Rapport TER

Reconnaissance d'empreintes digitales sans contact

Josua Philippot - Félix Yriarte
Master 1 IMAGINA
FAJ-BYP

Encadré par : Pauline Puteaux - Iuliia Tkachenko

Janvier 2021 - Juin 2021

Nous tenons à remercier nos encadrantes Pauline Puteaux et Iuliia Tkachenko pour leur sollicitude et conseils avisés.

Table des matières

Résumé

Les systèmes biométriques offrent une couche de sécurisation supplémentaire face à la recrudescence des attaques cybercriminelles. Il est cependant d'autant plus critique de vérifier la non-accessibilité aux informations biométriques personnelles présentes dans de tels systèmes : celles-ci ne changent généralement pas au cours d'une vie. C'est pourquoi nous nous intéressons ici à la mise en place d'un système de reconnaissance d'empreintes digitales, tout en portant une attention particulière au chiffrement de ces données personnelles.
Pour ce faire, nous avons mis en place et comparé différentes méthodes de récupération de points caractéristiques. Nous avons alors légèrement modifié ces méthodes, dans l'optique de garantir le bon fonctionnement d'une *Fuzzy Vault*, qui offre une sécurisation supplémentaire de ces données sensibles, tout en permettant un accès aux données chiffrées par notre système biométrique quand bien même de légères différences apparaîtraient lors de la requête. Nous avons obtenu les meilleurs résultats en utilisant des *Crossing Numbers* sur des images pré-traitées, en ne prenant en compte que les points d'intérêt les plus proches du centroïde de masse. Nous n'avons finalement pas pu implémenter de *Fuzzy Vault* par manque de temps, mais pensons avoir réfléchi à tous les aspects nécessaires à son bon fonctionnement.

1 Introduction

Le sujet de notre TER porte sur le traitement d'empreintes digitales. Nous avons choisi ce sujet parce que les concepts de chiffrement d'information, et de biométrie en général, nous intéressent fortement. Ces thèmes s'ancrent dans l'utilisation contemporaine des appareils multimédia et smartphones. En effet, de nos jours le nombre de téléphones portables proposant une authentification biométrique, que ce soit de la reconnaissance faciale ou bien des empreintes digitales (qui est la plus répandue), est de plus en plus grand. Comprendre la manière dont fonctionnent des outils que nous utilisons tous les jours, afin de se faire une idée plus claire du monde qui nous entoure nous plaît fortement.

2 Contextualisation

2.1 Analyse d'empreinte : historique

On retrouve des traces d'identification d'empreintes primitives dans la préhistoire, dans *la civilisation babylonienne*, où elles servaient de signature avec empreintes sur les poteries, ou encore dans *l'antiquité chinoise*, avec des techniques qui se sont affinées avec le temps.
À partir du 19e siècle, on retrouve un usage qui s'intensifie, notamment *aux Indes* pour s'assurer qu'une personne ne touche une pension de l'armée qu'une seule fois, mais c'est au 20e siècle que l'identification d'individu par empreintes digitales est démocratisée dans la résolution des affaires criminelles.

Les empreintes digitales sont un moyen sûr d'identifier des individus : Elles sont uniques à chaque personne et ne changent que très peu au cours d'une vie.

2.2 Les systèmes biométriques

Les systèmes biométriques, dans le cadre de l'informatique, sont des systèmes de reconnaissance et d'authentification des êtres vivants à partir de caractéristiques physiques ou comportementales. Il existe de nombreuses applications, notamment la reconnaissance faciale ou vocale, beaucoup utilisées pour l'identification d'individus (ex : vidéo-surveillance, Home Assistants ...).

Les systèmes biométriques se décomposent en deux grands axes : la recherche de caractéristiques (*features*), c'est à dire des informations biométriques discriminantes, et la comparaison de ces dernières dans une base de données afin de déterminer à qui elles pourraient appartenir, dans le but d'identifier un individu.

Pour ce qui est de la recherche de caractéristiques, elle doit se faire en suivant ces principes :

— Les points d'intérêt récupérés doivent être le plus possible **discriminants**.
Un système biométrique doit représenter un individu d'une manière unique. Deux personnes différentes doivent donc avoir des caractéristiques différentes. Ainsi on s'assure que les informations biométriques récupérées font office de mot de passe unique.

— La récupération de caractéristiques doit être **robuste**.
Ainsi, il ne faut ni que ces informations discriminantes soient sensibles aux changements d'environnement (une voix doit pouvoir être reconnaissable même avec un léger bruit de fond), ni aux changements physiques légers (un changement de coupe de cheveux ne doit pas influer sur la reconnaissance de visage d'un individu).

2.3 Utilisation d'empreintes digitales dans un système biométrique

D'après ce que nous venons de voir, les empreintes digitales se prêtent particulièrement bien à un système biométrique de par leur unicité et faible changement chez une personne au cours du temps.
Les minuties sont, dans notre cas, extraites à partir des motifs de nos empreintes. On peut ensuite les répertorier dans une base de données afin d'authentifier un individu.
Par ailleurs, de nos jours leur usage ne s'arrête pas au domaine du judiciaire, et un bon nombre d'appareils multimédia proposent une identification biométrique pour remplacer les mots de passe. Nous nous sommes donc posés la question :

“ Comment pouvons-nous garantir l'identification sécurisée d'un individu à partir de ses empreintes digitales ? ”

Pour répondre à cette question, il nous est essentiel de regarder ce qui se fait de nos jours dans ce domaine.

2.4 Analyse d'empreintes : état de l'art

Les systèmes de traitement d'empreintes modernes fonctionnent le plus souvent de la manière suivante :

— On commence par *squelettiser* l'empreinte, dans le but de récupérer une image où les motifs de cette dernière sont les plus fins possible (un pixel d'épaisseur). La figure 1 présente un exemple de squelettisation d'empreinte digitale : On peut remarquer que les lignes restent bien contiguës, et que les lignes les plus épaisses ne font bien qu'un pixel d'épaisseur après squelettisation.

FIGURE 1 – Exemple de squelettisation (*thinning*) d'une empreinte digitale [1].

— On analyse alors cette image, afin d'identifier les points d'intérêt de celle-ci. On appelle ces points d'intérêt des caractéristiques (*features*), ou encore, pour des empreintes digitales, *minuties*. Les points considérés sont souvent les extrémités de lignes ainsi que les points de bifurcation entre deux lignes. Sur la figure 2 sont présentées les différentes caractéristiques remarquables d'une empreinte digitale. La plupart de ces caractéristiques sont réductibles à des extrémités et/ou à des bifurcations, étant donné que ces caractéristiques sont composées d'extrémités/bifurcations.

