



HAL
open science

Etude d'algorithmes d'authentification pour petits capteurs d'empreinte digitale

Mathilde Bourjot, Régis Perrier, Jean-François Mainguet

► **To cite this version:**

Mathilde Bourjot, Régis Perrier, Jean-François Mainguet. Etude d'algorithmes d'authentification pour petits capteurs d'empreinte digitale. CORESA 2017, Nov 2017, Caen, France. hal-01667407

HAL Id: hal-01667407

<https://hal.science/hal-01667407>

Submitted on 19 Dec 2017

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.

Etude d'algorithmes d'authentification pour petits capteurs d'empreinte digitale

Mathilde Bourjot

Régis Perrier

Jean-François Mainguet

CEA Leti - Grenoble *
Département Systèmes
{prenom.nom}@cea.fr

Résumé

Les nouveaux capteurs d'empreintes digitales intégrés dans les smartphones ont déclenché une petite révolution dans le domaine de la biométrie. Ils permettent aux utilisateurs de s'authentifier sans avoir à retenir de code, tout en garantissant une sécurisation des données annoncée comme très bonne par les fabricants. Cependant, la taille de ces capteurs de l'ordre de quelques millimètres carrés proscrit l'usage d'algorithmes d'authentification à minuties. A notre connaissance, peu d'études se sont penchées sur ce problème, malgré l'existence de nombreux algorithmes de reconnaissance de formes et de solutions commerciales supposées robustes. Cette étude cherche à apporter une première réponse en analysant les performances de trois algorithmes pour cette application.

Mots clefs

empreinte digitale, petits capteurs, authentification, descripteurs, points saillants, corrélation.

1 Introduction

L'intérêt des empreintes digitales pour l'identification humaine n'est plus à démontrer depuis longtemps. Si la police les utilise depuis des décennies, leur usage dans les produits commerciaux tels que les ordinateurs et les PDA (*Personal Digital Assistant*) a commencé vers 2004 où le nombre de capteurs vendus se comptait en millions.

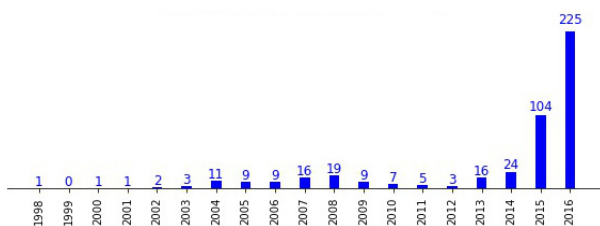


Figure 1 – Nouveaux modèles de téléphone portable avec un capteur d'empreinte (source : mainguet.org).

Les smartphones ont emballé le mouvement en 2013 lorsqu'Apple a intégré la reconnaissance d'empreinte dans son iPhone. Aujourd'hui la production de capteur atteint le milliard par an.

Si les applications gouvernementales s'accroissent de

* Cette étude a été financée par un projet Carnot interne au CEA Leti

grands capteurs optiques, ce n'est pas le cas des smartphones où ces capteurs pour des raisons de coût et d'intégration sont de plus en plus petits, ce qui n'est pas sans poser de problèmes. En effet, l'écart de taille entre les capteurs gouvernementaux de plusieurs centimètres carrés enregistrant le doigt entier comparée à celle des capteurs intégrés aux smartphones des quelques millimètres carrés¹ pose question quant à la fiabilité de ces derniers.

Les applications gouvernementales utilisent systématiquement des algorithmes à minuties ; c'est un standard international. Il a déjà été démontré que ces algorithmes avaient une limite sur le recouvrement minimal entre deux images se situant aux alentours de 7mm pour un capteur à balayage [1]. Une étude plus récente [2] démontre également que plus la surface du capteur est importante, meilleurs sont les taux de vrai refusés (FRR) à un taux de faux acceptés (FAR) fixe de 1 pour 10000 : le FRR passe de l'ordre de 1% pour un grand capteur à 20% pour un capteur de 8x8mm².

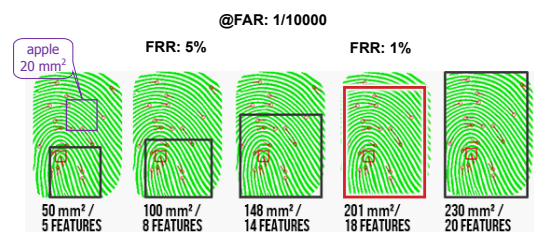


Figure 2 – Taux de reconnaissance suivant la taille du capteur pour un algorithme à minutie [2].

