are displayed on the images. The two ellipses correspond to the normalization contours and the blue region to the segmentation mask. Figure (a) : Off-angle image where contours fitting has failed. Figure (b) : Image correctly segmented but with a cosmetic contact lens, making it unsuitable for recognition. Figure (c) : Image correctly segmented, but with very few information available due to occlusions. Figures (d) (e) (f) : Complete failures of the segmentation.

2.3 Databases

Databases are a key aspect of biometric systems evaluation : to assess the recognition performance of a system, a large number of comparisons is required. In this thesis, we focus on the databases ICE2005, ND-Iris and CASIA V3/V4.

We choose to study these databases because they are the largest databases acquired with low acquisition constraints available at the beginning of this thesis. Therefore, the databases contain several kind of image degradations such as illumination inhomogeneities, hard or synthetic lenses, occlusions, poor focus or off-angle images. It is also interesting to note that the CASIA databases have been acquired with a different sensor than the one to acquire the ICE2005 and ND-Iris databases that allows validating our results for different sensors.

2.3.1 ICE 2005



Figure 13: Examples of ICE2005 images

The Iris Challenge Evaluation [55] (ICE) was constructed by the NIST in 2005 to proceed the first large scale iris recognition challenge, namely evaluating iris recognition systems on degraded images. The ICE database consists of 2,953 images from 132 subjects including degradations such as illumination inhomogeneities, occlusions, poor focus or off-angle images.

2.3.2 Nd Iris

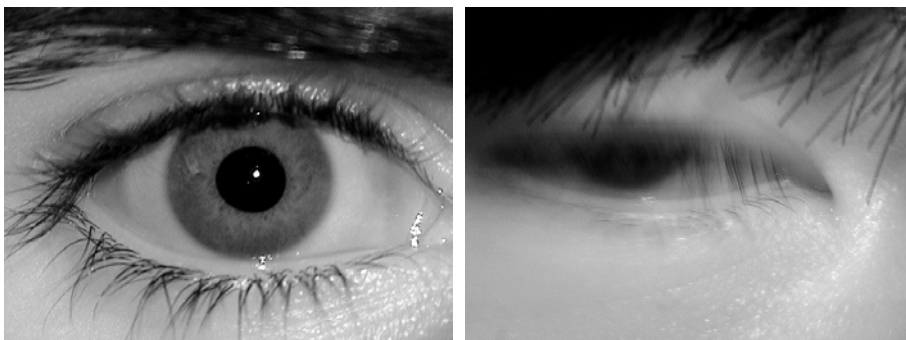


Figure 14: Examples of ND-Iris images

The Nd-Iris-0405 Database [56] is an extension of the ICE2005 database. This Database is one of the largest public databases available for iris recognition with 64,980 iris images obtained from 356 subjects (712 unique irises). This database is acquired with low constraints. It contains off-angle images, hard or synthetic lenses, large occlusions or images with illumination variations.

2.3.3 CASIA

The CASIA databases have been constructed by the Chinese Academy of Sciences' Institute of Automation [9]. These databases are mainly composed of Asian people. This population tends to have thin eyelashes oriented downwards for the upper eyelid, thus covering a significant part of the iris and therefore inducing segmentation errors.

CASIA V1 was the first public irises database. Therefore, this database has been widely studied in the early ages of iris literature. The database is composed of 756 irises of 108 different eyes. The acquisition was highly constrained and the pupil was manually edited to be a uniform region of constant intensity. Therefore, this database should not be used for Iris recognition anymore as it does not match real acquisition conditions [54].

CASIA V3 was released later and is composed of three subsets :

Interval This subset contains 2,655 images from 396 eyes of 249 persons. Images are acquired using the same sensor as CASIA V1, but without manual alteration of the pupil. The in-house sensor used to create this database ensure a strong illumination showing very rich texture in the iris.

Lamp This subset contains 16,213 images captured from 819 eyes of 411 subjects. This database contains strong illumination inhomogeneities and several other degradations such as hair occlusions. Since the eyes are taken under different illumination conditions, the pupil dilation may vary largely from one eye image to another. The authors expect this database to be suitable for studying non-linear distortions of the iris texture due to pupil dilation.

Twins This subset contains 3,183 images captured from 400 eyes of 200 subjects (100 pairs of Twins). The authors expect this database to demonstrate that iris is strictly different even between Twins.

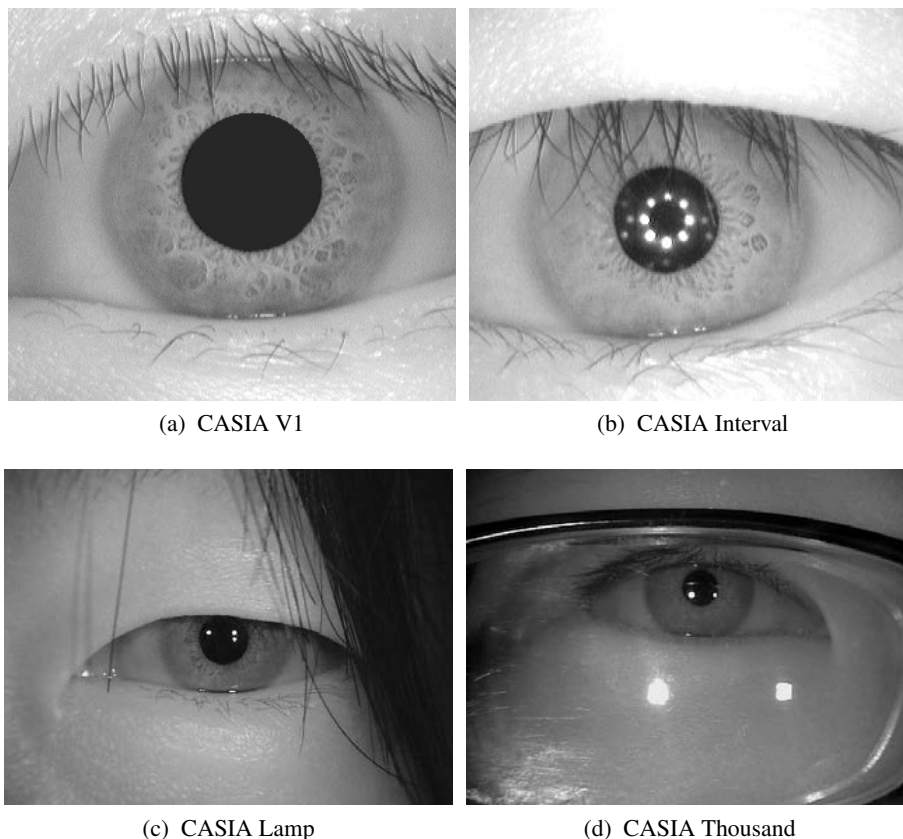


Figure 15: Examples from the different CASIA databases.

CASIA V4 extends CASIA V3 adding three new subsets :

Distance This subset contains images captured at a distance using an in-house imaging system. The acquisition system also acquired other biometric samples such as high quality face images in order to evaluate multi-modal fusion.

Thousand This database contains 20,000 iris images from 1,000 subjects. This is the first publicly available iris database containing one thousand different subjects.

Syn CASIA-Iris-Syn contains 10,000 synthetic iris images representing 1,000 chimeric subjects. The irises are synthesized using normalized images from CASIA V1.

2.4 Conclusion

In this chapter, we have first recalled how to evaluate the recognition performance of a given biometric system. This evaluation is done offline by comparing a large number of biometric samples.

This chapter is centered on how to use the recognition performance of a complete iris recognition system to characterize the accuracy of an iris segmentation algorithm. Indeed, the better the segmentation algorithm is, the better the system's recognition performance should be. We propose to carry out the evaluation at two levels :

- At a global level (the whole recognition system) using ROC curves.
- At an image level using the image False Match Rate (iFMR) and image False Non-Match Rate (iFNMR).

Finally, we have described the different databases we used during this thesis for validating our results. We have selected these databases because they have been acquired with low acquisition constraints and therefore, are representative of the different problems addressed during this thesis.

In next chapter, we will study the specificity of iris segmentation. We will make a survey of some classical iris segmentation methods. We will use this survey to explain technical choices made in this thesis and the consequences for our segmentation system.

CONFIDENTIAL

Chapter 3

Segmentation Techniques for Iris Recognition

Iris segmentation refers to two distinct actions :

1. The identification of the pixels belonging to the iris in the eye image (generating a so called segmentation mask).
2. The estimation of a parametric description for the iris inner and outer boundaries (in the image, but also under possible occlusions)

Such information is critical to perform recognition. The parametric contours are used to unwrap the iris texture in order to produce the normalized image (Equation (3) of Chapter 1) that is used at matching step. The segmentation mask is used to remove non iris area from the matching process (Equation (5) of Chapter 1).

The first segmentation algorithms of Daugman [16] and Wildes [77] assume that the iris can be modeled by two non concentric circles possibly occluded by the eyelids. Therefore, they reduce the segmentation problem to the search of circles with simple eyelids detection schemes. These approaches showed good results when subjects are highly cooperative. However, when acquisition conditions are relaxed, the iris borders are more likely to be occluded and simple segmentation models tend to fail. For this reason, new algorithms have been proposed in order to handle degradations appearing in this context.

The iris segmentation task thus produces a segmentation mask and a parametric description of the iris borders. Accordingly, we identify two main strategies that differ in the processing order. The first strategy, which we call *Contours First*, first finds parametric descriptions of the iris borders and then removes occlusions in the

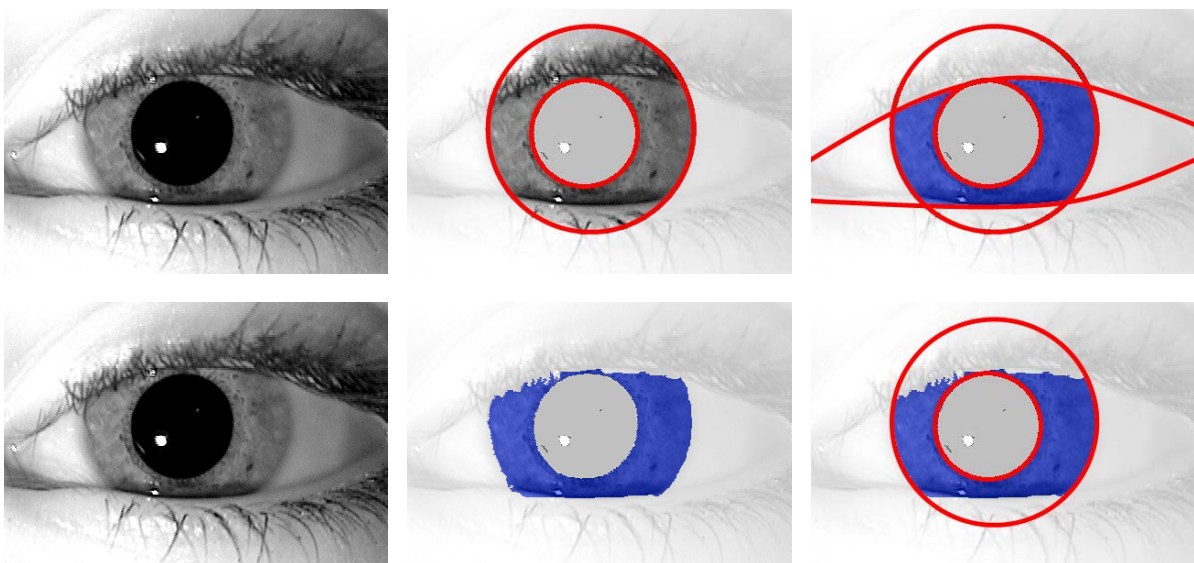


Figure 16: Two main segmentation strategies. On the upper row, the *Contours First* strategy : the borders of the iris are found first and then the segmentation mask is obtained by removing occlusions. On the lower row, the *Texture First* strategy : the texture of the iris is found first and then a model is fitted on the iris borders.

segmentation mask. The second strategy, which we call *Texture First*, first determines a precise segmentation mask of the iris texture and then, finds a parametric description of the iris borders. The difference between such two strategies is illustrated in Figure 16. At the end, both strategies will produce close results, but this choice has some consequences on the design of the segmentation algorithms :

- *Contours First* strategies (first row of Figure 16) first find parametric descriptions of the iris borders and then remove the occlusions. This first stage must be both robust and accurate in finding the visible iris borders, and in predicting the shape of the occluded ones. Indeed, mistakes made during this first stage will directly affect the image normalization. Once the shape of the iris is known, occlusions are removed from the segmentation mask using simple methods like eyelids detection.
- *Texture First* strategies (second row of Figure 16) first precisely segment the iris texture in the image and then estimate the iris borders. This segmentation can be performed directly on the image using classical region segmentation algorithms. However, the region segmentation does not provide parametric contours for normalization. Accordingly, the search of the parametric contours is done using the information from the region segmentation.

In this chapter, we review some methods proposed in the literature to segment the iris. First, in Section 3.1, we recall briefly the early segmentation approaches of Daugman and Wildes. Then, we present more recent works on iris segmentation. We describe *Contours First* strategies in Section 3.2 and then present *Texture First* strategies in Section 3.3. Finally, in Section 3.4, we present the key problems to solve in the framework of the chosen strategy for iris segmentation and the proposed solutions.

3.1 Early Segmentation Approaches

The first article proposing an automated system able to identify people based on the texture of their irises was proposed by Daugman in [16] in 1993. This article was followed in 1997 by an article of Wildes [77] based on a different comparison algorithm. These two articles address the problem of identifying people whose irises have been acquired in a controlled acquisition scenario. In these studies, the eye images are clear and the irises always visible in the image. They both model the iris borders using circles but propose different algorithms to find these circles in the image. Finally, they remove the eyelids from the image using simple detection methods.

Using the terminology we proposed in the introduction of this chapter, these two articles apply a *Contours First* strategy because they first search the parametric contours (the circles) and then remove the occlusions (the eyelids).

3.1.1 Daugman's approach

In [16], Daugman proposes to find circles in the image using the following integrodifferential operator (IDO) :

$$\max_{(r,x_0,y_0)} \left\| G_\sigma(r) * \frac{\partial}{\partial r} \oint_\theta \frac{I(x_0 + r \cos \theta, y_0 + r \sin \theta)}{2\pi r} d\theta \right\|, \quad (13)$$

where (r, x_0, y_0) are the parameters of a circle with center (x_0, y_0) and radius r , I is the image, G_σ is a smoothing Gaussian of scale σ , and $*$ denotes convolution. This operator finds the circle where the change of intensity is maximal. For practical implementations, the author proposes using a first order approximation of the operator and evaluating the operator on a fixed grid. These approximations allow to speed up the process compared to a naive implementation of equation (13).

In the original article [16], Daugman makes an empirical assumption on the expected location of the eyelids and always remove there areas from the segmentation mask. In [15], Daugman proposes to locate the eyelids

based on an integrodifferential operator designed to locate parabola similarly to what he proposes for circles in Equation (13).

Although the operator (13) has shown good result in controlled acquisition scenarios, it lacks robustness in less constraint acquisition conditions. Indeed, as the circle detection is based on an energy formulation, it can be affected by irrelevant optima created by other elements of the image such as the eyelids, eyelashes or shadows.

3.1.2 Wildes' approach

In [77], Wildes proposes to find circles in the image using classical image processing methods : he first looks for edges in the image using a simple edge detector (Canny in [77]) followed by a CHT (Circular Hough Transform [53]). In addition to this circle finding process, the author proposes to segment the eyelids to remove this area from the comparison process. This detection is also very classical and is based on a linear Hough Transform. The critical issue of this method is to select an accurate edge detector. Indeed, if the iris's edges are not present in the edge image, the Hough transform can not find the iris contours. On the opposite, if the edge detector is not selective enough, the time computation of the CHT may dramatically increase. For these reasons, most of the articles based on Wildes' idea modify the edge detector to improve performance [36] [41] [39]. This method is very robust because it uses very stable image processing tools. However, the precision is directly affected by the choice of edge detector and its settings.

3.2 Contour First Strategy

In this section we describe methods we identified as *Contours First* strategies. These methods first accurately locate the borders of the iris visible in the image. Then, they use this information to predict the shape of the iris in occluded regions. Finally, they remove the occlusions (like eyelids or eyelashes) from the segmentation mask. We give a brief summary of the key elements of each article in Table 1.

In [17], Daugman proposes a set of methods to solve some of the main issues of early iris segmentation systems, like handling persons which have an anatomically non-circular iris and gaze deviations. To that aim, he models the iris borders with parametric contours described by Fourier coefficients instead of circles. He first roughly locates the pupil and the iris in the image and then, uses this initialization to segment the iris with snakes described by Fourier series expression. The interest of this formulation is twofold : on one hand it gives a very effective scheme to identify the iris borders, and on the other hand, the contours are parametrically described and can be used to normalize the image instead of circles. To address the off-angle problem, the author uses the first two coefficients of the contours' Fourier expansion to rectify the image deviation (i.e. he assumes an elliptic shape). Finally, the eyelashes overlapping the iris are removed based on a simple statistical test on the intensity distribution.

This method is a good illustration of modern *Contours First* strategies because the author focuses on the research of a parametric description of the iris anatomic borders and then uses a simple scheme to remove occlusions. Therefore, the main difficulty of the proposed model is to use only the "true" iris borders in the image to fit the snake. Indeed, if the snake fitting is done using some edges generated by occlusions (for instance an edge separating the iris texture and the eyelid), the overall shape of the contour will be corrupted.

In [46], Miyazawa et al. propose a new approach to compare irises using global phase correlation. This comparison method provides an interesting alternative to the Daugman's reference method described in Section ?? and 1.5 of Chapter 1. However, the method proposed in [46] also requires a normalized image generated using a parametric description of the iris contours. To get these parametric contours the authors propose a quite unique *Contours First* segmentation strategy. They first locate the pupil by looking for dark areas close to specular reflections. Then, they apply a Parametric Deformable Model based on 10 parameters to segment the iris. This model is composed of one ellipse for the pupil border and one for the iris border. The idea of this method is to minimize a certain energy of the contour by iteratively evolving the parameters of the model.

This method is interesting because the Parametric Deformable Models are known to be very robust and are likely to work even with degraded images. However, they assume that the iris can only be occluded by eyelids and other types of occlusions are not handled by this model.

In [63], Ryan et al. adapt an eye tracking method called starburst to iris segmentation. They first locate the pupil with a simple thresholding method assuming that the pupil should be far from white areas (the skin). Then, they propose to fit an ellipse on the pupil and the iris contours using starburst. The idea of starburst is to send rays from the initial location of the pupil and assume that the occurrences of larger gradient found along the ray correspond to the pupil or the iris borders. Several directions are considered to get a set of points that are likely to belong to the borders. Those points are used to generate an ellipse for the pupil and the iris. The eyelids are segmented using a 1-D snake. Unlike most biometric-related articles the authors do not evaluate the segmentation accuracy using the performance of their system. Instead, they rely on manually segmented images from ICE2005. The authors define a distance metric for ellipses and compare their algorithm to some reference algorithms of the literature on images which have been manually annotated.

In [26], He et al. propose a complete *Contours First* segmentation strategy. They first propose an accurate and effective way to get a parametric description of the iris borders, and then, they propose an effective scheme to remove occlusions generated by eyelashes and eyelids. The pupil is first roughly localized using an adaboost classifier. This rough location is used to initialize a "pulling and pushing" algorithm to generate the parametric description of the iris borders. This algorithm starts by a rough detection of the edges points that are likely to belong to the pupil (resp. the iris) border, and then the "pulling and pushing" algorithm is used to keep only edge points belonging to the pupil (resp. the iris) border. This set of edge points is used to fit a spline which will give the parametric description of the pupil (resp. the iris) border. Finally, the eyelids and eyelashes are segmented using statistical inferences of the intensities in the regions.

Like [17], the method proposed in [26] is a good illustration of *Contours First* strategies because they apply a very effective scheme to first find a parametric description of the iris borders and then remove the occlusions.

In [10], Chou et al. focus on the correction of the gaze deviation for off-angle images. The authors first propose a new acquisition device taking simultaneously the three visible channels and a NIR image of the eye. The pupil is first roughly segmented using thresholding in the NIR image. Eyelash area are removed using hypotheses on the pupil shape. Then, an elliptic Direct Least Square fitting is applied in order to get an elliptic contour for the pupil area. This elliptic contour is used to perform a circular rectification, i.e. applying a global transform on the image morphing the pupil ellipse into a circle. Once the four channel image has been rectified, the authors look for a circle in the corrected image using the RANSAC [21] algorithm on the four channels of the image. The performance evaluation is done both in terms of recognition performance and using manual annotation. The authors show that a significant improvement of performance is achieved compared to a circle based method and that the improvement is proportional to the strength of the gaze deviation.

The approach proposed in [10] is interesting because it explicitly studies the influence of an elliptic contour model on off-angle images. The results of this study show that elliptic methods are pertinent to solve the off-angle problem. However, this approach can not be used directly without the sensor proposed by the authors to simultaneously acquire the NIR and the visible image of the eye.

In [84], Zuo et al. propose a complete system for iris segmentation designed to handle most of the degradations commonly found in iris images. The authors start by applying a preprocessing to the image in order to remove specular reflections and noise. The pupil is segmented based on a rough thresholding and morphological operations. Finally, an elliptic contour is fitted to the pupil border. The iris contour is found using an adaption of Daugman's integrodifferential operator for ellipses. Finally, illumination inhomogeneities are corrected, and the eyelash occlusions are removed using morphological operations. To estimate the segmentation quality the authors propose to use a subjective evaluation based on the distance between the resulting ellipses and the actual borders of the iris.

The proposed approach differs from the other *Contours First* strategies we presented. Indeed, the authors propose to correct the degradations in the image at every step of the algorithm whereas other methods usually

| Article | Initialization | Segmentation | Refinements | Unwrapping contours |
|---------------------------|--|---|---|---|
| Daugman 1993 [16] | - | Pupil and iris found using IDO | Empiric suppression of the upper and lower part of the iris | Circles from the IDO |
| Wildes 1997 [77] | - | Pupil and iris found using CHT | Eyelids detection based on a Linear Hough Transform | Circles from the CHT |
| Daugman 2007 [17] | Rough localization, eyelids segmentation | Snakes using Fourier series | Off-axis correction and eyelash suppression | Parametric contours given by Fourier series |
| Miyazawa et al. 2008 [46] | Rough pupil segmentation | Deformable iris model with 10 parameters | Eyelash suppression | Ellipses from the deformable model |
| Ryan et al. 2008 [63] | Rough pupil localization | Starburst segmentation for the pupil and the iris contours | Eyelid segmentation using snakes | Ellipses from the starburst segmentation |
| He et al. 2009 [26] | Rough location based on Adaboost classifier | Pupil and iris segmentation using pulling and pushing method | Border fitting using cubic splines. Eyelid segmentation and eyelash suppression | Refined polynomial curves |
| Chou et al. 2010 [10] | Rough pupil segmentation | Elliptic DLS for the pupil and RANSAC algorithm for the iris | Global transform to map the pupil ellipse onto a circle | Circles in the rectified image |
| Zuo et al. 2010 [84] | Inpainting and contrast enhancement | Thresholding for the pupil and the iris. Fitting using elliptic Daugman's IDO | Illumination correction and eyelid removal | Ellipses from the elliptic IDO |
| Uhl et al. [3] | Rough initialization using weighted Adaptive Hough transform | Line search in polar coordinates and elliptic DLS fitting. Second contour is found using ellipsopolar transform w.r.t. to the first contour | Eyelid and eyelash removal | Segmented ellipses |

Table 1: Relevant iris segmentation/normalization algorithms from the literature which use a *Contours First* strategy. (IDO stands for Integrodifferential Operator and DLS for Direct Least Square)

remove the occlusions at the end. Moreover, the pupil region is first accurately segmented before the elliptic fitting. Therefore, the pupil segmentation stage can be seen as a *Texture First* approach. We decided to classify this algorithm as *Contours First* strategy because the iris segmentation stage is done using a *Contours First* strategy.

Most iris segmentation methods we described so far are designed to segment images acquired under Near Infrared illumination. For this type of illumination, the pupil is the most stable element in the image and is segmented first. For image acquired in visible wavelength, the most stable element in the image is the limbus. Accordingly, the outer boundary of the iris is segmented first. In [3], Uhl et al. propose a method to segment the inner and outer border of the iris without assumptions on the processing order. Therefore, this method can be used either for NIR and visible images. The process is first initialized by finding a rough localization of the iris center using a weighted adaptive Hough transform. Then, the image is transformed into a polar image and the first contour is found by searching for lines in the polar image. The second contour is found using an ellipsopolar transform. Finally, the contours are identified as either pupil or iris contours based on the strength of the image's gradient at the location of the contour.

We can note that all the articles we presented as *Contours First* strategy focus on finding the visible borders of the iris in the image to fit a parametric model under possible occlusions. These models can be either circles [16, 77], ellipses [46, 63, 10, 84, 3] or general parametric contours [17, 26] and these parametric descriptions are used to generate the normalized image of the iris texture. However, this process must be very robust and accurate because any mistake made during the localization of the contours will directly impact the normalized image.

3.3 Texture First Strategy

In this section we describe *Texture First* methods. These methods segment the iris texture first and then apply a fitting method based on the region segmentation results to get the parametric description of the iris borders. Such methods can handle arbitrary types of occlusion and achieve a very precise segmentation of the texture. We give a synthesis of the key points of the presented articles in Table 2.

In [76], Vatsa et al. propose a complete iris recognition system from the *Image Segmentation* to the *Pattern Matching*. They propose to segment the iris using the Chan and Vese model for Active Contours [8]. Then they propose several quality enhancements applied on the normalized image to improve the recognition performance of the system. Finally, they propose two comparison algorithms they fuse using a Support Vector Algorithm (SVM).

The segmentation process starts by a rough estimation of the pupil and iris borders based on an iterative algorithm designed to find ellipses in the image. They initialize a circle at the center of the eye and randomly select some points on the circumference to decide if the contour should shrink or expand in order to maximize the variation of intensities. Then, based on this rough initialization, a Chan and Vese active contour is used to get the accurate segmentation mask of the iris and the iris is normalized using the initial ellipse parameters. This approach was one of the first *Texture First* methods of the literature. One of the main limitation of this approach is that the contours used to generate the normalized image are the same as the one used to initialize the Active Contours. It is a problem because this process is done quite empirically and have a low accuracy compared to the region segmentation proposed in the article.

In [66], Shah et al. want to segment the iris region without making assumption on the shape of the iris. Therefore, they propose to use Geodesic Active Contours (GAC [7]) to segment the iris texture. The process is initialized using a rough segmentation of the pupil based on thresholding. In order to distinguish the pupil area from other dark elements in the eye the authors propose to apply a circular least square fitting on the regions borders and select the region closest to a circle. Then, they use this rough initialization of the pupil to segment the iris using a Geodesic Active Contours. Finally, in order to get the parametric description of the contours for normalization, they propose to apply a circular fitting on some relevant points of the iris mask.

This article is the first complete *Texture First* strategy proposed in the literature. The authors first accurately segment the iris region in the image and then propose a way to estimate the shape of the contours using the shape of the region. This article was used as a basis for our work on iris texture segmentation and we describe more deeply its technical aspects in Section 5.3.1 of Chapter 5.

In [62], Roy et al. propose a complete recognition system to handle non-ideal iris images. They first segment the pupil and the iris using two different kinds of active contours and then they propose several feature extraction algorithms. These features are merged using genetic algorithms and Support Vector Machines. The pupil is first segmented using Geodesic Active Contours in a similar way Shah et al. [66] did for the iris segmentation, and then they segment the iris texture using a Chan and Vese active contour [8] to segment the iris. Finally, they propose to refine the segmentation mask by segmenting the eyelids assuming a parabolic shape. Then they remove the eyelashes using the process described in [37]. Normalization contours are found the same way [66] do.

The segmentation method proposed in this article is very close to the article of Shah et al. [66] but they propose to describe the iris region using its intensity distributions rather than the gradient of its boundaries.

In [59], Proenca proposes an interesting extension of He et al.'s method [26] for segmentation. He focuses on the definition of image quality criteria for images acquired in visible wavelength. The author proposes to apply the segmentation method [26] to generate the segmentation mask of the iris region. Then Proenca retrieves the normalization contours by robust variational fitting of a contour contours described by Fourier series coefficients.

The proposed method uses a *Contours First* strategy [26] to generate the segmentation masks, but, instead of using the contours of the original method, Proenca proposes an alternative method to get the iris borders from the segmentation mask. The interest of this formulation is that it can basically be used as a complement to any method first generating a segmentation mask. However, the fitting is done on the border of the segmentation mask. Therefore, errors in the accuracy of the segmentation mask may corrupt the contour fitting stage.

In [68], Sutra et al. propose a method in between *Contours First* and *Texture First* strategies. They propose to segment the boundaries of the iris region in the image (different from the anatomic iris borders) using the Viterbi algorithm. This algorithm is an adaptation of a classical algorithm from the signal transmission theory. Applied to the iris segmentation problem, it can be seen as finding the contour of minimal length going around the iris region while following areas of strong gradient. The authors first roughly localize the pupil to transform the image into polar coordinates. In the polar image they search for the path of minimal length linking the two sides of the image and passing through areas of strong gradient. The path closest to the center is the pupil one and the further one is the iris path. The image region contained between these two paths is the iris region. In order to obtain the parametric description of the iris borders for normalization, the Viterbi algorithm is applied a second time at a lower resolution to get a rougher representation of the iris borders. The interest of this rough representation is that more strength is given to the areas where it is more likely to have an iris border (right and left regions) and less to areas where it is more likely to have occlusions (upper and lower regions). These contours are used to apply a circular Direct Least Square to fit the parametric description of the iris contours.

Although this article focuses on a contour finding method the authors clearly apply a *Texture First* strategy because they first segment the iris region and then propose a way to get the parametric description of the iris borders. An interesting aspect of this approach is that they do not directly rely on the region segmentation to fit the parametric contour, therefore they do not depend too much on the accuracy of the first run of the Viterbi algorithm.

An interesting aspect of *Texture First* strategies is their ability to precisely segment iris texture almost independently of the occlusion types. However, they require an effective scheme to get the parametric description of the iris contours based on the shape of the segmented region. A limitation of the presented articles is that they often strongly rely on the accuracy of the region segmentation results to estimate the shape of the iris borders. Therefore, inaccuracies in the region segmentation process will lead to inaccuracies on the borders'

description.

| Article | Initialization | Segmentation | Refinements | Unwrapping contours |
|------------------------|--|---|---|---|
| Vatsa et al. 2008 [76] | Pupil and iris initialized as ellipses | Level set based on Chan and Vese formulation [8] | Eyelid segmentation | Initial ellipses |
| Shah et al. 2009 [66] | Segmentation of the pupil and circular DLS | Segmentation of the iris using Geodesic Active Contours | Iris fitting using circular DLS on points of the limbus boundary | Refined circles |
| Roy et al. 2011 [62] | Pupil contour initialized using elliptic DLS | Geodesic Active Contours for the pupil. Level set based on Chan and Vese formulation [8] for the iris | Circles are found in the same way as Shah [66]. Eyelid segmentation and eyelash suppression | Refined Circles |
| Proenca 2010 [59] | - | Segmentation using the method [26] | Recursive fitting of a Fourier series to the iris borders | Parametric contours given by Fourier series |
| Sutra et al. 2012 [68] | Rough pupil localization | Viterbi algorithm in polar coordinates for the pupil and the iris | Downsampled Viterbi algorithm and circular DLS | Refined circles |

Table 2: Relevant iris segmentation/normalization algorithms from the literature using *Texture First* strategies. DLS stands for Direct Least Square

3.4 Focus of this Thesis

We have identified two main strategies for iris segmentation, namely *Contours First* and *Texture First* strategies. In this thesis, we have decided to apply a *Texture First* strategy for two main reasons. The first is that *Contours First* strategies are closer to the original articles of Daugman and Wildes. Therefore, they have been more intensively studied than *Texture First* strategies which are relatively new (the first article we identified as *Texture First* has been published in 2008). The second reason is that these strategies allow to make very few assumptions on the content of the image, as we have discussed in Section 3.3. We find this property particularly interesting for handling the degradations that may appear in the image when relaxing acquisition conditions (See Section 1.6 of Chapter 1).

The overview of the complete system that we developed during this thesis is described in Appendix A. During the development of this system, we faced four critical issues for which we have proposed original solutions compared to the literature (presented in Sections 3.2 and 3.3) :

- (i) *Rough Pupil Localization* : Localization of the pupil area in the whole image (Chapter 4).
- (ii) *Iris Region Segmentation* : Segmentation of the iris region in the image (Chapter 5).
- (iii) *Contour Fitting* : Fitting of a parametric contour on the iris borders (Chapter 6).
- (iv) *Segmentation Quality* : Evaluation of the quality of the segmentation results (Chapter 7).

Concerning part (i), every article presented in Sections 3.2 and 3.3 first roughly localizes the pupil region and then applies some more complex algorithms based on this localization. Therefore, this step of *Rough Pupil Localization* is of crucial importance indeed, a failure at the *Rough Pupil Localization* step will result in the failure of the complete segmentation process. The difficult aspect of this localization is to distinguish the pupil from other dark elements in the eye such as eyebrows, packed eyelashes, hair or the arms of glasses. To solve this issue, we propose to first identify eyebrows and eyelashes regions and then to remove them from the pupil localization process. We describe this process in Chapter 4.

Concerning part (ii), as we have decided to use a *Texture First* strategy, we first segment the regions before fitting the parametric contour in the image. Therefore, *Iris Region Segmentation* is challenging because the algorithm should not only reach a good precision, but it must also be very robust to handle image variability. Accordingly, we focused on Active Contours for iris similarly as the articles [76, 66, 62]. We present several implementations of such algorithms in Chapter 5 and discuss their advantages and limitations.

Concerning part (iii), *Texture First* strategies described in our survey usually find the parametric contours of the iris borders by making a strong use of segmentation results. This tends to create a lack of robustness in the contour fitting scheme because inaccuracies in segmentation results may lead to large failures in the contour fitting process. To solve this issue, we developed a new algorithm based on the formalism of Active Contours. This algorithm allows to fit robustly and accurately an elliptic contour on the iris borders. We describe our new model in Chapter 6.

Concerning part (iv), we were concerned about detecting large failures of the segmentation process. Indeed, segmentation failures have a large impact on the recognition performance of the global system [46]. In order to reduce the error rate of the system, it is necessary to detect such segmentation failures ; besides, this opens other possibilities like reacquiring the image if possible, or changing the algorithm's settings in order to correctly segment the image. Unlike the pupil and iris segmentation issues that are widely studied in the literature, the estimation of the segmentation quality and the detection of failures have not been widely studied so far. Indeed, few work in the literature target this critical problem [38, 29, 81, 30, 80]. Traditionally, the algorithms estimating the image quality are often placed either at the image acquisition level to check the quality of the raw image, or after the segmentation module to assess if the biometric sample has enough information to characterize the iris. In Chapter 7, we propose a set of new segmentation quality measures. We explain why we choose them and how they are fused to produce a final segmentation quality score.

Part II

Contributions of this Thesis

CONFIDENTIAL

CONFIDENTIAL

Chapter 4

Automatic Localization of the Pupil Area in the Eye

The pupil is the dark hole located inside the iris allowing the light to enter the eye. This area is usually very dark and homogeneous. For this reason, it is considered as the most stable element in an eye image acquired in near infrared illumination. Therefore, most iris segmentation algorithms start by a rough localization of the pupil area to initialize more complex algorithms. However, if the localization of the pupil fails, the algorithms may be initialized very far from the actual location of the iris, leading to a complete failure of the segmentation process.

Although the pupil is usually one of the darkest elements in the eye image, it may not be the only dark region : eyelashes, eyebrows or hair may have intensities very close to those of the pupil area (see Figure 17). It is obviously critical to be able to distinguish the pupil from these other dark elements in the eye. Our proposal to handle this problem is inspired from the work of Pundlik et al. [61] and aims at roughly segmenting the eyelashes and eyebrows in order to remove them from the pupil localization process. Let us note that this approach differs from most eyelash segmentation methods : usually eyelashes are removed directly from the segmentation mask assuming that the pupil's location is known.

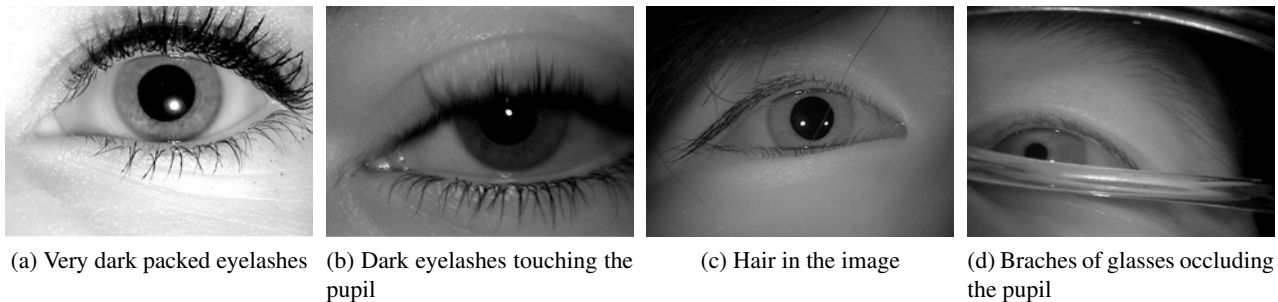


Figure 17: Some difficult images for pupil localization. (Figure (a) and Figure (b) are taken from the ND-Iris database and Figure (c) and Figure (d) are taken from the CASIA Thousand database)

In order to identify eyelash and eyebrow areas we design two algorithms that we describe in Section 4.1. This information on eyelash location allows enhancing the pupil area in the image as we explain in Section 4.2. Enhancing the pupil area makes its detection easier and, therefore, reduces the amount of mis-localization of the pupil area. Finally, in Section 4.3 we evaluate the two algorithms we have developed for eyelashes detection on two different databases : ND-Iris and Casia Thousand.

4.1 Segmentation of eyelashes and eyebrows

Eyelashes are usually highly textured whereas the pupil is rather homogeneous. In [61], Pundlik et al. propose to segment the eyelashes using this textural information. To that aim, they propose to use Harris corner detection framework [25] in order to identify textured regions. Let's consider the operator :

$$G(\mathbf{x}) = \sum_{\mathbf{x}' \in N_g(\mathbf{x})} \begin{bmatrix} I_x^2(\mathbf{x}') & I_x(\mathbf{x}')I_y(\mathbf{x}') \\ I_x(\mathbf{x}')I_y(\mathbf{x}') & I_y^2(\mathbf{x}') \end{bmatrix}, \quad (14)$$

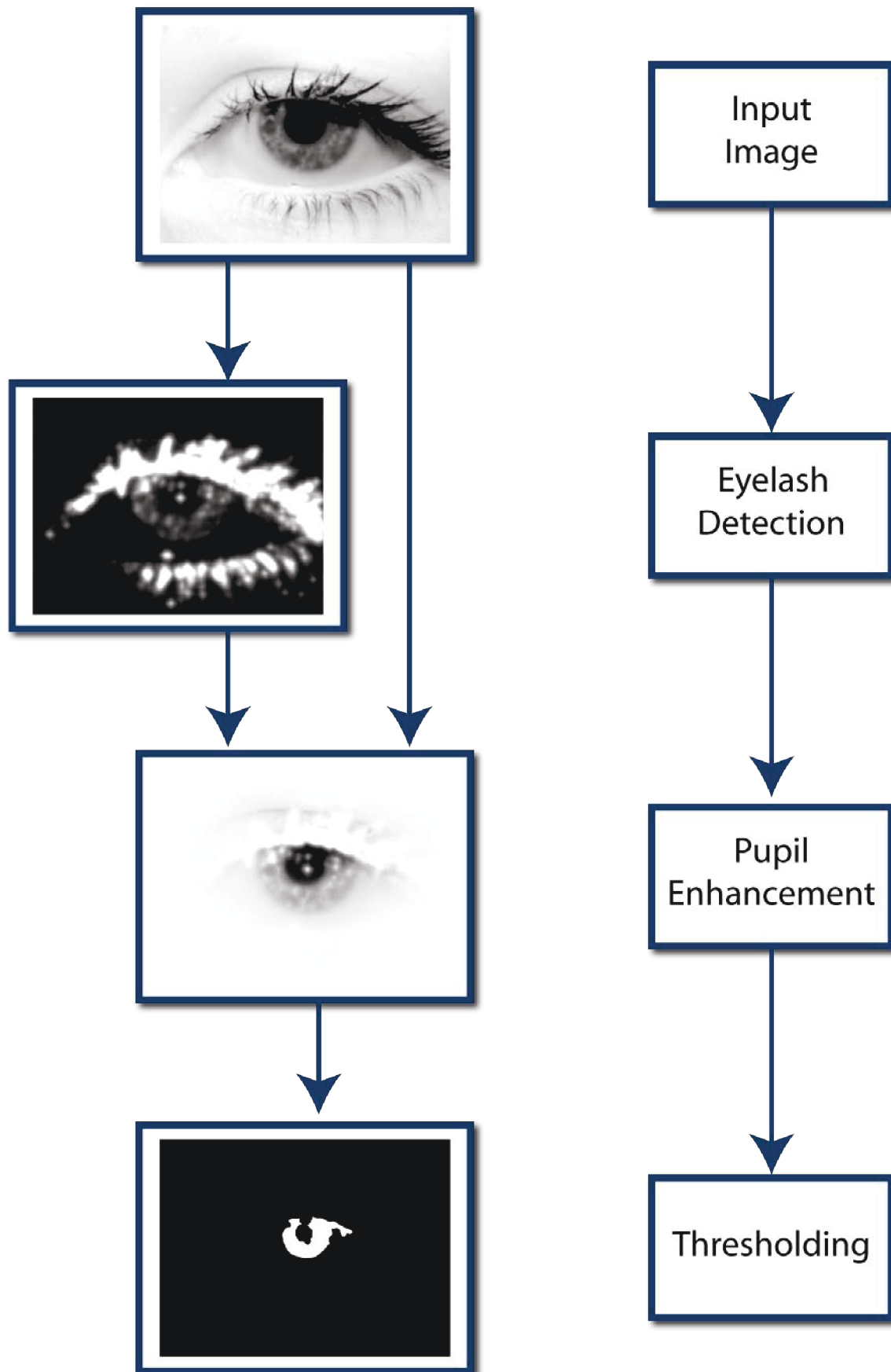


Figure 18: Rough localization of the pupil area. First, the eyelash regions are detected in the original image. This information is used to enhance to pupil region in the original image. Enhancement is done in such a way that the pupil area can be retrieved using simple thresholding techniques.

where \mathbf{x} is a given pixel of the image I , $N_g(\mathbf{x})$ is a neighborhood of \mathbf{x} , $I_x(\mathbf{x})$ the horizontal gradient at pixel \mathbf{x} and $I_y(\mathbf{x})$ the vertical gradient at pixel \mathbf{x} . Let e_1 and e_2 be the two eigenvalues of $G(\mathbf{x})$ with $e_1 \leq e_2$. The eigenvector associated to the largest eigenvalue represents the dominant direction of the texture's orientation in the neighborhood $N_g(\mathbf{x})$. The strength of the eigenvalues characterize the nature of the region around \mathbf{x} . If both e_1 and e_2 are small, \mathbf{x} belongs to an homogeneous region. If $e_1 \gg e_2$ and e_2 is small, \mathbf{x} belongs to an edge. If both e_1 and e_2 are large, \mathbf{x} belongs to a region where the gradient is isotropic, namely a corner point. To detect the corner points in the image, Pundlik et al. set an empirical threshold τ and select the points where $h(\mathbf{x}) = \min\{e_1, e_2\} > \tau$.

This process generates a binary map of corner points in the image. Eyelash regions are expected to contain a higher density of corner points than the other regions in the image. In order to estimate the density of corner points, the authors apply a weighted sum of the radial histogram at each point (this can be seen as a smoothing of the binary image). This process results in a dense map of the textured areas.

The main limitation of the approach proposed in [61] is the definition of one global empiric threshold τ . Indeed, the strength of the eigenvalues can however vary depending on the blur in the image. Thus, influencing the optimal value of τ . Therefore, we propose to use the same idea as Pundlik et al. [61] (eyelashes are in areas of high frequency), but we propose two different methods based on an adaptive thresholding to identify these regions. The first method we call *Texture-Based* is a direct extension of the works of Pundlik et al. with an adaptive thresholding process. The second method we call *Probabilistic-Based* is based on a *Reductio ad absurdum* on the intensity distribution of the high frequency image.

In order to obtain the final eyelash location, We fuse these results using the scheme presented in Figure 19. In addition to the two eyelash location information, we also favor dark regions in the image. Therefore, our fusion scheme is the product of three images of probability : probability of the *Texture-Based* approach, probability of the *Probabilistic-Based* approach and probability of being in a dark region. As the product

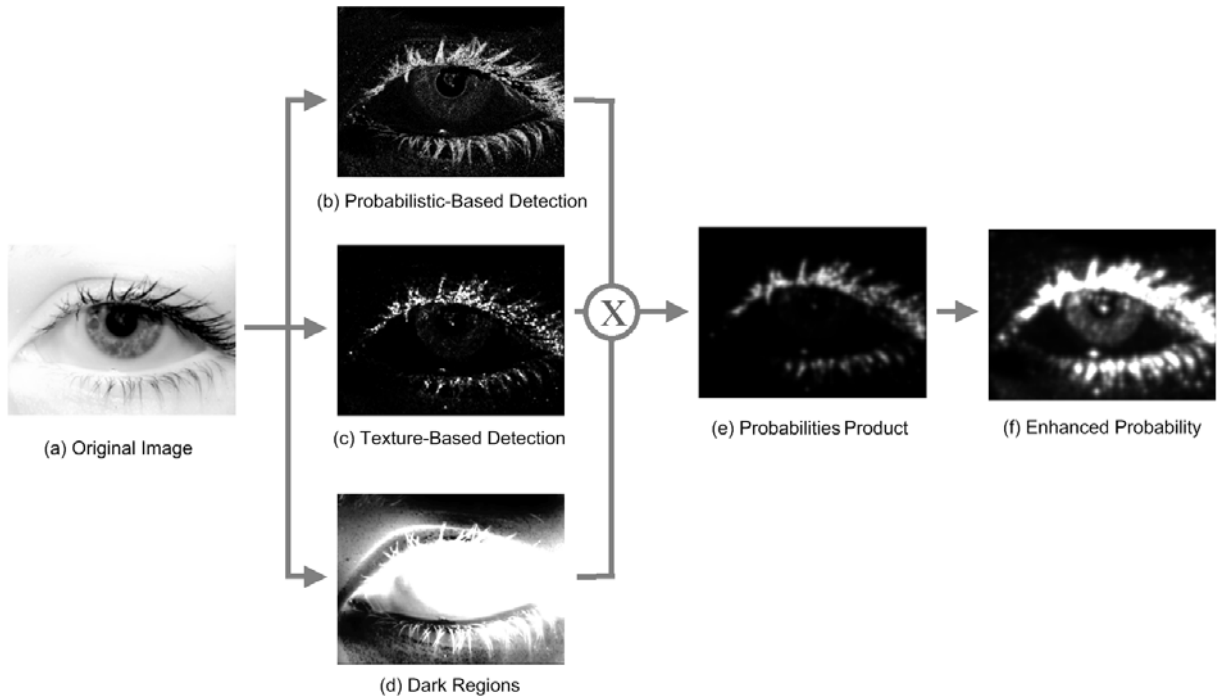


Figure 19: Figure (a) : Original image. Figure (b) : Probability P_p . Figure (c) : Probability P_t . Figure (d) : Probability P_d . Figure (e) : Product of the three probabilities P_{ee} . Figure (f) : Final enhanced probability P'_{ee} .

operation tends to generate a few high intensity pixels and mostly low intensity pixels, we correct the intensity in the image to have a better repartition in the intensity distribution.

4.1.1 Texture-Based Eyelash Detection

An empiric analysis of the operator $h(\mathbf{x}) = \min\{e_1, e_2\}$ presented in the above section shows that its histogram follows a decreasing exponential law. If we note H the histogram of the operator h normalized between 0 and 1 we assume that we have :

$$\begin{aligned} [0, 1] &\longrightarrow [0, 1] \\ H(t) &= \exp\left(-\frac{t}{\sigma}\right), \end{aligned} \quad (15)$$

where σ is the time constant of the exponential law. The value of σ is related to the blur in the image : it is low for blurred images and high for sharp images. Figure 20 gives examples of such histogram for three different images. The first two images are rather sharp, so the slope of their histogram H is very high. On the opposite, the last image is rather blurry so the slope of the histogram is quite low compared to the two previous images.

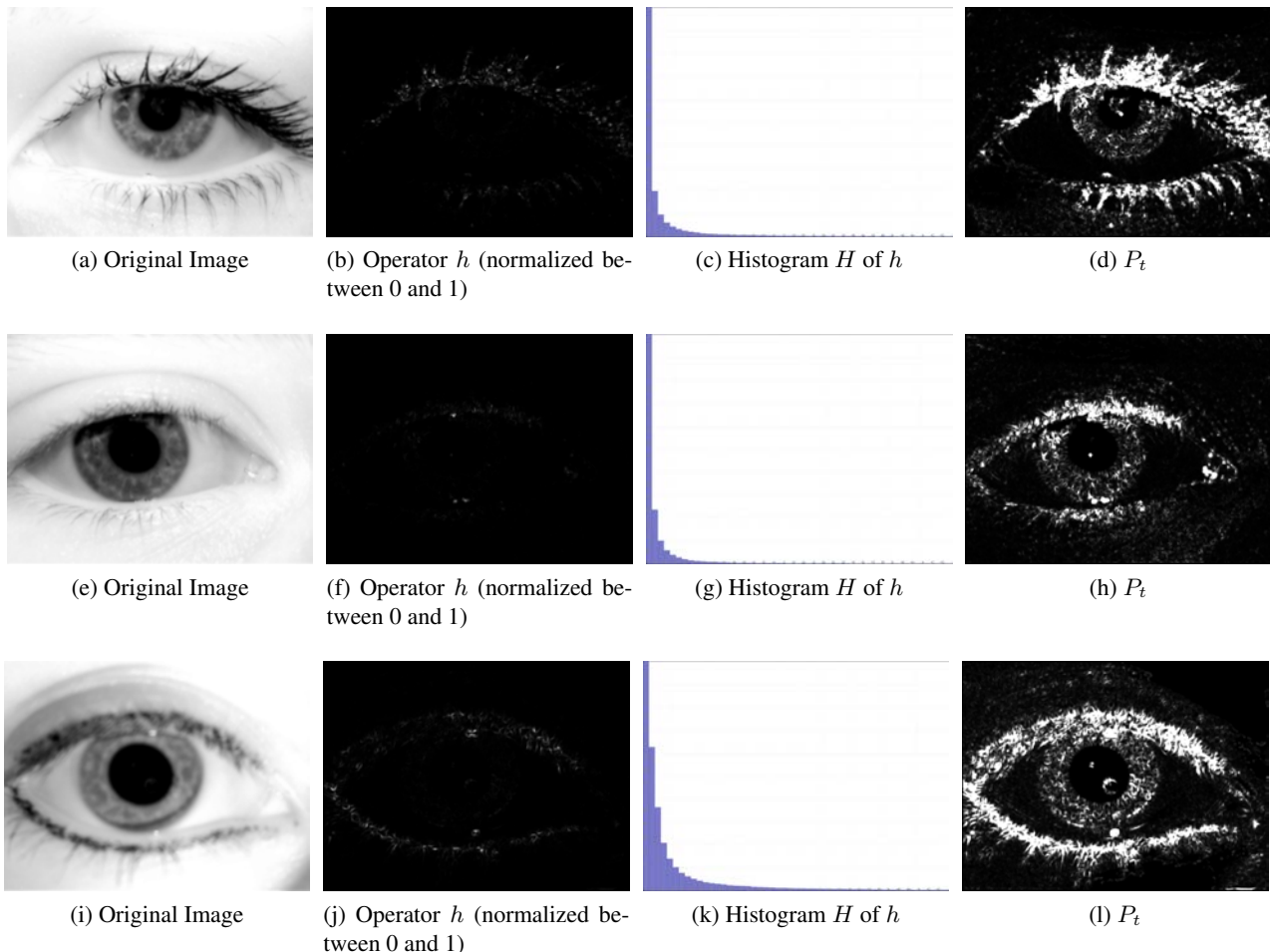


Figure 20: Eyelids area enhancement based on the *Texture-Based* model. The first column presents the original image. The second column presents the operator h normalized between 0 and 1. This operator gives a indication on the eyelash location but their is a large gap between pixels having high intensity and pixels having low intensity. The third column presents the histogram H of the operator h . Finally, the last column presents P_t , the result of the *Texture-Based* eyelash detection process.

Therefore, we propose to use this information to automatically adapt the empirical threshold τ of Pundlik et al. [61].

We first estimate $\hat{\sigma}$ using the slope of the histogram at the origin $H'(0) = -\frac{1}{\hat{\sigma}}$ and apply the following transform to the operator h of the previous section :

$$P_t(\mathbf{x}) = \min\left\{\frac{h(\mathbf{x})}{10 \cdot \hat{\sigma}}, 1\right\}, \quad (16)$$

P_t can be seen as a limitation of the intensity at a maximum of $10\hat{\sigma}$ with a linear normalization between 0 and 1. The interest of P_t as opposed to the process proposed in [61] is that the threshold is set adaptively, and P_t has values between 0 and 1 instead of strictly binary values.

Figure 20 illustrates the process for several images with different types of eyelashes. The image in Figure 20a has very dark and packed eyelashes. The image in Figure 20e has very thin eyelashes. The image in Figure 20i has very thin and dark eyelashes with a large focal blur. The results of the proposed process are displayed in the last column and show that our process can automatically highlight the eyelashes in the three images.

4.1.2 Probabilistic-Based Eyelash Detection

In [33], Kang et al. detect eyelashes using high frequency of the image. They consider the average amount of high frequencies in the image to adapt a two-fold algorithm for eyelash detection. They use a region based algorithm to select the packed eyelashes and a filter-based scheme to select the separable eyelashes.

To extract this high frequencies map the authors take the difference between the original image and a blurred image :

$$HF = I - K * I, \quad (17)$$

where K is a blurring kernel. The approach proposed in [33] shows that it is possible to identify eyelashes in the high frequency image HF . However, this image HF also contains other high frequency elements such as edges and noise.

Based on the idea of Kang et al. [33], our proposition to find eyelashes in HF is divided in two steps :

- (i) Defining an adaptive threshold to identify pixels of higher frequency in HF .
- (ii) Distinguish eyelashes areas from edges and noise.

