



EDITE - ED 130



2015-ENST-00xx

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Antoine DEBLONDE

Algorithmes rapides et fiables de recherche dans une base d'images d'empreintes digitales

Directeur de thèse : **Andrés ALMANSA**

Co-encadrement de la thèse : **Saïd LADJAL, Cédric BELIARD**

Jury

Prof. Bernardette DORIZZI, INT - Telecom SudParis

Dr. Christophe CHARRIER, GREYC Univ Caen

Prof. Amine NAIT-ALI, Université Paris-Est Créteil

Prof. Florence TUPIN, Telecom ParisTech

Dr. Andrés ALMANSA, Telecom ParisTech

Dr. Saïd LADJAL, Telecom ParisTech

Dr. Cédric BELIARD, Morpho

Rapporteur

Rapporteur

Rapporteur

Examineur

Directeur de thèse

Encadrant

Encadrant

TELECOM ParisTech

école de l'Institut Mines-Télécom - membre de ParisTech

46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

pour Aurore, Christian et Jean, partis beaucoup trop tôt

Remerciements

Je souhaite remercier d'abord les membres de mon jury et mes relecteurs pour le temps et l'attention qu'ils m'ont accordé : Bernadette Dorizzi, Christophe Charrier, Amine Nait-Ali, je remercie en particulier Florence Tupin pour ses conseils et son écoute et pour avoir présidé mon jury, après avoir déjà assisté à ma soutenance de mi-parcours avec Agnès Desolneux. Je remercie également Jean-Christophe Fondeur pour avoir représenté Morpho à ma soutenance.

Je remercie aussi chaleureusement mes directeurs de thèse, Andrés Almansa et Saïd Ladjal, qui ont permis à cette thèse d'aboutir après maints aléas de parcours.

J'en profite pour remercier également tous ceux qui ont pu un moment ou un autre contribuer à cette thèse, en particulier mes collègues de l'URT. Un grand merci à l'équipe empreinte, à Cécile, Milena, Emilie, Fantin (courage d'ailleurs pour la fin de la tienne !), Jonathan, Jean-Noël. Merci notamment à Mélanie pour les articles, à Benoît pour les relectures, à Eric pour les conseils. Merci à Lauriane, Marie, Constance, Catherine, Emine, Vincent, Julien, Xavier, Kaz, et tous les autres.

Merci également à Raphaël (tu m'as bluffé le jour de la soutenance), à mes frères, Jean-Philippe et Arnaud, qui y sont passés aussi et ont su me conseiller, à mes parents, à Anne-Marie, et enfin et surtout à Anne-Flore, qui m'a supporté et soutenu pendant tout ce temps, et sans qui je ne serais pas arrivé au bout !

Table des matières

Introduction générale	9
1 Introduction aux algorithmes de reconnaissance d’empreintes digitales	13
1.1 Généralités	13
1.2 Systèmes de reconnaissance biométriques par empreintes digitales	15
1.3 Comparaison automatique d’empreintes digitales	16
1.3.1 Analyse des performances en vérification	17
1.3.2 Analyse des performances en identification	19
1.3.3 Algorithmes de filtrage	21
1.4 Problématique	23
2 Recherche d’images par le contenu : Application à l’empreinte digitale	25
2.1 Généralités	25
2.2 Les types de représentation d’image	28
2.2.1 Approche globale	28
2.2.2 Approche locale	28
2.3 Agrégation des descripteurs locaux	30
2.3.1 Agrégation directe	30
2.3.2 Consolidation de l’agrégation directe : transformée de Hough généralisée	35
2.3.3 Représentation par graphes	46
2.3.4 Bag of Words	47
2.4 Accélération de la recherche	50
2.4.1 Application de la clusterisation aux représentations d’images	50
2.4.2 Outils de clusterisation et d’accélération de la recherche	52
2.5 Application à un algorithme de filtrage d’empreintes digitales	55
2.5.1 Distance agrégée	56
2.5.2 Score agrégé	57
2.5.3 Score a contrario	58
3 Comparaison d’empreintes digitales : construction d’un algorithme en force brute	61
3.1 Extraction des données	61
3.1.1 Crêtes et minuties	62
3.1.2 Orientation locale de crête	63

3.1.3	Singularités du flux de crêtes	66
3.2	Descripteurs globaux d’empreintes digitales	66
3.2.1	Corrélation d’images	66
3.2.2	Comparaison de cartes d’orientation	68
3.2.3	Analyse des approches globales	70
3.3	Descripteurs locaux d’empreintes digitales	70
3.3.1	Types de descripteurs locaux	70
3.3.2	Méthodes de comparaison	74
3.3.3	Analyse des approches locales	77
3.3.4	Bases de données	77
3.4	Descripteurs choisis	78
3.4.1	Minutiae Cylinder Code	78
3.4.2	Triplets de minuties	82
3.4.3	Patch d’orientation locale	84
3.4.4	Patch d’orientation locale - décomposition parcimonieuse	86
3.4.5	Algorithmes de comparaison	89
3.5	Evaluation et analyse	93
3.5.1	Analyse locale	94
3.5.2	Distance agrégée	102
3.5.3	Effets de la correction de distance	104
3.5.4	Score agrégé	107
3.5.5	Score a contrario	110
3.5.6	Effets des transformées de Hough	114
3.5.7	Score en proportion de Hough	117
3.5.8	Comparaison avec l’état de l’art	119
3.5.9	Conclusion sur les méthodes en force brute	121
4	Accélération	123
4.1	Généralités sur les méthodes de recherche de plus proches voisins	123
4.1.1	Méthodes de recherche accélérée	126
4.1.2	Comparaison des coûts et contraintes théoriques	127
4.2	Application aux descripteurs d’empreinte digitale choisis	129
4.2.1	Estimation du nombre de plus proches voisins optimal	129
4.2.2	Comparaison avec l’état de l’art	134
4.2.3	Grands MCC	137
4.2.4	Petits MCC	139
4.2.5	Patchs de carte d’orientation	141
4.2.6	Triplets de minuties	142
4.2.7	Comparaison générale	143
4.3	Bag of Words	146
4.3.1	Dictionnaire inversé sur les grands MCC	148
4.3.2	Dictionnaire inversé sur les patchs de carte d’orientation	149
4.3.3	Dictionnaire inversé sur les triplets de minuties	150

4.4	Fusion des données	152
4.4.1	Décorrélacion des descripteurs utilisés	153
4.5	Fusion des LSH MCC et coefficients parcimonieux	155
4.6	Fusion des distances BoW	158
4.7	Application aux requêtes latentes	160
4.8	Conclusion	163
	Conclusion	165
	Annexes	175
	Bibliographie	183

Introduction générale

Depuis quelques décennies, l'accroissement du nombre d'appareils capables de capter des images, appareils photos ou caméras de tout type, est allé de pair avec l'accroissement des capacités de stockage de données numériques. La taille des bases de données d'images s'est ainsi accrue considérablement, et les besoins pour des algorithmes de recherche dans de telles bases, qui soient à la fois précis en termes de reconnaissance et efficace en termes de temps de calcul, se sont fortement développés. Google Image apparaît aujourd'hui comme l'exemple le plus typique de l'utilisation de tels algorithmes, mais les besoins se sont également fait sentir dans l'imagerie médicale, la vidéosurveillance, la classification d'œuvres d'art, etc.

Les algorithmes de recherche d'images dans des grandes bases de données doivent concilier deux objectifs généralement contradictoire : la *précision* de la recherche, c'est à dire la capacité de l'algorithme à trouver au sein de la base de données des images ressemblants à une requête avec suffisamment de pertinence, avec le *temps de calcul*, c'est à dire la capacité à obtenir des résultats pertinents en un temps raisonnable, y compris dans de très grandes bases de données.

L'utilisation des points d'intérêt locaux en reconnaissance d'image est un concept qui remonte à plus de trente ans, aux travaux fondateurs de Hans Moravec. Les travaux de Harris et surtout de Lowe ont progressivement imposé les méthodes à base de points d'intérêts et de descripteurs locaux comme les plus performantes en coûts de calcul et les plus efficaces en qualité de comparaison, robustes à tout type et toute qualité d'image.

Les empreintes digitales constituent un type très particulier d'images, utilisées pour la reconnaissance d'individus. Les algorithmes académiques de recherche d'images appliquées à l'empreinte digitale ont jusqu'à récemment surtout insisté sur l'obtention de bonnes performances en précision. Avec la mise à disposition de bases de données publiques d'empreintes digitales de taille croissante, et la demande croissante pour des outils de recherche d'empreintes digitales dans des bases de données privées ou policières de très grande taille, la recherche en empreinte digitale s'est orientée depuis quelques années vers l'obtention de bonnes performances en précision à des coûts plus faibles en temps de calcul.

Les caractéristiques visuelles des empreintes digitales facilitent naturellement la construction de points d'intérêts locaux, pourtant, le transfert systématique des méthodes et des algorithmes développés en recherche d'image par le contenu (*CBIR : Content-Based Image Retrie-*

val) avec des images génériques est encore très peu développé.

L'objectif de cette thèse est le développement d'un algorithme de recherche dans des bases de données d'images d'empreintes digitales, qui soit non seulement précis, mais également robuste à des requêtes de mauvaise qualité, bruitées ou occultées, et qui soit capable de gérer des bases de données de très grande taille en un temps raisonnable, donc avec un temps de calcul le plus faible possible.

Avec de telles contraintes, nous avons naturellement orienté nos travaux vers l'adaptation à l'empreinte digitale des méthodes de recherche d'image à base de descripteurs locaux, méthodes intrinsèquement rapides et robustes dans le cas d'images génériques.

Contributions

Dans nos travaux, nous avons procédé à l'analyse systématique d'un ensemble de descripteurs locaux adaptés à l'empreinte digitale. A l'exception du premier de ces descripteurs, le Minutiae Cylinder Code, directement issu de l'état de l'art, ces différents descripteurs sont une implémentation nouvelle de concepts existants. Cette nouvelle implémentation est essentiellement destinée à disposer de descripteurs faciles à manipuler dans le cadre de l'application de méthodes de recherche rapides d'images génériques. L'un de ces descripteurs, la représentation parcimonieuse de patches de carte d'orientation, est original et vient d'une réinterprétation d'un descripteur plus ancien à la lumière des nouvelles avancées dans les méthodes de décomposition parcimonieuse.

Suite à l'analyse de ces différents descripteurs, nous avons pu proposer l'application d'un ensemble de méthodes de recherche accélérée, tirés de l'état de l'art en recherche d'images par le contenu, aux empreintes digitales. Si plusieurs méthodes d'indexation et de recherche accélérée ont été proposées récemment dans l'état de l'art des travaux en empreinte digitale, la mise en perspective systématique des méthodes développées pour les images génériques à des descripteurs construits spécifiquement pour l'empreinte digitale constitue une nouveauté.

Les résultats obtenus en terme de temps de calcul constituent une avancée par rapport à l'état de l'art, et s'expliquent justement par le fait que les résultats théoriques envisageables avec les méthodes génériques dépassaient les performances que l'on peut attendre de la majorité des algorithmes actuels en empreinte digitale.

Organisation du mémoire

Dans le chapitre 1, nous proposons une introduction générale aux images d'empreintes digitales et à leurs particularités. Nous présentons les scénarii typiques dans lesquels elles sont utilisées, et quel type de scénario est concerné par nos travaux, puis nous détaillons les modalités d'évaluation des performances d'un algorithme de reconnaissance d'empreintes digitales, afin de poser le cadre de notre problématique.

Dans le chapitre 2, nous passons en revue les différentes méthodes de reconnaissance d'images

génériques de l'état de l'art, et comment ces méthodes se prêtent à l'accélération de la recherche par indexation des bases de données. Après analyse des différentes méthodes possibles, nous sélectionnons celles qui se prêtent le mieux à notre problématique, et nous proposons l'ensemble de la procédure de construction et d'optimisation d'un algorithme de recherche accélérée d'images digitales, appliquée au cas particulier de bases de données d'empreintes digitales avec requêtes potentiellement de mauvaise qualité.

Dans le chapitre 3, nous présentons les différentes méthodes de l'état de l'art spécifiquement en reconnaissance d'empreinte digitale, en reprenant la grille d'analyse des méthodes des images génériques. Cette analyse nous permet de sélectionner parmi les algorithmes et méthodes existants ceux qui s'insèrent au mieux dans notre approche utilisant des points caractéristiques et des descripteurs locaux.

Une fois choisis les descripteurs locaux qui nous semblent les plus adaptés à notre problématique, nous procédons à une évaluation approfondie des performances en précision de la reconnaissance qui peuvent être atteintes, par évaluation sur une base de données de petite taille. Cette évaluation nous permet de déterminer les avantages et inconvénients des différents descripteurs, ainsi que leur adaptabilité à l'accélération de la procédure de recherche.

Dans le chapitre 4, nous présentons les méthodes de l'état de l'art en recherche accélérée dans des bases de données d'images de très grande taille. Nous avons procédé à l'adaptation de ces méthodes, développées et éprouvées sur des images génériques, de type photographies par exemple, au cas particulier que constitue les empreintes digitales. Les méthodes de recherche accélérée ainsi exposées et sélectionnées sont alors appliquées sur la base de données de petite taille utilisée pour l'apprentissage des algorithmes, puis sur des bases de données de grande taille et des bases de données d'images de mauvaise qualité.

Notations

Dans la suite du mémoire, sauf mentions contraires, les notations suivantes seront utilisées

Terme	signification
n_s	nombre de minuties dans l'image requête
n_r	nombre de minuties dans l'image référence
n_I	nombre de descripteurs locaux dans l'image I
$z, \{z_i\}$	descripteur local, ensemble des descripteurs locaux
$z_{k,i}$	descripteur local i de l'image k
E	espace des descripteurs locaux
$p, \{p_i\}$	coefficient de pondération du descripteur local, ensemble des coefficients de pondération des descripteurs locaux $\{z_i\}$
$s, \{s_{ij}\}$	coefficient de pondération de l'appariement de descripteurs locaux z_i, z_j
\mathbb{T}_{SR}	similitude de superposition optimale de l'image S vers l'image R
FA	<i>False Acceptance</i> , fausse acceptance
FAR	<i>False Acceptance Rate</i> , taux de fausse acceptance
FR	<i>False Reject</i> , faux rejets
FRR	<i>False Reject Rate</i> , taux de faux rejets
ROC	<i>Receiver Operating Curve</i> , courbe $FRR = F(FAR)$
Acc	<i>Accuracy Rate</i> , taux d'acceptance
CMC	<i>Cumulative Match Characteristic</i> , courbe $Acc = F(rang)$

Chapitre 1

Introduction aux algorithmes de reconnaissance d'empreintes digitales

1.1 Généralités

Les caractéristiques biologiques permettant d'identifier et de distinguer les êtres humains entre eux sont couramment regroupées sous le terme de *biométries*. Parmi ces caractéristiques, on trouve notamment les traits du visage, l'iris, les empreintes digitales.

Utilisée en criminalistique depuis plus d'un siècle, avec les travaux fondateurs de Francis Galton [1] et Edward Henry[2], et dans les systèmes civils et policiers de reconnaissance d'individus depuis quelques décennies, l'empreinte digitale constitue l'une des biométries les plus étudiées. Le succès de l'empreinte digitale comme biométrie repose essentiellement sur deux faits : d'abord, elles sont en pratique uniques pour chaque individu [1] [2] [3], ensuite, elles sont relativement faciles à acquérir,

Une empreinte digitale est l'image de l'épiderme d'une extrémité d'un doigt, acquise généralement sur une surface plane comme une feuille de papier ou un scanner, mais aussi sur des surfaces quelconques plus accidentées, notamment sur les scènes de crime, ou encore acquise directement dans l'espace par photo ou caméra [5]. L'épiderme des extrémités des doigts est caractérisée par un motif complexe de relief, composé de crêtes et de vallées. Ce motif d'épiderme se forme chez le fœtus vers le septième mois de grossesse. Par rapport à d'autres biométries comme les traits du visage, l'empreinte digitale présente l'avantage de très peu varier durant l'existence d'un individu, et même après le décès, à l'exception de dégradations majeures de la peau dues aux cicatrices ou aux maladies.

Les experts en biométrie distinguent plusieurs types d'empreintes digitales en fonction de leurs conditions d'acquisition. Les types les plus couramment répertoriés sont les suivants :

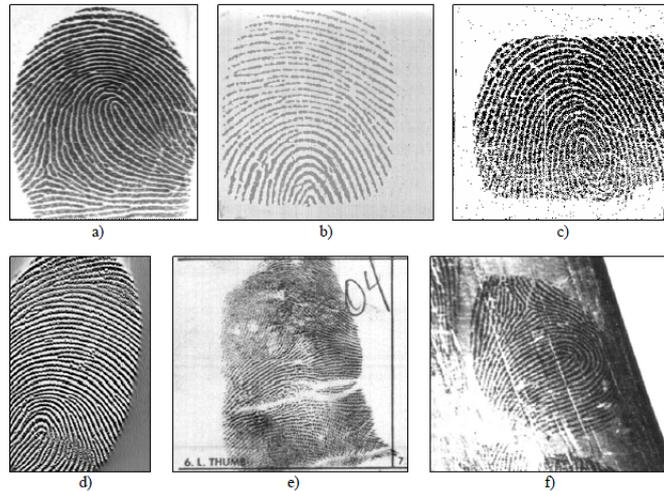


FIGURE 1.1 – Images d’empreintes digitales extraites a) d’un scanner optique FTIR b) d’un scanner capacitif c) d’un scanner piézoélectrique d) d’un scanner thermique e) d’une impression à l’encre sur papier f) d’une latente de scène de crime (source [4])

- les acquisitions de type *enroulé* (*rolled*), où un individu fait rouler son doigt sur une surface d’acquisition, de manière à récupérer un maximum de surface de doigt.
- les acquisitions de type *posé* (*slap*), où un individu pose simplement l’extrémité de son doigt sur une surface d’acquisition.
- les acquisitions de type *latente*, qui sont des traces de doigts collectées sur des objets quelconques, généralement par des experts policiers sur des scènes de crime

Les caractéristiques d’une empreinte digitale sont classées sur trois niveaux.

Au niveau 1, on trouve les crêtes et vallées. Une crête est un segment continu de relief de peau ascendant, et les crêtes sont séparées par des vallées, reliefs de peau descendant. La largeur des crêtes est couramment situées entre $100\mu m$ et $300\mu m$, avec une période entre crêtes de l’ordre de $500\mu m$. La forme générale du flux de crêtes des empreintes est utilisée pour définir les *classes de Henry*, illustrées par la figure 1.2 : boucles (*loop*) orientées à gauche ou à droite, arches, et spirales (*whorl*) [2]. Le flux de crêtes présente également des singularités, nommées *cores* et *deltas*, correspondant respectivement à des points singuliers de type foyer et de type col. Le core peut aussi être défini empiriquement comme le point le plus haut de la boucle de crête la plus intérieure de l’empreinte digitale. Ces singularités sont illustrées en figure 1.3.



FIGURE 1.2 – Les cinq classes de Henry, avec de gauche à droite : boucle à gauche, boucle à droite, spirale, arche, arche pincée (*tented arch*) ; (tiré de[4])

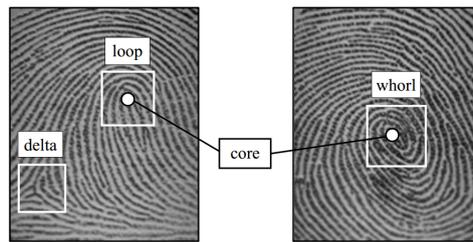


FIGURE 1.3 – Zone de singularités (boîtes blanches) et core (cercles) (tiré de[4])

Au niveau 2, on considère un autre type de caractéristique : les terminaisons et intersections des crêtes. Ces objets sont ponctuels, orientés par leur crête d'appartenance [6] [7], et sont appelés *minuties*

Au niveau 3, on trouve les détails les plus précis des empreintes digitales : déformations fines et largeur des crêtes, micro-crêtes incises dans des crêtes plus larges, pores de sudation sur les crêtes. Ces détails peuvent être détectés sur une empreinte digitale quand la résolution de l'image est suffisamment élevée, de l'ordre de 1000 dpi [5].

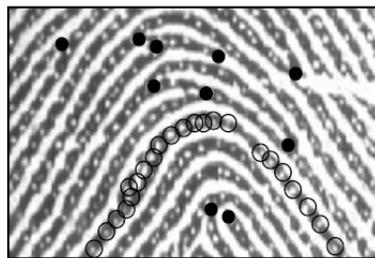


FIGURE 1.4 – Minuties (cercles pleins) sur une portion d'empreinte digitale ; pores de sudation (cercles vides) sur une crête (tiré de [4])

1.2 Systèmes de reconnaissance biométriques par empreintes digitales

Depuis les débuts de l'étude scientifique des empreintes digitales, les systèmes de reconnaissance fondés sur ces éléments biométriques ont essentiellement été utilisés pour deux types de procédures : la *vérification*, aussi appelée *authentification*, et l'*identification*.

Dans une procédure de vérification, le système biométrique est destiné à vérifier si l'identité d'un individu caractérisée par ses empreintes digitales correspond effectivement à l'identité qu'il revendique. Cela signifie en pratique que le système vérifie si les empreintes digitales de l'individu considéré acquises au moment de la vérification correspondent aux images d'empreintes digitales préalablement acquises et stockées dans une base de données de référence. Une utilisation type de procédure de vérification est le contrôle de douane, quand les agents des douanes doivent vérifier si l'identité revendiquée par un voyageur sur son passeport correspond à son identité réelle, définie par ses empreintes digitales.