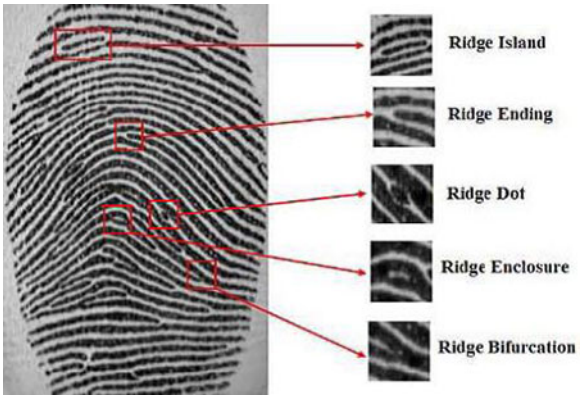

FIGURE 2 – Exemples de points d'intérêt dans une empreinte digitale [2].

— On associe alors à chaque minutie un code. Ce code doit donner une description des particularités de cette minutie, par exemple, de sa position ainsi que de son orientation.

— À partir de ces codes, on peut donc définir ou identifier les informations biométriques d'une personne en les comparant à des entrées dans une base de données.

2.5 Sécurisation : Coffre-fort flou (*Fuzzy Vault*)

Comme nous l'avons dit dans notre problématique, le sujet de la sécurisation de la base de données nous semble important. En effet, une des particularités des systèmes biométriques est la qualité le plus souvent immuable des informations sur lesquelles on travaille. Il est en effet très peu probable de changer d'empreintes, ou de visage, du tout au tout, en particulier sans intervention volontaire. De ce fait, il est primordial de garantir que les informations sensibles stockées dans les bases de données soient sécurisées.

Pour garantir cette sécurité, nous avons opté pour la mise en place d'une Fuzzy Vault [1].

L'intérêt d'une Fuzzy Vault est de garder un secret, ou plus particulièrement dans notre cas, un mot de passe. Il est possible de transcrire ce mot de passe en un polynôme $P(X)$: on associe à chaque lettre un entier, chaque caractère du mot de passe devient ainsi un coefficient du polynôme.

1. La traduction littérale de Fuzzy Vault est Coffre-fort flou. Nous n'avons cependant trouvé aucune référence à une telle appellation dans la documentation, nous utiliserons donc le terme Fuzzy Vault dans ce rapport.

Ce polynôme (secret) sera chiffré par l'information biométrique d'un individu. On appelle ces informations biométriques "originelles" des *minuties modèles (template minutiae)*.

Pour déduire ces *template minutiae*, on procède de la même manière qu'énoncé plus haut : Après pré-traitement de l'image d'empreintes digitales, les points d'intérêt sont mis en avant, puis encodés.

Nous disposons donc d'un secret, encodé dans un polynôme, ainsi que d'un ensemble $\mathbb{X}$ représentant l'encodage des minuties modèles.

$\forall x \in \mathbb{X}$, on insère dans la Fuzzy Vault les couples de $(x\,;P(x))$, soit, les couples "encodages de minuties ; évaluation de $P(X)$ en ces valeurs". On notera qu'il est nécessaire, pour pouvoir retrouver le secret, que $|\mathbb{X}| > degree(P)$.

La Fuzzy Vault contient donc assez de points d'évaluations d'un polynôme secret pour qu'il soit possible de l'interpoler. Pour assurer la protection de ce secret, on insère également dans notre Fuzzy Vault de l'aléa (appelé *chaff points* dans la littérature). En cas d'attaque visant à récupérer le message secret, la présence de cet aléa garantit la dissimulation des points d'intérêt. Cet aléa n'a pas de signification particulière, il ne sert qu'à brouiller les pistes, et empêche un déchiffrage en temps linéaire de notre Fuzzy Vault. On peut noter qu'une Fuzzy Vault contient généralement une bien plus grande proportion d'aléa que de points d'intérêt (10 à 100 fois plus).

Pour accéder au secret contenu dans la Fuzzy Vault, un individu fournit une image d'empreinte digitale. Cette image est alors traitée de la même manière que l'image modèle, et on récupère ainsi des minuties, que l'on encode.
Soit $\mathbb{X}'$ l'ensemble contenant ces nouvelles minuties encodées. Pour retrouver le secret, on cherche dans la Fuzzy Vault les points ayant une abscisse assez proche des $x \in \mathbb{X}'$. Par interpolation lagrangienne [3], il est alors possible, *tant qu'un nombre suffisant de minuties correspondent*, de retrouver le polynôme, et donc, le secret.
Vous trouverez dans la figure 3 un schéma explicatif du fonctionnement d'une Fuzzy Vault :