Concerning (i), we apply a *Reductio ad absurdum* assuming that HF is only composed of a Gaussian white noise, i.e. that there is no information in the image (no edge nor corner points). As we know this assumption is wrong, we can check for incoherency between the model and the actual HF to extract the eyelash. This process is illustrated in Figure 21. First we compute HF as the difference between the original image and its blurred version Equation (17). Then, we compute σ the variance of HF . If the white noise assumption was true, more than 99% of the pixels in the image should fall in the interval $[-3\sigma; 3\sigma]$ and they should be widespread randomly in the image. Accordingly, we mark all the pixels in the image above 3σ and below -3σ :

$$E_p(\mathbf{x}) = \begin{cases} 0 & \text{if } HF(\mathbf{x}) \in [-3\sigma; 3\sigma] \\ 1 & \text{otherwise} \end{cases}. \quad (18)$$

Concerning (ii) : eyelashes are usually located in dense regions of E_p , whereas edges are usually located along thin lines and the noise is spread randomly in the image. Therefore, blurring the binary image removes both the edges and the noise, whereas it homogenize the eyelash regions. Accordingly, we generate the eyelash probability map by blurring E_p with a Gaussian kernel of variance τ : G_τ .

$$P_p = G_\tau * E_p, \quad (19)$$

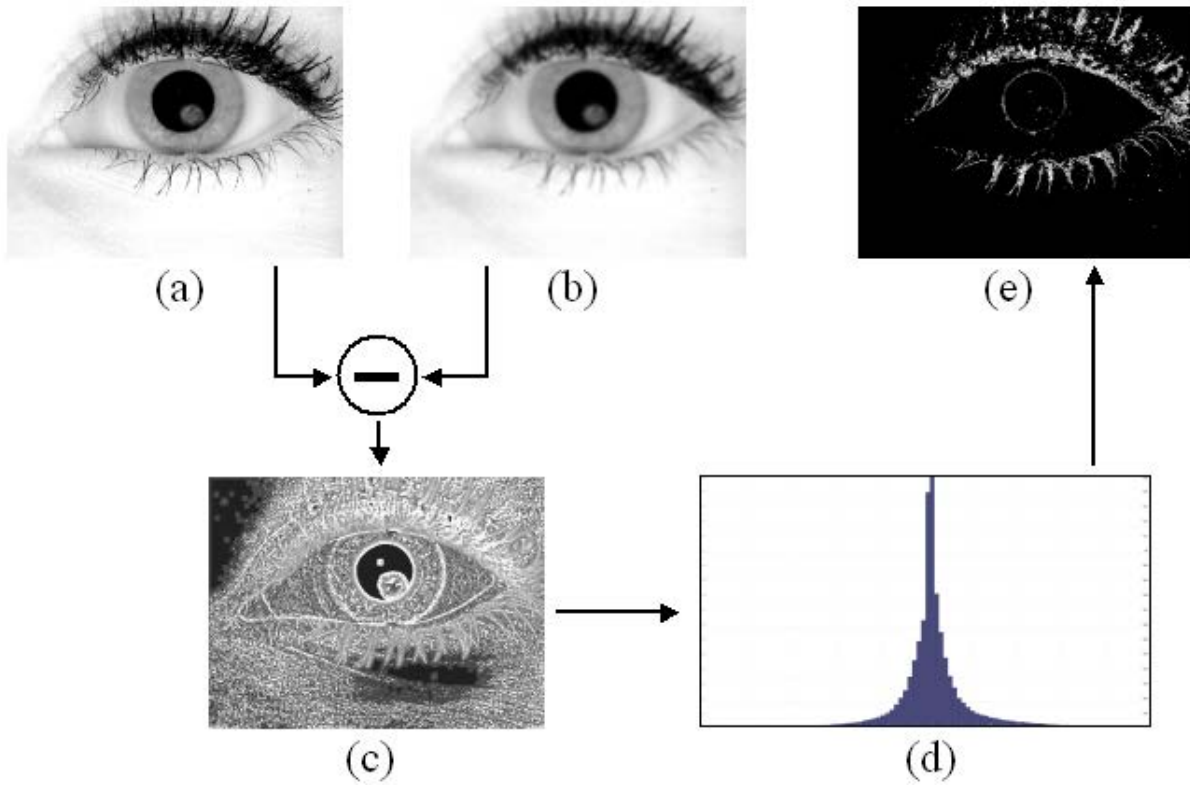


Figure 21: Figure (a) : Original image. Figure (b) : Blurred image. Figure (c) : High frequency image obtained as the difference between the first two images (histogram equalization is applied to improve visualization). Figure (d) : Histogram of the high frequency image, values are between $[-50, 50]$ and the histogram spike is at 0. Figure (e) : Pixels of Figure (c) having a value below -3σ or above 3σ , where σ is the variance of the histogram in (d).

4.1.3 Retrieving the Eyelash Map

We merge the two operators P_p and P_t of the previous sections in order to get the expected location of eyelashes and eyebrows. As we are only interested in eyelashes and eyebrows that can be wrongly detected as pupil area, we add a constraint P_d to select only dark areas in the image.

$$P_d(\mathbf{x}) = \max\left\{\frac{255 - 5 * I(\mathbf{x})}{255}, 0\right\}, \quad (20)$$

The Figure 19 we presented earlier in this section illustrates the complete eyelash localization process. $P_p(\mathbf{x})$, $P_t(\mathbf{x})$ and $P_d(\mathbf{x})$ are computed separately and merged using equation :

$$P_{ee}(\mathbf{x}) = P_p(\mathbf{x}) \cdot P_t(\mathbf{x}) \cdot P_d(\mathbf{x}). \quad (21)$$

The product of Equation (21) allows getting an estimation of the eyelash's expected location in the dark regions. However, the product operator tends to produce a large difference between high and low values in P_{ee} . Therefore, we propose to enhance the probabilities using the same scheme we used for P_p in Section 4.1.1. We also apply a Gaussian blur of variance 5 to smooth the image leading to the equation :

$$P'_{ee}(\mathbf{x}) = G_5 * \min\left\{\frac{h_{P_{ee}}(\mathbf{x})}{10 \cdot \sigma}, 1\right\}, \quad (22)$$

where $h_{P_{ee}}$ is the histogram of P_{ee} normalized between 0 and 1, and G_5 is a Gaussian blur of variance 5.

4.2 Rough Pupil Segmentation

In this section, we propose to build an image where the pupil is the darkest element. To do so, we use the prior knowledge we have on the properties of the pupil :

- The pupil is a dark region in the original image.
- The pupil is not in the eyelash regions.
- The pupil should be close to the specular reflections generated by the illumination device [46].

Accordingly, we use these three assumptions to enhance the pupil region as presented in Figure 22. First, we use the *Eyelash map* we built in Section 4.1 to discard the eyelash areas. Then we consider the input image I_n normalized between 0 and 1. Finally, we use the location of the specular reflexions S to localize the pupil region (see Annexe A for details about the generation of S). As S is a binary mask, we blur this mask using a Gaussian kernel of large variance to generate a smoother map using the equation :

$$S_n = 1 - \frac{G_\sigma * S}{\max_{\mathbf{x} \in S} G_\sigma * S(\mathbf{x})} \quad (23)$$

In order to fuse these three probability information ($I_n(\mathbf{x})$, P'_{ee} and S_n), we simply use a product of the three probabilities :

$$I_{pupil}(\mathbf{x}) = I_n(\mathbf{x}) \cdot P'_{ee}(\mathbf{x}) \cdot S_n(\mathbf{x}). \quad (24)$$

where \mathbf{x} is a pixel in the image.

This overall process ensures that the pupil is the darkest region in the image. The rough segmentation of the pupil is achieved using a simple hysteresis thresholding. The image I_{pupil} is normalized between 0 and 1 and the first threshold is set to 0.01 and the higher threshold is set to 0.1. If several disconnected components are generated using this process, the region of largest area is labeled as being the pupil region.

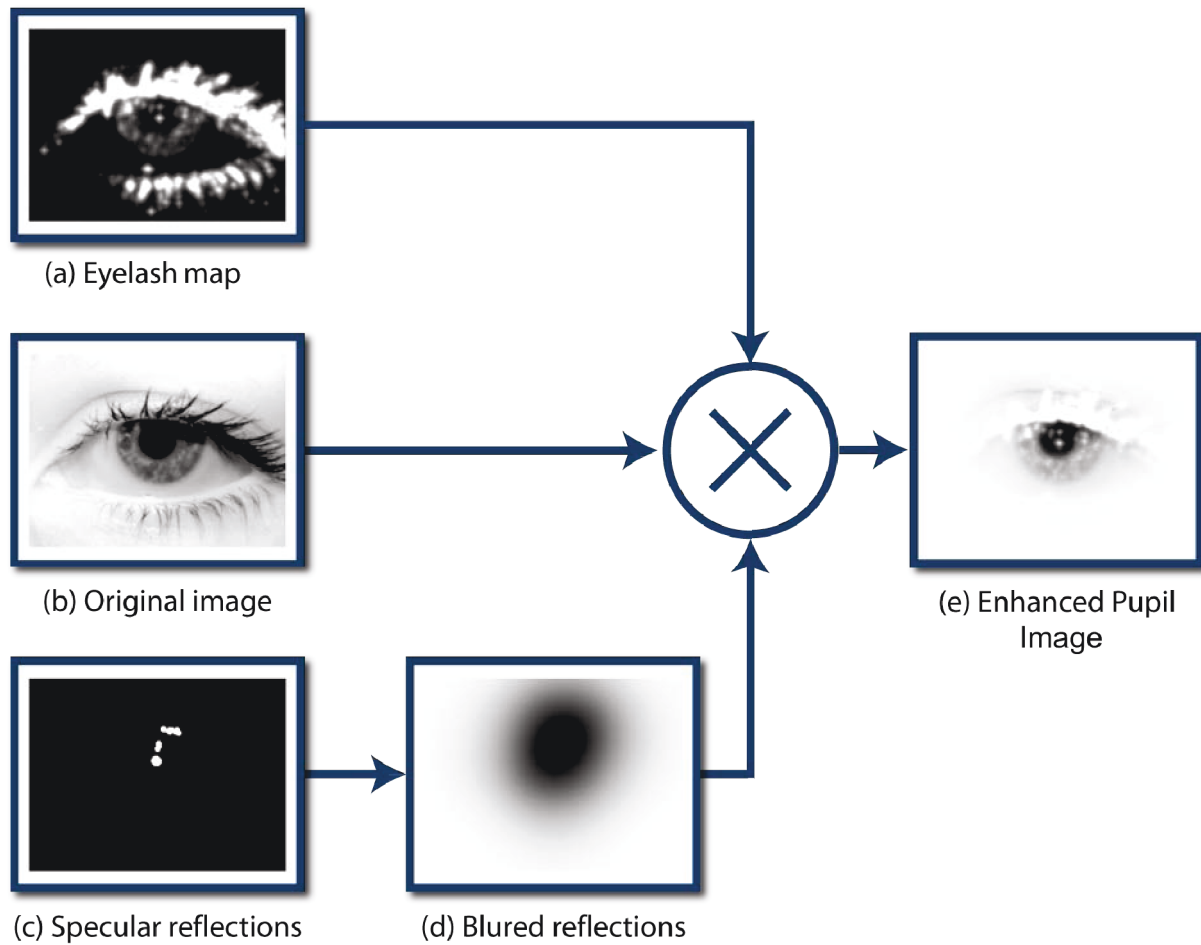


Figure 22: Generation of the enhanced map. Figure (a) : the eyelashes map P_{ee} . Figure (b) : the original image normalized between 0 and 1 I_n . Figure (c) : mask of the specular reflections in the image. Figure (d) : blurred and inverted image of the specular reflections. Figure (e) : composite image enhancing the pupil region.

4.3 Evaluation

In this section, we propose to evaluate the *Texture-Based* and the *Probabilistic-Based* eyelash detection methods presented in Section 4.1.1 and 4.1.2. We use either one of the algorithm, none of them or both of them to generate the pupil enhanced image I_{pupil} of Section 4.2. Then we assess the impact on the system recognition performance. This leads to four different system configurations :

Simple Model Pupil location extracted without eyelash information (i.e. we only use the intensities and specular reflections locations).

Texture-Based Model Pupil location extracted with eyelash information provided by the method described in Section 4.1.1

Probabilistic-Based Model Pupil location extracted with eyelash information provided by the method described in Section 4.1.2

Complete Model We use the information of both Section 4.1.1 and Section 4.1.2 to locate the pupil.

Figure 23 shows the ROC curves associated to the four above scenarios on the ND-Iris and CASIA-Thousand databases. We decide to evaluate the recognition performance of our system on these two databases because they do not share the same kind of degradations. The ND-Iris database contains a significant amount of images with packed eyelashes and eyebrows like the image shown in Figure 17a and Figure 17b in the introduction of this chapter. The CASIA Thousand database contains mostly hair and glasses that we do not specifically handle with our proposed algorithms (Figure 17c and 17d).

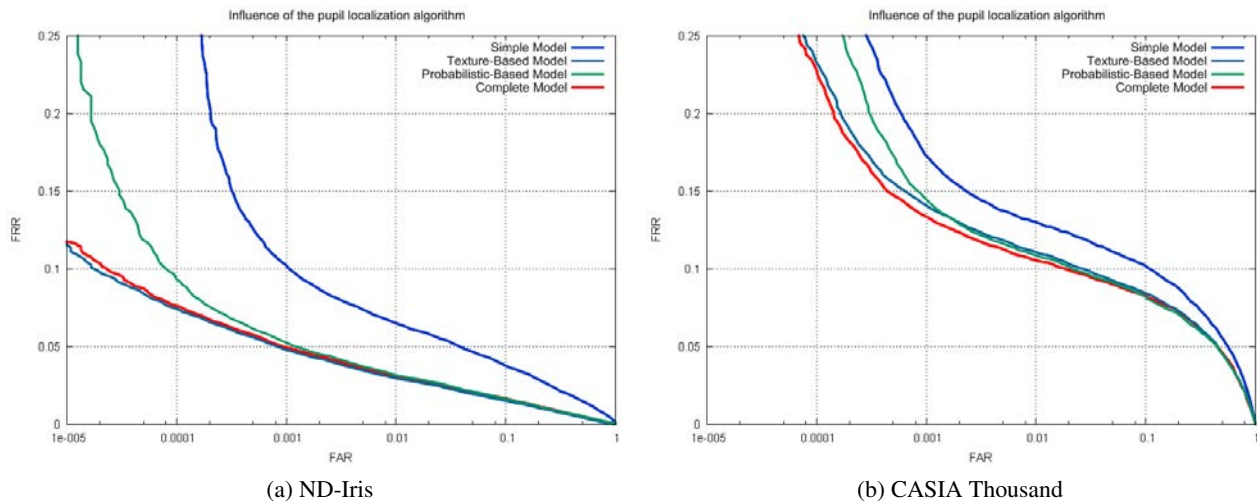


Figure 23: ROC curve of the recognition performance of the system when using different eyelash detection schemes to help the localization of the pupil. Figure (a) shows the results on the ND-Iris database and Figure (a) shows the results on the CASIA Thousand database.

First we can note that for both databases the ranking of the four models is the same: The *Simple model* performs the worst followed by the *Probabilistic-Based Model*. Finally, the *Texture-Based Model* and *Complete model* perform roughly as well.

The *Texture-Based Model* performs the best because it can very efficiently identify eyelashes in almost any kind of images thanks to the adaptive thresholding process. Though the *Probabilistic-Based Model* has a good influence on the recognition performance, it does not perform as well as the *Texture-Based Model*. We noticed that, a fusion of the two models does not provide a significant improvement compared to the

single *Texture-Based Model*. This behavior illustrates that the *Probabilistic-Based Model* provides information that are redundant with the one provided by the *Texture-Based Model*. It is an unexpected result because the two methods are based on different assumptions so we would have expected to get uncorrelated results and, therefore, an improvement when merging these two information. As the *Complete Model* does not provide a significant improvement compared to the *Texture-Based Model*, we propose to simply use the *Texture-Based Model* to detect the eyelash regions in the image.

It is also interesting to note that the improvement between the *Simple Model* and the other experiments is more significant on the ND-Iris database than on the CASIA-Thousand database. This statement is coherent with the knowledge we have on the degradations contained in the databases. As the ND-Iris database contains images with packed eyelashes and eyebrows, both the *Texture-Based Model*, the *Probabilistic-Based Model* and the *Complete Model* significantly improve the performance compared to the naive *Simple Model* based on a thresholding algorithm. On the other hand, for the CASIA-Thousand database, the worst degradations are due to glasses, a degradation that our model do not specifically handle. Figure 24 illustrates some pupil localization failures due to the presence of glasses. In this case, it would be necessary to perform glasses detection and adapt the segmentation process accordingly.

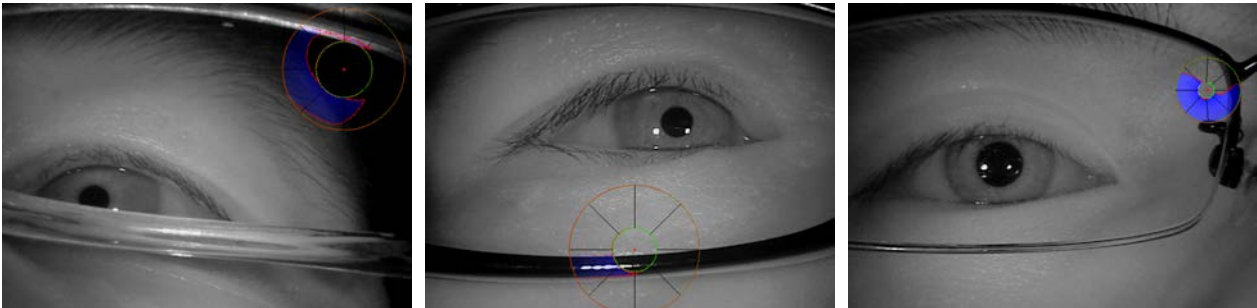


Figure 24: Pupil localization failures due to the presence of glasses in the image. (CASIA Thousand)

4.4 Conclusion

In this chapter, we have described our algorithm to localize the pupil area in the eye. This algorithm is based on an enhancement of the pupil region by removing other dark areas. This strategy then allows to obtain the pupil area using a simple thresholding algorithm.

We have designed two algorithms for segmenting eyelash and eyebrow areas in the image. The first one, called *Texture-Based*, is a direct extension of the works of Pundlik et al. [61]. We have proposed an effective way to define an adaptive threshold instead of setting it empirically. The second algorithm, that we called *Probabilistic-Based*, is an automatic method for identifying eyelash areas in regions of high frequency.

We showed that using these algorithms provides a significant improvement of the system's recognition performance compared to a naive pupil localization approach. This improvement is due to a decrease of the amount of pupil mis-localization. The *Texture-Based* algorithm performs better than the *Probabilistic-Based* algorithm on the two databases that we considered. Interestingly, we also noticed that the fusion of the two algorithms does not improve the recognition performance compared to the single *Texture-Based* algorithm.

This rough localization of the pupil area allows initializing the complete segmentation process close to the iris location. In the following chapter, we will discuss the next stage of our segmentation system : the accurate segmentation of the iris region.

Chapter 5

Iris Region Segmentation

The iris texture can vary largely from one person to another and it is likely to be occluded by eyelids, eyelashes or other exterior elements such as hair or the arms of glasses. Accordingly, segmentation algorithms should be very precise because inaccuracies in the iris region segmentation will provide wrong information at the matching step, leading to erroneous comparison results. Indeed, in order to segment automatically large databases, segmentation algorithms must be very robust to handle the high variability encountered in iris images.

Active Contours have lately received a large interest in the iris recognition community because they meet both requirements of precision and robustness for iris segmentation. These methods are popular for iris segmentation since Daugman [17] represented the iris borders using parametric Active Contours described by Fourier coefficients. This work was followed by the articles [66, 76, 62] which we already discussed in Chapter 3.

Although Active Contours have become popular for iris region segmentation, there are no comparisons between different implementations of such algorithms for iris segmentation. Our proposal in this chapter is precisely to compare different types of Active Contours in the framework of a complete iris recognition system.

In Section 5.1, we first present the general framework of *Deformable Models* in order to illustrate the differences existing between Active Contours we describe from Section 5.2 to 5.4. In Section 5.5, we evaluate the performance of the considered algorithms. Finally, in Section 5.6, we conclude this chapter and propose some perspectives to our work.

5.1 Deformable Models

Deformable Models were first introduced by Kass et. al in [35] as an explicit deformable contour used for object segmentation. This work was the starting point of a very active branch of the Computer Vision community. In addition to their historical application, i.e. segmentation, *Deformable Models* have been used for a broad variety of applications such as pattern recognition [19, 2], computer animation [73], geometric modeling [42], simulation [12], boundary tracking [5] and image segmentation [49, 48, 69, 32].

Active Contours are a particular type of *Deformable Models* dedicated to segmentation. Active Contours iteratively make a contour evolve so that it fits the boundaries of the object to segment. Figure 25 illustrates the segmentation of an image using Active Contours.

As any *Deformable Model*, an Active Contour is defined by the association of two components :

- The *Shape Description* gives the contour's mathematical model.
- The *Deformation Description* that expresses a set of constraints on the contour as an energy function. This energy should be minimal when the contour fits the object's boundaries.

In this section, we first present these two elements and then, shortly present the Active Contours we have implemented for iris region segmentation in this chapter.

5.1.1 Shape Description

The *Shape Description* expresses how the contour is generated. There are two main families of *Shape Descriptions* resulting in two families of Active Contours : *Parametric Contours* and *Geometric Contours*.

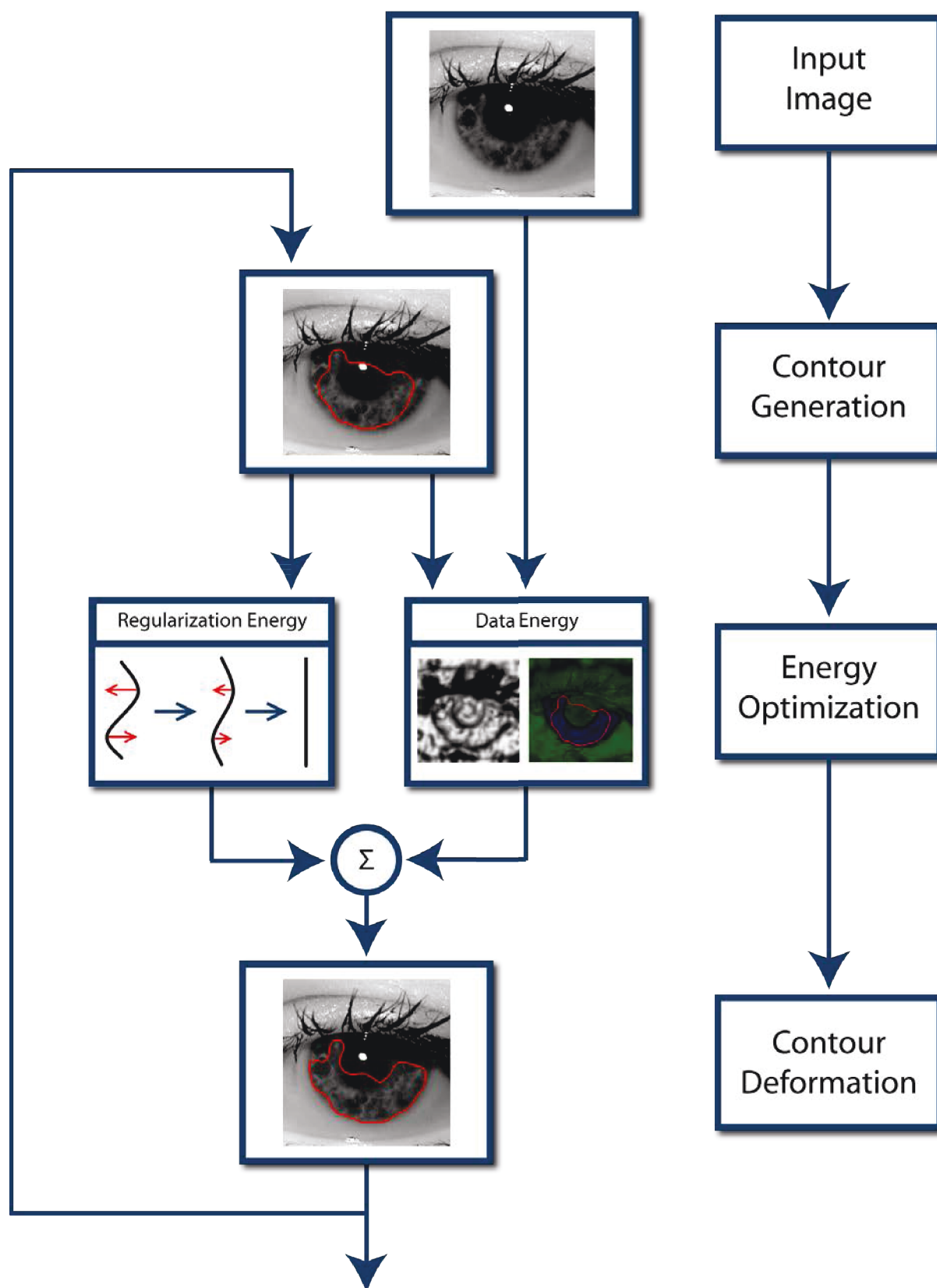


Figure 25: Iterative segmentation work-flow using Deformable Models. This iterative work-flow is divided in three main stages. First, the contour is generated using the *Shape Description* of the model. Then, the variation of the energies defined by the *Deformation Description* are computed. This formulation (*Deformation Description*) is usually composed of two terms : a Regularization Energy and a Data Energy. The regularization Energy ensures that the contour remains smooth whereas the Data Energy pushes the contour onto the borders of the object. Finally, the contour is deformed in order to minimize the energies of the *Deformation Description*.

In case of *Parametric Contours*, the contour is generated using an explicit parametric equation. This parametric equation is characterized by a shape vector $\Theta = \{\theta_i, i = 1 \dots n\}$ with n degrees of freedom of the model. In this context, the contour C is defined as :

$$\begin{aligned} C_{\Theta} : \mathbb{R}^n \times \mathbb{R} &\rightarrow \mathbb{R}^2 \\ (\theta_1 \dots \theta_n, s) &\mapsto \mathbf{x}(s) = (x(s), y(s)) \end{aligned} \quad (25)$$

with $\mathbf{x} = (x, y) \in L^2(\mathbb{R}) \times L^2(\mathbb{R})$, where $L^2(\mathbb{R})$ is the set of the square integrable functions of \mathbb{R} . Using this kind of shape representation allows explicit accessing of the points on the contour by means of parameter s . *Parametric Contours* are so called because of the parametric equations $\mathbf{x}(s) = (x(s), y(s))$.

Geometric Contours rely on the Level Set framework [1] to represent the contour. In this framework, the contour is represented as the 0 Level-Set of a 2-dimensional manifold embedded in a 3-dimensional space. Formally, for a function ϕ of \mathbb{R}^3 , C is defined as $C = \{\mathbf{x} \in \Omega | \phi(\mathbf{x}) = 0\}$. In this framework, the contour is seen as a geometrical object, and therefore, does not have an explicit parametric description like *Parametric Contours* do. Figure 26 illustrates such a geometric definition. The red contour is defined as the intersection between the manifold ϕ and a plane.

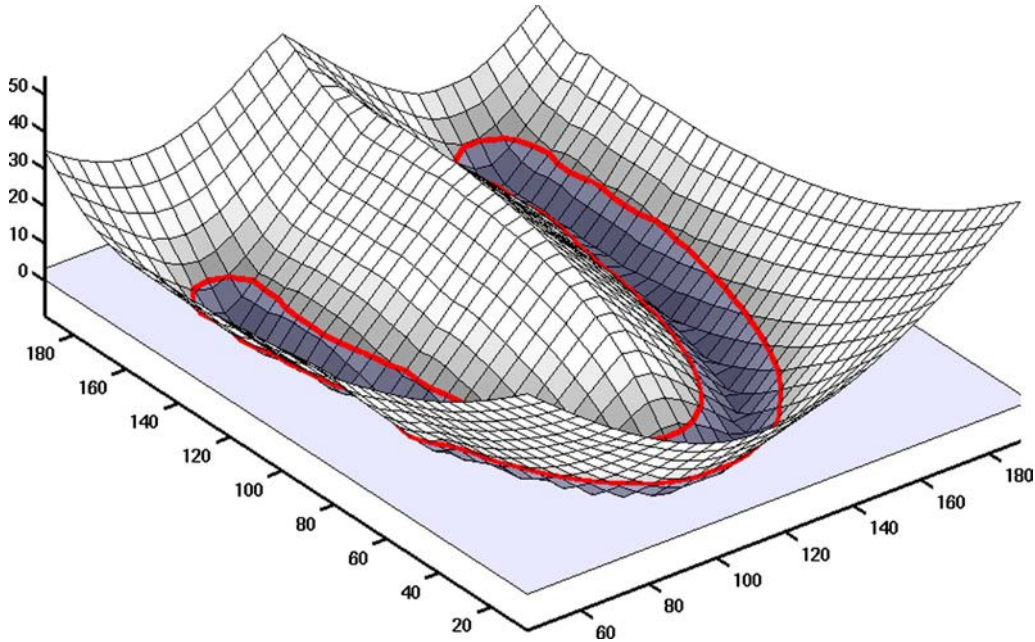


Figure 26: The definition of a curve as being the zero-level of a 2D surface embedded in a 3D space (image taken from [18]).

Parametric Contours are usually more intuitive than *Geometric Contours* are (the curve is generated using an explicit equation). However, due to this formulation, they can only segment shapes having a parametric expression. For instance, they can hardly segment objects composed of multiple disconnected regions.

At the opposite, *Geometric Contours* can take almost any shape during the evolution. For instance, they can naturally split in several regions if the object to segment has several disconnected components. However, the evolution of the contour is done through the evolution of a 3D surface, that is much slower than the evolution of a classical 2D contour.

5.1.2 Deformation Description

The *Deformation Description* is usually expressed as the minimization of an energy functional defined along the contour. Such an energy E is composed of two terms, a regularization term E_{int} , ensuring that the contour

remains smooth, and a data term E_{data} pulling the contour toward specific features of the image. This leads to :

$$\begin{aligned} E : L^2(\mathbb{R}) \times L^2(\mathbb{R}) &\rightarrow \mathbb{R} \\ C &\mapsto E_{int}(C) + E_{data}(C), \end{aligned} \quad (26)$$

The *data energy* E_{data} is related to the representation of the object to segment. Two main families of data terms exist in the literature : an *Edge-Based data energy* and a *Region-Based data energy*. In the *Edge-Based* formalism, the borders of the objects are modeled as areas of strong gradient in the image. Therefore, in case of an *Edge-Based* formalism, the contour is forced to pass through areas of strong gradient. In the *Region-Based data energy* case, the borders of the object are modeled as boundaries between regions; thus, the contour is forced to separate regions having different statistical properties.

The minimization of E in Equation (26) does not have a general analytic solution ; therefore, the minimization is handled using iterative minimization schemes. Figure 25 illustrates such a minimization in the segmentation context. The curve at a given iteration step is generated using its corresponding *Shape Description*, then the energy terms are computed in order to make the contour evolve. This minimization is achieved by a gradient descent algorithm :

$$\begin{aligned} C^{t+\delta t} &= C^t - \delta t \frac{\partial E}{\partial t}(C) \\ &= C^t - \delta t \left(\frac{\partial E_{int}}{\partial t}(C) + \frac{\partial E_{data}}{\partial t}(C) \right) \end{aligned} \quad (27)$$

We have focused on the description of deformable models in an energetic framework which is the one we use in this chapter. However, let us note that there are some alternative frameworks that can be considered for the *Deformation Description*, such as an optimal alignment of the shape with the image using a Markovian framework. We let the reader refer to the surveys [47], [44] and [24] for more details about alternative formulations of *Deformable Models*.

5.1.3 Active Contours for Iris Segmentation

In this chapter, we propose to compare several Active Contours implementations for iris segmentation. They differ either in term of *Shape Description* or *Deformation Description*. Table 3 summarizes these algorithms :

| Data term | <i>Parametric Contours</i> | <i>Geometric Contours</i> |
|----------------------|----------------------------------|---|
| <i>Edge-Based</i> | B-Snakes (Section 5.2.2) | Geodesic Active Contours (Section 5.3.1) |
| <i>Region-Based</i> | B-Snakes (Section 5.2.2) | Chan and Vese (Section 5.3.2) |
| <i>Texture-based</i> | Texture B-Snakes (Section 5.4.5) | Integrated Active Contours for Texture Segmentation (Section 5.4) |

Table 3: The different Active Contours that we implemented for iris segmentation.

In Section 5.2, we first describe some classical *Parametric Contours*. We present both an *Edge-Based* and *Region-Based* data terms for this kind of contours. Then, we present an efficient way to apply the deformations associated to these energies on the contour using B-Snakes [27].

In Section 5.3, we describe some classical *Geometric Contours*. We first present one contour using an *Edge-Based* data term (Geodesic Active Contours [7]) and then, one using a *Region-Based* data term (Chan and Vese [8]). Let us note that, these two algorithms have already been used for iris segmentation. For this reason, we also describe some specific issues identified in the literature when using these algorithms for iris segmentation.

The algorithms that are presented in Section 5.2 and 5.3 are rather classical and make a direct use of the image intensity information in order to make the contours evolve. In Section 5.4, we present instead a specific type of Active Contours that involves texture instead of intensity. In such section, we first describe the *Geometric Contour* described in the original article [64] and then we extend this formulation to *Parametric Contours*.

5.2 Parametric Contours

In this section, we describe some classical *Parametric Contours*. We first present how to express an *Edge-Based* and a *Region-Based data energy* on a contour C described explicitly as :

$$C : \mathbf{x}(s) = (x(s), y(s)), s \in [0, 1]. \quad (28)$$

These formulations are given in the continuous domain. However, their direct discretization leads to some impractical issues like numerical instabilities. For this reason, in Section 5.2.2, we describe an effective way to represent the curve by using a discrete set of control points and interpolation functions.

5.2.1 Data Energies on Explicit Parametric Contours

We first describe the Active Contour introduced by Kass et al. [35] in 1988 that exploits an *Edge-Based data energy*. Then, in Section 5.2.1.2, we describe an extension of this work, proposed by Zhu and Yuille [82] in 1996, relying on an *Region-Based data energy*. Finally, we describe how to merge these two formulations in an unified *data energy*.

5.2.1.1 Edge-Based Data Energy

The first Active Contours were introduced by Michael Kass, Andrew Witkin and Demetri Terzopoulos in their article [35] published in 1988. The objective of the authors is to move the contour on the borders of the object to segment, while still imposing some regularity on the contour shape. The authors express such properties using an objective energy function we described in Section 5.1.2 for *Deformable Models* :

$$E(C) = E_{int}(C) + E_{data}(C), \quad (29)$$

E_{int} is defined to keep the contour smooth :

$$E_{int} = \int_0^1 \frac{1}{2} [\alpha \mathbf{x}'(s)^2 + \beta \mathbf{x}''(s)^2] ds, \quad (30)$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are parameters, $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ are the first and second derivatives of the curve parametric vector \mathbf{x} at point at s . The first term represents a minimization of the contour's length and the second one ensures that the curvature remains small along the contour.

In the article [35], the authors expect the contour to fit the borders of the object. Therefore, they propose to define data energies E_{data} that are small when the contour is on areas of strong gradient (i.e. the borders of the object). There are several classical energies that can be considered for the energy term E_{data} such as :

$$\begin{aligned} E_{data}^{(1)}(\mathbf{x}) &= -\|\nabla I(\mathbf{x})\|, \\ E_{data}^{(2)}(\mathbf{x}) &= -\|\nabla[G_\sigma * I](\mathbf{x})\|, \\ E_{data}^{(3)}(\mathbf{x}) &= \frac{1}{1 + \|\nabla I(\mathbf{x})\|}, \\ E_{data}^{(4)}(\mathbf{x}) &= \frac{1}{1 + \|\nabla[G_\sigma * I](\mathbf{x})\|}, \end{aligned} \quad (31)$$

where ∇ is the gradient operator, $I(\mathbf{x})$ the image intensity at \mathbf{x} , $*$ the convolution operator and G_σ is a Gaussian kernel of variance σ . All these energies are minimum when \mathbf{x} is in an area of strong gradient.

The above energy is minimized using a gradient descent scheme. For an artificial time parameter t , each point of the contour should move according to the equation :

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}''(s, t) - \beta \mathbf{x}'''(s, t) - \nabla E_{data}(\mathbf{x}(s, t)) \quad (32)$$

where $s \in [0, 1]$. Equation (32) can be seen as the motion equation of the contour points.

This first work of Kass et al. is interesting because it defines some key elements leading to the general formulation of *Deformable Models*. However, their work suffers from some limitations :

- The evolution of Equation (32) suffers from numerical instabilities, especially for the term $\mathbf{x}'''(s, t)$ of Equation (32) (third derivative of the position).
- The energies in Equation (31) are defined using the image gradient. However, if the contour is initialized in an homogenous region, there is no gradient information and $E_{data} = 0$. In this situation, the contour has no information to evolve.

5.2.1.2 Region-Based Data Energy

In the article [82], Zhu and Yuille propose an evolution of the classical Snake formulation. Traditional Active Contours (like Snakes) focus on finding areas of strong gradient in the image. At the opposite, Zhu and Yuille see the contour as an interface separating regions. The contour evolves in order to maximize the difference between such regions. Zhu and Yuille call this interpretation the Region Competition principle.

In the Region Competition formalism, regions are represented based on their statistical properties. The contour is represented in terms of a partition $\Omega = \Omega_1 \dots \Omega_N$ of the image I . Each region of the partition Ω_i , $i \in \{1 \dots N\}$ can be described using a set of parameters θ_i (for instance the mean of the region, or its histogram). The authors in [82] propose to maximize the posterior probability of the partition Ω , $p(\Omega|I)$:

$$p(\Omega|I) = p(I|P(\Omega)) p(P(\Omega)), \quad (33)$$

$p(P(\Omega))$ describes the shape of the partition Ω and $p(I|P(\Omega))$ indicates how well the regions match specific features of the image. The authors [82] propose to maximize this posterior probability by iterative evolution of the partition Ω .

The authors [82] consider that the length of the regions' borders should be as small as possible, leading to the following equation for $p(P(\Omega))$:

$$p(P(\Omega)) = e^{-\nu|C|}, \nu > 0, \quad (34)$$

where $|C|$ is the total length of the boundaries in the partition Ω . By definition of an image partition, the regions Ω_i , $i \in \{1 \dots N\}$ do not overlap, so we can write :

$$p(I|P(\Omega)) = p(I|\Omega_1, \dots, \Omega_N) = \prod_{i=1}^N p(I, \Omega_i). \quad (35)$$

Assuming that the statistical properties of the regions are independent, for a probability density p_i in region Ω_i , Equation (35) becomes :

$$p(I|P(\Omega)) = p(I|\Omega_1, \dots, \Omega_N) = \prod_{i=1}^N \prod_{\mathbf{x} \in \Omega_i} p_i(I(\mathbf{x})). \quad (36)$$

In order to have a linear expression, the authors propose to minimize the opposite of the logarithm : $-\log p(\Omega|I)$ instead of maximizing Equation (33). Using the equations (34) and (36), this minimization becomes :

$$E(\{\Omega_1, \dots, \Omega_2\}) = -\sum_i \int_{\Omega_i} \log p_i(I(\mathbf{x}))d\mathbf{x} + \nu|C|, \quad (37)$$

Using any parameter θ_i describing the region Ω_i , the equation becomes :

$$E(\{\Omega_i, \theta_i\}_{i=1..N}) = -\sum_i \int_{\Omega_i} \log p(I(\mathbf{x})|\theta_i)d\mathbf{x} + \nu|C|. \quad (38)$$

If we reduce the problem to a partition of the image into two regions (namely an object Ω_{in} and its background Ω_{out}), Equation (38) can be embedded in the *Deformable Models* framework as follows :

$$\begin{aligned} E(C) &= E_{int}(C) + E_{data}(C), \\ \text{with } \begin{cases} E_{int}(C) &= \nu|C|, \\ E_{data}(C) &= -\int_{\Omega_{in}} \log p(I(\mathbf{x})|\theta_{in})d\mathbf{x} - \int_{\Omega_{out}} \log p(I(\mathbf{x})|\theta_{out})d\mathbf{x}, \end{cases} \end{aligned} \quad (39)$$

First, the optimization of energy (39) w.r.t. the parameters θ_i is achieved by maximizing their conditional probability :

$$\begin{aligned} \hat{\theta}_{in} &= \arg \max_{\theta_{in}} \prod_{\mathbf{x} \in \Omega_{in}} p(\theta_{in}|I(\mathbf{x})) \\ \hat{\theta}_{out} &= \arg \max_{\theta_{out}} \prod_{\mathbf{x} \in \Omega_{out}} p(\theta_{out}|I(\mathbf{x})) \end{aligned} \quad (40)$$

For example, if θ_i models a Gaussian distribution, $\hat{\theta}_i$ corresponds to the mean and the variance of the intensities inside the region Ω_i .

Zhu and Yuille [82] show that the minimization of the energy (39) w.r.t. C leads to the following motion equation for the contour :

$$\frac{d\mathbf{x}}{dt}(s) = -\mu\kappa(s) \vec{n}(s) + \log \left(\frac{p(I(\mathbf{x}(s))|\theta_{in})}{p(I(\mathbf{x}(s))|\theta_{out})} \right) \vec{n}(s), \quad (41)$$

with $s \in [0, 1)$, $\vec{n}(s)$ being the normal outward vector to the curve at point $\mathbf{x}(s)$ and $\kappa(s)$ the curvature at point $\mathbf{x}(s)$. The first term tends to minimize the contour's curvature which ensures that the contour remains smooth. The second term tends to maximize the likelihood of the inner and outer region w.r.t their parameters θ_{in} and θ_{out} . For instance, if it is more likely that a given intensity $I(\mathbf{x}(s))$ belongs to the region inside the contour, we have :

$$p(I(\mathbf{x}(s))|\theta_{in}) > p(I(\mathbf{x}(s))|\theta_{out}) \Rightarrow \log \left(\frac{p(I(\mathbf{x}(s))|\theta_{in})}{p(I(\mathbf{x}(s))|\theta_{out})} \right) > 0. \quad (42)$$

Therefore, the contour will move outward.

The interesting aspect of this formulation is that the data term in Equation (39) is defined as long as $\theta_{in} \neq \theta_{out}$. Therefore, the contour can evolve even if it is initialized in homogeneous regions.

5.2.1.3 Unifying Edge and Region Energy

The main difficulty to unify these two formulations is that the Edge-Based energy is defined on the contour C whereas the Region-Based energy is defined in terms of regions. In [27], the authors propose to unify these

two approaches by expressing these energies on the inner surface S of the contour C :

$$\begin{aligned} E_{data}(C) &= \int_S T_{unified}(s) ds, \\ &= \int_S \alpha T_{edge}(s) + (1 - \alpha) T_{region}(s) ds, \end{aligned} \quad (43)$$

where $T_{unified}$ is the unified data term relating the Edge-Based and the Region-Based energy terms, T_{edge} and T_{region} are the new corresponding Edge-Based and the Region-Based terms, α is a parameter setting the relative importance between the region and the edge energy and S is the surface inside the contour C .

The region energy is defined as :

$$\begin{aligned} E_{region}(C) &= \int_S -\log \frac{p_{in}(I(\mathbf{x}(s)))}{p_{out}(I(\mathbf{x}(s)))} ds, \\ &= - \int_S T_{region}(s) ds, \end{aligned} \quad (44)$$

with p_{in} and p_{out} are the histograms inside and outside the region S .

The edge energy E_{edge} defined as :

$$E_{edge}(C) = \oint_C \vec{n} \cdot (\vec{e}_I(r)) dr, \quad (45)$$

where \vec{n} is the inward unit normal vector to the curve at point r and \vec{e}_I is a vector field highlighting the edges in the image (for instance ΔI). The authors [27] transfer the edge energy expressed on the curve into an energy expressed on the surface S using Green's theorem [34] :

$$\begin{aligned} E_{edge}(C) &= \int_S \nabla \cdot \vec{e}_I(s) ds, \\ &= \int_S T_{edge}(s) ds, \end{aligned} \quad (46)$$

where $T_{edge} = \nabla \cdot \vec{e}_I$ denotes the divergence of the vector field \vec{e}_I .

5.2.2 B-Snake Implementation

The energy formulations described in the previous section are defined on an explicit continuous contour. However, those expressions must be discretized for implementation. To that aim, we have implemented the algorithm proposed in [27]. In this article, the contour is represented using a set of control points and B-spline functions are used to interpolate the contour between the control points. Therefore, there is an explicit relation between the continuous curve (generated using the B-spline) and the discrete representation of the curve (the control points). The key idea of the algorithm is to transfer the evolution equations presented in Sections 5.2.1 onto the set of control points.

5.2.2.1 Curve Generation Using B-splines

We first need to come back to the definition of a deformable model given in Section 5.1. In this framework, the contour is defined using a parameter vector $\Theta = \{\mathbf{c}_k, k = 1 \dots M\}$ where $\mathbf{c}_k \in \mathbb{R}$ are the control points. The curve C is generated using :

$$C(t) = \sum_{k=-\infty}^{\infty} \psi(t-k) [\mathbf{c}_k \quad \mathbf{c}_{k+1} \quad \mathbf{c}_{k+2} \quad \mathbf{c}_{k+3}] \quad (47)$$

where ψ is a uniform cubic B-spline defined for $t \in \mathbb{R}$:

$$\psi(t) = \frac{1}{6} \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}. \quad (48)$$

This formulation allows obtaining the location of any point on the curve with parameter $t \in \mathbb{R}$, namely, formalizing the *Shape Description* of the contour. Figure 27 shows the representation of a circle using the above formulation for a different number of control points.

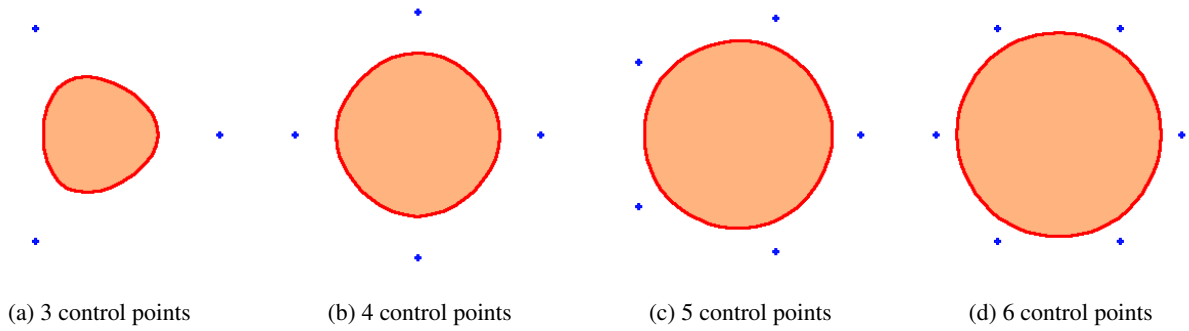


Figure 27: Representation of a circle using a different number of control points.

5.2.2.2 Energy Formulation

Like done in the previous sections, the objective of the authors is to express the objective energy using the formulation :

$$E(C) = E_{int}(C) + E_{data}(C). \quad (49)$$

For the internal energy E_{int} , the authors [27] propose to force the control points to be regularly spaced :

$$E_{int}(C) = - \int_0^M \left\| \|\mathbf{x}'(s)\|^2 - c^2 \right\| ds, \quad (50)$$

where $\mathbf{x}'(s)$ is the tangential vector to the curve at point s and c is the average spacing between the control points. This energy is minimum when all the control points are equally spaced.

The authors use the unified data term we introduced in Section 5.2.1.3 :

$$\begin{aligned} E_{data}(C) &= \int_S T_{unified}(s) ds, \\ &= \int_S \alpha T_{edge}(s) + (1 - \alpha) T_{region}(s) ds, \end{aligned} \quad (51)$$

where α is a parameter setting the relative importance between the region and the edge energy and S is the surface inside the contour C .

5.2.2.3 Partial Derivatives and Evolution of the Contour

The evolution of the contour C is obtained by the gradient descent algorithm with the partial derivatives of the energy in Equation (49) computed on the set of control points :

$$\begin{pmatrix} c_{x,k}^{p+1} \\ c_{y,k}^{p+1} \end{pmatrix} = \begin{pmatrix} c_{x,k}^p \\ c_{y,k}^p \end{pmatrix} - \begin{pmatrix} \frac{\partial}{\partial c_{x,k}^p} E_{data} \\ \frac{\partial}{\partial c_{y,k}^p} E_{data} \end{pmatrix} - \begin{pmatrix} \frac{\partial}{\partial c_{x,k}^p} E_{int} \\ \frac{\partial}{\partial c_{y,k}^p} E_{int} \end{pmatrix} \quad (52)$$

Where the upper index refers to an artificial time parameter p , the first lower index is either x or y referencing the first or second coordinate of the control point, the second lower index refers to the index of the control point in $\Theta = \{\mathbf{c}_k, k = 1 \dots M\}$ and E_{int} , E_{data} are defined in Equations (50) and (51).

We let the reader refer to the original article [27] for calculus details, but the partial derivatives of E_{data} can be expressed a follow :

$$\begin{bmatrix} \frac{\partial}{\partial c_{x,k}^p} E_{data} \\ \frac{\partial}{\partial c_{y,k}^p} E_{data} \end{bmatrix} = \sum_{l=0}^{M-1} \begin{bmatrix} -c_{y,l}^p \\ c_{x,l}^p \end{bmatrix} Q_{T_{unified}}(k, l) \quad (53)$$

where :

$$Q_{T_{unified}}(k, l) = \int_{-\infty}^{\infty} T_{unified}(I)(t+k)\psi(t)\psi'(t+k-l)dt. \quad (54)$$

This term can be computed in an efficient way because the functions ψ and ψ' have a finite support, so the integral is null in a large portion of the integration domain. The term $Q_{T_{unified}}$ can be seen as the influence of the curve on the closer control points using the unified data term to guide the evolution.

On the other hand, the partial derivatives of E_{int} can be expressed a follow :

$$\begin{aligned} \frac{\partial}{\partial c_{x,k}^p} E_{int} &= \sum_{|l|,|m|,|n|<N} c_{x,k-l}^p c_{x,k-m}^p c_{x,k-n}^p h_1(l, m, n) \\ &+ \sum_{|l|,|m|,|n|<N} c_{x,k-l}^p c_{y,k-m}^p c_{y,k-n}^p h_1(l, m, n) \\ &- 4c \sum_{|l|<N} c_{x,k-l}^p h_2(l) \end{aligned} \quad (55)$$

$$\begin{aligned} \frac{\partial}{\partial c_{y,k}^p} E_{int} &= \sum_{|l|,|m|,|n|<N} c_{y,k-l}^p c_{y,k-m}^p c_{y,k-n}^p h_1(l, m, n) \\ &+ \sum_{|l|,|m|,|n|<N} c_{y,k-l}^p c_{x,k-m}^p c_{x,k-n}^p h_1(l, m, n) \\ &- 4c \sum_{|l|<N} c_{y,k-l}^p h_2(l) \end{aligned} \quad (56)$$

where :

$$\begin{aligned} h_1(l, m, n) &= \int_{-\infty}^{\infty} \psi'(t)\psi'(t+l)\psi'(t+m)\psi'(t+n)dt \\ h_2(l) &= \int_{-\infty}^{\infty} \psi'(t)\psi'(t+l)dt. \end{aligned} \quad (57)$$

5.2.2.4 Iris Segmentation Using B-Snakes

There are two main elements to define in order to use the B-Snakes algorithm for segmentation : the number of control points used to generate the contour and the balance between the edge and the region information in Equation (43).

The number of control points is related to the degrees of freedom on the contour. The more there are control points, the less the shape of the contour is constrained. At the opposite, the less there are control points, the more the shape of the contour is restricted. Setting a high number of control points could seem interesting, but giving too much freedom to the contour that can lead to incorrect segmentation results. Indeed, the contour may be attracted by some artifacts in addition to the iris region. From our experiments, we found out that 10 control points seem to set a fair balance between constraint and freedom of the contour shape.

The edge term and the region term defined in Equation (43) have different properties. The edge term E_{edge} is only defined in the proximity of the edges in the image, but can lead the contour to a very accurate location of the object borders. At the opposite, the region term E_{region} is always defined, even far from the actual borders of the object. However, the location of borders is usually less accurate using the region information instead of gradient information. Accordingly, we propose to split the evolution of the contour into two stages. First, the contour evolves mostly using the region energy E_{region} and, once the contour is close to the edges of the object, the contour evolves mostly using the edge energy E_{edge} . Such a behavior can be achieved by setting the α parameter in Equation (43) close to 0 at the beginning of the process and close to 1 at the end.

Figure 28 illustrates the evolution on an iris image using the B-Snake algorithm with 10 control points. As we have explained above, we divide the evolution into two phases : we first give a strong weight to the region forces in order to guide the contour from the initial location to the approximate border of the iris. Then we give more strength to the edge forces ensuring that the contour fits precisely the borders of the iris. It is interesting to note that because of the limited number of control points, the contour has a quite constrained shape and can not catch small details such as eyelashes or shadows.

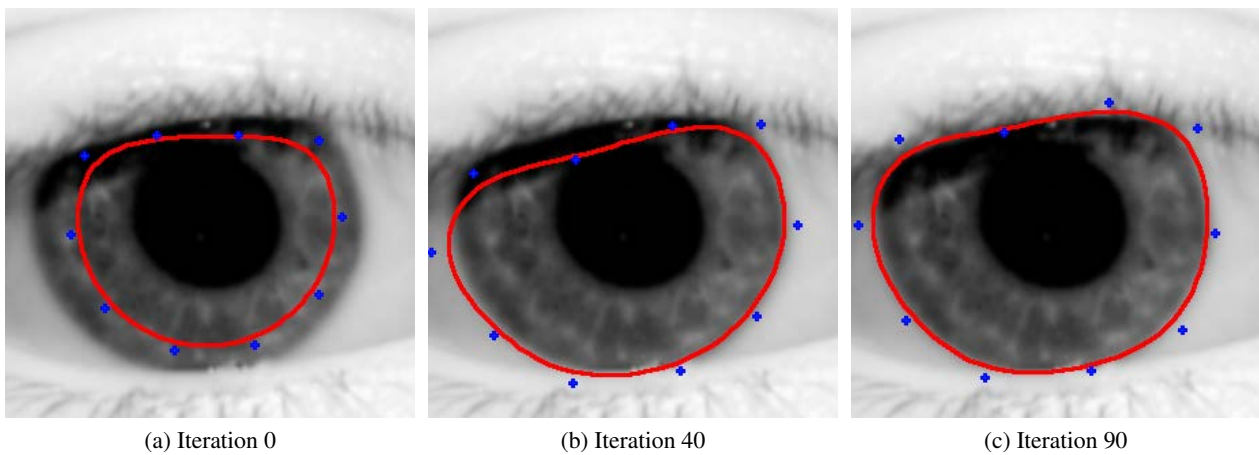


Figure 28: Segmentation of the iris using B-Snakes. The red curve represents the contour and the blue points are the corresponding control points.

5.3 Geometric Contours

In this section, we present some classical *Geometric Contours*. They differ from *Parametric Contours* described in Section 5.2 in terms of *Shape Description*. In case of *Geometric Contours*, the contour is generated using the Level-Set framework described in Section 5.1.2. In this framework, the curve is defined as the intersection of a 3D manifold and a plane. The evolution of the curve is done by deforming the manifold.

In next Section 5.3.1, we describe Geodesic Active Contours that use an Edge-Based Data Energy for *Geometric Contours*. Then, in Section 5.3.2 we describe the Chan and Vese model that extends the Region Competition principle we described in Section 5.2.1.2 to *Geometric Contours*.

5.3.1 Geodesic Active Contours

Geodesic Active Contours (GAC) were introduced by Casselles et al. [7] in 1997. We study GAC because they are the first *Geometric Contours*. Moreover, they have already been successfully used for iris segmentation in [66].