Dans une procédure d'identification, le système biométrique est destiné à déterminer si un individu, défini par une ou plusieurs de ses empreintes digitales, appartient ou non à une base de données de référence d'individus définis également par des acquisitions de leurs empreintes digitales, et si c'est le cas, à identifier cet individu, en fournissant une liste de candidats potentiels susceptibles de correspondre à l'individu requête. Parmi les applications types de la procédure d'identification, on peut citer :

- les accès sécurisés d'administrations ou d'entreprises, où on l'on cherche à vérifier si un individu appartient à une liste de personnes autorisées
- l'enregistrement des procédures administratives effectuées par un citoyen (enregistrement d'un vote, demande de carte d'identité, etc)
- les enquêtes criminelles, où l'on cherche à vérifier si les empreintes digitales d'un individu relevées sur une scène de crime appartiennent à un criminel déjà identifié

1.3 Comparaison automatique d'empreintes digitales

Historiquement, les procédures de vérification étaient systématiquement réalisées par des experts humains qui estimaient le degré de similarité entre deux empreintes digitales et décidaient si elles correspondaient ou non. La comparaison d'empreintes digitales par des experts humains est toujours pratiquée en vérification, mais il s'agit d'une procédure intrinsèquement lente. Pour des procédures d'identification, l'expertise humaine est envisageable sur des très petites bases de données, mais devient complètement impraticable pour des bases de données de grande taille (les bases de données du FBI atteignent plusieurs millions d'empreintes digitales [4]).

Pour accélérer les procédures de vérification, des algorithmes de comparaison d'empreintes digitales, capables de fournir automatiquement un score de comparaison reflétant la probabilité de correspondance entre deux empreintes digitales, ont été développés. Une fois ce score calculé, à charge alors à un algorithme de décision ou à un opérateur humain, de décider si les probabilités ainsi estimées permettent d'identifier l'individu dans la base de références.

Dans la suite de notre exposé, on désignera l'empreinte d'un individu soumis à vérification ou identification sous le terme de *requête* (couramment désignées *search* ou *query* en anglais). On utilisera le terme *référence* (*reference*) pour désigner soit l'empreinte dont l'identité est revendiquée dans le cas d'une procédure de vérification, soit les empreintes de la base de données à explorer dans le cas d'une procédure d'identification.

Lorsque l'on compare deux empreintes digitales, deux situations sont envisageables :

- Soit les deux empreintes sont deux acquisitions d'un même doigt d'une même personne, auquel cas on dira que les empreintes sont *correspondantes* ou *matchantes*. On désignera cette hypothèse comme H_M
- Soit les deux empreintes ne sont pas issues d'un même doigt, auquel cas on dira que les empreintes sont *non correspondantes* ou *non matchantes*. On désignera cette hypothèse comme H_{NM}

Pour un couple requête/référence S, R (*search/query*), on note D_M la décision que S et R sont matchantes, et D_{NM} la décision qu'elles sont non matchantes. On notera également $s(S, R)$ le score de similarité entre une requête et une référence, que l'on prend à valeurs dans \mathbb{R}^+ .

Deux types d'erreurs de décision peuvent alors être faites :

- les *fausses acceptances* (D_M est prise alors que H_{NM} est vraie)
- les *faux rejets* (D_{NM} est prise alors que H_M est vraie)

1.3.1 Analyse des performances en vérification

En vérification, on considère un seul couple requête/référence S, R , et donc un seul événement de type H_M ou H_{NM}

Le taux de fausse acceptation (*False Acceptance Rate* - FAR) désigne la probabilité de fausse acceptation :

$$FAR = Pr(D_M | H_{NM})$$

Le taux de faux rejets (*False Reject Rate* - FRR) désigne la probabilité de faux rejets :

$$FRR = Pr(D_{NM} | H_M)$$

On parlera également régulièrement du taux d'acceptation (ou *accuracy*) Acc , défini par

$$Acc = Pr(D_M | H_M) = 1 - FRR$$

L'évaluation d'un algorithme de comparaison d'empreintes digitales peut se faire selon différents critères.

La prise de décision D_M ou D_{NM} est faite selon un seuil de score de similitude δ : D_M est prise si $s(S, R) > \delta$, D_{NM} est prise sinon.

On a alors $FAR = P(s > \delta | H_{NM})$ et $FRR = P(s \leq \delta | H_M)$

En collectant un grand nombre de scores de similarités générés sur des paires d'empreintes digitales matchantes et non matchantes, on peut estimer les fonctions de distribution de $p(s | H_M)$ et $p(s | H_{NM})$

On a alors

$$FRR = \int_0^{\delta} p(s | H_M) ds$$

et

$$FAR = \int_{\delta}^{+\infty} p(s | H_{NM}) ds$$

Le choix de δ constitue un arbitrage entre les fausses acceptations et les faux rejets : une valeur faible réduira le taux de faux rejets mais augmentera le taux de fausses acceptations, et inversement, FAR et FRR étant des fonctions de δ . La visualisation de cet arbitrage se fait au moyen d'une courbe ROC (*Receiver Operating Characteristic*), ou d'une courbe DET (*Detection-Error Tradeoff*).

La courbe ROC est définie comme $FAR(\delta)$ fonction de $Acc(\delta)$, la courbe DET comme $FAR(\delta)$ fonction de $FRR(\delta)$. En pratique, au plus l'algorithme est performant, au plus il parvient à séparer les distributions de $p(s | H_M)$ et $p(s | H_{NM})$, au plus la courbe DET se rapproche de l'axe des abscisses et de l'axe des ordonnées.

L'effet du choix d'un seuil de décision sur le score, ainsi que les deux taux d'erreurs peuvent être visualisés via les distributions de distances entre empreintes matchantes et non matchantes, sur les figures 1.5, 1.6, 1.7, 1.8.

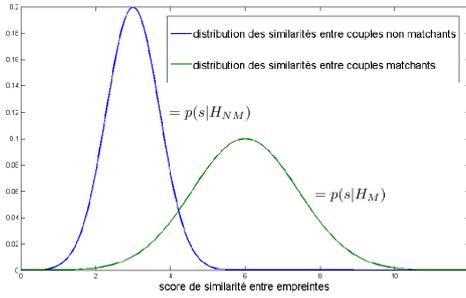


FIGURE 1.5 – Exemple de distributions de scores de similarité entre couples matchants (vert) et non matchants (bleu) d’empreintes digitales - en abscisse le score de similarité, en ordonnée la densité de probabilité

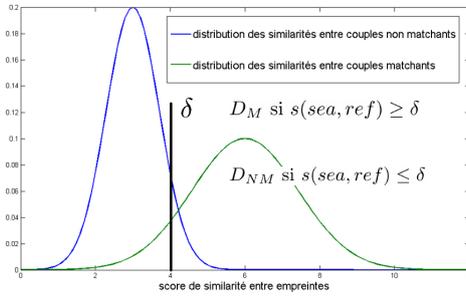


FIGURE 1.6 – Sur les courbes de la figure précédente, illustration de l’effet d’un seuil de décision sur les scores de similarité

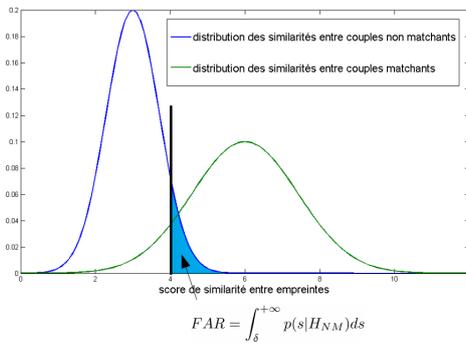


FIGURE 1.7 – Définition graphique du taux de fausses acceptances

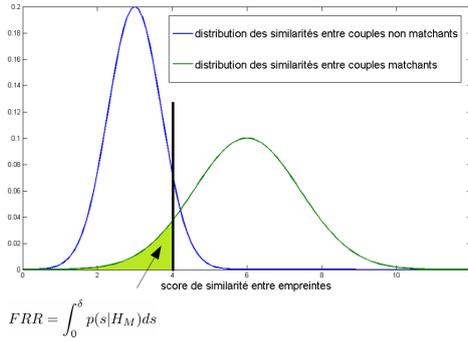


FIGURE 1.8 – Définition graphique du taux de faux rejets

1.3.2 Analyse des performances en identification

En identification, on compare une requête S avec N références (R_1, \dots, R_N) , N événements de type H_M ou H_{NM} sont donc générés. Pour tout i entre 1 et N on note alors respectivement H_{M^i} et H_{NM^i} les hypothèses (S, R_i) vérifient H_M et (S, R_i) vérifient H_{NM} , ainsi que respectivement D_{M^i} et D_{NM^i} les décisions (S, R_i) sont matchantes et (S, R_i) sont non matchantes.

Les algorithmes d'identification pouvant fournir une liste, de taille éventuellement variable, de références potentiellement correspondantes, on définit les deux types d'erreur suivants :

- la requête appartient à la base de références, et la liste de candidats ne contient pas la référence matchante. On parle alors de faux rejets en identification ou de faux négatifs en identification (*False Negative Identification error*)
- la requête n'appartient pas à la base de références, et la liste de candidats est non vide. On parle alors de fausses acceptances en identification ou de faux positifs en identification (*False Positive Identification error*)

On définit la probabilité de faux rejets en identification

$$FNIR = Pr(\forall i D_{NM^i} | \exists i / H_{M^i})$$

et la probabilité de fausse acceptance en identification

$$FPIR = Pr(\exists i / D_{M^i} | \forall i H_{NM^i})$$

En supposant qu'il n'existe qu'une seule empreinte digitale par doigt dans la base de référence, on a alors $FNIR = FRR$ et $FPIR = 1 - (1 - FAR)^N$. Si le FAR est petit, le FPIR peut être simplement approché par $FPIR \cong N \cdot FAR$, ce qui signifie que le nombre de fausses acceptances croît linéairement avec la taille de la base de données.

Les performances d'un algorithme de comparaison en identification peuvent également être visualisées en courbes ROC ou DET, en remplaçant FAR par FNIR et Acc par 1-FPIR.

Il est également possible d'analyser des performances en identification en visualisant, pour des requêtes ayant toutes une référence matchante dans la base de données, le rang, dans la

liste triée des candidats, où se situe la référence matchante. Le graphe présentant le taux d'acceptance $1-FNIR$ en fonction du rang dans la liste de candidats est appelé *Cumulative Match Characteristic (CMC)*.

La figure 1.9 propose un exemple de distribution des similarités $s(S, R)$ pour des couples d'images vérifiant respectivement H_M et H_{NM} . Ces distributions permettent de construire les courbes ROC et CMC présentées respectivement dans les figures 1.10 et 1.11. LA construction de la courbe ROC est illustrée dans la figure 1.12

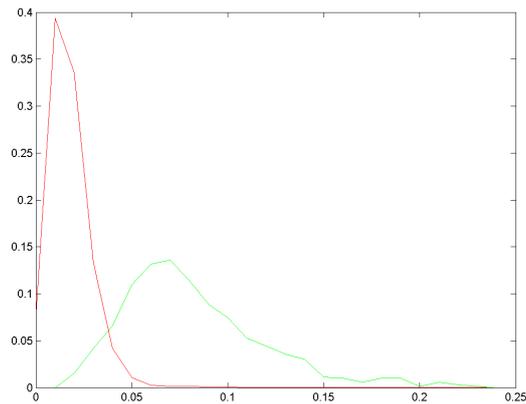


FIGURE 1.9 – Exemple de distributions de scores de similarité entre couples matchants (vert) et non matchants (rouge) d’empreintes digitales - en abscisse le score de similarité, en ordonnée la densité de probabilité

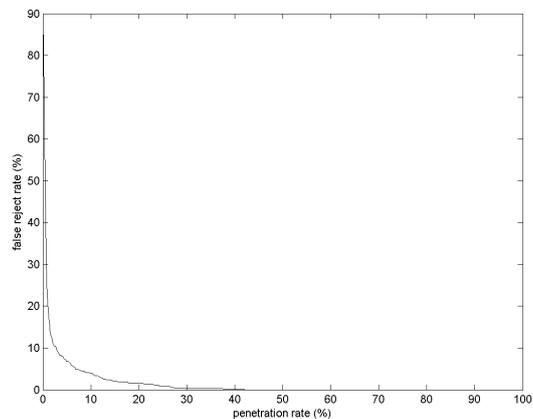


FIGURE 1.10 – Exemple de courbe ROC associée aux distributions de la figure 1.9 - en abscisse le taux de fausses acceptances, en ordonnée le taux de faux rejets

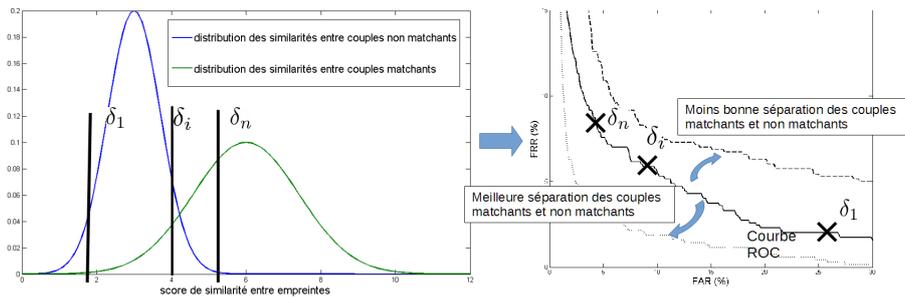


FIGURE 1.12 – Construction de la courbe ROC en utilisant les taux de FA et FR atteints pour différentes valeurs de seuils de similarité

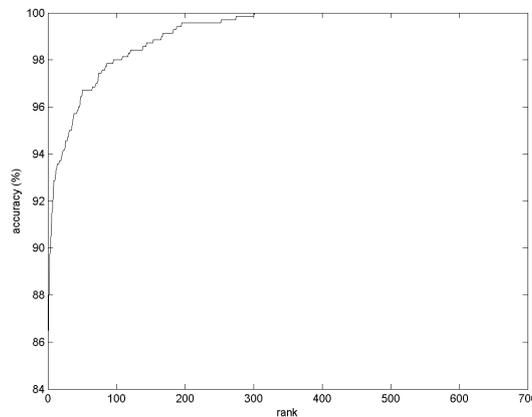


FIGURE 1.11 – Exemple de courbe CMC associée aux distributions de la figure 1.9

1.3.3 Algorithmes de filtrage

La taille des bases de données à analyser pour les algorithmes d'identification constitue une difficulté pour deux raisons :

- d'une part, comme vu précédemment, le taux de fausses acceptances croît linéairement avec la taille de la base de données
- d'autre part, pour chaque requête, la similarité s doit être calculée pour toutes les empreintes digitales de la base de référence

Maintenir un taux de fausses acceptances raisonnable en identification nécessite des algorithmes de comparaison générant de faibles taux de fausse acceptance en vérification pour un taux de faux rejets donné, c'est-à-dire des algorithmes très performants. Or, de tels algorithmes sont généralement coûteux en temps de calcul de similarité. Pour conserver des temps de calcul raisonnables, une approche courante consiste à utiliser des algorithmes de *filtrage*. Les algorithmes de filtrage sont des méthodes de comparaison d'empreintes digitales permettant de

calculer des scores de similarité peu coûteux en temps de calcul, tout en atteignant des taux de faux rejets très faibles à des taux de fausse acceptation raisonnables en vérification.

Un algorithme de filtrage peut alors être combiné avec un algorithme d'identification de la façon suivante : l'algorithme de filtrage calcule un score de similarité approximatif s' entre la requête et les empreintes de référence, et un seuil δ' sur s' permet de sélectionner une sous-partie de la base de données de référence sur laquelle est appliqué l'algorithme d'identification.

La proportion de la base de données vérifiant une similarité approximative s' supérieure à δ' , c'est-à-dire la proportion de la base de données transférées à l'algorithme d'identification, est appelée *taux de pénétration (penetration rate)*. Les performances d'un algorithme de filtrage sont exprimées en terme de taux de faux rejets en fonction du taux de pénétration.

Le taux de faux rejets en identification avec ajout d'un algorithme de filtrage se réécrit alors

$$FNIR = PRR(\delta') + (1 - PRR) \cdot FRR$$

où PRR est le taux de faux rejets de l'algorithme de filtrage (*Preselection Error Rate*)

Le taux de fausse acceptation en identification se réécrit également

$$FPIR = 1 - (1 - FAR)^{PR(\delta')M}$$

où PR est le taux de pénétration de l'algorithme de filtrage.

Un algorithme de filtrage permet donc de réduire le nombre de fausses acceptations en identification, tout en réduisant le temps d'exécution total de l'algorithme, au prix d'un accroissement du nombre de faux rejets en identification matérialisé par le taux de faux rejets en filtrage PRR .

Il est important de noter que les taux de fausse acceptation et de faux rejets du ou des algorithmes d'identification qui suivent un algorithme de filtrage, doivent être considérés *après filtrage*. Les probabilités $FNIR$ et $FPIR$ doivent alors être comprises comme

$$FNIR_{post\ filtrage} = Pr(\forall i D_{NM^i} | \exists i / H_{M^i} \text{ et } s'(S, R^i) > \delta')$$

et

$$FPIR_{post\ filtrage} = Pr(\exists i / D_{M^i} \text{ et } s'(S, R^i) > \delta' | \forall i H_{NM^i})$$

Si les informations utilisées par l'algorithme de filtrage et l'algorithme d'identification sont décorrélées, alors

$$Pr(\forall i D_{NM^i} | \exists i / H_{M^i} \text{ et } s'(S, R^i) > \delta') = Pr(\forall i D_{NM^i} | \exists i / H_{M^i})$$

et

$$Pr(\exists i / D_{M^i} \text{ et } s'(S, R^i) > \delta' | \forall i H_{NM^i}) = Pr(\exists i / D_{M^i} | \forall i H_{NM^i})$$

Si ce n'est pas le cas, il est alors nécessaire de réévaluer les probabilités post-filtrage.

Cascade d'algorithmes

Il est tout à fait envisageable, et même nécessaire dans le cas de grandes bases de données, d'enchaîner successivement plusieurs algorithmes de filtrage, chaque algorithme recevant en entrée une sous-partie de la base de données initiales, opérant un filtrage dessus avec la requête initiale, et transmettant à l'algorithme suivant une sous-base encore restreinte, jusqu'à arriver à l'algorithme d'identification final.

Le but est de placer en début de traitement les algorithmes les plus rapides et les filtrants pour affiner le problème, c'est-à-dire rejeter le plus d'images non matchantes possible, en gardant les algorithmes les plus fiables, et donc les plus coûteux en temps de calcul, en fin de requête.

Formellement, si on considère K algorithmes de filtrage, chacun caractérisés par des fonctions de similitudes respectives s'_1, \dots, s'_K , des seuils de décision respectifs $\delta'_1, \dots, \delta'_K$ et des couples de fonctions respectifs

$((PRR_1, PR_1), \dots, (PRR_K, PR_K))$, alors, sous l'hypothèse que chaque algorithme utilise une information indépendante des algorithmes précédents, le taux de faux rejets $FNIR_i$ en identification de la cascade d'algorithme pour l'algorithme de rang i sera défini de manière récursive par

$$FNIR_1 = PRR(\delta'_1) + (1 - PRR(\delta'_1)) \cdot FRR(\delta'_2)$$

et pour $i > 1$

$$FNIR_i = FNIR_{i-1} + (1 - FNIR_{i-1}) \cdot FRR(\delta'_i)$$

le taux de faux rejets $FNIR_{cascade}$ de l'ensemble de la cascade d'algorithmes est défini par $FNIR_{cascade} = FNIR_K$, et le taux de fausse acceptation en identification $FPIR_{cascade}$ de la cascade est défini par

$$FPIR_{cascade} = 1 - (1 - FAR)^{\prod_{i=1}^K PR_i(\delta'_i)}$$

Les algorithmes de filtrage successifs et leurs seuils de décision respectifs $\delta'_1, \dots, \delta'_K$ sont choisis de manière à arbitrer entre l'ajout de faux rejets, le taux de fausse acceptation en identification total, et le coût en temps de calcul de l'algorithme.

1.4 Problématique

Les problématiques de performances des algorithmes de comparaison d'empreintes digitales étant posées, définissons maintenant l'objectif principal de cette thèse. Nous nous intéressons au développement d'un algorithme de filtrage initial de très grandes base de données d'empreintes digitales en identification, et nous nous plaçons dans le cas particulier où les requêtes sont susceptibles d'être des empreintes digitales de type *latentes*, bruitées ou partiellement occultées.

Les contraintes sont donc les suivantes :

- S'agissant d'un algorithme de filtrage *initial*, opérant en première passe sur des très grandes bases de données, le coût en temps de calcul du filtrage rapporté à la taille des bases doit être le plus faible possible, pour rendre le coût de calcul de l'ensemble de la cascade raisonnable.

- L'accélération de la procédure d'identification par ce filtrage ne doit pas être obtenue au prix d'un grand nombre de faux rejets. Les cascades d'algorithmes en identification avec requêtes latentes sont typiquement des applications d'identification de criminels à partir de données de scènes de crime, et les faux rejets à l'identification sont à proscrire autant que possible.
- Dans la mesure du possible, l'algorithme choisi ne doit pas amener à une augmentation significative des coûts de stockage des empreintes digitales, ou de la mémoire vive nécessaire à la procédure d'identification.

Le cadre de la problématique peut être synthétisé ainsi : une empreinte digitale est d'abord une image, dont les caractéristiques principales à notre disposition sont les crêtes et les minuties. Nous aborderons donc notre problématique en deux temps : d'abord une revue des méthodes de recherche d'image de l'état de l'art (chapitre 2), puis une approche plus spécifique à l'empreinte digitale, où nous verrons comment les méthodes développées en recherche d'image en général peuvent s'adapter à ces images particulières, pour assurer la précision de la recherche d'abord (chapitre 3). Enfin, nous verrons quelles méthodes d'accélération du processus de recherche peuvent s'adapter du cas de figure général au cas des empreintes digitales (chapitre 4).

