A biometric reference system for iris

OSIRIS version 2.01

howto last updated - 27/10/2009

Foreword. *The reference system was developed by Telecom & Management Sud Paris (previously GET-INT) (Emine Krichen, Anouar Mellakh, Phan Viet Anh, Sonia Salicetti and Bernadette Dorizzi). The following 'how to' was written Phan Viet Anh (Telecom & Management Sud Paris).*

Abstract. *The OSIRIS (Open Source for IRIS) reference system is an open source iris recognition system developed in the framework of the BioSecure project. The system is inspired by Daugman works [1]. It is composed of segmentation and classification steps. The segmentation part uses the cicular Hough transform and an active contour approach tp detect the contours of iris and pupil. The classification part is based on Gabor phase demodulation and Hamming distance classification.*

Introduction

This document describes the version 2 of the OSIRIS reference system and explains how to use it in practice to make an automatic verification experiment based on iris modality. This document is divided into four parts. In the first part, we present how to install the system. Then, in the second part, the database used and the protocol which have been defined are described. In the third part, we move on to the explanation of each module and finally, the experimental results obtained in authentication are shown in the last part.
This document refers to version 2 of the software in which the segmentation part has been greatly improved over the previous one in version 1. A simple Hough transform for fitting the circles arround the iris and pupil was implemented in version 1.

I.Installation

The full system has been tested under Linux (Fedora 8.0). For the compilation step, we have used g++ 4.1.2.

1.Downloading

The OSIRIS reference system is currently available on the Telecom & Management Sud Paris subversion server at the following address :

http://share.int-evry.fr/svnview-eph/

The source files are available in the part *ref_syst/Iris_Osiris/download/*. Download the archive *Iris_Osiris.tar.gz* in your home directory */home/Iris_Osiris/*, and then decompress it as follows :

```
>tar zxvf Iris_Osiris.tar.gz
```

The new segmentation module uses some function of the OpenCV library. So, you need to install OpenCV before compiling OSIRIS source code. You can download the OpenCV library at the following address :

http://www.sourceforge.net/projects/opencvlibrary

2. Compilation

To compile the source code, you must perform the following steps :

- enter in the *src* directory :
  ```
  >cd Iris_Osiris/src
  ```
- open the *makefile* and change the value of *OpenCvDir* to the directory where the OpenCV library is installed. For example :
  ```
  OpenCvDir = /home/opencv
  ```
- compile the source code :
  ```
  >make all
  ```

This compilation step creates the executable `osiris` in your directory */home/Iris_Osiris/src*.

II. Reference database

The National Institute of Standards and Technology (NIST) has provided researchers the database ICE_2005 [5]. This database consists of 2953 grayscale eye images of 132 people. They are acquired with a dedicated LG2200 camera. Each image captures one eye and has a size of 640x480 pixels. This database has been divided into two sub-databases : one for images of the right iris (1425 iris images from 124 persons) and another one for images of the left iris (1528 iris image of 120 persons). In most cases, images of the right and left irises are acquired at the same time. The images in this database contain reflections, blur and interlacing distorsion. In addition, the iris in the database can be covered by eyelids, eyelashes.

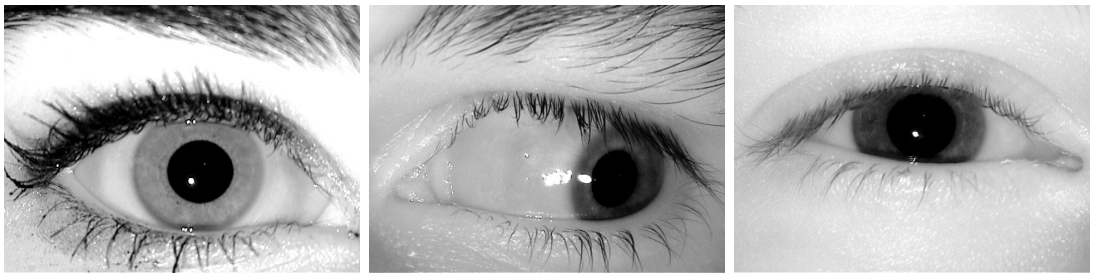
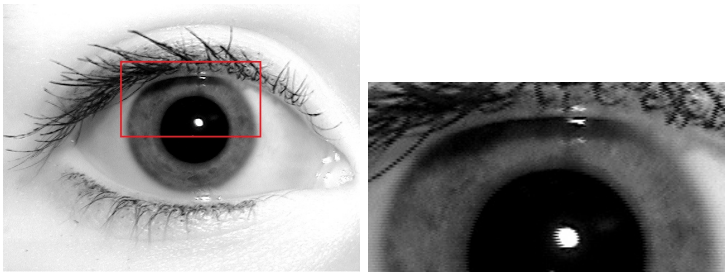