5.3.1.1 General Formulation

Geodesic Active Contours aim at minimizing the functional :

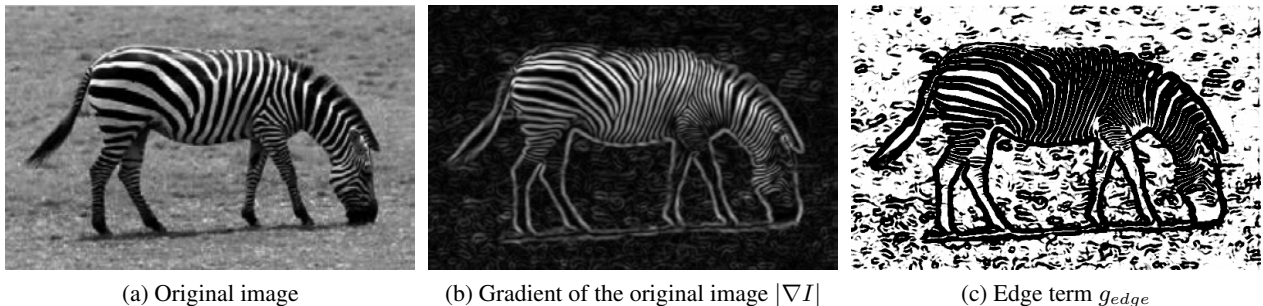
$$E_{GAC}(C, I) = \int_0^1 g_{edge}(I(C(q))) |C'(q)| dq. \quad (58)$$

where C is a closed differentiable contour in \mathbb{R}^2 , $C : [0; 1) \rightarrow \mathbb{R}^2$, I is the input image and g_{edge} is a term designed to respond on image edges. As explained previously, this functional is minimized by iteratively evolving the contour C until it reaches a minimum of g_{edge} . Accordingly g_{edge} is called the stopping term of the functional because it indicates where the contour should stop in the image. A physical interpretation of Equation (59) is that g_{edge} is a potential energy, and the global energy of the contour is minimal when it reaches areas of low potential. Therefore, one of the key ideas of this method is to build the term g_{edge} so that its minima represent the boundaries of the object to segment.

A classical formulation for g_{edge} is :

$$g_{edge}(I(\mathbf{x})) = \frac{1}{1 + |\nabla I(\mathbf{x})|}, \quad (59)$$

where ∇I is the image gradient. Using this definition of g_{edge} implies that $E_{GAC}(C, I)$ is minimal when the contour C passes through areas of strong gradient in I . Let us note here that other features can be exploited instead of edges, as we will see in Section 5.4. Figure 29 shows an example of the energy term g_{edge} for a natural image.



(a) Original image

(b) Gradient of the original image $|\nabla I|$ (c) Edge term g_{edge}

Figure 29: Exemple of edge term g_{edge} for a natural image

As explained in Section 5.1, the minimization of E_{GAC} is performed using a gradient descent algorithm, i.e., for an artificial time parameter t , the evolution of the contour is given by (see[7] for calculus details) :

$$\begin{aligned} \frac{\partial C}{\partial t} &= -\frac{\partial E_{GAC}}{\partial t} \\ &= \left[g_{edge}(I)\kappa - \langle \nabla g_{edge}, \vec{n} \rangle \right] \vec{n}, \end{aligned} \quad (60)$$

where \vec{n} the outward normal vector of the contour and κ the contour's curvature.

Equation (60) comes from the derivation of a single energy term in Equation (58). However, the two terms of Equation (60) can be interpreted in a similar fashion to the reference Equation (26) given for *Deformable models*. The first term of Equation (60) tends to minimize the curvature along the curve, i.e. it ensures that the curve remains smooth similarly to the internal term $\frac{\partial E_{int}}{\partial t}$. The second term indicates that the contour is attracted by the minimums of the function $\nabla g_{edge}(I)$ that can be seen as the influence of the data term $\frac{\partial E_{data}}{\partial t}$.

This evolution could be performed on the contour like the *Parametric Contours* described in Section 5.2. However, Casselles et al. noted that this minimization can be handled very efficiently by the Level-Set framework [1] described in Section 5.1.2. Indeed, in this framework the contour is represented as the 0 level set of a 2-dimensional manifold embedded in a 3-dimensional space. Formally, for a function ϕ of \mathbb{R}^3 , C is defined as $C = \{\mathbf{x} \in \Omega | \phi(\mathbf{x}) = 0\}$. Most geometric properties of C can be expressed using only the ϕ function. For instance :

$$\begin{aligned} \vec{n} &= -\frac{\nabla \phi}{\|\nabla \phi\|}, \\ \kappa &= \text{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right). \end{aligned} \quad (61)$$

We can rewrite Equation (60) :

$$\frac{\partial C}{\partial t} = -\frac{\partial E(C)}{\partial C} = F \cdot \vec{n}, \quad (62)$$

with F the speed function of the model. As we have $\phi(C(t), t) = 0$ for all t we can write :

$$\frac{d}{dt}\phi(C(t), t) = \nabla \phi \frac{\partial C}{\partial t} + \frac{\partial \phi}{\partial t} = \nabla \phi \cdot \vec{n} + \frac{\partial \phi}{\partial t}. \quad (63)$$

Using the formula $\vec{n} = -\frac{\nabla \phi}{\|\nabla \phi\|}$ we can transfer the evolution equation from the curve C to ϕ :

$$\frac{\partial \phi}{\partial t} = \|\nabla \phi\| F. \quad (64)$$

Therefore, the minimization flow of equation (60) expressed on C can be expressed on ϕ instead :

$$\frac{\partial \phi}{\partial t} = \left[g_{edge}(I) \text{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) + \langle \nabla g_{edge}(I), \frac{\nabla \phi}{\|\nabla \phi\|} \rangle \right] \|\nabla \phi\|, \quad (65)$$

leading to the evolution equation of the ϕ function :

$$\begin{aligned} \phi^{t+\Delta t} &= \phi^t - \Delta t \frac{\partial E(C)}{\partial C} \\ &= \phi^t + \Delta t \frac{\partial \phi}{\partial t} \end{aligned} \quad (66)$$

This equation can be efficiently implemented using the narrow band algorithm [1]. This algorithm proposes to compute the function ϕ only in the neighborhood of its zero-level (the curve C) to speed up the process. However, the function ϕ needs to be re-evaluated regularly in order to ensure the stability of the evolution.

5.3.1.2 Adapting GAC to Iris Segmentation

The GAC framework is successfully used for iris segmentation by Shah et. al in [66]. They propose to use an evolution of the traditional GAC formulation and adapt it to the iris segmentation task. First, they define an alternative stopping term $g_{edge'}$ defined as :

$$g_{edge'}(I(\mathbf{x})) = \frac{1}{1 + \left(\frac{\|\nabla[G*I](\mathbf{x})\|}{k} \right)^\alpha}, \quad (67)$$

where G is a Gaussian kernel and k and α are parameters enhancing the edges of the image. Changing these parameters acts on the sharpness on the energy valleys in $g_{edge'}(I)$. The authors also remove the pupil area from $g_{edge'}(I)$ in order to ensure that the contour does not get stuck in pupil edges.

They also propose to add a balloon force [11] to the contour forcing the contour to evolve in one direction. This leads to the evolution equation :

$$\frac{\partial \phi}{\partial t} = g_{edge'}(I) |\nabla \phi| \left(c + \epsilon \operatorname{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) \right) + \langle \nabla g_{edge'}(I), \nabla \phi \rangle, \quad (68)$$

where the parameter c sets the strength of the evolution and can be seen as a balloon force pushing the contour outward and ϵ the regularity of the contour based on its curvature.

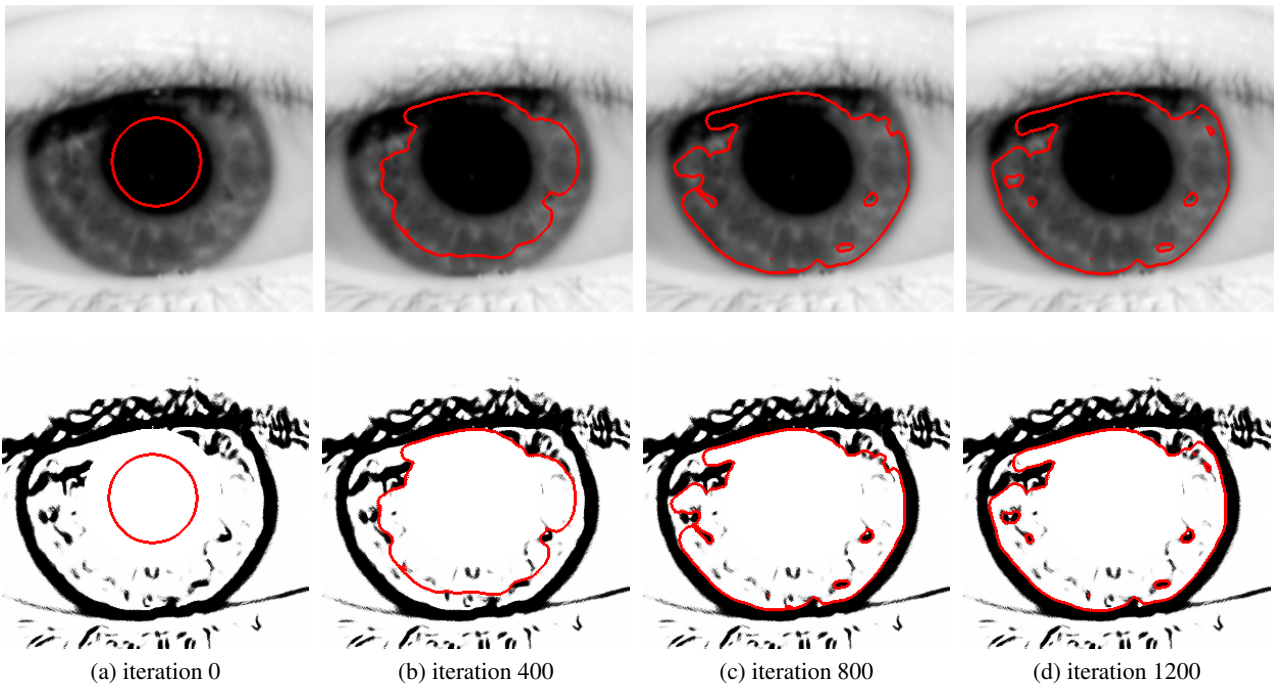


Figure 30: Iris segmentation using GAC. The first row shows the evolution on the original image. The second row shows the evolution on the $g_{edge'}(I)$.

In order to segment the iris, we set the same parameters as proposed in [66] : $k = 2.8$, $\alpha = 8$, $c = 0.65$ and $\epsilon = 1$. Using these settings tends to produce an edge term $g_{edge'}(I)$ where the transitions between homogenous regions and high frequency regions are very sharp. Moreover, the balloon force is regulated by c and is quite high compared to the other elements. Therefore, the contour is expected to expand mostly due to the balloon force, until it reaches low energies regions of $g_{edge'}(I)$ where it should stop.

Figure 30 shows an example of GAC evolution for iris segmentation. The first row shows the evolution on the original image and the lower one on the edge term $g_{edge'}$. Like Shah et. al [66], we mask the pupil area in

$g_{edge'}$. The contour is initialized inside the pupil and first expands mainly thanks to the balloon force c . Around iteration 400, the contour reaches textured areas of the iris (dark areas in $g_{edge'}$). The contour slows down in these areas, and even stops in highly textured areas. Between iterations 400 and 800 the contour crosses the textured areas thanks to the regularization and balloon forces. Finally, between iterations 800 and 1200, the contour progressively fits the iris border until it reaches an equilibrium between the forces.

The main problem with equation (68) is the setting of parameter c (as highlighted by the authors [66]). On one hand, if this term is not set strong enough, the contour stops before reaching the edges of the iris. On the other hand, if it is set too high, the contour may cross the borders of the iris. Moreover, this force assumes that the contour has been initialized either inside or outside the object to segment, but the overall evolution is one sided (if $c > 0$ the contour only moves outward, if $c < 0$ the contour only moves inward). Figure 31 illustrates and over-segmentation due to the balloon force.

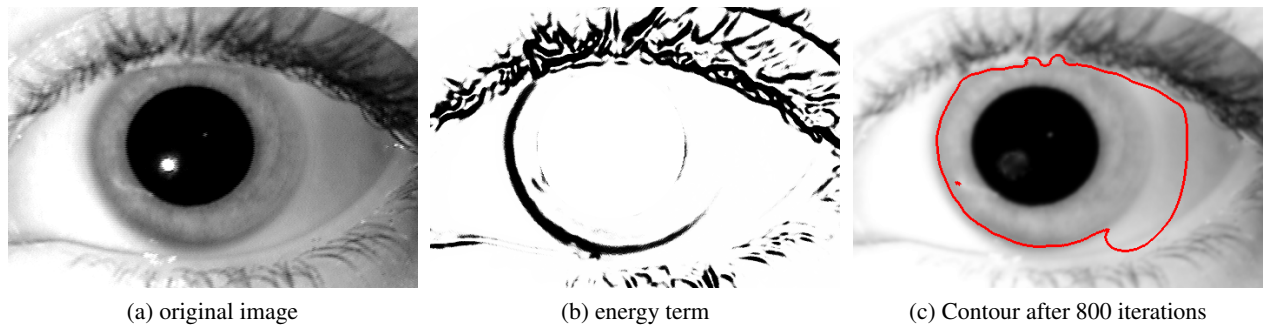


Figure 31: Example of a segmentation failure due to the balloon force c . The iris edges are very weak on the right side and almost non-existent in the energy term $g_{edge'}$. Accordingly, the contour crosses the edge due to the balloon force.

5.3.2 Chan And Vese Formulation

The Chan and Vese formulation is an extension of the GAC using the Region Competition approach introduced by Zhu and Yuille [82] we already described in Section 5.2.1.2 for *Parametric Contours*.

5.3.2.1 General Formulation

In the following, we note R_{in} the region of the image inside the contour C and R_{out} the region of the image outside the contour C . Chan and Vese proposed to minimize the energy E_{CV} :

$$E_{CV}(C, c_1, c_2) = E_{int}(C) + E_{data}(C, c_1, c_2),$$

$$\text{with } \begin{cases} E_{int}(C) &= \mu \text{Length}(C) + \nu \text{Area}(\text{inside}(C)), \\ E_{data}(C, c_1, c_2) &= \lambda_1 \int_{R_{in}} (I(\mathbf{x}) - c_1)^2 d\mathbf{x} + \lambda_2 \int_{R_{out}} (I(\mathbf{x}) - c_2)^2 d\mathbf{x}, \end{cases} \quad (69)$$

where C is the contour, I the considered image, c_1 and c_2 respectively the expected means of the pixels' intensities inside and outside C and μ , ν , λ_1 and λ_2 are parameters. Values of c_1 and c_2 can be either defined based on a prior knowledge about the object to segment, or recomputed during the evolution of the contour as the mean of the pixels' intensities inside and outside C if no assumption is made about such objects' intensity.

$E_{int}(C)$ tends to minimize the length and the area of the contour, which forces the contour to remain smooth and keep limited area. $E_{data}(C, c_1, c_2)$ is low when the pixels inside C have a mean close to c_1 and the pixels outside C have a mean close to c_2 . This formulation is a simplification of the Mumford and Shah functional [51]. In [51], Mumford and Shah set the segmentation problem as the minimization of a

given functional in a very formal and mathematical way. However, the resolution of their functional is known to be extremely slow which usually requires to use approximations. The Chan and Vese formulation is a simplification (but exact) resolution of the Mumford and Shah functional assuming a 2 regions problem and that the solution of the segmentation is a piecewise constant image (regions are only described by their means).

Chan and Vese show that this formulation is well suited for a minimization using the Level-Set framework described in Section 5.1.2. The idea is to express the functional E_{CV} of equation (69) using the Heaviside function H :

$$\begin{cases} H(x) = 1 & \text{if } x \geq 0 \\ = 0 & \text{otherwise} \end{cases} \quad (70)$$

the Dirac function δ :

$$\begin{cases} \delta(x) = 1 & \text{if } x = 0 \\ = 0 & \text{otherwise} \end{cases} \quad (71)$$

and the function ϕ defined as:

$$\begin{cases} C = \partial\omega = \{(x, y) \in \Omega : \phi(x, y) = 0\}, \\ R_{in} = \omega = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ R_{out} = \Omega \setminus \bar{\omega} = \{(x, y) \in \Omega : \phi(x, y) < 0\}, \end{cases} \quad (72)$$

where Ω is the image domain and $\omega \in \Omega$ is an open set whose border is the current curve $C = \partial\omega$. The functional E_{CV} can be written using ϕ instead of C :

$$\begin{aligned} E_{CV}(\phi, c_1, c_2) = & \mu \int_{\Omega} \delta(\phi(\mathbf{x})) \|\phi(\mathbf{x})\| d\mathbf{x} + \nu \int_{\Omega} H(\phi(\mathbf{x})) d\mathbf{x} \\ & + \lambda_1 \int_{\Omega} (I(\mathbf{x}) - c_1)^2 H(\phi(\mathbf{x})) d\mathbf{x} + \lambda_2 \int_{\Omega} (I(\mathbf{x}) - c_2)^2 (1 - H(\phi(\mathbf{x}))) d\mathbf{x}. \end{aligned} \quad (73)$$

The main interest of this formulation compared to the one expressed on C in equation (69) is that all the integrals are computed on the whole image domain Ω . This statement allows an efficient use of calculus to minimize E_{CV} w.r.t. ϕ , c_1 and c_2 .

First, fixing ϕ and minimizing E_{CV} w.r.t. c_1 and c_2 leads to :

$$\begin{aligned} c_1(\phi) &= \frac{\int_{\Omega} I(\mathbf{x}) H(\phi(\mathbf{x})) d\mathbf{x}}{\int_{\Omega} H(\phi(\mathbf{x})) d\mathbf{x}} = \text{average}(I) \text{ in } \{\phi \geq 0\} \\ c_2(\phi) &= \frac{\int_{\Omega} I(\mathbf{x}) (1 - H(\phi(\mathbf{x}))) d\mathbf{x}}{\int_{\Omega} 1 - H(\phi(\mathbf{x})) d\mathbf{x}} = \text{average}(I) \text{ in } \{\phi < 0\} \end{aligned} \quad (74)$$

Then, by fixing c_1 and c_2 , the minimization of E_{CV} w.r.t. ϕ is done by gradient descent. For an artificial time parameter t , we have the minimization flow :

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left(\mu \operatorname{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) - \nu - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2 \right), \quad (75)$$

where δ_{ϵ} is a continuous approximation of the Dirac function such that $\delta_{\epsilon} \rightarrow \delta$ for $\epsilon \rightarrow 0$.

Like done in the previous section, the function ϕ evolves using the scheme :

$$\phi^{t+\Delta t} = \phi^t + \Delta t \frac{\partial \phi}{\partial t}. \quad (76)$$

| GAC | Chan & Vese | Meaning |
|--|---|---------------------|
| $\epsilon g_{edge'}(I) \nabla\phi \operatorname{div} \left(\frac{\nabla\phi}{\ \nabla\phi\ } \right)$ | $\mu \delta_\epsilon(\phi) \operatorname{div} \left(\frac{\nabla\phi}{\ \nabla\phi\ } \right)$ | mean curvature flow |
| $c g_{edge'}(I) \nabla\phi $ | $\nu \delta_\epsilon(\phi)$ | balloon force |
| $\langle \nabla g_{edge}(I) \cdot \nabla\phi \rangle$ | $\delta_\epsilon(\phi) (-\lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2)$ | data term |

Table 4: Interpreting the mathematical terms in GAC versus in the Chan and Vese formulation

It is interesting to compare this equation with the traditional evolution of GAC in Equation (68) :

The mean curvature flow and the balloon force have the same influence on the contour evolution in both formulations, but the data term is very different. The term $\langle \nabla g_{edge}(I) \cdot \nabla\phi \rangle$ from the GAC formulation only influences the evolution of the contours in areas where $\nabla g_{edge}(I) \neq 0$. On the other hand, for the Chan and Vese model, the term $\delta_\epsilon(\phi) (-\lambda_1(I - c_1)^2 + \lambda_2(I - c_2)^2)$ influences the evolution of the whole contour as long as $c_1 \neq c_2$. This means that the data term of the GAC has a local definition, whereas the data term of the Chan and Vese model has a global definition.

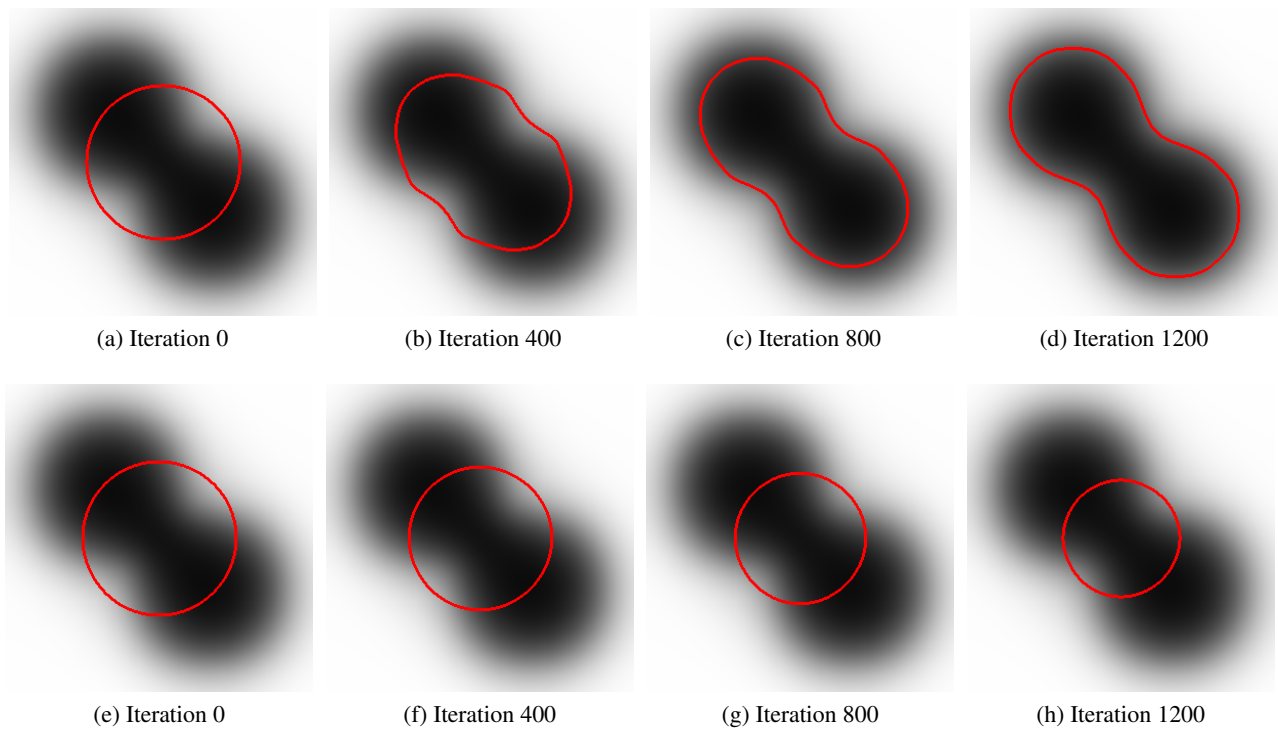


Figure 32: Segmentation of a synthetic image using the Chan and Vese formulation for the first row and the Geodesic Active Contours in the second row.

Figure 32 shows the segmentation of a synthetic image using the Chan and Vese evolution scheme in Equation (75) compared to the Geodesic Active Contour scheme in Equation (68). The objects' borders are very smooth, so the data term in the GAC formulation is null almost everywhere ($\nabla g_{edge}(I) \approx 0$ in Equation (68)). Therefore, the contour slowly shrinks due to the mean curvature flow (the balloon force has been removed for this experiment). At the opposite, the contour in the framework of the Chan and Vese formulation catches the intensity properties of the object ($c_1 \neq c_2$ in Equation (75)) and converges to the expected borders of the object.

5.3.2.2 Iris Segmentation

The classical formulation of the Chan and Vese contour has already been used for iris segmentation [76][62], but from our point of view using directly the Chan and Vese model does not seem suitable for iris segmentation. Indeed, as discussed above, the Chan and Vese model assumes that the solution of the problem is a piecewise constant image. Therefore the object to segment and the background should be distinguishable by their mean intensity value. Although, even if the iris has a quite stable intensity, no assumptions can be made on "background intensities" (namely, everything that is not the iris texture). In an iris image, it often happens that the background contains very dark elements such as eyelashes, and very bright ones as well, such as the skin or the limbus. In this situation, the iris and the background have very close means and the use of the above formulation will not lead to a correct segmentation.

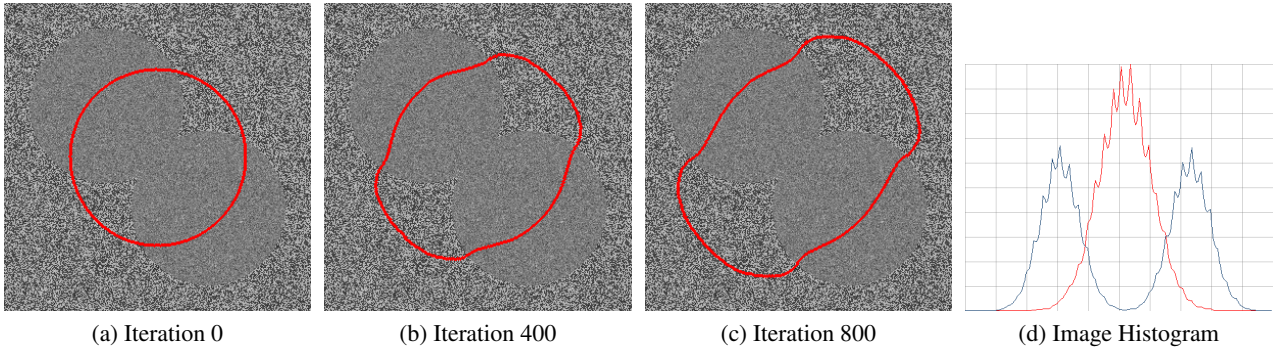


Figure 33: Segmentation of a synthetic image using Chan and Vese traditional formulation. Figure 33d shows the intensity distributions of the regions (red for the object, blue for the background)

Figure 33 illustrates this problem with a synthetic image. The object and the background have different intensity distributions but the same means. The object's intensities follow a Normal distribution of mean 124 and the background intensities follow a mixture of two Normal distributions of mean 74 and 174. As the object and the background have the same mean, the traditional Chan and Vese model is not able to distinguish between both.

In order to solve this problem, we propose to model the image regions by their intensity distributions instead of their means. Formally, we now aim at minimizing :

$$\begin{aligned}
 E_{CV'}(\phi, I, p_{in}, p_{out}) &= \mu \int_{\Omega} \delta(\phi(\mathbf{x})) \|\phi(\mathbf{x})\| d\mathbf{x} + \nu \int_{\Omega} H(\phi(\mathbf{x})) d\mathbf{x} \\
 &\quad + \lambda_1 \int_{\Omega} -\log(p_{in}(I(\mathbf{x}))) H(\phi(\mathbf{x})) d\mathbf{x} + \lambda_2 \int_{\Omega} -\log(p_{out}(I(\mathbf{x}))) (1 - H(\phi(\mathbf{x}))) d\mathbf{x}. \\
 \text{subject to} &\quad \int_{\mathbb{R}} p_i(a) da = 1, \quad i = \{1, 2\}
 \end{aligned} \tag{77}$$

It can be shown (see [50]) that the optimization of $E_{CV'}$ w.r.t. p_{in} and p_{out} while fixing ϕ , leads to a continuous version of the Parzen window estimate of the densities inside and outside the contour C . For $a \in \mathbb{R}$:

$$\begin{aligned}
 p_{in}(a) &= \frac{\int_{R_{in}} K_{\epsilon}(I(\mathbf{x}) - a) d\mathbf{x}}{\int_{R_{in}} d\mathbf{x}}, \\
 p_{out}(a) &= \frac{\int_{R_{out}} K_{\epsilon}(I(\mathbf{x}) - a) d\mathbf{x}}{\int_{R_{out}} d\mathbf{x}},
 \end{aligned} \tag{78}$$

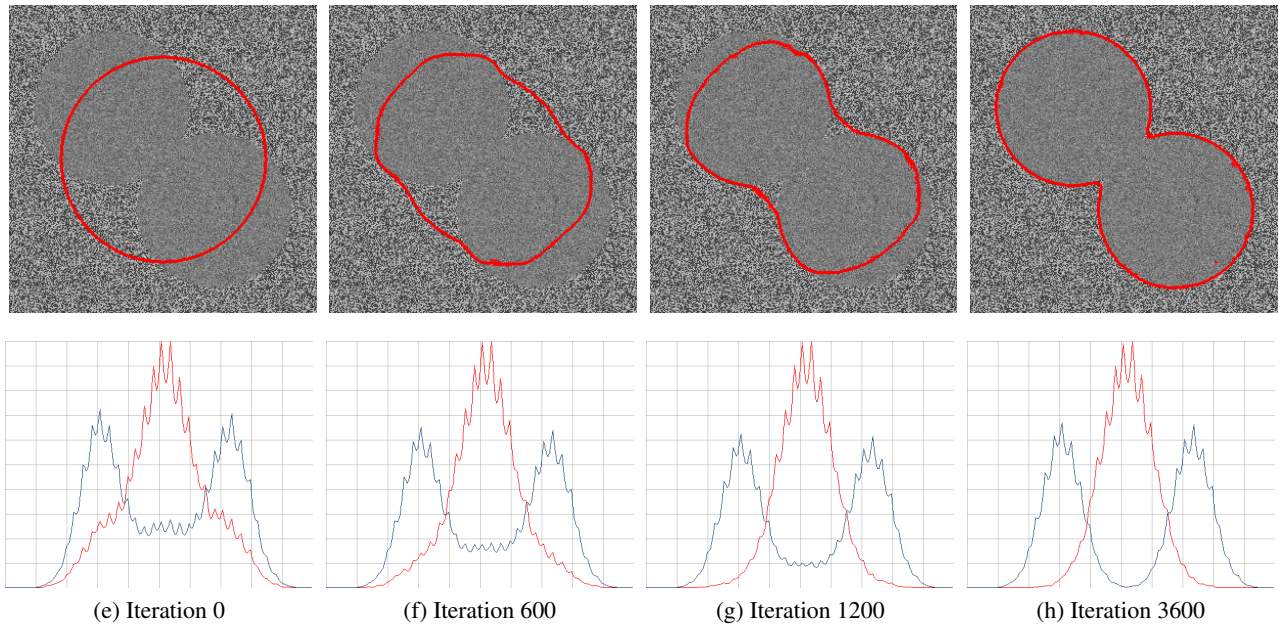


Figure 34: The convergence of the contour minimizing the energy $E_{CV'}$ in the image of Figure 33. The first row shows the evolution of the contour and the second shows the intensity histograms inside (red) and outside (blue) the contour (excluding the pupil area from the computation).

where K_ϵ is a Gaussian Kernel of small variance ϵ . The optimization of $E_{CV'}$ w.r.t. ϕ also uses a gradient descent approach leading to:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left(\mu \operatorname{div} \left(\frac{\nabla \phi}{\|\nabla \phi\|} \right) - \nu - \lambda_1 \log(p_{in}(I)) + \lambda_2 \log(p_{out}(I)) \right), \quad (79)$$

Figure 34 shows an illustration of the minimization scheme in Equation (79), considering only the influence of the data term from the Chan and Vese model for the segmentation of the synthetic image from Figure 33.

As discussed above, one of the main interest of the Chan and Vese Model is that the data energy is defined globally as opposed to the data term of the GAC model that is defined locally. Therefore the balloon term in Equation (79) can be set to 0. Thanks to the region data energy, the contour evolves even if the initialization is done far from the expected final location of the contour. However, in order to speed up the convergence, we do not set the balloon force to 0, but to a low value compared to the other terms.

Figure 35 illustrates the segmentation of an iris image using the Chan and Vese Model. The contour is initialized around the pupil region and progressively evolves until it segments the iris region. It is interesting to note that as the balloon force has little influence on the contour evolution, the contour can shrink in some parts and expand in some others simultaneously. Indeed, the upper part (initialized inside some eyelashes and shadows) shrinks while the lower part (initialized inside the iris region) expands. Such a behavior could not have been achieved with a strong balloon force such as the one used for GAC.

The Chan and Vese Model and the GAC presented in the previous Section are based on different *Deformation Descriptions*, but they tend to fail on the same kind of images : images in which the iris texture has intensities very close to its surrounding elements, like the limbus for instance. For such images, the gradient between the iris region and the limbus region is very weak, which leads to segmentation failures as we already discussed in Section 5.3.1. Similarly, as there is no clear difference between the intensities of the iris and the limbus ones, the Chan and Vese Model is not able to distinguish the two regions based on their intensities, which leads to segmentation failures. Therefore, in the next section, we propose to describe the iris region using texture instead of intensities in order to avoid such failures.

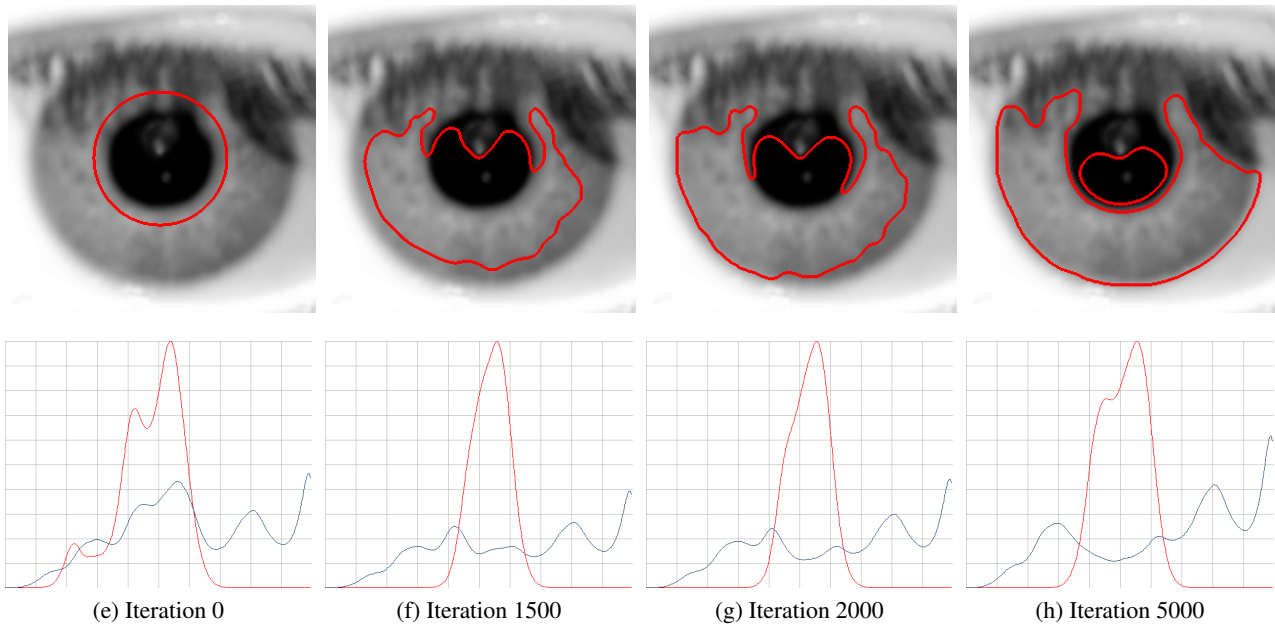


Figure 35: The convergence of the contour minimizing $E_{CV'}$ energy. The upper row presents the evolution the contour and the lower presents the separation of the inner (red) and outer (blue) histograms.

5.4 Texture Based Active Contours

As explained at the end of Section 5.3, representing the iris region using only the intensity information can lead to segmentation failures. Therefore, we propose to use this textural information to segment the iris region instead of the intensity information. To do so, we have implemented some ideas described in [64]. In this paper, Chen et al. propose a *Geometric Contour* exploiting textural information in Active Contours. This textural information is used to build both a new *Edge-Based* and a new *Region-Based data energy*. The *Edge-Based data energy* is built in order to define a notion of gradient between areas with different textural properties. The *Region-Based data energy* is a simple extension of the Chan and Vese Model for multichannel images.

5.4.1 Texture Descriptors

Our implementation follows [64] using Gabor filters to extract texture information. A filter is constructed according to three parameters :

- the orientation θ , i.e. the direction of a pattern's repetition.
- the radial frequency f , i.e. the frequency at which a sinusoidal pattern repeats itself.
- the scale σ , determines how a pattern spreads itself in the image.

Let $h : \mathbb{R}^2 \rightarrow \mathbb{C}$ be the impulse response of a 2D-Gabor filter such that:

$$\Re(h(x, y)) = g(x_\theta, y_\theta) \cos(2\pi(Ux + Vy)) \quad (80)$$

$$\Im(h(x, y)) = g(x_\theta, y_\theta) \sin(2\pi(Ux + Vy)) \quad (81)$$

where $U = f \cos \theta$ and $V = f \sin \theta$ represent the 2-D spatial frequencies, (x_θ, y_θ) is the image of (x, y) under a rotation of θ radians, g is a centered elliptical Gaussian window function with covariance matrix Σ defined as:

$$\Sigma = \begin{bmatrix} \lambda\sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}. \quad (82)$$

Our purpose is to use such Gabor filters to characterize the texture of a given point in the image. Therefore, we apply a set of Gabor filters and, for each point, we memorize the filter giving the strongest response. We use this filter to build a multichannel image characterizing texture in the original image : $M = \{C_0, C_\theta, C_f, C_\sigma, C_{\Re}, C_{\Im}\}$ where, for a given pixel \mathbf{x} :

- $C_0(\mathbf{x})$ is the magnitude of the filter giving the largest response at \mathbf{x} .
- C_θ is the orientation of the filter giving the largest response.
- C_f is the frequency of the filter giving the largest response.
- C_σ is the scale of the filter giving the largest response.
- C_{\Re} and C_{\Im} the real and imaginary components of the filter's response.

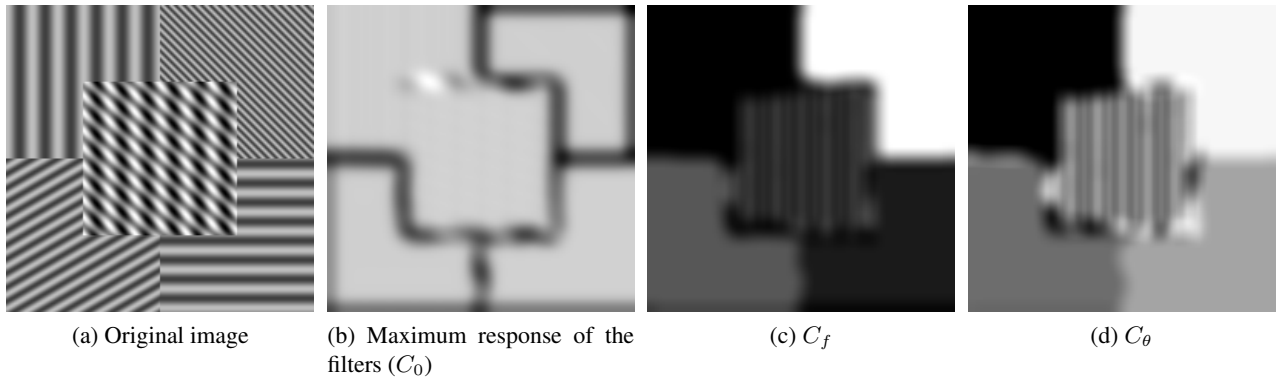


Figure 36: Synthetic image and some channels of the Gabor feature space.

5.4.2 Edge Indicator

Before detailing how the authors in [64] build their texture edge detector, let us make a digression and discuss a geometric interpretation of the GAC energetic functional of equation (59) in Section 5.3.1. A classical gray-scale image can be seen from a strictly geometric point of view as a 2D Riemannian surface (manifold), with (x, y) as local coordinate, embedded in \mathbb{R}^3 with (X, Y, Z) as local coordinates. The relation between (x, y) and (X, Y, Z) is given by $(X = x, Y = y, Z = I(x, y))$. Using this formulation, the minimization of the GAC energetic functional of equation (59) can be seen as finding the contour of minimal length on the 2D Riemannian manifold w.r.t. a certain distance measure. Measuring the distance on this manifold can be done using :

$$d_{GS}(x, y) = \begin{pmatrix} 1 + I_x I_x & I_x I_y \\ I_x I_y & 1 + I_y I_y \end{pmatrix}, \quad (83)$$

where d_{GS} is the distance on the gray-scale manifold, I_x and I_y correspond to the image derivatives w.r.t. x and y . This metric indicates how fast the manifold's shape changes : for instance, an edge in the image can be seen as a cliff on the manifold, therefore $\det(d_{GS})$ will be high along edges. d_{GS} can be seen as an edge indicator, and the inverse of $\det(d_{GS})$ can be seen as a good stopping function for the GAC. Actually, we even have the equivalence :

$$g_{edge}(I(x, y)) = \frac{1}{\det(d_{GS}(x, y))}, \quad (84)$$

where $\det(d_{GS}(x, y))$ is the determinant of the matrix $d_{GS}(x, y)$ and g_{edge} the stopping term of Section 5.3.1.

The key idea of [64] is to extend this geometric interpretation of the GAC using the Gabor features space instead of the image space. Formally, the image is seen as a 2D Riemannian surface embedded in a \mathbb{R}^7 space

with $\{x, y, C_\theta(x, y), C_f(x, y), C_\sigma(x, y), C_{\mathfrak{R}e}(x, y), C_{\mathfrak{I}m}(x, y)\}$. Though less intuitive than the traditional image space, the feature space denoted $(F^1(x, y), \dots, F^n(x, y))$ allows defining the non-Euclidian distance on the manifold :

$$d_{texture}(x, y) = \begin{pmatrix} 1 + \sum_{i=1}^n F_x^i F_x^i & \sum_{i=1}^n F_x^i F_y^i \\ \sum_{i=1}^n F_x^i F_y^i & 1 + \sum_{i=1}^n F_y^i F_y^i \end{pmatrix}, \quad (85)$$

and the associated edge stopping function for the GAC becomes :

$$g_{texture} = \frac{1}{\det(d_{texture})}. \quad (86)$$

We let the readers refer to [64] for calculus details of the above formulation, but the underlying idea is the following : if there is a discontinuity in the feature space (border between two regions with different textural properties), $\det(d_{texture})$ will be high, and accordingly $g_{texture}$ will be low. For this reason, we are able to substitute g_{edge} by $g_{texture}$ in the GAC formulation of Section 5.3.1. Figure 37 illustrates $\det(d_{texture})$ and $g_{texture}$ for a synthetic image composed of regions with different textural properties.

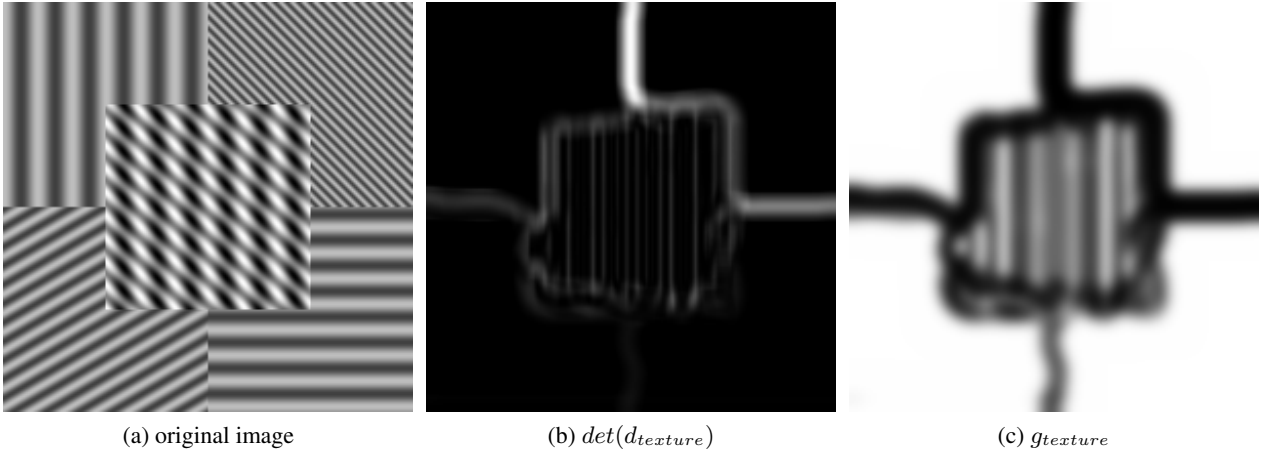


Figure 37: Example of the distance term and the stopping term for a synthetic image.

5.4.3 Minimization Flow

We define our optimal energy $E_{Texture}$ on the multichannel image $M = \{C_\theta, C_f, C_\sigma, C_{\mathfrak{R}e}, C_{\mathfrak{I}m}\}$ introduced in Section 5.4.1 :

$$\begin{aligned} E_{Texture}(\{p_{in}^i, 1 \dots n\}, \{p_{out}^i, 1 \dots n\}, C) &= E_{GAC-Texture} + E_{Region-Texture} \\ &\text{with} \\ E_{GAC-Texture} &= \int_0^1 g_{texture}(I(\mathcal{C}(q))) |\mathcal{C}'(q)| dq \\ E_{Region-Texture} &= + \int_{inside(C)} \frac{1}{N} \sum_{i=1}^N \log(p_{in}^i(M^i(\mathbf{x}))) d\mathbf{x}, \\ &\quad - \int_{outside(C)} \frac{1}{N} \sum_{i=1}^N \log(p_{out}^i(M^i(\mathbf{x}))) d\mathbf{x}, \end{aligned} \quad (87)$$

where p_{in}^i and p_{out}^i are computed using the Parzen windows formula given in Equation (78) of Section 5.3.2.

$E_{GAC-Texture}$ makes a straight use of the edge term defined in Section 5.4.2. $E_{Region-Texture}$ is an extension of classical Chan and Vese term for multichannel images.

The resulting minimization flow expressed as a function of ϕ is :

$$\begin{aligned} \frac{\partial \phi}{\partial t}(\mathbf{x}) = & \delta\epsilon(\phi) \left[\text{div} \left(g_{texture}(\mathbf{x}) \frac{\nabla \phi}{\|\nabla \phi\|}(\mathbf{x}) \right) \right. \\ & - \frac{1}{N} \sum_{i=1}^N \log(p_{in}^i(M^i(\mathbf{x}))) \\ & \left. + \frac{1}{N} \sum_{i=1}^N \log(p_{out}^i(M^i(\mathbf{x}))) \right], \end{aligned} \quad (88)$$

5.4.4 Iris Segmentation Using a Texture-Based Geometric Contour

In the algorithm presented in this section, we first generate the multichannel image M_{iris} describing the textural information in an iris image. To that aim, we must design the Gabor filters in such a way that the differences between the the iris region and other surrounding elements (such as the limbus, the eyelashes, the shadows or the skin) are highlighted. We empirically assess that the filters should verify $\sigma \geq \frac{1}{2f}$ for producing meaningful descriptors. Imposing this constraint on the filters allows characterizing the different elements in the image, as shown in Figure 38. We also noted that studying the texture on both the original image and on an up-sampled (by a factor 2) version of the image allows extracting some extra information from the image (we up-sampled the image instead of changing the filters because we faced some numerical instabilities for filters with a too small spatial extension in the original image).

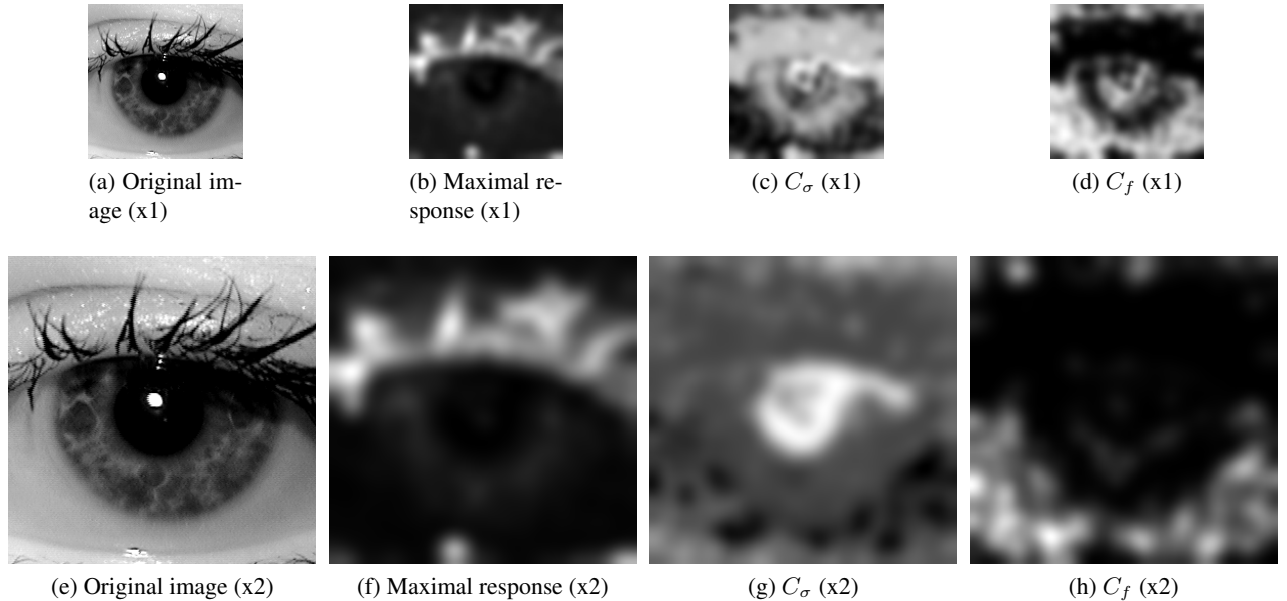


Figure 38: Examples of channels of the Gabor feature space for an iris image. The first row shows the results for the original image and the second row for the image up-sampled by a factor 2.

Now that the filters have been designed, the next step of the process is the generation of the multichannel image M_{iris} describing the textural information in the iris image. At both scales, the maximum response of the filters C_0 , the frequency information C_f and the scale information C_σ provide meaningful information. Moreover, we also found interesting to add the original intensity information I to this textural information in order to make the process more robust when few textural information is available

(because of blur for instance). Therefore, the multichannel image we considered for iris segmentation is $M_{iris} = \{C_0^{(1x)}, C_f^{(1x)}, C_\sigma^{(1x)}, C_0^{(2x)}, C_f^{(2x)}, C_\sigma^{(2x)}, I\}$.

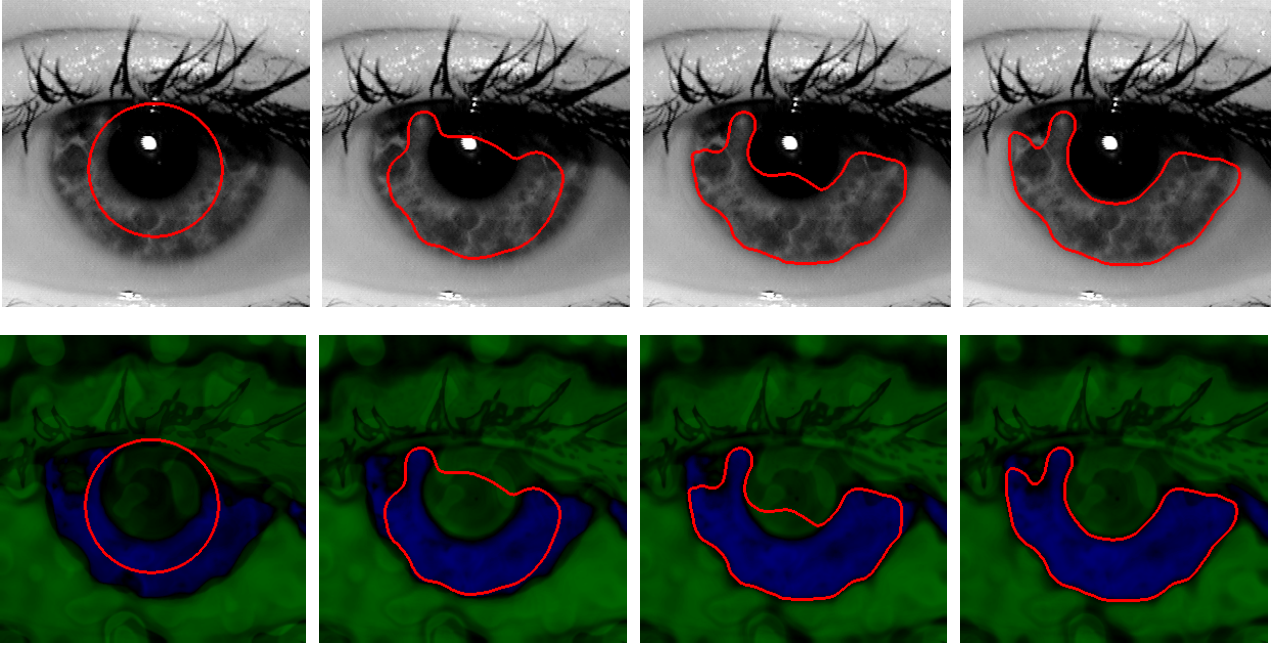


Figure 39: Evolution of the active contour driven by the Gabor framework. The upper row presents the evolution on the original image. The lower one on the region term computed on the whole image showing if it is more likely that a pixel belongs to the inside of the contour (blue) or to the outside (green).

Figure 39 illustrates the evolution of the contour driven by the framework presented in this section. The lower row presents the evolution of the contour and the region term :

$$-\frac{1}{N} \sum_{i=1}^N \log(p_{in}^i(M_{iris}^i(\mathbf{x}))) + \frac{1}{N} \sum_{i=1}^N \log(p_{out}^i(M_{iris}^i(\mathbf{x}))), \quad (89)$$

from equation (88) evaluated for the whole image.

We can see in Figure 39 that the contour produces a very accurate segmentation of the iris borders. It is especially interesting to note that the contour can exclude the shadows on the left part of the iris. Indeed, this shadow has intensities very close to the average iris intensity, but the contour can identify the shadow as not being part of the iris thanks to textural information. It is also interesting to note the evolution of the region term computed on the whole image as the contour evolves. The closer the contour is from its final location, the sharper is the difference between the inner and outer region. This illustrates that this term is pertinent to distinguish the iris region from the other elements in the eye.

5.4.5 Extension of the Textural Information For B-Snakes

In this final section, we propose to extend the ideas we have presented for *Geometric Contours* to *Parametric Contours*. To that aim, we extend the classical B-Snakes formulation given in Section 5.2.2 to exploit the textural information presented in Section 5.4.

The *Classical B-Snakes* implementation of the B-Snake based on [27] evolves using the terms :

$$\begin{aligned} T_{edge}(I) &= \Delta I \\ T_{region}(I) &= -\log \frac{p_{in}(I(\mathbf{x}))}{p_{out}(I(\mathbf{x}))} \end{aligned} \quad (90)$$

with the notations of Section 5.2.2.

Instead, we propose to implement a *Texture B-Snakes* using the terms :

$$\begin{aligned} T_{edge}(I) &= g_{texture}(I) \\ T_{region}(I) &= -\log \sum_{i=1}^N \frac{p_{in}(M_{iris}^i(\mathbf{x}))}{p_{out}(M_{iris}^i(\mathbf{x}))} \end{aligned} \quad (91)$$

with M_{iris} the multichannel image introduced in Section 5.4.4 and $g_{texture}(I)$ the textural edge indicator described in Section 5.4.2.

Figure 40 and 41 illustrates the difference between *Classical B-Snakes* and *Texture B-Snakes*. In the particular image of Figure 40, the upper eyelashes have intensities close to those of the iris because of blur. Therefore, during the first steps of the algorithm, where it mostly relies on region information, *Classical B-Snakes* consider eyelashes as being part of the iris, which will lead to a segmentation failure. At the opposite, *Texture B-Snakes* are able to distinguish eyelashes and the iris thanks to textural information. Therefore, during the first step, the contour evolves roughly to its correct final location, and in the second part of the evolution, the contour accurately fits the iris borders thanks to edge information.

