République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Benyoucef BENKHEDDA-Alger1

****

Faculté des Sciences

Département des Mathématiques et Informatique

Mini Projet du TP Traitement d’Images Numériques

1ere Année Master ISII

Sujet

|  |
| --- |
| **Réalisation d’un système biométrique d’identification d’individus par l’iris** |

**Professeur : Réalisé par :**

- Mme  BELATTAR Khadidja - Bouzid Moussaab

- El-moubarek fayçal

2020/2021

# Table des matières

[Table des matières 2](#_Toc68215768)

[Introuction 4](#_Toc68215769)

[Plan de travaille 4](#_Toc68215770)

[Praitretement 4](#_Toc68215771)

[Egalisation d’histograme (utilisé) 5](#_Toc68215772)

[Ajustement du conrast (non utilisé) 6](#_Toc68215773)

[Filtre median pondéré (utilisé) vs Filtre median (non utilisé) 6](#_Toc68215774)

[Lissage moyen et moyen pondéré (non utilisé) 7](#_Toc68215775)

[Segmentation clustering K-means et operation morphologiques 8](#_Toc68215776)

[(Non utilisé) 8](#_Toc68215777)

[Filtre de difference (utilisé) 8](#_Toc68215778)

[Extraction des caréctistiques 9](#_Toc68215779)

[Extraction de l’iris 9](#_Toc68215780)

[Detecteur et descripteur sift 10](#_Toc68215781)

[Mise en corresependence 10](#_Toc68215782)

[Distence eucledienne 10](#_Toc68215783)

[Affichage de l’interface 10](#_Toc68215784)

[Echec d’Identification 11](#_Toc68215785)

[Identification reussit 11](#_Toc68215786)

[Tableau du code 12](#_Toc68215787)

## Introuction

La realisation d’un système d’identification biométrique est basé sur la qualité de l’image contenant les informations necéssaires pour l’empreint de l’iris. Et afin d’assurer les bonnes informations, un ensemble de traitements est obligatoir pour sortir avec des images de meilleure qualité.

## Plan de travaille

Prétraitements utilisés

* Egalisation d’histograme
* Lissage (filtre mediane pondéré)
* Extraction de l’iris
* Filtre contour

Extraction des carecteristiques

* + - Deteteur Sift et descripteur sift

Mise en correspedence

* Calcule de distences eucledienne
* Affichage du resultat

## Praitretement

### Egalisation d’histograme (utilisé)

L’objectif de l’histograme est d’afficher le redondonces des intecités des pixels dans l’image, et afin

D’amelioere la qualité de l’image on procede a l’egalisation d’histogramme.

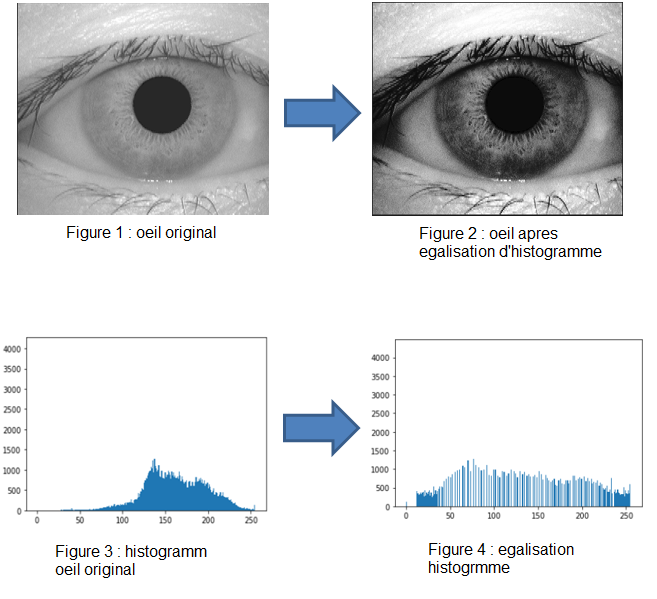
Comme on le voit dans la figure 3, l’histogramme de l’image originale est sous exposée

Ce qui montre que l’image est de mauvaise qualité donc on passe a l’egalisation :

1/ calcule la probabilité de chaque pixel avec la formule

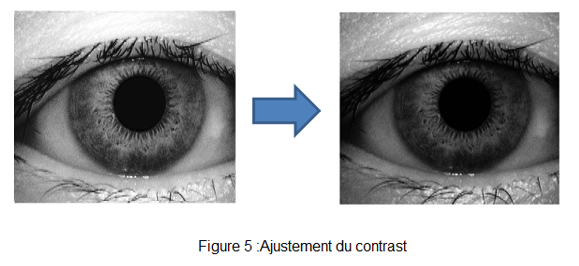
2/ calculer le cumuler des probabilités

3/ a la fin on multiplie fois le nombre d’intencité ‘255’