Chapitre 2

Recherche d'images par le contenu : Application à l'empreinte digitale

2.1 Généralités

Dans ce chapitre, nous nous intéressons à l'empreinte digitale en tant qu'image, et nous allons considérer la problématique sous l'angle d'un processus de *recherche d'images*.

Dans la littérature scientifique, le terme de recherche d'images fait couramment référence à un processus de recherche et d'exploration dans de grandes bases de données d'images digitales.

Depuis une quinzaine d'années, l'intérêt pour ce domaine de recherche a cru de façon exponentielle, pour la simple raison que les besoins pour des techniques de recherche d'images efficaces ont cru au même rythme. Les dispositifs d'acquisition d'images, essentiellement des caméras et appareil photos mais aussi des outils d'imagerie médicale ou des capteurs pour les longueurs d'ondes en dehors du domaine visible, sont de plus en plus répandus, faciles à acheter et à manipuler, augmentant ainsi drastiquement la quantité d'images digitales acquises chaque jour dans le monde. Cette croissance du volume d'acquisitions s'est faite en parallèle de l'accroissement des capacités de stockage. Pour de nombreuses applications informatiques, la quantité de données disponibles n'est plus un problème : le problème actuel est désormais la gestion efficace d'immenses quantités de données.

Le concept d'image retrieval est extrêmement large, et peut englober de nombreux outils, depuis une simple fonction de similarité entre images à des méthodes avancées de clusterisation de bases de données d'images. Une première distinction très basique peut être faite parmi les méthodes de recherche d'images entre celles faisant usage des *métadonnées* et celles utilisant exclusivement le contenu de l'image. Les méthodes de la première catégorie, aussi appelées *métarecherche d'image* (*image meta search*), utilisent les métadonnées, c'est-à-dire toutes les informations non picturales supplémentaires ajoutées à une image, comme les titres, les dates, les propriétés des instruments d'acquisitions, les annotations humaines, etc, pour procéder à

la recherche. Quand elles sont disponibles, les métadonnées peuvent considérablement contribuer à la qualité et la rapidité des méthodes de recherche d'images. Des métadonnées comme l'âge ou le sexe des individus, ou encore de quel doigt de la main est issue l'empreinte digitale, peuvent grandement améliorer la précision et le temps de calcul d'un algorithme de recherche d'image appliqué à l'empreinte digitale. Cependant, dans de nombreuses situations, les métadonnées sont soit non disponibles, soit sont peu utiles pour les scénarii de recherche courant.

Dans le cas d'un algorithme de filtrage pour empreintes digitales, nous nous placerons dans le cas le plus général où les métadonnées ne sont pas disponibles, ou ne contribuent d'aucune manière à la vérification de correspondance entre empreintes.

Nous nous concentrerons donc exclusivement sur la deuxième catégorie de méthodes, généralement appelées méthodes de recherche d'images par le contenu (*content-based image retrieval*).

Une étude très complète et exhaustive des méthodes de recherche d'images par le contenu a été réalisée il y a quelques années par [8]. Des revues plus récentes sont disponibles, notamment [9] [10], [11], mettant à jour les dernières avancées du domaine, en conservant les grandes lignes tracées par [8].

Dans la vaste majorité des méthodes de CBIR, l'objectif de l'exploration de la base de données est la reconnaissance d'un *objet* ou d'un *concept* dans les images de la base. L'objet à rechercher peut être soit défini formellement par une sémantique humaine, soit uniquement défini par une image, sans plus de caractérisation.

Dans les deux cas, un processus de reconnaissance de l'objet est nécessaire, qui intervient en deux temps :

- d'abord, une étape de *capture* durant laquelle un enregistrement digital, c'est-à-dire une image, de l'objet réel est acquise
- ensuite, une étape d'*interprétation*, durant laquelle l'information digitale de l'enregistrement est traitée de manière à produire une représentation mathématique de l'objet, pouvant être comparée de façon automatique avec les représentations des objets extraits des autres enregistrements de la base de données

Ces deux étapes définissent les deux principaux défis à relever par toute méthode de CBIR :

- l'écart d'acquisition (*sensory gap*)
- l'écart sémantique (*semantic gap*)

L'écart d'acquisition définit la différence de représentation entre l'objet lui-même dans le monde réel, et l'enregistrement digital qui en est fait par un appareil d'acquisition tel qu'une caméra.

L'écart sémantique définit l'écart de représentation entre l'information directement extractible de l'enregistrement digital, aussi appelées *caractéristiques de bas niveau*, comme la couleur, la texture, la forme, les points saillants, et leur interprétations en caractéristiques complexes,

de *haut niveau*, comme des bâtiments, des personnes, des animaux, etc.

Formulé différemment, l'écart d'acquisition détermine quelle information d'un objet l'ordinateur reçoit du monde réel, et l'écart sémantique détermine comment l'ordinateur utilise ces informations pour reconnaître l'objet, ou la catégorie de l'objet, pour ce qu'il est. Ces deux écarts constituent deux difficultés à franchir pour réaliser la reconnaissance d'un objet ou d'un concept dans une image.

La difficulté de franchissement de l'écart d'acquisition dépend fortement du type d'image que l'on traite et des appareils d'acquisitions utilisés. Dans le cas d'images photographiques par exemple, cela dépend essentiellement de la résolution de la caméra et de la précision de la colorimétrie.

Une analyse approfondie de l'écart d'acquisition n'entre pas dans le cadre de nos travaux. Les problèmes d'acquisition relèvent essentiellement de problématiques matérielles, et dans une moindre mesure d'algorithmes de retraitement d'images brutes, déterminés par des contraintes de coût et de facilité d'utilisation du matériel d'acquisition qui priment sur les contraintes de gestion des bases de données par la suite.

Pour la suite de nos travaux, on prendra en compte les éléments suivants : la majorité des images que nous considérons, c'est-à-dire des images d'empreintes digitales, sont des images en niveaux de gris, acquise à des résolutions allant couramment de 500 à 1000 dpi.

Si nous revenons sur l'étape sémantique, celle-ci peut encore être subdivisée en deux sous-étapes :

- l'extraction de motifs de base
- la construction de *signatures visuelles*

Lors de l'extraction des motifs de base, l'information brute issue de l'image digitale est traitée afin de construire des motifs visuels de bas niveau, tels que couleurs, orientation, ou points saillants.

L'extraction des motifs de base, appelée aussi couramment *encodage* en biométrie, est assurée par des algorithmes de traitement d'image de bas niveau.

A l'heure actuelle, plusieurs algorithmes d'encodage, académiques ou commerciaux, sont disponibles pour la construction de systèmes de reconnaissance biométrique par empreinte digitale. Les performances en termes de fiabilité de la comparaison d'un même algorithme de construction et de comparaison de signatures visuelles peuvent varier significativement en fonction de la qualité des données produites par l'encodage. Nos travaux se concentrent sur la construction et la comparaison des signatures visuelles, aussi nous n'utiliserons pour l'ensemble de nos expérimentations qu'un unique algorithme d'encodage. La qualité intrinsèque de l'algorithme d'encodage utilisé est suffisamment établie par divers benchmarks pour s'assurer que la qualité des motifs de base qu'il extrait ne soit pas le facteur limitant à la performance des algorithmes de haut niveau que nous construirons.

2.2 Les types de représentation d'image

Le motif de base d'une image digitale, à savoir le niveau ou la valeur d'un pixel, est par nature une information extrêmement locale. Toutes les méthodes de construction d'une signature mathématique d'image à partir de ses données digitales brutes consistent à mettre en perspective cette information locale avec des données plus ou moins voisines, afin d'en extraire une information de plus en plus globale. C'est cette mise en perspective qui constitue la clé du franchissement de l'écart sémantique : l'ensemble de l'information disponible dans une image permet d'y reconnaître des formes et des objets complexes.

Il existe deux grandes approches pour la construction de signatures d'images : une approche globale et une approche locale.

2.2.1 Approche globale

Dans l'approche globale, on cherche à agréger l'information de base de l'image directement dans des descripteurs de haut niveau, descripteurs susceptibles de couvrir une grande partie de l'image voire la totalité, si bien qu'on ne produit qu'un nombre très réduit de descripteurs, voire un seul. Ces descripteurs peuvent être assez simple, un exemple fameux étant le descripteur Gist de [12], ou plus complexes, par exemple le degré de correspondance de l'image à un modèle de forme ou d'objet, comme dans le cas de certains algorithmes de reconnaissance de visage [13].

L'approche globale présente plusieurs intérêts. D'abord, en produisant un petit nombre de descripteurs voire un descripteur unique, elle facilite le processus de comparaison entre signatures d'image, en réduisant au minimum le nombre d'opérations de comparaison. Ensuite, elle est intrinsèquement conçue pour tenir compte de la globalité de l'information présente sur l'image, évitant la perte d'information.

Elle présente néanmoins un défaut majeur : si la construction du descripteur global repose sur l'ensemble de l'information de l'image, alors le calcul de la signature risque d'être peu robuste au bruit, et surtout d'être très peu robuste à l'occultation d'une partie de l'image.

Parmi les contraintes que nous nous sommes fixés dans notre étude, figure la robustesse de l'algorithme de comparaison au bruit et à l'occultation des images requêtes, aussi nous ne conserverons pas l'approche globale pour la suite de nos travaux.

2.2.2 Approche locale

Dans l'approche locale, l'information de base de l'image est, dans un premier temps, agrégée dans des *descripteurs locaux* de bas niveau ou de niveau intermédiaire, ne couvrant chacun qu'une fraction de la surface de l'image. Ces descripteurs locaux sont généralement standardisés : l'information est agrégée de la même manière en tout point de l'image.

Dans un second temps, ces descripteurs locaux peuvent être enrichis d'une information globale, en particulier la disposition spatiale des descripteurs locaux entre eux.

Enfin, dans un troisième temps, l'information des descripteurs locaux peut être agrégée à une échelle supérieure, pour obtenir une représentation globale agrégée de l'empreinte. On notera que cette représentation globale agrégée est d'une nature différente de l'approche globale

précédemment décrite : ici, le franchissement de l'écart sémantique se fait en plusieurs étapes, du local au global, alors que dans l'approche globale directe, on s'efforce de construire une représentation globale sans passer par des étapes intermédiaires locales.

L'approche locale présente comme intérêt majeur pour notre problématique une grande robustesse au bruit et à l'occultation des images, aussi nous privilégierons cette approche par rapport à l'approche globale. Cependant, l'approche locale présente comme défaut majeur une complexité accrue de la signature visuelle de l'image, définie comme un ensemble potentiellement grand de descripteurs locaux, là où l'approche globale pouvait représenter une image par un seul objet mathématique. Pour maintenir des temps de calcul raisonnables avec l'approche locale, il sera nécessaire de développer des heuristiques et des méthodes d'accélération de la recherche.

Descripteurs locaux

Depuis les apports fondateurs du descripteur SIFT par [14] et reformulés par [15], de nombreux descripteurs locaux ont été proposés en reconnaissance d'image. Plusieurs revues des descripteurs locaux couramment utilisés en reconnaissance d'image couvrent les travaux dans ce domaine [16] [17], ou plus récemment [10], et certains descripteurs locaux prometteurs comme ORB[18] ou FREAK [19]) ont été proposés.

La majorité de ces descripteurs s'efforcent de proposer une signature mathématique qui soit à la fois invariante aux transformations affines et robuste au bruit, voire aux occultations. Cet effet peut être atteint soit par extraction d'un référentiel géométrique local [14] soit par extraction de caractéristiques intrinsèquement affine-invariantes, comme des moments ou des histogrammes.

L'intérêt d'un descripteur local affine-invariant est qu'il n'est pas nécessaire de tenir compte des différences de pose entre deux motifs locaux pour procéder à leur comparaison. Dans ce cas, le descripteur peut alors souvent être ramené à une représentation vectorielle, la dissimilarité entre deux descripteurs pouvant être calculée avec des métriques peu coûteuses en temps de calcul (normes ℓ_1 , ℓ_2 , distances de Minkowski, distances de Hamming, etc).

Pour des descripteurs non-affine invariants, il faut tenir compte de la différence de pose entre les images avant de pouvoir comparer les descripteurs. Cela passe généralement par le calcul d'une transformation affine permettant de superposer au mieux les deux images entières ou simplement les deux motifs locaux, processus généralement coûteux en temps de calcul.

La signature mathématique obtenue par construction de descripteurs locaux à partir d'une image I se présente donc dans le cas le plus général comme un ensemble de n_I vecteurs $\{z_i\}_{1 \leq i \leq n_I}$, où le nombre n_I de descripteurs locaux et leur localisation est variable d'une image à l'autre. En sus de ces descripteurs locaux, on dispose souvent d'une information globale sur l'image, par exemple un ensemble de pondérations $p_{i1 \leq i \leq n_I}$ associés à leurs descripteurs locaux respectifs, ou une information de localisation spatiale des descripteurs locaux entre eux.

2.3 Agrégation des descripteurs locaux

Dans cette section, nous supposons que nous disposons sur les images à comparer d'une signature visuelle définissant des descripteurs locaux z à valeur dans un espace métrique E . Chaque image peut alors être caractérisée par un ensemble fini de descripteurs locaux (z_1, \dots, z_n) . A chaque descripteur local z_i , on associe un facteur de pondération p_i proportionnel à l'importance par rapport à l'ensemble de l'image du descripteur local considéré. Les p_i sont le plus souvent fixé au préalable de la comparaison entre images. Un cas particulier couramment rencontré est le cas où $\forall i, p_i = \frac{1}{n}$, c'est-à-dire que tous les descripteurs locaux sont également importants.

2.3.1 Agrégation directe

Distances courantes

Une approche naturelle pour comparer deux images définies comme deux ensembles de vecteurs de descripteurs locaux $I_1 = \{(z_{1,1}, p_{1,1}), \dots, (z_{1,n_1}, p_{1,n_1})\}$, $I_2 = \{(z_{2,1}, p_{2,1}), \dots, (z_{2,n_2}, p_{2,n_2})\}$ proposée par [20] est l'affectation d'un poids s_{ij} à chaque appariement $(z_{1,i}, z_{2,j})$ de descripteurs locaux.

Une distance agrégée I_1 et I_2 peut alors être calculée comme suit :

$$D(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} d(z_{1,i}, z_{2,j}) \quad (2.1)$$

Les poids (s_{ij}) permettent de définir des contraintes heuristiques sur les appariements des descripteurs locaux, ils peuvent être fixés au préalable, ou bien être recalculés sous certaines contraintes pour chaque comparaison d'images.

Une heuristique courante consistera par exemple à imposer que $\forall j \sum_i s_{ij} = p_{2,j}$ et $\forall i \sum_j s_{ij} = p_{1,i}$. Cette heuristique signifie que le poids $\sum_j s_{ij}$ de tous les appariements auquel un descripteur participe doit être égal à la pondération p_j de ce descripteur.

Cette heuristique participe notamment à la définition de l'*Earth Moving Distance* (EMD) :

$$\text{EMD}(I_1, I_2) = \min_{s_{ij}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} d(z_{1,i}, z_{2,j}) \quad (2.2)$$

sous les contraintes $\forall j \sum_i s_{ij} = p_{2,j}$, $\forall i \sum_j s_{ij} = p_{1,i}$ et $\forall i, j s_{ij} \geq 0$.

$p_{1,i}$ et $p_{2,j}$ peuvent également correspondre à des distributions de probabilité fixées au préalable, dans ce cas les contraintes seront $\sum_i p_{1,i} = 1$ et $\sum_j p_{2,j} = 1$. On peut alors considérer la première distance de Mallows (ou première distance de Wasserstein) D_M pour les distributions

discrètes [21] :

$$D_M(I_1, I_2) = \min_{s_{ij}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} d(z_{1,i}, z_{2,j}) \quad (2.3)$$

sous les contraintes $(\forall j \sum_i p_{1,i} = 1, \forall i \sum_j p_{2,j} = 1) \Leftrightarrow \sum_{ij} s_{ij} = 1$.

[20] propose une autre heuristique, connue comme l'*Integrated Region Matching* (IRM).

Dans ce cas de figure, les coefficients de poids (s_{ij}) sont déterminés en cherchant le meilleur appariement $d(z_{1,p}, z_{2,q})$ parmi $\{d(z_{1,i}, z_{2,j})\}$ (i.e. c'est-à-dire la plus faible distance), et en attribuant la valeur $\min(p_p^1, p_q^2)$ au poids correspondant s_{pq} .

Ensuite, on soustrait la valeur de s_{pq} à p_p^1 et à p_q^2 , mettant l'une des deux valeurs à zéro, puis on itère à nouveau l'opération avec le meilleur appariement suivant, jusqu'à ce que tous les poids de la requête ou tous les poids de la référence soient à zéro.

Heuristiquement, cela signifie que la pondération p_i du vecteur caractéristique $z_{1,i}$ est attribuée en priorité aux appariements $d(z_{1,i}, z_{2,j})$ qui répondent le mieux à $z_{1,i}$.

L'ensemble de l'algorithme [20] est décrit dans le pseudo-code suivant 1, et illustré dans les figures 2.1, 2.2, 2.3.

Algorithm 1 IRM matching

entrée : On considère les images $I_1 = \{(z_{1,1}, p_{1,1}), \dots, (z_{1,n_1}, p_{1,n_1})\}$, $I_2 = \{(z_{2,1}, p_{2,1}), \dots, (z_{2,n_2}, p_{2,n_2})\}$

1. Soit $L = \{\}$, $M = \{(i, j)\}_{1 \leq i \leq m, 1 \leq j \leq m, \forall i, j s_{i,j} = 0}$
 2. On trouve (i', j') tel que $d(z_{1,i'}, z_{2,j'}) = \min_{i,j} d(z_{1,i}, z_{2,j})$
 3. $s_{i',j'} \leftarrow \min(p_{1,i'}, p_{2,j'})$
 4. Si $p_{1,i'} < p_{2,j'}$
on pose $s_{i',j} = 0$
 $p_{1,i'} \leftarrow 0$
 $p_{2,j'} \leftarrow p_{2,j'} - p_{1,i'}$
Sinon
on pose $s_{ij'} = 0$
 $p_{1,i'} \leftarrow p_{1,i'} - p_{2,j'}$
 $p_{2,j'} \leftarrow 0$
 5. Si $\sum_i p_{1,i} > 0$ et $\sum_j p_{2,j} > 0$ retour à l'étape 2, autrement arrêter
-

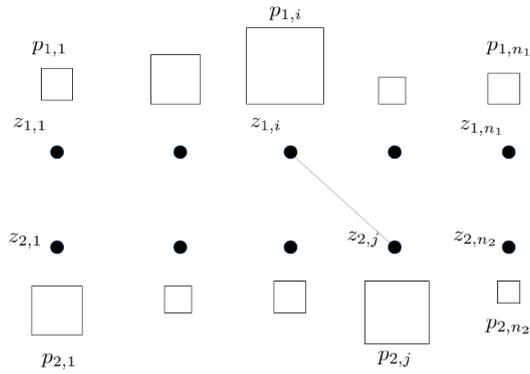


FIGURE 2.1 – Sélection du meilleur appariement

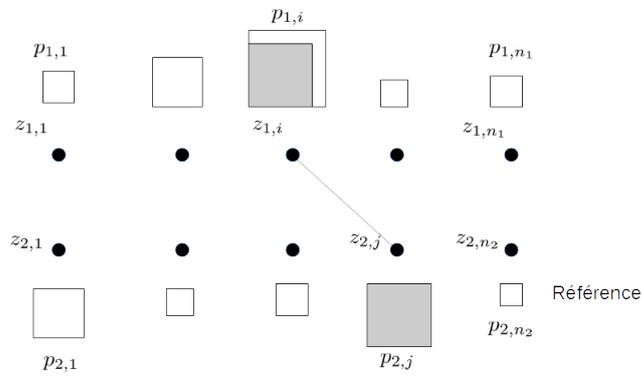


FIGURE 2.2 – Correction des poids

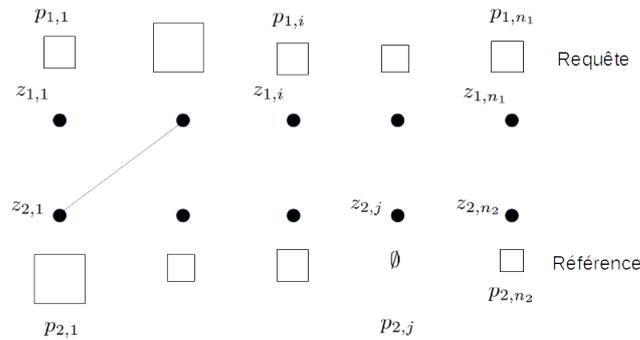


FIGURE 2.3 – Répétition de l'opération jusqu'à ce que tous les poids d'une des deux images soient à zéro

Dans le cas où toutes les régions de l'images ont même pondération, on a $p_i^k = \frac{1}{n^k}$, et la distance IRM devient

$$D(I_1, I_2) = \begin{cases} \frac{\sum_{i=1}^{n_1} \min_j d(z_{1,i}, z_{2,j})}{n_1} & \text{si } n_1 < n_2 \\ \frac{\sum_{j=1}^{n_2} \min_i d(z_{1,i}, z_{2,j})}{n_2} & \text{sinon} \end{cases} \quad (2.4)$$

Une autre méthode de comparaison est la *distance de Hausdorff*, définie comme suit :

$$D(I_1, I_2) = \max \left(\max_i \left(\min_j (d(z_{1,i}, z_{2,j})) \right), \max_j \left(\min_i (d(z_{2,j}, z_{1,i})) \right) \right) \quad (2.5)$$

Heuristiquement, chaque $z_{1,i}$ de I_1 est comparé à son plus proche descripteur local dans I_2 , puis la distance $D(I_1, I_2)$ est calculée comme le maximum des distances. Cette distance a été utilisée entre autres par [22]

Distance a contrario

Dans l'approche a contrario telle que définie par [23], l'objectif est de comparer un motif requête S avec d'autres motifs de références issus d'une base de données B , et de reconnaître les motifs issus d'un même objet. On suppose que chaque motif S est représenté par un *code*, c'est à dire un ensemble de K caractéristiques locales $(x_i)_{1 \leq i \leq K}$. Chacune de ces caractéristiques x_i est définie sur un ensemble E_i , doté d'une pseudo-distance ou dissimilarité d_i . La pseudo-distance "produit" d est alors définie par $\forall (S, S') \in B^2$ telle que $S = (x_i)_{1 \leq i \leq K}$, $S' = (x'_i)_{1 \leq i \leq K}$, on a

$$d(S, S') = \max_{i \in [1 \dots K]} d_i(x_i, x'_i) \quad (2.6)$$

On suppose ici que les ensembles E_i contiennent des informations comparables, et que les pseudo-distances d_i définies dessus sont comparables.