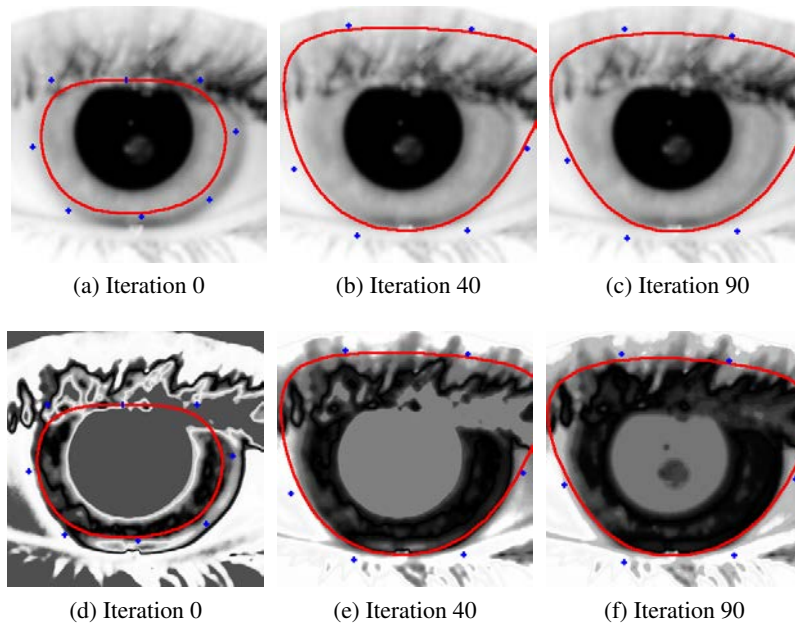


Figure 40: Segmentation of the iris using the *Classical B-Snakes* implementation. The evolution is shown on the original image (upper row) and on the region term T_{region} evaluated on the whole image (lower row). Dark regions in the region term's image show pixels that are more likely to belong to the iris and light pixels that are more likely to belong to the background.

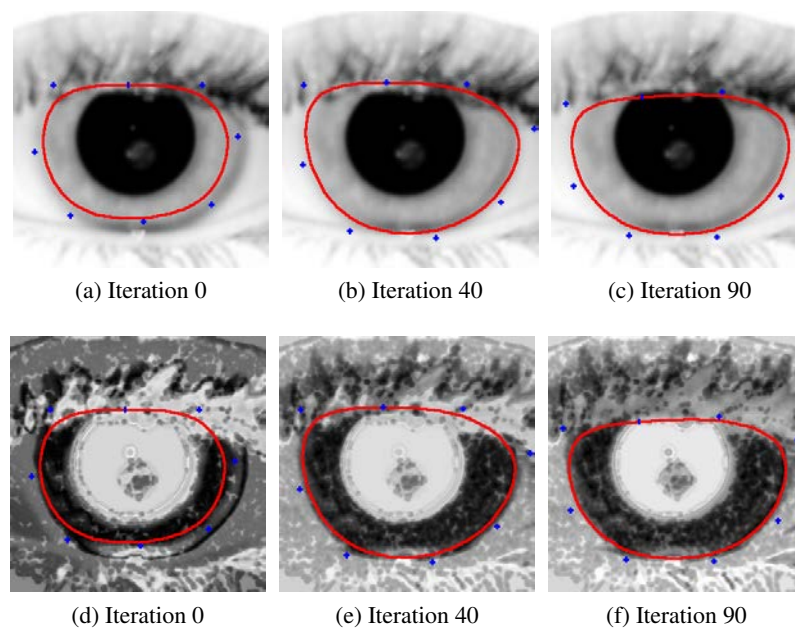


Figure 41: Segmentation of the iris using *Texture B-Snakes* implementation. The evolution is shown on the original image (upper row) and on the region term T_{region} evaluated on the whole image (lower row). Dark regions in the region term's image show pixels that are more likely to belong to the iris and light pixels that are more likely to belong to the background.

5.5 Evaluation

In order to evaluate the performance of the algorithms presented in this Chapter, we select 12,872 left and right irises of the ND-Iris database. We segment each image using the considered Active Contours and evaluate the recognition performance of the complete system for 117,200 intra-class comparisons and 13,790,000 inter-class comparisons. We perform these comparisons using the *Feature Extraction* and *Pattern Matching* algorithms described in Sections ?? and 1.5 of Chapter 1.

For evaluation, we study the differences between the Active Contours described in Sections 5.2, 5.3 and 5.4. We first evaluate the impact of the *Deformation Description* for different *Geometric Contours*. Then, we compare the classical B-Snakes described in Section 5.2.2 with the implementation using texture information described in Section 5.4.5. Finally, we compare *Parametric Contours* versus *Geometric Contours* when using the same *Deformation Description*.

5.5.1 Geometric Contours

In this section, we compare the performance of three *Geometric Contours* presented in this chapter : the *Geodesic Active Contours*, the *Chan and Vese Model*, and the *Integrated Active Contours for texture segmentation*. The evaluation based on the ROC curve displayed in Figure 42 shows that *Geodesic Active Contours* performs the best followed by the *Integrated Active Contours for texture segmentation* and finally by the *Chan and Vese model*.

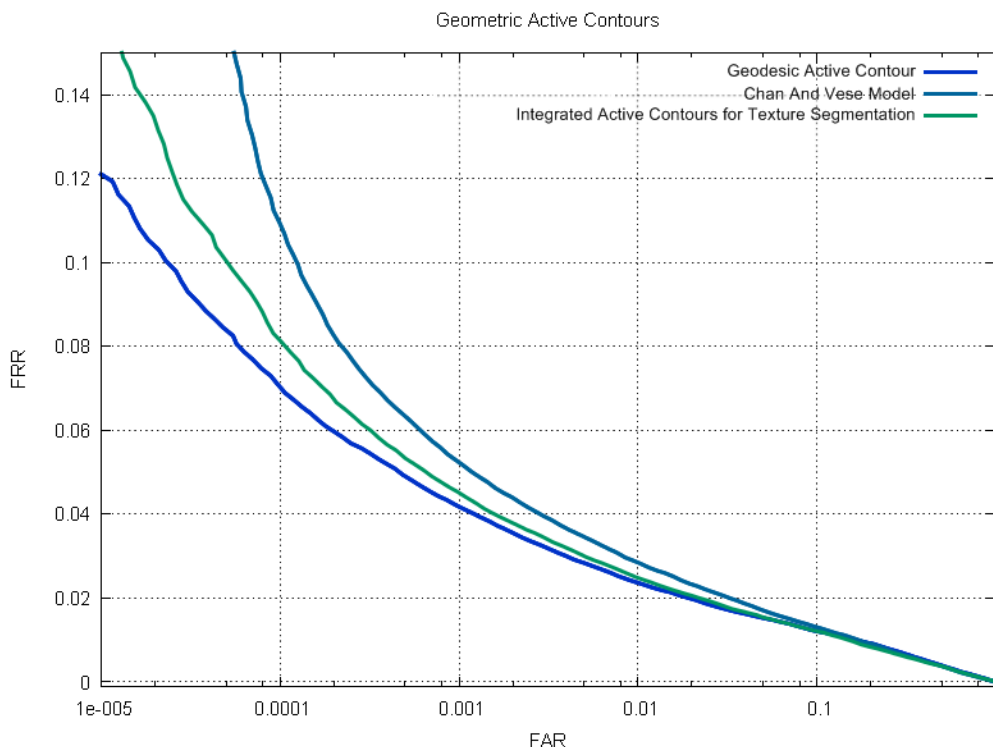


Figure 42: Performance of the three *Geometric Contours* on the ND-Iris database : the *Geodesic Active Contours*, the *Chan and Vese Model*, and the algorithm *Integrated Active Contours for texture segmentation*.

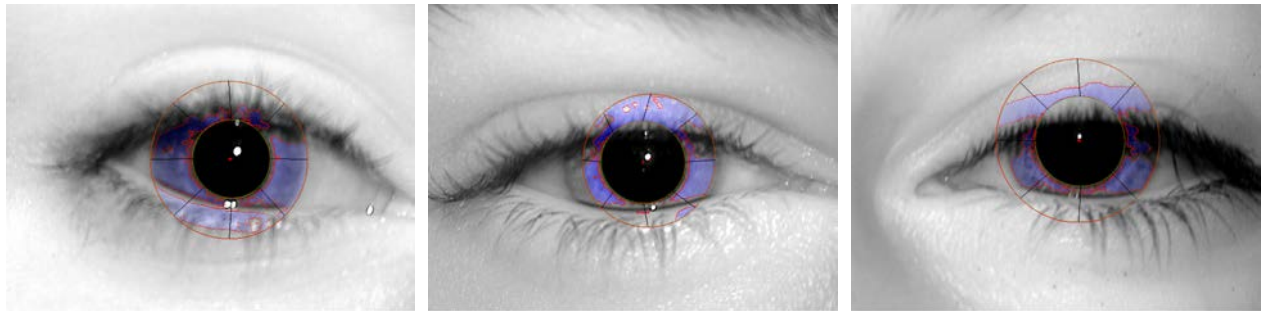
However, random visual inspection of the segmentation results showed that the *Integrated Active Contours for texture segmentation* produce more accurate segmentation masks. On the other hand, we noted that the segmentation failures are not the same for the three algorithms (See Figure 43).

Figure 43 illustrates some segmentation failures of the three considered algorithms. *Geodesic Active Contours* have a tendency to produce over-segmentations due to the balloon force discussed in Section 5.3.1. On the other hand, the *Chan and Vese model* and *Integrated Active Contours for texture segmentation* have a tendency to produce under-segmentations because the iris texture might be inhomogeneous and the contour only segments one part of the iris.

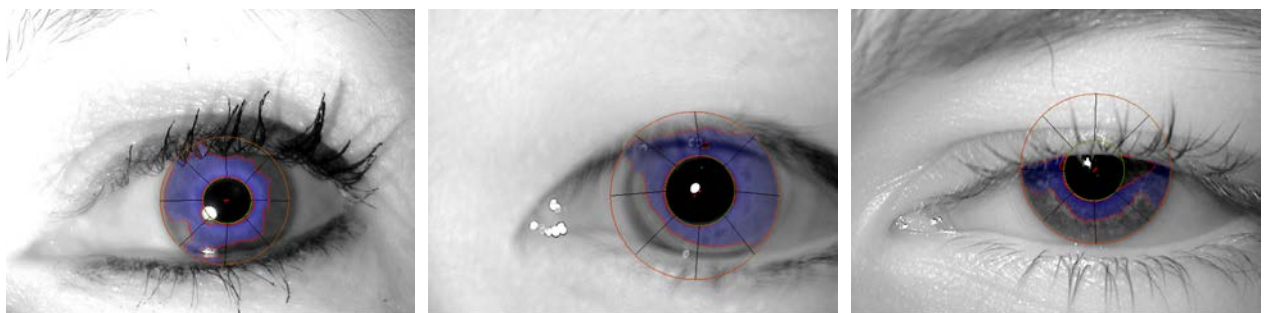
This difference in the nature of failures actually has a severe impact on the recognition performance displayed in Figure 42. Indeed, over-segmentations mostly provide wrong information to the matching process. This wrong information produce an increase of the False Rejection Rate that pushes the whole curve "upper" but does not change its overall shape. On the other hand, under-segmentations result in a lack of information for the matching algorithm. This lack of information has a tendency to produce an increase of False Acceptances that affects the shape of the ROC curve. Indeed, this increase of False Acceptances has no influence on the ROC curve for high FAR but highly decreases the recognition performance for a low FAR.

We noted that *Integrated Active Contours for texture segmentation* and *Chan and Vese Model's* algorithm tend to produce more under-segmentation whereas *Geodesic Active Contours* tend to produce over-segmentation. The case of under-segmentation is more critical for the matching algorithm since information is lost. Moreover, the pattern matching here used is extremely sensitive to the number of bit comparisons that is carried out when two images are matched : this is discussed by Daugman in [16] and Cremer in [13, 14]. Indeed, matching algorithm is based on a test of statistical independence which requires a large number of bit comparisons in order to be statistically relevant.

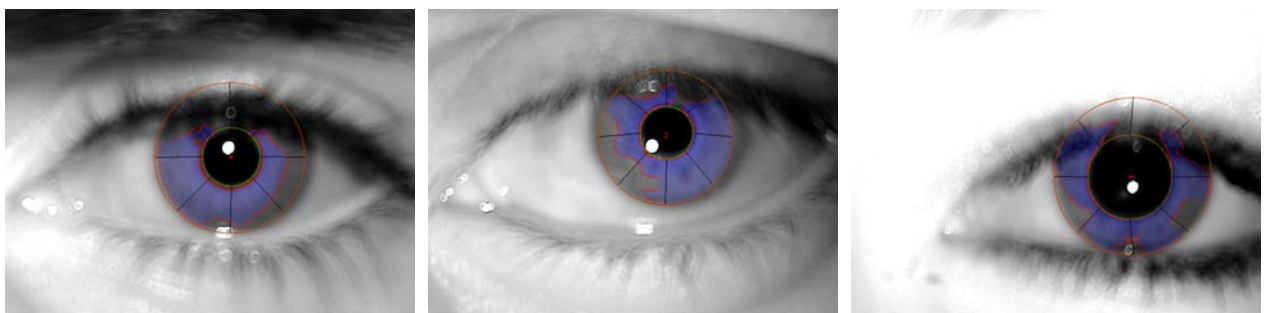
This trait of the recognizer here used for evaluating the segmentation leads to the following conclusion : a sever lack of information (case of under-segmentation) can be more damaging for the complete system here considered (segmentation system and recognizer) than providing inaccurate segmentation mask (case of over-segmentation).



(a) failure of the *Geodesic Active Contours* (b) failure of the *Geodesic Active Contours* (c) failure of the *Geodesic Active Contours*



(d) failure of the *Chan and Vese Model* (e) failure of the *Chan and Vese Model* (f) failure of the *Chan and Vese Model*



(g) failure of the *Integrated Active Contours for texture segmentation* (h) failure of the *Integrated Active Contours for texture segmentation* (i) failure of the *Integrated Active Contours for texture segmentation*

Figure 43: Segmentation failures resulting from the three variations of *Geometric Contours* here studied.

5.5.2 Parametric Contours

In this section, we compare the classical B-Snakes described in Section 5.2.2 with the implementation using texture information described in Section 5.4.5. To do so, we use the same settings for the two algorithms, but they only differs in terms of the *Deformation Description* : the *classical B-Snake* uses the image intensity information to compute the edge term and the region term, whereas the *textural B-Snake* uses texture information instead.

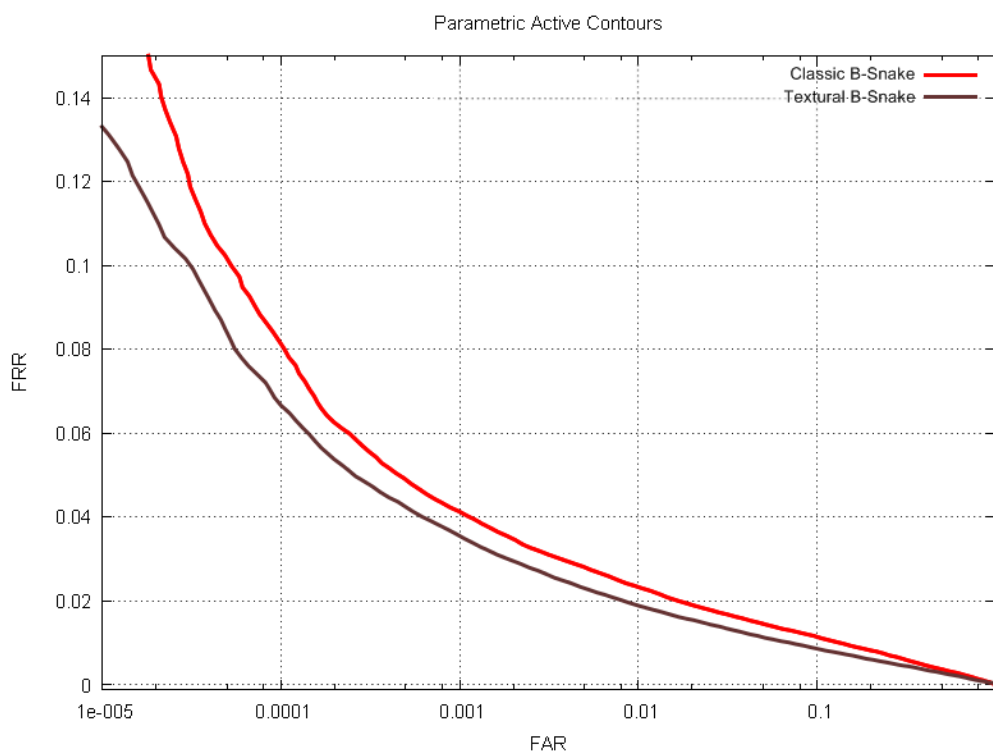
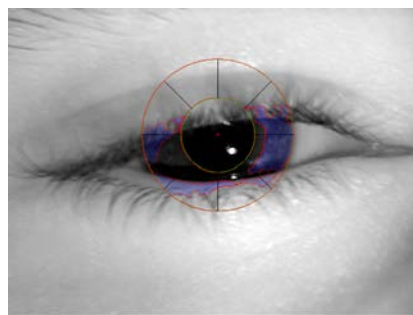


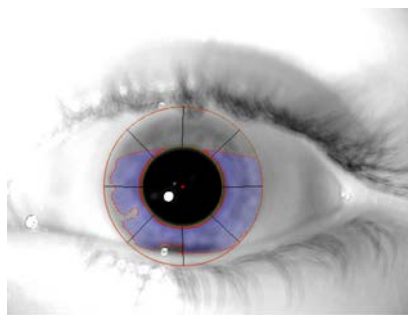
Figure 44: Performance of two *Parametric Contours* on the ND-Iris database : the *classical B-Snake* and the *textural B-Snake*.

Figure 44 shows the ROC curves for the two B-Snakes implementations. They have roughly the same behavior but the B-Snake based on texture information performs better than the one based on the classical formulation. This result shows that the textural information provide a surplus of information compared to the intensity information.

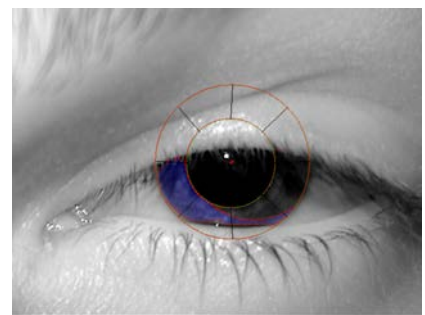
Figure 45 illustrates some segmentation failures of the B-Snakes algorithms. We can see that the two algorithms have a tendency to fail on images in which the iris is highly occluded. Figure 45b illustrates an image in which the *classical B-Snake* fails to fully segment the iris. This comes from the fact that the limbus has intensities close to those of the iris, thus weakening the strength of the region force. Figure 45e shows a failure of the *textural B-Snake* because the contour is initialized mostly on eyelashes. Therefore, the contour mostly catches the properties of eyelashes instead of those of the iris, leading to a wrong segmentation.



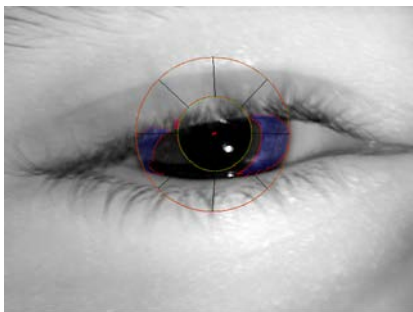
(a) *Classical B-Snakes* failure



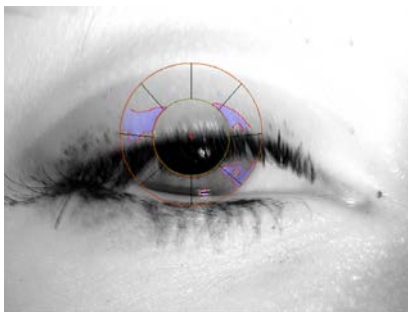
(b) *Classical B-Snakes* failure



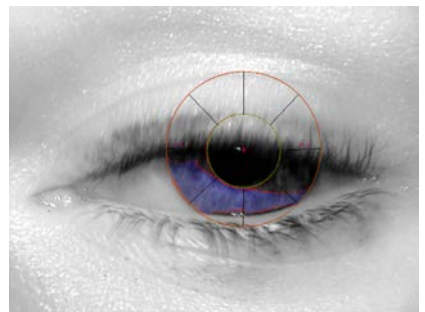
(c) *Classical B-Snakes* failure



(d) *Texture B-Snakes* failure



(e) *Texture B-Snakes* failure



(f) *Texture B-Snakes* failure

Figure 45: Segmentation failures of the *Classical B-Snakes* and *Texture B-Snakes* algorithms.

5.5.3 Deformation Description

In this section, we compare *Parametric Contours* versus *Geometric Contours* when they use the same *Deformation Description*. We first compare B-Snakes based on intensity information described in Section 5.2.2 with the Chan and Vese Model described in Section 5.3.2. Then, we compare B-Snakes based on textural information described in Section 5.4.5 with the *Integrated Active Contours for texture segmentation* described in Section 5.4.

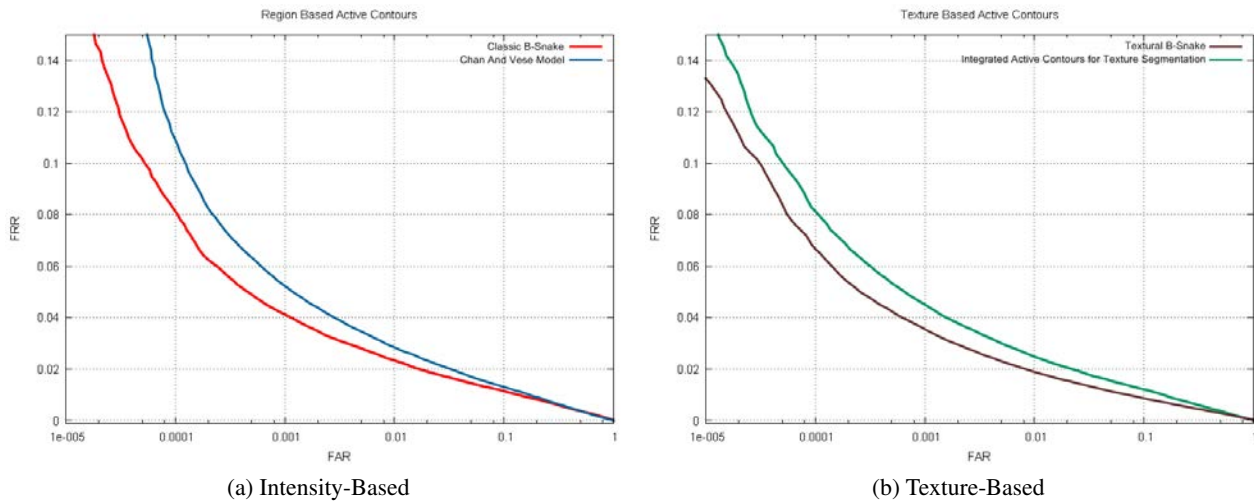


Figure 46: Comparison between *Parametric Contours* and *Geometric Contours* that work either with an Intensity-Based or a Texture-Based *Deformation Description*.

Figure 46 shows the ROC curves of the two experiments. We can see that in both experiments *Parametric Contours* perform better than *Geometric Contours*. This is an *a priori* unexpected result because random visual inspection of segmentation results shows that *Geometric Contours* tend to produce more accurate segmentation masks than *Parametric Contours* do.

Indeed, *Geometric Contours* can take almost any shape during the evolution, leading to very accurate segmentation masks. However, they are also more likely to produce results which are very different from the expected segmentation : when the contour starts being attracted by non iris regions (eyelashes or limbus for instance) they can freely evolve far from the iris region. On the other hand, *Parametric Contours* are more constrained during their evolution because of the limited number of control points that we used (10 in our experiments). Therefore, they are less likely to move far from the iris region.

This experiment shows that despite *a priori* better visual results, the lack of robustness of *Geometric Contours* has a negative impact on a large scale evaluation.

5.6 Conclusion

In this chapter, we have studied several Active Contours implementations for the iris region segmentation. We implemented *Parametric Contours* and *Geometric Contours* for which we studied several *Deformation Descriptions*. The evaluation conducted in Section 5.5 allowed us to identify general results :

- *Parametric Contours* perform better than *Geometric Contours* for the same *Deformation Description* because *Parametric Contours* are more robust than *Geometric Contours*.
- It is not pertinent to evaluate a segmentation algorithm by the accuracy of the generated segmentation masks.
- In our case, algorithms producing over-segmentations must to be favored compared those giving under-segmentations, as the *Pattern Matching* is done with Daugman's algorithm (described in Section 1.5).

Setting the balance between accuracy and robustness is a traditional issue in segmentation. Our experiments have shown that for iris segmentation, the robustness of the segmentation must be favored w.r.t. segmentation accuracy. This can be explained by the *Feature Extraction* and *Pattern Matching* stages that we used in this thesis is based on a test of statistical independence on a large number of bit comparisons. Therefore, this process can handle some small inaccuracies and still be discriminative. This behavior can be interpreted in terms of bias-variance trade-off. Indeed, the *Geometric Contours* have a small bias and a large variance, and the *Parametric Contours* a small variance but a large bias. From our experiments, it appears that having a small variance is actually more important than having a small bias when using the Daugman's matching algorithm.

However, if the segmentation of one image fails completely, the *Feature Extraction* and *Pattern Matching* algorithms will have only false information to rely on. Therefore, this wrongly segmented image will provoke a large number of False Acceptances and False Rejections, which will highly decrease the recognition performance of the system. For this reason, we also consider the issue of automatic detection of segmentation failures in order to remove such images from the evaluation process. We address this last issue in Chapter 7.

In order to improve the segmentation stage on its own, we propose to use more shape-constrained segmentation algorithms. Indeed, the *Deformable Models* formalism allows to force the contour to stay close to a reference shape. For instance, we could impose Active Contours to remain close to a circle. This extra shape information should be extremely effective to reduce the amount of under-segmentations that have a very negative impact on the recognition performance of the complete system here considered.

In the next chapter, we will discuss the last processing stage of our iris segmentation system : the accurate estimation of the iris borders in order to generate the normalized image of iris texture.

CONFIDENTIAL

Chapter 6

Contour Fitting for Normalization

In iris recognition systems, following the approach in [16], matching is done by aligning the features of specific points (application points) from normalized images, i.e., the rectangular unwrapped texture of the two irises (see Section 1.4 of Chapter 2). If the unwrapping is not done properly, the two textures, and therefore, the corresponding points, do not correctly align and recognition performance is affected. Consequently, the robustness and precision of the unwrapping is a critical aspect of iris recognition systems.

This unwrapping is done using a parametric description of the iris borders independently of the occlusions. Most methods of the literature apply a simple estimator procedure, such as the Hough Transform or Least Square Fitting to get this parametric description. Those fitting methods rely on segmentation results for estimating the model's parameters, and therefore, inaccuracies in segmentation results lead to errors in the resulting contour.

The iris borders have traditionally been modeled as a pair of circles [16, 77]. However, this assumption is incorrect for off-angle images or for people with a pupil or an iris which is anatomically non circular. To address this shortcoming, more general shapes need to be considered, such as ellipses or free-shape parametric contours.

Ellipses are the intuitive extension of circles, but the two classical operators used to find circles in iris literature (Daugman's Integrodifferential Operator (IDO) and the Circular Hough Transform (CHT)) do not extend well to ellipses. These operators require the evaluation of an accumulator whose dimension is the number of parameters in the model. Spanning from 3 parameters for the circle to 5 for the ellipse dramatically increases computation time, making these operators unsuitable for real-time applications. For this reason, most articles working with ellipses use an elliptic Direct Least Square Fitting (DLS) [22] to fit ellipses onto the iris borders. This method is very fast, but suffers from the usual drawback of least square methods : sensitivity to outliers.

Free-shape parametric contours are able to handle any shape for iris borders [17, 26, 57]. They can achieve very good recognition performance but are strongly dependent on the segmentation stage. Hence, in order to apply them, it is necessary to efficiently distinguish between the pixels belonging to the anatomic borders of the iris and the edges generated by occlusions. If some pixels are misclassified, a large part of the resulting contours may be affected, leading to a decrease of system's recognition performance.

We favor a fixed-shaped model (ellipses) over a free-shape model because the latter is too sensitive to segmentation accuracy. Accordingly, we propose a precise, robust and effective way to fit ellipses on iris borders in order to perform iris normalization. Our system acts at two levels :

- First, we define an effective iterative scheme based on deformable models working specifically with ellipses. Our method, which we have called Elliptic Variational Fitting (EVF) allows us to morph a roughly initialized circle onto the exact borders of the iris. The interesting aspect of this approach is that unlike most fitting methods, it is independent of segmentation results' accuracy. As a consequence, our proposed method can even correct some segmentation inaccuracies in addition to giving suitable contours for normalization.
- The second level is dedicated to the initialization of the iterative process. We propose to initialize the EVF framework for several initial locations and select the ellipse that reaches the lowest minimum with respect to the energetic functional of the model. The complete process is summarized in Figure 47.

This Chapter is organized as follows: Section 6.1 presents the method we developed to fit the ellipses.

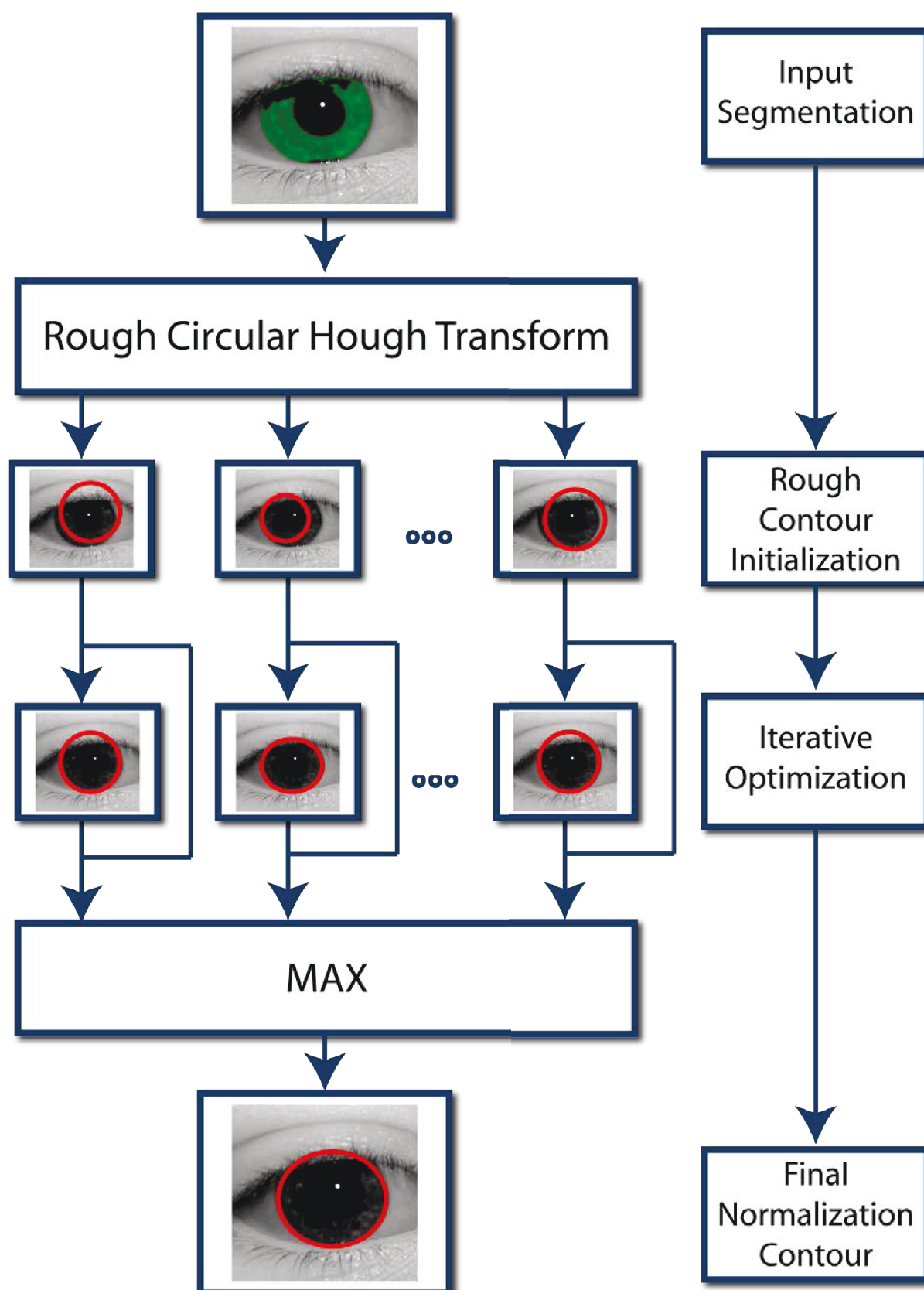


Figure 47: Our elliptic contour fitting process in order to generate the contours for normalization. The input region segmentation is used to initialize a rough Circular Hough Transform. This Circular Hough Transform provides a set of initial circles close to the expected solution. Each circle is morphed into an ellipse using our EVF framework. Finally, the ellipse reaching the best location is selected for normalization.

We first present the general formulation of our model and give an explicit formulation of the equations for the circles and the ellipses. Then, in Section 6.2, we expose how we have implemented the complete contour fitting process presented in Figure 47 into our complete system presented in Chapter A. Finally, Section 6.3 evaluates the performance of our system for different contour fitting algorithms on reference databases. In this section, we also address specific issues like the evaluation on off-angle images and computation times.

6.1 Parametric Deformable Template Optimization

In this section, we describe how we express the contour fitting problem as an optimization problem using the general framework of deformable models introduced in Section 5.1 of Chapter 5. Then, we give the explicit formulation of the contour evolution for circles and ellipses.

6.1.1 General Formulation

As explained in the introduction, we use a fixed-shape contour as opposed to a free-shape contour used for segmentation in Chapter 5. To that aim, we use the formalism of deformable template introduced by Yuille et al. in [78]. In this context, the contour is described using a small number of parameters $\Theta = \{\theta_i, i = 1 \dots n\}$ describing its complete shape using a set of parametric equations.

For free-shape contours, the evolution is expressed locally. For instance, B-Snakes of Section 5.2.2 model the contour by a set of control points. Each control point defines the shape of the contour in its neighborhood. At the opposite, using the deformable template framework, changing one of the parameter affects the overall shape of the contour. For instance, for a circle we have $\Theta_{circle} = \{x_c, y_c, r\}$ with $\mathbf{x}_c = (x_c, y_c)$ the center of the circle and r its radius. The change of any parameter of Θ_{circle} affects the whole circle.

Parametric Deformable Templates are very classical in Computer Vision but are almost absent from the iris recognition literature, the only reference being Miyazawa et al. in [46] where a model with 10 parameters is used to find simultaneously the inner and outer iris ellipses and both eyelids. Although the model assumes an ellipse for pupil and iris borders, they are enforced to keep a circular shape in the equations. In the present work, we explicitly provide a new algorithm to solve the optimization problem in the elliptic case.

If we go back to the formalism of deformable models introduced in Section 5.1 of Chapter 5, the set of parameters $\Theta = \{\theta_i, i = 1 \dots n\}$ gives the *Shape Description*. To describe the *Deformation Description*, we define an objective energy in the same fashion to what has been done in Chapter 5. Our energy functional contains three terms : a regularization energy E_{int} , ensuring that the contour remains coherent with some prior assumption of shape and location. An edge energy E_{edge} , ensuring that the contour passes through areas of strong gradient and a region energy E_{region} to force the contour to separate regions that have different statistical properties.

The traditional deformable model formulation is defined on the contour C as follows :

$$E_{PDM}(C) = E_{edge}(C) + E_{region}(C) + E_{int}(C). \quad (92)$$

It can be expressed on the set of parameters $\Theta = \{\theta_i, i = 1 \dots n\}$ instead :

$$E_{PDM}(\Theta) = E_{edge}(\Theta) + E_{region}(\Theta) + E_{int}(\Theta) \quad (93)$$

One way to minimize E_{PDM} w.r.t. C is to fix all but one parameter θ_j^k and optimize the functional w.r.t. such parameter using the gradient descent algorithm :

$$\theta_j^{k+1} = \theta_j^k - \frac{\partial}{\partial \theta_j^k} [E_{edge}(\Theta) + E_{region}(\Theta) + E_{int}(\Theta)]. \quad (94)$$

We define the edge term E_{edge} so that it promotes areas having a strong normal gradient and a low tangent one:

$$E_{edge}(C) = E_n(C) + E_t(C), \quad (95)$$

$$\begin{aligned} E_n(C) &= \int_C \left\| \nabla I(\mathbf{x}) \cdot \vec{n}(\mathbf{x}) \right\| \frac{1}{\|C'(\mathbf{x})\|} d\mathbf{x}, \\ E_t(C) &= - \int_C \left\| \nabla I(\mathbf{x}) \cdot \vec{t}(\mathbf{x}) \right\| \frac{1}{\|C'(\mathbf{x})\|} d\mathbf{x}, \end{aligned} \quad (96)$$

where I is the image, $\vec{n}(\mathbf{x})$ the outward unit normal vector to the curve at point \mathbf{x} , $\vec{t}(\mathbf{x})$ the unit tangent vector to the curve at point \mathbf{x} and $C'(\mathbf{x})$ derivative of the curve at point \mathbf{x} .

For the region term, we use the Region Competition formulation introduced in Section 5.3.2 of Chapter 5 working with the intensity distributions.

$$E_{region}(C) = - \int_{R_{in}} \log p_{in}(I(\mathbf{x})) d\mathbf{x} - \int_{R_{out}} \log p_{out}(I(\mathbf{x})) d\mathbf{x}, \quad (97)$$

where p_{in} and p_{out} are respectively the probabilities of a given intensity to be inside or outside the curve (we note R_{in} the region inside C and we note R_{out} the region outside C):

$$\begin{aligned} p_{in}(k) &= \frac{\int_{R_{in}} \{I(\mathbf{x}) = k\} d\mathbf{x}}{\int_{R_{in}} d\mathbf{x}}, \\ p_{out}(k) &= \frac{\int_{R_{out}} \{I(\mathbf{x}) = k\} d\mathbf{x}}{\int_{R_{out}} d\mathbf{x}}. \end{aligned} \quad (98)$$

In the following section we will describe how we apply this general framework to circles and ellipses.

6.1.2 Circular Variational Fitting (CVF) Model

The use of circles is motivated by two main reasons. First it is the simplest way to model the iris borders, and therefore many methods in the literature still rely on circles for normalization. Having an accurate circular model allows us to compare our results with those of the literature. The second interesting aspect comes from the simplicity of the model : as the model is extremely constrained, it is also extremely robust.

In this context, the set of parameters $\{\theta_i, i = 1..n\}$ from the previous section becomes $\{x_c, y_c, r\}$ with $\mathbf{x}_c = (x_c, y_c)$ the center of the circle and r its radius. The related parametric equations are :

$$\begin{aligned} x(\theta) &= x_c + r \cos(\theta), \\ y(\theta) &= y_c + r \sin(\theta), \end{aligned} \quad (99)$$

with $\theta \in [0, 2\pi[$.

We give the derivatives of E_{edge} and E_{region} w.r.t. the parameters $\mathbf{x}_c = (x_c, y_c)$ and r in Appendix B :

$$\begin{aligned} \frac{\partial E_n}{\partial \mathbf{x}_c} &= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial r} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] \cdot \vec{n}_\theta dir(\vec{n}_\theta) d\theta, \end{aligned} \quad (100)$$

and :

$$\begin{aligned} \frac{\partial E_t}{\partial \mathbf{x}_c} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{t}_\theta \right] dir(\vec{t}_\theta) d\theta, \\ \frac{\partial E_t}{\partial r} &= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] \cdot \vec{t}_\theta dir(\vec{t}_\theta) d\theta, \end{aligned} \quad (101)$$

with :

$$\text{dir}(\vec{n}_\theta) = \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right], \quad (102)$$

$$\text{dir}(\vec{t}_\theta) = \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{t}_\theta \right], \quad (103)$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_c} E_{region} &= \frac{1}{2} \int_0^{2\pi} D(\mathbf{x}) \vec{n}_\theta d\theta, \\ \frac{\partial}{\partial r} E_{region} &= \frac{1}{16} \int_0^{2\pi} D(\mathbf{x}) d\theta, \end{aligned} \quad (104)$$

To define the regularization term, we consider the order in which normalization contours are found in our system. As the pupil contour is found first, we have no additional information on the pupil's location, so we set the pupil regularization energy $E_{int}^p = 0$. The iris contour is found after the pupil contour in order to exploit the pupil's location. Accordingly, we set the iris regularization energy $E_{int}^i = \|\mathbf{x}_c^i - \mathbf{x}_c^p\|^2$ where \mathbf{x}_c^i is the center of the iris and \mathbf{x}_c^p the center of the pupil. This energy acts as a string restricting the iris center to stay close to the pupil's center. The resulting partial derivatives are :

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}_c^i} E_{int}^i &= 2(\mathbf{x}_c^p - \mathbf{x}_c^i), \\ \frac{\partial}{\partial r^i} E_{int}^i &= 0, \end{aligned} \quad (105)$$

The above equations defined on a circular contour are interesting because they have a very simple expression that allow understanding the evolution of the contour. For instance, the evolution of the radius r according to the region energy E_{region} (see Equation (104)) can be seen as a vote for all the points of the contour to decide whether the circle should shrink or expand. If for a majority of contour points \mathbf{x} , we have :

$$D(\mathbf{x}) = -\log \frac{p_{in}(I(\mathbf{x}))}{p_{out}(I(\mathbf{x}))} < 0, \quad (106)$$

then a majority of contour points vote that they are "inside". Therefore, the radius should become larger. Accordingly, we define the evolution equation for an arbitrary time parameter k as :

$$r^{k+1} = r^k - \frac{\partial}{\partial r} E_{region}. \quad (107)$$

This definition ensures that if $\frac{\partial}{\partial r} E_{region} < 0$ the radius r will increase.

Figure 48 illustrates the evolution of a circle using the above framework. As we can see, the fitting is rather accurate, but it is not perfect because the circular shape assumption does not match the exact shape of the iris. Therefore we extend the above formulation to ellipses.

6.1.3 Elliptic Variational Fitting (EVF) Model

In the context of ellipses, the set of parameters $\{\theta_i, i = 1..n\}$ becomes $\{x_c, y_c, a, b, \phi\}$ where $\mathbf{x}_c = (x_c, y_c)$ is the center of the ellipse, a and b are the major and minor axes of the ellipse and ϕ is the angle of the ellipse with the reference vector of the image domain. The resulting parametric equations are :

$$\begin{aligned} x(\theta) &= x_c + a \cos(\theta) \cos(\phi) - b \sin(\theta) \sin(\phi), \\ y(\theta) &= y_c + a \cos(\theta) \sin(\phi) + b \sin(\theta) \cos(\phi), \end{aligned} \quad (108)$$

where $\theta \in [0; 2\pi[$. Using the vectors \vec{n}_ϕ and \vec{t}_ϕ :

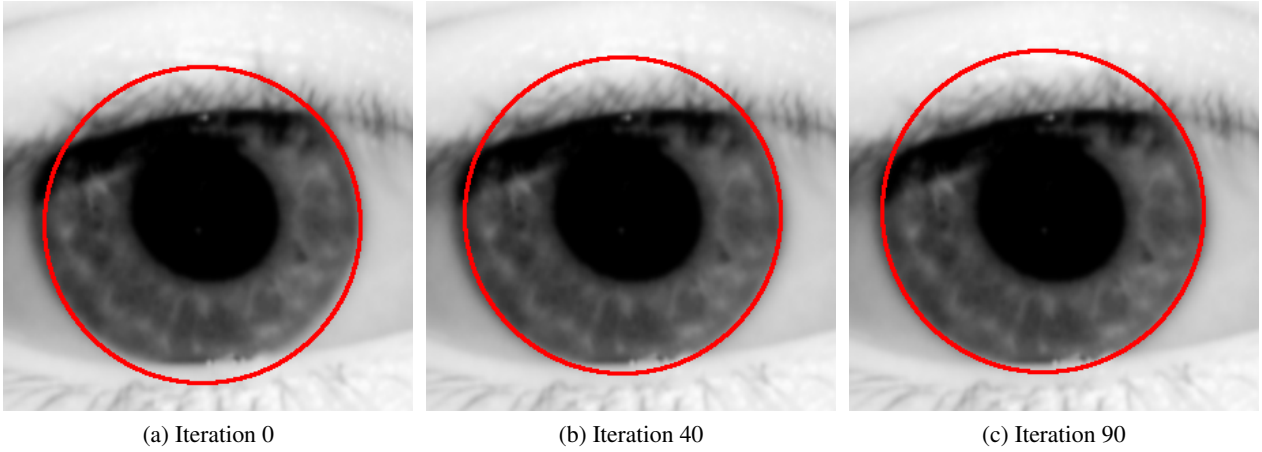


Figure 48: The circle is initialized close to the iris borders and move onto the iris borders using our Circular Variational Fitting (CVF) method.

$$\vec{n}_\phi = \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}, \quad (109)$$

and

$$\vec{t}_\phi = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \end{pmatrix}, \quad (110)$$

the ellipse can be compactly written as :

$$\mathbf{x}(\theta) = \mathbf{x}_c + a \cos(\theta) \vec{n}_\phi + b \sin(\theta) \vec{t}_\phi. \quad (111)$$

We give the derivation of E_{edge} and E_{region} in the Appendix B : So :

$$\begin{aligned} \frac{\partial E_n}{\partial \mathbf{x}_c} &= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial a} &= \int_0^{2\pi} \left[\cos(\theta) \left[H(\mathbf{x}) \vec{n}_\phi \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\sin(\theta)}{\|N(\theta)\|} \left[\vec{t}_\phi + a \vec{n}_\theta \frac{\sin(\theta)}{\|N(\theta)\|} \right] \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial b} &= \int_0^{2\pi} \left[\sin(\theta) \left[H(\mathbf{x}) \vec{t}_\phi \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\cos(\theta)}{\|N(\theta)\|} \left[\vec{n}_\phi + b \vec{n}_\theta \frac{\cos(\theta)}{\|N(\theta)\|} \right] \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial \phi} &= \int_0^{2\pi} \left[\left[H(\mathbf{x}) (-b \sin(\theta) \vec{n}_\phi + a \cos(\theta) \vec{t}_\phi) \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \vec{t}_\theta \right] dir(\vec{n}_\theta) d\theta, \end{aligned} \quad (112)$$

And :

$$\begin{aligned} \frac{\partial E_t}{\partial \mathbf{x}_c} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{t}_\theta \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_t}{\partial a} &= - \int_0^{2\pi} \left[\cos(\theta) \left[H(\mathbf{x}) \vec{n}_\phi \right] \cdot \vec{t}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\sin(\theta)}{\|T(\theta)\|} \left[\vec{n}_\phi + a \vec{t}_\theta \frac{\sin(\theta)}{\|T(\theta)\|} \right] \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_t}{\partial b} &= - \int_0^{2\pi} \left[\sin(\theta) \left[H(\mathbf{x}) \vec{t}_\phi \right] \cdot \vec{t}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\cos(\theta)}{\|T(\theta)\|} \left[-\vec{t}_\phi + b \vec{t}_\theta \frac{\cos(\theta)}{\|T(\theta)\|} \right] \right] dir(\vec{n}_\theta) d\theta, \\ \frac{\partial E_t}{\partial \phi} &= - \int_0^{2\pi} \left[\left[H(\mathbf{x}) (-b \sin(\theta) \vec{n}_\phi + a \cos(\theta) \vec{t}_\phi) \right] \cdot \vec{t}_\theta + \nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] dir(\vec{n}_\theta) d\theta, \end{aligned} \quad (113)$$

$$\begin{aligned}
\frac{\partial E_r}{\partial \mathbf{x}_c} &= \int_0^{2\pi} [AN(\theta)] D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial a} &= \frac{1}{8a} \int_0^{2\pi} \cos^2(\theta) D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial b} &= \frac{1}{8b} \int_0^{2\pi} \sin^2(\theta) D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial \phi} &= \frac{1}{8} \left(-\frac{1}{\sigma_x^2} + \frac{1}{\sigma_y^2} \right) \int_0^{2\pi} \left[N(\theta)^T \begin{pmatrix} \sin(2\phi) & \cos(2\phi) \\ \cos(2\phi) & -\sin(2\phi) \end{pmatrix} N(\theta) \right] D(\mathbf{x}) \|T(\theta)\| d\theta.
\end{aligned} \tag{114}$$

We define two different behaviors for E_{int} . For the pupil, it is necessary that the minor and major axis do not differ much, i.e. $a \approx b$. This means that if there is a lack of information, the pupil prefers circular shapes. For the iris, the two focus points of the iris ellipse must stay close to the pupil ones, which are already fixed.

$$E_{int}^p = \|a_p - b_p\|^2, \tag{115}$$

$$E_{int}^i = \left\| \mathbf{x}\mathbf{f}_i^{(1)} - \mathbf{x}\mathbf{f}_p^{(1)} \right\|^2 + \left\| \mathbf{x}\mathbf{f}_i^{(2)} - \mathbf{x}\mathbf{f}_p^{(2)} \right\|^2, \tag{116}$$

where $\mathbf{x}\mathbf{f}_p^{(1)}$ and $\mathbf{x}\mathbf{f}_p^{(2)}$ are the two focus points of the pupil, $\mathbf{x}\mathbf{f}_i^{(1)}$ is the iris ellipse's focus point closest to $\mathbf{x}\mathbf{f}_p^{(1)}$ and $\mathbf{x}\mathbf{f}_i^{(2)}$ the one closest to $\mathbf{x}\mathbf{f}_p^{(2)}$.

The calculus of the derivatives for the pupil shape energy is straightforward :

$$\begin{aligned}
\frac{\partial E_{int}^p}{\partial \mathbf{x}_c} &= 0, \\
\frac{\partial E_{int}^p}{\partial a} &= (a - b), \\
\frac{\partial E_{int}^p}{\partial b} &= (b - a), \\
\frac{\partial E_{int}^p}{\partial \phi} &= 0.
\end{aligned} \tag{117}$$

Using the notations of Appendix B we have :

$$\begin{aligned}
\frac{\partial E_{int}^i}{\partial \mathbf{x}_c} &= 2(\mathbf{x}_c^i - \mathbf{x}_c^p), \\
\frac{\partial E_{int}^i}{\partial a} &= \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \\
\frac{\partial E_{int}^i}{\partial b} &= -\delta_{pi} \frac{b}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \\
\frac{\partial E_{int}^i}{\partial \phi} &= -2\delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \sqrt{(a^p)^2 - (b^p)^2} \sin(\phi_i - \phi_p).
\end{aligned} \tag{118}$$

These partial derivatives are used to make the contour evolve using the gradient descent scheme of Equation 94. An example of such evolution is given in Figure 49

Though less intuitive than the equations of the circular model, the underlying ideas remain the same. For instance, the evolution of the big axis a according to E_{region} is very close to the evolution of the radius for the circular model. The only difference is the weighting of the region term $D(\mathbf{x})$ by factor $\cos^2(\theta) \frac{\|T(\theta)\|}{a}$. This weighting corresponds to the relative importance of the major and minor axis.

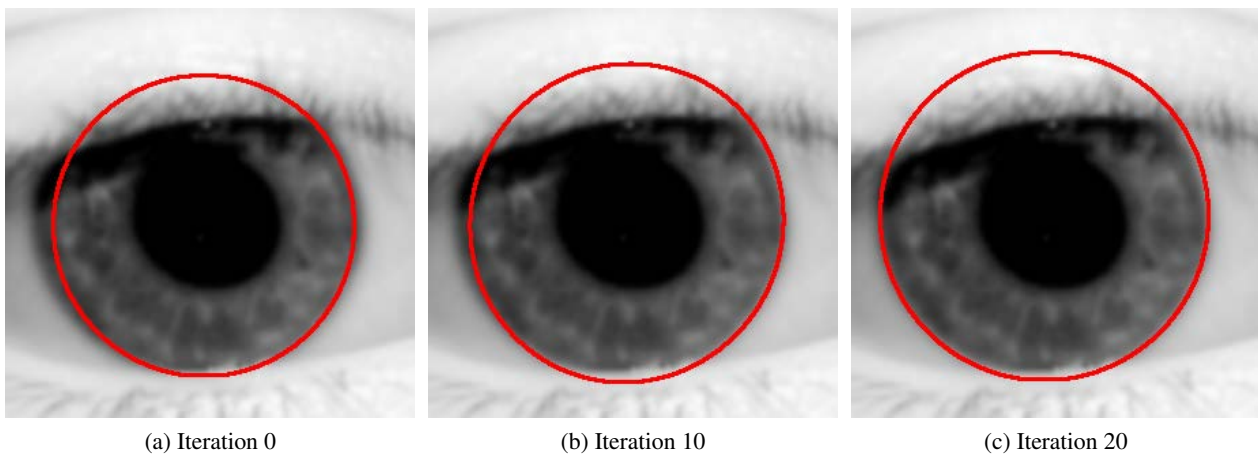


Figure 49: The circle is initialized close to the iris borders and move onto the iris borders using our Elliptic Variational Fitting (EVF) method. Accordingly, the intial circle morphs into an ellipse during the process.

6.2 Model Initialization

Section 6.1 gives an effective way of finding ellipses for normalization, but as with every method based on variational optimization, final results depend on the initialization. If the initialization is too far from the optimal solution, the process may stop in a local minimum of equation (93). In this section, we propose an effective way to initialize the EVF process in order to make it more robust to local minima.

The key idea is to run the algorithm for several initial locations and to select afterward the most suitable ellipse for normalization. The proposed algorithm is composed of three consecutive steps illustrated in Figure 50:

- (i) *Circular Init.* Segmentation results are used to perform a sub sampled Circular Hough Transform to obtain a set of initial circles (see Figure 50b).
- (ii) *EVF Opt.* EVF optimization is applied to all circles from the initial set to generate a set of optimized ellipses (see Figure 50c).
- (iii) *Final Selection* The ellipse reaching the lowest minimum for equation (93) after the *EVF Opt.* step is selected as input to the normalization process (see Figure 50d).

We generate the set of initial circles from the segmentation results displayed in Figure 50a (here resulting from the *Classical B-Snakes* algorithm of Section 5.2.2 of Chapter 5). As the purpose is only to get a rough initialization, it is possible to heavily sub sample the Hough Transform : it makes the algorithm run faster and allows a larger exploration of the parameter space to be performed. Circles obtained at the stage *Circular Init.* are optimized using the EVF scheme more precisely detailed in Section 6.1.3. Finally, we select the ellipse reaching the lowest minimum for equation (93) within the set of ellipses from the previous step.

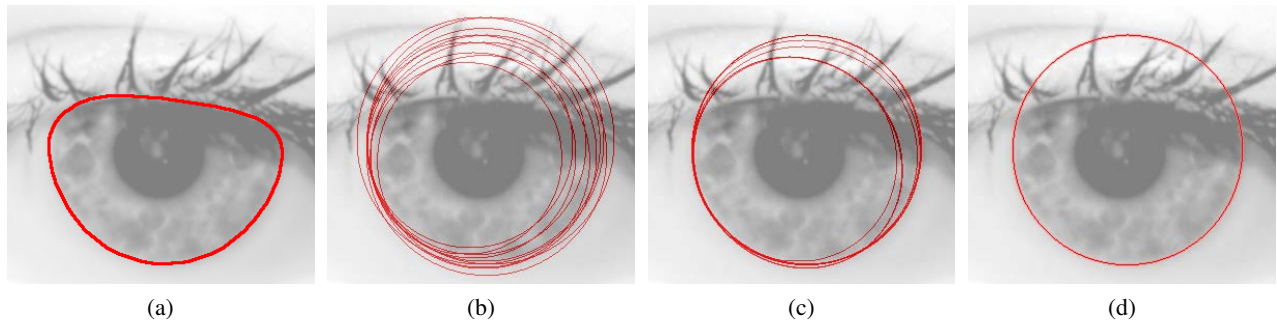


Figure 50: Complete Ellipse Fitting Process : the algorithm uses the results of the segmentation (Figure (a)) to generate a set of initial circles (Figure (b)). The EVF framework is used to morph the initial circles into optimized ellipses (Figure (c)). The ellipse that best fits the border of the iris, i.e. the ellipse that has the lowest energy value (Equation (93)) is selected to perform the normalization (Figure (d))

6.3 Evaluation

In this section, we first evaluate our proposed EVF algorithm compared to other boundary fitting methods. We then address some specific issues : evaluation on off-angle images and computation times.