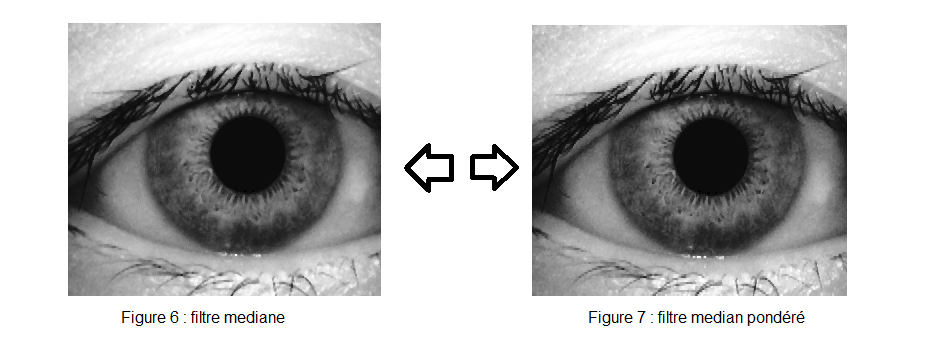
### Ajustement du conrast (non utilisé)

On appliquant l’ajustement du contrast sur le resultat de l’histogramme on obtient une photo aillant des intencité plus sombre et moins lumineuse

Pour notre cas l’ajustement du contraste n’était pas tres benfique car apres l’avoir appliqué, l’image est devenu plus sombre ce qui a affecté nos resultats du filtre de difference qu’on va voir apres, et en raison de la bonne repartition de nos instences dans l’etape de l’egalisation de l’histogramme l’image était preparé a passer au filtre mediane pondéré a fin de reduire le bruit et la rendre plus nette et d'affiner et de rehausser les contours

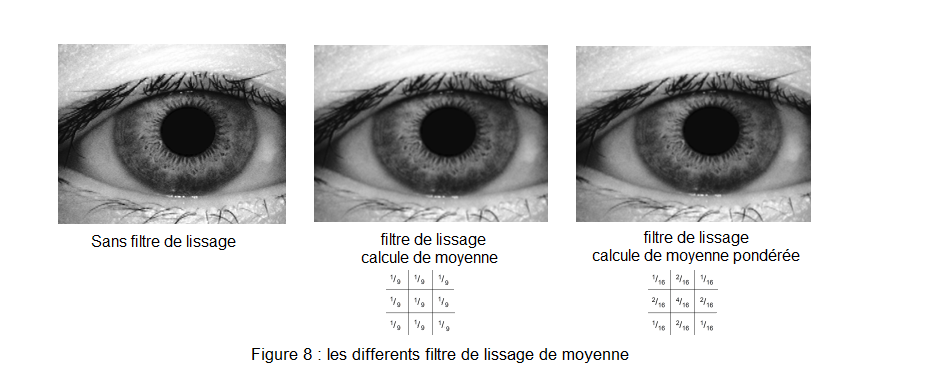
### Filtre median pondéré (utilisé) vs Filtre median (non utilisé)

Afin de trouver le meilleur filtre à utiliser on a comparé les deux filtres median et median pondéré pour sortir avec le meilleur filtre qui nous donnera plus de detaills dans l’etape de l’application du filtre de difference



On remarquant les deux photos on voit vaguement que les detaills de l’iris sans moins claire dans le filtre median « figure 6 »  contraitement au filtre median pondéré « figure 7 » qui est entrain de reduire le bruit de l’image sans avoir flouter les informations de l’iris, et le filte de difference qu’on va appliquer apres sera capable de nous donner en details les informations de l’iris avec le filtre mediane pondéré contrairement au filtre median

### Lissage moyen et moyen pondéré (non utilisé)

En appliquant les filtre de lissage sur le resultat de l’egalisation de l’histogramme on a vu que l’image est devenu plus flouté, dans le cadre de notre projet on a besoins que les details de l’iris soients plus claire pour la bonne detection, et on comparant avec le filtre median pondéré on voit bien que ce dernier est un meilleur outil que les lissages de calcule de moyenne