Cependant, ces études sont fondées sur des algorithmes « classiques » dits à minuties. Ces dernières sont des caractéristiques d'une empreinte digitale dites de niveau 2 ; le niveau 1 étant la forme générale de l'empreinte. Etant donnée les taux de reconnaissance annoncés, et a priori jamais testés ou publiés, par les vendeurs de smartphones², il paraît certain que les algorithmes utilisés pour des petits capteurs utilisent d'autres caractéristiques, comme celles dites de niveaux 3 (pores, variations d'épaisseur des lignes, ...). Dans cet article, nous comparons trois algorithmes extraits de la littérature en vision par ordinateur pour cette application d'authentification d'empreinte digitale acquise par un petit capteur : une méthode dite directe et deux approches

1. l'iPhone5 avait un capteur de 4.5x4.5mm²

2. par exemple Apple annonce un FAR de 1/50000 sans FRR fixé

par descripteurs. A notre connaissance, peu de travaux ce sont intéressés à l'utilisation de méthodes alternatives aux minuties pour la reconnaissance d'empreinte digitale, à des exceptions près comme [3, 4] qui utilisent des bancs de filtres de Gabor et dont les résultats s'approchant des méthodes à minuties, et très récemment [5] qui effectue une comparaison de nombreuses méthodes à descripteurs sur des images d'empreintes de mauvaise qualité et dans laquelle SIFT se démarque. Les méthodes à minuties ont largement satisfait la demande en algorithmes robustes et rapides depuis les années 90, mais l'introduction de petits capteurs a changé la donne. Plusieurs solutions industrielles existent sans qu'il y ait eu d'analyse précise dans la littérature sur les moyens de parvenir à un algorithme de reconnaissance satisfaisant dans ce contexte. Notre étude cherche à apporter une première réponse dans ce sens à l'aide d'algorithmes classiques et éprouvés.

2 Vocabulaire en biométrie

Les systèmes biométriques ont deux phases [6] :

- Une première, l'**enrôlement**, capture les caractéristiques biologiques de l'utilisateur, ici l'image de l'empreinte digitale. Un traitement réduit l'image initiale à des caractéristiques qui sont enregistrées en mémoire.
- Une seconde, la reconnaissance ou l'**authentification**, consiste à capturer une image **candidate**, qui est comparée à l'enregistrement réalisé lors de l'enrôlement. Un algorithme de comparaison retournera un score de similarité, et décidera si l'image candidate est suffisamment proche pour retourner un résultat positif, qui dépendra d'un seuil pré-réglé.

Dans le cas des petits capteurs, l'utilisateur doit présenter son doigt de manière répétitive pour enregistrer une surface de doigt suffisante afin d'augmenter les chances plus tard d'en reconnaître une portion. Il est logiquement nécessaire de présenter la même portion de peau à l'enrôlement et à la reconnaissance pour reconnaître quelqu'un.

On définit habituellement deux taux :

- **FAR = faux accepté** ; il s'agit pour celui-ci d'un problème de sécurité. Il doit être de l'ordre d'au moins 1 pour 10000 pour un système commercial, alors que la police requiert des taux de 1 pour 1 million pour exécuter des recherches dans des bases de données de millions de personnes.
- **FRR = vrai refusé** ; c'est généralement ce taux que les utilisateurs expérimentent *de facto*, il correspond à un problème de commodité. Il faut qu'il soit suffisamment faible pour ne pas rejeter le système, et est de l'ordre de 1% pour un bon système.

Ces deux taux sont liés : si on diminue le taux de vrai refusé pour que le système soit pratique, alors on augmente forcément la possibilité pour un imposteur d'entrer dans celui-ci. Ils sont couramment représentés sous forme de courbe ROC (*Receiver Operating Characteristic*) comme nous le verrons dans la partie résultat.

3 Calcul du score de similarité

Nous présentons ici les trois méthodes permettant de calculer un score de similarité entre deux images enrôlée et candidate, respectivement notées E et C par la suite. Ces méthodes sont extraites de l'état de l'art et servent de référence en terme de performance de reconnaissance dans notre contexte de capteur biométrique de petite surface.