6.3.1 Global Performance evaluation

We evaluate the influence of the contour fitting algorithm on our complete recognition system on the databases ICE2005, ND-Iris and Casia-Lamp using the following contour fitting algorithms :

EVF Framework from Section 6.1.3, as described in Section 6.2 taking 10 circles for the initialization.

CVF Framework from Section 6.1.2, as described in Section 6.2 taking 10 circles for the initialization.

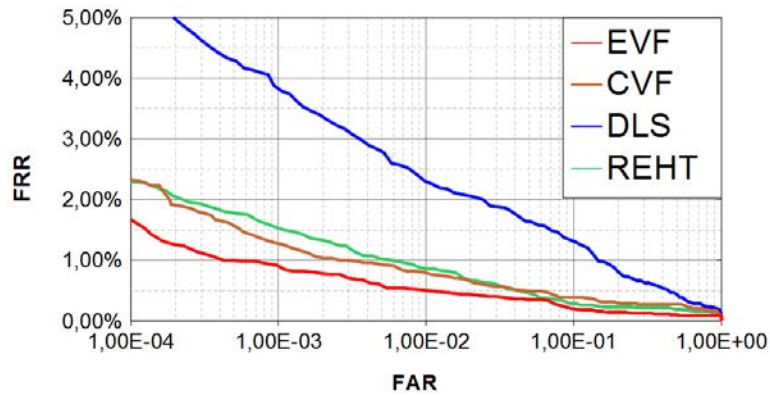
DLS Direct Least Square (DLS) algorithm presented in [22] because it is the most common way to find ellipses in the iris literature.

REHT Random Elliptic Hough Transform [45] (REHT) as an extension of the Circular Hough Transform. Though faster than the standard Elliptic Hough Transform, this algorithm is by far the slowest algorithm considered in this evaluation as we will discuss in the following.

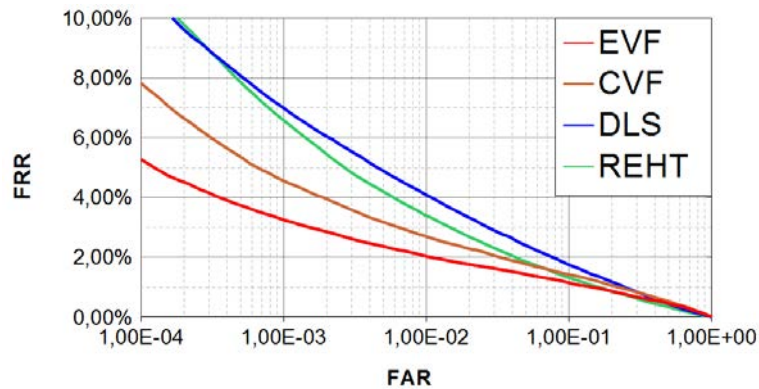
Figures 51a, 51b and 51c display the ROC curves corresponding to the four algorithms on the three databases. EVF performs the best on the three databases followed by CVF. REHT performs better than DLS on ICE2005 and ND-Iris but DLS performs better than REHT on CASIA-Lamp. EVF and CVF perform best because they are very robust : big fitting mistakes are mainly due to important segmentation failures. On the contrary, DLS and REHT apart from giving bad estimations in the case of big segmentation errors, also fail on some images where the segmentation is just slightly inaccurate, explaining the gap in the global system's recognition performance.

We now highlight specific behaviors of the four algorithms on some examples displayed in Figure 52. DLS performs well when the iris borders are clearly visible (Figures 52f, 52n) but when failures occur, the resulting ellipse is far from actual iris borders (Figures 52b, 52j, 52r). These errors are mainly due to segmentation inaccuracies (skin for Figure 52r and right limbus in Figure 52j) and so, the only way to prevent these errors is to adjust the segmentation stage.

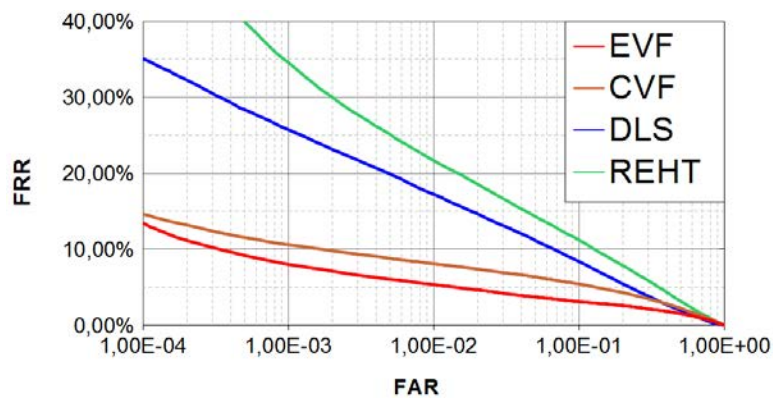
REHT is more robust than elliptic DLS but inaccurate fittings still occur (Figures 52q, 52i). Let us note here that this algorithm performs the best on the off-angle image (Figure 52a). We will discuss this statement in the following section.



(a) ICE2005



(b) ND-Iris



(c) CASIA-Lamp

Figure 51: ROC curves of the experiments for the algorithms Elliptic Variational Fitting (EVF), Circular Variational Fitting (CVF), Elliptic Direct Least Square (DLS) and Random Elliptic Hough Transform (REHT) on the databases ICE2005, ND-Iris and CASIA-Lamp

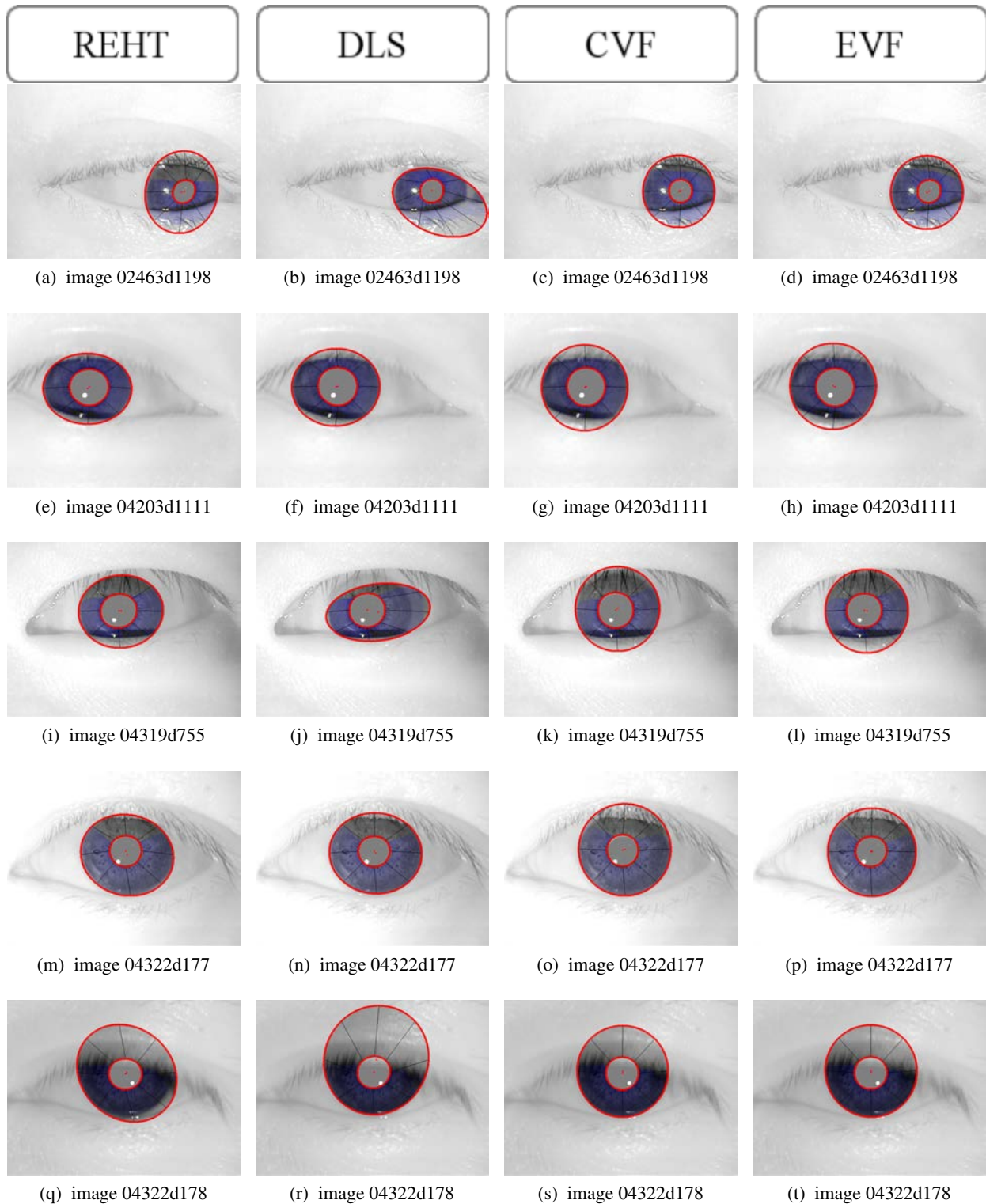


Figure 52: This Figure shows some fitting results for the algorithms we considered in Section 6.3.1. The first column (Figures (a) (e) (i) (m) (q)) corresponds to the results of the REHT algorithm. The second column (Figures (b) (f) (j) (n) (r)) corresponds to the results of the DLS algorithm. The third column (Figures (c) (g) (k) (o) (s)) corresponds to the results of the CVF algorithm. The last column (Figures (d) (h) (l) (p) (t)) corresponds to the results of the EVF algorithm.

EVF gives very consistent results on all the images of Figure 52. Figure 52h is especially interesting because elliptic DLS and REHT tend to flatten the iris contour because of occlusions but EVF stays coherent with the shape we assume behind the eyelids. Thanks to the regularization terms E_s (equations (115 of Section 6.1.3) and (116)) some extra information is added about the expected shape of the borders behind occlusions. However, this assumption can lead to some inaccuracies as in Figure 52p where the upper and right part of the ellipse do not exactly fit the outer iris borders. This is because we used a fixed setting for the constraint terms γ_p and γ_i . We expect to correct this behavior by adapting this term to the image : on images of good quality with visible iris borders, shape assumptions should be low, but for images of bad quality with unclear borders or heavy occlusions, the shape assumptions should be high.

It is interesting to note that CVF performs better than REHT and DLS. This is because our variational framework exposed in Section 6.1 is very robust. Figure 52 illustrates the reason for this statement. For the considered images, none of the CVF-based circles are exactly on the iris borders, but they are all close to the optimal fitting. A conclusion from this experiment is that it is better to have an efficient circular scheme for normalization than a non-optimal elliptic one.

6.3.2 Off-angle correction evaluation

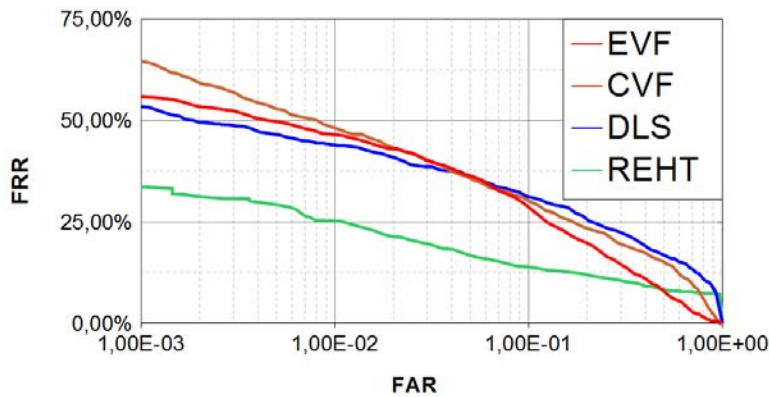


Figure 53: ROC curves of off-angle images of ND-Iris

In this section, we evaluate the performance on off-angle images. In this experiment, we selected a list of off-angle images from the ND-Iris database. The selection was made using the criterion described in [40] and we manually removed falsely selected images. This process gives a sub-list of 50 images. For the evaluation, we kept all comparisons of the previous experiment where at least one of the off-angle images was involved. It generated a sub-list of 800 intra-class comparisons and 102,200 inter-class comparisons. This number of comparisons is far from being sufficient for an accurate evaluation, but as we can see in Figure 53, it is sufficient to get a ranking of the four algorithms considered.

As expected, CVF performs the worst on this subset. The standard EVF performs as well as DLS. Finally, REHT performs the best on this database. The ranking of the algorithms is different in this experiment compared to the whole ND-Iris database because the precision of contour fitting is even more critical for off-angle images than for frontal images.

REHT may lack robustness as discussed above, but when the fitting goes well, it achieves a very good precision as shown in Figure 52a. EVF is more robust, but the fitting might not be perfect for images with strong gaze deviation because of the regularization term E_s . We have discussed this issue above, but we expect to correct this behavior by adapting the regularization strength to the image properties. For instance, by using the criterion [40] which we used to create this image subset as an indicator of gaze deviation, we could

automatically adapt the constraint terms γ_p and γ_i . When the criterion indicates an off-angle image, constraints should be lowered.

Let us note here that a common claim in iris literature [17, 63, 10] is to assume that an elliptic shape for iris contours is a consistent approach to correct gaze deviation. DLS and EVF which perform worse than CVF on the complete database perform better on the off-angle subset. This result confirms that elliptic models are better than circular ones for off-angle images. However, the recognition performance is still very low. It seems that this problem still needs further studies to be extended beyond the elliptic model.

6.3.3 Computation Time

| | Initialization | Contour Fitting | Selection | Total |
|---------------------------------|-------------------------|----------------------------|--------------------------|----------------------------|
| DLS | — | $2 \pm 0.8\text{ms}$ | — | $2 \pm 0.8\text{ms}$ |
| REHT | — | $30223 \pm 22294\text{ms}$ | — | $30223 \pm 22294\text{ms}$ |
| CVF 1 contour (40 iterations) | $139 \pm 46 \text{ ms}$ | $15 \pm 3 \text{ ms}$ | $0.8 \pm 0.2 \text{ ms}$ | $185 \pm 49 \text{ ms}$ |
| CVF 10 contours (40 iterations) | $174 \pm 66 \text{ ms}$ | $198 \pm 34 \text{ ms}$ | $0.8 \pm 0.2 \text{ ms}$ | $372 \pm 100 \text{ ms}$ |
| EVF 1 contour (40 iterations) | $139 \pm 46 \text{ ms}$ | $21 \pm 4 \text{ ms}$ | $0.8 \pm 0.2 \text{ ms}$ | $160 \pm 67 \text{ ms}$ |
| EVF 10 contours (40 iterations) | $174 \pm 66 \text{ ms}$ | $257 \pm 37 \text{ ms}$ | $0.8 \pm 0.2 \text{ ms}$ | $431 \pm 103 \text{ ms}$ |

Table 5: Computation time for different contour fitting scenarios

Table 5 gives the average time for each algorithm used in this evaluation. The algorithms are implemented in C++ without particular optimizations. The evaluation was done on a computer equipped with a 2.4 GHz Intel processor running under a 32bit operating system.

The first statement is that DLS fitting clearly outperforms the other algorithms in term of computation time, explaining its popularity in iris recognition literature [10, 62, 3].

On the other hand, REHT has a huge computation time. As explained earlier, the purpose of this implementation is only to get an extension of what would have been the classical Circular Hough Transform of the literature when transposed to ellipses. Computation time displayed here clearly shows that this algorithm is by no mean suitable for real-time applications.

It is also interesting to note that both the initialization of EVF/CVF (based on a Circular Hough Transform) and REHT have very high variability in computation time. This is because the complexity of those algorithms is proportional to the contour length : the complexity is proportional to the power of three of the contours length for the Circular Hough Transform and to the power of five for the REHT.

Our novel EVF algorithm performs in-between the two other algorithms. We performed 40 iterations for the iterative scheme of Equation (94) and the initial Circular Hough Transform is down-sampled by a factor of 3. Fitting of a single contour is 15 ms for a circle and 21 ms for an ellipse. This difference comes from the reduction from 5 equations for the ellipse to three for the circle (radius $r = (a + b)/2$, and $\phi = 0$). If we look at the complete system with 10 contours evaluated in Section 6.3.1, the computation time is 372 ms for circles and 431 ms for ellipses. We can note that roughly 40% of the computation time is spent in the initialization (Circular Hough Transform). It is possible to reduce this initialization time using the fast rough location described in [3] or using DLS.

6.4 Conclusion

In this chapter, we have proposed a precise and effective way to estimate iris contours by mean of ellipses for iris normalization, the EVF framework. This algorithm relies on the classical Active Contours approach leading to a compact formulation for ellipses. We have compared our algorithm to classical Direct Least Square fitting for ellipses and to a Randomized Elliptic Hough Transform and have shown that our algorithm performs better on several reference databases when integrated in a complete recognition system. After this global evaluation, we have addressed some specific issues such as evaluation on off-angle images and computation times.

An interesting theoretical aspect of our model is that it can be easily extended to use other region competition terms like the texture descriptor that we described in Section 5.4 of Chapter 5. Another theoretical interpretation of our model is to see the evolution as a rigid deformation of an elliptic template. A perspective to extend our model is to consider non-rigid deformations to better adapt the template shape to the actual border of the iris. This extension should not reduce the robustness of the process but should help characterizing more precisely pupils or irises which are anatomically non elliptic.

The estimation of normalization contours of the iris is the last critical stage of our iris segmentation stream. At this stage, detecting possible segmentation failures is of main importance for the complete recognition system that we consider in this thesis. This will be the aim of the next chapter.

Chapter 7

Image Quality and Automatic Detection of Segmentation Failures

Recognition performance of iris recognition systems is highly related to input data quality. For this reason, evaluations of iris image quality have received a significant interest over the last years. One of the most significant illustrations of this trend is the IREXII-IQCE challenge [71] which gathered most of the industrial and academic actors of the iris recognition community, to study the impact of image quality on iris recognition systems. However, most of the quality metrics of the literature make direct use of segmentation results without further studying the segmentation quality itself.

Our contribution in this thesis is to propose a module detecting segmentation failures, i.e. large errors in the location of the normalization contours or a very bad segmentation mask. To build this module, we exploit the idea that correctly segmented images may or may not have good recognition performance depending on the quality of their information content, but images for which segmentation did fail, have always very poor recognition performance. Accordingly, we propose in this chapter a set of metrics designed to predict an image's False Non Match Rate introduced in Chapter 2. Let us note that these metrics are built to detect segmentation failures so they are not able to distinguish two "correct" segmentation results.

In order to build these metrics, we use an intermediate Rough Region Segmentation (RRS) of the image. This RRS is generated using simple and robust algorithms exploiting the pupil and iris location provided by the fine segmentation under evaluation. We use this intermediate RRS to build two kinds of metrics. The first type of metrics which we call *Relative Metrics* assumes that the RRS is a reference ground truth. In other words, these metrics are based on the difference between the input segmentation and the RRS. The second type of metrics which we call *Empirical Metrics* evaluates the quality of the segmentation of the RRS image with respect to some prior knowledge on what a good segmentation is.

Computing all these quality metrics results in a vector of individual quality measures. However, in order to decide if the segmentation is valid or not, it is necessary to fuse these individual metrics in order to generate a final segmentation quality score. We propose to fuse these metrics by predicting the image's False Non Match Rate of the segmented image. To solve this problem, we train an ϵ -Support Vector Regression (ϵ -SVR) algorithm and show that the output of the ϵ -SVR is coherent with the expected behavior of a meaningful quality measure.

We start this chapter by a brief overview of what iris data quality is (Section 7.1) and then follow the process described in Figure 54. In Section 7.2, we detail how we generate the RRS. In Section 7.3, we explain how we build the quality metrics used to detect segmentation failures. In Section 7.4 we merge the individual quality metrics to produce the final quality score. Finally, in Section 7.5, we make some comparative evaluation for several segmentation algorithms and databases.

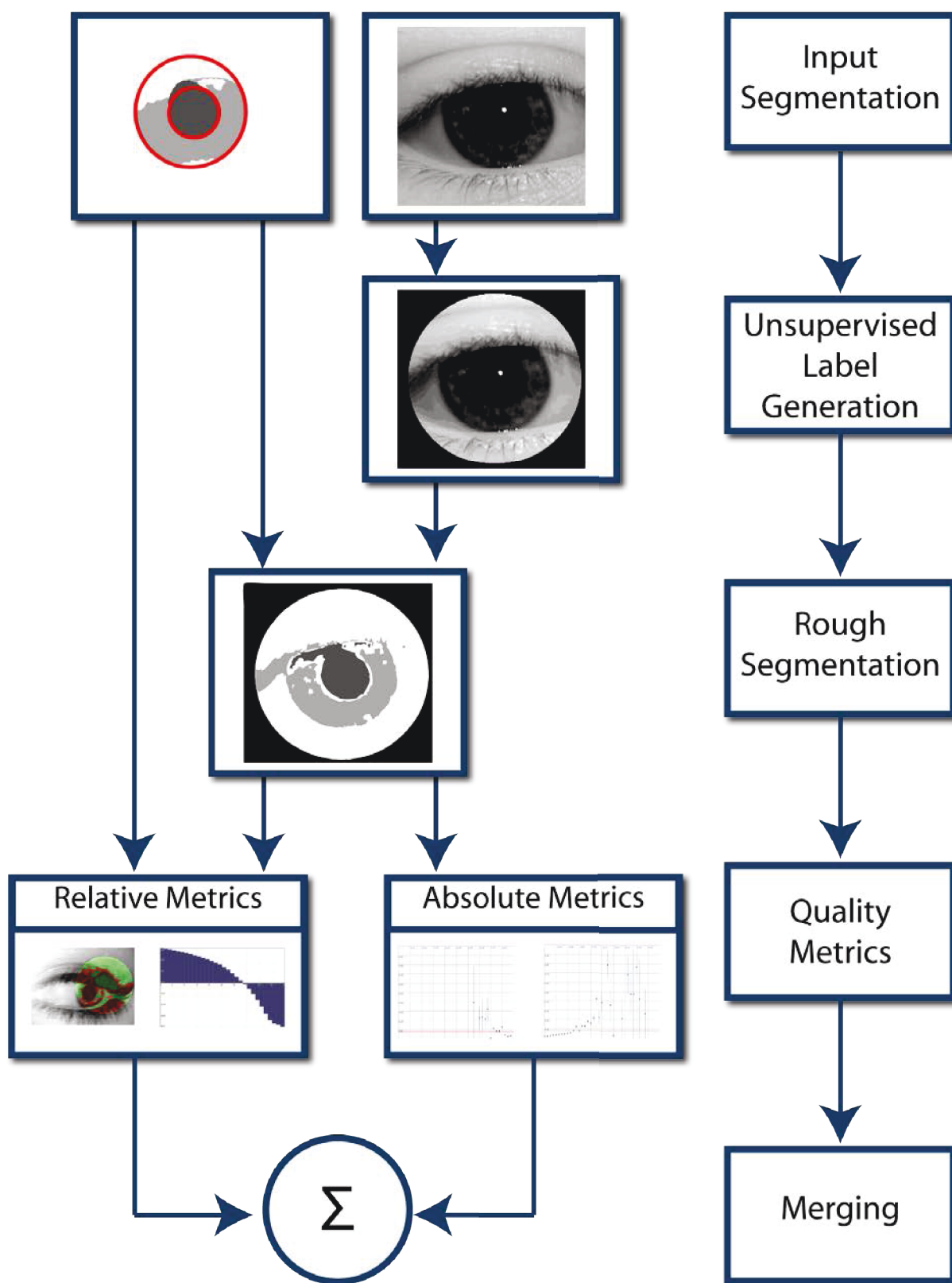


Figure 54: Main steps of the segmentation evaluation process. A rough unsupervised labeling is applied to the image based on Finite Gaussian Mixture Models. Then labels are merged using the input segmentation information in order to generate the intermediate Rough Region Segmentation (RRS). RRS is then used to generate a set of quality metrics which are either *Relative Metrics* or *Absolute Metrics*. Finally, results are merged to generate a final segmentation quality score.

7.1 What Is Iris Data Quality?

The quality of a biometric sample is correlated to its recognition performance in a specific system. Indeed, degradations in the input image quality tend to make the recognition harder, leading to a decrease of the system's performance. Degradations in iris images can be either blur, illumination inhomogeneity, strong occlusion of the iris texture, low contrast along the iris borders, etc. (See Section 1.6 of Chapter 1 where we describe such degradations).

Classical iris quality evaluation algorithms usually assume that the segmentation is known in order to evaluate the quality of the image. For instance, evaluating the contrast between the pupil and the iris requires the location of the border between the two regions. However, such computation becomes inconsistent in case of large failures of the segmentation process. Therefore, we propose to detect such large failures in order to ensure the consistency of traditional iris quality algorithms.

In this section, we first describe usual iris quality evaluation methods. Then, we make a brief description of segmentation evaluation principles. Finally, we describe the few articles in the literature which are specifically devoted to iris segmentation quality.

7.1.1 Traditional Iris Quality

We start this section by describing one of the most significant pieces of work of the community on the subject: the IREXII-IQCE challenge. This challenge aims at defining some general quality metrics that should be implemented in iris recognition systems in order to ensure correct data quality. Then we describe some more specific articles of the literature. Our aim is not to give a complete state of the art of iris quality evaluations, but only to highlight main trends that motivate our further work.

7.1.1.1 Iris Quality Calibration and Evaluation Challenge

The IREXII-IQCE challenge [71] (Iris Quality Calibration and Evaluation) was organized by the NIST in 2010 and gathered the main industrial and academic actors of the iris recognition community. Each participant was asked to give an implementation of 14 quality metrics expected to describe the quality of an iris image. The challenge consists in an objective evaluation of the most consistent metrics, independently of the implementation and the database. Among the 14 considered metrics, the most significant ones are pupil/iris contrast, pupil shape, iris/sclera contrast, gaze angle and sharpness.

The IQCE evaluation created an awareness about the importance of iris quality assessment for operational systems. Moreover, conclusions of this evaluation are relevant regardless of the system used for recognition. However, a drawback of this evaluation is that most competitors were industrial companies which did not give access to their quality metrics implementation.

7.1.1.2 Some Reference Work on the Iris Quality

Quality metrics can be divided into two categories: global metrics that evaluate the quality at an image level and local metrics that evaluate the quality at a pixel level. Local quality metrics are important for feature extraction and recognition algorithms in order to weight good quality regions more than poor quality ones. On the other hand, global quality metrics evaluate whether the overall image quality is sufficient for recognition.

In [31], Kalka et al. give a good overview of most quality metrics that were available before the IQCE challenge. The authors propose to evaluate the impact of defocus blur, off-angle, occlusion/specular reflections, lighting and iris resolution. Then, they build the associated quality metrics using the segmentation results of their algorithm described in [83]. Finally, they merge these metrics using Dempster-Shafer criterion. They show that the final quality score is correlated to the recognition performance of their system. However, they also outlined in their discussion that the proposed metrics require correct segmentation results.

In [59], Proenca proposes to extend most existing metrics for NIR imaging to Visible Wavelength. Accordingly, he gives a good state of the art of quality assessment methods in iris recognition. The metrics he considered can be applied either on the raw image, the segmented image or the normalized image. Among the 22 cited articles, 16 correspond to the normalized or segmented images. Then, Proenca describes quality metrics for measuring focus, motion blur, occlusions, iris pigmentation, pixel count (area) and pupillary dilation. Interestingly, the author also highlights that the considered quality metrics are dependent on the segmentation results.

To conclude this section, we note that most quality metrics of the literature are built using the segmentation results. Therefore, they are no longer consistent when the segmentation process fails. Our proposal in this chapter differs from the classical quality approaches because we propose to detect large segmentation failures instead of doing a fine evaluation of data quality.

7.1.2 Segmentation Quality

The evaluation of segmentation quality is a difficult problem because it is very hard to assess what a good segmentation is. In [79], Zhang et al. propose a very complete survey of segmentation quality evaluation where they attempt to answer this question. Although dedicated to unsupervised segmentation metrics, their work gives a very good description of the different possible definitions of what can be called a good segmentation. The authors identify 5 corresponding evaluation methods :

Subjective Evaluation The quality of a segmentation is judged by a human evaluator.

System-level Evaluation Evaluation is done by examining the impact of different segmentation methods on a specific system.

Analytical Evaluation Evaluation of the structure and the complexity of the segmentation algorithms almost independently of its output.

Supervised Evaluation Evaluation assuming the presence of a ground truth. The quality is based on the difference to the ground truth.

Unsupervised Evaluation Evaluation of segmentation result's properties (for instance the number of regions, the regions homogeneity, the shape of the regions...).

As iris segmentation is naturally embedded in a complete recognition system, evaluating an iris segmentation by looking at its impact on the complete recognition system is a natural evaluation method, i.e. performing a *System-level Evaluation*. We have discussed the way such an evaluation can be performed in Section 2.2 of Chapter 2, but in the present chapter, our purpose is to create metrics that can evaluate the iris segmentation quality of a single image in order to detect segmentation failures.

Supervised Evaluation are strongly limited by the requirement of a ground truth. Generation of such a ground truth is usually done manually by a human operator. Therefore, such a generation is impossible for operational applications. However, *Supervised Evaluation* methods can be very effective for benchmarking, like for instance the competition NICEII [60] which provides a ground truth for every image.

Unsupervised Methods are based on prior assumptions on what a good segmentation should be and check to what extent the input segmentation matches the assumptions. Without further specifications, these assumptions are usually based on the homogeneity of the segmented regions. From our first experiments, we find out that the homogeneity criterion is not suitable to evaluate the correctness of an iris segmentation because iris texture may be highly textured and yet, correctly segmented. Therefore, in the following we propose more complex quality metrics that are more specific to the iris segmentation problem.

7.1.3 Iris Segmentation Quality

As explained in the introduction, the estimation of the segmentation quality in an iris recognition system has been little studied so far. To the best of our knowledge, only five articles have studied this problem.

The earliest paper trying to predict the segmentation failures in iris recognition systems is the thesis dissertation of Lee [38] from West Virginia University. In this thesis, the author computes a set of image quality metrics based on regions' homogeneity and heterogeneity. These metrics are combined using PCA to compute a final score predicting segmentation failures. Thesis [38] is interesting because it is the first study of the iris segmentation evaluation problem. However, the metrics used to predict segmentation failures are too simple because they are only based on regions homogeneity and heterogeneity and do not take into account the specificities of the iris segmentation task. Moreover, as we mentioned in the previous section, our preliminary experiments indicate that region homogeneity does not seem pertinent to describe iris segmentation quality. Indeed, the iris region can be more textured, and therefore less homogenous, than other regions in the eye.

In [?] Zuo et al. first show the impact of the degradation on the reference segmentation algorithms and then propose to evaluate a set of empirical assumptions about the expected properties of a "correct" segmentation. First the pupil should be of sufficient size. Then the cumulative gradient along the pupil and iris border should be high. Finally, the pupil should have a lower intensity than the iris, and the iris a lower intensity than the sclera.

In [81], Zhou et al. propose to add several quality modules to existing iris recognition systems in order to make them suitable for non ideal iris recognition. One of the quality modules is dedicated to the evaluation of iris segmentation quality. This module takes as input the segmentation mask and the two normalization contours of the pupil and the iris. The module first re-segment the pupil using the pupil contour. If the input pupil segmentation and the re-segmented pupil region overlap, the pupil is assumed to be correctly segmented. Then, the quality of the iris segmentation is evaluated by assessing the homogeneity of concentric rings inside and outside the iris contour. Finally the two scores are fused to generate a final segmentation quality score.

The article of Kalka et al. [30] was our main inspiration for the use of an intermediate RRS to evaluate segmentation quality. In this article, the authors propose to build an intermediate segmentation using the input segmentation and probability inferences. Then, they build a metric based on the mis-classified pixels with respect to their intermediate segmentation. An interesting aspect of their metric is that they evaluate the contours used for normalization (circles in the article), which is critical for correct recognition. However, from our point of view, their intermediate segmentation is too simple because it is only based on the histogram information. Indeed, this histogram may be incoherent if the segmentation has strongly failed. This statement motivated us for developing a more robust system for this intermediate segmentation result we describe in Section 7.2.

In [80], Zhang et al. train a classifier for identifying mis-localized iris segmentations. The idea is to divide the image into 12 sub-regions and to learn the average curvature of the image manifold for a correctly segmented image in each sub-regions. From our point of view, this approach is very limited because it strongly relies on the learning step of the classifier, and may change if the sensor or acquisition conditions change.

7.2 Rough Region Segmentation

As explained previously, simple quality metrics based on region homogeneity are not sufficient to quantify the correctness of an iris segmentation. On the other hand, manual generation of a segmentation ground truth is impossible for operational applications. Therefore, our proposal is to automatically generate a Rough Region Segmentation that can be interpreted as a ground truth for evaluation. We compute these metrics on the RRS because the RRS generation tends to enhance failures of the segmentation, making their detection easier.

Such approach is close to the idea of Kalka et. al [30] where they build a RRS based on the histogram of the iris region. However, from our experience, a histogram is not a robust enough feature for building such a RRS. Indeed, it may be largely corrupted in case of large failure of the segmentation. Accordingly, we propose a robust way to build such a RRS. Our process is twofold :

- We propose to start by an unsupervised labeling of the image based on a Gaussian Mixture Model. The idea is to learn on the image 16 Gaussian components. Each pixel of the image will be associated to one of such components, namely to run "unsupervised label" generation (Section 7.2.1). The idea is that describing the regions using a few Gaussians is actually more robust than using directly the intensities.
- Finally, we "merge" the unsupervised labels into three classes : pupil (label L_p), iris (label L_i) or background (label L_b). To that aim we use the spacial information provided by the input segmentation we want to evaluate (Section 7.2.2).

The main idea of this process is that if the segmentation is correct, the resulting RRS should be very close to the input segmentation, but if the segmentation has failed, the RRS is expected to be very different as we will see in the following.

7.2.1 Unsupervised Labels Generation

The first step of our process is to generate an unsupervised set of labels to represent image intensities. To that aim, we use the spatially variant Gaussian Mixture Model (GMM) algorithm introduced in [65] with 16 labels. We set the number of labels empirically as a tradeoff between computation time and the fidelity to the image (the more the labels, the better the fidelity is, but also the slower the generation is). This algorithm belongs to the field of finite mixture models which assume that each pixel in the observed image is a sample of a finite mixture distribution. Therefore, estimating the underlying distribution mixture allows to associate a label to every pixel in the image (the distribution having the largest contribution in the pixel's representation).

Figure 55 illustrates the evolution of the algorithm. The 16 Gaussian distributions are initialized with similar variance and with their means covering the complete intensity range. After a few iterations, the Gaussian distributions start to catch some specific properties of the image.

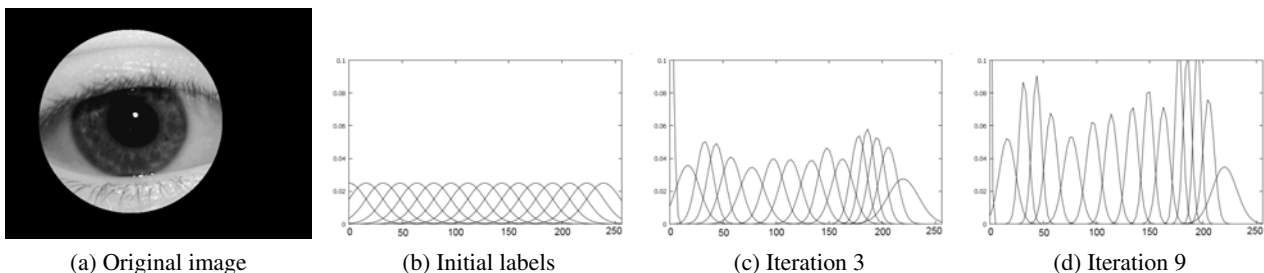


Figure 55: Evolution of the Gaussian mixture.

The interesting aspect of this unsupervised labeling is that every region in the image is represented using a few Gaussians. Accordingly, in the next section we identify the regions that should belong to the iris and the pupil regions.

7.2.2 From Unsupervised Labels to Iris Regions

At this point, our goal is to merge the unsupervised labels generated in the previous section into one of the classes : pupil labels L_p , the iris labels L_i and the background labels L_b . To that aim, we use the input segmentation results.

Our assumption is that labels associated to the pupil should cover at least 80% of the pupil segmentation mask. Remaining labels associated to the iris should cover 80% of the iris mask. This idea is that the resulting masks of L_i and L_p will contain all the objects in the image overlapped by the segmentation masks of the pupil and the iris. The background labels L_b are all the unaffected unsupervised labels. Figures 56 and 57 illustrate the generation of L_i for a correctly and wrongly segmented image.

- In Figure 56, the Gaussians covering the largest portion of the iris segmentation masks are progressively added until L_i covers 80% of the iris segmentation mask. As the iris segmentation mask is correct, the final L_i (Figure 56h) and the segmentation mask (Figure 56i) are very alike.
- In Figure 57, the localization of the pupil has failed, leading to a complete failure of the segmentation process. Accordingly, the segmentation mask only covers a part of the iris region and mostly eyelashes. As the iris segmentation mask covers mostly non-iris region, the first two labels added to L_i correspond to eyelashes and shadows. However, the two last Gaussians actually correspond to the iris region. Therefore, the final label mask L_i (Figure 57j) is actually closer to a "good" segmentation mask than the input segmentation mask (Figure 57k). Such a behavior could not always be achieved, but in any case the label mask L_i and the input segmentation mask will be very different for a poorly segmented image.

An advantage of this label merging process is that it merges all the unsupervised labels covered by the segmentation mask. If the segmentation mask covers only one region, then the associated merged label will be very close to the segmentation mask. However, if the segmentation mask relies on several regions, the merging process will catch labels associated to all the regions. Therefore, the resulting merged label will be very different from the original segmentation mask. Formally, if we note U the set of unsupervised labels of Section 7.2.1 the algorithm for the generation of L_p , L_i and L_b is :

Data: U the set of unsupervised labels

$$L_p = L_i = L_b = \emptyset$$

while L_p does not cover at least 80% of the pupil mask **do**

Select the label $U_{Pmax} \in U$ covering the largest part of the pupil segmentation mask.
 add U_{Pmax} to L_p
 remove U_{Pmax} from U

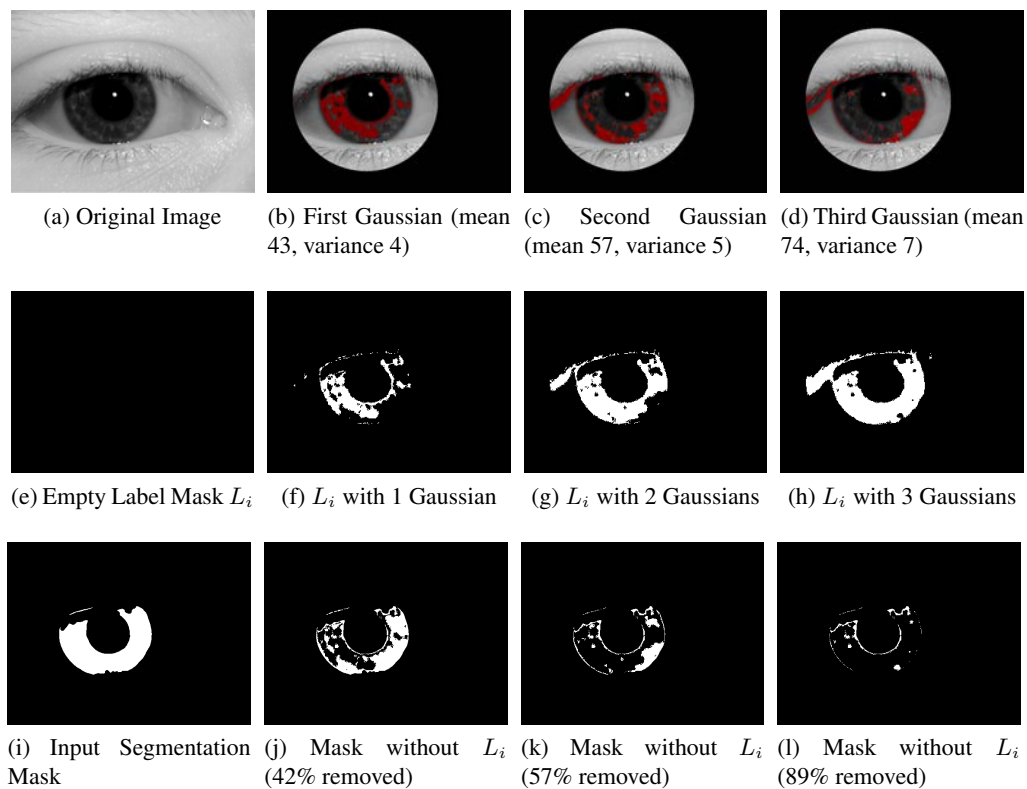
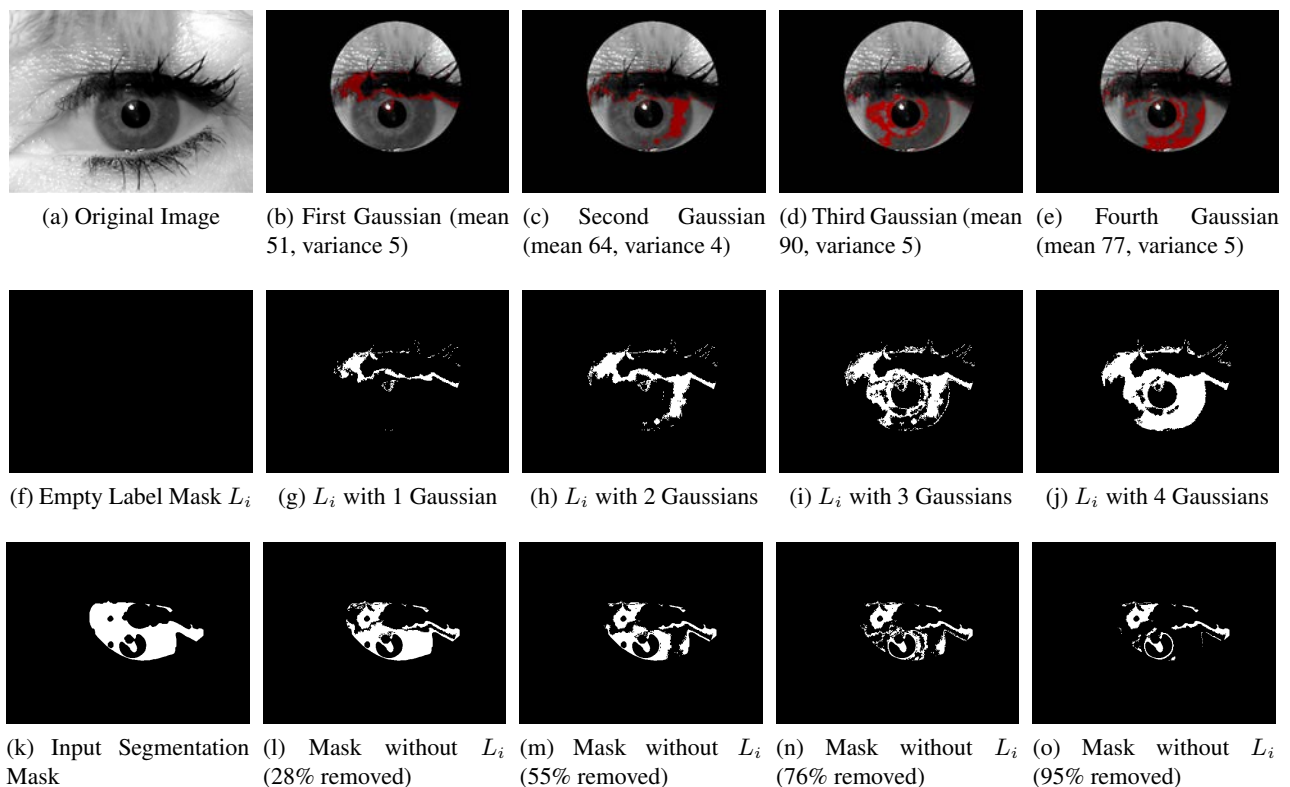
while L_i does not cover at least 80% of the iris mask **do**

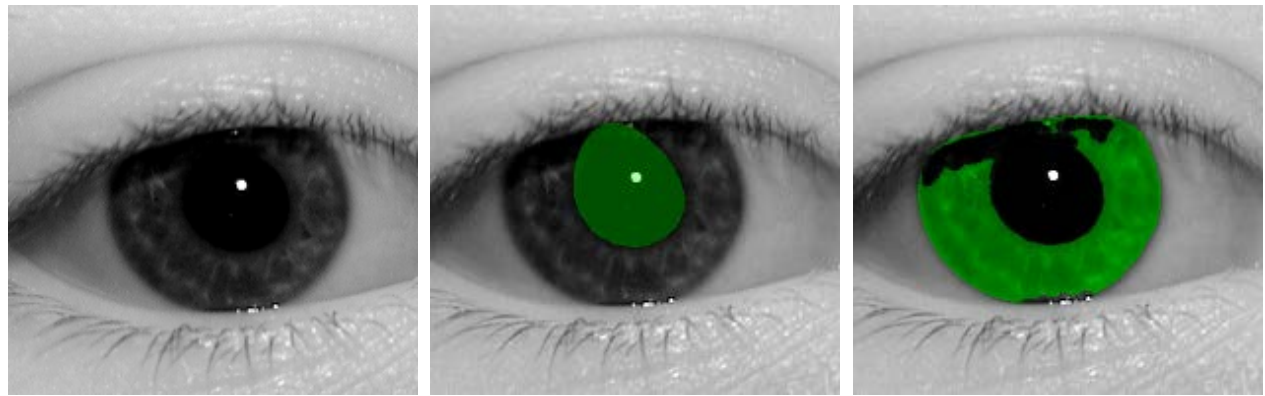
Select the label $U_{Imax} \in U$ covering the largest part of the iris segmentation mask.
 add U_{Imax} to L_i
 remove U_{Imax} from U

$$L_b = U$$

Algorithm 1: Generation of the pupil labels L_p , the iris labels L_i and the background labels L_b

Figure 58 illustrates the process of Algorithm 1 for a correctly segmented image. As the pupil segmentation mask is covered mainly (over 80%) by one label, this single label is associated to the pupil label L_p . The iris is quite textured, therefore 4 unsupervised labels are required to cover the iris segmentation mask. These 4 labels are associated to the iris label L_i . The remaining labels are associated to the background label L_b .

Figure 56: Generation of L_i for a correctly segmented image.Figure 57: Generation of L_i for an incorrectly segmented image.



(a) Original Image

(b) Pupil Mask

(c) Iris Mask

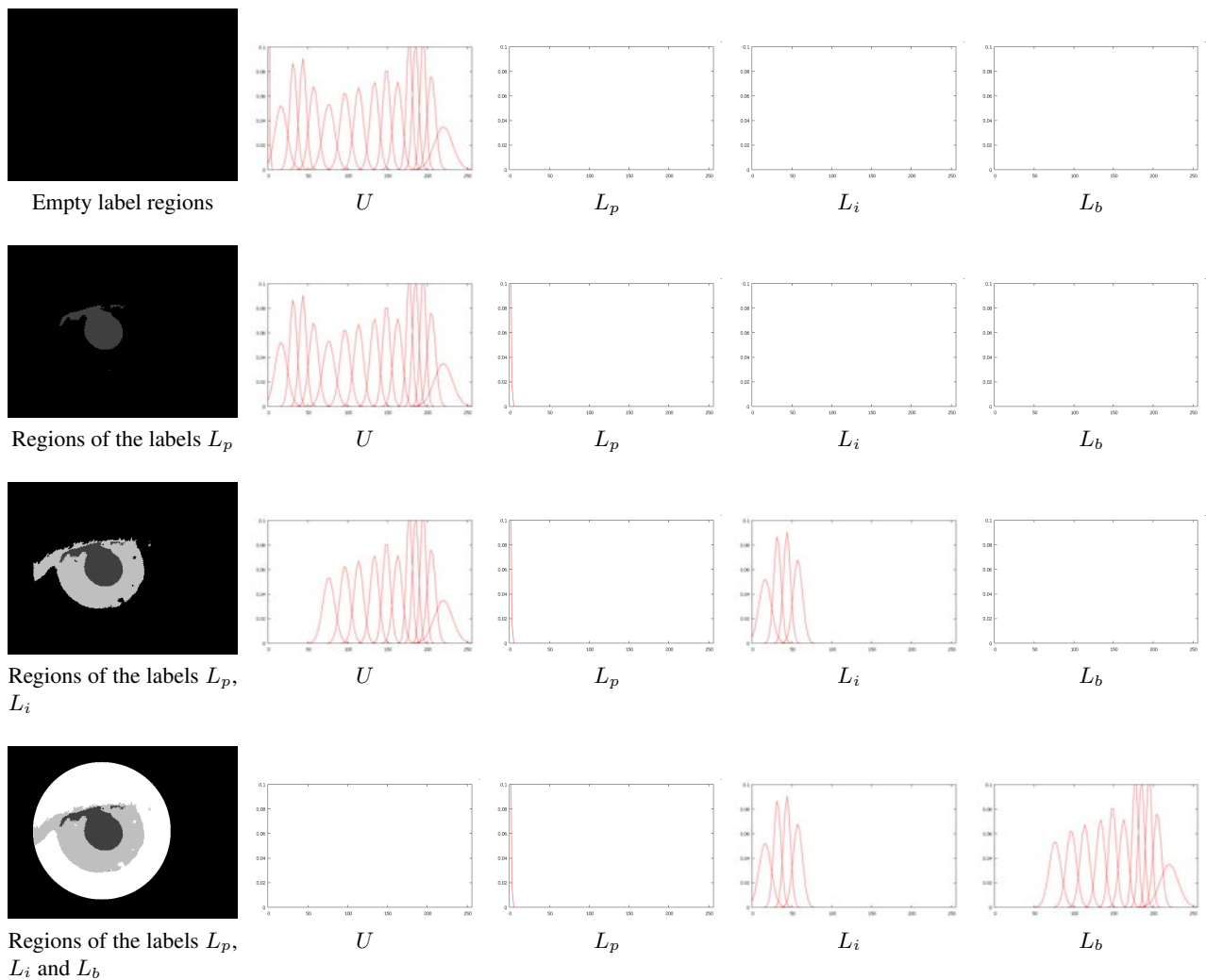


Figure 58: Construction of the labels L_p, L_i and L_b using the set of unsupervised labels U . The first row displays the original image, the pupil segmentation mask and the iris segmentation mask. The second row shows the initial situation : all the Gaussians are in the unsupervised set U and L_p, L_i and L_b are empty. The third row illustrates the association of one Gaussian to L_p . This Gaussian is chosen such that the labels in L_p covers 80% of the pupil segmentation mask shown in Figure (b). The fourth row illustrates the association of four Gaussians to L_i . These Gaussians are chosen such that the labels in L_i covers at least 80% of the iris segmentation mask shown in Figure (c). The final row shows the association of remaining labels to the background.

7.3 Quality Metrics

We have developed two kinds of quality metrics using the Rough Region Segmentation presented in the previous section. On one hand, we considered metrics comparing the RRS to the input segmentation results. The idea of these metrics is to check for incoherence between the two segmentations. On the other hand, we computed some quality metrics on the RRS to assess for the prior knowledge we have on what a good segmentation is.

As explained in Section 2.2 of Chapter 2, a poorly segmented image looks different from any other image of its own image class, i.e. poorly segmented image have a high iFNMR (Equation 12 of Chapter 2). Therefore, we assess the pertinence of a metric based on its ability to "predict" the iFNMR of an image. To that aim, we generate the iFNMR for 12,872 images of the ND-Iris database. We select these 12,872 images by taking 20 images of each eye of every subject in the database¹. So the iFNMR is computed on the 20 intra-class comparisons involving each image for a threshold of 0.36 for a match/non-match decision. Setting this threshold corresponds to an operating point with a FAR of 7.10^{-5} and a FRR of 0.08 for the system we used. The segmentation is generated using B-snakes for iris segmentation (see Section 5.2.2 of Chapter 5) and our EVF method presented in Chapter 6 for contour fitting.

Figure 59 shows the expected behavior of what we call a "good" quality metric. An increase (or a decrease) of the quality metric implies an increase (or a decrease) of the average iFNMR, i.e the metric is correlated to the iFNMR.

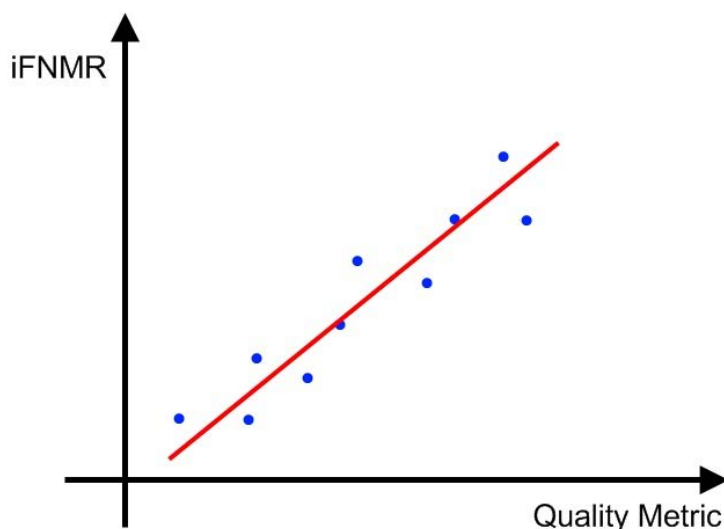


Figure 59: Expected behavior of a good quality metric.

7.3.1 Relative Metrics

The principle of relative metrics is illustrated in Figure 60. The idea is to compare the differences between the original segmentation and the RRS image which corresponds to a novel segmentation of the image. We build two kinds of quality metrics based on this idea. The first metric evaluates the quality of the pupil and iris segmentation masks and the second metric evaluates the quality of the normalization contours' location.

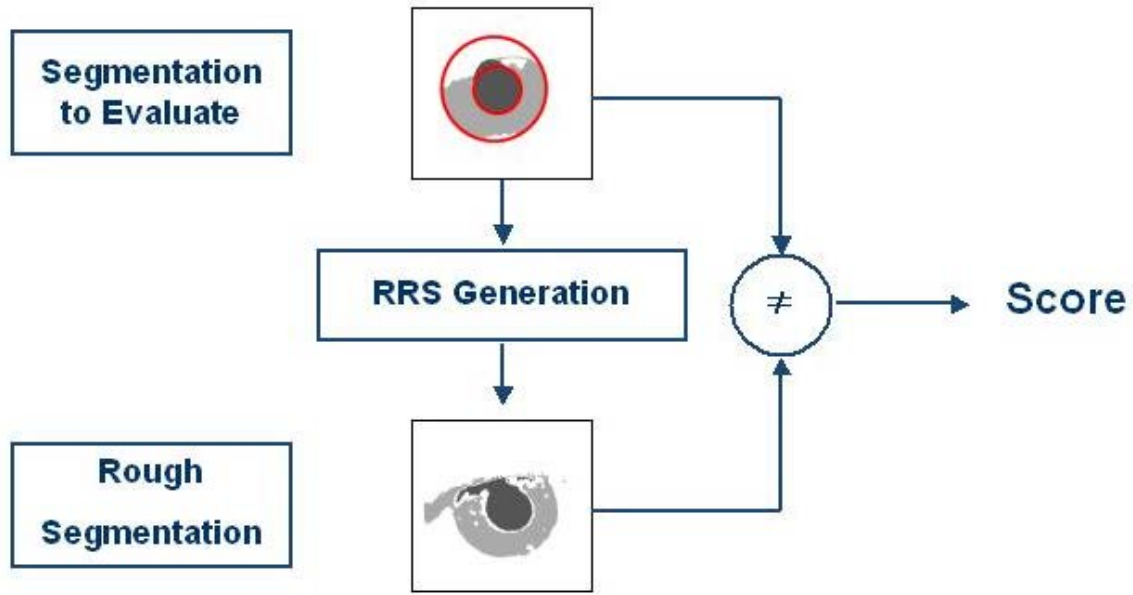


Figure 60: Relative quality metrics.