FIGURE 1 – Examples of iris images in the ICE_2005 database



[a] [b]

FIGURE 2 – Examples of original iris image presenting interlacing distorions (a) and zoom of the red region of this image (b)

III.Presentation of the software

The reference system has several sub-modules, which are described in this part. The execution of each sub-module is based on the use of an executable (`osiris`) and/or a script.
We use the script with the following format for the segmentation :

1. `Options` : options required by the executable `osiris`.
2. `input_Dir` : The path of the directory where the database is stored.
3. `fileList` : The file that contains the list of images to be processed. The number of images in the list is quoted in the first line of this file. The following rows contain the name of the image (one image per line). See example in figure 5.

```
5
239219.bmp
239224.bmp
239226.bmp
239229.bmp
239230.bmp
```

FIGURE 3 – Example of a file that contains a list of images to be processed

4. `mask_Dir` : The directory where the mask will be saved.
5. `segment_Hough_Dir` : This directory contains the images with the initial contours (we need this parameter if option `-i` has been used).
6. `snake_Dir` : This directory contains the images with the final contours (we need this parameter if option `-i` has been used).

For the *Normalisation Module* and the *Featrure Extraction and Matching Module*, we use the script which is made of 7 fields. If one wants to use the script, the 7 fields must be filled with the following pieces of information which are the same for each sub-module (the *Normalisation Module* and the *Featrure Extraction and Matching Module*) :

1. `Options` : options required by the executable `osiris`.

2. `Filter and Points` : path to the text format filter file and the text format points file. These two files are needed for the features extraction (see part III.3) and are respectively located at */home/Iris_Osiris/OsirisParam/GaborFilterBank.txt* and */home/Iris_Osiris/OsirisParam/Point.txt*.
3. `Path_source` : path to the directory where the input files are located.
4. `Path_results` : path to the directory where the output file will be stored.
5. `Result_file` : name of the text results file. This file is only produced at the matching step (see part III.3).
6. `Match` : field containing the names of the input files which are processed by the sub-module.

N.B :
- Even if all these pieces of information are not needed by the sub-module, all fields have to be filled in order to assure a good functioning of the script.
- This system work only with the BMP (8 bit/pixel or 24 bit/pixel) image. If your image database is not in this format, you must convert it into BMP format before using this system.

1.Segmentation

a.Description

The *segmentation* module is used for localizing the iris.
First, this module roughly search the pupil region in the image using a binarization and the fact that the pupil is circular.
Second, this module looks for the contour between the pupil and the iris and the iris and the eye white part.This is done in two steps :
- a rough localization using Hough transform [2],[3].
- an active contour approach [4] initialized on the circle founded in the previous step using Hough transform is done to refine the contours (See figure 3 (a) and (b)).