La première utilise globalement l'intensité des pixels des images pour calculer un score ; elle fait partie de la classe des méthodes appelées directes [7, 8]. Les deux suivantes utilisent des descripteurs pour mettre en correspondance les images, elles sont aujourd'hui largement plus répandues que les premières [9, 10].

Nous considérons que les images sont définies sur une grille régulière discrète de taille $(m \times n)$ pixels de sorte que $E(x, y)$ corresponde à l'intensité à valeur dans \mathbb{R} de l'image E au pixel de position $\mathbf{p} = [x, y]^T$.

3.1 Approche directe

La première méthode utilise la corrélation croisée centrée et normalisée pour calculer une surface de corrélation $\gamma(u, v)$ entre les deux images par translation de l'une par rapport à l'autre :

$$\gamma(u, v) = \frac{\sum_{x, y \in \mathcal{S}} (E(x, y) - \mu^E)(C(x - u, y - v) - \mu^C)}{\sigma^E \sigma^C} \quad (1)$$

pour $u \in [-n+1, n-1]$ et $v \in [-m+1, m-1]$. Le support \mathcal{S} contient l'ensemble des coordonnées des pixels $\{(x, y)\}$ tel que $x - u \in [1, n]$ et $y - v \in [1, m]$; il varie donc selon u et v . Les paramètres statistiques sont recalculés pour chaque valeur de u et v avec :

$$\mu^E = \frac{\sum_{x, y \in \mathcal{S}} E(x, y)}{\text{Card}(\mathcal{S})} \text{ et } \mu^C = \frac{\sum_{x, y \in \mathcal{S}} C(x - u, y - v)}{\text{Card}(\mathcal{S})},$$

ainsi que :

$$\sigma^E = \sqrt{\sum_{x, y \in \mathcal{S}} (E(x, y) - \mu^E)^2},$$

$$\sigma^C = \sqrt{\sum_{x, y \in \mathcal{S}} (C(x - u, y - v) - \mu^C)^2}.$$

Le score de similarité final retenu est le maximum de la surface de corrélation :

$$s_{\text{cor}} = \max \gamma(u, v) \quad (2)$$

On peut relever au moins deux avantages à cette méthode : elle est invariante à un changement de luminosité global dans les images et il existe une implémentation rapide de celle-ci [11] ; ces deux critères l'ont souvent avantagée en pratique au dépend de métriques plus robustes [12]. En revanche, elle n'est pas invariante aux rotations et ses performances se dégradent suivant l'amplitude des déformations géométriques locales entre les deux images.

Pour compenser sa sensibilité aux rotations dans cette étude, nous calculons plusieurs scores de corrélation entre l'image E et l'image C tournée par pas de 5 degrés entre -30 et 30 degrés, et retenons le score maximal.

3.2 Approches par descripteurs

L'utilisation de descripteurs a supplanté depuis de nombreuses années les approches directes sur la question du recalage d'image [8], en plus d'étendre ses compétences dans des domaines comme la reconnaissance de forme. Le schéma d'utilisation de ceux-ci est assez commun selon les applications visées [7], nous résumons ici les étapes pertinentes pour notre sujet :

1. **identification de points saillants** : L'objectif est de rechercher dans les deux images E et C des coordonnées de pixels, \mathbf{p}^E et \mathbf{p}^C respectivement, tels que leur mise en correspondance soit facilitée par la suite. La littérature est abondante sur le sujet [13, 9, 10], mais il est admis à minima qu'un point saillant doit présenter des variations de contraste autour de lui ; autrement dit les dérivées spatiales de l'image dans l'entourage de ce point ne doivent pas être nulles. A l'issue de cette étape, nous obtenons les ensembles $\{\mathbf{p}^E\}$ et $\{\mathbf{p}^C\}$ pour les deux images, non nécessairement de même dimension.
2. **calcul de descripteurs** : Toujours dans l'optique de la mise en correspondance, le descripteur cherche à caractériser au mieux la région autour du point saillant. Il doit être discriminant et compact dans la quantité d'information qu'il contient pour un appariement efficace en terme de robustesse et vitesse de calcul. Une nouvelle fois les travaux sont nombreux sur ce sujet, exploitant la répartition spatiale des points pour capturer une forme [14], des histogrammes sur la distribution des gradients de l'image [9], ou des patches orientés [7]; une vue plus exhaustive est donnée par [8]. Après cette étape, chaque point \mathbf{p}_i est donc associé à un descripteur \mathcal{D}_i qui peut prendre la forme d'un vecteur ou d'une matrice d'information variée (histogramme, patch, ...).
3. **mise en correspondance** : cette étape recherche les liaisons une à une entre les points $\{\mathbf{p}^E\}$ et $\{\mathbf{p}^C\}$ par comparaison de leurs descripteurs respectifs. La stratégie la plus simple consiste à rechercher pour chaque point d'indice i de E celui qui minimise la fonction distance h de son descripteur avec ceux de C :