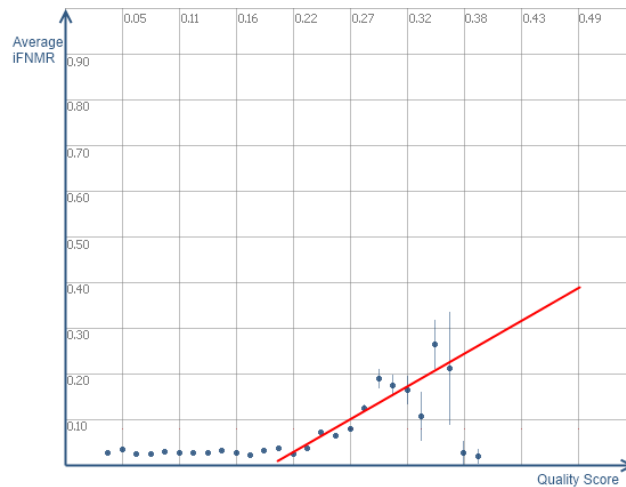


Figure 61: Average iFNMR for $s_{SegDiff}$

7.3.1.1 Differential Metric

This first metric that we propose is directly inspired from supervised evaluation methods ; it simply counts the pixels labeled differently in the two segmentations. Formally, for S the input segmentation with value S_i at pixel location i , R the associated Rough Region Segmentation and n the number of pixels in the image we have :

$$s_{SegDiff} = \frac{1}{n} \sum_{i=1}^n \{S_i \neq R_i\} \tag{119}$$

¹If only less than 20 images were available for a subject, then we took all available images for this subject

Figure 61 displays the average iFNMR of images w.r.t. their quality score $s_{SegDiff}$. First, there is no evolution of the iFNMR when $s_{SegDiff}$ increases from 0 to 0.22. But after 0.22, increasing of $s_{SegDiff}$ imply an increase of the average iFNMR. This means that if the segmentation of an image and its RRS differ from more than 22%, it is more likely that there is a segmentation error.

Figure 62 shows some specific examples of $s_{SegDiff}$ computations. The first row shows an example of a correctly segmented image. The RRS catches most of the iris area and the only difference comes from inaccuracies around region borders and some eyelashes not caught by the RRS generation process. The second row shows an example of a segmentation failure. As the pupil and the iris regions of the input segmentation cover several distinct regions, the RSS generation tends to cover most of these regions, producing a bad misclassification score. The final row shows an image where the RRS is responsible of the bad score value. The large dark area has exactly the same intensities as the pupil, hence the association of the pupil label to this region. However, we assume this last image should not pass traditional image quality checks.

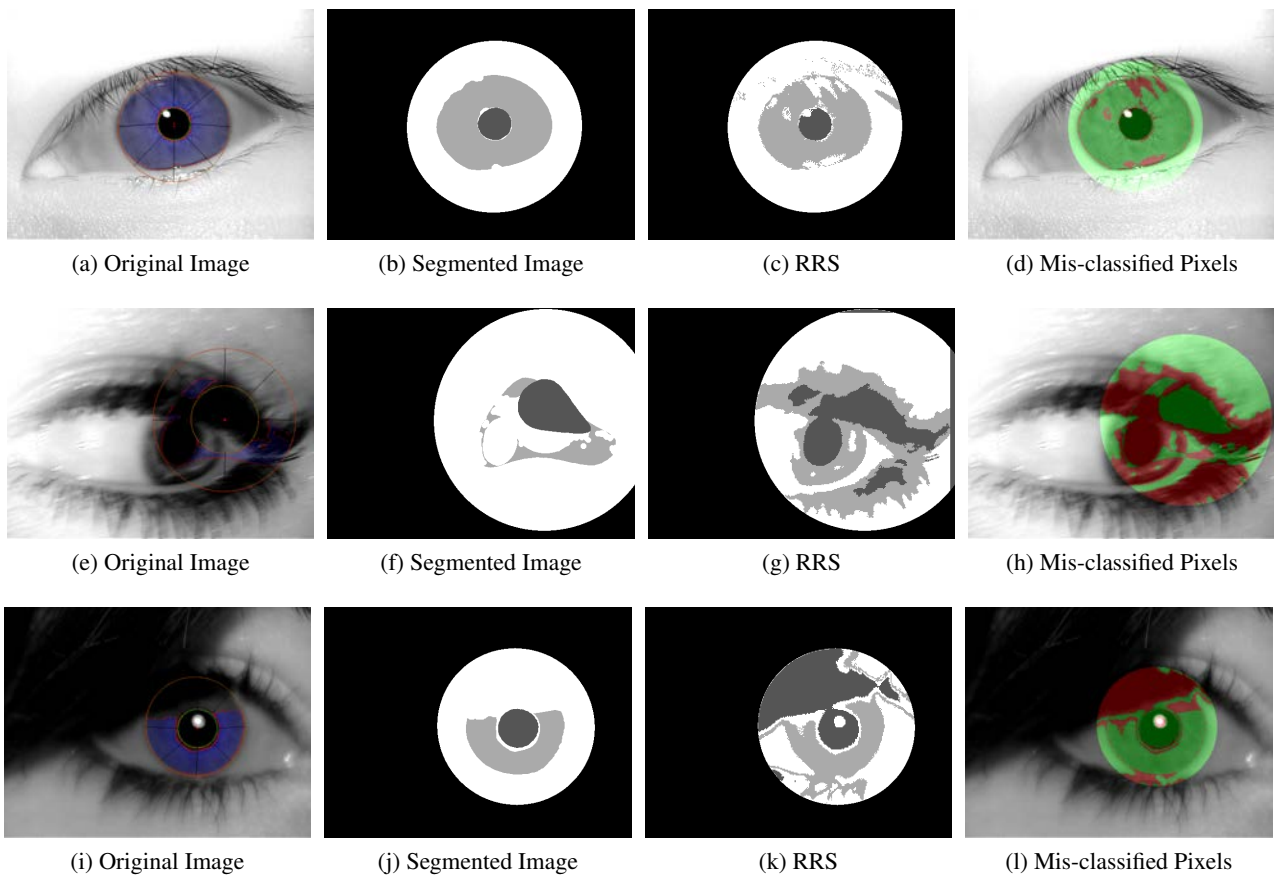


Figure 62: Examples of $s_{SegDiff}$ computations. The first row shows an example of a correct segmentation; we obtain $s_{SegDiff} = 0.1$. The second row shows an example of segmentation failure, we have $s_{SegDiff} = 0.4$. The last row shows an example of an image where the RRS generation is incorrect, we have $s_{SegDiff} = 0.5$.

7.3.1.2 Accuracy of the Contour Location

Precision on the contours used for the normalization step is critical for iris recognition system (as discussed in Chapter 6). We built a set of metrics inspired by the ideas of [81] and [30] in order to evaluate the correctness of these contours location. From [81], we extend the idea of looking for concentric rings inside and outside the contour, but instead of evaluating the regions homogeneity, we propose to count mis-classified pixels w.r.t. the RRS similarly as [30].

We propose to note $R(C_{ref}, t)$ the ring between a contour C_{ref} and the same contour dilated by a factor t , as illustrated in Figure 63. Formally, if we note $d(C_{ref}, t)$ the dilation of C_{ref} by a factor t , we have :

$$R(C_{ref}, t) = \begin{cases} \{\mathbf{x} \in C_{ref}\} \cap \{\mathbf{x} \notin d(C_{ref}, t)\} & \text{if } t \leq 1 \\ \{\mathbf{x} \in d(C_{ref}, t)\} \cap \{\mathbf{x} \notin C_{ref}\} & \text{if } t > 1 \end{cases} \quad (120)$$

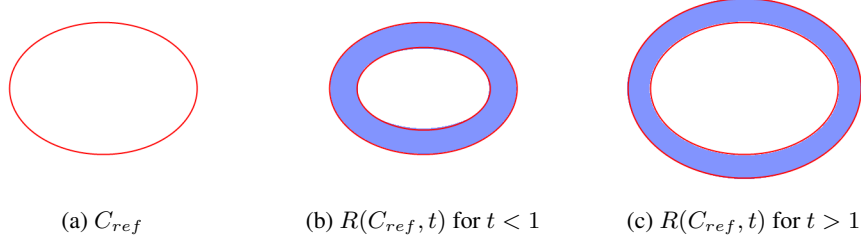


Figure 63: Illustration of $R(C_{ref}, t)$ for t values below and above 1.

In order to estimate the quality of the pupil contour location, we consider the quantity :

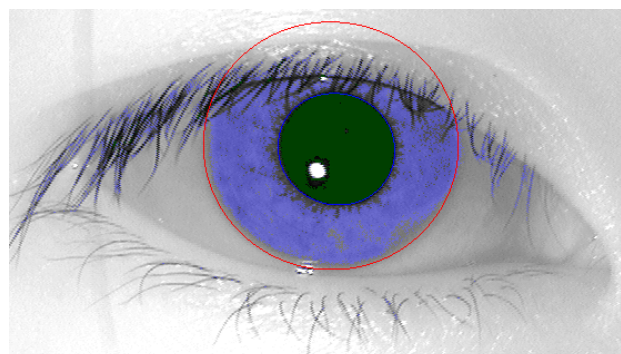
$$M_p(t) = \frac{\sum_{\mathbf{x} \in R(C_{pupil}, t)} \{L(\mathbf{x}) = L_p\} - \sum_{\mathbf{x} \in R(C_{pupil}, t)} \{L(\mathbf{x}) = L_i\}}{\sum_{\mathbf{x} \in R(C_{pupil}, t)} \{L(\mathbf{x}) = L_p\} + \sum_{\mathbf{x} \in R(C_{pupil}, t)} \{L(\mathbf{x}) = L_i\}}. \quad (121)$$

where C_{pupil} is the pupil contour and $L(\mathbf{x})$ is the label associated to \mathbf{x} in the RRS image. $M_p(t)$ is bounded between $[-1; 1]$ and evaluates the ratio between the pixels labeled as L_p and those labeled L_i inside the ring $R(C_{pupil}, t)$.

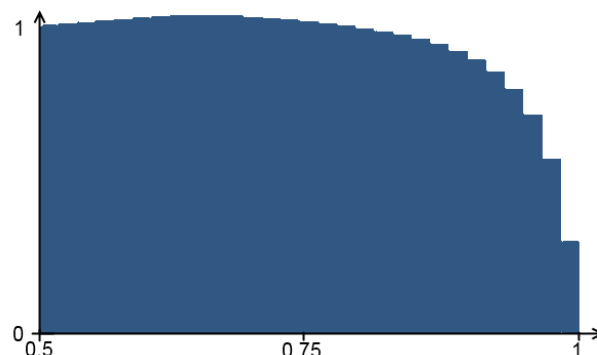
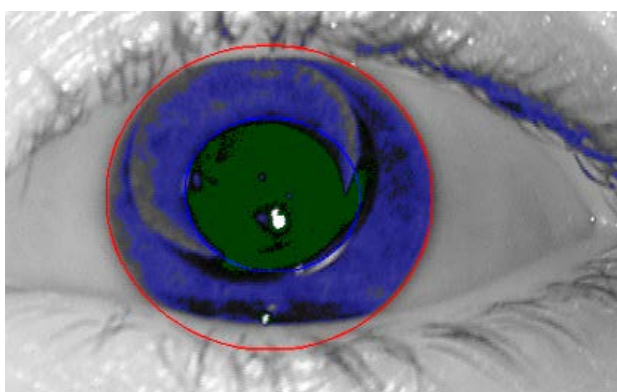
- If $M_p(t) = 1$, there are no pixels labeled as L_i in the ring $R(C_{pupil}, t)$.
- If $M_p(t) = 0$, there are as many pixel labeled as L_p as there are pixels labeled L_i .
- If $M_p(t) = -1$, the ring $R(C_{pupil}, t)$ contains only pixels labeled as L_i .

If C_{pupil} relies exactly on the pupil's border, we should have $M_p(t) = 1$ for $t < 1$ and $M_p(t) = -1$ for $t > 1$. Therefore, if the sign of $M_p(t)$ changes far from $t = 1$ it indicates an incorrect location of the pupil contour. Figure 64 shows $M_p(t)$, $t \in [0.5; 1]$ in 3 different situations.

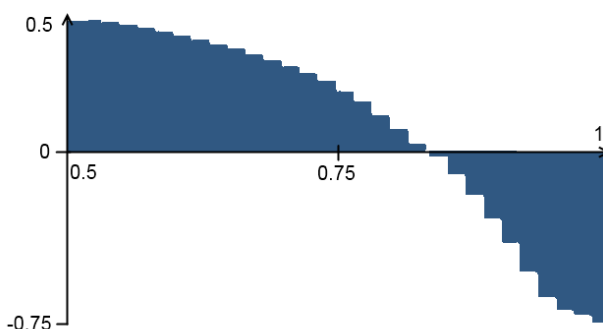
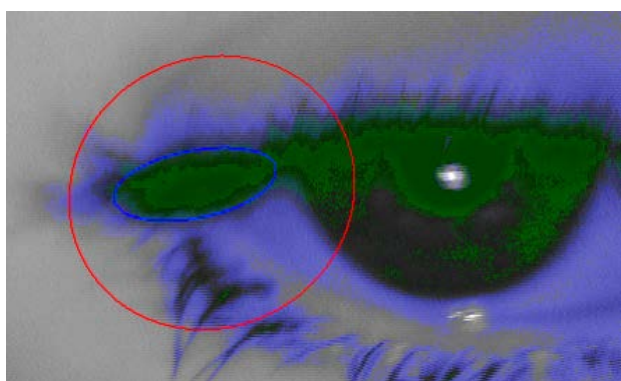
- Figure 64b shows the evolution of $M_p(t)$ for a correctly segmented image 64a. When $t \approx 0.5$ the ring $R(C_{pupil}, t)$ covers a large part of the pupil. Therefore, $M_p(0.5) \approx 1$ as amount of pixel labeled as L_i is small regarding to the one labeled as L_p . As t increases, $M_p(t)$ remains close to 1 until $t \approx 1$ where $M_p(t)$ starts dropping. This last drop is due to the rough nature of the RSS : the borders of the pupil are not as clear in the RRS as they are in the input segmentation, resulting in some mis-classification close to the pupil borders.
- Figure 64c shows an example with C_p outside its expected location. The overall evolution of $M_p(t)$ for this example is roughly the same as it is for the correctly segmented image, but $M_p(t)$ becomes negative for $t = 0.85$. This change of sign is significantly far from $t = 1$, that indicates an inaccurate location of the pupil contour. Moreover, the maximum value obtained for $M(0.5) = 0.5$ is quite low. Indeed, we would have expected a value close to 1 for a contour at the correct location.
- Figure 64e shows an example of a complete segmentation failure. $M_p(t)$ decreases linearly and becomes negative for $t = 0.75$ and the maximum value is only $M(0.5) = 0.3$.



(a) Image correctly segmented

(b) $M_p(t)$ corresponding to (a)

(c) Pupil contour outside the pupil

(d) $M_p(t)$ corresponding to (c)

(e) Complete segmentation failure

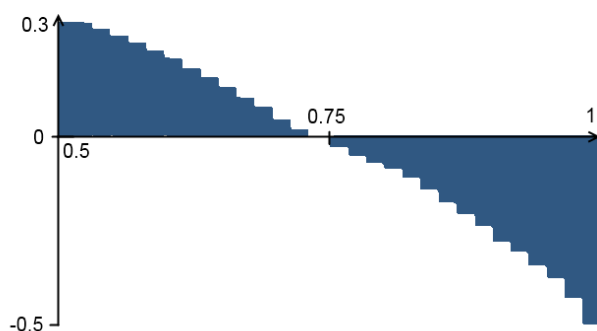
(f) $M_p(t)$ corresponding to (e)

Figure 64: Examples of $M_p(t)$ evolution for different scenarios. Figure 64a and 64b : Illustration of an image where the pupil contour is at the right location. Accordingly, the histogram of $M_p(t)$ between 0.5 and 1 remains close to 1 until $t \approx 1$. Figure 64c and 64d : Illustration of an image where the pupil contour is actually found outside the pupil. Accordingly the value of $M_p(t)$ decreases as t increases. This means that the closer the ring is to the pupil contour C_p , the more mistakes are made. Figure 64e and 64f : Illustration of an image where the pupil contour is at the wrong location. In this situation, $M_p(t)$ decreases linearly as t increases.

The examples of Figure 64 show that $M_p(t)$ can characterize good and bad locations of C_p . We found interesting to use more information in $M_p(t)$ than a simple value like the value of t for which $M_p(t) = 0$. Therefore, we propose to use the integral of $M_p(t)$ to characterize the quality of the contour location. For correctly segmented image (like in Figure 64b) we expect to have $\int M_p \approx 1$, and for poorly segmented image (like in Figure 64d) we expect to have $\int M_p \approx 0$. Formally we define :

$$s_{Pout} = 2 \int_{t=0.5}^1 M_p(t) dt \quad (122)$$

where s_{Pout} aims at characterizing if the pupil contour is outside its expected location.

In the same way, we define s_{Pin} that aims at characterizing if the pupil contour is inside its expected location :

$$s_{Pin} = 2 \int_{t=1}^{1.5} M_p(t) dt \quad (123)$$

We define the same kind of metrics for the iris contour :

$$M_i(t) = \frac{\sum_{\mathbf{x} \in R(C_{iris}, t)} \{L(\mathbf{x}) = L_i\} - \sum_{\mathbf{x} \in R(C_{iris}, t)} \{L(\mathbf{x}) = L_b\}}{\sum_{\mathbf{x} \in R(C_{iris}, t)} \{L(\mathbf{x}) = L_i\} + \sum_{\mathbf{x} \in R(C_{iris}, t)} \{L(\mathbf{x}) = L_b\}}, \quad (124)$$

for counting the amount of pixels labeled as L_i as opposed to pixels labeled as L_b in the ring $R(C_{iris}, t)$. In the same way we did for the pupil contour we define :

$$\begin{aligned} s_{Iout} &= 2 \int_{t=0.5}^1 M_i(t) dt \\ s_{Iin} &= 2 \int_{t=1}^{1.5} M_i(t) dt \end{aligned} \quad (125)$$

where s_{Iin} aims at characterizing if the iris contour is inside its expected location and s_{Iout} if it is outside.

Figure 65 shows that all the computed metrics have a good correlation with the iFNMR.

- The iFNMR decreases linearly with an increase of s_{Pin} . Therefore, images having a low value of s_{Pin} ($s_{Pin} < 0$) are more likely to produce False Matching.
- The iFNMR is first stable w.r.t. increases of s_{Pout} and then starts to increase for $s_{Pout} > 0$. Therefore, images having a high value of s_{Pout} ($s_{Pout} > 0$) are more likely to produce False Matching.
- The iFNMR first decreases sharply with increase of s_{Iin} and then stops evolving for $s_{Iin} > 0$. Therefore, images having a low value of s_{Iin} ($s_{Pin} < 0$) are more likely to produce False Matching.
- The iFNMR increases linearly with an increase of s_{Iout} . Therefore, images having a high value of s_{Iout} ($s_{Iout} > -0.6$) are more likely to produce False Matching.

Figure 66 shows examples of images among the worst w.r.t. these 4 metrics. s_{Pin} , s_{Pout} and s_{Iout} have the expected behavior, but s_{Iin} identifies images with large occlusions despite a correct location of the contour. It is not the behavior we expected for s_{Iin} but it is known [71] that the occlusion ratio has a strong correlation with the recognition performance of the system, explaining why we observe a correlation between s_{Iin} and the iFNMR.

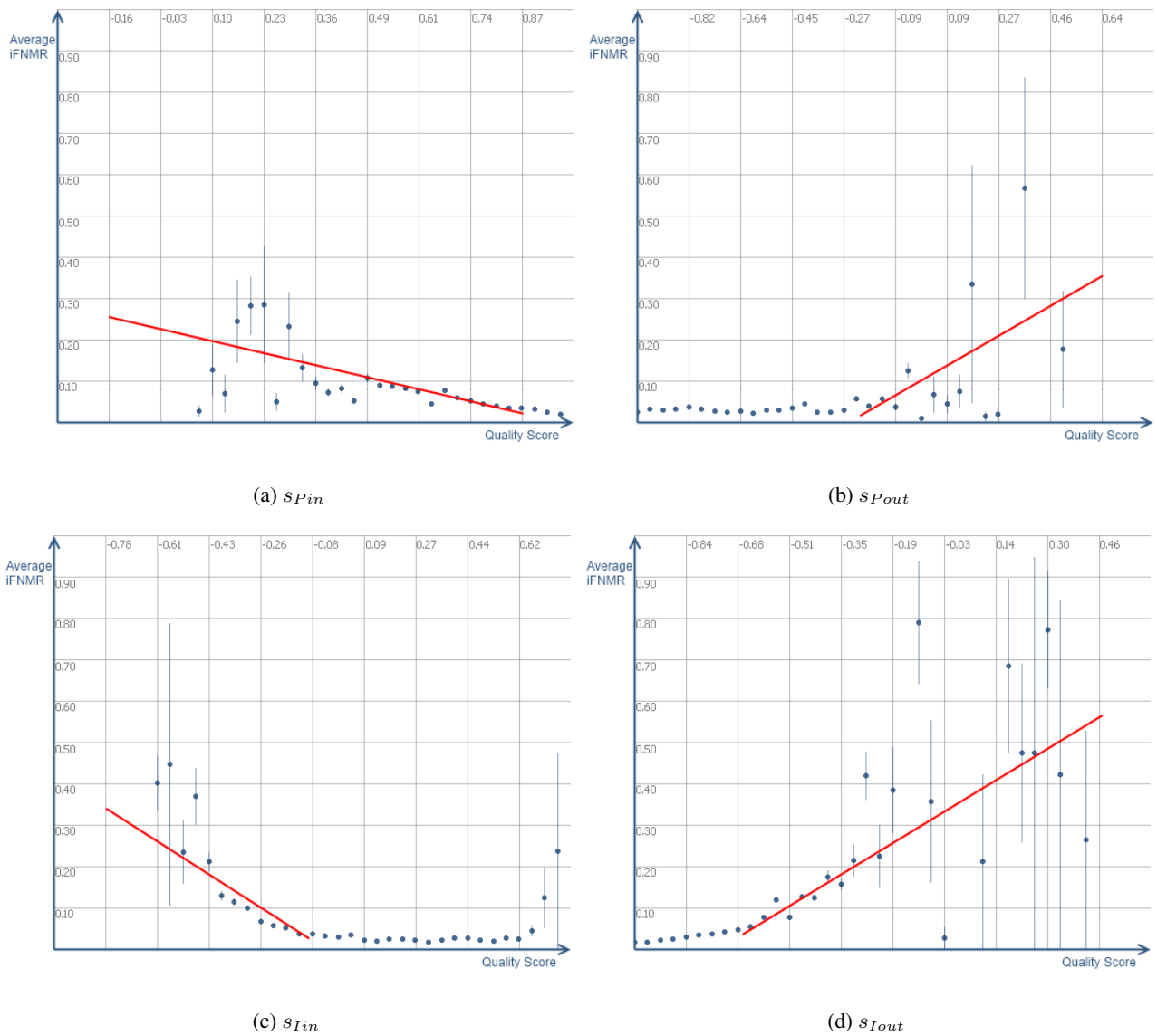


Figure 65: Average iFNMR for s_{Pin} , s_{Pout} , s_{Iin} and s_{Iout}

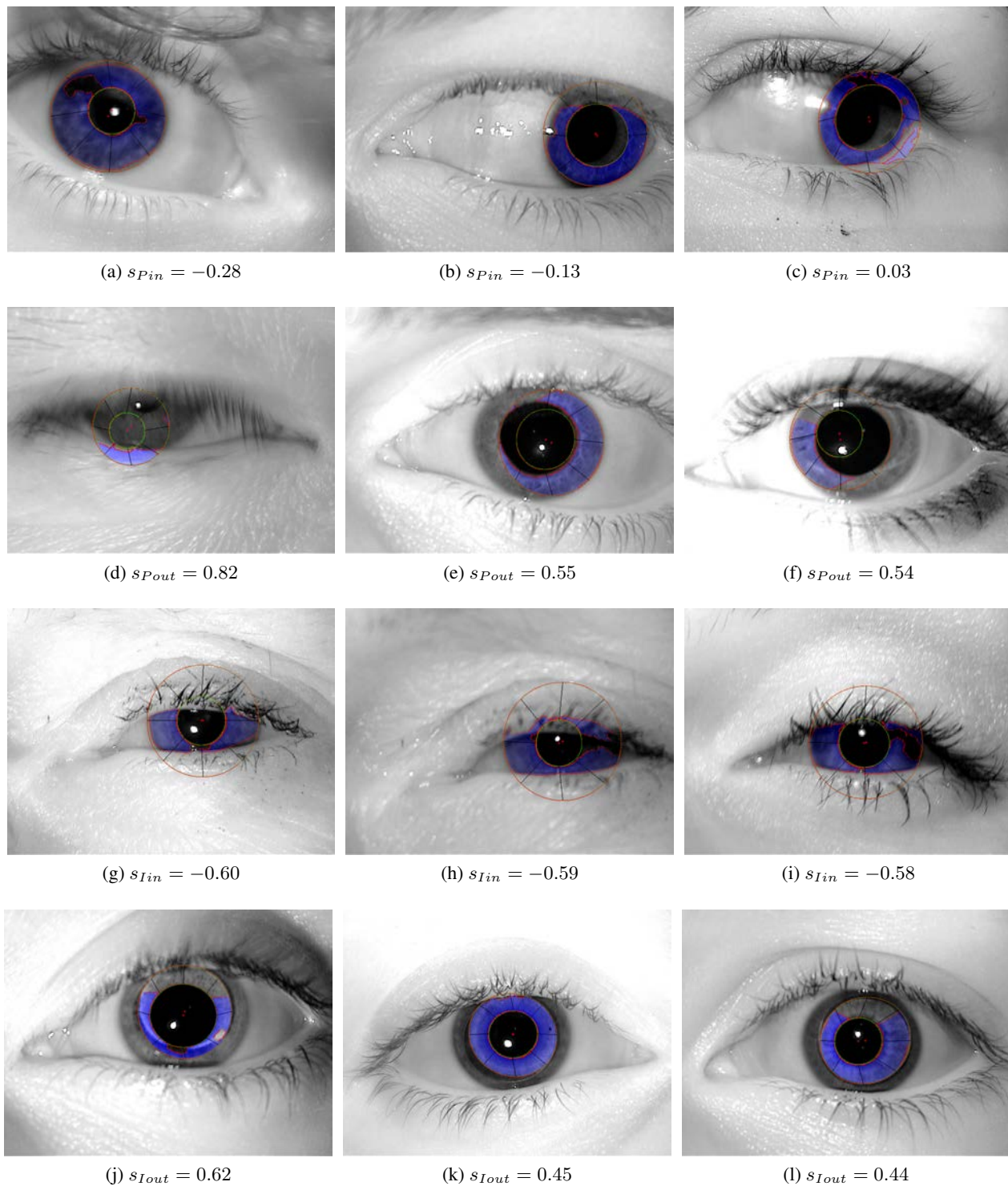


Figure 66: Example of images with the worst quality score for metrics s_{Pin} , s_{In} , s_{Pout} and s_{Iout}

7.3.2 Empirical Metrics

The basic principle of empirical metrics is illustrated by Figure 67. The idea is to evaluate how well a segmentation matches a set of empirical assumptions on what should be a good segmentation. However, as illustrated in Figure 67, we find more relevant to evaluate these metrics on the RRS image instead of on the input segmented image because the RRS process tends to enhance segmentation failures ; indeed, some problems that cannot be seen in the original segmentation may be visible in the RRS image.

Our assumptions we have for a correct segmentation are the following :

- The pupil and the iris regions should be close to a circle, or at least, have a compact shape.
- The distribution of the pupil and iris intensities should take a limited range of intensity.

We have considered some other unsupervised metrics proposed in the survey [79] like simple intra-region entropy : $\int_R h(\mathbf{x}) \log(h(\mathbf{x})) dx$ with h the histogram of the region R . However, our experiments showed that metrics assuming intra-region homogeneity are not directly correlated to the iFNMR.

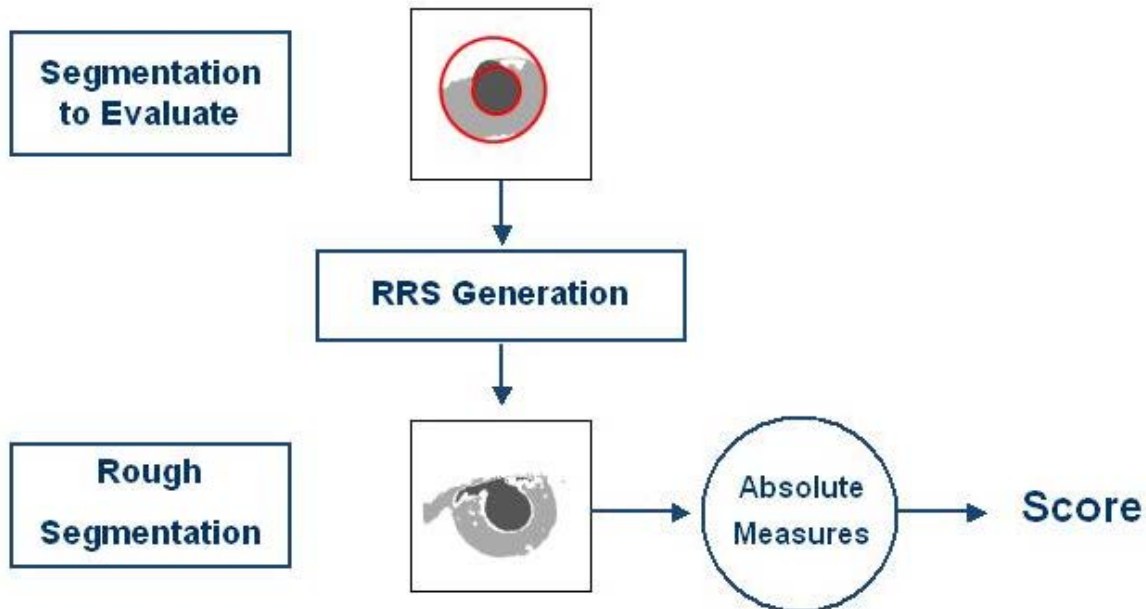


Figure 67: Empirical quality metrics.

7.3.2.1 Circularity and Compactness

We use the criterion presented in [79] to evaluate the circularity and the compactness of a region, i.e. a set of pixels. Let p_k be the perimeter and S_k be the area of the region (region k being either the pupil or iris region). We have :

$$\begin{aligned} compactness &= \frac{p_k^2}{S_k} \\ circularity &= \frac{4\pi S_k}{p_k^2} \end{aligned} \quad (126)$$

Figure 68 shows the average iFNMR for the pupil and iris circularity and compactness :

- The iFNMR does not change with an increase of the pupil circularity, the iris circularity and the pupil compactness. Therefore, there is no correlation between them and the iFNMR.

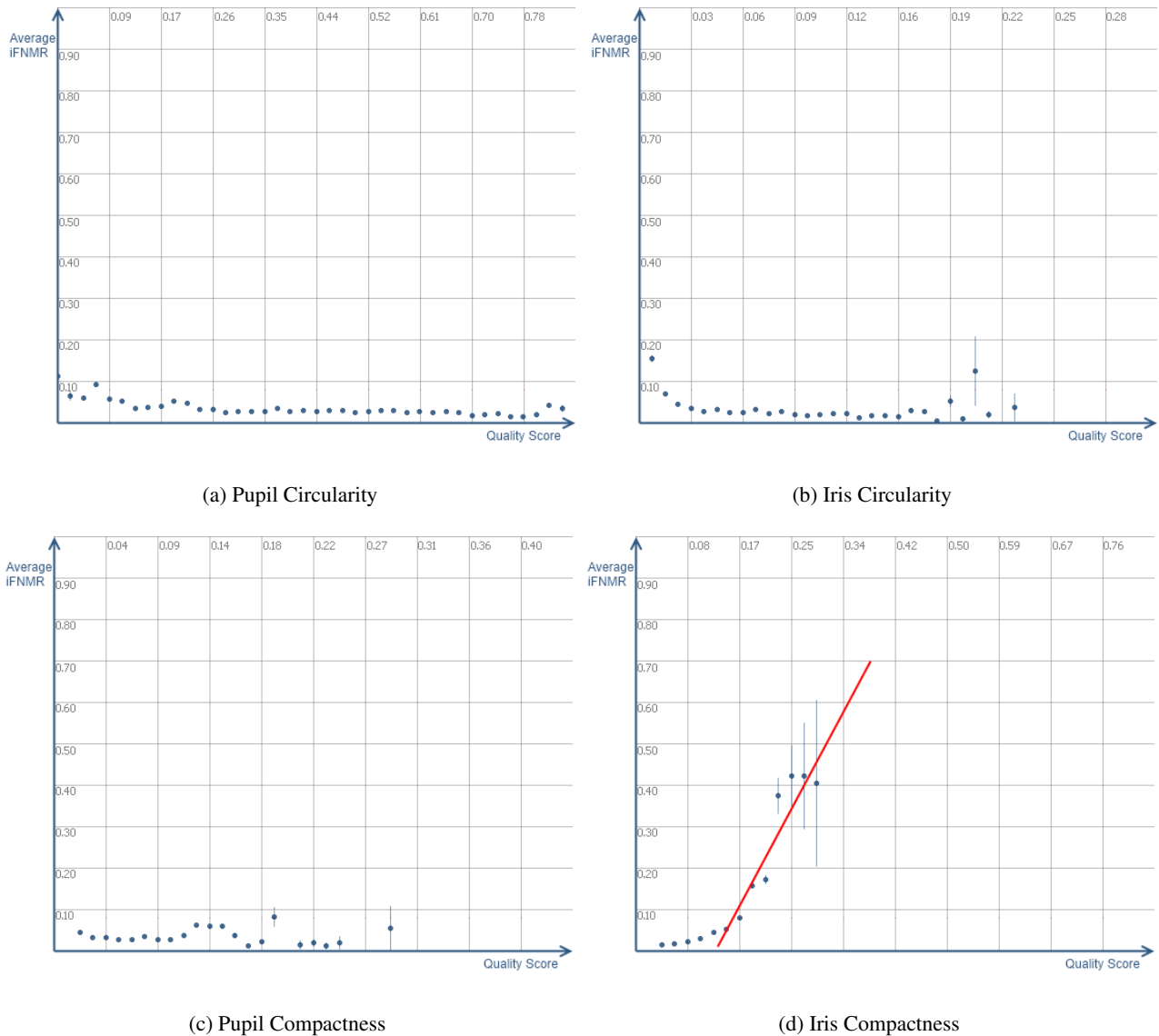


Figure 68: Average iFNMR for the pupil and iris compactness and circularity

- The iFNMR clearly increases w.r.t. an increase of the metric of the iris compactness. This increase is especially notable for a compactness above 0.15.

Accordingly we only select the metric of the iris compactness (we note s_{Icomp}) as being a relevant metric.

7.3.2.2 Contrast

Iris and pupil contrast are among the most discriminative features for image quality according to the IREXII-IQCE challenge. For this reason, we implemented the corresponding criteria to check whether they can be used to estimate a segmentation's quality. Let m_p be the average intensity of the pupil region, m_i the average intensity of the iris region and the m_b the average intensity of the background. Then the pupil/iris contrast and the iris/background contrast are defined as :

$$\begin{aligned} \text{contrast pupil/iris} &= \frac{\|m_i - m_p\|}{255} \\ \text{contrast iris/background} &= \frac{\|m_b - m_i\|}{255} \end{aligned} \quad (127)$$

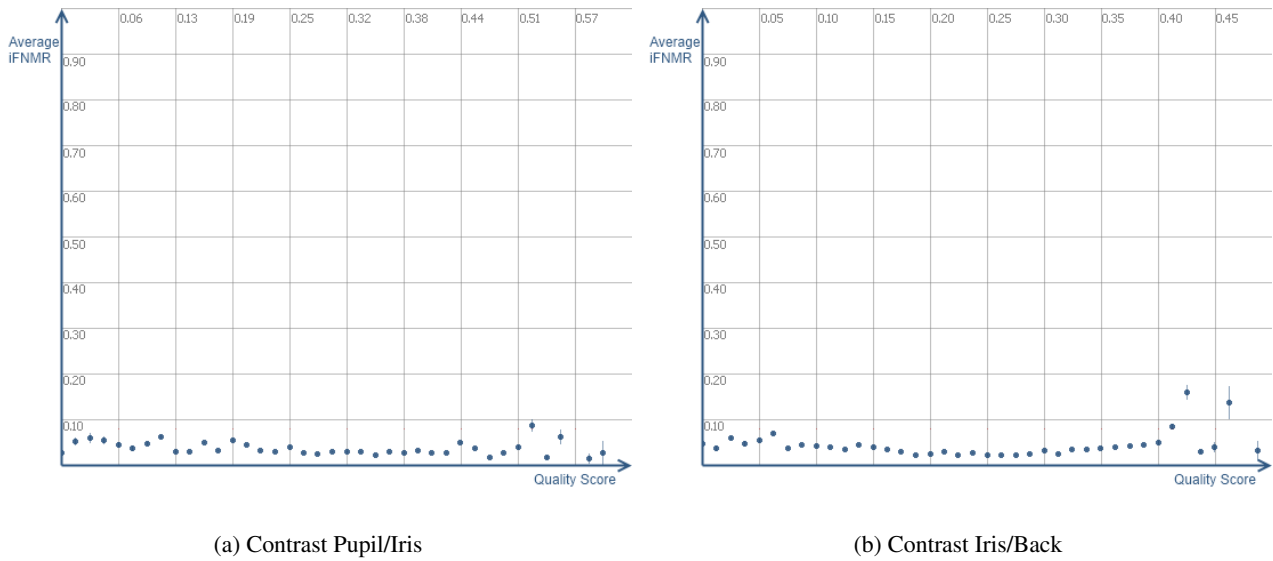


Figure 69: Average iFNMR for the pupil/iris and iris/background contrast

Figure 69 shows that the iFNMR does not evolve with changing in these metrics. Therefore there is no correlation between the contrast and the iFNMR.

Our results differ from the IREXII-IQCE challenge benchmark because we do not use the same protocol to assess the metric's performance. In the IREXII-IQCE challenge the contrast is evaluated for the two compared images and the evaluation is performed on the difference between the contrast of the two images. Therefore, the conclusion of the IREXII-IQCE challenge is that the contrast should be close in the two compared image. What we can conclude from our experiments is that the contrast of a single image is not informative on the quality of the image.

7.3.2.3 Variance of L_i and L_p

If the segmentation is correct, the associated Gaussian Mixture Model L_i and L_p generated during the RRS process should cover a limited range of intensity. Therefore, we evaluate the variance of L_i and L_p to assess respectively the quality of the iris and pupil region.

- The iFNMR does not change with an increase of the variance of L_p . Therefore, there is no correlation between this metric and the iFNMR.
- The iFNMR increases with the variance of L_i . This increase is important for a variance above 15.

Accordingly, we only select variance of L_i (we note s_{Ivar}) as being a relevant metric.

7.3.2.4 Distance between L_i and L_p and a Gaussian Model

The quality score presented in the previous section assumes that the pupil and the iris regions should cover a limited range of intensities. In this section, we propose an even more constraint model : we assume that the regions' intensity should follow a Normal law. Indeed, if the segmentation has failed, the mixture should be composed of very different Gaussians producing a Gaussian mixture very different from a Normal law. Therefore, we evaluate the distance between the GMM of L_p and L_i and the Normal distribution of same mean and same variance. We compute this distance using the Battacharya distance :

$$d_B(p, q) = \int_{-\infty}^{\infty} \sqrt{p(x)q(x)} dx, \quad (128)$$

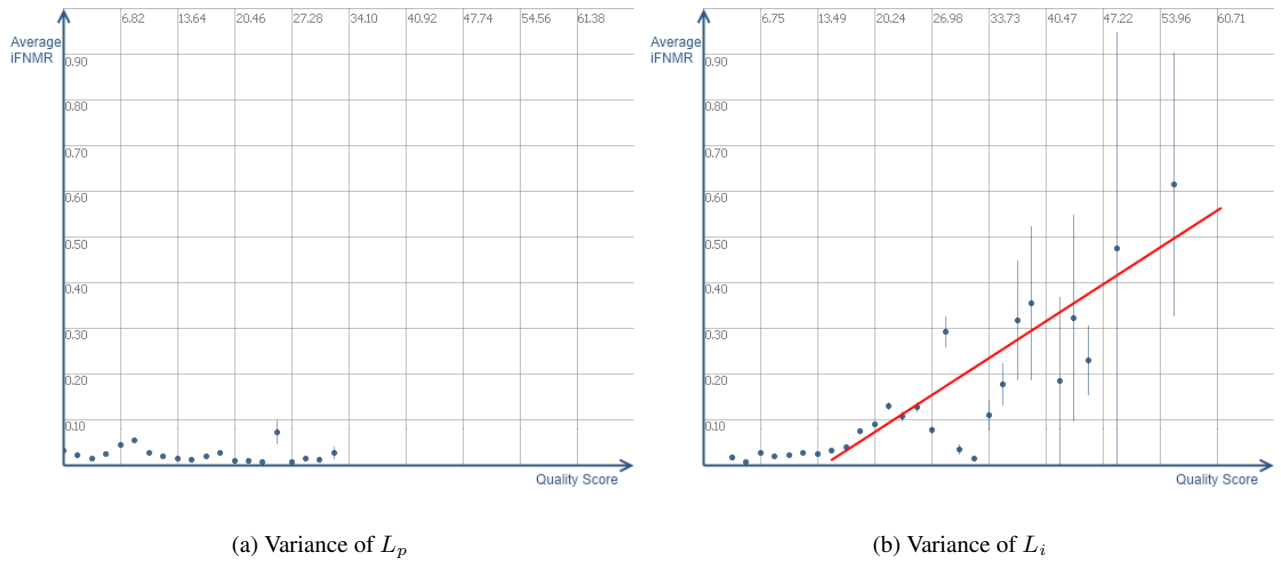


Figure 70: Average iFNMR for the variance of L_i and L_p .

where p and q are two probability distributions defined on \mathbb{R} .

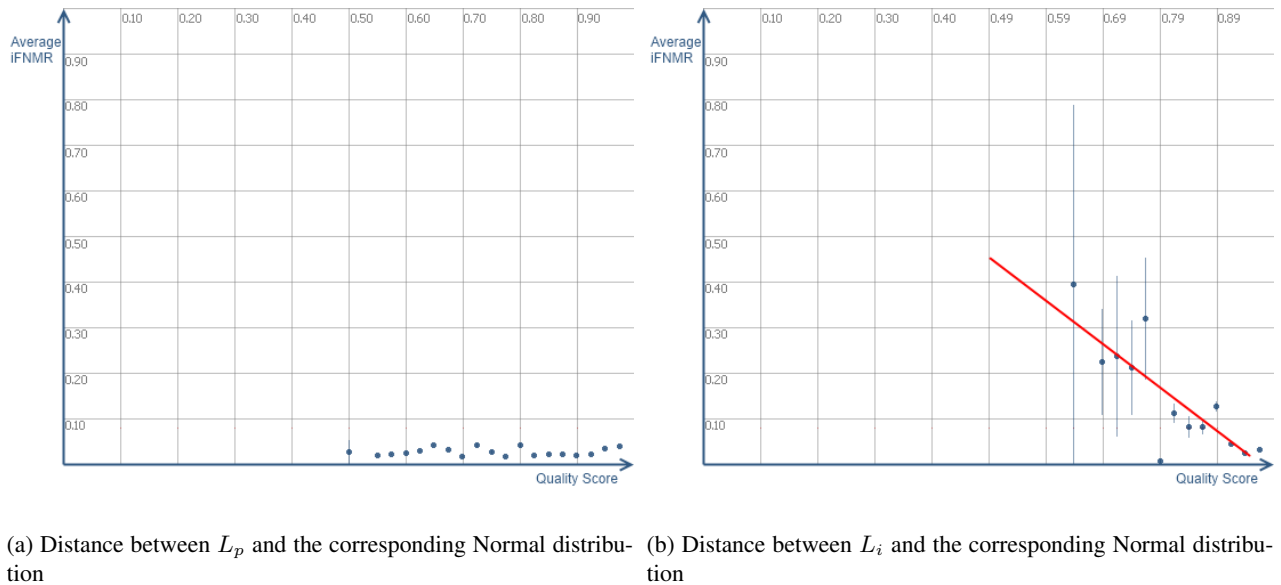


Figure 71: Average iFNMR for L_p and L_i distance to its corresponding Normal distribution

- The iFNMR does not change with an increase of the distance between L_p and the corresponding Normal distribution. Therefore, there is no correlation between this metric and the iFNMR.
- The iFNMR decreases for an increase of the distance between L_i and the corresponding Normal distribution. As this evolution is linear, there is a high correlation between the iFNMR and the distance between L_i and the corresponding Normal distribution.

Accordingly, we only select the distance between L_i and the corresponding Normal distribution (we note $sIDistGMM$) as being a relevant metric.

7.3.3 Selected Metrics

Finally, we selected the following metrics : $s_{SegDiff}$, s_{Pin} , s_{Pout} , s_{Iout} , s_{Icomp} and $s_{IDistGMM}$ since those metrics show a high correlation with the iFNMR. In Figure 72, we evaluate their impact on the complete recognition system by removing the worst 1% and 5% of the database with respect to these metrics in Figure 72.

In the context of a global recognition system, a good quality metric should rank images w.r.t. their influence on the system's recognition performance. Therefore, removing images of worst quality should improve the overall system's recognition performance. Moreover, for a good quality metric, the more images are removed, the better the system's recognition performance should be.

For the metrics we display in Figure 72, removing 1% of the "worst" images in the database significantly improve the system's recognition performance. However, removing 5% of the "worst" images instead of 1% does not improve system's recognition performance. This means that these quality metrics are pertinent for identifying the worst images of the database but they do not to give a ranking between normal images.

Let us note that we chose not to retain the metrics s_{Iin} and s_{Ivar} despite of a good correlation with the iFNMR. We did not select s_{Iin} because, as we can see in Figure 66, this criterion evaluates mainly the occlusion rate of the image and not the segmentation quality in itself. Accordingly, we decided not to use this metric. We also did not retain s_{Ivar} because it is too similar to $s_{IDistGMM}$, but with a lower correlation with the iFNMR than $s_{IDistGMM}$.

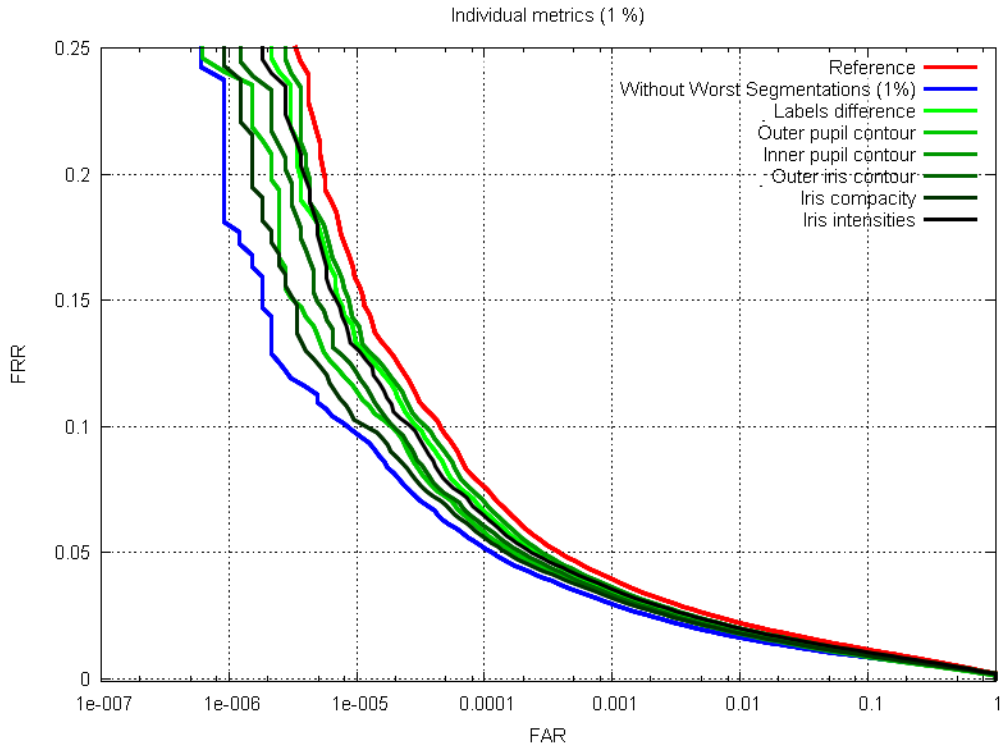
Table 6 shows the correlation between the selected quality metrics (normalized between 0 and 1). The correlation ρ between two sets of values $x = \{x_i; i = 1..n\}$ and $y = \{y_i; i = 1..n\}$ is defined as :

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{\frac{1}{2}}} \quad (129)$$

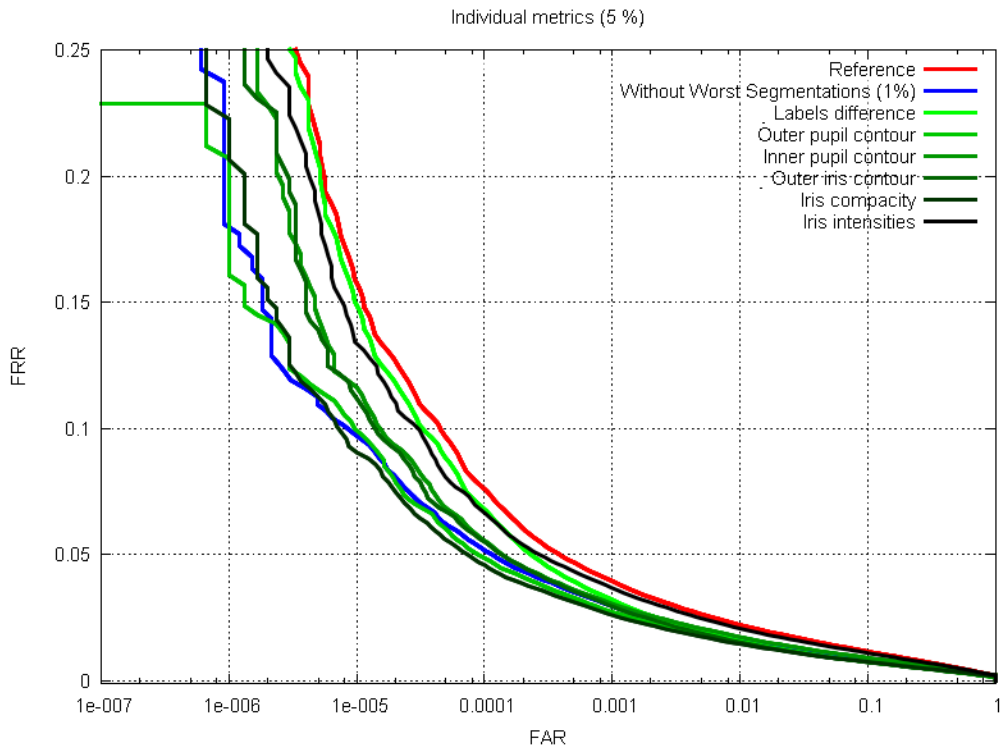
Most of the possible pairs of metrics are very slightly correlated. The pair of metrics showing the highest correlation is $(s_{SegDiff}, s_{Iout})$ with a correlation of 0.425.

| | $s_{SegDiff}$ | s_{Pin} | s_{Pout} | s_{Iout} | $s_{iriscmpactness}$ | $s_{IDistGMM}$ |
|----------------------|---------------|-----------|------------|------------|----------------------|----------------|
| $s_{SegDiff}$ | 1 | 0.025 | 0.054 | 0.425 | 0.112 | -0.001 |
| s_{Pin} | | 1 | -0.220 | 0.226 | -0.209 | -0.100 |
| s_{Pout} | | | 1 | 0.049 | 0.207 | 0.166 |
| s_{Iout} | | | | 1 | -0.059 | -0.042 |
| $s_{iriscmpactness}$ | | | | | 1 | 0.299 |
| $s_{IDistGMM}$ | | | | | | 1 |

Table 6: Correlation between the selected quality metrics



(a) Removing 1% of the database



(b) Removing 5% of the database

Figure 72: Impact on the system’s recognition performance when removing the worst 1% and 5% images w.r.t selected quality metrics.

7.4 Failure Detection

For failure detection, we propose to fuse the quality metrics presented above in order to compute a final segmentation quality score. Indeed, fusion allows exploiting the complementarity between such metrics. We selected our quality metrics based on their correlation with the iFNMR. Therefore, we consider the fusion problem as a regression one, trying to predict for an image the corresponding iFNMR. To tackle this problem, we decided to use a Machine Learning approach and train a Support Vector Regression (SVR) to predict the iFNMR of the images.

In this section, we first make a brief description of SVR methods, then we explain how we use them to predict the iFNMR of an image.

7.4.1 Support Vector Regression

Support Vector Regression (SVR) is a particular kind of Support Vector Machines (SVM) dedicated to regression problems. SVM are a traditional type of supervised machine learning algorithms. SVM are trained using a set of examples for learning the underlying properties of the input set in order to predict the properties of a new input. SVM have been developed in the 90th by Vapnik and his co-workers at AT&T Bell laboratories [6]. The idea to use SVM for regression was first proposed by Vapnik et. al [75] in 1996 with the ϵ -SVR we will describe here. In the following, we take the formulation introduced in the survey [67].

We note $\{(\mathbf{m}_1, s_1) \dots (\mathbf{m}_n, s_n)\}$ the training set, where $\mathbf{m}_i \in R^n$ represents the space of input patterns (the quality metrics for our purpose) and $s_i \in R$ is the expected pattern to learn from the input ones (the iFNMR in our case). The objective of traditional ϵ -SVR is to find a function f that has as many output patterns s_i as possible inside the band of size ϵ : $[f(\mathbf{m}_i) - \epsilon; f(\mathbf{m}_i) + \epsilon]$. Figure 73 shows such a linear function f with all the training samples lying inside a band of width ϵ .