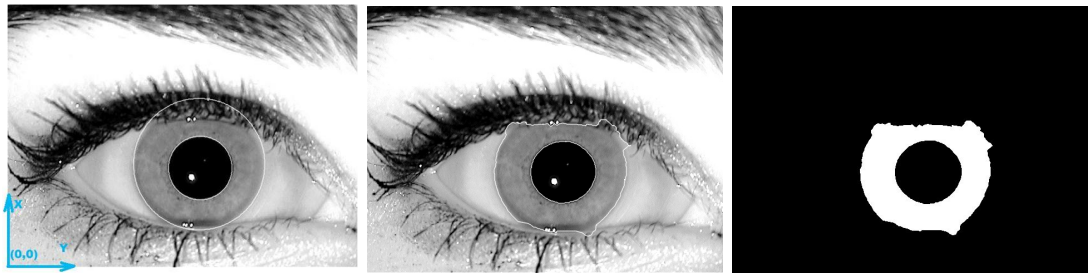
The module also generates a mask that shows the region containing the texture of the iris and a file that contains the coordinates of the two circles closest to the final contour. These circles are used for normalizing the iris region and the mask ; this is done in the normalization module.

b.Software presentation

Use : `./osiris -s script`

Input : The file that contains options and settings
Output :

- Image that contains the initial contours (figure 3a)
- Image with the final contours (figure 3b)
- Image of the mask. The region containing the texture of iris is the white region in this image (figure 3c)
- File containing the parameters needed for the normalization (figure 4)



[a] [b] [c]

FIGURE 4 – Initial contours (a), final contours (b) and the associated mask (c)

```
XPUP : 316
YPUP : 255
RPUP : 49

XIRIS : 321
YIRIS : 260
RIRIS : 120
```

FIGURE 5 – An example of a parameters file produced by the *segmentation* module

c.Script file

The script file contains the options and settings that are needed for running this module.

Options :

1. `-d` : only segmentation is performed on the original eye image.
2. `-i` : Save an image with the initial contours in a file called `segment_Hough_Dir/image_init.jpg` and another one with the final contours in a file called `snake_Dir/image_snake.jpg`.
3. `-l` : the parameters of the segmentation are saved in a file called `mask_Dir/image.Param`.

Here is an example of a script file :

```
Options : -d -l -i
input_Dir = /home/IRIS_OSIRIS/image/
fileList = /home/IRIS_OSIRIS/param/ImageList.txt
mask_Dir = /home/IRIS_OSIRIS/image/
snake_Dir = /home/IRIS_OSIRIS/image/
segment_Hough_Dir = /home/IRIS_OSIRIS/image/
```

FIGURE 6 – An example of script for the *segmentation* module

2.Normalisation

a.Description

The sub-module *normalisation* aims at normalizing the iris area in terms of size.

b.Software presentation

Use : `./osiris -s script`

The execution of the sub-module is based on the use of a script (see the introduction of part III for a global description of such a script) whose path (`script`) is specified with the standard option `-s`. The fields of this script used by the sub-module are described below :

Path_source : path to the directory which contains the input files, i.e. the original eye images (*.bmp*) and their parameters files (these latest files are produced by the previous sub-module).

Match : on each line of this field, there are two image filenames followed by their corresponding parameters filenames (each filename is separated by the tabulation character). An illustration of this field is shown in figure 7.

Match :239219.bmp	239219_mask.bmp	239219.Param	239219.Param
Match :239224.bmp	239224_mask.bmp	239224.Param	239224.Param
Match :239226.bmp	239226_mask.bmp	239226.Param	239226.Param
Match :239229.bmp	239229_mask.bmp	239229.Param	239229.Param
Match :239230.bmp	239230_mask.bmp	239230.Param	239230.Param
Match :239234.bmp	239234_mask.bmp	239234.Param	239234.Param
...			

FIGURE 7 – Illustration of the field `Match` required by the *normalisation* module

Options :

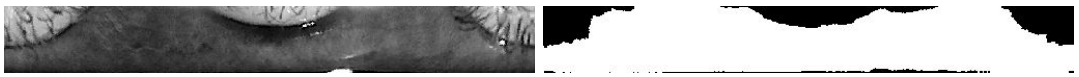
1. `-k` : only normalisation is performed.
2. `-p` : the parameters files are taken into account for the normalisation step.
3. `-w` : from an input image called `image.bmp`, the normalized image will be saved as `image_Norm.bmp`.