$$\hat{k} = \underset{k}{\operatorname{argmin}} h(\mathcal{D}_i^E, \mathcal{D}_k^C), \quad (3)$$

où \hat{k} est l'indice de l'élément de $\{\mathcal{D}^C\}$ choisi pour la mise en correspondance avec \mathcal{D}_i^E . La fonction h peut par exemple être une distance euclidienne ou de Hamming suivant la caractéristique du descripteur choisi. Pour obtenir une bijection entre les ensembles de points, la même procédure peut être réalisée en inversant les rôles des images, ainsi seuls les appariements qui satisfont le critère de coût minimum de façon symétrique sont retenus ; c'est l'approche qui est utilisée dans ce travail. Nous obtenons alors un nouvel ensemble de points liés entre les deux images $\{\bar{\mathbf{p}}^E, \bar{\mathbf{p}}^C\}$. Notons qu'il existe d'autres stratégies plus élaborées tel que l'algorithme hongrois pour trouver un couplage de poids minimum

dans un graphe biparti [14], ou encore des méthodes de tri des descripteurs dans des arbres ou table de hash pour accélérer leur mise en correspondance [9, 7]. Il est également possible de considérer deux minimums plutôt qu'un pour améliorer la robustesse de cette étape par un test du ratio [9], la motivation étant de déterminer à quel point le choix d'une mise en correspondance serait meilleure que la suivante dans la liste.

Afin de satisfaire une contrainte géométrique liée à notre problème selon laquelle le doigt peut subir une rotation et une translation dans le plan du capteur entre deux acquisitions, une étape supplémentaire de minimisation de fonction permet d'exclure les appariements aberrants :

$$\hat{\theta}, \hat{\mathbf{t}} = \underset{\theta, \mathbf{t}}{\operatorname{argmin}} \sum_i \rho(\bar{\mathbf{p}}_i^E - R(\theta)(\bar{\mathbf{p}}_i^C + \mathbf{t})) \quad (4)$$

où $R(\theta)$ désigne la matrice de rotation dans le plan image et ρ est une fonction permettant un calcul robuste de $\hat{\theta}$ et de $\hat{\mathbf{t}}$. Ces derniers ne sont pas directement d'intérêt, ils ne servent ici que de contrainte dans la détermination d'un nouvel ensemble de points $\{\bar{\mathbf{p}}^E, \bar{\mathbf{p}}^C\}$ qui satisfont chacun :

$$\|\bar{\mathbf{p}}_i^E - R(\theta)(\bar{\mathbf{p}}_i^C + \mathbf{t})\| < \epsilon, \quad (5)$$

pour chaque indice i des éléments de l'ensemble de points, où $\|\cdot\|$ désigne la norme euclidienne et ϵ est un scalaire de valeur suffisamment petite. Ceci nous permet de définir le score de similarité final entre E et C :

$$s_{\text{desc}} = \operatorname{Card}(\{\bar{\mathbf{p}}^E, \bar{\mathbf{p}}^C\}). \quad (6)$$

Littéralement le score correspond au nombre de bonnes mises en correspondance entre les deux images à l'issue de toutes les étapes précédentes. La fonction ρ peut prendre des formes multiples, comme appartenir à la classe des M-estimateurs. En pratique nous utilisons l'algorithme de RANSAC [15] pour sa bonne adéquation avec le problème à résoudre.