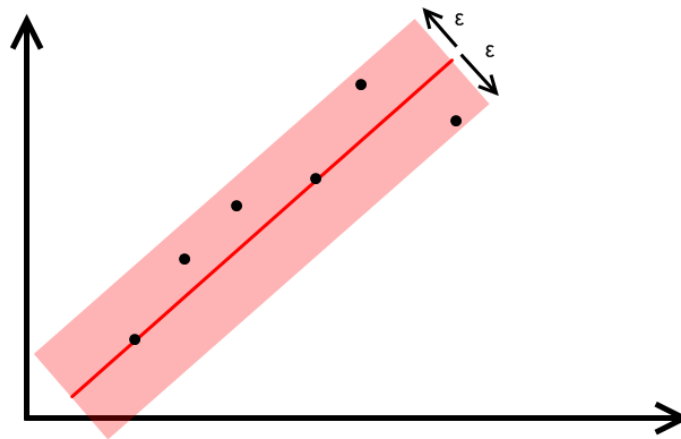


Figure 73: Illustration of a linear ϵ - SVR

A linear function f can be written as :

$$f(\mathbf{m}) = \langle \omega, \mathbf{m} \rangle + b, \text{ with } \omega \in R^n, b \in R \text{ and } \langle \cdot \rangle \text{ the scalar product in } R^n. \quad (130)$$

The problem to solve becomes :

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\omega\|^2 \\ & \text{subject to} && \begin{cases} s_i - \langle \omega, \mathbf{m}_i \rangle - b \leq \epsilon \\ \langle \omega, \mathbf{m}_i \rangle + b - s_i \leq \epsilon \end{cases} \end{aligned} \quad (131)$$

This problem can be solved using a dual problem (see [67] for calculus details) and function f can be written :

$$f(\mathbf{m}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \mathbf{m}_i, \mathbf{m} \rangle + b, \quad (132)$$

where α_i and α_i^* are Lagrangian multipliers representing the constraints of the minimization. Equation (132) is called the support vector extension of the function because f is only described using the training parameters \mathbf{m}_i (b can also be computed directly using \mathbf{m}_i). An interesting aspect of this formulation is that the complexity of the function f is more dependent on the training parameters than of the dimension of the input space.

We have described here only the basic idea of SVR with a linear kernel. It is possible to consider non linear kernels instead of linear ones, yet the final function is always represented using the training parameters. For a more detailed description of SVR possibilities, see [67].

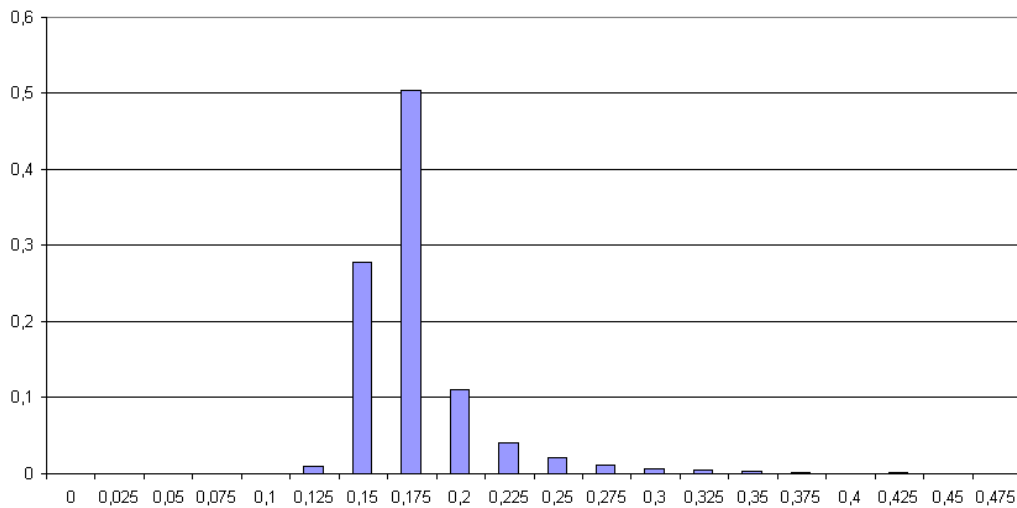
7.4.2 iFNMR Prediction

The first problem for every Machine Learning procedure is the construction of the training set. To do so, we considered two classes of images.

Class I The image has an iFNMR of almost 0 (below 0.2)

Class II The image has a high iFNMR (above 0.2)

Most of the images fall in *Class I*, so to avoid over-learning, we randomly select 20% of the images in *Class II* and add the same amount of images from *Class I* (selection also random). The resulting learning data base is composed of 350 images containing as many *Class I* images as *Class II* images. The remaining images are used for validation.



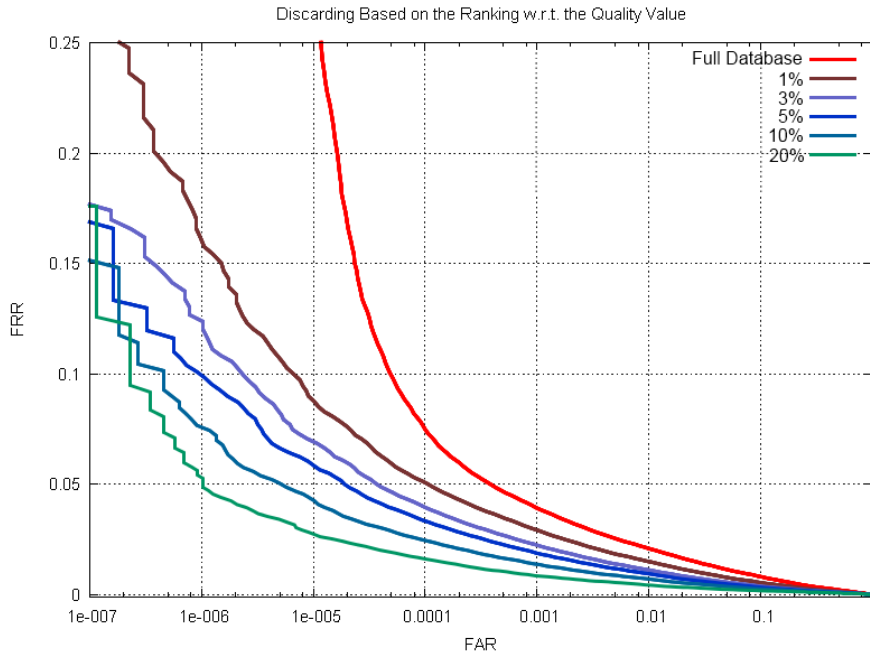
(a) Quality Score Distribution

Figure 74: Quality score distribution of the SVR results. As the SVR predict the iFNMR of an image, the lower the score is, the better the quality is.

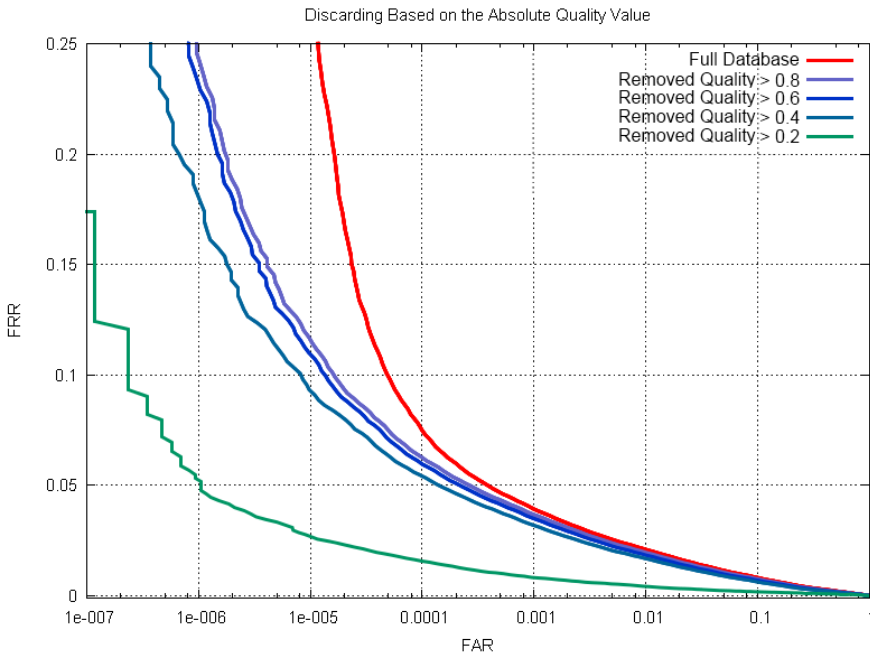
In order to carry out the learning and regression steps of the SVR, we use the toolbox LibSVM [52] with a ϵ -SVR parametrization. We tried several kernel types and learning options and find out (based on cross-validation and its influence on the recognition performance) that cubic kernel seems to be the most suitable for our problem. Figure 74 gives the score distribution of the SVR results on the full testing database (12,872 images).

Figure 75 shows how the removal of the worst images in the database according to SVR results has an impact on recognition performance. In Figure 75a we remove images based on their ranking w.r.t. their predicted iFNMR. In Figure 75b, we remove images based on the absolute value of their predicted iFNMR.

The final quality score (predicted iFNMR) has a better influence on the system's recognition performance than any of the individual metric. We also remind that removing 1% or 5% of the database for the individual metrics had roughly the same impact on recognition performance (see Figure 72 in Section 7.3.3). On the contrary, SVR results seems to give a good ranking of the image quality since the more images are removed, the better the recognition performance.



(a) Rank based evaluation



(b) Score Based evaluation

Figure 75: Evaluation of recognition performance when removing the worst images according to the score generated by the SVR on the ND-Iris Database

7.5 Evaluating the Impact of the Fused Quality Score

As our final quality score is based on a Machine Learning algorithm, we consider important to assess that the resulting scores are neither dependent on the image database used for training (here ND-Iris) nor on the algorithm used to generate the input segmentation (here B-Snakes). Therefore we have applied our SVR (without retraining) on a different database and on images generated with a different segmentation algorithm.

Let us note that in the following, we consider that our quality score is the prediction of the image's iFNMR. Therefore, a high output value corresponds to a bad quality score.

7.5.1 Evaluation on the Casia-Thousand Database

We chose to evaluate the consistency of our quality score on Casia-Thousand for two reasons. First, the database has been generated using a different sensor than the one used to generate the ND-Iris database. Therefore, the images of the two databases are very different. The second point is that this database contains challenging images for our segmentation algorithm (like for instance glasses, that may generate large specular reflections on the image or corrupting pupil initialization). Therefore, we expect to have worse segmentation results on this database than on the ND-Iris database.

Among the 3000 randomly selected images that we considered, 276 did not pass trivial quality checks like having a non-empty segmentation mask, or neither too big nor too small normalization contours. They were therefore discarded and the score evaluation is done on the remaining 2724 images.

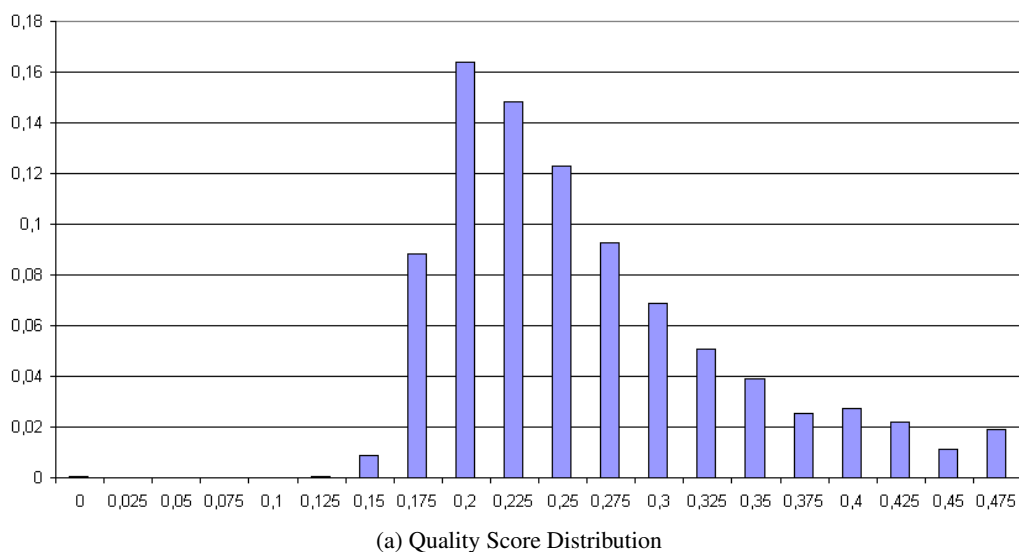
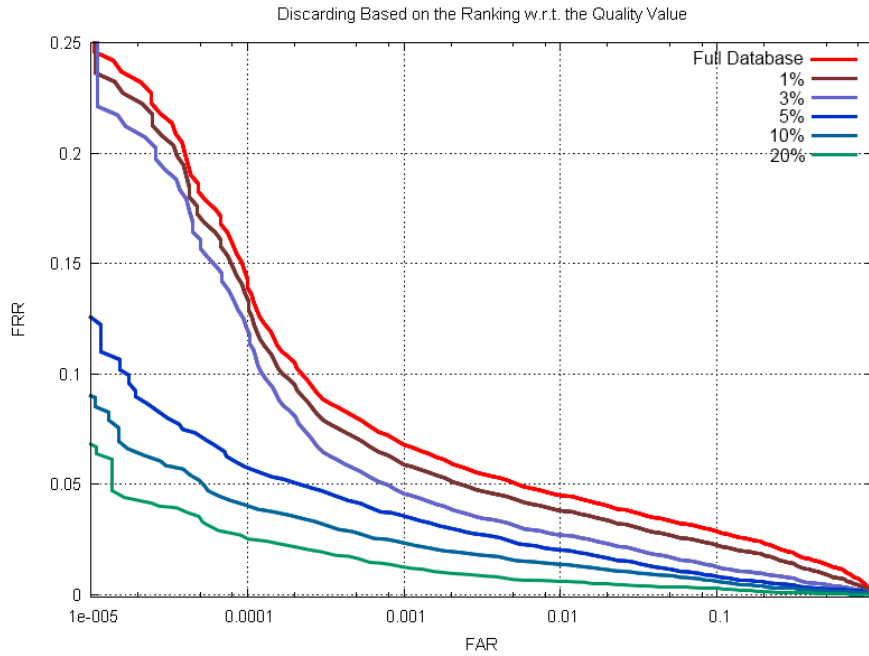


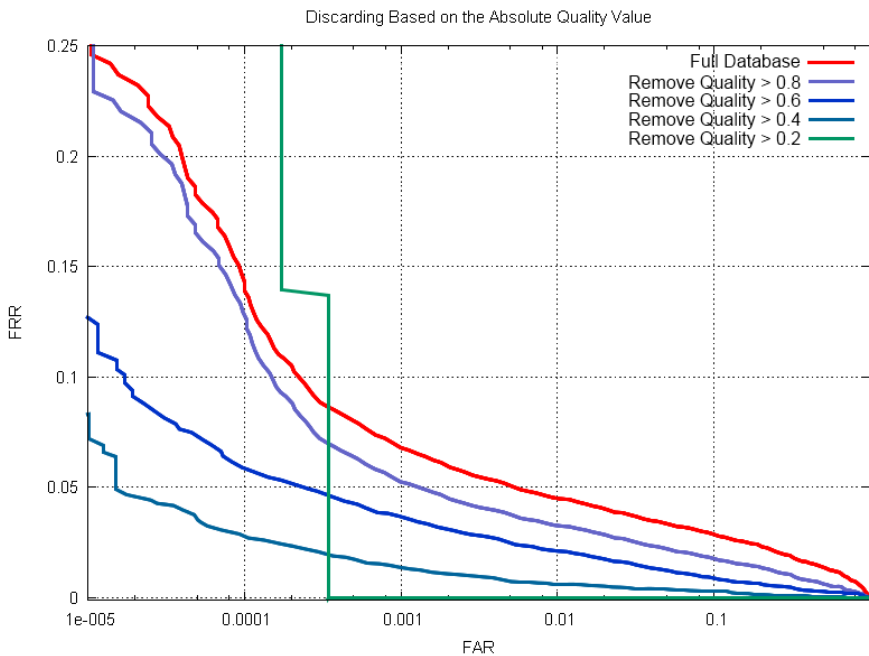
Figure 76: Quality score distribution of SVR trained on the ND-Iris database and applied on the Casia-Thousand database

Figure 76 shows the distribution of the quality score of the 2724 images. The overall distribution is shifted on the right compared to the reference experiment we conducted on the ND-Iris database in Section 7.4. This means that the images of this database have a tendency to have a higher predicted iFNMR on the CASIA-Thousand database than on the ND-Iris database. This is coherent with our expectation that our segmentation algorithm performs worse on this database.

Figure 77 shows that our quality score provides a good ranking of the image quality in the Casia-Thousand database since the more images are removed, the better the recognition performance. We manually check the segmentations of the images with the worst quality score. 113 images have a quality score above 0.8 (some of these images are displayed in Figure 78).



(a) Rank based evaluation



(b) Score Based evaluation

Figure 77: Evaluation of the recognition performance when removing the worst images according to the score generated by the SVR for the Casia-Thousand database

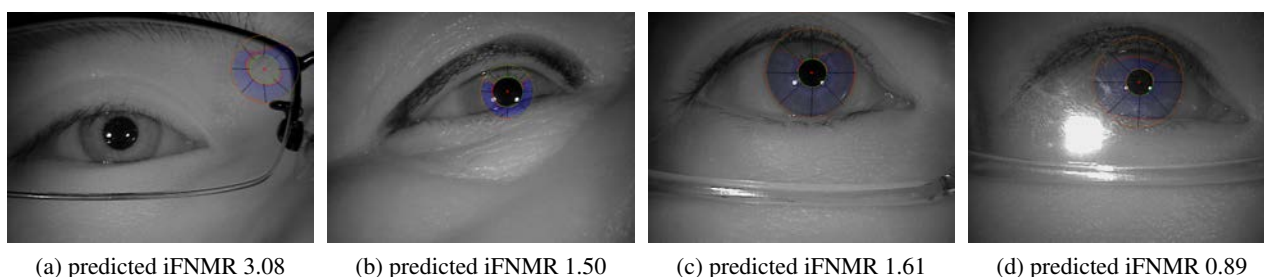


Figure 78: Some of the images of the Casia-Thousand database with the worse quality scores.

This experiment shows that quality score defined in Section 7.4 on the ND-Iris database is still pertinent when used on an image database different from the one used for training. However, some errors could be corrected by using a composite database composed of images from different image databases.

7.5.2 Evaluation Using GAC algorithms

In order to study the sensitivity of our trained SVR to the algorithm used to generate the segmentation, we segment the images using Geodesic Active Contour (GAC ; Section 5.3.1 of Chapter 5) instead of B-snakes used for the training.

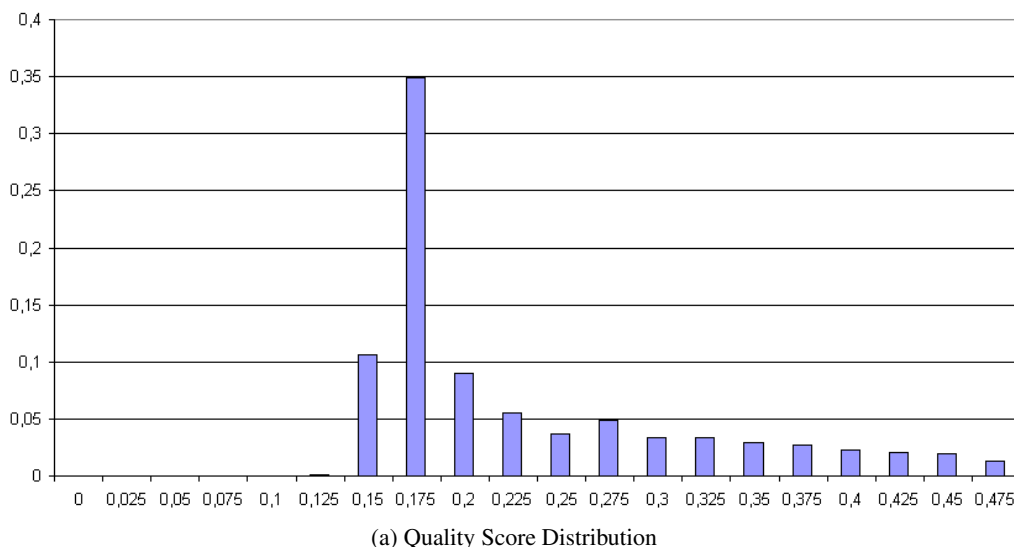
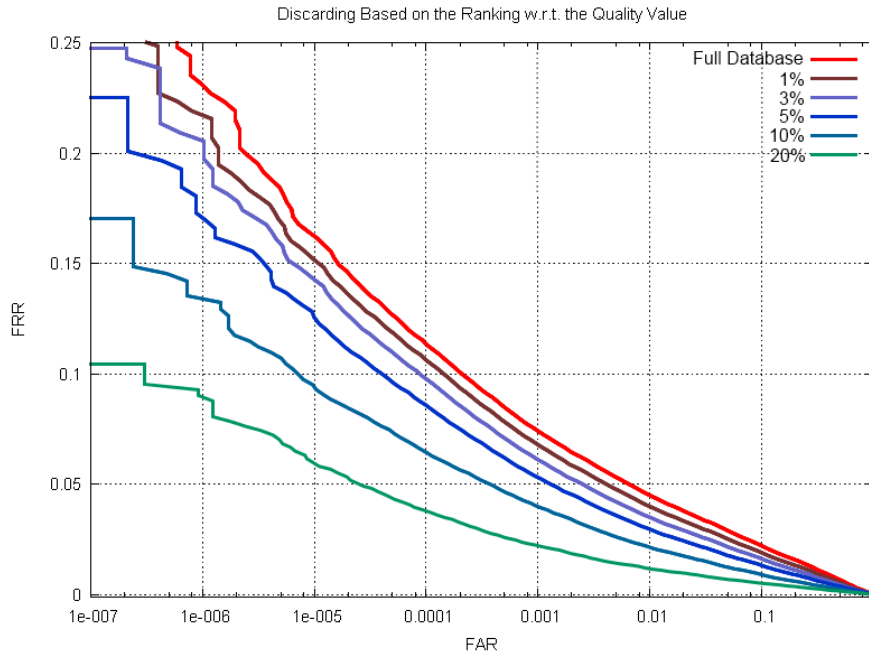


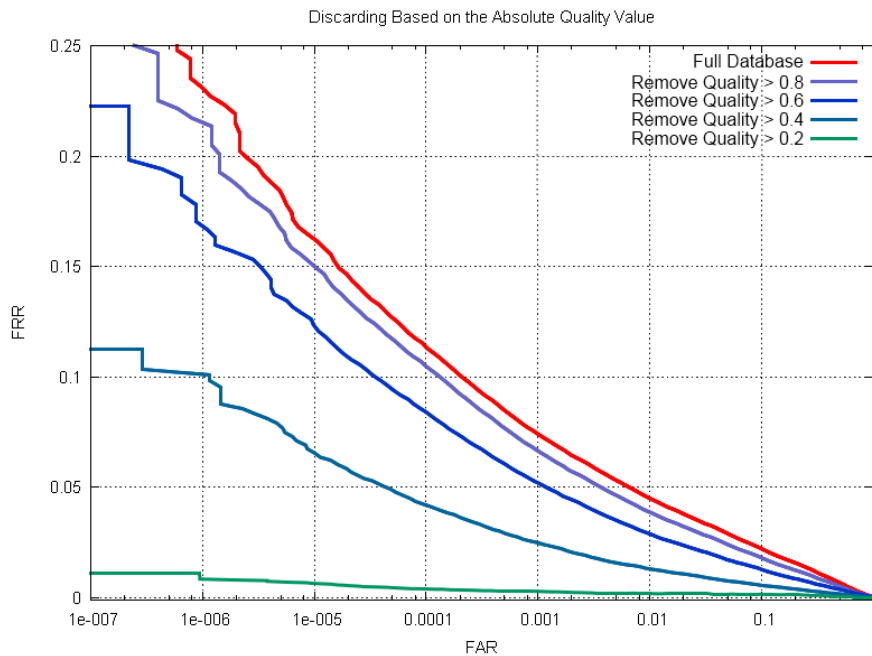
Figure 79: Quality score distribution of the SVR results using GAC segmentation algorithm evaluated on the reference sub-set of the ND-Iris database

Figure 79 shows the distribution of the quality scores when using GAC for segmentation. The distribution has roughly the same shape as the reference one for B-snakes (described in Section 7.4). However, for B-snakes, the distribution almost drops to 0 after 0.3. On the opposite, for the GAC, the distribution decreases very slowly to 0. This means that there are more images with bad quality scores with the GAC algorithm than with the B-snakes algorithm.

Figure 80 shows that the quality score still behaves correctly in the sense that the more images are removed, the better the recognition results are. However, gains in recognition performance increase slowly with the amount of removed images compared to the reference experiment (Section 7.4), in which a bigger gap is obtained when removing the first few percents of the worst images.



(a) Rank based evaluation



(b) Score Based evaluation

Figure 80: Evaluation of recognition performance when removing the worst images according the score generated by the SVR using GAC segmentation algorithm (on the ND-Iris database)

In order to understand this behavior, we made a visual inspection of 200 images with a very bad quality score (above 0.8). Among these 200 images, 114 correspond to complete segmentation failures, 42 correspond to images poorly segmented but suitable for recognition and 44 correspond to correctly segmented images (see Figure 81d for example).

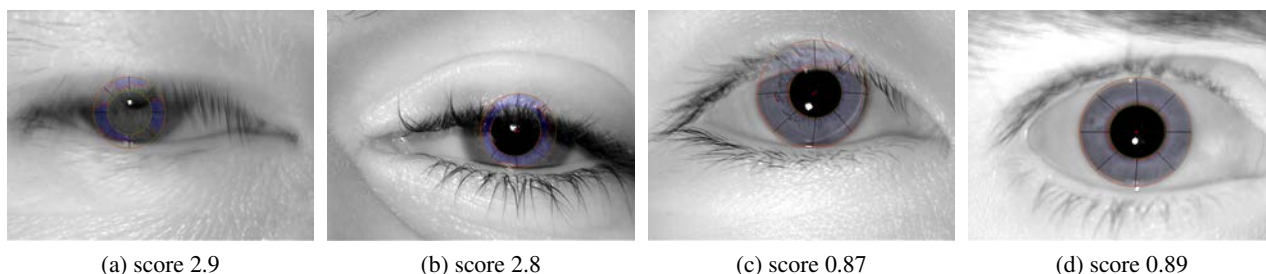


Figure 81: Some of the segmentation generated using GAC with the worst quality scores. The first two ones are large segmentation failures. The third one includes skin area in the iris mask, explaining the bad quality score. The last segmentation seems correct if restricted to the normalization contours.

The problem with the above evaluation is that the iris segmentation mask is restricted to the inner part of the two normalization contours. This is due to the fact that everything that lies outside the normalization contour is not relevant for recognition because it will not appear in the normalized image. However, when looking at the complete segmentation mask of the 86 "correct" images at visual inspection, they almost always correspond to images in which there is an over-segmentation of the iris region.

In order to explain this behavior, we have to recall some properties highlighted in Section 5.5 of Chapter 5. The GAC algorithm evolves using a balloon force may push the contour outward. This tends to produce over-segmentation of the iris region because the contour may cross the iris borders. However, in Section 5.5 of Chapter 5, we have assessed that this kind of segmentation failure does not have a strongly negative impact on the recognition performance.

In Figure 82 we illustrates why over-segmentations are identified as being images of poor quality by our quality measure. Because of the balloon force and the softness of the right iris border, the GAC has leaked out the iris region (See Section 5.3.1 of Chapter 5). Therefore, the RRS generation process associated to the iris labels both the iris and the limbus. Accordingly, the value of s_{ML} is rather bad for this image.

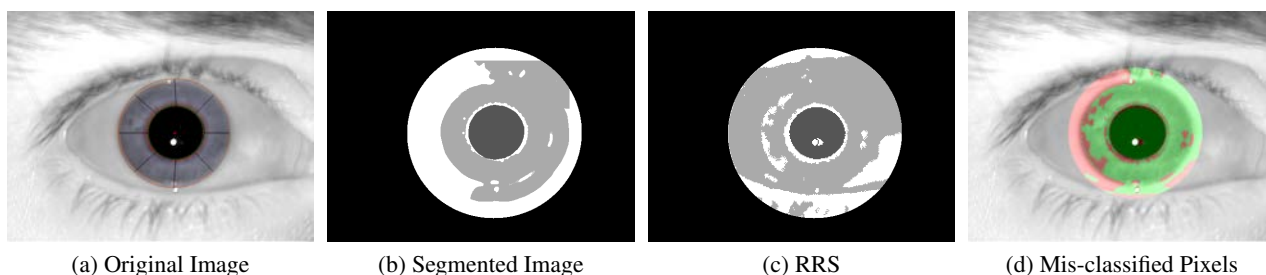


Figure 82: Example of over-segmented image using GAC with $s_{ML} = 0.28$.

These results are interesting because they allow identifying poorly segmented images despite the fact that they have good recognition performance. This shows that our quality metric is pertinent from an image processing perspective. However, identifying these images as being of poor quality can be discussed from a global system perspective. Indeed, in the complete recognition system, we should only remove images having a bad impact on the system's recognition performance. Therefore, we should correct our algorithm's behavior.

7.6 Conclusion

In this chapter, we have proposed a quality metric to evaluate the quality of an iris segmentation. This quality metric is the output of a Machine Learning algorithm trained using basic quality metrics dedicated to different types of segmentation failures.

These metrics use an intermediate Rough Region Segmentation to replace a ground truth for the segmentation evaluation. The generation of this RRS is robust in the sense that the generation process tends to enhance inaccuracies in the original segmentation, but stays coherent with this last one if the segmentation is correct. Though based on a Machine Learning algorithm, we have validated that the learned pattern is still coherent on different image database and for different segmentation algorithms.

We see two main perspectives for our work : first, as we have explained in this chapter, most traditional segmentation quality metrics based on the homogeneity of the regions do not work for iris segmentation evaluation because the iris can be highly textured. Therefore, instead of evaluating the homogeneity of a region using only histograms, a metric able to evaluate the homogeneity based on textural information could most certainly help to incorporate traditional homogeneity-based quality metrics to our system. Then, we have trained our SVR on a single database for a specific segmentation algorithm. We have also assessed that results remain pertinent when changing the iris database and the segmentation algorithm. Yet, we think that building a composite database for training, mixing different databases and generated using different segmentation algorithms, could improve the generalization performance of the SVR.

CONFIDENTIAL

Chapter 8

Conclusion

In this thesis, we have studied the problem of segmenting iris images acquired with low acquisition constraints. Our segmentation system is embedded in a complete recognition system using Daugman's [16] matching algorithm to compare irises. This context led to designing a complete segmentation system and to evaluate it in the framework of its interaction with the pattern matching stage. Therefore, the "best" segmentation system is which has the best complementarity with the matching stage.

We have split the complete segmentation task into several modules, each one dedicated to a singular task. This allows simpler individual tasks to be addressed, however, we have to consider all of them in order to segment the iris. As described at the end of Chapter 3, we have chosen to apply a *Texture First* strategy to segment the iris. This means that we first segment the iris region in the image, and then retrieve the iris borders in order to generate the normalized image. During the development of our segmentation system, some modules naturally appeared as being more critical than others, namely

- Rough localization of the pupil in the eye image in order to initialize the complete segmentation process.
- Accurate detection of the pixels belonging to the iris texture in the image, i.e. iris region segmentation.
- Estimation of the iris borders under possible occlusions in order to generate the normalized image.
- Evaluation of the quality of segmentation results in order to identify failures of the segmentation process.

As a consequence, we have focused on these modules for which we have proposed some novel solutions discussed below.

The localization of the pupil area is extremely critical because an error in its localization usually results in a complete failure of the segmentation process. Most iris segmentation algorithms assume that the pupil is the darkest element in the image and that its location can be obtained using thresholding algorithms [46, 63, 10, 84, 66]. However, this process can fail because of other dark elements in the eye, such as eyelashes or eyebrows. To solve this problem, we have proposed to first segment the eyelashes and eyebrows in the image in order to remove them from the pupil localization process. This approach differs from classical eyelash and eyebrow segmentation algorithms [17, 72, 33, 37] because they assume that the pupil location is known. With this aim, we developed two eyelash and eyebrow segmentation algorithms and assessed their performance on the ND-Iris and Casia-Thousand databases.

We have studied the segmentation of the iris region using Active Contours. Though Active Contours have become popular for iris segmentation [17, 76, 66, 62], to our knowledge, there is actually no comparative evaluation of such algorithms. In Chapter 5, we have conducted a comparative evaluation of different Active Contours implementation for iris segmentation. The novelty of our approach is that the evaluation is performed in a fixed segmentation framework (same initialization, same contour fitting method and same database) and for a given recognition algorithm. Therefore, we can really evaluate the influence of the segmentation stage only. We have studied several variations of Active Contours' implementations in terms of contour generation and energy formulation. This allows us identifying some desirable properties that a segmentation algorithm should have in order to be used in our complete processing stream (from acquisition to recognition).

We have proposed a new precise and effective way to estimate elliptic contours for iris normalization : the EVF framework. This algorithm relies on the Active Contours approach and leads to a compact formulation for ellipses. We have compared our algorithm to classical Direct Least Square fitting for ellipses [22] and to a Randomized Elliptic Hough Transform [45] and have shown that our algorithm performs better on several

reference databases when integrated in the complete processing stream. After this global evaluation, we have addressed some specific issues, such as the evaluation of off-angle images and computation times.

Finally, we have proposed an original module to detect failures of the segmentation process. To the best of our knowledge, only 5 articles [38, 29, 81, 30, 80] study the evaluation of iris segmentation quality. In order to identify poorly segmented images, we first re-segment the iris using an original process that we called Rough Region Segmentation (RRS). This RRS can be interpreted as a ground truth and allows us to design a new set of segmentation quality metrics specific to iris segmentation. We fuse these metrics using a Support Vector Regression (SVR) in order to build a final quality score. We have assessed that our trained SVR is not dependent on the iris database nor on the segmentation algorithm used for training.

Different main problems concerning iris segmentation are thus studied in this thesis, covering a wide range, from basic image processing tasks to high level tasks such as quality measurement. The methodological framework is original since we also study the interaction between the segmentation and the recognition steps of the global system. Conclusions consist in the following :

- (i) Most contour fitting methods used in *Texture First* segmentation strategies [76, 66, 62] make a direct use of the region segmentation results. Therefore, inaccuracies in the segmentation mask can lead to large errors in the contour's location. On the contrary, our EVF framework is barely dependent on the accuracy of the segmentation mask. It only uses this information for initialization and then uses the image's intensity. Moreover, our method is also more robust than classical elliptic fitting methods that we considered : Direct Least Square fitting for ellipses [22] and to a Randomized Elliptic Hough Transform [45]. Indeed, as our EVF framework works on the image instead of on segmentation results, it can even correct inaccuracies of the segmentation mask.
- (ii) The evaluation conducted in Chapter 5 allows us to identify the general properties a segmentation algorithm should have in order to be embedded in our system (i.e. a *Texture First* segmentation strategy using Daugman's [16] matching algorithm). We noticed that segmentation algorithms' performance is more characterized by the nature of their failures than by the accuracy of their generated segmentation mask. This behavior is due to the matching algorithm we used in our complete recognition system. Indeed, this matching algorithm is based on a test of statistical independence on a large number of bit comparisons. Therefore, this process can handle some small inaccuracies in the segmentation mask and still be discriminative ; however, a lack of information results in fewer comparisons of bits and, therefore, in a less reliable matching result. For this reason, under-segmentations have a very negative impact on our system's recognition performance.
- (iii) The evaluation of segmentation quality is a difficult problem because it is very hard to assess what a good segmentation is. The most common method to evaluate the accuracy of a segmentation is to use a manually generated ground truth representing the expected segmentation result. This method can be used for benchmarking but it is not possible to get such a ground truth in operational systems. Therefore, we have proposed a new method to re-segment the iris so that the new segmentation can be seen as a ground truth of the expected segmentation. This original process, which we called Rough Region Segmentation (RRS), allows some traditional segmentation evaluation methods to be applied even in operational systems for which no ground truth is available.
- (iv) We also define a general framework to select and fuse segmentation quality metrics. We select individual quality metrics that showed a correlation with the iFNMR of images. As all the chosen metrics are correlated to the iFNMR of images, we propose to fuse them using a regression approach, i.e. predicting the iFNMR of a given image using the individual metrics. With this aim, we trained an SVR and used the predicted iFNMR as an indicator of segmentation quality. The interesting aspect of our approach is that it is a general framework that could be used with any other segmentation quality metrics.

The segmentation system proposed here fulfilled the objective of being able to segment irises acquired with low acquisition constraints. However, we see some points that could be improved in our current system :

First, the contour fitting and the normalization stages should receive important attention. Indeed, in systems based on Daugman's recognition algorithm [16], matching is done by aligning the features of specific points from the normalized image of the two irises. If the normalization is not done properly, the two textures, and therefore, the corresponding points do not correctly align and recognition performance is affected. In this thesis, we have proposed an effective way to model the iris border using ellipses. This approach has shown improvements compared to a similar circle-based formulation and reference elliptic-based models. However, this elliptic model is still an approximation of the "true" anatomic borders of the iris. Studying more flexible models that can handle non-rigid deformations should improve our system. However, we have noticed that it is more critical to have a robust fitting scheme than an accurate one. Therefore, increasing the contour's flexibility should not decrease the contour fitting's robustness.

On the other hand, our study on Active Contours for region segmentation showed that most contours considered were actually too flexible. This flexibility allows them to take almost any shape during the segmentation and thus, to produce very accurate segmentation masks. However, this flexibility can also lead to the generation of segmentation masks which are very different from the expected solution. For this reason, considering more constrained models should improve the final recognition performance of the whole processing stream even if the segmentation masks are less accurate.

Finally, in order to assess the quality of a segmented image, we have focused on the prediction of the images' iFNMR. This is an intuitive choice as the iFNMR represents to what extent an image is similar to other images of its own class, and a poorly segmented image is not similar to other images of its class (high iFNMR). Therefore, it is possible to identify large segmentation failures because this kind of image tends to look different from a correctly segmented image. However, during our study of segmentation algorithms, we noted that False Acceptances have more impact on the system's recognition performance than False Rejections. It would be interesting to be able to predict the tendency of images to produce False Acceptances as we did for False Rejections. Therefore, we would study the possibility to predict the images' iFMR as a complement to the approach we have presented for iFNMR. Having both a prediction of iFMR and iFNMR should allow a better characterization of segmentation failures. Nevertheless, it is more difficult to give an interpretation to such an approach because the iFMR of an image is influenced by other classes of images used during the training of the algorithm.

CONFIDENTIAL

Bibliography

- [1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1994.
- [2] A. S. Aguado, M. S. Nixon, and M. Montiel. Parameterizing arbitrary shapes via fourier descriptors for evidence-gathering extraction. *Computer Vision and Image Understanding*, 69(2):202 – 221, 1998.
- [3] U. Andreas and W. Peter. Weighted adaptive hough and ellipsoidal transforms for real-time iris segmentation. In *In Proceedings of the 5th IAPR/IEEE International Conference on Biometrics (ICB'12), New Delhi, India*, pages 1–7, 2012.
- [4] AOPTIX. InSight. <http://www.aoptix.com/>, 2012.
- [5] A. Blake and M. Isard. 3d position, attitude and shape input using video tracking of hands and lips. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94*, pages 185–192, New York, NY, USA, 1994. ACM.
- [6] B. E. Boser and et al. A training algorithm for optimal margin classifiers. In *PROCEEDINGS OF THE 5TH ANNUAL ACM WORKSHOP ON COMPUTATIONAL LEARNING THEORY*, pages 144–152. ACM Press, 1992.
- [7] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.
- [8] T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.
- [9] Chinese Academy of Sciences' Institute of Automation. CASIA V4 Iris Image Database . <http://biometrics.idealtest.org/>, 2011.
- [10] C.-T. Chou, S.-W. Shih, W.-S. Chen, V. Cheng, and D.-Y. Chen. Non-orthogonal view iris recognition system. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(3):417–430, March 2010.
- [11] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Underst.*, 53:211–218, March 1991.
- [12] S. Cotin, H. Delingette, N. Ayache, I. S. Antipolis, and R. D. Lucioles. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation, 2000.
- [13] S. Cremer. Adapting iris feature extraction and matching to the local and global quality of iris images, 2012.
- [14] S. Cremer, B. Dorizzi, S. Garcia, and N. Lemperiere. How a local quality measure can help improving iris recognition. *BIOSIG*, 2012.
- [15] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, 2002.
- [16] J. G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1148–1161, November 1993.
- [17] J. G. Daugman. New Methods in Iris Recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(5):1167–1175, September 2007.
- [18] T. Dietenbeck, M. Alessandrini, D. Barbosa, J. D'hooge, D. Friboulet, and O. Bernard. Detection of the whole myocardium in 2d-echocardiography for multiple orientations using a geometrically constrained level-set. *Medical Image Analysis*, 16(2):386 – 401, 2012.
- [19] C. Dorai and A. Jain. Cosmos-a representation scheme for free-form surfaces. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 1024–1029, June 1995.
- [20] M. Fairhurst and M. Erbilek. Analysis of physical ageing effects in iris biometrics. *Computer Vision, IET*, 5(6):358–366, November 2011.
- [21] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [22] M. Fitzgibbon, A. W. and Pilu and R. B. Fisher. Direct least-squares fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480, May 1999.
- [23] L. Flom and A. Safir. Iris recognition system. u.s. patent no. 4.641.349, 1987.
- [24] V. Hadziavdic. A comparative study of active contour models for boundary detection in brain images. Technical report, UNIVERSITY OF TROMSO, 2000.
- [25] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceeding Fourth Alvey Vision Conference*, pages 147–151, 1988.

- [26] Z. He, T. Tan, Z. Sun, and X. Qiu. Toward accurate and fast iris segmentation for iris biometrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1670–1684, September 2009.
- [27] M. Jacob, T. Blu, and M. Unser. Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13(9):1231–1244, 2004.
- [28] A. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, January 2004.
- [29] N. Kalka, N. Bartlow, and B. Cukic. An automated method for predicting iris segmentation failures. In *Proceedings of the 3rd IEEE international conference on Biometrics: Theory, applications and systems*, BTAS'09, pages 9–16, Piscataway, NJ, USA, 2009. IEEE Press.
- [30] N. Kalka, N. Bartlow, and B. Cukic. An automated method for predicting iris segmentation failures. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on*, pages 1–8, September 2009.
- [31] N. Kalka, J. Zuo, N. Schmid, and B. Cukic. Estimating and fusing quality factors for iris biometric images. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(3):509–524, May 2010.
- [32] M. Kamber, R. Shinghal, D. Collins, G. Francis, and A. Evans. Model-based 3-d segmentation of multiple sclerosis lesions in magnetic resonance brain images. *Medical Imaging, IEEE Transactions on*, 14(3):442–453, September 1995.
- [33] B. J. Kang and K. R. Park. A robust eyelash detection based on iris focus assessment. *Pattern Recognition Letter*, 28(13):1630–1639, 2007.
- [34] W. Kaplan. *Advanced Calculus*. Addison-Wesley, 4th ed. edition, 1991.
- [35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.
- [36] L. Kennell, R. Ives, and R. Gaunt. Binary morphology and local statistics applied to iris segmentation for recognition. In *Image Processing, 2006 IEEE International Conference on*, pages 293–296, October 2006.
- [37] A. W.-K. Kong and D. Zhang. Detecting eyelash and reflection for accurate iris segmentation. *IJPRAI*, pages 1025–1034, 2003.
- [38] S. Lee and W. V. University. *Quality of Iris Segmentation as a Predictor of Verification Performance*. West Virginia University, 2007.
- [39] P. Li, X. Liu, L. Xiao, and Q. Song. Robust and accurate iris segmentation in very noisy iris images. *Image and Vision Computing*, 28(2):246–253, 2010. Segmentation of Visible Wavelength Iris Images Captured At-a-distance and On-the-move.
- [40] X. Li, Z. Sun, and T. Tan. Comprehensive assessment of iris image quality. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3117–3120, September 2011.
- [41] X. Liu, K. Bowyer, and P. Flynn. Experiments with an improved iris segmentation algorithm. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 118–123, October 2005.
- [42] J.-L. Mallet. Discrete smooth interpolation in geometric modelling. *Computer-Aided Design*, 24(4):178–191, 1992.
- [43] J. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. LoIacono, S. Mangru, M. Tinker, T. Zappia, and W. Zhao. Iris on the move: Acquisition of images for iris recognition in less constrained environments. *Proceedings of the IEEE*, 94(11):1936–1947, November 2006.
- [44] T. Mcinerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1:91–108, 1996.
- [45] R. A. McLaughlin. Technical report - randomized hough transform: Improved ellipse detection with comparison. Technical report, 1997.
- [46] K. Miyazawa, K. Ito, T. Aoki, K. Kobayashi, and H. Nakajima. An effective approach for iris recognition using phase-based image matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30:1741–1756, October 2008.
- [47] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: Topology, geometry and deformation. *Image and Vision Computing*, 19:1023–1040, 2001.
- [48] J. Montagnat, H. Delingette, and P. Epidaure. Volumetric medical images segmentation using shape constrained deformable models. In *In Computer Vision, Virtual Reality and Robotics in Medicine*, pages 13–22. Springer Verlag, 1997.

- [49] A. Moreau-Gaudry, P. Cinquin, and J.-P. Baguet. Active model based carotid ultrasonic data segmentation. In *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '99, pages 176–183, London, UK, UK, 1999. Springer-Verlag.
- [50] B. Mory. *Interactive Segmentation of 3D Medical Images with Implicit Surfaces*. PhD thesis, Lausanne, 2011.
- [51] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [52] National Taiwan University. LibSVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [53] S. J. K. Pedersen. Circular Hough Transform. 2007.
- [54] P. Phillips, K. Bowyer, and P. Flynn. Comments on the casia version 1.0 iris data set. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1869–1870, October 2007.
- [55] P. Phillips, K. Bowyer, P. Flynn, X. Liu, and W. Scruggs. The iris challenge evaluation 2005. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–8, 29 2008-oct. 1 2008.
- [56] P. Phillips, W. Scruggs, A. O'Toole, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe. Frvt 2006 and ice 2006 large-scale experimental results. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):831–846, May 2010.
- [57] H. Proenca. Iris recognition: On the segmentation of degraded images acquired in the visible wavelength. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1502–1516, 2010.
- [58] H. Proenca. Short communciation: Iris recognition: Analysis of the error rates regarding the accuracy of the segmentation stage. *Image Vision Computation*, 28:202–206, January 2010.
- [59] H. Proenca. Quality assessment of degraded iris images acquired in the visible wavelength. *Information Forensics and Security, IEEE Transactions on*, 6(1):82–95, March 2011.
- [60] H. Proenca and L. Alexandre. Toward covert iris biometric recognition: Experimental results from the nice contests. *Information Forensics and Security, IEEE Transactions on*, 7(2):798–808, April 2012.
- [61] S. Pundlik, D. Woodard, and S. Birchfield. Non-ideal iris segmentation using graph cuts. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, June 2008.
- [62] K. Roy, P. Bhattacharya, and C. Y. Suen. Towards nonideal iris recognition based on level set method, genetic algorithms and adaptive asymmetrical svms. *Engineering Applications of Artificial Intelligence*, 24(3):458–475, April 2011.
- [63] W. Ryan, D. Woodard, A. Duchowski, and S. Birchfield. Adapting starburst for elliptical iris segmentation. pages 1–7, October 2008.
- [64] C. Sagiv, N. Sochen, and Y. Zeevi. Integrated active contours for texture segmentation. *Image Processing, IEEE Transactions on*, 15(6):1633–1646, June 2006.
- [65] S. Sanjay-Gopal and T. Hebert. Bayesian pixel classification using spatially variant finite mixtures and the generalized em algorithm. *Image Processing, IEEE Transactions on*, 7(7):1014–1028, July 1998.
- [66] S. Shah and A. Ross. Iris segmentation using geodesic active contours. *Information Forensics and Security, IEEE Transactions on*, 4:824–836, December 2009.
- [67] A. J. Smola and B. Schölkopf. A tutorial on support vector regression, 2004.
- [68] G. Sutra, S. Garcia-Salicetti, and B. Dorizzi. The viterbi algorithm at different resolutions for enhanced iris segmentation. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 310–316, April 2012.
- [69] G. Szekely and et al. Segmentation of 2-d and 3-d objects from mri volume data using constrained elastic deformations of flexible fourier contour and surface models, 1996.
- [70] E. Tabassi. Image specific error rate: A biometric performance metric. In *ICPR*, pages 1124–1127, 2010.
- [71] E. Tabassi, P. J. Grother, and W. J. Salamon. Iris quality calibration and evaluation (iqce): Evaluation report. NIST Interagency/Internal Report (NISTIR) - 7820, September 2011.
- [72] T. Tan, Z. He, and Z. Sun. Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition. *Image and Vision Computing*, 28(2):223–230, 2010. Segmentation of Visible Wavelength Iris Images Captured At-a-distance and On-the-move.
- [73] D. Terzopoulou, J. Platt, A. Barr, and K. Fleischert. Elastically deformable models. *Computer Graphics*, 21:205–214, 1987.
- [74] Unique Identification Authority of India. UIDAI. <http://uidai.gov.in/>, 2011.
- [75] V. Vapnik, S. E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems 9*, pages 281–287. MIT Press, 1996.

- [76] M. Vatsa, R. Singh, and A. Noore. Improving iris recognition performance using segmentation, quality enhancement, match score fusion, and indexing. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(4):1021–1035, aug. 2008.
- [77] R. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, September 1997.
- [78] A. Yuille, D. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on*, pages 104–109, June 1989.
- [79] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.
- [80] H. Zhang, Z. Sun, and T. Tan. Statistics of local surface curvatures for mis-localized iris detection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4097–4100, September 2010.
- [81] Z. Zhou, Y. Du, and C. Belcher. Transforming traditional iris recognition systems to work in nonideal situations. *Industrial Electronics, IEEE Transactions on*, 56(8):3203–3213, August 2009.
- [82] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:884–900, September 1996.
- [83] J. Zuo, N. Kalka, and N. Schmid. A robust iris segmentation procedure for unconstrained subject presentation. In *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pages 1–6, August 2006.
- [84] J. Zuo and N. A. Schmid. On a methodology for robust segmentation of nonideal iris images. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40:703–718, June 2010.

CONFIDENTIAL

Part III

Appendix

CONFIDENTIAL

CONFIDENTIAL

Appendix A

Modular System for Iris Segmentation

In this Appendix, we describe the complete segmentation system that we have developed during this thesis. This system takes as input a row eye image acquired under Near Infrared (NIR) illumination and produces as output a normalized image for recognition. This segmentation system integrates the key algorithms we have described in Chapters 4 to 7, but also some classical algorithms for iris segmentation. As these algorithms do not show a clear novelty compared to the state of the art, we propose to describe them in this Appendix instead of in the core part of this manuscript.

A.1 System Overview

Figure 83 summarizes our system's work-flow and the connection between the different modules. These modules are :

Image Acquisition The image is acquired under Near Infrared illumination. This acquisition can be made with different sensors and different illumination devices introducing variability in the input images.

Preprocessing We first apply some global correction on the image to limit the influence of the sensor.

Pupil Segmentation The pupil segmentation aims at finding the pupil region in the eye and the related normalization contour. The process is divided into three sub-steps :

- (i) Rough localization of the pupil area in the image (Chapter 4)
- (ii) Accurate segmentation of the pupil region using B-Snakes (Section A.3.2 of this Appendix)
- (iii) Fitting of an ellipse on the pupil borders (Chapter 6)

Iris Segmentation The iris segmentation aims at locating the iris region in the eye and the related normalization contour. In analogy to the pupil segmentation module, we divide the process into three sub-steps :

- (i) Building of a rough region of interest containing the iris region (Section A.4.1 of this Appendix)
- (ii) Accurate segmentation of the iris region using one of the segmentation algorithms we describe in Chapter 5.
- (iii) Fitting of an ellipse on the iris borders (Chapter 6)

Segmentation Quality This module is dedicated to the detection of poorly segmented images. If the segmentation process fails, the image should be discarded or re-segmented (Chapter 7).

Image Normalization The iris texture and the normalization mask are unwrapped using the normalization contours (Section 1.3 of Chapter 1).

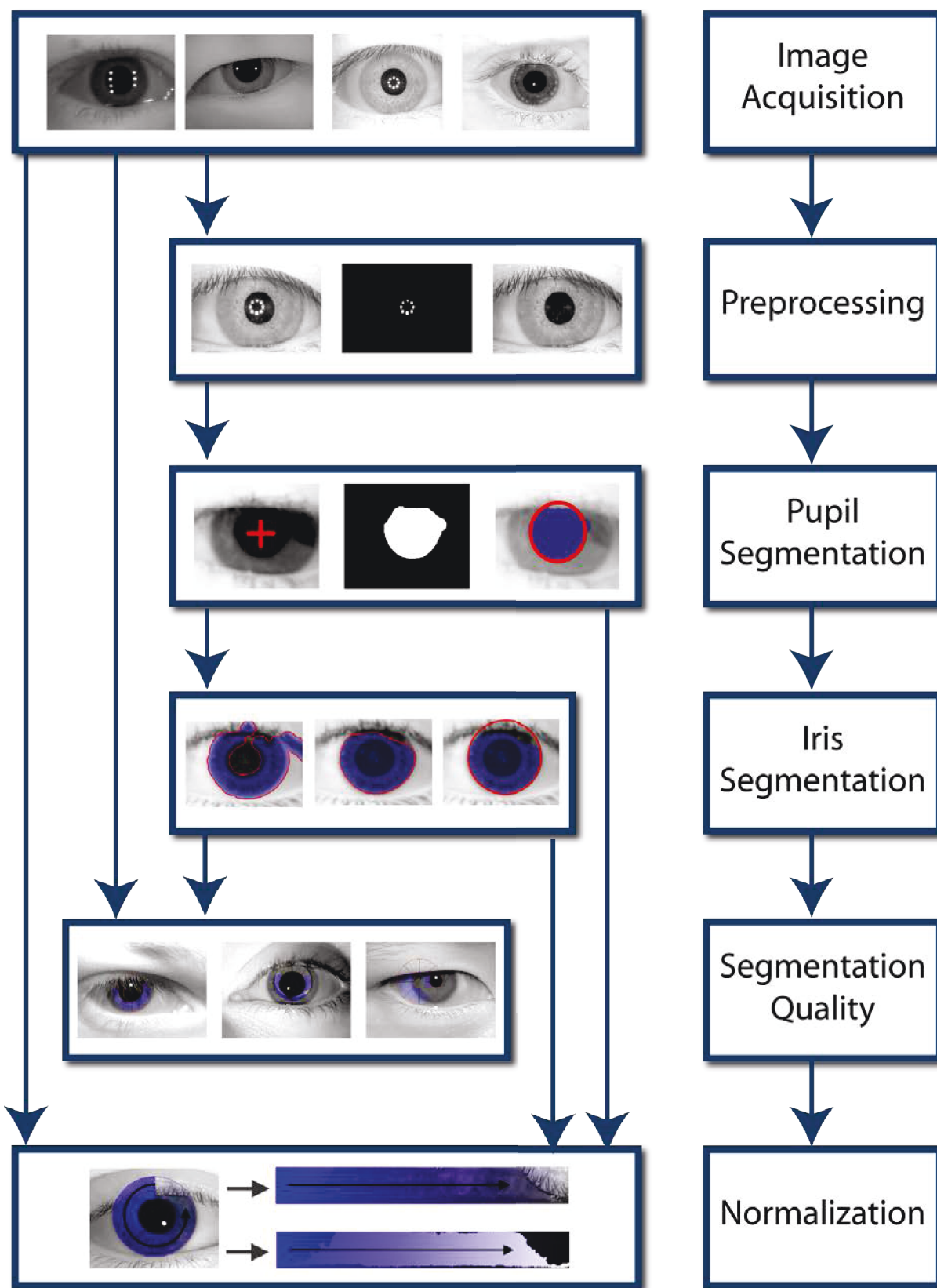


Figure 83: Complete work-flow for iris segmentation. The arrows indicate the inputs and outputs of every module.