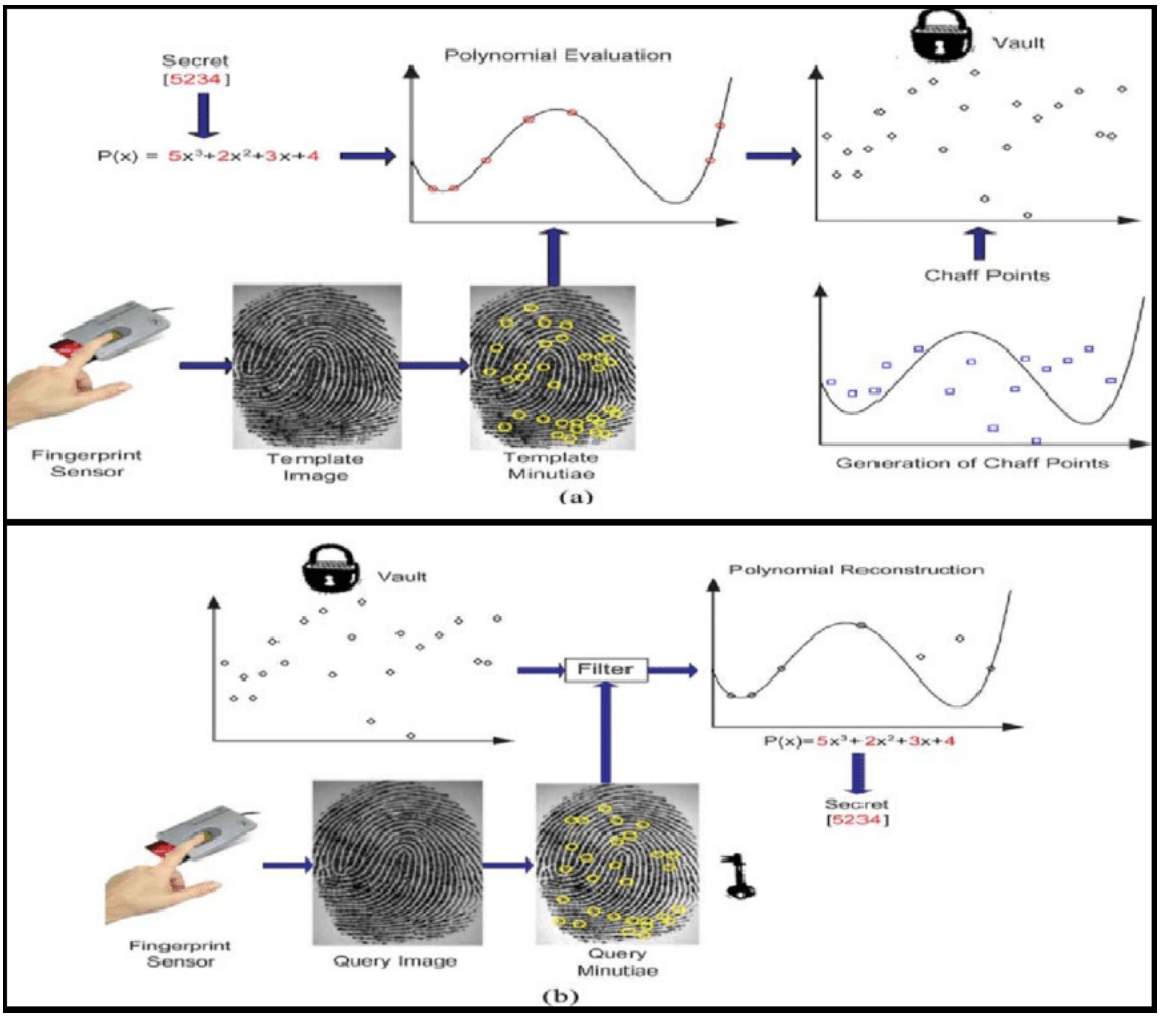

FIGURE 3 – Fonctionnement d'une Fuzzy Vault : en (a) le chiffrement, en (b) le déchiffrement [4].

3 Démarche initiale

3.1 Planification

Pour répondre à notre problématique, nous avons mis en place un premier plan :

1. Il nous fallait d'abord nous renseigner sur les différentes méthodes qui traitent de la reconnaissance d'empreintes et des Fuzzy Vault et choisir une base de données sur laquelle travailler. Étant donné que ce domaine nous était totalement inconnu, c'était une étape primordiale pour mener à bien notre projet.
2. Nous avions ensuite prévu de séparer notre travail en deux parties : D'un coté il nous fallait mettre en place un programme de pré-traitement des empreintes, pour nous permettre de squelettiser ces dernières, d'autre part on voulait commencer à mettre en place les mécanismes de chiffrement nécessaires pour l'utilisation d'une Fuzzy Vault. Nous avions vu en cours de traitements d'images le principe d'opérations morphologiques [5], et pensions

pouvoir récupérer un bon résultat en appliquant une suite d'érosions à notre image.

3. Nous comptions alors travailler sur la mise en évidence des points d'intérêt des images pré-traitées, c'est à dire sur la récupération de minuties.
4. Enfin, nous pensions finaliser le projet par la mise en place du processus de déchiffrement de la Fuzzy Vault. À ce stade nous étions censés avoir un programme fonctionnel et relativement stable, et avoir commencé la rédaction du présent rapport.

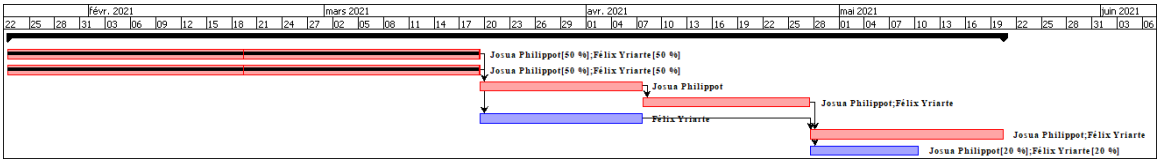

FIGURE 4 – Diagramme de Gantt de notre planification initiale.

Nous avions effectué un diagramme de Gantt initial, avant d'avoir totalement cerné les différentes difficultés du projet, celui-ci s'est ainsi avéré être très peu représentatif du temps que nous avons réellement passé sur les différentes parties de notre travail.

3.2 Base de données

Comme dit précédemment, nous avons réfléchi à une base de données sur laquelle débuter nos travaux. Notre choix s'est porté sur "DB3_B" du jeu de base de données FVC2000. Cette base de données est libre de droits, et est constituée de 80 empreintes de 19 volontaires âgés de 5 à 73 ans. Les images utilisées sont en 500 dpi pour du 448×478, capturées par des capteurs optiques "DF-90" par *Identicator Technology*.

Nous avons fait le choix de cette base de données parce qu'elle est celle avec la meilleure qualité d'image (pour la mise en place de notre programme, on désire travailler avec le meilleur cas possible, pour assurer que la théorie peut être mise en pratique) comparée aux autres proposées sur le site, en plus d'être assez vaste et libre d'utilisation.

3.3 Problèmes rencontrés

Alors que l'un d'entre nous essayait d'implémenter une fonction de squelettisation, l'autre utilisait des librairies *OpenCV* pour mettre en place une première version d'algorithme de récupération de minuties. Nous comptions assurer dans un premier temps le bon fonctionnement de notre Fuzzy Vault. Nous nous sommes ainsi rendus compte que les librairies OpenCV implémentaient déjà la plupart des traitements d'images dont nous avions besoin, tandis que nos implémentations ne fonctionnaient pas comme nous le souhaitions, et nécessitaient un long temps de développement.

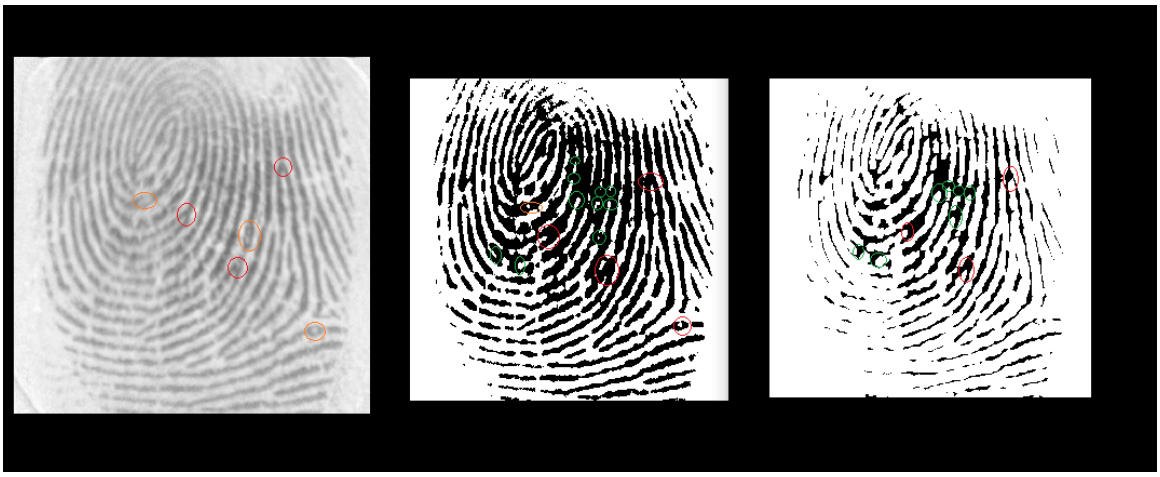

FIGURE 5 – Conjecture de minuties sur notre tentative de squelettisation.