Il faut préciser que la mise en correspondance proposée ici n'est pas optimale en terme de vitesse de calcul ; ce n'est pas l'objectif de cette étude qui se concentre sur les 2 premières étapes. Par expérience, nous avons constaté que RANSAC rendait l'algorithme général insensible au choix d'une méthode d'appariement de descripteurs plus élaborée que l'approche naïve et exhaustive que nous utilisons. Cependant il serait intéressant de chercher à s'affranchir de RANSAC par un meilleur appariement de descripteurs au départ. Nous pouvons maintenant détailler les différences entre les deux approches considérées dans ce papier.

SIFT. Cet algorithme qui a pour acronyme *Scale Invariant Feature Transform* a été très populaire depuis le début des années 2000 [9] et reste une référence incontournable en vision par ordinateur. Il optimise toutes les étapes précédemment décrites pour aboutir à une détection de points saillants et une mise en correspondance ayant pour caractéristiques principales :

- une grande robustesse aux transformations affines et aux changements d'échelle entre les images, ainsi qu'aux changements de luminosité (étape 1);
- une efficacité dans le temps de calcul par la construction de descripteurs compacts et discriminants (étape 2), et par leur indexation dans des arbres pour faciliter leur mise en correspondance (étape 3).

En pratique nous avons fait le choix de n'utiliser dans ce travail que la partie identification des points saillants et calcul du descripteur de la méthode SIFT. Ceci nous permet une certaine maîtrise de la mise en correspondance en conservant le schéma décrit précédemment, et facilite également la comparaison avec la seconde méthode. Les conséquences sont un temps de calcul supérieur dans notre cas, et un nombre de mise en correspondance erronée plus grand qui est corrigé par la suite à l'aide de RANSAC.

Les points saillants de SIFT, également définis comme points clés, sont détectés comme des extremas de la fonction DoG (*Difference of Gaussians*) appliquée à l'image pour différents facteurs d'échelle. Intuitivement, les contours de l'image sont estimés à différentes échelles par filtrage, et les extremas locaux le long de ces contours et entre échelles adjacentes sont sélectionnés. Après suppression des points de faible contraste et de faible courbure le long du contour, un raffinement subpixelique permet une localisation plus précise de ceux-ci. Enfin à chaque point est associée une orientation, déterminée par un calcul de gradient dans un environnement local qui dépend de l'échelle à laquelle celui-ci a été détecté; ceci finalise l'étape 1.

Le descripteur SIFT cherche à caractériser l'orientation des gradients dans une fenêtre (16x16) pixels autour de chaque point clé par un classement ingénieux de celles-ci dans des histogrammes[9]. La robustesse du descripteur est assurée notamment par pondération des orientations estimées par la magnitude du gradient servant à leur calcul ainsi que par leur distance au point saillant, mais également par un calcul d'orientation qui est relatif à l'orientation même du point clé, et aussi par un seuillage des histogrammes pour améliorer l'indépendance aux conditions d'illumination. Le descripteur SIFT est résumé dans un vecteur contenant les valeurs des histogrammes, la fonction de distance h entre deux descripteurs est une distance euclidienne :

$$h(\mathcal{D}_i, \mathcal{D}_k) = \|\mathcal{D}_i - \mathcal{D}_k\|. \quad (7)$$

Ceci complète les étapes 2 et 3 dans le cas de SIFT.

Harris-SSD. Nous désignons par Harris-SSD l'utilisation conjointe de points saillants extraits par la méthode d'Harris [13] et de descripteurs dont la comparaison s'effectue par calcul d'une somme des différences entre éléments au carré (SSD : *Sum Square Differences*).

La détection de points saillants par l'algorithme d'Harris et Stephen s'appuie sur un développement de Taylor de la différence des intensités au carré entre l'image concernée et une version de celle-ci légèrement translatée. Cette expression peut être astucieusement réécrite sous une forme

matricielle localement pour chaque pixel :