Une méthode de décision envisageable peut alors consister à estimer si la distance $d(S, S')$ a été observée "par hasard", ou parce que S et S' sont deux occurrences du même objet.

On formalise ces deux hypothèses ainsi :

- \mathcal{H}_0 : la distance entre S et S' a été observée *par hasard*, du fait de la grande fréquence de ce type de motif dans la base de données
- \mathcal{H}_1 : la distance entre S et S' a été observée parce que les deux motifs sont liés, et sont issus d'un même objet sous-jacent

Etant données ces deux hypothèses, pour un seuil de distance δ , on définit deux probabilités :

- la probabilité de fausse alarme $FA = Pr(d(S, S') \leq \delta | \mathcal{H}_0)$
- la probabilité de faux rejet $FR = Pr(d(S, S') > \delta | \mathcal{H}_1)$

En notant Ω l'ensemble des motifs locaux S , on note $Pr(\cdot | \mathcal{H}_0)$ et $Pr(\cdot | \mathcal{H}_1)$ les probabilités respective d'observer \mathcal{H}_0 et \mathcal{H}_1 sur Ω

Le choix entre ces deux hypothèses peut être fait en comparant $d(S, S')$ avec un seuil de distance δ . On supposera que \mathcal{H}_0 est vraie si $d(S, S') < \delta$, et que \mathcal{H}_1 est vraie autrement.

La probabilité de fausse alarme α est alors définie par

$$\alpha = Pr(d(S, S') \leq \delta | \mathcal{H}_0)$$

La probabilité de faux rejet α' est définie par

$$\alpha' = Pr(d(S, S') \leq \delta | \mathcal{H}_1)$$

Pour certaines données, il est possible de construire un modèle statistique des motifs S , permettant d'estimer $Pr(\cdot|\mathcal{H}_1)$. Mais certains types de données ne se prêtent que très difficilement à la construction d'un modèle statistique, généralement du fait de leur très forte variabilité. Dans ce genre de cas, qui inclut notamment les empreintes digitales, les méthodes de décision nécessitant une estimation de $Pr(\cdot|\mathcal{H}_1)$ comme le test bayésien ne sont pas envisageables.

Si l'on dispose d'un échantillon d'observation suffisamment vaste de motifs S de Ω , on peut calculer un *modèle de fond* des motifs S permettant d'estimer $Pr(\cdot|\mathcal{H}_1)$. Formellement, il s'agit d'un espace de probabilité (Ω, A, P) , Ω étant l'espace des motifs S , A l'ensemble des événements \mathcal{H}_0 et \mathcal{H}_1 , et P est une mesure sur Ω définie par $P = Pr(\cdot|\mathcal{H}_1)$.

On supposera pour la suite que ce modèle de fond vérifie que les variables aléatoires

$$\Sigma_i \longrightarrow d_i(x_i, x'_i), 1 \leq i \leq K$$

définies de Ω sur R^+ sont mutuellement indépendantes.

La probabilité de fausse alarme FA entre deux motifs S et S' pour un seuil δ s'écrit alors

$$\begin{aligned} FA(S, \delta) &= Pr(d(S, S') \leq \delta | H_0) \\ &= Pr\left(\max_{i \in [1..K]} d_i(x_i, x'_i) \leq \delta | H_0\right) \end{aligned} \tag{2.7}$$

et sous l'hypothèse d'indépendance des codes, on a

$$FA = \prod_{i \in [1..K]} Pr(d_i(x_i, x'_i) \leq \delta | H_0)$$

Si l'on dispose déjà d'un algorithme ou d'une heuristique construisant une injection c permettant d'associer chacun des descripteurs locaux $\{z_{1,i}\}$ de I_1 à un des descripteurs locaux $\{z_{2,i}\}$ de I_2 ,

$$c : z_i \in I_1 \longrightarrow c(z_i) \in I_2$$

alors la probabilité de fausse alarme peut être interprétée comme une distance *a contrario* D_{ac} entre I_1 et I_2 , avec

$$D_{ac}(I_1, I_2) = \prod_{i \in [1..K]} Pr(d_i(z_i, c(z_i)) \leq \delta | H_0) \tag{2.8}$$

La distance *a contrario* peut être considérée comme une distance entre descripteurs locaux, quantifiant la probabilité que la proximité entre deux ensembles de descripteurs ne soit pas due au hasard.

2.3.2 Consolidation de l'agrégation directe : transformée de Hough généralisée

La transformée de Hough a été introduite par les travaux par [24]. L'algorithme dans sa version classique consistait à reconnaître des lignes dans une image en associant à chaque point remarquable de l'image un couple de paramètres géométriques, et à "faire voter" les paramètres calculés dans un espace de paramètres, appelé espace de Hough.

La transformée de Hough généralisée est une variante de l'algorithme classique de [24], développée par [25]. Dans l'algorithme de base, on cherche à reconnaître un objet dans une image, caractérisé par un modèle géométrique, en cherchant à retrouver la position du modèle dans l'image. Dans la version généralisée, on transforme le problème en recherchant les paramètres de transformation géométriques qui projettent le modèle sur l'image.

En pratique, chaque motif caractéristique de l'image analysée se voit attribuer un ensemble de paramètres correspondants à sa pose géométrique. En 2 dimensions, cela correspondra à l'orientation, l'échelle, la localisation en x et en y , soit 4 degrés de liberté. En 3 dimensions, le nombre de degrés de liberté passe à 6.

En notation complexe, la pose d'un descripteur local z_i d'une scène bidimensionnelle I_p est entièrement définie par son origine $\rho e^{i\alpha}$, son échelle s et son orientation θ .

Lors de la comparaison de deux images bidimensionnelles I_1 et I_2 , entre chaque appariement d'un descripteur local z_1 de I_1 vers un descripteur local z_2 de I_2 , caractérisé respectivement par deux jeux de paramètres géométriques $(\rho_1 e^{i\alpha_1}, s_1, \theta_1)$ et $(\rho_2 e^{i\alpha_2}, s_2, \theta_2)$ il existe une unique similitude T transformant z_1 en z_2 .

$$T(z_1) = z_2$$

T est entièrement définie par

$$\forall t \in \mathbf{C}, T(t) = \frac{s_1}{s_2} e^{i(\theta_2 - \theta_1)} (t - \rho_1 e^{i\alpha_1}) + \rho_2 e^{i\alpha_2}$$

A chaque appariement correspond donc une similitude T dans un espace de transformations \mathcal{T} borné à 4 dimensions. [25] propose ensuite de discrétiser l'espace T en clusters réguliers. Chaque transformation affine "vote" ensuite pour le cluster dans lequel elle tombe. Si deux images contiennent un même objet, alors les appariements de motifs caractéristiques de cet objet dans les deux images vont massivement voter pour un même cluster. Ce vote permet d'éliminer du calcul d'un score agrégé entre images tous les appariements qui n'ont pas voté pour le meilleur cluster, appariements qui ont une forte probabilité d'être des fausses acceptances.

La transformée de Hough généralisée peut alors se formuler simplement de la manière suivante :

Pour tout couple d'images I_1, I_2 définis par leurs ensembles respectifs de descripteurs locaux $(z_i^1)_{1 \leq i \leq n_1}$ et $(z_{2,i})_{1 \leq i \leq n_2}$, descripteurs locaux caractérisés par leurs ensembles respectifs

$(p_{1,i}, s_{1,i}, \theta_{1,i})_{1 \leq i \leq n_1}$ et $(p_{2,i}, s_{2,i}, \theta_{2,i})_{1 \leq i \leq n_2}$ de paramètres géométriques,

On note H la fonction qui à toute paire $(z_{1,p}, z_{2,q})$ de $I_1 \times I_2$ associe l'unique similitude T_{pq} qui transforme z_p dans le référentiel de I_1 en z_q dans le référentiel de I_2

$$H : I_1 \times I_2 \longrightarrow \mathcal{T} \quad (2.9)$$

$$(z_p, z_q) \in \longrightarrow T_{pq} \quad (2.10)$$

Soit $\mathbf{H} = [cl_1, \dots, cl_M]$ une clusterisation de l'espace borné \mathcal{T} des similitudes en M clusters, définie par la fonction de quantification h qui à toute similitude T de \mathcal{T} associe un unique cluster $h(T) \in H$

Algorithm 2 transformée de Hough généralisée

Soit $\mathbf{H} = [cl_1, \dots, cl_M]$ une clusterisation de \mathcal{T} formée de M clusters, et définie par la fonction de quantification $h : \mathcal{T} \longrightarrow \mathbf{H}$

Soit CL la matrice d'accumulation des similitudes

Soient $I_1 = (z_{1,i}, p_{1,i}, s_{1,i}, \theta_{1,i})_{1 \leq i \leq n_1}$ et $I_2 = (z_{2,i}, p_{2,i}, s_{2,i}, \theta_{2,i})_{1 \leq i \leq n_2}$ deux images avec leurs descripteurs locaux respectifs

for all $i \in [cl_1, \dots, cl_M]$ **do**

$CL[i] = 0$

end for

for all $(p, q) \in [1, n_1] \times [1, n_2]$ **do**

$CL[h(T_{pq})] \leftarrow CL[h(T_{pq})] + 1$

end for

$cl_H(I_1, I_2) = cl_{\hat{i}}$ tel que $\hat{i} = \arg \max_i CL[i]$

On peut alors obtenir une approximation de la meilleure similitude entre I_1 et I_2 en prenant le centre du cluster le plus voté $cl_H(I_1, I_2)$

[15] propose de raffiner cette approche, en explicitant la meilleure similitude. Pour cela, il prend en compte uniquement les appariements qui ont voté pour le meilleur cluster cl_H , et procède à une minimisation aux moindres carrés entre ces appariements, de la façon suivante :

La similitude T entre un point (x, y) de l'image I_1 et un point (u, v) de l'image I_2 peut s'écrire

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2.11)$$

où (m_i) sont les paramètres de rotation, échelle et distorsion affine, et t_x, t_y sont les paramètres

de translation. L'équation précédente peut être réécrite sous la forme

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.12)$$

Pour un nombre n d'appariements $(x_1, y_1, u_1, v_1), \dots, (x_n, y_n, u_n, v_n)$, on a l'équation

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ \dots & & & & & \\ x_n & y_n & 0 & 0 & 1 & 0 \\ 0 & 0 & x_n & y_n & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix} \quad (2.13)$$

L'égalité peut ainsi être réécrite

$$\mathbf{Ax} = \mathbf{b} \quad (2.14)$$

qui admet comme solution aux moindres carrés

$$\mathbf{x} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b} \quad (2.15)$$

La solution aux moindres carrés permet d'éliminer encore d'éventuels appariements fausse acceptance, parmi les transformations affines éloignées de la solution.

Ajout de la transformée de Hough généralisée à un score ou une distance agrégée

Dans l'expression générale de la distance agrégée 2.1, on peut envisager de corriger les poids (s_{ij}) selon leur appartenance ou non au cluster le plus rempli de l'espace de Hough $cl_H(I_1, I_2)$. On définit des poids corrigés $(\tilde{s}_{ij})_{1 \leq i \leq n_1, 1 \leq j \leq n_2}$ de la façon suivante :

$$\tilde{s}_{ij} = \begin{cases} s_{ij} & \text{si } T_{ij} \in cl_H(I_1, I_2) \\ 0 & \text{sinon} \end{cases} \quad (2.16)$$

On définit également une distance agrégée corrigée \tilde{D} :

$$\tilde{D}(I_1, I_2) = \min_{\tilde{s}_{ij}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \tilde{s}_{ij} d(z_{1,i}, z_{2,j}) \quad (2.17)$$

On notera pour la suite que pour éviter les effets de bords dûs à la quantification de l'espace des paramètres, il est nécessaire de procéder au remplissage de 2^N grilles de Hough, où N désigne le nombre de paramètres de la similitude. Pour chaque paramètre i , si on note Δ_i le

pas de quantification de la dimension correspondante de l'espace de Hough, il faut en effet tester deux grilles de quantification, l'une étant translatée de $\Delta_i/2$ par rapport à l'autre, et ne conserver pour l'estimation de la similitude finale que la grille de Hough présentant la cellule la plus remplie parmi les 2^N grilles.

La transformée de Hough généralisée entre deux ensembles de points peut être abordée de deux façons différentes :

- soit on réalise la transformée en première étape de l'algorithme de comparaison, auquel cas on accumule dans l'espace des transformations tous les appariements possibles entre un descripteur requête et un descripteur référence. On parlera dans ce cas de transformée de Hough *a priori*. Cette approche est illustrée par les figures 2.4 et 2.5.
- soit on ne fait voter dans l'espace des transformations que les appariements de descripteurs validant des conditions d'appariement local, par exemple un score d'appariement au delà d'un certain seuil. On parlera alors de transformée *a posteriori*, car l'information de cohérence géométrique globale calculée par la transformée de Hough est utilisée après l'information locale.

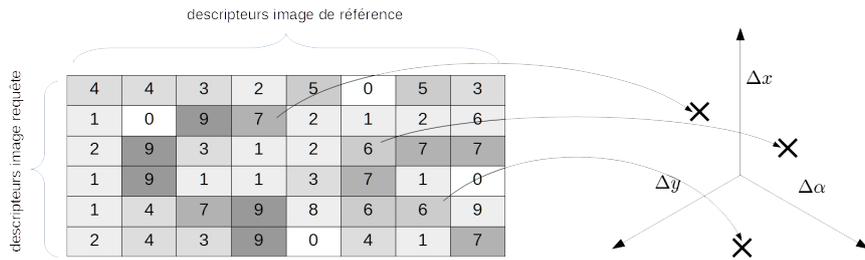


FIGURE 2.4 – On considère les distances entre appariements : tous les appariements entre descripteurs de l’image requête et de l’image référence votent dans l’espace des similitudes

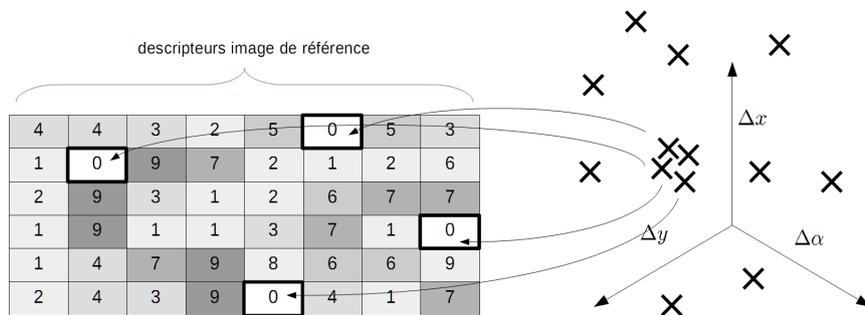


FIGURE 2.5 – Seuls les appariements ayant voté pour le principal cluster d’appariements dans l’espace des similitudes sont conservés pour la distance ou le score agrégé

Limites de la transformée de Hough

Heuristiquement, l’effet attendu de la correction des s_{ij} est une dégradation des distances agrégées D pour les couples d’images non matchantes, car en moyenne, seuls $\frac{n_1 \times n_2}{M}$ poids s_{ij} seront non nuls. Pour les couples d’images matchantes, on s’attend à ce que la majorité des appariements (p, q) vérifiant $\max_j d(z_{1,p}, z_{2,j}) = d(z_{1,p}, z_{2,q})$ vérifient également $T_{pq} \in cl_H(I_1, I_2)$

Cependant, si elle apporte une information essentielle de cohérence géométrique, la transformée de Hough généralisée est susceptible de poser plusieurs problèmes pouvant dégrader la fiabilité des algorithmes de comparaison.

1. Dans le cas d’une transformée de Hough *a priori*, si $n_1 \times n_2$ est grand par rapport au nombre de clusters de la quantification \mathbb{H} de l’espace des similitudes, alors les clusters

de la matrice CL vont être remplis par un "bruit de fond" ω lié au nombre important d'appariements possibles.

L'espérance de ω vaut $E(\omega) = (n_1 \times n^2)/M$ où M est le nombre de clusters de \mathbb{H} . Si I_1 et I_2 forment un couple d'images matchantes, l'espérance de remplissage du meilleur cluster $cl_H(I_1, I_2)$ vaut $n^1 \times n^2/M + n_1$

Si $n^2/M \gg 1$, $n_1 \times n^2/M \gg n_1$, et le rapport signal/bruit du remplissage du meilleur cluster de \mathbb{H} sera mauvais : la bonne similitude entre images matchantes ne sera pas discernable du bruit de fond.

La transformée de Hough a posteriori permet de résoudre ce type de problème, en réduisant le nombre d'appariements votant en les filtrant par les scores ou distances locales.

2. Si le nombre d'appariements locaux matchants entre deux images matchantes est intrinsèquement très faible, ce qui arrive si la requête est très bruitée ou si elle contient peu de descripteurs du fait de l'occultation, alors le meilleur cluster de Hough sélectionné ne sera pas forcément celui contenant la bonne similitude entre les deux images, même avec un faible bruit de fond.

Une solution à ce type de problème peut être de se passer de transformée de Hough quand le nombre de descripteurs locaux requêtes devient trop faible.

3. Les appariements locaux matchants entre deux images ne sont pas nécessairement ou systématiquement les appariements présentant le meilleur score ou la plus faible distance, des appariements fausses acceptances peuvent parfois présenter de meilleurs scores. Il s'agit là d'un défaut potentiel du descripteur utilisé ou de la distance associée, que la transformée n'est pas en mesure de corriger. Il se peut alors que les appariements validés par la transformée de Hough ne soient pas les meilleurs appariements, conduisant à un score agrégé plus faible ou une distance agrégée plus forte que si la transformée de Hough n'était pas utilisée.

La pertinence de la transformée de Hough dans ces cas-là dépend de l'arbitrage entre la dégradation des scores ou distances observés sur les couples d'images matchantes, et ceux observés sur les couples d'images non matchantes

4. Si les appariements locaux matchants ne sont pas les appariements présentant le meilleur score ou la plus faible distance, il existe en plus du problème précédent le risque qu'un filtrage des appariements votant pour la transformée de Hough élimine des appariements matchants. Dans ce cas, le cluster de Hough contenant la bonne similitude entre deux images matchantes risque également de ne pas être sélectionné car pas suffisamment voté.

On remarquera que le nombre d'appariements votant pour le calcul de $cl_H(I_1, I_2)$ dans la transformée *a posteriori* étant significativement plus faible que dans le cas *a priori* (n_1 au lieu de $O(n_1 \times n^2)$), la probabilité que cela arrive est significativement plus faible également.