### Segmentation clustering K-means et operation morphologiques

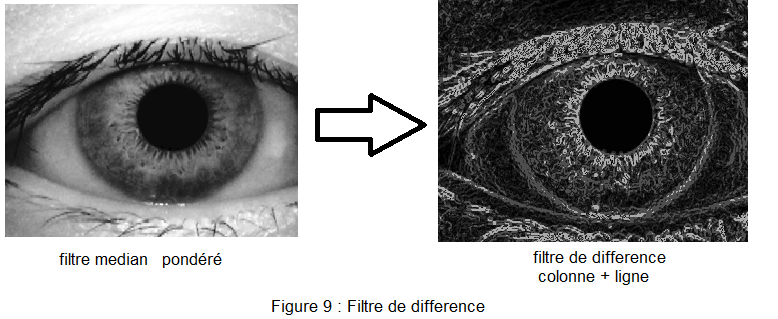
### (Non utilisé)

L’utilité de la segmentation K-means est de regrouper les points plus proches par couleur ou par intensité, dans notre cas on l’a appliqué par intensité sachant qu’on a des images en mode gris, et en appliquant le K-means, les segmentations de nos points n’étaient pas tres benefiques pour la phase de la mise en correspendence avec le SIFT. Le K-means à repositionner nos points clé de l’iris ce qui a rendu l’identification par le SIFT loins du scors du matching souhaité. La decision du non utilisation du K-means n’était pas prise dans la phase du pré-traitement, Meme avec de nombreuses tentatives apres les operation morphologiques (erosion et dilatation) appliquées sur les images, les resultats sur le matching du sift n’étaient pas correcte.

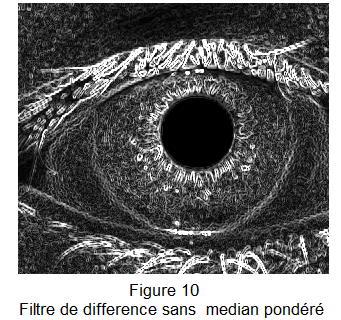
Ce qui nous a menés a appliqué le filtre de difference

### Filtre de difference (utilisé)

Le filtre de difference ou bien l’affichage des contours par colonne ou bien par ligne est une phase tres importante durant notre approche, et pour éclaircir et afficher le maximum de details pour les informations de l'iris on a prit judicieux d’appliquer le filtre par ligne et aussi par colonne



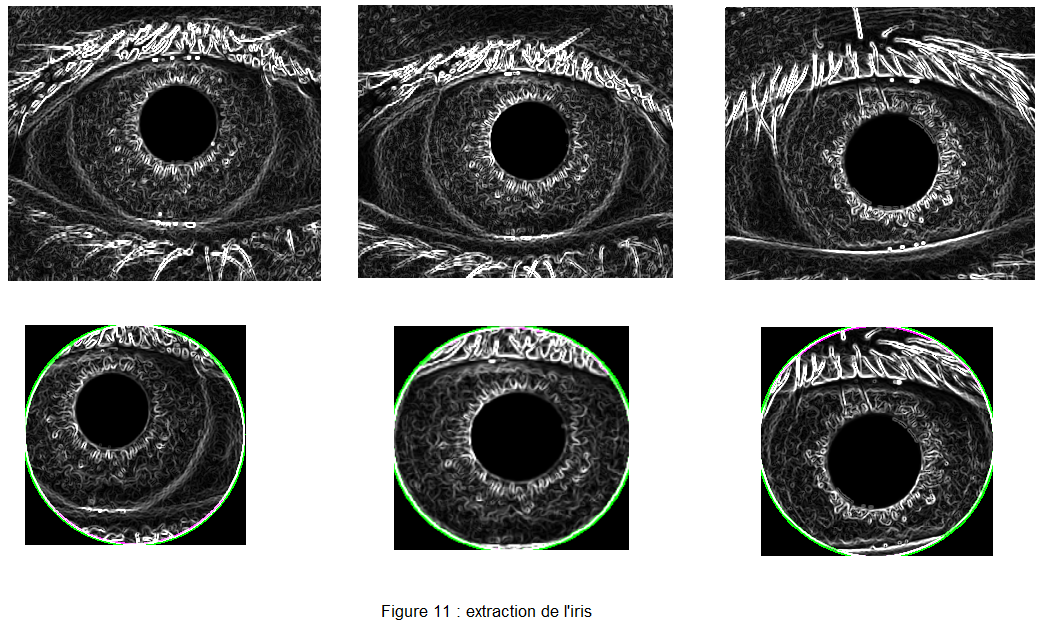
Si on avait utilisé le filtre de difference sans le filtre de median pondéré on aura beaucoup de bruit qui seront affiché (la figure 10)



L’utilité du filtre median pondéré en reduissant le bruit, est de permétre le filtre de difference d’extraire seulement les informations non bruitées