La figure 5 est le résultat de notre prototype de squelettisation, sur lequel nous avons cherché "à la main" des minuties potentiellement utilisables. La perte d'information au fur et à mesure que l'on traite l'image est assez flagrante.
Alors que nous érodons progressivement l'image pour créer un squelette, nos minuties initiales (en rouge et orange) disparaissent, tandis que de nouvelles potentielles minuties (qui sont donc des faux positifs) apparaissent en vert.

C'est pourquoi nous avons décidé d'abandonner notre implémentation manuelle, au profit des fonctions déjà présentes sur OpenCV.

4 Méthodologie employée

4.1 Squelettisation d'une image d'empreinte

Dans un premier temps, il est nécessaire de seuiller l'image. Une valeur de seuil est définie automatiquement par la méthode d'Otsu [6]. L'image est alors traitée de cette manière : pour chaque pixel, si sa valeur en niveau de gris est inférieure à la valeur de seuil, le pixel est classé comme faisant partie du fond, il devient ainsi noir (0). Si, à l'inverse, sa valeur est supérieure à la valeur de seuil, le pixel est classé comme faisant partie de l'objet ; il devient blanc (1).

On obtient ainsi une binarisation de l'image de base. Pour squelettiser l'image, une heuristique est utilisée par OpenCV. Cette heuristique étudie pour chaque pixel (appelé pixel de référence) la disposition de ses voisins :

— Si trop ou trop peu de ces voisins sont considérés comme appartenant à l'empreinte, on ne considère pas le pixel de référence comme appartenant à l'empreinte.
— De même, si plusieurs "morceaux" d'empreinte entourent le pixel de référence (et non pas une ligne contiguë), celui-ci n'est pas considéré comme appartenant à l'empreinte.

4.2 Récupération de points d'intérêt

Pour pouvoir comparer deux empreintes digitales, il est nécessaire de sélectionner des points particuliers qui seront a priori sélectionnés à nouveau lors d'une future analyse d'empreinte.

4.2.1 Méthode de Harris

La base de code issue d'OpenCV utilise une détection de points d'intérêt par méthode de Harris pour identifier les minuties. Cette méthode se base sur la dérivation d'image : un gradient est appliqué à l'image après seuillage, les points à plus grande valeur absolue de dérivée sont marqués comme points d'intérêt.

Cette méthode est notamment utilisée pour la reconnaissance d'objet dans la création d'image panoramique ou encore pour effectuer un suivi d'objet (*tracking*) dans une vidéo.

Pour augmenter nos chances de sélectionner des caractéristiques qui seront choisies à nouveau lors de futurs traitement, potentiellement dans un environnement différent, il est nécessaire de sélectionner des points à grande valeur de dérivée. Ces points sont en effet particulièrement *différents* de leur voisinage (par définition du gradient d'une image).

Pour ce faire, on définit un seuil (*threshold*) sur les valeurs de dérivée au-dessus duquel on va garder les points d'intérêt pour la mise en place de nos minuties.
Nous avons commencé à utiliser la méthode de Harris sur notre base de données, et nous avons eu un premier résultat :

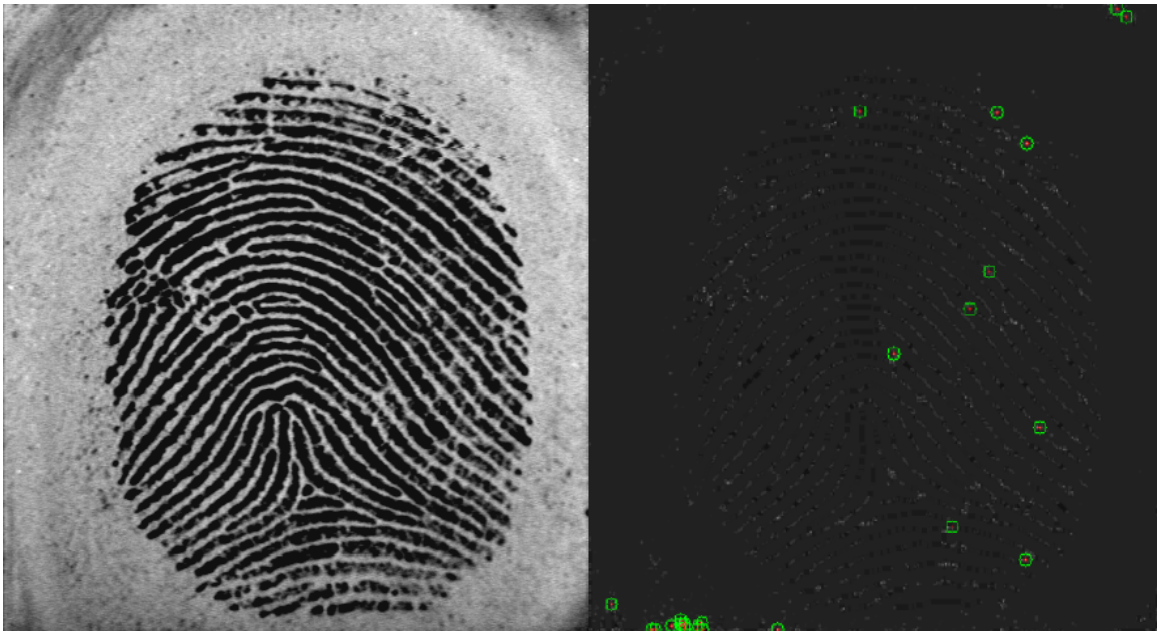

FIGURE 6 – Résultat de l'algorithme de Harris (droite) par rapport à l'empreinte originale (gauche).

Sur l'image de droite, figure 6, les cercles rouges (entourés de vert) représentent les points d'intérêt relevés par méthode de Harris. Malgré un seuillage censé filtrer les points d'intérêt (dans l'optique de garder seulement "les plus pertinents"), on remarque qu'une grosse partie de points marqués se trouve sur les bords de l'image.
Ces points, que l'on peut considérer comme des faux positifs, sont liés aux impuretés présentes sur l'image. En effet, même si cette dernière est d'assez bonne qualité, le capteur n'est pas parfait et est sensible au *bruit*. C'est ce bruit que les points de Harris relèvent. Pour remédier à cela, il va nous falloir "nettoyer" l'image, c'est à dire enlever au maximum les zones qui engendrent des faux positifs, sans supprimer d'informations liées à l'empreinte.