Pour tenir compte de ces limitations, nous pouvons envisager plusieurs méthodes :

- ne faire voter que les appariements locaux au-delà d'un seuil fixe de score

Un seuil faible fera naturellement tendre les résultats vers ceux de la transformée *a priori*

- ne faire voter, pour chaque descripteur requête, qu'un seul appariement, celui correspondant au meilleur score ou à la plus faible distance

- variante de la méthode précédente, destinée à éviter les cas de figure où l'appariement matchant n'est pas celui présentant le meilleur score ou la plus faible distance : ne faire voter, pour chaque descripteur requête, que les n meilleurs appariements, avec n raisonnablement faible

Transformée de Hough avec seuillage

Une première approche consiste à ne faire voter dans la transformée de Hough que les appariements (i, j) vérifiant $d(z_{1,i}, z_{2,j})$ inférieur à un seuil δ_H , fixé empiriquement de manière à faire voter la majorité des appariements locaux matchants et à ne pas faire voter la majorité des appariements locaux non matchants. L'algorithme est détaillé comme suit (3) et illustré par les figures 2.6 et 2.7 :

Algorithm 3 transformée de Hough avec seuillage

Soit $\mathbb{H} = [cl_1, \dots, cl_M]$ une cluserisation de \mathcal{T} formée de M clusters
 Soit CL la matrice d'accumulation des transformations affines
 Soient $I_1 = (z_{1,i}, p_{1,i}, s_{1,i}, \theta_{1,i})_{1 \leq i \leq n_1}$ et $I_2 = (z_{2,i}, p_{2,i}, s_{2,i}, \theta_{2,i})_{1 \leq i \leq n_2}$ deux images avec leurs descripteurs locaux respectifs
for all $i \in [cl_1, \dots, cl_M]$ **do**
 $CL[i] = 0$
end for
for all $(p, q) \in [1, n_1] \times [1, n_2]$ **do**
 if $d(z_{1,p}, z_{2,q}) < \delta_H$ **then**
 $CL[h(T_{pq})] \leftarrow CL[h(T_{pq})] + 1$
 end if
end for
 $cl_H(I_1, I_2) = cl_{\hat{i}}$ tel que $\hat{i} = \arg \max_i CL[i]$

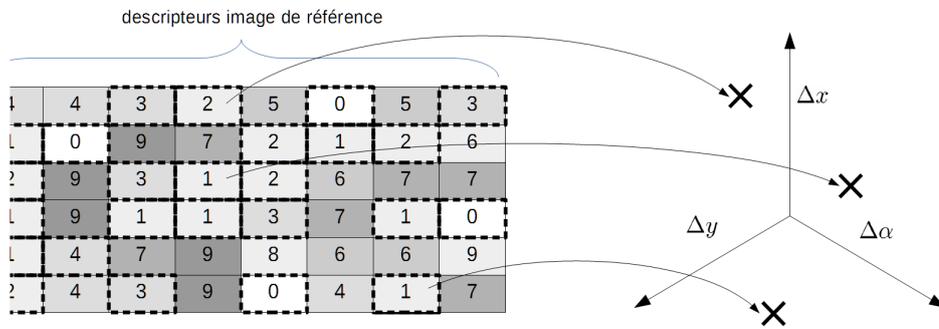


FIGURE 2.6 – Les appariements entre descripteurs de l'image requête et de l'image référence validant un seuil de distance (ici 3) votent dans l'espace des similitudes

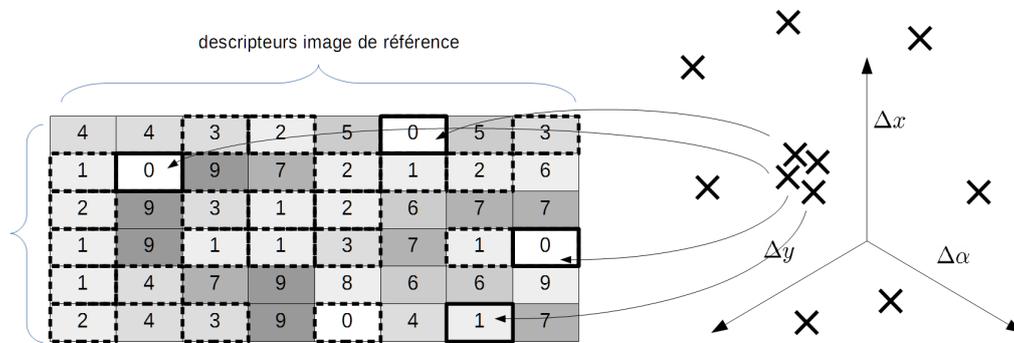


FIGURE 2.7 – Seuls les appariements ayant voté pour le principal cluster d’appariements dans l’espace des similitudes sont conservés pour la distance ou le score agrégé

Transformée de Hough avec meilleurs appariements

Une deuxième approche pour consolider la distance agrégée D avec la transformée de Hough consiste à ne faire voter dans l’espace de Hough que les appariements (p, q) de descripteurs locaux vérifiant $\min_j d(z_{1,p}, z_{2,j}) = d(z_{1,p}, z_{2,q})$. On définit toujours des scores corrigés (s_{ij}) et une distance agrégée \bar{D} de la même manière que définit en 2.16 et 2.17, mais le calcul de $cl_H(I_1, I_2)$ est modifié comme suit (5), et illustré par les figures 2.8 et 2.9 :

Cette variante de la transformée de Hough est cependant sensible au quatrième type de problème, c’est-à-dire à éliminer du vote dans l’espace de Hough les appariements matchants quand ceux-ci ne sont pas les meilleurs appariements.

Transformée de Hough avec N meilleurs appariements

Pour limiter le cas d’erreur potentielle évoqué dans la deuxième version, on propose une troisième version de la transformée de Hough, qui consiste à faire voter dans l’espace de Hough non seulement les appariements (p, q) de descripteurs locaux vérifiant $\min_j d(z_{1,p}, z_{2,j}) = d(z_{1,p}, z_{2,q})$, mais aussi, $\forall p$, les N appariements $(p, q_k)_{1 \leq k \leq N}$ correspondants aux N plus petites distances locales $d(z_{1,p}, z_{2,q})$, avec N fixé empiriquement en fonction du rang moyen r de l’appariement matchant dans la liste triée des distances locales. L’algorithme est détaillé comme suit (5) et illustré par les figures 2.10 et 2.11 :

Algorithm 4 transformée de Hough avec meilleurs appariements

Soit $\mathbb{H} = [cl_1, \dots, cl_M]$ une cluserisation de \mathcal{T} formée de M clusters
 Soit CL la matrice d'accumulation des transformations affines
 Soient $I_1 = (z_{1,i}, p_{1,i}, s_{1,i}, \theta_{1,i})_{1 \leq i \leq n_1}$ et $I_2 = (z_{2,i}, p_{2,i}, s_{2,i}, \theta_{2,i})_{1 \leq i \leq n_2}$ deux images avec leurs descripteurs locaux respectifs
for all $i \in [cl_1, \dots, cl_M]$ **do**
 $CL[i] = 0$
end for
for all $(p, q) \in [1, n_1] \times [1, n_2]$ **do**
 if $\min_j d(z_{1,p}, z_{2,j}) = d(z_{1,p}, z_{2,q})$ **then**
 $CL[h(T_{pq})] \leftarrow CL[h(T_{pq})] + 1$
 end if
end for
 $cl_H(I_1, I_2) = cl_{\hat{i}}$ tel que $\hat{i} = \arg \max_i CL[i]$

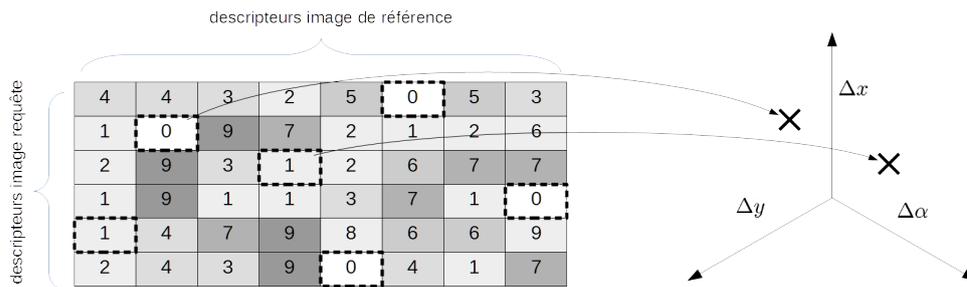


FIGURE 2.8 – Seuls les meilleurs appariements de chaque descripteur requête à tous les descripteurs de référence votent dans l'espace des similitudes

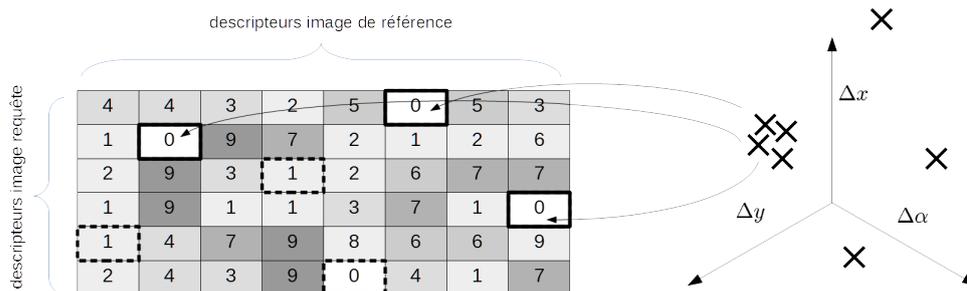


FIGURE 2.9 – Seuls les appariements ayant voté pour le principal cluster d'appariements dans l'espace des similitudes sont conservés pour la distance ou le score agrégé

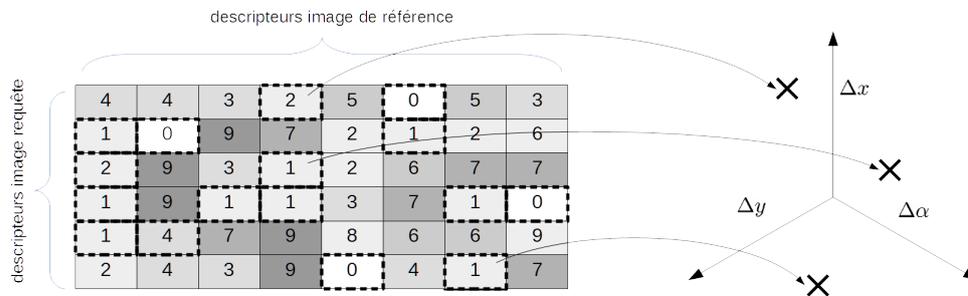


FIGURE 2.10 – Seuls les N meilleurs appariements de chaque descripteur requête à tous les descripteurs de référence votent dans l'espace des similitudes

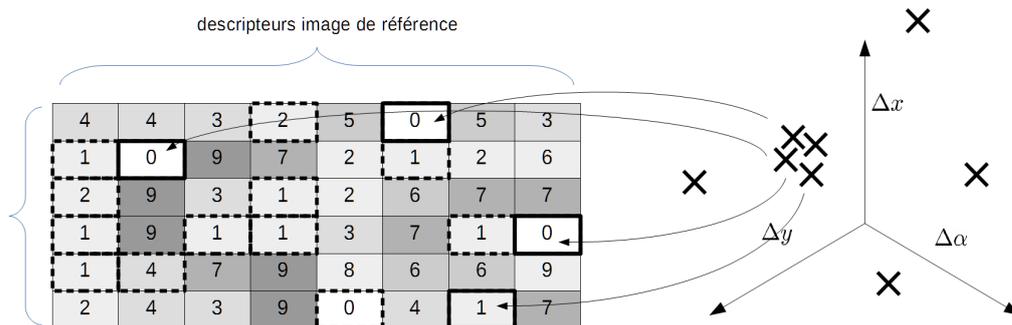


FIGURE 2.11 – Seuls les appariements ayant voté pour le principal cluster d'appariements dans l'espace des similitudes sont conservés pour la distance ou le score agrégé

Algorithm 5 transformée de Hough avec N meilleurs appariements

Soit $\mathbb{H} = [cl_1, \dots, cl_M]$ une cluserisation de \mathcal{T} formée de M clusters
Soit CL la matrice d'accumulation des transformations affines
Soient $I_1 = (z_{1,i}, p_{1,i}, s_{1,i}, \theta_{1,i})_{1 \leq i \leq n_1}$ et $I_2 = (z_{2,i}, p_{2,i}, s_{2,i}, \theta_{2,i})_{1 \leq i \leq n_2}$ deux images
avec leurs descripteurs locaux respectifs
for all $i \in [cl_1, \dots, cl_M]$ **do**
 $CL[i] = 0$
end for
for all $(p, q) \in [1, n_1] \times [1, n_2]$ **do**
 if $d(z_{1,p}, z_{2,q}) \in NS(z_{1,p})$ où $NS(z_{1,p})$ est la liste des N plus petites valeurs
 ($d(z_{1,p}, z_{2,j})_{1 \leq j \leq n_2}$) **then**
 $CL[h(T_{pq})] \leftarrow CL[h(T_{pq})] + 1$
 end if
end for
 $cl_H(I_1, I_2) = cl_{\hat{i}}$ tel que $\hat{i} = \arg \max_i CL[i]$

2.3.3 Représentation par graphes

La transformée de Hough généralisée utilise directement l'information de localisation spatiale des descripteurs locaux au sein des images à comparer. On peut cependant envisager une approche plus qualitative pour représenter les relations entre descripteurs locaux. Les structures de graphes sont particulièrement adaptées aux représentations d'images comme nuage de descripteurs locaux : chaque descripteur local peut être défini comme un noeud du graphe, et les arêtes du graphe permettent de définir les relations de voisinages, géométriques ou autres entre descripteurs locaux.

L'utilisation de graphes de voisinages entre objets extraits de l'image est ainsi proposée par [26] en imagerie médicale, développé notamment par [27], qui formalisent cette approche sous le terme d'*Attributed Relational Graphs* (ARG). La comparaison de deux représentations ARG d'images se fait généralement par la recherche d'un isomorphisme optimal entre sous-graphes des graphes issus des deux images, minimisant une fonction de coût. La résolution exacte d'un tel problème se fait au moyen d'algorithmes au coût exponentiel, notamment l'algorithme de Kuhn-Munkres ou algorithme hongrois.

Une approche similaire est également utilisée par [28], avec la technique de *comparaison élastique* (*elastic matching*). Dans cette méthode, utilisée précisément pour des empreintes digitales, la comparaison ne se fait pas par reconnaissance de structures topologiques communes entre les graphes, mais par déformation géométrique des graphes pour aboutir à un recouvrement des noeuds. L'algorithme procède par itération de deux étapes : un appariement deux à deux des noeuds du graphe par proximité géométrique, puis le calcul d'une déformation élastique minimale par TPS (*Thin Plate Spline*) destinée à faire se recouvrir les descripteurs appariés. Cette méthode a été enrichie dans les travaux de [29] et [30].

Cependant, de par leur coût très élevé en temps de calcul pour comparaison, les représentations par graphe sont difficilement envisageables pour des méthodes de CBIR où la rapidité d'exécution est une contrainte majeure.

2.3.4 Bag of Words

La reconnaissance d'image par agrégation directe de descripteurs locaux, si elle est plus complexe dans son implémentation que l'approche globale directe, reste une approche relativement simple pour franchir l'écart sémantique. En particulier, l'agrégation directe prend difficilement en compte les distributions des descripteurs locaux dans l'image.

Un autre type d'approche a été développé, qui ajoute une étape supplémentaire dans le franchissement de l'écart sémantique : l'approche par *Bag of Words*. Le Bag of Words consiste à revenir à une représentation globale de l'image non par construction directe d'un descripteur global unique, mais par regroupement de descripteurs locaux, afin de caractériser l'image par une *distribution* de descripteurs locaux plutôt que comme un ensemble non nécessairement ordonné de ces descripteurs locaux.

Clusterisation des descripteurs locaux

La construction de distributions de descripteurs locaux nécessite généralement la construction de classifications de ces descripteurs en catégories plus ou moins fines. Pour obtenir ces classifications on fait appel à des techniques d'apprentissage non supervisé, ou clusterisation, destinées à reconnaître dans l'ensemble des descripteurs locaux des motifs particuliers.

[8] distingue trois grandes catégories d'algorithmes de clusterisation :

- Clusterisation par appariement, par exemple la clusterisation spectrale
- Clusterisation fondée sur l'optimisation d'une mesure globale de qualité de clusterisation, par exemple le K-means
- Modélisation statistique : chaque cluster est considéré comme un motif généré par une distribution aléatoire, l'ensemble des descripteurs locaux étant la combinaison de ces distributions.

Une fois la clusterisation des descripteurs locaux obtenue, on peut construire la représentation globale de l'image sous la forme d'une distribution ou d'un histogramme de descripteurs. Là encore, plusieurs méthodes ont été développées pour construire ces représentations.

Bag-of-Words classique

L'attribution d'un descripteur local à un ou plusieurs clusters de l'espace des descripteurs est appelée phase de *codage* [31]. Les *codes* obtenus, c'est-à-dire les clusters d'attribution correspondants à chaque descripteur local, peuvent ensuite être agrégés facilement dans l'histogramme Bag-of-Words.

Il existe plusieurs méthodes de codage, parmi les plus courantes on trouve :

– **Hard vector quantization :**

Il s'agit de l'approche la plus simple, utilisée par [32]. Chaque descripteur local est attribué une et une seule fois, au cluster dont le centroïde est le plus proche. De manière formelle, le code du descripteur sur un ensemble de K clusters de centroïdes respectifs $(d_k)_{1 \leq k \leq K}$ se note $\alpha_1, \dots, \alpha_K$ avec $\forall i \in [1K]$,

$$\alpha_i = \begin{cases} 1 & \text{si } i = \arg \min_k d(x_i - d_k) \\ 0 & \text{sinon} \end{cases}$$

– **Sparse coding :**

Dans cette approche, on cherche à exprimer le descripteur x comme une combinaison linéaire d'un nombre restreint de centroïdes, à une faible erreur d'approximation près. [33] formalise l'approche ainsi :

$$\alpha = (\alpha_1, \dots, \alpha_K) = \operatorname{argmin}_{\alpha_0} L_i(\alpha_0, D) \triangleq \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_0$$

Pour un vecteur $x = (x_1, \dots, x_n) \in \mathbf{R}^n$ on note $\|x\|_0 = \operatorname{card}(x_i, 1 \leq i \leq n/x_i \neq 0)$. $\|\cdot\|_0$ est appelée la pseudo-norme l_0 . Un vecteur est dit k -parcimonieux s'il possède au plus k composantes non nulles, c'est-à-dire, si $\|x\|_0 \leq k$.

Un dictionnaire $D = [d_1, \dots, d_p]$ est un ensemble de vecteurs (ou *mots*) dans $\mathbf{R}^{n \times p}$. Un vecteur $y \in \mathbf{R}^n$ est k -parcimonieux pour le dictionnaire D s'il existe un vecteur k -parcimonieux x tel que $y = Dx$. Dans ce cas, y admet une décomposition parcimonieuse sur le dictionnaire D , avec un vecteur de coefficients parcimonieux x .

Les signaux réels sont rarement parfaitement parcimonieux, mais ils peuvent être *compressibles* : ils peuvent être correctement approximé par un autre signal, qui lui s'avère parcimonieux sur un dictionnaire bien choisi. Trouver une décomposition parcimonieuse d'un signal y sur un dictionnaire D signifie soit trouver le signal parcimonieux $\tilde{y} = Dx$, x k -parcimonieux, qui approxime le mieux y , ou bien, pour une erreur d'approximation maximale ϵ , trouver le signal $\tilde{y} = Dx$ le plus parcimonieux approchant y à cette erreur d'approximation près. Le premier problème peut être formalisé ainsi

$$\min_{x \in \mathbf{R}^p} \|x\|_0 \text{ tel que } \|y - Dx\|_2 \leq \epsilon \quad (2.18)$$

et le deuxième ainsi

$$\min_{x \in \mathbf{R}^p} \|y - Dx\|_2 + \lambda \|x\|_0 \quad (2.19)$$

La résolution de 2.18 est un problème NP-hard [34], cependant des algorithmes calculant des solutions approchées efficacement sont disponibles [35]. Une version relaxée de 2.19 peut être résolue par des méthodes de programmation linéaire standards :

$$\min_{x \in \mathbf{R}^p} \frac{1}{2} \|y - Dx\|_2^2 + \lambda \|x\|_1 \quad (2.20)$$

où λ est un paramètre de contrôle de la parcimonie de la solution.

Pyramide spatiale

La représentation en pyramide spatiale proposée par [36] à partir des travaux de [37] consiste à partitionner l'image en sous-régions de plus en plus petites enchâssées les unes dans les autres, en concaténant les histogrammes de descripteurs locaux localisés dans chaque sous-région. Cette représentation n'est pas parfaitement invariante aux transformations affines, car le canevas de découpage en sous-régions dépend de l'orientation de l'image. Cependant, elle intègre l'information globale de disposition mutuelle des descripteurs de façon bien plus efficace que le Bag-of-Word, permettant des performances en reconnaissance accrue dans de nombreuses situations.

Distances entre histogrammes

Les histogrammes de descripteurs locaux comme le BoW ou la pyramide spatiale se présentant sous la forme de vecteurs dans R^+ , des distances entre histogrammes simples comme la distance euclidienne ou des distances de Minkowski quelconques peuvent être envisagées.

Cependant, ces distances ne tiennent pas compte de la nature particulière des histogrammes, qui sont par nature des distributions de probabilité discrètes.

Parmi les distances entre histogrammes prenant en compte cette particularité, on peut citer à nouveau la distance de Mallows [21] et l'Earth-Moving Distance (EMD) [38], déjà introduite pour des distances entre ensemble de descripteurs locaux non ordonnées en histogrammes, mais aussi la divergence de Kullback-Leibler (KL) ou la distance de Bhattacharyya. Parmi ces distances, on notera que la distance EMD nécessite la construction d'une métrique entre les clusters associés à chaque coefficient de l'histogramme.

Le Bag-of-Words présente comme principal avantage de représenter une image par un unique objet mathématique, permettant des comparaisons entre images peu coûteuses en temps de calcul, en $O(1)$ plutôt qu'en $O(n)$ ou $O(n^2)$ avec n le nombre moyen de descripteurs par image, comme c'est le cas pour la majorité des méthodes de comparaison entre ensemble de descripteurs locaux.

Les principaux défauts de cette représentation sont :

- une perte potentielle en fiabilité de la reconnaissance de motif pour les descripteurs localisés en bordure de clusters
- un manque de robustesse à l'occultation de données
- la perte de l'information de localisation spatiale des descripteurs locaux entre eux

Le premier défaut peut être mitigé par l'utilisation simultanée de plusieurs jeux de clusters non recouvrants, et la combinaison des résultats associés, afin de réduire les effets de bord. Il peut l'être également par la distance de type EMD, pondérant les écarts entre histogrammes par la proximité entre les clusters.

Le deuxième défaut peut être lui mitigé par l'utilisation de distances ou pseudo-distances entre histogrammes mettant en valeur la quantité d'information partagée plutôt que les différences.

Le troisième défaut peut enfin être mitigé par l'utilisation de distances entre histogrammes susceptibles d'intégrer partiellement de l'information de localisation spatiale, ou par l'utilisation de descripteurs locaux de grande taille, c'est-à-dire construits sur une surface importante d'image, susceptible de contenir une information de cohérence semi-globale.

2.4 Accélération de la recherche

Si la construction d'une clusterisation de l'espace des descripteurs locaux est couramment utilisée pour franchir l'écart sémantique entre les descripteurs locaux et les modèles d'objet que l'on cherche à reconnaître en CBIR, la clusterisation peut également être mise à profit pour accélérer la reconnaissance d'image.

Une revue générale des méthodes de recherche accélérée de descripteurs locaux d'images a été proposée récemment par les travaux de [39].