Path_results : path to the directory which contains the output files, i.e. the normalized images of iris (*.bmp*). An example of a such image is given in figure 8.

3.Feature extraction and Matching

a.Description

The sub-module *feature extraction* performs a convolution operation between the normalized iris image and a set of Gabor filters at pre-fixed points. At the end of this



[a] [b]

FIGURE 8 – An example of a normalized image [a] and its mask [b] (512 × 64 pixels) produced by the *normalisation* module

operation, we have at disposal a set of coefficients representing the iris. Each coefficient is then coded depending on its sign, resulting in a binary code of fixed length (denoted as iris code).
The sub-module *matching* provides a score. A comparison between two iris codes using the Hamming distance is performed. The score is a number between 0 and 1, and it equals 0 when the two iris codes are strictly identical.

b.Software presentation

Use : `./osiris -s script`

The execution of the two sub-modules is based on the use of a single script (see the introduction of part III to have a global description of a such script) whose path (`script`) is specified with the standard option `-s`. The fields of this script used by the sub-modules are described below :

Path_source : path to the directory which contains the input files, i.e. the normalized images of iris (*.bmp*) which are provided by the previous sub-module (*normalisation*).

Filter : path to the filter file (*.txt*) which contains a set of Gabor filters.

Points : path to the points file (*.txt*) containing the coordinates of the points at which the convolution between the normalized image and the set of filters is performed.

Match : on each line of this field, are the filenames of the two normalized images which have to be compared (the two filenames are separated by the tabulation character). An illustration of this field is shown in figure 9.

Match :242862_Norm.bmp	246727_Norm.bmp	242862_mask_Norm.bmp	246727_mask_Norm.bmp
Match :242862_Norm.bmp	240963_Norm.bmp	242862_mask_Norm.bmp	240963_mask_Norm.bmp
Match :242862_Norm.bmp	240568_Norm.bmp	242862_mask_Norm.bmp	240568_mask_Norm.bmp
Match :242862_Norm.bmp	240010_Norm.bmp	242862_mask_Norm.bmp	240010_mask_Norm.bmp
Match :242862_Norm.bmp	239485_Norm.bmp	242862_mask_Norm.bmp	239485_mask_Norm.bmp
Match :242862_Norm.bmp	241382_Norm.bmp	242862_mask_Norm.bmp	241382_mask_Norm.bmp
...			

FIGURE 9 – Illustration of the field `Match` required by the modules *features extraction* and *matching*

Options :

1. `-n` : segmentation is not performed. The input images have already been normalized.
2. `-e` : segmentation is not performed. The input images have already been normalized. Use the mask for calculating the score.
3. `-c` : facultative option used to save the binary iris code files. From an input normalized image called `image_Norm.bmp`, the binary code file will be saved as `image_Norm.cod`.

Path_results : path to the directory which contains the output files, i.e. the text results file (called `Result_file`) and the binary code files (if you have mentioned the standard option `-c`).

Result_file : this file (which is located at `Path_results`) contains on each line the score (Hamming distance) of a comparison between two iris codes. A sample of a such file is given in figure 10.

242862_Norm	246727_Norm	0.191529
242862_Norm	240963_Norm	0.19798
242862_Norm	240568_Norm	0.287356
242862_Norm	240010_Norm	0.207804
242862_Norm	239485_Norm	0.209034
...		

FIGURE 10 – A sample of `Result_file` produced by the *matching* module (name of the target iris on the first column, name of the test iris on the second and score on the third)

IV. Experiments

To test the performance of the new version of the OSIRIS reference system, we used the rights iris in the database ICE_2005. There were totally 7704 intra-class comparisons. For the inter-class, we selected randomly 219231 comparisons. These scripts files can be found in the directory `IRIS_OSIRIS/script/`.
We get an EER equal to 5.14% and the FRR at FAR of 0.1% is equal to 16.03%. We also ploted the DET curve in figure 11.
This new version allows a big amelioration compared to version 1 which gave an EER of 26.86% and the FRR at FAR of 0.1 % of 90.37% on the same database with the same protocol.