De plus, de par le nombre assez faible de points sélectionnés sur cet exemple, on peut constater que cette méthode est très sensible à la valeur de seuil choisie. La valeur de seuil idéale pour une image ne sera de plus pas la même quelle que soit l'image.

4.2.2 Nombre de croisements (*Crossing Number*)

Nos résultats en utilisant la méthode de Harris nous paraissaient perfectibles : il semblait délicat de bien choisir une valeur de seuil, ou d'automatiser ce choix. De plus, les minuties renvoyées par méthode de Harris ne correspondaient pas forcément aux résultats avec d'autres images de la même empreinte.

En parallèle de nos essais avec la méthode de Harris nous avons effectué des essais avec les Crossing Number [2]. La méthode par Crossing Number nous a été conseillée par nos encadrantes, qui l'utilisent dans le cadre de leurs travaux de reconnaissance de caractères.
Les Crossing Number sont définis comme les configurations du voisinage des point ne faisant pas partie du fond. Ainsi, suivant la valeur de Crossing Number d'un pixel et de son voisinage, on peut déduire s'il a des chances d'être un point d'intérêt.
Comme montré sur la figure 7, la valeur de CN représente le nombre de "changements de valeur" parmi les voisins d'un pixel ne faisant pas partie du fond. Ainsi, la position relative de ces changements n'influe pas sur la valeur de CN d'un pixel : celui-ci aura beau être une extrémité "gauche ou droite" de ligne, sa valeur de CN restera 1 dans les deux cas.

2. Une traduction de Crossing Number serait *nombre de croisements*, que nous n'avons pas retrouvé dans la littérature, c'est pourquoi nous utilisons le terme Crossing Number dans ce rapport.

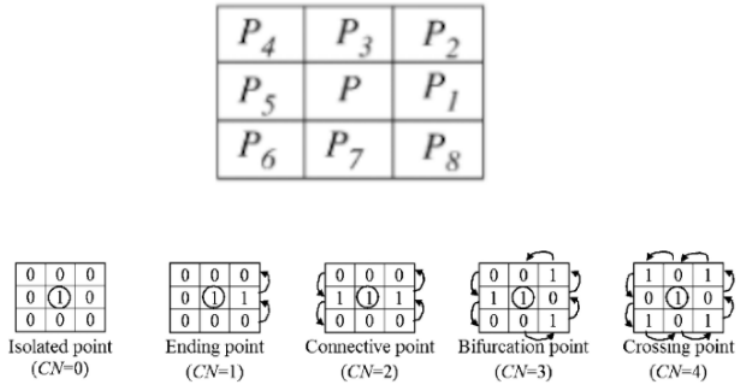

FIGURE 7 – Principe des Crossing Number [7].

La valeur de Crossing Number pour un pixel P, pour lequel le voisinage est défini par les P_i (avec une configuration semblable à la figure 7 par exemple) est :

$$CN(P) = \frac{1}{2}\sum_{i=1}^{8} |P_i - P_{i+1}| \quad avec\ P_1 = P_9$$

Pour notre utilisation, les valeurs de Crossing Number qui nous paraissaient intéressantes étaient surtout les CN = 1 et CN = 3. Celles-ci représentent respectivement les extrémités de lignes, et les points de bifurcation. Nous avons décidé de traiter ces configurations en particulier.

Un autre avantage conséquent à l'utilisation de cette méthode est l'absence de paramètres qu'il est nécessaire de régler pour son bon fonctionnement.

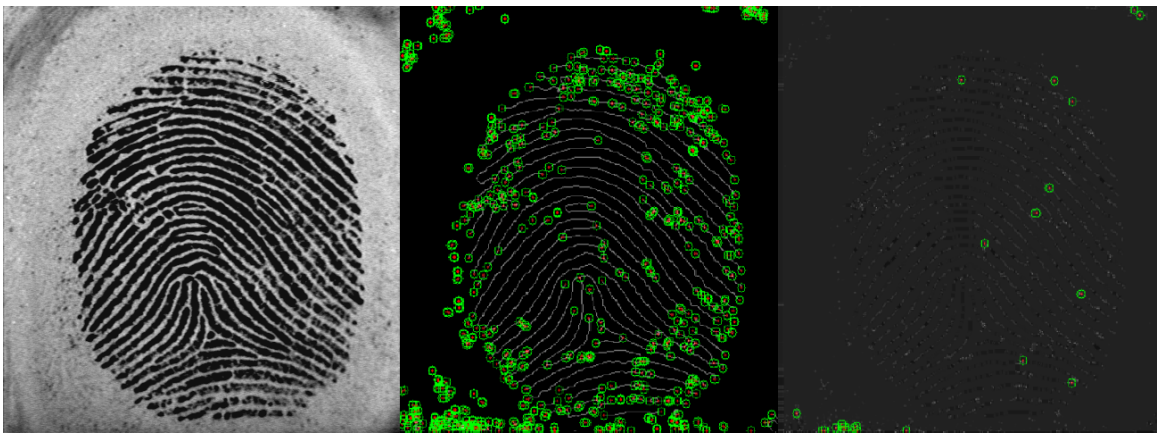

FIGURE 8 – Résultat de la méthode Crossing Number (centre) par rapport à l'empreinte originale (gauche) et Harris (droite).

On remarque sur ces résultats, présentés figure 8, que la méthode Crossing Number détecte un nombre bien plus important de minuties comparé à la méthode de Harris, sans pour autant

qu'elles soient moins pertinentes. On voit comme précédemment des minuties trouvées dans des zones de bruit (extérieures à l'empreinte), comme plus tôt, ce genre de minuties ne sera plus présent sur une empreinte "nettoyée". Nous avons décidé de ne plus traiter que des images nettoyées, afin de pouvoir avant tout vérifier le fonctionnement pratique de notre modèle théorique. Sur la figure 9 sont présentés nos résultats pour les mêmes méthodes que précédemment, mais sur l'image, une fois nettoyée :

FIGURE 9 – Empreinte *101_1* après nettoyage, mêmes traitements.

En appliquant nos analyses de points d'intérêt à des images nettoyées, les points récupérés ne sont que des points appartenant à l'empreinte ; ce qui nous permet de ne pas passer du temps à traiter le potentiel bruit amené par de mauvaises images.

4.3 Encodage de minuties

Une fois qu'un point a été considéré comme "d'intérêt", il est nécessaire de l'encoder, c'est à dire lui associer une valeur décrivant ses propriétés, telles que sa position, ou encore, son orientation.

La définition d'une bonne fonction d'encodage est primordiale, étant donné qu'une comparaison entre deux minuties ne portera dans notre système que sur des *distances* entre encodages. On définit un premier encodeur "naïf" sur 16 bits tel qu'il associe à un point la concaténation de son abscisse et de son ordonnée, sur 8 bits chacun. Il est alors possible de comparer 2 encodages par distance de Hamming.