## Extraction des caréctistiques

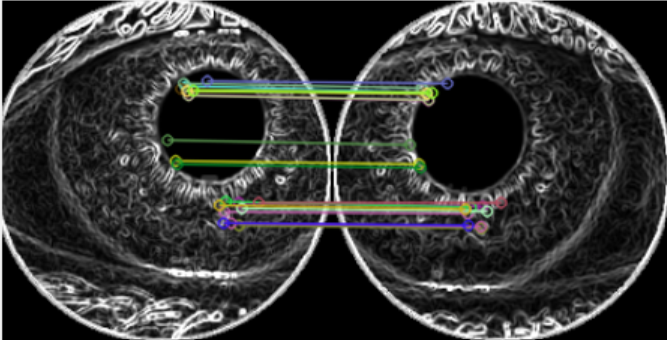
### Extraction de l’iris

Apres avoir terminé la phase du prétraitement, on passe a l’extraction de l’iris, car meme apres avoir ameliorer les images, on doit tout de meme laisser seulement la partie essentiel pour la detection des zones clé de l’iris et enlever informations non necessaire pour l’identification comme les cils

De cette forme, nos données seront stoqué dans la BDD et en demarant le processus de l’identification l’interface prendra en chagre la photo de la requété pour lui passer les memes traitements puis la compare avec l’ensemble des images de la BDD

### Detecteur et descripteur sift

A travers l’interface on ajoute une image afin de lancer le processus de l’identification. L’image va passer par un pré-traitement puis dans la phase du sift on va comparer ses carecteristique avec les differentes images de la BDD en utilisant la distence eucledienne



## Mise en corresependence

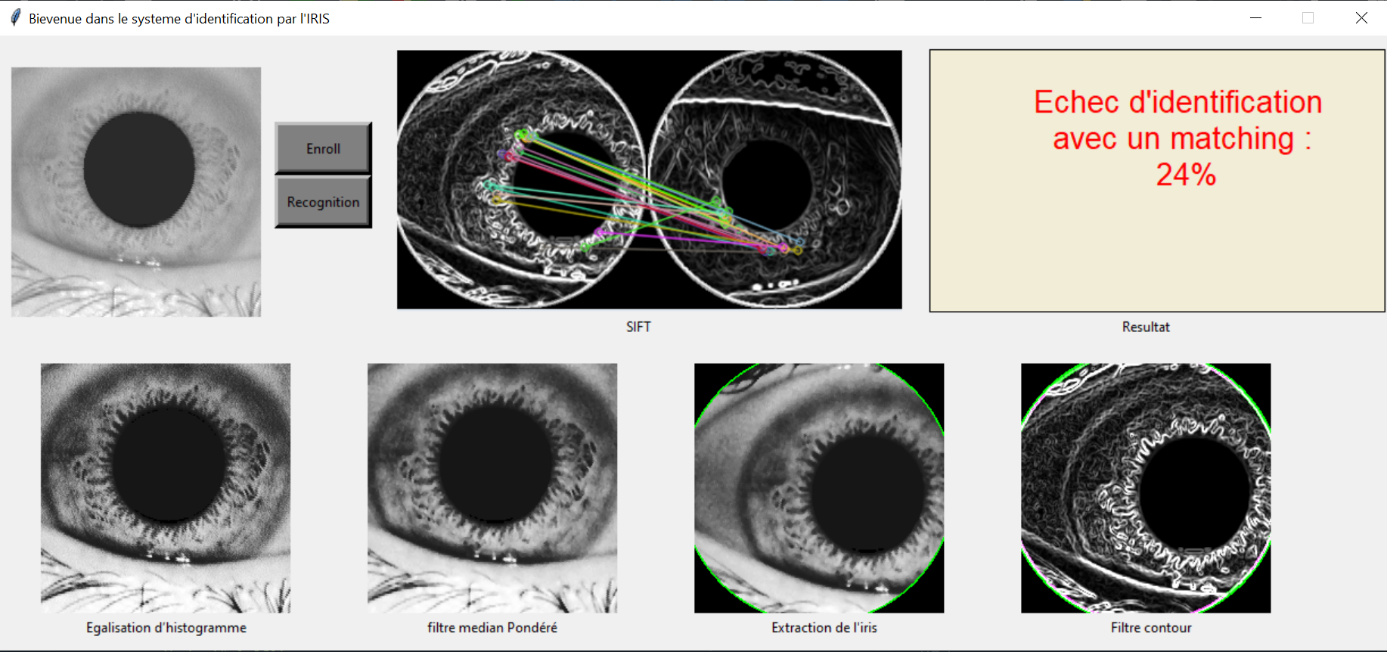
### Distence eucledienne

Dans cette phase on va comparer la distences entres les point du matching realisé dans le SIFT entre une image test et image BDD (sachant que l’image test sera comparé par toutes les images de la BDD), la selection de l’image compatible avec l’image test sera choisi en fonction du nombre de ressemblence du point par rapport a leur nombre de similarité de distence, et l’image de la BDD aillant un maximum de score de similarité sera la plus compatible a l’image test .