Quel que soit le type de représentation choisi pour les images traitées, les méthodes présentées jusqu'à présent pour calculer une similarité ou une dissimilarité entre images nécessitent le calcul exhaustif explicite des distances entre la totalité des signatures des images requêtes et la totalité des signatures des images de référence. Ce parcours exhaustif est extrêmement coûteux dès que les bases de données croissent en taille, rendant un algorithme de filtrage exhaustif inutilisable. Pour conserver des temps de calcul raisonnables, qui ne croissent pas exponentiellement avec la taille des bases de données référencées, l'utilisation de méthodes non exhaustives est indispensable.

2.4.1 Application de la clusterisation aux représentations d'images

Cas de la représentation par ensemble de descripteurs locaux

Revenons sur le cas général du calcul d'une distance agrégée entre deux images I_1 et I_2 , avec $I_1 = \{(z_{1,1}, p_{1,1}), \dots, (z_{1,n_1}, p_{1,n_1})\}$, $I_2 = \{(z_{2,1}, p_{2,1}), \dots, (z_{2,n_2}, p_{2,n_2})\}$.

On rappelle la forme générale de la distance agrégée 2.1 :

$$D(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} d(z_{1,i}, z_{2,j})$$

Le calcul de cette distance entre une image requête I_s et les N images de la base de référence $(I_r)_{1 \leq r \leq N}$ nécessite, pour chaque descripteur local requête $z_{1,i}$, le parcours de l'ensemble de la base de données des descripteurs locaux de référence et le calcul explicite de la distance locale d à tous ces descripteurs.

Dans la plupart des cas, s_{ij} vaut 1 si $z_{2,j}$ minimise la distance $d(z_{1,i}, z_{2,j})$ et 0 sinon, c'est-à-dire en recherchant explicitement les meilleurs appariements de descripteurs locaux, ceux qui minimisent la pseudo-distance ou maximisent la similarité entre appariements. Parmi les variantes de la transformée de Hough généralisée, on peut également se limiter à faire voter dans l'espace des transformations affines les meilleurs appariements plutôt que la totalité des appariements possibles.

Lors de la soumission d'une image requête à l'algorithme d'identification, on peut donc envisager de ne calculer explicitement les distances et scores entre descripteurs locaux, que pour les descripteurs de références les plus proches des descripteurs requêtes.

Dès lors, on peut structurer la base de données des descripteurs locaux, de manière à pouvoir obtenir, pour chaque requête locale $z_{1,i}$, une liste de k références $[z_{r_1}, \dots, z_{r_k}]$, telle que les (z_{r_j}) constituent les k plus proches voisins de la requête $z_{1,i}$ selon la distance locale d , cela de manière certaine (*k-nearest neighbors*) ou approchée (*approximate k-nearest neighbors*).

L'intérêt de cette méthode, est qu'il est possible de récupérer explicitement les k plus proches voisins sans parcourir la base de descripteurs locaux de façon exhaustive. Le coût d'extraction des plus proches voisins peut être sous-linéaire par rapport à la taille N_r de la base de données de descripteurs locaux, ce coût étant souvent en $O(\log(N_r))$ quand le coût du parcours exhaustif est en $O(N_r)$.

En notant $knn(z)$ la liste des k plus proches voisins selon la distance locale d d'un descripteur local z dans la base de données des descripteurs locaux références, on peut alors remplacer la distance d entre descripteurs locaux dans la formule (2.1) par une distance approchée \hat{d} , définie par

$$\hat{d}(z_{1,i}, z_{2,j}) = \begin{cases} d(z_{1,i}, z_{2,j}) & \text{si } z_{2,j} \in knn(z_{1,i}) \\ \Delta & \text{n sinon} \end{cases} \quad (2.21)$$

où Δ est une distance arbitrairement grande fixée empiriquement

La formule (2.1) peut alors être réécrite avec une distance agrégée corrigée \hat{D} :

$$\hat{D}(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} \hat{d}(z_{1,i}, z_{2,j}) \quad (2.22)$$

La manipulation des cas où $\hat{d}(z_{1,i}, z_{2,j}) = \Delta$ peut s'avérer délicate : deux images assez semblables I_1 et I_2 peuvent se retrouver à une distance agrégée \hat{D} importante du fait de certaines valeurs de $\hat{d}(z_{1,i}, z_{2,j})$ à Δ . C'est pourquoi on préfère généralement utiliser les *scores* agrégés avec les algorithmes de *k-nearest neighbors*.

Ainsi, on remplace le score s entre descripteurs locaux par un score approché \hat{s} :

$$\hat{s}(z_{1,i}, z_{2,j}) = \begin{cases} s(z_{1,i}, z_{2,j}) & \text{si } z_{2,j} \in knn(z_{1,i}) \\ 0 & \text{n sinon} \end{cases} \quad (2.23)$$

La formule du score agrégé S peut alors être réécrite avec un score corrigé \hat{S} :

$$\hat{S}(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} \hat{s}(z_{1,i}, z_{2,j}) \quad (2.24)$$

L'ensemble des résultats proposés dans la section 2.3 restent valides avec les distances et scores approchés \hat{D} et \hat{S} .

On remarquera que la consolidation par transformée de Hough voit également son coût fortement réduit par l'utilisation des distances approchées car on ne fait voter dans l'espace de Hough que les appariements (i, j) tels que $z_{2,j} \in knn(z_{1,i})$

Cas de la représentation par Bag-of-Words

Dans le cas de la représentation par Bag-of-Words, une clusterisation de l'espace des descripteurs locaux a déjà été réalisée, pour construire l'histogramme de fréquence que constitue le Bag-of-Words. Cette première clusterisation a déjà pour but de fournir une représentation plus efficace à comparer qu'une représentation complète.

En assimilant l'histogramme Bag-of-Words à un descripteur local unique, on peut accélérer la recherche dans une base de représentation d'images par Bag-of-Words de la même façon qu'on accélérerait la recherche pour un unique descripteur local, tel que décrit plus haut : la distance D_{BoW} entre histogrammes est simplement remplacé par une distance approchée \hat{D}_{BoW}

$$\hat{D}_{BoW}(z^1, z^2) = \begin{cases} D_{BoW}(z^1, z^2) & \text{si } z^2 \in knn(z^1) \\ \Delta & \text{n sinon} \end{cases} \quad (2.25)$$

et le score S_{BoW} est remplacé par un score approché \hat{S}_{BoW}

$$\hat{S}_{BoW}(z^1, z^2) = \begin{cases} S_{BoW}(z^1, z^2) & \text{si } z^2 \in knn(z^1) \\ 0 & \text{n sinon} \end{cases} \quad (2.26)$$

2.4.2 Outils de clusterisation et d'accélération de la recherche

Toutes les méthodes de recherche non exhaustive de plus proches voisins reposent à des degrés divers sur l'application d'un système d'*index inversé* (*inverted file system*) à une clusterisation de l'espace des données. La clusterisation de l'espace des descripteurs permet d'associer à chaque descripteur une clé de codage correspondant à l'indice du ou des clusters auxquels il est attribué. Le système d'index inversé consiste à indexer la base des descripteurs en fonction de leur clé de codage. L'extraction des plus proches voisins d'une requête peut alors se faire en deux étapes :

- extraction de la clé de codage de la requête
- recherche en distance réelle des plus proches voisins restreinte aux descripteurs partageant la même clé de codage

Si le remplissage moyen d'un cluster sur une base de données est significativement inférieur aux nombres de descripteurs dans la base, alors la recherche par système d'index inversé sera plus rapide qu'une recherche exhaustive sur la base de données avec la distance exacte.

Une clusterisation dense de l'espace des descripteurs présente comme avantage un faible remplissage moyen des clusters, et donc une forte accélération de l'algorithme de recherche. Cependant, la probabilité que des descripteurs proches en distance exacte se retrouvent dans des clusters différents est accrue également, réduisant la précision de la recherche. Un algorithme de k-nearest neighbors présente donc un arbitrage entre accélération de la recherche et précision. Une heuristique couramment employée pour accroître la précision sans trop augmenter le temps de calcul consiste à visiter des ensembles de clusters voisins plutôt qu'un cluster unique, selon des critères propres à chaque algorithme. Le choix d'une heuristique efficace dans le choix de ces voisinages de clusters est le principal facteur déterminant l'efficacité d'un algorithme de k-nearest neighbors.

Kd-trees

D'après [40], un *arbre kd* ou *kd-tree* est la généralisation d'un *arbre binaire de recherche* destiné à la recherche des plus proches voisins d'une requête parmi un ensemble de données.

Un arbre binaire de recherche est une structure de données hiérarchique (un *graphe connexe acyclique* en théorie des graphes), dont chaque élément est appelé *noeud*. Les arêtes orientées de l'arbre binaire de recherche définissent une relation hiérarchique. Un noeud en amont d'une arête orientée est appelé *noeud père*, un noeud en aval d'une arête est appelé *noeud fils* ou *noeud successeur*. Chaque noeud a au plus deux fils. Un noeud n'ayant pas de fils est appelé *noeud terminal* ou encore *feuille*. Un noeud non terminal est appelé également *noeud interne*. Dans un arbre binaire de recherche, un seul noeud ne dispose pas de père, il s'agit de la *racine* de l'arbre. Tous les noeuds de l'arbre descendent de la racine par relations hiérarchiques successives.

La finalité d'un arbre binaire de recherche est l'indexation d'une base de données, afin d'accélérer le processus de recherche de la base en question. A chaque noeud d'un arbre binaire de recherche correspond un sous-ensemble de la base de données à indexer. La racine représente la totalité de la base de données. A chaque noeud est associé un sous-ensemble de la base de données, tel qu'à chaque fils d'un noeud, correspondent aux deux sous-ensembles issus du partitionnement binaire de l'ensemble de données associé au noeud considéré. L'ensemble des feuilles de l'arbre correspond à un partitionnement sans recouvrement de l'espace des données.

Dans le cas d'un espace de données unidimensionnel, chaque point (ou *enregistrement*) de l'espace de données est représenté par une *clé*. A chaque noeud non terminal de l'arbre kd est associé une *valeur*. Tout enregistrement appartenant au sous-ensemble associé à ce noeud, est également associé au noeud fils de gauche si sa clé est inférieure ou égale à la valeur du noeud, et associée au noeud fils de droite si sa clé est strictement supérieure à la valeur du noeud.

Quand l'espace de données est en dimension k , chaque enregistrement est représenté par k clés de codage. N'importe laquelle de ces clés peut être utilisée pour procéder au partitionnement de l'ensemble des enregistrements associés à un noeud donné. La clé choisie pour procéder au partitionnement est appelée le *discriminant*. L'arbre kd originel proposé par [41] choisit la clé associée à un noeud en fonction de la hauteur de ce noeud dans l'arbre binaire. Le choix de la clé est réalisé en bouclant dans l'ordre des clés.

[40] propose la construction de *kd-tree optimisé*, en choisissant la valeur et le discriminant de chaque noeud, ainsi que la profondeur de l'arbre, permettant de minimiser le temps de recherche moyen des plus proches voisins d'une requête dans la base de données.

A chaque noeud, on choisit comme discriminant la dimension présentant la plus forte variance des enregistrements associés à ce noeud. Heuristiquement, ce choix assure que la recherche de plus proche voisin dans le kd-tree ne devra procéder à l'exploration de plusieurs branches que pour des profondeurs faibles.

On choisit ensuite comme valeur du noeud la médiane des clés des enregistrements associés à ce noeud pour le discriminant choisi. Ce choix assure de construire un arbre binaire équilibré.

En notant N le nombre d'enregistrements dans la base de données, le processus de recherche du plus proche voisin de la requête nécessite 2^N comparaisons, et aboutit à une feuille unique de l'arbre, correspondant à un premier candidat au statut de plus proche voisin. On peut associer chaque feuille de l'arbre kd à une cellule de \mathbb{R}^k . La hauteur de l'arbre de recherche est 2^N . Le premier candidat n'est pas nécessairement l'enregistrement le plus proche de la requête :

la première recherche doit être complétée par un processus de *backtracking*, pendant lequel d'autres cellules doivent être également visitées à la recherche du meilleur candidat.

[40] démontre que l'utilisation d'un kd-tree optimisé permet d'attendre un coût de calcul moyen du plus proche voisin en $O(\log N)$, sous certaines conditions sur la distribution des enregistrements de la base de données, notamment dans le cas où les points de la base de données sont réparti aléatoirement.

Cependant, dans le cas où la dimensionalité de l'espace de données est élevée (typiquement de l'ordre de 10), le coût moyen de la recherche exacte du plus proche voisin explose, et tend rapidement vers le coût d'un parcours exhaustif de la base de données.

Des variantes de Kd-trees ont été proposées par [15] permettant le calcul approximatif des plus proches voisins en contenant le coût de recherche. [42] a montré que les Kd-trees représentent un bon arbitrage entre le temps de calcul du plus proche voisin, au prix d'un coût de stockage potentiellement élevé de la structure de données.

K-Means et Hierarchical K-Means

L'algorithme k-means est l'une des méthodes de clusterisation les plus utilisées en traitement de données. Pour un ensemble de points $(x_i)_{1 \leq i \leq n}$, l'algorithme tend à partitionner l'ensemble en k clusters $(S_j)_{1 \leq j \leq k}$ sous la contrainte de minimisation de l'erreur quadratique de quantification intra-cluster :

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

où μ_i est la moyenne des points de données contenus dans S_i .

L'algorithme standard de calcul d'une clusterisation par k-means est, étant donné un ensemble initial de centroïdes $\mu_1^{(1)}, \dots, \mu_k^{(1)}$ choisis aléatoirement dans l'espace ou selon une heuristique quelconque, la répétition successive de deux étapes :

- assignation : Chaque point de données est assigné au cluster de plus proche centroïde : les points de données sont partitionnés selon le diagramme de Voronoï induits par les centroïdes

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\}$$

- mise à jour : Pour chaque cluster on calcule un nouveau centroïde comme moyenne des points de données attribués à ce cluster

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

Il faut garder à l'esprit que cet algorithme ne fournit pas de garantie de convergence au sein de chaque cluster des sommes des carrés des distance au centroïde vers un minimum global. Le résultat dépend souvent fortement de la localisation initiale des centroïdes. L'algorithme ne fournit pas non plus de garantie sur l'équilibrage du nombre de points par cluster.

Le K-Means hiérarchique (*Hierarchical K-Means* - HKM) est une variante du K-Means d'origine, qui consiste à clusteriser récursivement les données à indexer. L'algorithme est caractérisé par un *facteur de branche* k . Lors de la première itération du HKM, les données sont

clusterisées en k clusters par le K-Means classique, puis le sous-ensemble de données contenus dans chaque cluster est à nouveau réparti en k sous-clusters par K-Means, et ainsi de suite. La structure finale de données est un *arbre de recherche*, dont les méthodes de parcours sont similaires à celles des Kd-trees, permettant de fouiller un ensemble de données de manière plus rapide que par parcours exhaustif.

Locality Sensitive Hashing

Le *Locality-Sensitive Hashing*, introduit par [43], est une méthode de clusterisation basée sur l'application d'un ensemble de fonctions de hachage sur les données à clusteriser, dont l'objectif est de maximiser le nombre de collisions sur les signatures, ou "buckets", des fonctions de hachage des points considérés heuristiquement comme "suffisamment proches". Cette méthode est sensiblement différente d'autres méthodes de hachage visant à éviter les collisions d'objets dissimilaires sur les mêmes signatures. Ainsi, le LSH est une méthode de clusterisation qui insiste sur l'évitement des faux rejets plutôt que des faux positifs.

Plus précisément, si on considère un espace métrique \mathcal{M} doté d'une distance d , un seuil R et un facteur d'approximation $c > 1$, une famille LSH \mathcal{F} est une famille de fonctions h telle que

$$\forall h \in \mathcal{M} \forall (p, q) \in \mathcal{M}$$

$$\text{si } d(p, q) \leq R, \text{ alors } Pr(h(p) = h(q)) > P_1$$

$$\text{si } d(p, q) \geq cR, \text{ alors } Pr(h(p) = h(q)) < P_2$$

Cela signifie que si deux points p et q sont suffisamment proches dans \mathcal{M} , alors ils entrent en collision avec une probabilité $> P_1$, et s'ils ne sont pas si proches ; ils tomberont sur des signatures de hachage différentes avec une probabilité $> 1 - P_2$. Une famille LSH est intéressante si $P_1 \gg P_2$, i.e., deux points semblables de \mathcal{M} tombent sur la même signature avec une plus forte probabilité que deux points dissemblables.

Parmi les différentes fonctions de hachage connues, on considère :

- le *bit sample*, où $h(x) = \tilde{x}[i]$, avec \tilde{x} la représentation de Hamming de x et i le i -ème bit de \tilde{x} sélectionné de manière aléatoire
- le ℓ_2 -LSH, où $h(x) = \lfloor \frac{\langle x, r \rangle + b}{w} \rfloor$, avec r un vecteur de projection aléatoire, b un offset aléatoire et w la largeur du bucket de la signature de hachage.

2.5 Application à un algorithme de filtrage d'empreintes digitales

Les différentes techniques de CBIR exposées ici ne sont pas toutes adaptées à tous les types de données, et a fortiori au cas qui nous intéresse, à savoir des empreintes digitales.

Si on reprend les contraintes de notre problématique, on peut vérifier si les différentes techniques exposées valident ces contraintes, et le cas échéant, on propose une application de la méthode adaptée à nos données.

Les contraintes sont les suivantes :

- robustesse au bruit
- robustesse aux occultations
- rapidité d'exécution en ligne
- coût de stockage raisonnable

Si les approches par descripteurs globaux ne sont pas adaptées à des images requêtes bruitées et occultées, les approches par représentation complètes sont adaptées à des images bruitées ou occultées. Cependant, toutes les distances agrégées ne sont pas adaptées à un coût de comparaison faible.

Historiquement, l'acquisition d'empreintes digitales s'est généralement faite sur des surfaces planes, feuilles de papiers ou capteurs optiques. L'acquisition d'empreintes digitales en trois dimensions, par capture photo ou vidéo de doigts sans contact avec un capteur, commence à se répandre, néanmoins à l'heure actuelle une part significative des bases de données d'empreintes digitales est toujours constituées d'acquisitions planes. Nous ferons donc ici l'hypothèse que les empreintes digitales analysées se présentent en deux dimensions, et à échelle unique.

Cette particularité signifie que la transformation affine superposant deux acquisitions d'une même image est restreinte à une rotation affine, si l'on fait abstraction des distorsions non linéaires de la surface du doigt.

La transformation affine optimale superposant au mieux deux empreintes digitales est donc restreinte à trois dimensions, ce qui en simplifie fortement le calcul par rapport à des algorithmes de reconnaissance d'images plus génériques.

Le calcul de cette rotation affine optimale s'avère un élément fondamental d'un grand nombre d'algorithmes de comparaison d'empreintes digitales, qui peut intervenir à différentes étapes de l'algorithme.

On suppose que l'on a construit un espace Z de motifs de descripteurs locaux z , doté d'une mesure de dissimilarité, ou pseudo-distance, d .

$$d : Z \times Z \longrightarrow \mathbf{R}^+ \quad (2.27)$$

Pour l'instant, nous ne supposons pas que Z soit un espace métrique ou euclidien ni que d soit une vraie distance.

Toute image I de l'ensemble F_i des empreintes digitales est caractérisée par un ensemble (z_i) de n^I descripteurs locaux.

Pour tout couple de descripteurs locaux (z_i, z_j) , on note $P_m(z_i, z_j)$ la probabilité que z_i et z_j soient extraits de la même région du même doigt

Soit $I_1 = \{z_{1,1}, \dots, z_{1,n_1}\}$ et $I_2 = \{z_{2,1}, \dots, z_{2,n_2}, p_{2,n_2}\}$ deux empreintes digitales dotées de leurs ensembles respectifs de descripteurs locaux, et n_1 et n_2 leur nombres respectifs de descripteurs locaux. On note I_1 l'image requête et I_2 l'image référence.

2.5.1 Distance agrégée

La distance IRM présente deux avantages : elle est relativement peu coûteuse à calculer, et elle reste valide en remplaçant la distance locale d par la distance approchée \hat{d}

On utilise une version non- symétrique de la distance IRM comme distance agrégée entre I_1 et I_2 . On définit D comme suit :

$$D(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \min_j d(z_{1,i}, z_{2,j})}{n_1} \quad (2.28)$$

On envisage une première heuristique sur cette distance agrégée : sachant que la majorité des $\min_j d(z_{1,i}, z_{2,j})$ sera faible entre deux images matchantes, et qu'elle sera élevée entre deux images non matchantes, on propose de corriger la fonction de distance locale d en une fonction $\tilde{d} = h(d)$, où h est une fonction de correction non linéaire qui réduit les faibles distances et accroît les fortes distances, jusqu'à un seuil de distance maximale.

On définit h comme suit :

$$h(d) = \begin{cases} \frac{(1 - \cos(\frac{d}{\alpha} \cdot \pi/2))^2}{2} & \text{si } d < \alpha \\ 1 - \frac{(1 - \cos(\frac{\beta-d}{\beta-\alpha} \cdot \pi/2))^2}{2} & \text{si } \alpha \leq d < \beta \\ 1 & \text{si } d > \beta \end{cases} \quad (2.29)$$

La fonction de correction est illustrée par la figure 2.12.