Il existe cependant d'autres encodeurs plus discriminants, notamment par ORB descriptor [8] : cette méthode calcule déjà des positions *relatives* au barycentre de l'image, permettant ainsi de garder un repère relativement constant, quelle que soit l'image d'empreinte (tant qu'il s'agit bien de la même empreinte). De plus, cette méthode prend en compte l'orientation des points d'intérêt (encore une fois, relativement à un centre de masse). Cette couche descriptive supplémentaire rend les points d'intérêt plus discriminants : même si deux points de deux empreintes différentes ont des positions semblables, ils ne seront mis en correspondance que s'ils ont une orientation

similaire.

4.4 Vérification de la validité

Contrairement à ce que nous avions initialement prévu, il nous a semblé nécessaire de s'assurer que ces étapes précédentes effectuaient bien ce que nous attendions d'elles ; et surtout, qu'elles vérifiaient bien les propriétés qui étaient nécessaires au bon fonctionnement de notre système biométrique. Par exemple, vérifier que les minuties trouvées sur une empreinte restent les mêmes pour une autre image de la même empreinte.

4.4.1 Définition d'une mise en correspondance (*match*) entre deux minuties

On définit la distance entre deux minuties encodées comme le nombre de bits qui diffèrent entre ces deux représentations (distance de Hamming [9]). On ne peut pas s'attendre à ce que deux minuties censées représenter un match aient une distance nulle ; il faut définir une distance maximale séparant 2 minuties considérées comme "mises en correspondance". Dans le but de trouver une valeur de distance optimale, nous avons mis en place un comparatif :

4.4.2 Recherche d'une distance maximale optimale

Si la distance maximale choisie est trop faible, deux minuties censées matcher risquent de ne pas matcher. Lors de l'encodage de celles-ci, ou même du pré-traitement appliqué, de légères différences peuvent apparaître, ce qui pourrait impliquer une distance les séparant trop importante pour les considérer comme représentant un match. À l'inverse, en choisissant une distance trop grande, des minuties différentes risquent de matcher. Notre Fuzzy Vault serait alors déverrouillable par d'autres personnes que la personne souhaitée. Nous avons ainsi décidé de faire varier la distance maximale (de 10 à 100, en 10 étapes). Pour chaque valeur de distance, on compte le nombre de matchs entre deux images, tout en notant si elles viennent d'une même empreinte ou non.

Cela nous permet de constituer une courbe ROC : les matchs peuvent avoir 4 valeurs de vérité :

— Vrai positif (**VP**) : correspond à un match correct, c'est à dire que la minutie sur l'empreinte 1 de la personne A est associée à une minutie sur l'empreinte 2 de la personne A

— Faux positif (**FP**) : correspond à un match incorrect, c'est à dire que la minutie sur l'empreinte 1 de la personne A est associée à au moins une minutie sur l'empreinte 1 de la personne B

— Vrai négatif (**VN**) : correspond à un "non match", c'est à dire que la minutie sur l'empreinte 1 de la personne A n'est associée à aucune minutie sur l'empreinte 1 de la personne B

— Faux négatif (**FN**) : correspond à un "non match" erroné, c'est à dire que la minutie sur l'empreinte 1 de la personne A n'est associée à aucune minutie sur l'empreinte 2, alors qu'elle est effectivement celle de la personne A.

Avec ces valeurs de vérité on peut calculer la **sensibilité**, qui correspond à la proportion des positifs détectés parmi tous les résultats réellement positifs, et la **spécificité** qui correspond à la

proportion de vrais négatifs parmi les résultats effectivement négatifs. On les calcule de la façon suivante :

$$\textbf{Sensibilité} : \frac{VP}{(VP + FN)}$$

$$\textbf{Spécificité} : \frac{VN}{(VN + FP)}$$

Une fois ces valeurs calculées, on peut tracer la courbe **ROC**, qui permet de mesurer la performance des classificateurs binaires, représentée par la *Sensibilité* en fonction de l'*Antispécificité* (soit 1 - *Spécificité*).

Les courbes colorées de la figure 10 représentent les courbes ROC associées à différentes empreintes de référence :

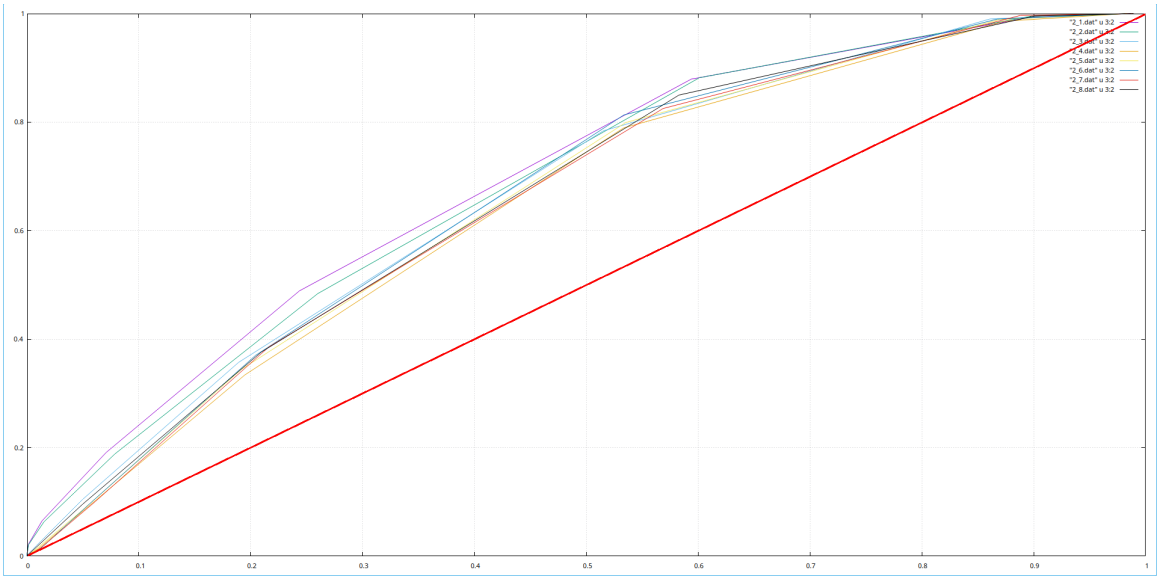

FIGURE 10 – ROC comparatifs des Crossing Number

Sur ce graphique, nous avons tracé les courbes ROC de plusieurs empreintes : on compare une empreinte avec toutes les autres. Notre idée est de visualiser une "forme moyenne" des courbes pour en déduire la meilleure distance pour considérer 2 minuties comme constituant un match. La meilleure valeur de distance est représentée par le point de la courbe ROC le plus proche de (0,1). Si la courbe ROC atteint ce point, c'est que tous les matchs sont de vrais positifs (et tous les non-matchs, de vrais négatifs), et donc que notre méthode est parfaitement fiable.