$$H_{x,y} = \nabla_{\sigma_H} E(x,y) \nabla_{\sigma_H} E(x,y)^T, \quad (8)$$

où $H_{x,y}$ désigne la matrice de Harris de taille (2x2) au pixel (x,y) de l'image E et $\nabla_{\sigma_H} E(x,y)$ le gradient selon x et y de E lissée par une fenêtre gaussienne de paramètre σ_H fixé à 1 dans notre étude. Une analyse en valeur propre de $H_{x,y}$ permet de déterminer si le pixel appartient à une région uniforme, à un contour, ou encore à un coin. Ce dernier ayant pour avantage de lever toute ambiguïté spatiale, il devient privilégié dans l'extraction des points caractéristiques après seuillage de la fonction d'Harris sur l'image. Pour chaque point saillants, nous ajoutons un calcul d'orientation θ à partir de la valeur locale du gradient estimée pour l'image lissée :

$$\theta(x,y) = \arctan2(\nabla_{\sigma_\theta}^y E(x,y), \nabla_{\sigma_\theta}^x E(x,y)), \quad (9)$$

où le paramètre de lissage σ_θ est fixé à 4. Ainsi chaque point saillant est caractérisé par une position \mathbf{p} et une orientation θ ; ceci complète l'étape 1.

Le descripteur utilisé ici correspond à un patch orienté extrait autour du point d'intérêt dont nous centrons et normalisons les intensités pour limiter sa sensibilité vis à vis des conditions de peau. Soit $I_{\mathbf{p},\theta}$ le patch de taille $(2w+1 \times 2w+1)$ extrait autour du point \mathbf{p} d'orientation θ , nous avons :

$$\mathcal{D} = \frac{I_{\mathbf{p},\theta} - \mu^{I_{\mathbf{p},\theta}}}{\sigma^{I_{\mathbf{p},\theta}}}. \quad (10)$$

Par convention, le patch est extrait par interpolation relativement à θ de sorte à ce que les plus forts gradients soit détectés selon l'axe x de $I_{\mathbf{p},\theta}$, de même nous avons choisi $w = 7$ notre étude. Par extension, \mathcal{D} est donc une matrice et la fonction de distance utilisée à un autre descripteur est :

$$h(\mathcal{D}_i, \mathcal{D}_k) = \sum_{x,y} (\mathcal{D}_i(x,y) - \mathcal{D}_k(x,y))^2. \quad (11)$$

Ici h est également connue comme une fonction d'erreur du type SSD (*Sum of Square Differences*). Ceci finalise l'étape 2 et 3 pour la méthode Harris-SSD; nous adoptons ici une version simplifiée de la méthode MOPS (*Multi-Scale Oriented Patches*) utilisée dans [7] pour la construction de mosaïque d'images.

4 Protocole d'évaluation

Nous détaillons dans cette partie le type des données utilisées pour comparer les méthodes présentées précédemment ainsi que les critères évalués. Les algorithmes ont été implémentés en python et utilisent en grande partie les fonctions de la bibliothèque scikit-image [16].

4.1 Données

Nous avons utilisé la base de données biométrique SDUMLA-HMT en libre accès sur internet [17]. Elle comprend des images acquises par 5 capteurs différents sur 106

personnes. Pour chaque personne, 6 doigts sont enregistrés : le pouce, l'index et le majeur de chacune des mains ; chacun des doigts est acquis 8 fois. Nos expérimentations sont faites sur les données du capteur capacitif FT-2BU³. De cette base, nous avons sélectionné 30 personnes aléatoirement, totalisant 1440 images de taille (152x200) pixels. Il s'agit de données labélisées puisque les acquisitions sont rangées par dossier selon les individus avec un sous dossier par doigt. En revanche, nous n'avons pas de connaissance sur la quantité de translation et de rotation d'un même doigt entre deux acquisitions.

4.2 Simulation d'un petit capteur

Pour simuler le processus d'acquisition d'un petit capteur, nous avons découpé une image de taille 70x70 pixels dans chaque acquisition. Ce format correspond à une surface de (5x5)mm pour un capteur de 350dpi, soit de l'ordre de grandeur de ceux du marché des smartphones.



Figure 3 – Extraction sur une acquisition à gauche (contours rouges), et exemples d'images extraites à droite.

Pour extraire une image centrée sur l'empreinte, nous utilisons un algorithme K-means qui sépare le fond de l'image de celle de l'empreinte. On détermine ensuite le centre comme le point situé au milieu du rectangle englobant l'empreinte segmentée. Enfin, on extrait la fenêtre de taille (70x70) centrée en ce point ; un exemple est en figure 3.