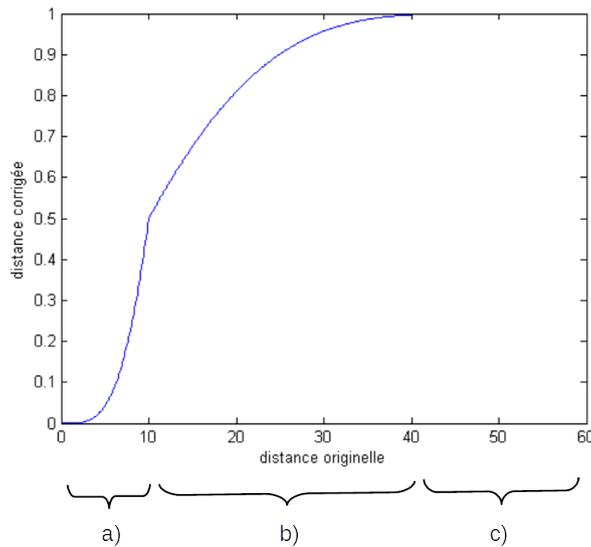


FIGURE 2.12 – Visualisation de la correction de distance : sur la section a), les faibles distances sont écrasées, sur la section b), le ratio aux distances de la section a) est accru, sur la section c), au delà d'un seuil de distance originelle, les distances corrigées sont toutes corrigées à une même valeur de distance standard arbitrairement forte

2.5.2 Score agrégé

De manière heuristique, lors de la construction de distances locales et de distances agrégées entre image, on cherche à séparer le plus possible les distributions de distances agrégées entre les

images matchantes et non matchantes. Cependant, une distance associée à un bon appariement ne peut pas être rendue plus petite que zéro. Pour accroître son éloignement aux appariements non matchants, il faut corriger la distance moyenne associée aux appariements non matchant.

Si on considère maintenant une *similarité* ou un *score* entre descripteurs locaux plutôt qu'une distance, on peut rendre le score associé à un bon appariement local arbitrairement grand, ce qui permettrait également de rendre un score agrégé entre images arbitrairement grand également dans le cas d'images matchantes.

L'utilisation de similarités plutôt que de pseudo-distances permet ainsi de focaliser l'optimisation de la fiabilité de l'algorithme de comparaison sur les appariements matchants, plutôt que sur les non matchants, beaucoup plus nombreux et présentant des cas beaucoup plus variés.

On définit un score de similarité local entre descripteurs comme suit :

$$\forall z_{1,i} \in I_1, \forall z_{2,j} \in I_2, s(z_{1,i}, z_{2,j}) = \frac{1}{1 + \alpha \cdot d(z_{1,i}, z_{2,j})} \quad (2.30)$$

On définit un score de similarité agrégé entre les images I_1 et I_2 comme suit :

$$S(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \max_j s(z_{1,i}, z_{2,j})}{n_1} \quad (2.31)$$

2.5.3 Score a contrario

Le concept de Probabilité de Fausses Alarmes proposé par [23] a été initialement conçu pour calculer une dissimilarité entre deux motifs locaux d'images. Cette approche reste cependant entièrement valide pour le cas où les s_i ne sont pas des sous-parties d'un motif local, mais où ils représentent des motifs locaux entiers, pour peu que l'on dispose d'une bijection entre deux ensemble de motifs locaux extraits de deux images à comparer.

Cette bijection est donnée par le calcul pour tout $z_{1,p}$ de l'image requête, d'un $z_{2,q}$ de l'image référence vérifiant $d(z_{1,p}, z_{2,q}) = \min_j d(z_{1,p}, z_{2,j})$

La probabilité de fausses alarmes entre les deux images I_1 et I_2 est alors donné par

$$FA = Pr(\min_j d(z_{1,1}, z_{2,j}) \leq \delta, \dots, \min_j d(z_{1,i}, z_{2,j}) \leq \delta, \dots, \min_j d(z_{1,n_1}, z_{2,j}) \leq \delta, |H_0) \quad (2.32)$$

sous l'hypothèse que les descripteurs locaux soient indépendants, on a alors

$$FA = \prod_i Pr(\min_j d(z_{1,i}, z_{2,j}) \leq \delta | H_0) \quad (2.33)$$

Or, $Pr(\min_j d(z_{1,i}, z_{2,j}) \leq \delta | H_0)$ est justement la fonction de répartition f de la distance entre meilleurs appariements de descripteurs locaux pour des couples d'images non matchantes, que l'on peut estimer expérimentalement

On définit alors un *score a contrario* agrégé S_{ac} de la façon suivante :

$$S_{ac}(I_1, I_2) = \sum_i -\log(f(\min_j d(z_{1,1}, z_{2,j}))) \quad (2.34)$$

où f est la fonction de répartition de H_0

Le score a contrario vérifie

$$\begin{aligned} S_{ac}(I_1, I_2) &= -\log\left(\prod_i f(\min_j d(z_{1,1}, z_{2,j}))\right) \\ S_{ac}(I_1, I_2) &= -\log(FA(I_1, I_2)) \end{aligned} \quad (2.35)$$

Indépendance des motifs locaux

L'équation 2.33 n'est valide que si les n_1 descripteurs de l'image requête I_1 sont indépendants.

Nous ferons l'hypothèse que les caractéristiques locales d'une empreinte digitale sont indépendantes, quand leurs zones d'extraction sont suffisamment éloignées. Cependant, étant donné la répartition des descripteurs locaux sur une empreinte, il peut arriver fréquemment que les surfaces d'empreinte encodés par deux descripteurs locaux se chevauchent, empêchant d'utiliser cette hypothèse.

Pour tenir compte de l'éventuelle non-indépendance des descripteurs locaux due à leur recouvrement dans le calcul du score a contrario, nous construisons une pondération des termes de l'équation 2.34

On considère la *matrice d'indépendance* $M = [m_{ij}]_{1 \leq i \leq n_1, 1 \leq j \leq n_1}$ définie par

$$\forall (i, j) \in [1, n_1]^2, m_{ij} = \begin{cases} 1 & \text{si } z_{1,i} \text{ et } z_j^1 \text{ se recouvrent} \\ 0 & \text{sinon} \end{cases} \quad \forall i, m_{ii} = 1$$

On construit alors le score a contrario corrigé de l'indépendance S_{acInd} :

$$S_{ac}(I_1, I_2) = \sum_i \frac{-\log(f(\min_j d(z_{1,1}, z_{2,j})))}{\sum_k m_{ik}} \quad (2.36)$$

A l'issue de ce chapitre, suite à une revue de l'état de l'art des principaux algorithmes de comparaison d'images, nous avons procédé à la sélection des algorithmes susceptibles de s'adapter à une recherche d'image avec requête de mauvaise qualité, permettant de remplir la contrainte de précision de la recherche de notre problématique. Les algorithmes choisis sont fondés sur des descripteurs locaux. Nous proposons plusieurs méthodes originales de calcul de similarités et dissimilarités entre images d'empreintes digitales à partir de descripteurs locaux dotés d'une pseudo-distance locale.

Nous allons détailler au chapitre 3 quels descripteurs locaux adaptés aux empreintes digitales nous allons construire, afin d'y appliquer les méthodes de comparaison que nous venons de décrire.

Ces algorithmes de comparaison sont susceptibles de s'adapter à des méthodes de recherche accélérée, qui permettront alors de remplir la contrainte de rapidité de la recherche, ce qui fera l'objet du chapitre 4.

Chapitre 3

Comparaison d'empreintes digitales : construction d'un algorithme en force brute

Les algorithmes de comparaison d'empreintes digitales peuvent être vus comme l'application à un type d'images particulier des méthodes générales décrites dans la précédente partie. Les caractéristiques propres aux images d'empreintes digitales vont naturellement restreindre les méthodes envisagées, ainsi que les types de représentation utilisées. Un aperçu exhaustif des différentes techniques d'analyse et de comparaison des empreintes digitales a été proposé par [4], dont nous reprendrons plusieurs notations et classifications.

Avant d'entrer dans une description plus précise des représentations d'empreintes digitales et des méthodes de comparaison associées, nous allons proposer un aperçu des principales méthodes d'extraction des informations de niveau 1 et 2 des empreintes digitales telles que décrite en section 1.1, c'est-à-dire respectivement d'une part les crêtes et vallées et d'autre part les minuties. Ces informations sont celles que nous utiliserons dans la suite de nos travaux, et nous nous concentrerons exclusivement dessus.

3.1 Extraction des données

Comme nous l'avons vu en première partie, les caractéristiques visuelles d'une empreinte digitale sont réparties en trois niveaux :

1. forme générale des crêtes et vallées
2. détails des crêtes, vallées et minuties
3. caractéristiques accessibles à résolution élevée, notamment les pores de sudation

En écartant de notre analyse les caractéristiques de niveau trois pour des raisons d'accessibilité des données expliquées en première partie, les informations de base les plus couramment utilisées par les algorithmes de comparaison d'empreinte digitales sont les suivantes :

- la localisation exacte des crêtes et des minuties
- l'orientation locale des crêtes
- les singularités (cores et delta)

Nous allons maintenant passer en revue les principales méthodes de l'état de l'art utilisées pour extraire ces informations de base à partir de l'image brute d'une empreinte digitale.

3.1.1 Crêtes et minuties

Les *minuties* sont les motifs les plus utilisés pour la comparaison d'empreintes digitales, depuis les travaux initiaux de [1]. Elles constituent la principale information de niveau 2 des empreintes digitales. Étymologiquement, le terme se réfère à des "petits détails". En pratique, il s'agit de la dénomination des discontinuités des crêtes d'un motif d'empreinte digitale. L'extraction des minuties d'une empreinte digitale suppose donc, simultanément, l'extraction d'une localisation suffisamment précise des crêtes et des vallées. La classification la plus générale des minuties consiste à distinguer les terminaisons de crêtes (*ridge ending*) et les croisements (*bifurcations*).

Deux types d'information sont utilisées pour la caractérisation des minuties dans la quasi-totalité des algorithmes de la littérature : la localisation spatiale en deux dimensions, et la direction de la minutie, définie sur $[0, 2\pi]$ comme étant l'orientation locale de la crête à laquelle elle appartient, dotée d'une direction fixée, de façon variable selon les conventions, par le côté vers lequel la crête s'étend ou bifurque.

Une troisième caractérisation de minutie couramment utilisée est la *classe*, qui définit la façon dont la crête définissant la minutie se termine ou bifurque. L'ANSI [44] définit 4 classes de minuties, le FBI [45] en définit 2. Dans la littérature en empreinte digitale, on peut cependant trouver plus que 4 classes, ainsi, jusqu'à 7 classes de minuties peuvent être couramment définies, par exemple les classes illustrées par la figure 3.1

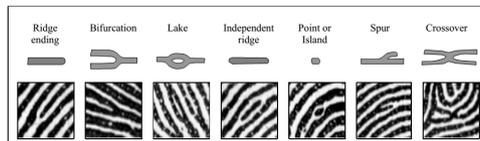


FIGURE 3.1 – Différentes classes de minuties répertoriées dans la littérature (tiré de[4])

Approches par binarisation

L'approche par binarisation de l'image d'empreinte est la plus répandue dans la littérature pour extraire les minuties d'une empreinte digitale [4]. Cette approche consiste à séparer explicitement au sein de l'image les zones de crêtes et les zones de vallées, puis à procéder à la squelettisation des zones de crêtes pour trouver les pixels terminaux et bifurcations, avec éventuellement une phase intermédiaire de débruitage et de correction de l'image binarisée.

L'utilisation d'un seuillage global pour la binarisation de l'ensemble de l'image d'empreinte n'est pas envisageable dans la grande majorité des cas, du fait de la variabilité parfois grande du contraste et de l'intensité au sein d'une même image. L'utilisation de seuils dynamiques, adaptés

localement, est nécessaire pour la très grande majorité des images. Le premier algorithme de détection automatique de minuties, reposant sur cette approche, a été proposé par [46].

[47] propose une méthode de binarisation de l'image d'empreinte par itérations successives d'un opérateur laplacien et un couple de seuils dynamiques : l'image initiale est fusionnée de façon itérative avec sa convolution avec un laplacien pour rehausser les contrastes, et des seuils dynamiques sont utilisés pour binariser progressivement les valeurs extrêmes à 0 ou 1. [48] propose une binarisation avec seuillage dynamique par fenêtre de voisinage local, de manière à équilibrer le nombre de 0 et de 1 dans chaque voisinage locale de l'image

[49] propose une sélection des pics d'intensité le long des directions orthogonales aux courbures locales, et une construction des crêtes binaires par sélection des deux pixels environnant le pic.

[50] propose de localiser les crêtes et vallées d'une image d'empreinte en calculant les maxima locaux et les points selles du hessien calculé en tout point de l'image.

[51] propose une approche où la surface de l'empreinte est considérée comme une réalisation bruitée d'une fonction continue sous-jacente de la surface du doigt. Cette fonction est approchée par des polynômes d'interpolation de Tchebychev. Crêtes et vallées sont ensuite extraites en utilisant le signe des maxima de la courbure normale de ces polynômes.

Une fois une représentation binarisée des crêtes et vallées obtenues, on calcule une squelettisation de l'ensemble des crêtes, par opérations topologiques classiques. Un grand nombre de méthodes existe pour régulariser l'image binaire et le squelette de l'empreinte digitale afin de lisser le bruit auquel le processus de squelettisation est extrêmement sensible. Une fois le squelette obtenu, on en déduit facilement la position et l'orientation des minuties, suivant la méthodologie décrite par [44].

Extraction directe

[52] propose une méthode de détection de crête reposant sur le suivi de directions orthogonales à l'orientation locale. A partir d'un point de départ sur une crête, une coupe de crête orthogonale à la courbure locale est déterminée. La direction de la crête est choisie comme la direction orthogonale à la courbure à partir du point de la coupe de la crête présentant la plus forte intensité. Cette direction est suivie le long d'un pas d'échantillonnage, jusqu'à un nouveau point de base, puis l'opération est réitérée, jusqu'au parcours intégral de la crête. Des variantes ont été proposées par [53],[54], [55], [56], [57]. Une variante notable a été également proposée par [28].

3.1.2 Orientation locale de crête

[4] définit l'orientation locale d'une crête en un pixel $[x, y]$ comme l'angle θ_{xy} qu'une crête forme avec l'axe horizontal de l'image, dans un voisinage arbitrairement petit de $[x, y]$. Les crêtes n'ayant pas de direction définie, θ_{xy} est un angle non orienté, compris entre 0 et π radians. Par la suite, on désignera sous le terme *orientation* un angle non orienté de crête compris entre 0 et π .

L'orientation des crêtes est une information suffisamment continue et régulière pour que la majorité des algorithmes de traitement d'empreinte digitale et d'extraction d'information ne

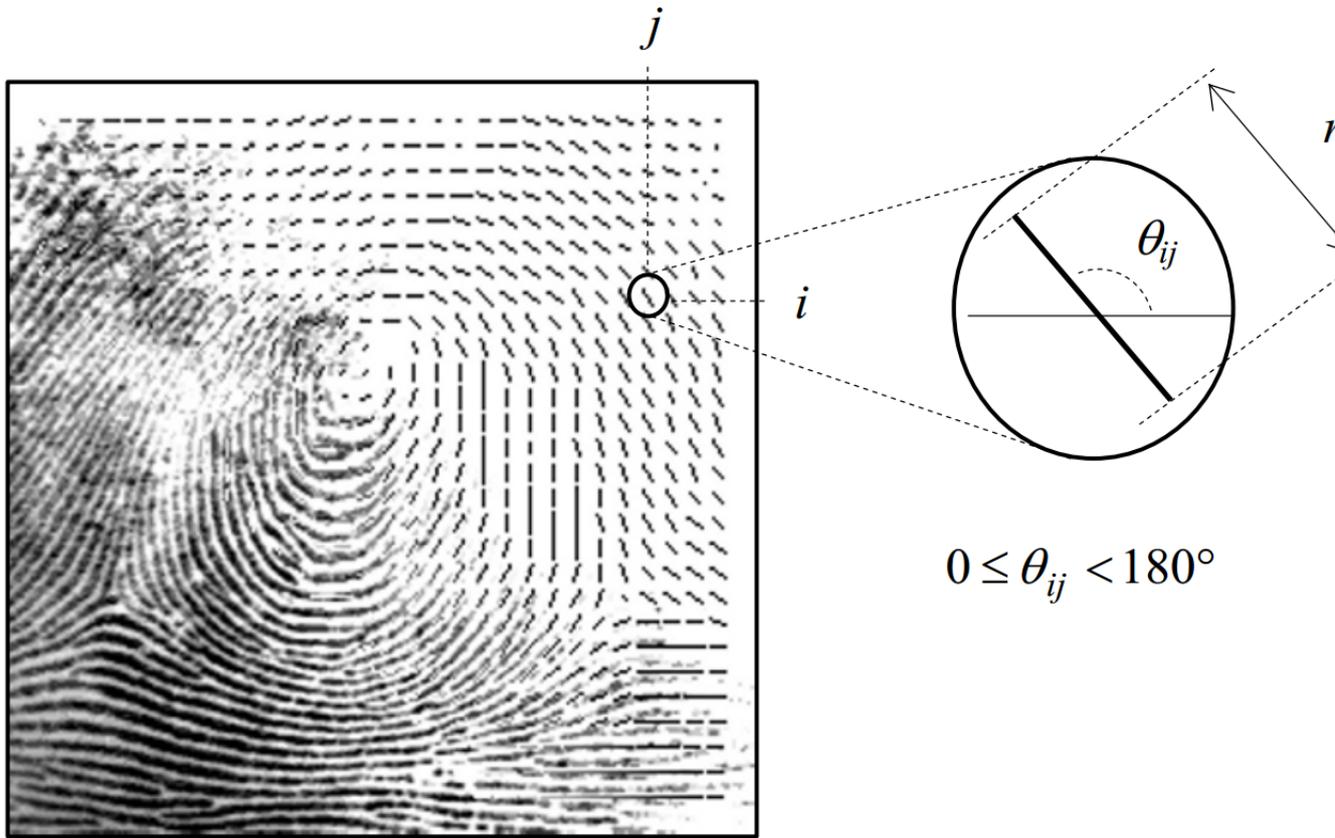


FIGURE 3.2 – Représentation courante de la carte d'orientation de crêtes d'une empreinte digitale (extrait de [4])

procèdent pas au calcul de l'orientation locale sur l'ensemble de l'image, mais seulement sur une grille de positions échantillonnées. Cela permet de réduire le coût de calcul de cette information, tout en laissant la possibilité d'estimer l'orientation locale en n'importe quel point par interpolation. [58] introduisit pour la première fois le concept d'*image d'orientation* ou *image de direction*, encore appelée *carte de flux* ou *carte d'orientation*. La carte d'orientation est une matrice $D = [\theta_{ij}]$ dont chaque élément θ_{ij} correspond à l'orientation locale au pixel (x_i, y_j) , les pixels (x_i, y_j) étant répartis sur l'image d'empreinte selon une grille d'échantillonnage régulière. Une valeur supplémentaire q_{ij} est souvent associée à chaque orientation locale θ_{ij} , qui correspond à une heuristique quantifiant la fiabilité du calcul de l'orientation locale au voisinage de (x_i, y_j) .

Approches par gradient

Le calcul de gradient local est la méthode la plus répandue pour l'estimation de l'orientation locale des crêtes. On peut citer parmi les implémentations notables [49], [59]. En assimilant une

image d’empreinte digitale I à un champ scalaire bidimensionnel $I(x, y)$, on peut calculer le gradient $\nabla I(x, y)$ de l’image :

$$\nabla I(x, y) = \begin{bmatrix} \nabla_x I(x, y) \\ \nabla_y I(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

La phase du gradient ainsi définie correspond à la direction de plus forte variation de l’intensité de l’image. L’orientation θ d’une crête qui traverserait le pixel $[x, y]$ est alors orthogonale au gradient local en $[x, y]$.

Les composantes ∇_x et ∇_y d’un gradient d’image peuvent être facilement calculées par application d’un masque de Sobel ou de Prewitt [60], et la phase peut être calculée comme $\arctan\left(\frac{\nabla_y}{\nabla_x}\right)$

Pour doter l’ensemble des orientations d’une structure d’espace euclidien, et procéder notamment à des moyennes d’orientation, [61] proposa de représenter l’orientation locale comme le *double* de l’orientation orthogonale à la phase du gradient.

Une orientation locale d peut alors s’écrire dans un espace euclidien comme

$$d = (\cos 2\theta, \sin 2\theta)$$

Une heuristique courante pour quantifier la fiabilité de l’orientation locale ainsi calculée a été proposée par [61] : pour l’ensemble des orientations locales d située dans une fenêtre d’estimation W , la fiabilité r de la fenêtre est donnée par

$$r = \frac{|\sum_W d|}{\sum_W |d|}$$

Approches par coupes et projections

L’approche par coupe a été suggérée dans les travaux de [46]. Le principe consiste à estimer une distribution des niveaux de gris le long d’un nombre prédéfini d’orientations-types. On s’attend à ce que les distributions soient très concentrées pour les orientations suivant une crête ou une vallée, et qu’elles soient le plus étalées pour les orientations orthogonales à l’orientation locale des crêtes. [62] propose de choisir l’orientation présentant la plus forte déviation standard comme orientation locale de crête.

[63] propose une variante de cette approche, qui consiste à calculer la distribution des niveaux de gris non pas uniquement le long d’une orientation d’un pixel de large, mais sur la projection de l’ensemble des pixels d’une fenêtre locale W le long de cette orientation

L’approche par gradient est moins coûteuse à calculer que l’approche par coupes et projection, et s’affranchit du problème de quantification des orientations locales. Cependant, les gradients ne fournissent qu’une seule information d’orientation locale en entrée, alors que les coupes peuvent fournir une distribution de probabilité de chaque orientation possible, ce qui peut ajouter une information utile pour la correction ultérieure de l’orientation sur l’ensemble de l’image.

Approches par banques d'orientations

L'estimation de l'orientation locale peut également se faire en fonction de la réponse locale de l'image à des filtres orientés. Ainsi [64] propose une estimation de l'orientation locale en fonction de la réponse de l'image locale dans le domaine *spatial* à une base de filtres de Gabor pré-orientées.

[65] propose une approche similaire, mais en cherchant la meilleure réponse de l'image à une banque de filtres dans le domaine des *fréquences*. Une variante de cette approche a été développée par [66], où l'on calcule la meilleure réponse à une banque de filtres pour des fenêtres purement locales de l'empreinte digitale.