A.2 Preprocessing

In operational systems, the image can be acquired by various sensors and under varying illumination conditions. In order to make our algorithms less sensitive to this variability it is necessary to apply some global corrections to the image.

We consider two main corrections, first we remove the specular reflections created by the NIR illumination and then we apply a global contrast correction depending on the sensor properties.

A.2.1 Specular reflections suppression

Illumination in NIR uses diodes to highlight the iris texture. However, these diodes create specular reflections appearing like white spots in the image (see Figure 84).

These spots are disturbing for segmentation because they create very high gradients in areas expected to be homogenous like the pupil. For this reason it is common to remove these reflections before the segmentation stage [84] [26].

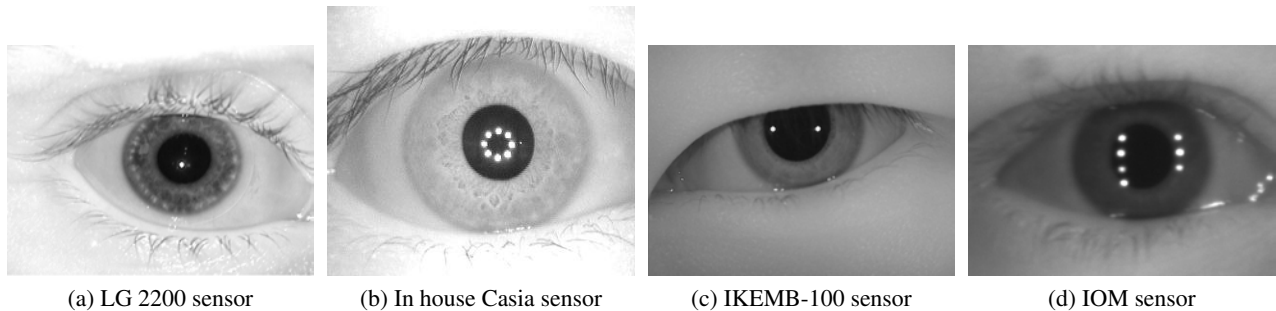


Figure 84: Specular reflections for various sensors

A.2.1.1 Spot Detection

The shape of the specular reflection depends on the sensor and the illumination device used to acquire the image (See Figure 84). However, they always appear as very white pixels in the image. Some other regions may have white intensities such as the skin or the limbus, so simple thresholding techniques are not sufficient to identify the reflections.

We apply a local contrast enhancement to highlight spots. These reflections are very white spots present in dark regions of the image (pupil). Hence, local increase of the contrast highlights these regions whereas the skin and limbus are left unchanged. We apply the following process to every pixel \mathbf{x} in the image :

- Compute the mean $M(\mathbf{x})$ of every pixel \mathbf{x} in the image estimated in a 5×5 neighborhood.
- Apply the transform $I'(\mathbf{x}) = I(\mathbf{x}) + k(I(\mathbf{x}) - M(\mathbf{x}))$ to every pixel \mathbf{x} in the image (with $I(\mathbf{x}) \in \mathbb{R}$).

If k is negative, this transform tend to smooth the image (for $k = -1$, $I'(\mathbf{x}) = M(\mathbf{x})$, i.e. a mean filtering). If k is positive, then the contrast of the image is locally increased (term $k(I(\mathbf{x}) - M(\mathbf{x}))$ correspond to the local contrast which is added to the original image intensity). We take $k = 2$, which heavily increases the local contrast. Then, it is possible to segment the region of the specular reflections with simple thresholding methods.

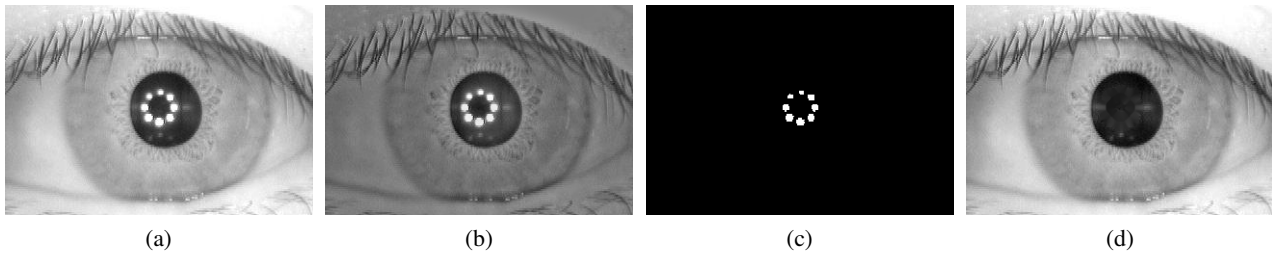


Figure 85: Specular reflections removal : Original image (a). Image after contrast enhancement (b). Thresholded image (c). Image after hole filling (d).

A.2.1.2 Hole filling

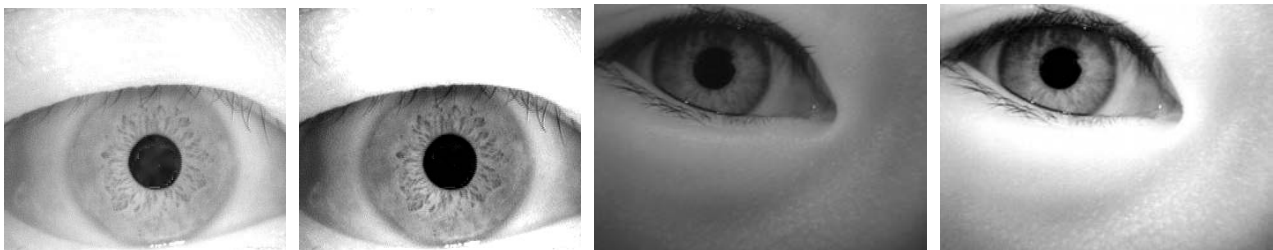
Once the spots are segmented, it is necessary to remove them from the original image. The most common technique is to use an inPainting algorithm, aiming at filling the holes in the image with the surrounding information (see [84] for details).

The method we implement uses the same idea but without the accuracy of the inPainting algorithm. The purpose is only to roughly fill the holes, and not to reconstruct precisely the missing information. Our idea is to propagate the mean from the outer regions to the inner ones :

1. Select the region to fill.
2. For each point on the border of the region, look for connected pixels where intensity is known and affect their mean to the current pixel.
3. Erode the region to fill by one pixel (i.e. remove the border of the region).
4. Go back to the first step until there is no more region to fill.

A.2.2 Intensity adaptation

Different sensors can produce images with very different intensity distributions. For instance, the image in Figure 86a has a good contrast between the pupil and the iris, whereas the contrast between the limbus and the iris is rather low. On the opposite, the image displayed in Figure 86c is acquired with a sensor generating darker images, and the contrast between the pupil and the iris is very low whereas the contrast between the limbus and the iris is quite good.



(a) Original image of Casia-Interval Database (b) Corrected image of Casia-Interval Database (c) Original image of Casia-Lamp Database (d) Corrected image of Casia-Lamp Database

Figure 86: Figures (a) (c): Original images from databases Casia-Interval and Casia-Twins. Figures (b) (d) : Images after application of the process presented in Section A.2.2 ($\gamma = 0.7$ for Casia-Interval and $\gamma = 1.3$ for Casia-Twins).

In order to reduce the sensitivity of the following algorithms to this intensity variability we apply a global contrast correction to the image depending on the nature of the acquisition sensor :

$$I'(\mathbf{x}) = I(\mathbf{x})^{\frac{1}{\gamma}}. \quad (133)$$

For sensor producing bright images, a gamma below 1 is used, and for sensors producing dark images, a gamma value above 1 is applied.

Finally, we apply an intensity stretching on the histogram to ensure that the intensity range is fully used. Moreover, we crop the 5% highest and lowest intensities in order to correct some artifacts and to ensure that the pupil is one of the darkest element in the image and the skin is the lighter one. Results of this transformation are shown in Figure 86b and Figure 86d.

A.3 Pupil Segmentation

The pupil is one of the darkest elements of the eye image acquired in NIR illumination. For this reason it is common to segment the pupil first. The segmentation of the pupil can be divided into three sub-problems illustrated in Figure 87. The first, and maybe most important step, is the rough localization of the pupil. Then the pupil region is accurately segmented. Finally a parametric contour is fitted to estimate the pupil borders under the occlusions.

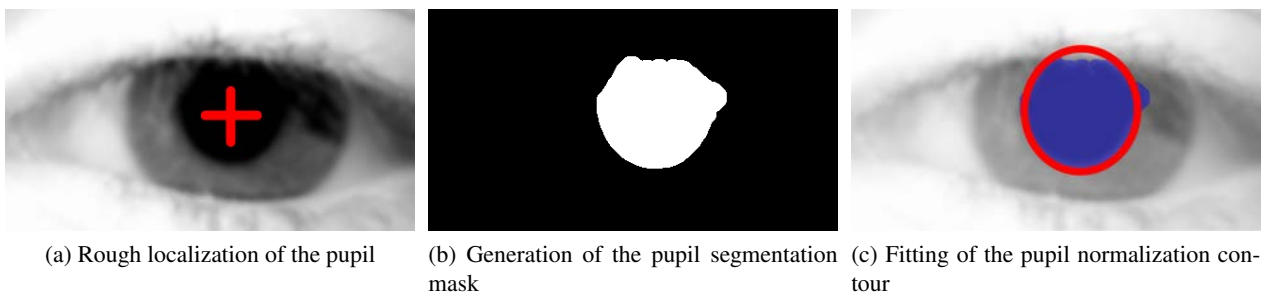


Figure 87: The three sub-steps of the pupil segmentation.

A.3.1 Rough Pupil Localization

The pupil localization aims at finding the rough location of the pupil area in the image. This task is critical because any mistake made at this point can no longer be corrected afterward. The key aspect of this step is to distinguish the pupil from other dark elements in the eye such as eyelashes and eyebrows. We addressed this issue in Chapter 4.

A.3.2 Pupil Region Segmentation

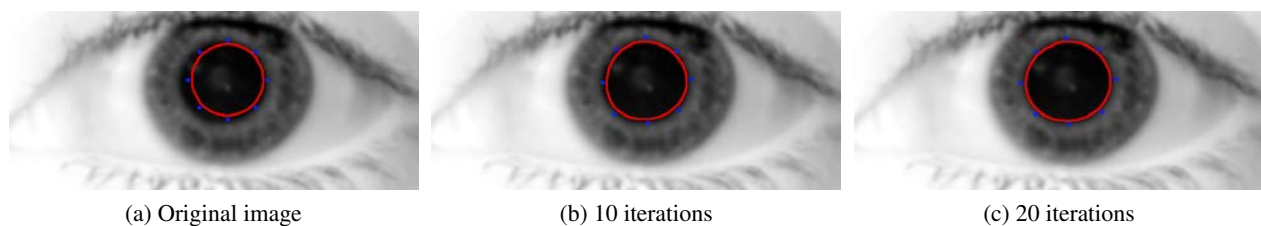


Figure 88: Fine pupil segmentation based on B-snakes.

The pupil usually has very sharp and clear borders, therefore segmentation of the pupil is quite simple once the pupil area has been identified.

Thresholding methods are a popular way to segment the pupil, but from our experiments, these techniques can lead to inaccurate segmentation when the upper eyelid or eyelashes touch the pupil. Accordingly, we use of B-Snakes described in detail for iris segmentation in Section 5.2.2 of Chapter 5.

A.3.3 Pupil Contour Fitting

The last step of pupil segmentation is estimating a parametric contour describing pupil border under possible occlusions. This task is challenging for the pupil, not because of the occlusions, but because the pupil may anatomically have a non-circular shape. In Chapter 6, we presented the module we developed to efficiently and robustly fit ellipses to the pupil borders.

A.4 Iris Segmentation

The iris segmentation follows the pupil segmentation. Therefore we can use the pupil information as a clue for the location of the iris region. Like we did for the pupil, we divide the process into three sub-problems. We first use the information provided by the pupil to build a region of interest where we expect to find the iris. This rough region of interest is followed by an accurate segmentation of the iris texture using algorithms derived from the active contours models. Finally we fit a parametric contour to the iris borders.

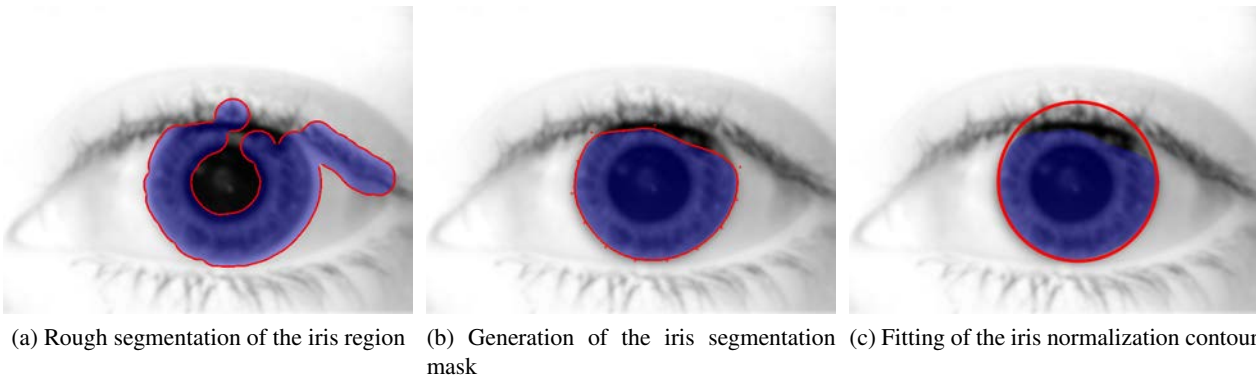


Figure 89: The three sub-steps of the iris segmentation.

A.4.1 Iris Location

In order to build this region of interest we use the pupil location. We build a rough estimation of the iris histogram and use it to estimate a region of interest including the iris region.

A.4.1.1 Iris Histogram estimation

Based on the pupil location, we consider three sub-regions where we expect to have a border between the iris and the limbus : left, right and down (we do not consider the upper part, as this area is often occluded by eyelashes). See Figure 90.

The idea is to find for every region the border maximizing the distance between the inner and outer histogram (the iris histogram and the background histogram). To that aim, we use the Bhattacharyya distance :

$$d_B(p_i, p_b) = \int \sqrt{p_i(x)p_b(x)} dx, \quad (134)$$

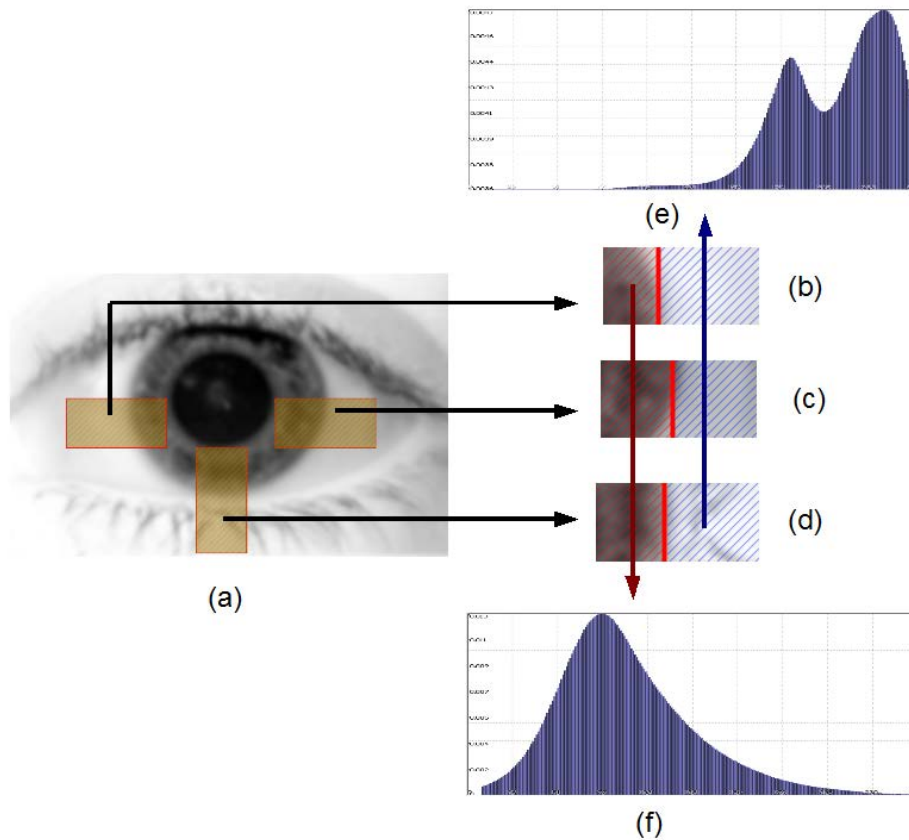


Figure 90: Estimation of the iris and background histograms. (a) : Original image. (b) (c) (d) : extracted sub-regions expected to have an edge between the iris and the limbus. For each of these regions, the abscissa best separating the background and the iris are computed (displayed in red in the image). The histograms of the three inner and outer regions are averaged and normalized in order to compute the estimated histogram for the iris (Figure (f)) and the background (Figure (e)).

where p_i is the probability distribution if iris intensities and p_b the background one. This distance is between 0 and 1, value of 0 means completely different distributions, and value of 1 means identical distributions.

Once the best separation has been found for the three sub-images, we average the three inner and outer histograms to generate an estimation of the iris and background histograms.

A.4.1.2 Region Of Interest

For every pixel in the image, we compute the log likelihood w.r.t. the two probabilities previously computed :

$$\begin{aligned}
 L(I(\mathbf{x})) &= \log\left(\frac{p_i(I(\mathbf{x}))}{p_b(I(\mathbf{x}))}\right), \\
 &= \log(p_i(I(\mathbf{x}))) - \log(p_b(I(\mathbf{x}))),
 \end{aligned}
 \tag{135}$$

p_i is the probability distribution of the iris intensities and p_b correspond to the background intensities. If the log likelihood for a given pixel is positive, the pixel is likely to belong to the iris, otherwise it is likely that it belongs to the background.

In the end, a simple thresholding around 0 of the log-likelihood image gives an estimation of the iris region. Finally, we apply a binary dilatation of the mask of the region of interest to ensure that the iris region is included in the mask.

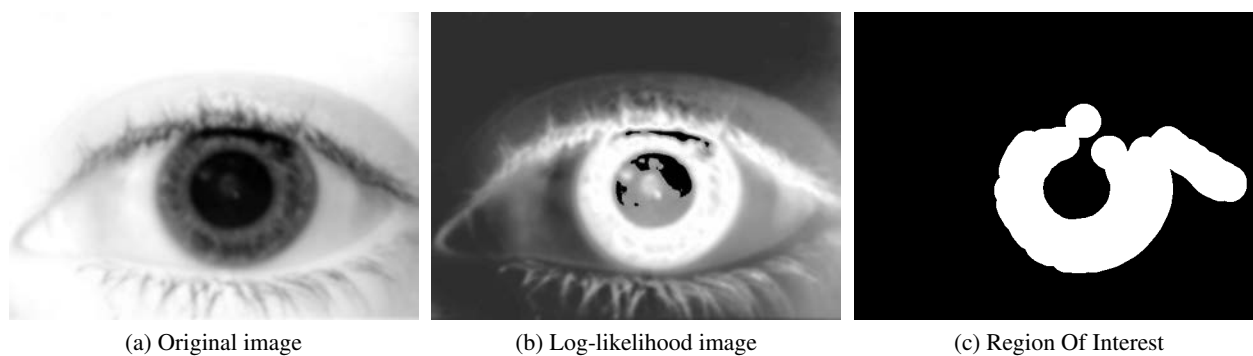


Figure 91: Building of the iris Region Of Interest.

A.4.2 Iris Region Segmentation

The segmentation of the iris region is a critical part of the complete segmentation process because the resulting segmentation mask indicates areas not belonging to the iris and which should be excluded from the recognition process. This task is challenging because the algorithm should not only reach a good precision, but they must also be very robust to handle a large variability in the acquisition conditions.

In order to solve this problem we implemented several Active Contours described in Chapter 5. The Region of Interest we found in the previous stage is used to initialize the region segmentation process and to reduce the computation time by restraining the process to this region.

A.4.3 Iris Contour Fitting

Like for the pupil, the last step of the iris segmentation is to find a parametric contour describing the iris borders under the occlusions. Unlike the pupil, the anatomic iris borders are usually rather close to a circle but the iris texture is very likely to be occluded by eyelids and eyelashes. We present a robust method which we developed to fit ellipses to the iris borders in Chapter 6.

A.5 Segmentation Quality

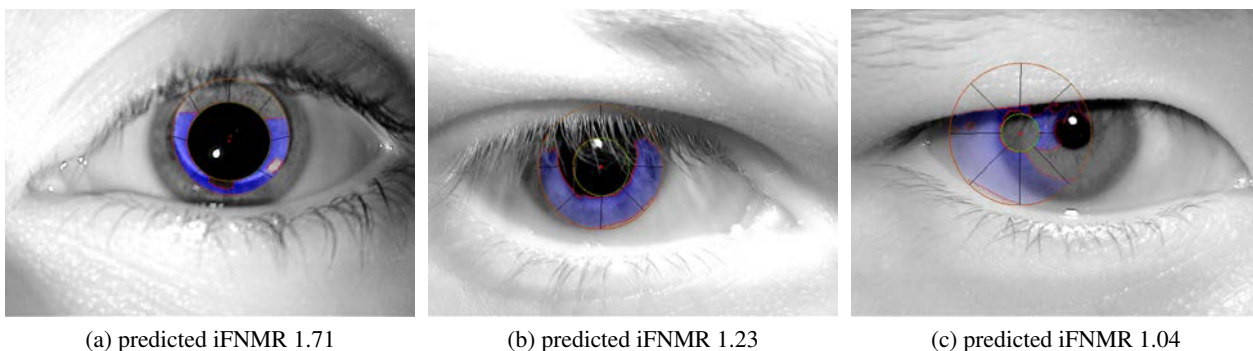


Figure 92: Example segmented images with bad segmentation quality scores

The assessment of the image quality is a critical issue in iris recognition system. Traditionally, the algorithms estimating the image quality are often placed either at the image acquisition level to check the quality of the raw image, or after the segmentation module to assess if the biometric sample has enough information to characterize the iris. Our proposition differs from the two traditional approaches in the sense that we aim at evaluating the image segmentation independently.

To that aim, we trained a SVR to predict the iFNMR of a segmented image using a set of new quality measures. We describe these measures and the training of the SVR in the Chapter 7. If the quality score of a given segmentation is too bad (like for the images of Figure 92), the image should be reacquired or re-segmented, with new settings.

CONFIDENTIAL

Appendix B

Variational Contour Optimization

In this appendix, we give the explicit derivation of energetic equations presented in Chapter 6. We first give the general formulation for ellipses and then describe the circle as a special case of ellipses, leading to some simplification in the equations.

B.1 General elements

As explained in Chapter 6, an ellipse $\mathcal{E}(\mathbf{x}_c, a, b, \phi)$ can be parametrized by equation :

$$\mathbf{x}(\theta) = \mathbf{x}_c + a \cos(\theta) \vec{n}_\phi + b \sin(\theta) \vec{t}_\phi, \quad (136)$$

for $\theta \in [0; 2\pi[$. Given this parametrization, the normal and tangential vectors are :

$$\begin{aligned} \vec{T}(\theta) &= \frac{d}{d\theta} \mathbf{x}(\theta), \\ &= -a \sin(\theta) \vec{n}_\phi + b \cos(\theta) \vec{t}_\phi, \end{aligned} \quad (137)$$

$$\vec{N}(\theta) = -b \cos(\theta) \vec{n}_\phi - a \sin(\theta) \vec{t}_\phi, \quad (138)$$

with the following properties :

$$\|\vec{T}(\theta)\| = \|\vec{N}(\theta)\| = \sqrt{a^2 \sin^2(\theta) + b^2 \cos^2(\theta)} \quad (139)$$

And the unit tangential and normal vectors are :

$$\begin{aligned} \vec{n}_\theta &= \frac{\vec{N}(\theta)}{\|\vec{N}(\theta)\|}, \\ \vec{t}_\theta &= \frac{\vec{T}(\theta)}{\|\vec{T}(\theta)\|}. \end{aligned} \quad (140)$$

B.2 Edge energy E_{edge}

As explained earlier, E_{edge} is :

$$E_e(\mathcal{E}) = E_n(\mathcal{E}) + E_t(\mathcal{E}), \quad (141)$$

with:

$$\begin{aligned} E_n(C) &= \int_C \left\| \nabla I(\mathbf{x}) \cdot \vec{n}(\mathbf{x}) \right\| \frac{d\mathbf{x}}{\|C'(\mathbf{x})\|}, \\ E_t(C) &= - \int_C \left\| \nabla I(\mathbf{x}) \cdot \vec{t}(\mathbf{x}) \right\| \frac{d\mathbf{x}}{\|C'(\mathbf{x})\|}. \end{aligned} \quad (142)$$

Considering that we have : $d\mathbf{x} = \|\vec{T}(\theta)\| d\theta = \|C'(\theta)\| d\theta$ for the parametrization of equation (136), the above equations become :

$$\begin{aligned} E_n(\mathcal{E}) &= \int_0^{2\pi} \left\| \nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right\| d\theta, \\ E_t(\mathcal{E}) &= - \int_0^{2\pi} \left\| \nabla I(\mathbf{x}) \cdot \vec{t}_\theta \right\| d\theta. \end{aligned} \quad (143)$$

So we have :

$$\begin{aligned}
\frac{\partial E_n}{\partial \alpha} &= \int_0^{2\pi} \frac{\partial}{\partial \alpha} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] d\theta, \\
&= \int_0^{2\pi} \left[\left[\frac{\partial \nabla I(\mathbf{x})}{\partial \alpha} \right] \cdot \vec{n}_\theta + \left[\nabla I(\mathbf{x}) \cdot \frac{\partial \vec{n}_\theta}{\partial \alpha} \right] \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] \right] d\theta, \\
&= \int_0^{2\pi} \left[\left[H(\mathbf{x}) \frac{\partial \vec{x}}{\partial \alpha} \right] \cdot \vec{n}_\theta + \left[\nabla I(\mathbf{x}) \cdot \frac{\partial \vec{n}_\theta}{\partial \alpha} \right] \right] \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] d\theta,
\end{aligned} \tag{144}$$

where $H(\mathbf{x})$ is the Hessian of the image I at point \mathbf{x} .

Considering :

$$\text{dir}(\vec{n}_\theta) = \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right], \tag{145}$$

$$\text{dir}(\vec{t}_\theta) = \text{sign} \left[\nabla I(\mathbf{x}) \cdot \vec{t}_\theta \right], \tag{146}$$

we finally have, for an arbitrary parameter α :

$$\begin{aligned}
\frac{\partial E_n}{\partial \alpha} &= \int_0^{2\pi} \left[\left[H(\mathbf{x}) \frac{\partial \vec{x}}{\partial \alpha} \right] \cdot \vec{n}_\theta + \left[\nabla I(\mathbf{x}) \cdot \frac{\partial \vec{n}_\theta}{\partial \alpha} \right] \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_t}{\partial \alpha} &= - \int_0^{2\pi} \left[\left[H(\mathbf{x}) \frac{\partial \vec{x}}{\partial \alpha} \right] \cdot \vec{t}_\theta + \left[\nabla I(\mathbf{x}) \cdot \frac{\partial \vec{t}_\theta}{\partial \alpha} \right] \right] \text{dir}(\vec{t}_\theta) d\theta.
\end{aligned} \tag{147}$$

We have the following partial derivatives for $\frac{\partial \mathbf{x}}{\partial \alpha}$, $\frac{\partial \vec{n}_\theta}{\partial \alpha}$ and $\frac{\partial \vec{t}_\theta}{\partial \alpha}$:

$$\begin{aligned}
\frac{\partial \mathbf{x}}{\partial \mathbf{x}_c} &= I_2, \\
\frac{\partial \mathbf{x}}{\partial a} &= \cos(\theta) \vec{n}_\phi, \\
\frac{\partial \mathbf{x}}{\partial b} &= \sin(\theta) \vec{t}_\phi, \\
\frac{\partial \mathbf{x}}{\partial \phi} &= -b \sin(\theta) \vec{n}_\phi + a \cos(\theta) \vec{t}_\phi,
\end{aligned} \tag{148}$$

$$\begin{aligned}
\frac{\partial \vec{n}_\theta}{\partial \mathbf{x}_c} &= 0, \\
\frac{\partial \vec{n}_\theta}{\partial a} &= -\frac{\sin(\theta)}{\|N(\theta)\|} \left[\vec{t}_\phi + a \vec{n}_\phi \frac{\sin(\theta)}{\|N(\theta)\|} \right], \\
\frac{\partial \vec{n}_\theta}{\partial b} &= -\frac{\cos(\theta)}{\|N(\theta)\|} \left[\vec{n}_\phi + b \vec{n}_\phi \frac{\cos(\theta)}{\|N(\theta)\|} \right], \\
\frac{\partial \vec{n}_\theta}{\partial \phi} &= -\vec{t}_\phi,
\end{aligned} \tag{149}$$

$$\begin{aligned}
\frac{\partial \vec{t}_\theta}{\partial \mathbf{x}_c} &= 0, \\
\frac{\partial \vec{t}_\theta}{\partial a} &= -\frac{\sin(\theta)}{\|T(\theta)\|} \left[\vec{n}_\phi + a \vec{t}_\phi \frac{\sin(\theta)}{\|T(\theta)\|} \right], \\
\frac{\partial \vec{t}_\theta}{\partial b} &= -\frac{\cos(\theta)}{\|T(\theta)\|} \left[-\vec{t}_\phi + b \vec{t}_\phi \frac{\cos(\theta)}{\|T(\theta)\|} \right], \\
\frac{\partial \vec{t}_\theta}{\partial \phi} &= \vec{n}_\phi,
\end{aligned} \tag{150}$$

So :

$$\begin{aligned}
\frac{\partial E_n}{\partial \mathbf{x}_c} &= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_n}{\partial a} &= \int_0^{2\pi} \left[\cos(\theta) \left[H(\mathbf{x}) \vec{n}_\phi \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\sin(\theta)}{\|N(\theta)\|} \left[\vec{t}_\phi + a \vec{n}_\theta \frac{\sin(\theta)}{\|N(\theta)\|} \right] \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_n}{\partial b} &= \int_0^{2\pi} \left[\sin(\theta) \left[H(\mathbf{x}) \vec{t}_\phi \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\cos(\theta)}{\|N(\theta)\|} \left[\vec{n}_\phi + b \vec{n}_\theta \frac{\cos(\theta)}{\|N(\theta)\|} \right] \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_n}{\partial \phi} &= \int_0^{2\pi} \left[\left[H(\mathbf{x}) (-b \sin(\theta) \vec{n}_\phi + a \cos(\theta) \vec{t}_\phi) \right] \cdot \vec{n}_\theta - \nabla I(\mathbf{x}) \cdot \vec{t}_\theta \right] \text{dir}(\vec{n}_\theta) d\theta,
\end{aligned} \tag{151}$$

And :

$$\begin{aligned}
\frac{\partial E_t}{\partial \mathbf{x}_c} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{t}_\theta \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_t}{\partial a} &= - \int_0^{2\pi} \left[\cos(\theta) \left[H(\mathbf{x}) \vec{n}_\phi \right] \cdot \vec{t}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\sin(\theta)}{\|T(\theta)\|} \left[\vec{n}_\phi + a \vec{t}_\theta \frac{\sin(\theta)}{\|T(\theta)\|} \right] \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_t}{\partial b} &= - \int_0^{2\pi} \left[\sin(\theta) \left[H(\mathbf{x}) \vec{t}_\phi \right] \cdot \vec{t}_\theta - \nabla I(\mathbf{x}) \cdot \frac{\cos(\theta)}{\|T(\theta)\|} \left[-\vec{t}_\phi + b \vec{t}_\theta \frac{\cos(\theta)}{\|T(\theta)\|} \right] \right] \text{dir}(\vec{n}_\theta) d\theta, \\
\frac{\partial E_t}{\partial \phi} &= - \int_0^{2\pi} \left[\left[H(\mathbf{x}) (-b \sin(\theta) \vec{n}_\phi + a \cos(\theta) \vec{t}_\phi) \right] \cdot \vec{t}_\theta + \nabla I(\mathbf{x}) \cdot \vec{n}_\theta \right] \text{dir}(\vec{n}_\theta) d\theta,
\end{aligned} \tag{152}$$

B.3 Region energy E_r

B.3.1 General context

We have :

$$E_r(\mathcal{E}) = - \int_{R_{in}} \log p_{in}(I(\mathbf{x})) d\mathbf{x} - \int_{R_{out}} \log p_{out}(I(\mathbf{x})) d\mathbf{x}. \tag{153}$$

where p_{in} and p_{out} are the probabilities of a given pixel to be inside or outside the contour.

To handle the minimization, we will consider the Chan and Vese [8] framework we described in Chapter 5. In this framework the object to be segmented is described using the Heaviside function H . The above problem can then be written as :

$$E_r(\mathcal{E}) = - \int_{\Omega} H(\Phi(\mathbf{x})) D(\mathbf{x}) d\mathbf{x}, \tag{154}$$

with Φ a function being positive inside the object and negative outside and :

$$D(\mathbf{x}) = - \frac{\log p_{in}(I(\mathbf{x}))}{\log p_{out}(I(\mathbf{x}))}. \tag{155}$$

The key idea is to select a function Φ whose level set are ellipses. We have considered the 2D Gaussian function :

$$\begin{aligned}
\Phi(\mathbf{x}) &= \exp \left(- (a'(x - x_c)^2 + 2b'(x - x_c)(y - y_c) + c'(y - y_c)^2) \right) - C, \\
\Phi(\mathbf{x}) &= \exp \left(- ((\mathbf{x} - \mathbf{x}_c)^T A (\mathbf{x} - \mathbf{x}_c)) \right) - C, \\
\Phi(\mathbf{x}) &= g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c) - C,
\end{aligned} \tag{156}$$

with $C \in]0, 1[$ and :

$$\begin{aligned}
a' &= \frac{\cos^2 \phi}{2\sigma_x^2} + \frac{\sin^2 \phi}{2\sigma_y^2}, \\
b' &= \frac{-\sin 2\phi}{4\sigma_x^2} + \frac{\sin^2 2\phi}{4\sigma_y^2}, \\
c' &= \frac{\sin^2 \phi}{2\sigma_x^2} + \frac{\cos^2 \phi}{2\sigma_y^2},
\end{aligned} \tag{157}$$

$$\begin{aligned}
A &= \begin{pmatrix} a' & b' \\ b' & c' \end{pmatrix}, \\
&= \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} \frac{1}{2\sigma_x^2} & 0 \\ 0 & \frac{1}{2\sigma_y^2} \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix}, \\
&= R^T(\phi)\Sigma(\sigma_x, \sigma_y)R(\phi).
\end{aligned} \tag{158}$$

The 0 level set of the Φ function is an ellipse with parameters $(\mathbf{x}_c, a, b, \phi)$. The relation between the center \mathbf{x}_c and this angle of the ellipse ϕ is straightforward. To show the relation between (σ_x, σ_y) and (a, b) we consider the case $\phi = 0$ in (156):

$$\begin{aligned}
\exp\left(-\left(\frac{(x-x_c)^2}{2\sigma_x^2} + \frac{(y-y_c)^2}{2\sigma_y^2}\right)\right) &= C, \\
\left(\frac{(x-x_c)^2}{2\sigma_x^2} + \frac{(y-y_c)^2}{2\sigma_y^2}\right) &= -\log(C), \\
\frac{(x-x_c)^2}{(\sqrt{-2\log(C)}\sigma_x)^2} + \frac{(y-y_c)^2}{(\sqrt{-2\log(C)}\sigma_y)^2} &= 1.
\end{aligned} \tag{159}$$

Setting the arbitrary constant $C = \exp(-\frac{1}{2})$ we get :

$$\begin{aligned}
\sigma_x &= a, \\
\sigma_y &= b.
\end{aligned} \tag{160}$$

Going back to the energy formulation, we consider the partial derivative w.r.t. an arbitrary parameter α :

$$\begin{aligned}
\frac{\partial E_r}{\partial \alpha} &= -\int_{\Omega} \delta(\Phi(\mathbf{x})) \frac{\partial}{\partial \alpha} (\Phi(\mathbf{x})) D(\mathbf{x}) d\mathbf{x}, \\
&= -\int_{\Phi=0} \frac{1}{\|\nabla(\Phi(\mathbf{x}))\|} \frac{\partial \Phi(\mathbf{x})}{\partial \alpha} D(\mathbf{x}) d\mathbf{x}.
\end{aligned} \tag{161}$$

B.3.2 Geometric Gradient Calculation

We have :

$$\begin{aligned}
\frac{\partial \Phi}{\partial \mathbf{x}} &= -\frac{\partial A}{\partial \mathbf{x}} g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c), \\
&= -(R^T \Sigma R(\mathbf{x} - \mathbf{x}_c) + (\mathbf{x} - \mathbf{x}_c)^T R^T \Sigma R) g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c), \\
&= -2A(\mathbf{x} - \mathbf{x}_c) g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c), \\
&= -2(\mathbf{x} - \mathbf{x}_c)^T A g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c),
\end{aligned} \tag{162}$$

So :

$$\|\nabla(\Phi(\mathbf{x}))\| = 2 \|A(\mathbf{x} - \mathbf{x}_c)\| g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \tag{163}$$

Considering that \mathbf{x} belongs to the 0 level of the Φ function, it can be parametrized for $\theta \in [0; 2\pi[$:

$$\mathbf{x}(\theta) = \mathbf{x}_c + R^T \Sigma^{-1} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}. \tag{164}$$

We have :

$$\begin{aligned} \|A(\mathbf{x} - \mathbf{x}_c)\| &= \left\| R^T \Sigma R R^T \Sigma^{-1} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \right\|, \\ &= \left\| R^T \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \right\|, \\ &= 1. \end{aligned} \quad (165)$$

So, for a point \mathbf{x} on the 0 level of the Φ function, we have :

$$\|\nabla(\Phi(\mathbf{x}))\| = 2g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \quad (166)$$

B.3.3 Partial derivatives

In the same way we get equation (162), we have :

$$\frac{\partial \Phi}{\partial \mathbf{x}_c} = 2A(\mathbf{x} - \mathbf{x}_c)g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \quad (167)$$

We now consider :

$$\begin{aligned} \frac{\partial \Phi}{\partial \sigma_x} &= -(\mathbf{x} - \mathbf{x}_c)^T R^T(\phi) \begin{pmatrix} -\frac{1}{4\sigma_x^3} & 0 \\ 0 & 0 \end{pmatrix} R(\phi)(\mathbf{x} - \mathbf{x}_c)g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c), \\ &= \frac{1}{4\sigma_x^3} (\cos(\phi)(x - x_c) + \sin(\phi)(y - y_c))^2 g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c), \\ &= \frac{1}{4\sigma_x^3} ((\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n}_\phi)^2 g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \end{aligned} \quad (168)$$

In the same way, we show :

$$\frac{\partial \Phi}{\partial \sigma_y} = \frac{1}{4\sigma_y^3} ((\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{t}_\phi)^2 g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \quad (169)$$

For the partial derivative w.r.t. ϕ we have :

$$\begin{aligned} \frac{\partial a'}{\partial \phi} &= \frac{\sin(2\phi)}{2} \begin{pmatrix} -\frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \\ \frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \end{pmatrix}, \\ \frac{\partial b'}{\partial \phi} &= \frac{\cos(2\phi)}{2} \begin{pmatrix} -\frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \\ \frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \end{pmatrix}, \\ \frac{\partial c'}{\partial \phi} &= -\frac{\sin(2\phi)}{2} \begin{pmatrix} -\frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \\ \frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \end{pmatrix}. \end{aligned} \quad (170)$$

So :

$$\frac{\partial A}{\partial \phi} = \frac{1}{2} \begin{pmatrix} -\frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \\ \frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \end{pmatrix} \begin{pmatrix} \sin(2\phi) & \cos(2\phi) \\ \cos(2\phi) & -\sin(2\phi) \end{pmatrix}. \quad (171)$$

Finally :

$$\frac{\partial \Phi}{\partial \phi} = -\frac{1}{2} \begin{pmatrix} -\frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \\ \frac{1}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \end{pmatrix} (\mathbf{x} - \mathbf{x}_c)^T \begin{pmatrix} \sin(2\phi) & \cos(2\phi) \\ \cos(2\phi) & -\sin(2\phi) \end{pmatrix} (\mathbf{x} - \mathbf{x}_c)g_{\sigma_x, \sigma_y, \phi}(\mathbf{x} - \mathbf{x}_c). \quad (172)$$

In the end we have :

$$\begin{aligned} \frac{\partial E_r}{\partial \mathbf{x}_c} &= -\int_{\Phi=0} A(\mathbf{x} - \mathbf{x}_c)D(\mathbf{x})d\mathbf{x}, \\ \frac{\partial E_r}{\partial \sigma_x} &= \frac{1}{8\sigma_x^3} \int_{\Phi=0} ((\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n}_\phi)^2 D(\mathbf{x})d\mathbf{x}, \\ \frac{\partial E_r}{\partial \sigma_y} &= \frac{1}{8\sigma_y^3} \int_{\Phi=0} ((\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{t}_\phi)^2 D(\mathbf{x})d\mathbf{x}, \\ \frac{\partial E_r}{\partial \phi} &= \frac{1}{8} \begin{pmatrix} -\frac{1}{\sigma_x^2} + \frac{1}{\sigma_y^2} \\ \frac{1}{\sigma_x^2} + \frac{1}{\sigma_y^2} \end{pmatrix} \int_{\Phi=0} (\mathbf{x} - \mathbf{x}_c)^T \begin{pmatrix} \sin(2\phi) & \cos(2\phi) \\ \cos(2\phi) & -\sin(2\phi) \end{pmatrix} (\mathbf{x} - \mathbf{x}_c)D(\mathbf{x})d\mathbf{x}. \end{aligned} \quad (173)$$

Considering the parametrization described in the previous section, we have :

$$\begin{aligned}
\frac{\partial E_r}{\partial \mathbf{x}_c} &= \int_0^{2\pi} [AN(\theta)] D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial \sigma_x} &= \frac{1}{8\sigma_x} \int_0^{2\pi} \cos^2(\theta) D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial \sigma_y} &= \frac{1}{8\sigma_y} \int_0^{2\pi} \sin^2(\theta) D(\mathbf{x}) \|T(\theta)\| d\theta, \\
\frac{\partial E_r}{\partial \phi} &= \frac{1}{8} \left(-\frac{1}{\sigma_x^2} + \frac{1}{\sigma_y^2} \right) \int_0^{2\pi} \left[N(\theta)^T \begin{pmatrix} \sin(2\phi) & \cos(2\phi) \\ \cos(2\phi) & -\sin(2\phi) \end{pmatrix} N(\theta) \right] D(\mathbf{x}) \|T(\theta)\| d\theta.
\end{aligned} \tag{174}$$

B.4 Regularization Energy E_s

The derivation for the pupil shape energy is straightforward :

$$\begin{aligned}
\frac{\partial E_s^p}{\partial \mathbf{x}_c} &= 0, \\
\frac{\partial E_s^p}{\partial a} &= (a - b), \\
\frac{\partial E_s^p}{\partial b} &= (b - a), \\
\frac{\partial E_s^p}{\partial \phi} &= 0.
\end{aligned} \tag{175}$$

For the iris shape energy, we take the following parametrization for the pupil focus points :

$$\begin{aligned}
\mathbf{x}f_p^{(1)} &= \mathbf{x}_c^p + \sqrt{(a^p)^2 - (b^p)^2} \vec{n}_\phi^p, \\
\mathbf{x}f_p^{(2)} &= \mathbf{x}_c^p - \sqrt{(a^p)^2 - (b^p)^2} \vec{n}_\phi^p.
\end{aligned} \tag{176}$$

Then the resulting parametrization for the iris focus points are :

$$\begin{aligned}
\mathbf{x}f_i^{(1)} &= \mathbf{x}_c^i + \delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \vec{n}_\phi^i, \\
\mathbf{x}f_i^{(2)} &= \mathbf{x}_c^i - \delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \vec{n}_\phi^i,
\end{aligned} \tag{177}$$

with :

$$\begin{aligned}
\delta_{pi} &= 1 && \text{if } \|\phi_p - \phi_i\| \leq \pi, \\
&= -1 && \text{otherwise.}
\end{aligned} \tag{178}$$

We have :

$$\begin{aligned}
\frac{\partial \mathbf{x}f_i^{(1)}}{\partial \mathbf{x}_c^i} &= \frac{\partial \mathbf{x}f_i^{(2)}}{\partial \mathbf{x}_c^i} = I_2, \\
\frac{\partial \mathbf{x}f_i^{(1)}}{\partial a} &= -\frac{\partial \mathbf{x}f_i^{(2)}}{\partial a} = \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \vec{n}_\phi^i, \\
\frac{\partial \mathbf{x}f_i^{(1)}}{\partial b} &= -\frac{\partial \mathbf{x}f_i^{(2)}}{\partial b} = -\delta_{pi} \frac{b}{\sqrt{(a^i)^2 - (b^i)^2}} \vec{n}_\phi^i, \\
\frac{\partial \mathbf{x}f_i^{(1)}}{\partial \phi_i} &= -\frac{\partial \mathbf{x}f_i^{(2)}}{\partial \phi_i} = -\delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \vec{t}_\phi^i.
\end{aligned} \tag{179}$$

For an arbitrary parameter α we have :

$$\frac{\partial E_s^i}{\partial \alpha} = (\mathbf{x}\mathbf{f}_i^{(1)} - \mathbf{x}\mathbf{f}_p^{(1)}) \frac{\partial \mathbf{x}\mathbf{f}_i^{(1)}}{\partial \alpha} + (\mathbf{x}\mathbf{f}_i^{(2)} - \mathbf{x}\mathbf{f}_p^{(2)}) \frac{\partial \mathbf{x}\mathbf{f}_i^{(2)}}{\partial \alpha}. \quad (180)$$

So :

$$\begin{aligned} \frac{\partial E_s^i}{\partial \mathbf{x}_c^i} &= (\mathbf{x}\mathbf{f}_i^{(1)} - \mathbf{x}\mathbf{f}_p^{(1)}) + (\mathbf{x}\mathbf{f}_i^{(2)} - \mathbf{x}\mathbf{f}_p^{(2)}), \\ &= 2(\mathbf{x}_c^i - \mathbf{x}_c^p), \end{aligned} \quad (181)$$

$$\begin{aligned} \frac{\partial E_s^i}{\partial a} &= \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \left[(\mathbf{x}\mathbf{f}_i^{(1)} - \mathbf{x}\mathbf{f}_p^{(1)}) - (\mathbf{x}\mathbf{f}_i^{(2)} - \mathbf{x}\mathbf{f}_p^{(2)}) \right] \cdot \vec{n}_\phi^i, \\ &= \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} n_\phi^p \cdot n_\phi^i \right], \\ &= \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \end{aligned} \quad (182)$$

$$\frac{\partial E_s^i}{\partial b} = -\delta_{pi} \frac{b}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \quad (183)$$

$$\begin{aligned} \frac{\partial E_s^i}{\partial \phi_i} &= \delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \left[(\mathbf{x}\mathbf{f}_i^{(1)} - \mathbf{x}\mathbf{f}_p^{(1)}) - (\mathbf{x}\mathbf{f}_i^{(2)} - \mathbf{x}\mathbf{f}_p^{(2)}) \right] \cdot \vec{t}_\phi^i, \\ &= 2\delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \sqrt{(a^p)^2 - (b^p)^2} \left[n_\phi^p \cdot t_\phi^i \right], \\ &= -2\delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \sqrt{(a^p)^2 - (b^p)^2} \sin(\phi_i - \phi_p). \end{aligned} \quad (184)$$

Summarizing the above results, we get :

$$\begin{aligned} \frac{\partial E_s^i}{\partial \mathbf{x}_c} &= 2(\mathbf{x}_c^i - \mathbf{x}_c^p), \\ \frac{\partial E_s^i}{\partial a} &= \delta_{pi} \frac{a}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \\ \frac{\partial E_s^i}{\partial b} &= -\delta_{pi} \frac{b}{\sqrt{(a^i)^2 - (b^i)^2}} \left[1 - \frac{\sqrt{(a^p)^2 - (b^p)^2}}{\sqrt{(a^i)^2 - (b^i)^2}} \cos(\phi_p - \phi_i) \right], \\ \frac{\partial E_s^i}{\partial \phi} &= -2\delta_{pi} \sqrt{(a^i)^2 - (b^i)^2} \sqrt{(a^p)^2 - (b^p)^2} \sin(\phi_i - \phi_p). \end{aligned} \quad (185)$$

B.5 Circular model

B.5.1 General elements

We consider the circles as a particular type of ellipses with :

$$\begin{aligned} \mathbf{x}_c^{circle} &= \mathbf{x}_c^{ellipse}, \\ r^{circle} &= \frac{1}{2}(a^{ellipse} + b^{ellipse}), \\ \phi^{circle} &= 0, \end{aligned} \quad (186)$$

therefore we have :

$$\begin{aligned} x(\theta) &= x_c + r \cos(\theta) \\ y(\theta) &= y_c + r \sin(\theta) \end{aligned} \quad (187)$$

with :

$$\|N(\theta)\| = \|T(\theta)\| = r, \quad (188)$$

and the unit tangential and normal vectors are :

$$\begin{aligned} \vec{t}_\theta &= \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix} \\ \vec{n}_\theta &= \begin{pmatrix} -\cos(\theta) \\ -\sin(\theta) \end{pmatrix}. \end{aligned} \quad (189)$$

B.5.2 Edge Term

We consider the energy of Equation (151) using the simplification presented above :

$$\begin{aligned} \frac{\partial E_n}{\partial \mathbf{x}_c} &= \int_0^{2\pi} [H(\mathbf{x}) \vec{n}_\theta] \text{dir}(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial a} &= \int_0^{2\pi} \left([H(\mathbf{x}) \begin{pmatrix} \cos(\theta) \\ 0 \end{pmatrix}] \cdot \vec{n}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\sin(\theta) \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \sin^2(\theta) \vec{n}_\theta \right] \right) \text{dir}(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial b} &= \int_0^{2\pi} \left([H(\mathbf{x}) \begin{pmatrix} 0 \\ \sin(\theta) \end{pmatrix}] \cdot \vec{n}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\cos(\theta) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \cos^2(\theta) \vec{n}_\theta \right] \right) \text{dir}(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial \phi} &= 0. \end{aligned} \quad (190)$$

Accordingly, we have :

$$\begin{aligned} \frac{\partial E_n}{\partial r} &= \frac{1}{2} \left(\frac{\partial E_n}{\partial a} + \frac{\partial E_n}{\partial b} \right) \\ &= \int_0^{2\pi} \left([H(\mathbf{x}) \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}] \cdot \vec{n}_\theta + \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} + (\sin^2(\theta) + \cos^2(\theta)) \vec{n}_\theta \right] \right) \text{dir}(\vec{n}_\theta) d\theta, \\ &= \int_0^{2\pi} \left([-H(\mathbf{x}) \vec{n}_\theta] \cdot \vec{n}_\theta + \frac{\nabla I(\mathbf{x})}{r} \cdot [-\vec{n}_\theta + \vec{n}_\theta] \right) \text{dir}(\vec{n}_\theta) d\theta, \\ &= \int_0^{2\pi} - [H(\mathbf{x}) \vec{n}_\theta] \cdot \vec{n}_\theta \text{dir}(\vec{n}_\theta) d\theta, \end{aligned} \quad (191)$$

So in the end we have :

$$\begin{aligned} \frac{\partial E_n}{\partial \mathbf{x}_c} &= \int_0^{2\pi} [H(\mathbf{x}) \vec{n}_\theta] \text{dir}(\vec{n}_\theta) d\theta, \\ \frac{\partial E_n}{\partial r} &= - \int_0^{2\pi} [H(\mathbf{x}) \vec{n}_\theta] \cdot \vec{n}_\theta \text{dir}(\vec{n}_\theta) d\theta, \end{aligned} \quad (192)$$

Similarly for the energy of Equation (152) :

$$\begin{aligned}
\frac{\partial E_t}{\partial \mathbf{x}_c} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{t}_\theta \right] \text{dir}(\vec{t}_\theta) d\theta , \\
\frac{\partial E_t}{\partial a} &= - \int_0^{2\pi} \left(\left[H(\mathbf{x}) \begin{pmatrix} \cos(\theta) \\ 0 \end{pmatrix} \right] \cdot \vec{t}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\sin(\theta) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \sin^2(\theta) \vec{t}_\theta \right] \right) \text{dir}(\vec{t}_\theta) d\theta , \\
\frac{\partial E_t}{\partial b} &= - \int_0^{2\pi} \left(\left[H(\mathbf{x}) \begin{pmatrix} 0 \\ \sin(\theta) \end{pmatrix} \right] \cdot \vec{t}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\cos(\theta) \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \cos^2(\theta) \vec{t}_\theta \right] \right) \text{dir}(\vec{t}_\theta) d\theta , \\
\frac{\partial E_t}{\partial \phi} &= 0.
\end{aligned} \tag{193}$$

Accordingly, we have :

$$\begin{aligned}
\frac{\partial E_t}{\partial r} &= \frac{1}{2} \left(\frac{\partial E_t}{\partial a} + \frac{\partial E_t}{\partial b} \right) \\
&= - \int_0^{2\pi} \left(\left[H(\mathbf{x}) \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \right] \cdot \vec{t}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[\begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix} + (\sin^2(\theta) + \cos^2(\theta)) \vec{t}_\theta \right] \right) \text{dir}(\vec{t}_\theta) d\theta , \\
&= - \int_0^{2\pi} \left(\left[-H(\mathbf{x}) \vec{n}_\theta \right] \cdot \vec{t}_\theta - \frac{\nabla I(\mathbf{x})}{r} \cdot \left[-\vec{t}_\theta + \vec{t}_\theta \right] \right) \text{dir}(\vec{t}_\theta) d\theta , \\
&= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] \cdot \vec{t}_\theta \text{dir}(\vec{t}_\theta) d\theta ,
\end{aligned} \tag{194}$$

So in the end we have :

$$\begin{aligned}
\frac{\partial E_t}{\partial \mathbf{x}_c} &= - \int_0^{2\pi} \left[H(\mathbf{x}) \vec{t}_\theta \right] \text{dir}(\vec{t}_\theta) d\theta , \\
\frac{\partial E_t}{\partial r} &= \int_0^{2\pi} \left[H(\mathbf{x}) \vec{n}_\theta \right] \cdot \vec{t}_\theta \text{dir}(\vec{t}_\theta) d\theta ,
\end{aligned} \tag{195}$$

B.5.3 Region Term

For the region term we have

$$\begin{aligned}
\frac{\partial E_r}{\partial \mathbf{x}_c} &= \int_0^{2\pi} [AN(\theta)] D(\mathbf{x}) \|T(\theta)\| d\theta , \\
&= \int_0^{2\pi} \left[\frac{1}{2r^2} r \vec{n}_\theta \right] D(\mathbf{x}) r d\theta , \\
&= \frac{1}{2} \int_0^{2\pi} D(\mathbf{x}) \vec{n}_\theta d\theta , \\
\frac{\partial E_r}{\partial r} &= \frac{1}{2} \frac{1}{8r} \int_0^{2\pi} (\cos^2(\theta) + \sin^2(\theta)) D(\mathbf{x}) r d\theta , \\
&= \frac{1}{16} \int_0^{2\pi} D(\mathbf{x}) d\theta ,
\end{aligned} \tag{196}$$