À l'inverse, une méthode peut être jugée comme peu fiable si sa courbe ROC associée passe de (0,0) à (1,1) en suivant une diagonale (ici en rouge). Cela revient pour un classificateur à ce qu'il réponde vrai ou faux, de manière aléatoire.

Ici, on peut voir que nos courbes sont toutes assez proches de la diagonale, aucune valeur de distance maximale ne donne de points sensiblement plus proches du point (0,1). On en conclut que notre méthode n'est pas très fiable.

4.4.3 Problèmes rencontrés et solutions mises en place

Même pour une comparaison de deux images issues de la même empreinte, les minuties peuvent différer : suivant l'orientation, la position du doigt, ou même la pression appliquée lors de la capture ; des petites différences sont perceptibles. Soient deux images Im_1 et Im_2 issues d'une même empreinte. Après traitement et analyse de celles ci, on récupère respectivement les ensembles de minuties $\mathbb{X}_1$ et $\mathbb{X}_2$. Il est très peu probable qu'on aie $\mathbb{X}_1 = \mathbb{X}_2$; en réalité, on aura seulement une partie des minuties en commun dans les deux ensembles, tandis qu'aucun match ne devrait être trouvé pour une majorité des minuties. Il est cependant délicat de savoir à l'avance et de manière automatique quelles minuties sont bien censées matcher, et quelles minuties ne le sont pas, ce qui est nécessaire à l'entraînement du classificateur (étant donné qu'il s'agit d'une classification supervisée).

En prenant un grand nombre de points, on augmente le risque de comparer des minuties qui ne devraient pas renvoyer de matchs. D'autre part, les points du "bord" de l'empreinte sont plus à même de varier suivant l'image choisie pour une même empreinte.

Il est important de noter que dans une Fuzzy Vault, quel que soit le nombre de minuties pour lesquelles aucun match n'est trouvé, tant qu'un *assez grand nombre* de match est présent, il est possible de retrouver le secret.

4.4.4 Sélection de minuties intéressantes

Nous avons donc décidé de mettre en place une sélection de minuties. Les minuties qui sont pour nous les plus intéressantes, sont celles qui sont présentes sur le plus d'images possible : nous souhaitons récupérer des minuties présentes sur l'empreinte, quel que soit l'angle de prise.

Nous avons trouvé dans la littérature [10] différentes manières d'implémenter de la sélection de minuties. Nous avons ainsi décidé de ne considérer que les minuties les plus centrées, étant donné qu'il s'agit de celles dépendant le moins de l'angle de prise, ou encore de la position du doigt lors de la capture. On rencontre alors un problème : comment trouver le centre de l'empreinte, alors qu'il est possible que l'image ne soit pas parfaitement centrée ?

Pour pallier cette problématique potentielle, on considère comme "centre" de nos minuties le centroïde de masse. Soit $\mathbb{X}$ l'ensemble de minuties récupérées, le centre de masse est défini comme :

$$C = \frac{1}{|\mathbb{X}|} \sum_{i=1}^{|\mathbb{X}|} x_i \quad , x_i \in \mathbb{X}$$

Il ne nous reste alors qu'à trier les minuties par proximité décroissante à ce centroïde de masse ; on ne considère que les premières valeurs dans cet ordre, c'est à dire les points les plus proches du centroïde.

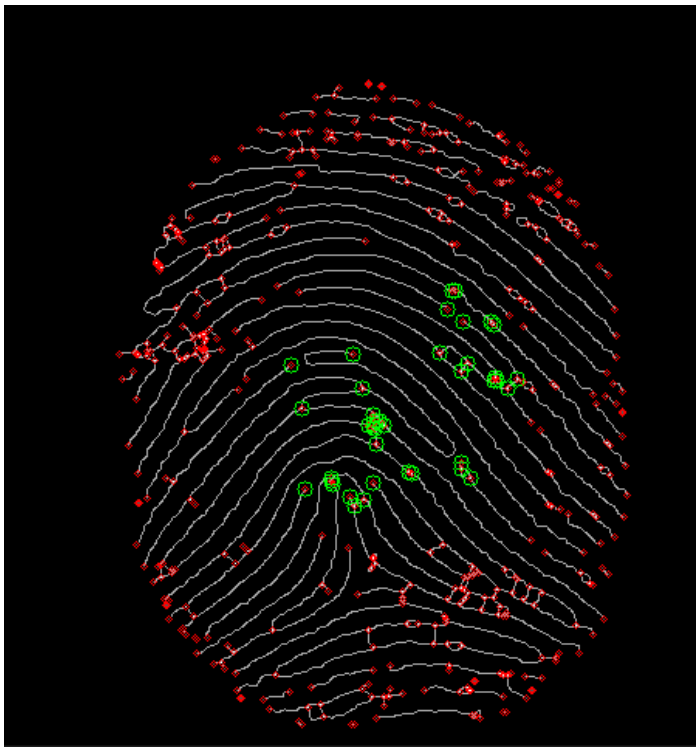

FIGURE 11 – Minuties considérées en imposant une proximité au centre de masse

Nous avons donc réalisé cette sélection, comme vous pouvez le voir sur la figure 11 : tous les cercles rouges correspondent à des points considérés comme minuties potentielles, c'est à dire tels que leur valeur de Crossing Number soit de 1 ou 3, tandis que les cercles verts représentent les minuties effectivement choisies, après calcul de leur distance au centre de masse. Les minuties "vertes" sont effectivement plus centrées ; on espère ainsi récupérer des minuties plus invariantes à la prise d'empreinte digitale.

5 Fuzzy Vault

Par manque de temps, nous n'avons pas pu mettre en place l'implémentation de la Fuzzy Vault. Nous avons cependant pu travailler sur les différentes démarches nécessaires au bon fonctionnement de celle-ci.

Pour implémenter l'approche Fuzzy Vault, nous avons décidé d'utiliser un encodeur naïf et simple plutôt que l'ORB descriptor. En effet, il nous est nécessaire de bien comprendre comment sont encodés les points pour définir des distances entre ceux-ci. L'utilisation de l'ORB descriptor nous semble complexe, et nous n'avons pas été à même de nous l'approprier.

Pour la création de la Fuzzy Vault, tout comme pour son déchiffrement, il est nécessaire que les minuties soient suffisamment espacées. Théoriquement, si on considère lors du déchiffrement que deux minuties matchent tant qu'elles sont au plus distantes de δ_1, il est nécessaire de considérer des minuties distantes au minimum de $2 \times \delta_1$. Si cette propriété n'est pas vérifiée,

plusieurs minuties proches risquent de matcher avec un unique point présent dans la Fuzzy Vault. Pour pallier cela, on décide de ne considérer sur l'empreinte de requête que des points *assez* éloignés entre eux. Pour ce faire, lors de la récupération des minuties proches du centroïde, une condition supplémentaire est mise en place : on calcule la distance d'un point considéré avec tous les autres points déjà acceptés, si une de ces distances est trop faible, on n'accepte pas le point. Dans l'exemple exposé figure 12, on peut voir que les minuties trouvées (en rouge) sont les mêmes que sur la figure 11. Cependant, les minuties sélectionnées (en rouge + vert) ne sont que les minuties assez distantes des autres minuties acceptées.