4.3 Séparation des données

Les images sont ensuite séparées en 2 catégories : images d'enrôlement qui correspondraient aux données enregistrées par l'utilisateur sur son smartphone lors de la première utilisation du système biométrique, et images candidates qui représenteraient des enregistrements authentiques ou frauduleux faits par le capteur lors de son utilisation. Pour chaque doigt, on sélectionne aléatoirement un nombre n d'acquisitions que l'on utilise comme données d'enrôlement, le reste devenant les données candidates.

4.4 Critères évalués

Une table de score est calculée en comparant chaque image candidate avec l'ensemble des images d'enrôlement de chaque doigt, nous obtenons un score par couple {image candidate, doigt enrôlé} qui est fonction des scores individuels de l'image candidate avec chacune des images d'enrôlement du doigt ; dans les résultats qui suivent nous uti-

3. capteur de Miaxis Biometrics qui utilise une technologie de Fingerprint Cards, a priori un clone du capteur FPC10xx à 363dpi

lisons le maximum de ceux-ci. La table de score permet de calculer les taux de FRR et FAR pour différents seuils. Les trois algorithmes sont comparés sur la base des courbes ROC obtenues sur les données extraites.

5 Résultats

La figure 4 résume les performances en reconnaissance des trois algorithmes en fonction du nombre d'images retenues au cours de l'enrôlement ; plus la courbe ROC s'approche du point $(0, 10^{-3})$, meilleur est l'algorithme.

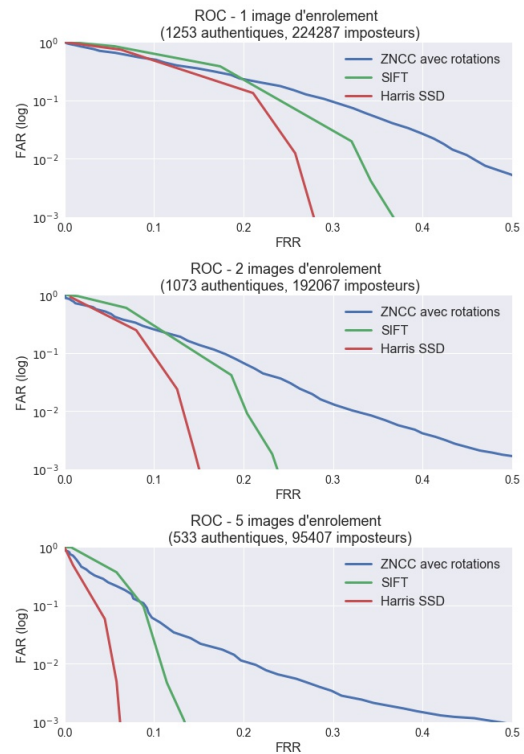


Figure 4 – Courbes ROC pour les trois algorithmes selon le nombre d'images d'enrôlement.

On constate assez logiquement qu'une augmentation du nombre d'images enrôlées permet de réduire nettement le FRR à FAR fixe, le FRR est presque divisé par trois pour les méthodes SIFT et Harris SSD en passant d'une à cinq images d'enrôlement. Il faut noter que ce nombre est bien supérieur à dix pendant la procédure d'enregistrement du doigt réalisée par un smartphone. La quantité des données de la base ne nous permet pas de reproduire un tel enrôlement, mais il est clairement déterminant pour augmenter les performances d'un algorithme d'authentification.

On remarque également que la méthode Harris SSD se démarque de SIFT, ceci contredit certains résultats de la littérature dont ceux de [5]. Une première explication est que SIFT est un algorithme performant dans des tâches génériques avec de nombreux degrés de liberté comme l'invariance au facteur d'échelle. Celle-ci n'est pas requise dans notre étude, et n'est d'ailleurs pas prise en compte dans Harris SSD. Ensuite, SIFT par conception cherche des ex-

tréma le long des lignes appartenant à l'image, or ce sont des zones à forte ambiguïté dans le cas des empreintes digitales. La figure 5 vient renforcer cette observation en comparant les points saillants détectés par les deux méthodes sur une même acquisition. On remarque que le détecteur d'Harris se concentre exclusivement sur les contours là où SIFT détecte des fonds de vallées ou des sommets de crêtes qui ont a priori moins de chances d'être discriminants.