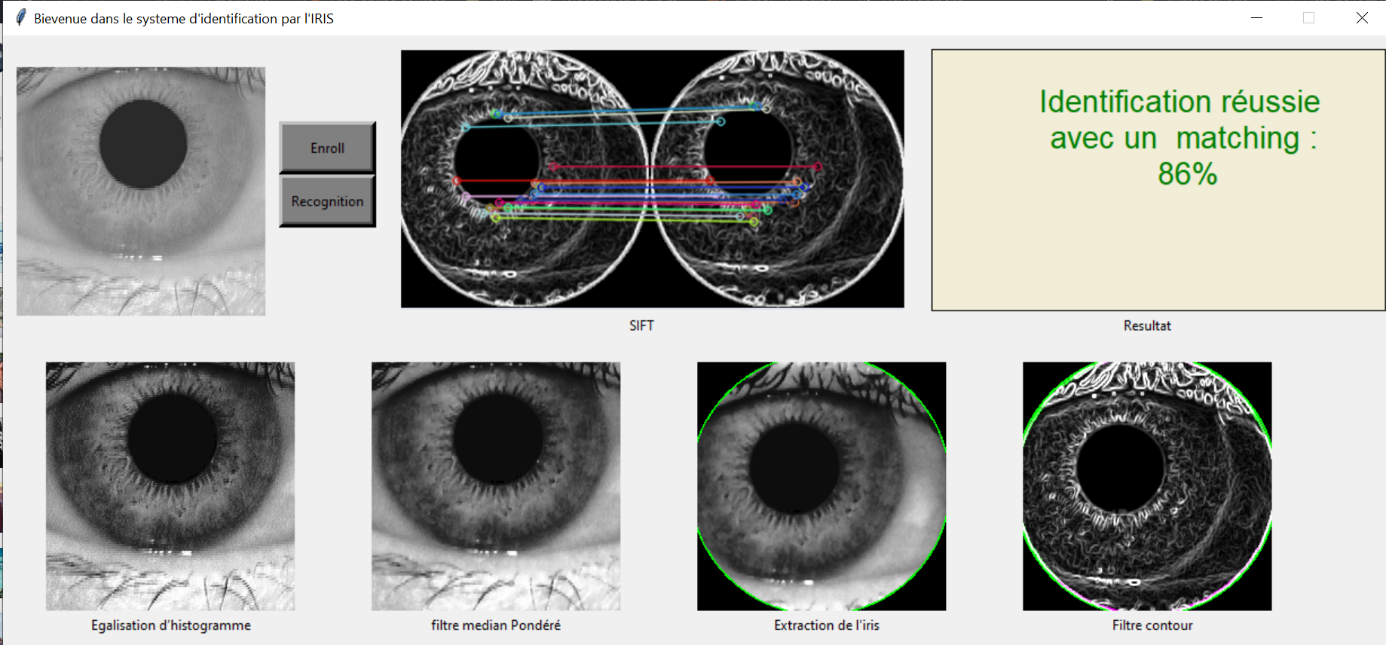
Se score du matching sera a la fin evalué en le comparant a un seuil de similarité bien definis

## Affichage de l’interface

### Echec d’Identification



### Identification reussit



## Tableau du code

Afin de vous facilité la tache pour parcourir les differents code du programme nous vous avons preparés la liste des fichiers code python et le traitement appliqué dans les fichiers.

Pour lancer le programme il suffit d’executer le fichier interface.py

|  |  |
| --- | --- |
| Traitement appliqué | Nom du fichier code python |
| Egalisation d’histogramme | histogram.py |
| Ajustement du contrast | Ajustement\_Du\_contrast.py |
| Lissage calcule moyenne | Lissage\_moyenne.py |
| Lissage calcule moyenne pondéré | Lissage\_moyenneP.py |
| Filtre median | Filtre\_Median.py |
| Filtre median pondéré | Filtre\_medianP.py |
| Filtre de difference (contours) | Contour2.py |
| Segmentation K-means | building\_k\_mean.py + building\_k\_mean\_test.py |
| Operation morphologique | Operation\_morphologique.py |
| Extraction de l’iris (haugh) | Haugh2.py |
| Matching du SIFT + distence eucledienne | SIFT.py |
| L’interface | Interface.py |