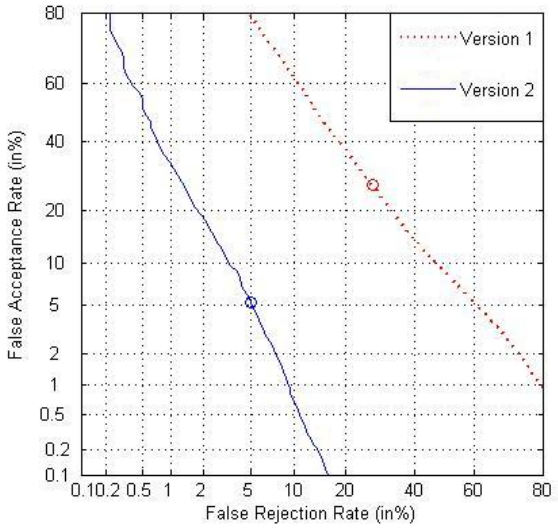


FIGURE 11 – DET curve of reference system

In the second test, we compare the perfomance of the system between 3 types of the segmentations :

- The segmentation manual : with this segmentation, we have only 2 circles whos represent the contour of the pupil and the contour of iris.
- The segmentation manual + active contour : use the segmentation manual as the initial contours for the active contour approach.
- Hough transform + active contour : use the Hough transform to find the initial contour and then use the active contour to refine it. This is done by using the segmentation module of OSIRIS version 2.0.

The Figure 12 shown us the performance of the system corresponding 3 types of segmentations. When we use the segmentation manual, we get an EER equal to 1.73% (green curved). But with the combination of the segmentation manual and the active contour (blue curved), we have improved the result. In this case, the EER equal to 1.17%. The segmentation module of OSIRIS version 2 gave an EER equal to 5.14% (red curved).

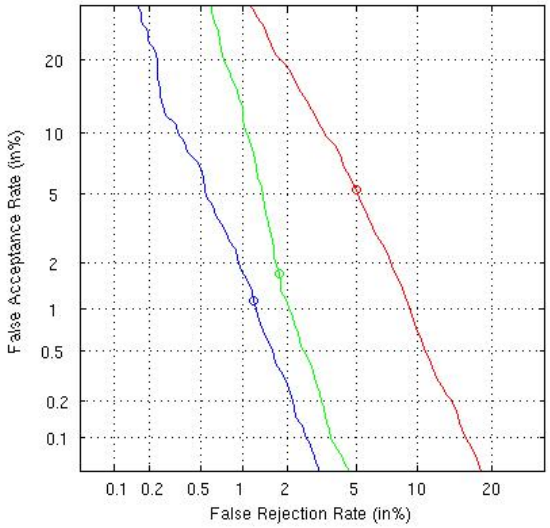


FIGURE 12 – The segmentation manual (green curved), The segmentation manual + active contour (blue curved), Hough transform + active contour (red curved)

Conclusion

The OSIRIS reference system has been described and tested on the ICE_2005 database. This reference system is made of several modules but it is not completely modular. Indeed, the two modules *feature extraction* and *matching* can currently not be dissociated. The segmentation module has been modified and improved in order to better adapt to degraded data such as the one present in ICE_2005.

Acknowledgments

The author express their sincere gratitude to Anouar Mellakh and Emine Krichen for their availability, explanations and kindness.

Reference

[1] J.G. Daugman, High confidence visual recognition of persons by a test of statistical independence, *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(11) :1148–1161, 1993.
[2] Kyewook Lee, Application of Hough Transform, *University of Massachusetts*, 2006.
[3] Simon Just Kjeldgaard Pedersen, Circular Hough Transform, *Aalborg University* , Novembre 2007.
[4] Chenyan Xu, Jerry L. Prince, Snakes, Shapes, and Gradient Vector Flow, *IEEE Transactions on image processing*, vol. 7, No. 3 March 1998.
[5] http ://iris.nist.gov/ice/