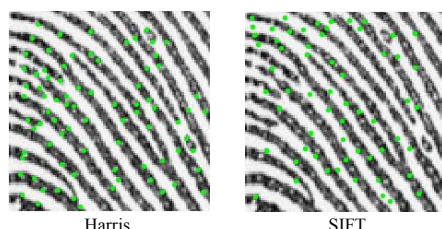


Figure 5 – Points saillants de Harris et SIFT en verts.

La seconde explication est liée à la nature des données. Les empreintes sont constituées d'une texture très répétitive avec une alternance de vallées et de crêtes. Un descripteur qui ne s'appuie que sur des histogrammes d'orientation des gradients peut perdre des informations locales sur la forme de l'empreinte autour du point saillant. La figure 6 montre des exemples de patches extraits par Harris SSD qui semblent difficiles à résumer à l'aide du descripteur SIFT.

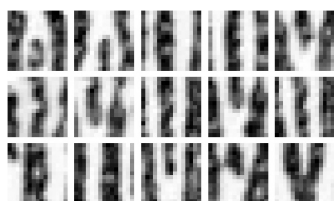


Figure 6 – Exemples de patches (15x15) pixels extraits autour des points saillants de Harris en figure 5.

6 Perspectives

Les résultats obtenus sont préliminaires, la robustesse des algorithmes doit être testée sur plus de données avec des variations dans les conditions de peau plus importantes. Certains paramètres ont été fixés empiriquement par observation des résultats, comme la taille des fenêtres ou les coefficients de lissage pour Harris SSD, ils nécessiteraient une analyse de sensibilité plus poussée.

Il reste des marges de manoeuvre importantes pour accélérer l'appariement et probablement se passer de l'étape de RANSAC. Enfin la similarité entre les patches retenus par la méthode d'Harris SSD sur l'ensemble des données nous suggère que des algorithmes d'apprentissage d'une base de représentation de ceux-ci pourraient ouvrir d'autres perspectives dans la façon de calculer le score de similarité.

Références

[1] J.F. Mainguet, W. Gong, et A. Wang. Reducing silicon fingerprint sensor area. Dans *Internatio-*

nal Conference on Biometric Authentication, volume 3072, pages 301–308, Hong-Kong, 2004.

- [2] R. Sanchez-Reillo. The Madrid Report, *Next Biometrics*, june 2017.
- [3] C. Lee et S. Wang. Fingerprint feature extraction using gabor filters. *Electronics Letters*, 35(4) :288–290, Feb 1999.
- [4] A. Jain, S. Prabhakar, L. Hong, et S. Pankanti. Filterbank-based fingerprint matching. *Transactions on Image Processing*, 9(5) :846–859, 2000.
- [5] A. Karlsson et J. Hagel. Fingerprint matching-hard cases, *Faculty of engineering, Lund University*, 2016.
- [6] A. Jain, A. Ross, et S. Prabhakar. An introduction to biometric recognition. *Transactions on Circuits and Systems for Video Technology*, 14(1) :4–20, Jan 2004.
- [7] M. Brown, R. Szeliski, et S. Winder. Multi-image matching using multi-scale oriented patches. Dans *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 510–517, Jun 2005.
- [8] R. Szeliski. *Computer vision : algorithms and applications*. Springer Science and Business Media, 2010.
- [9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.
- [10] K. Mikolajczyk et C. Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005.
- [11] J. P. Lewis. Fast normalized cross-correlation. *Vision interface*, 10(1) :120–123, 1995.
- [12] S. Chambon et A. Crouzil. Mesures de corrélation robustes aux occultations. Dans *ORASIS-Journées jeunes chercheurs en vision par ordinateur*, Gerardmer, 2003.
- [13] C. Harris et M. Stephens. A combined corner and edge detector. Dans *Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [14] S. Belongie, J. Malik, et J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence*, 24(4) :509–522, 2002.
- [15] M. Fischler et Ro. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, Juin 1981.
- [16] S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, et the scikit-image contributors. scikit-image : image processing in Python. *PeerJ*, 2 :e453, 6 2014.
- [17] Y. Yin, L. Liu, et X. Sun. Sdumla-hmt : A multi-modal biometric database. Dans *Chinese Conference on Biometric Recognition*, pages 260–268. Springer-Verlag, 2011.