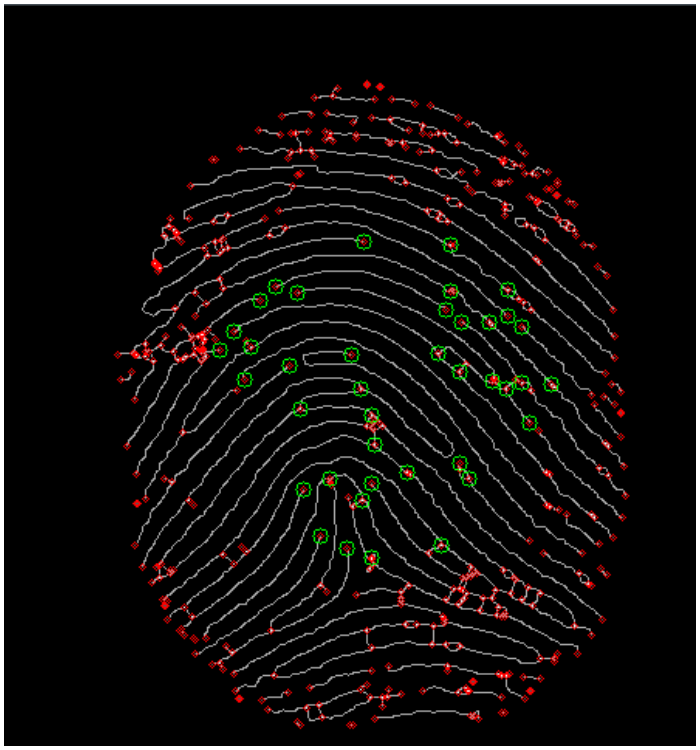

FIGURE 12 – Minuties considérées en imposant un espace entre minuties

Nous avons fixé arbitrairement le nombre de minuties souhaité à 40 : avec 40 minuties, il est théoriquement possible d'encoder un secret faisant jusqu'à 40 caractères (caractères encodés sur autant de bits que l'encodage d'une minutie ; soit 16 dans notre cas). Cependant, une Fuzzy Vault repose notamment sur le fait qu'une partie des minuties peut ne pas correspondre ; il serait donc contraire au but même de notre Fuzzy Vault d'encoder un secret de 40 caractères. En utilisant un secret de 40 caractères, on ne s'autoriserait aucune correction d'erreur, ce qui n'est pas pertinent. On estime donc plutôt pouvoir encoder un secret faisant jusqu'à 20 caractères, ce qui parait être du même ordre qu'une taille ordinaire de mot de passe.

6 Conclusion

Nous avons réussi à mettre en place un système de pré-traitement d'images d'empreintes digitales, afin de détecter, puis sélectionner des caractéristiques discriminantes et invariantes sur celles-ci. Nous avons tenté d'optimiser ces opérations, afin de garantir un bon fonctionnement, dans l'optique de pouvoir utiliser le concept de Fuzzy Vault, que nous n'avons pas eu le temps d'implémenter jusqu'au bout. Il nous semble cependant que nos travaux vérifient bien les propriétés nécessaires à une telle implémentation.

Bien que nous n'ayons pas particulièrement travaillé sur l'aspect "sans contact" d'analyse d'empreintes digitales, notre chaîne de traitements se prête bien à une telle opération. Il faudrait appliquer aux images prises sans contact des transformations morphologiques peu différentes de celles que nous avons présentées ici. Une propriété absolument nécessaire à cette démarche est que ces images soient d'une résolution suffisante pour pouvoir y discerner une empreinte.

Nous avions travaillé pour un différent projet sur un système biométrique de reconnaissance de visages. Pour traiter les caractéristiques d'un visage, une toute autre approche est utilisée : on considère des *Local Binary Patterns*, soit une description approximative des zones du visage. Tandis que par la méthode de *Crossing Number* nous décrivons des points particulièrement intéressants, avec la méthode par *Local Binary Patterns*, une description **vague** des zones du visage est utilisée. Pour aller plus loin, on pourrait s'intéresser à l'application de cette méthode dans le cadre du traitement d'empreintes digitales.

Références

[1] A. Bhargava and S. Bhargava. Techniques for minutiae based matching of low quality fingerprints images - a general overview. *International Journal of Scientific Research and Review*, 7(7) :1–6, 2018.
https://www.researchgate.net/figure/Fingerprint-image-before-left-and-after-right-ridge
fig5_326682707.

[2] Danny Thakkar. Minutiae based extraction in fingerprint recognition. *Bayometric*, 2016.
https://www.bayometric.com/minutiae-based-extraction-fingerprint-recognition/.

[3] Wikipédia. Interpolation lagrangienne. *Wikipédia L'encyclopédie libre*.
https://fr.wikipedia.org/wiki/Interpolation_lagrangienne.

[4] Aliakbar Nasiri ; Mahmood Fathy and Mina Zolfy Lighvan. A new approach to alignment-free fingerprint cryptosystem using fuzzy vaults. *Journal of Electrical Systems and Signalskk*, 2(2) :39, 2015.
https://www.researchgate.net/publication/329130147_A_New_Approach_to_
Alignment-Free_Fingerprint_Cryptosystem_Using_Fuzzy_Vaults.

[5] Wikipédia. Morphologie mathématique. *Wikipédia L'encyclopédie libre*.
https://fr.wikipedia.org/wiki/Morphologie_math%C3%A9matique.

[6] Wikipédia. Méthode d'otsu. *Wikipédia L'encyclopédie libre*.
https://fr.wikipedia.org/wiki/M%C3%A9thode_d%27Otsu.

[7] Feng Zhao and Xiaoou Tang. Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. *Pattern Recognition 40*, 1270 — 1281 :1272 – 1273, 2006AbstractIn.
https://www.researchgate.net/publication/222823428_Preprocessing_and_
postprocessing_for_skeleton-based_fingerprint_minutiae_extraction.

[8] OpenCV. Orb (oriented fast and rotated brief) adapted from *Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski : ORB : An efficient alternative to SIFT or SURF. OpenCV documentation*.
https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html.

[9] Wikipédia. Distance de hamming. *Wikipédia L'encyclopédie libre*.
https://fr.wikipedia.org/wiki/Distance_de_Hamming.

[10] B Vibert ; Christophe Charrier ; Jean-Marie Le Bars and Christophe Rosenberger. Comparative study of minutiae selection algorithms for iso fingerprint templates. 2015.
https://hal.archives-ouvertes.fr/hal-01120639/document.