3.1.3 Singularités du flux de crêtes

[67] propose une méthode simple et efficace pour la localisation des singularités du champ d'orientation des crêtes de l'empreinte digitale par l'index de Poincaré. Pour un champ vectoriel G et une courbe C plongée dans G , l'index de Poincaré $P_{G,C}$ est défini comme l'intégrale des rotations de G sur C . Cette méthode est illustrée par la figure 3.3 Dans le cas où G est un champ vectoriel discret, par exemple la carte d'orientation locale discrète D de l'empreinte digitale, et C un chemin défini comme une suite ordonnée de positions dans D , alors $P_{G,C}$ est défini comme la somme algébrique des orientations locales de crête le long des éléments de D Sur un chemin clos, l'index de Poincaré ne peut prendre que l'une des valeurs suivantes : $0, \pm\pi, \pm 2\pi$ radians. Ces valeurs permettent de définir les singularités de l'empreinte digitale.

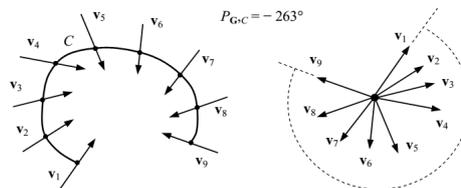


FIGURE 3.3 – Index de Poincaré calculé le long d'une courbe C plongé dans un champ vectoriel G (tiré de [4])

3.2 Descripteurs globaux d'empreintes digitales

Les représentations d'empreintes digitales par descripteurs globaux sont directement reliées aux approches globales pour les images en général telles que décrites dans la précédente partie. Dans l'approche globale, on cherche à représenter l'ensemble de l'information de l'image sous la forme d'une signature mathématique unique.

3.2.1 Corrélation d'images

La représentation globale la plus courante pour les empreintes digitales consiste à utiliser directement l'image d'empreinte elle-même, ou éventuellement une version rehaussée ou bina-

risée.

Sous l'hypothèse que l'image requête et l'image de référence à comparer sont à la même échelle, et que les transformations géométriques possibles entre les deux images peuvent être restreintes aux seules rotations affines en deux dimensions (ce qui est le cas pour les images d'empreinte acquises sur une surface plane), alors on peut définir une pseudo-distance entre deux images comme la somme des carrés des différences d'intensité pixel à pixel (*SSD* - sum of squared differences), à une rotation affine en deux dimensions près.

En notant T et I respectivement l'image de référence et l'image requête, on définit cette pseudo-distance *SSD* ainsi :

$$SSD(T, I) = \min_R \|T^\top - I \circ R\| \quad (3.1)$$

$$= \min_R \|T\|^2 + \|I\|^2 - 2T^\top I \circ R \quad (3.2)$$

où R parcourt l'ensemble des rotations affines.

On remarque que la pseudo-distance *SSD* est obtenue quand la corrélation croisée (CC) $T^\top I$ est maximisée. On peut alors également définir une similarité S entre T et I de la façon suivante :

$$S(T, I) = \max_{\Delta x, \Delta y, \Delta \alpha} T^\top I^{\Delta x, \Delta y, \Delta \alpha}$$

La rotation affine optimale peut être obtenue en force brute, en quantifiant l'espace des rotations affines et en y recherchant de manière exhaustive les paramètres de rotation maximisant la corrélation croisée entre les deux images. Cette méthode est cependant extrêmement coûteuse en temps de calcul [68], aussi des variantes de la méthode de corrélation globale ont été développée pour pallier ce défaut. On retiendra notamment le passage par l'espace de Fourier, puisque un produit de corrélation dans le domaine spatial est équivalent à un produit terme à terme dans le domaine de Fourier [60]. Le produit dans l'espace de Fourier présente également l'avantage d'être invariant aux translations, réduisant de fait la recherche exhaustive de la rotation affine optimale à l'espace des rotations. [69] et [70] proposent ainsi un produit dans l'espace de Fourier avec normalisation des transformées de Fourier avant produit terme à terme (*Symmetric Phase Only Filter*) :

$$T \otimes I = F^{-1} \left(\frac{F(T)}{|F(T)|} \times \frac{(F(I))^*}{\|(F(I))^*\|} \right)$$

[71] propose de raffiner la méthode en utilisant une transformée de Fourier-Mellin, qui produit une représentation indépendante à la fois de la translation et de la rotation de l'image.

Le produit normalisé dans l'espace de Fourier (*Phase-Only Correlation*) est repris par [72], combiné à un descripteur de Fourier-Mellin, et par [73] et [74], qui proposent de l'améliorer en sélectionnant dans la transformée de Fourier inverse du produit les fréquences des crêtes et

inférieures, c'est-à-dire les fréquences les plus porteuses d'information sur l'image d'empreinte digitale.

3.2.2 Comparaison de cartes d'orientation

Un des plus populaires des descripteurs globaux à base de carte d'orientation est le *FingerCode* de [75] : l'empreinte est caractérisée par un pavage circulaire, centré sur un core de l'empreinte. Dans chaque secteur de ce pavage, on calcule la valeur moyenne de la carte d'orientation locale au moyen d'une base de filtres de Gabor, la représentation finale étant un vecteur de taille fixe, permettant des calculs de distance simples entre représentation d'empreintes digitales. Les FingerCodes ainsi obtenus sont invariants aux translations mais pas aux rotations, aussi [75] propose de pré-calculer lors du codage un ensemble de rotations du FingerCode, et de conserver comme distances entre deux images la plus petite des distances entre tous les appariements de FingerCode.

Le concept de descripteur global centré et éventuellement orienté sur le core de l'empreinte a été plusieurs fois repris, notamment par [76] qui propose de construire des patches de carte d'orientation et de fréquence de crêtes autour du core.

Une variante récente du FingerCode a été proposée par [77], qui propose de combiner un score à base de FingerCode et un score utilisant des distances entre des patches de carte d'orientation centrées et orientées sur le core.

Autres approches globales

Diverses autres méthodes ont été développées pour fournir une représentation globale de l'empreinte digitale. Parmi les plus notables, nous retiendrons celle de [78], qui propose d'extraire de l'empreinte un repère absolu reposant sur la localisation d'un delta dans l'empreinte et d'une ligne de crête le traversant. La ligne de crête fournit alors un axe de pseudo-abscisse, avec une pseudo-ordonnée perpendiculaire, dotant les minuties d'un ordre absolu. Cet ordre absolu simplifie alors considérablement le processus de comparaison, qui se ramène à un algorithme de comparaison de deux séries ordonnées. Cette méthode est illustrée dans la figure 3.5.

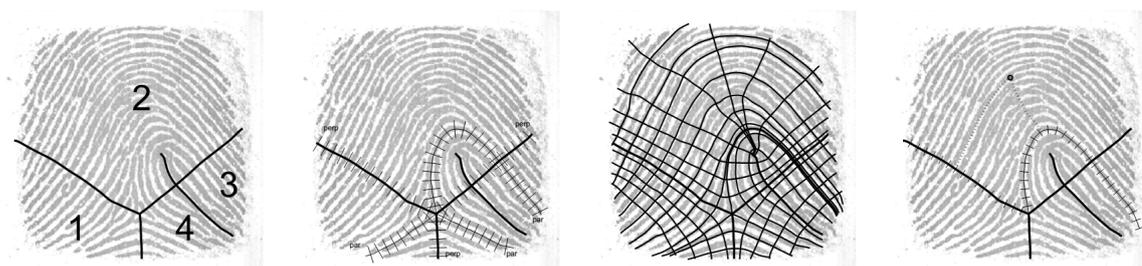


FIGURE 3.5 – Construction d'un système de coordonnées intrinsèques (tiré de [78]). De gauche à droite : découpage en régions, calcul des axes de référence du système de coordonnées, visualisation du quadrillage de l'empreinte dans le système de coordonnées intrinsèques, localisation d'une minutie dans le système de coordonnées

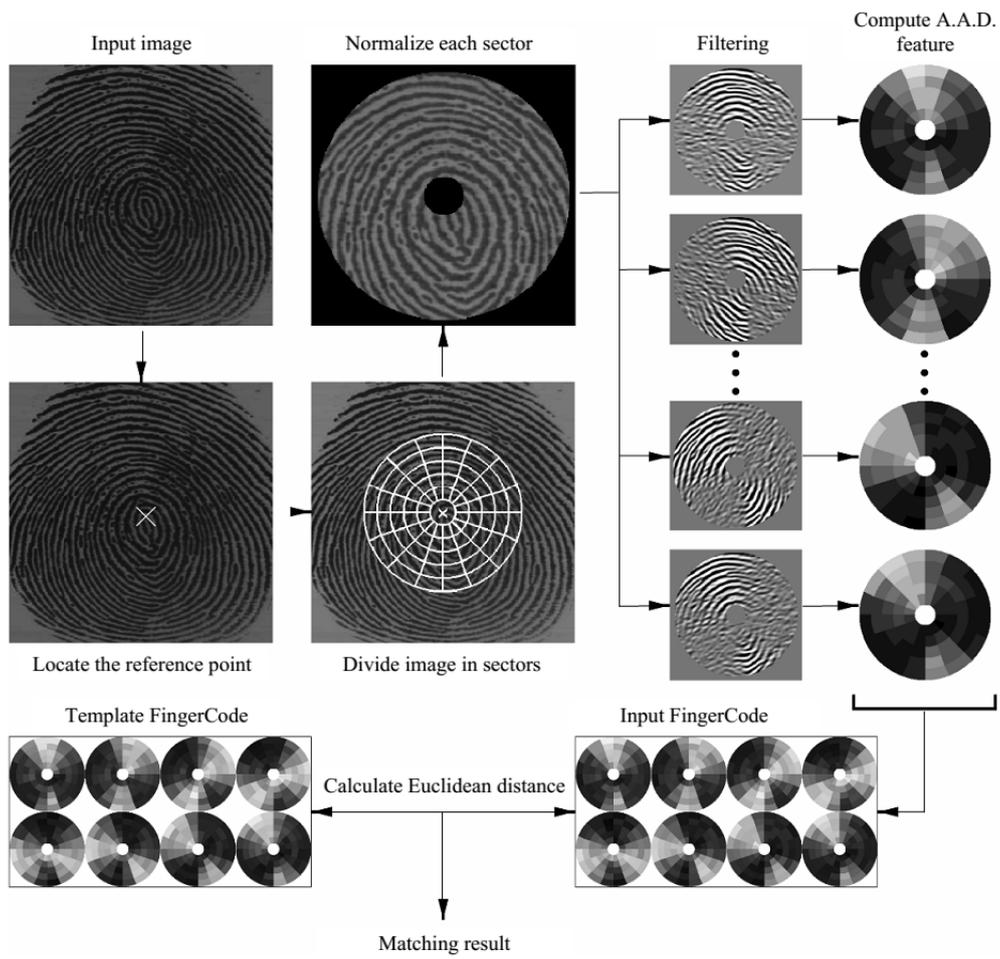


FIGURE 3.4 – Processus de construction du FingerCode (extrait de [75])

3.2.3 Analyse des approches globales

Les représentations par approches globales des empreintes digitales présentent des avantages et inconvénients directement liés aux avantages et inconvénients décrits pour les approches globales génériques dans notre précédente partie.

Les approches par corrélation d'images ou par comparaison de cartes d'orientation nécessitent souvent une phase d'optimisation de la superposition entre les images requête et référence, qui peut s'avérer très coûteuse en temps de calcul. Cette phase d'optimisation peut éventuellement être évitée ou réduite, à condition de disposer d'informations globales fiables dans les deux images, notamment la localisation et l'orientation des cores ou des deltas. Cette condition n'est pas compatible avec notre contrainte de robustesse à des images requêtes de mauvaise qualité de type latente, ou cette information sera peu fiable ou non disponible. Aussi, nous écarterons les représentations globales d'empreintes digitales pour la suite de nos travaux, cela en conformité avec nos conclusions du chapitre 2.

3.3 Descripteurs locaux d'empreintes digitales

Les représentations par descripteurs locaux pour empreintes digitales sont directement reliées aux approches locales pour les algorithmes de comparaison d'images génériques : l'image est représentée comme un nuage de points caractéristiques, stockant une information de nature locale, et pouvant être enrichi et ordonné par l'ajout d'une information globale.

3.3.1 Types de descripteurs locaux

Les descripteurs locaux d'empreintes digitales peuvent être regroupés en 5 grandes catégories :

- les structures locales de minuties, centrées sur une minutie
- les structures locales de minuties, non centrées (triplets et n-uplets de minuties)
- les patch de texture locale
- les descripteurs à base de crêtes
- les descripteurs hybrides

Structures locales de minuties centrées

[79] classe les structures locales de minuties en deux catégories, utilisant toutes les deux comme référentiel local une minutie centrale et son orientation :

a) Structures par plus proches voisins

Dans les structures par plus proches voisins, on définit le voisinage soit en terme de distance euclidienne, soit par des distances topologiques, par exemple les sommets reliés d'une triangulation de Delaunay de l'ensemble des minuties. Le principal avantage de ces structures est leur taille fixe, permettant des comparaisons locales rapides, comme des distances entre vecteurs. Le principal défaut de ces structures est leur grande sensibilité à l'occultation ou au déplacement de minuties par distorsion de la peau. Ce défaut oblige à restreindre l'utilisation

de ce type de descripteurs à des faibles distances entre minuties, ou à introduire des heuristiques coûteuses pour tenir compte des minuties déplacées.

Les descripteurs locaux à nombre fixe de voisins ont été introduits par [80], qui définissait un descripteur par une minutie centrale et ses deux plus proches voisins. Les distances entre minuties fournissent alors des caractéristiques du descripteur invariantes aux rotations affines. Les caractéristiques couramment retenues dans les différents algorithmes utilisant ces représentations sont :

- La distance à la minutie centrale de chacun des K plus proches voisins
- La différence d’orientation entre la minutie centrale et chacun des K plus proches voisins
- L’angle entre l’orientation de la minutie centrale et chacun des K segments joignant la minutie centrale à chacune de ses K plus proches voisines

D’autres caractéristiques peuvent compléter ces caractéristiques de base :

- Type de chacune des minuties, par exemple selon les classification [6]
- Comptage de crête le long de chacun des segments formés par la minutie centrale et chacun des K plus proches voisins

Des variantes ont été développées par [81], [82] qui étend le descripteur aux K plus proches voisins, par [83], qui propose de limiter les faux rejets dus aux effets de bords en dédoublant chaque K -uplet de rayon R par un K -uplet de rayon $R + r$, et plus récemment par [84].

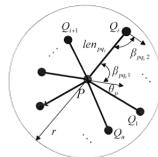


FIGURE 3.6 – Le descripteur minutie centré de [83]. Le descripteur est caractérisé par les caractéristiques $(len_{PQ_i}, \beta_{PQ_{i1}}, \beta_{PQ_{i2}}$. Les minuties Q_1, \dots, Q_n sont triées de 1 à n selon leur distance len_{PQ_i} à la minutie centrale P . Les angles $\beta_{PQ_{i1}}$ et $\beta_{PQ_{i2}}$ correspondent respectivement aux angles de PQ_i avec l’orientation de la minutie centrale P et de la minutie Q_i avec PQ_i

Pour restreindre le nombre potentiellement très élevé de minuties voisines quand K augmente, [85] propose de sélectionner les minuties les plus proches parmi les sommets liés d’une triangulation de Delaunay.

b) Structures à rayon fixe

Dans les structures à rayon fixe, les voisins de la minutie centrale sont définis comme toutes les minuties situées dans un rayon fixe ρ autour de la minutie centrale. Le nombre de descripteurs étant variable, le processus de comparaison entre descripteurs locaux est plus complexe que pour des descripteurs par plus proches voisins, mais aboutit également à une plus grande robustesse aux données bruitées ou occultées.

Les descripteurs à rayon fixe ont été introduits par [86], qui propose de construire un graphe orienté de minuties, où chaque minutie se voit dotée d'un ensemble d'arêtes vers les minuties voisines dans un rayon fixe. Ces descripteurs ont été réactualisés par [79], qui propose le *Minutiae Cylinder Code* (MCC) un descripteur centré et orienté sur chaque minutie, invariant aux rotations affines, et stockant simultanément, sous forme d'un cylindre quantifié, la localisation en deux dimensions et l'orientation des minuties dans un rayon fixe, permettant une distance vectorielle entre descripteurs. Le même type de descripteur est proposé par [87], mais sous une forme de listes de minuties voisines, nécessitant une distance plus complexe en calcul que le MCC.

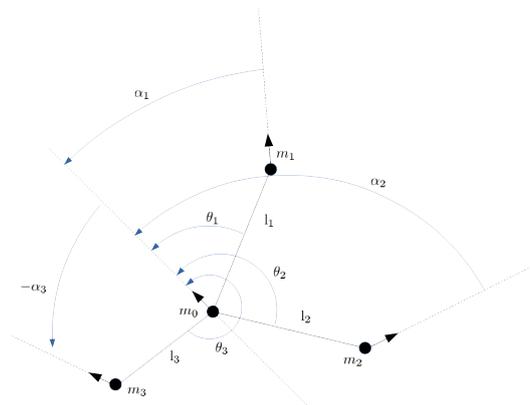


FIGURE 3.7 – Représentation d'une structure locale de minutie : les minuties m_1, m_2, m_3 sont définies comme voisines de m_0 , les paramètres $l_1, l_2, l_3, \alpha_1, \alpha_2, \alpha_3, \theta_1, \theta_2, \theta_3$ caractérisent le descripteur local

Structures locales de minuties non centrées - triplets de minuties

Les triplets de minuties forment un autre type de descripteur à base de voisinages de minuties, mais contrairement aux structures à K plus proches voisins décrites ci-dessus, ils ne sont pas définis autour d'une minutie centrale, mais directement comme un ensemble de triplets voisins. Le concept de triplet a été développé par [88] qui les introduit comme descripteur local de référence. Des variantes notables de triplets ont été proposées depuis par [89], [90], [91], [92], [93], [94], [95]. Les principales différences entre ces variantes reposent sur le type de caractéristiques géométriques utilisées pour caractériser les triplets.

De la même manière que pour les descripteurs à plus proches voisins minutie-centrés, des restrictions du nombre de triplets générés ont été proposées, en restreignant le choix des triplets voisins aux sommets d'une triangulation de Delaunay [89]. Pour plus de robustesse de la triangulation aux occultations et déplacements de minuties, des triangulations de Delaunay d'ordre k ont également été proposées [96] [97]. Le concept des triplets a également été élargi à des n-uplets de minuties : [98] propose ainsi des quadruplets de minuties.

Patch de texture locale

La texture de l’empreinte digitale, représentée directement en niveaux de gris, ou bien sous forme de carte d’orientation locale ou sous d’autres formes, peut être utilisée comme descripteur global de l’image, en utilisant un repère absolu comme un core ou un delta. Elle peut également être utilisée comme descripteur local en découpant des patchs de textures autour de points caractéristiques de l’empreinte digitale, minuties ou autres.

Ainsi, [99] propose de découper des patchs locaux de niveaux de gris, de taille variable, centrés et orientés sur les minuties, dans les zones où l’orientation locale des crêtes est la moins *cohérente*, c’est-à-dire avec une forte variance des orientations locales (mesurée par exemple via des histogrammes locaux d’orientation), ou encore dans les zones présentant la plus faible corrélation avec le reste de l’empreinte, c’est-à-dire les zones les plus susceptibles de caractériser l’empreinte dans sa totalité. Ce type de patchs a été proposé par [100], en niveau de gris, et exclusivement centrés sur les minuties.

En plus de patchs extraits de l’image brute de l’empreinte, des patchs d’orientation locale du flux de crêtes ont également été proposés par [101]. Dans cette approche, les descripteurs locaux sont fournis par les valeurs de l’orientation locale des crêtes calculées sur deux cercles de points régulièrement espacés autour d’une minutie centrale fournissant origine et orientation. Des variantes de l’échantillonnage pour cette méthode ont été proposées par [102] et [103]. [104] propose une autre variante pour orienter les patchs, en utilisant la moyenne des orientations locales échantillonnées plutôt que l’orientation de la minutie centrale. Enfin, [105] propose de réutiliser directement la structure du FingerCode autour de chaque minutie pour construire des descripteurs locaux.

Le descripteur SIFT de [14], très utilisé en reconnaissance d’images naturelles, est utilisé comme descripteur par [106], qui propose un descripteur SIFT adapté à l’empreinte digitale, centré et orienté sur les minuties et utilisant la carte d’orientation comme information de base dans la construction des histogrammes locaux sur lesquels le descripteur est basé. Le SIFT d’orientation locale est également proposé par [107], cette fois en centrant les SIFT en 4 points régulièrement disposés autour de chaque minutie.

Descripteurs à base de crêtes

Si la méthode d’extraction des informations de base de l’image d’empreinte permet leur localisation de manière suffisamment précise, les crêtes peuvent également constituer une base intéressante de descripteurs locaux.

[108] propose ainsi d’ajouter à chaque minutie une description de sa crête d’appartenance. On discrétise la crête à intervalles réguliers, et on projette les points discrétisés sur l’axe donné par l’orientation de la minutie, donnant une représentation de la crête comme vecteur de longueur variable. Une distance entre crête appariée peut alors être calculée comme l’erreur sur la longueur commune aux deux crêtes. Une approche similaire est utilisée par [109] et [110] pour enrichir des triplets de minuties.

[111] utilise les crêtes comme descripteur autonome. Chaque crête est caractérisée par sa longueur et sa courbure en des points régulièrement espacés. L’erreur en longueur et courbure est utilisée comme distance pour appairer les crêtes, puis la topologie des crêtes (placement

relatif des crêtes entre elles) est utilisée pour vérifier la cohérence des appariements estimés.

Descripteurs hybrides

D'autres types de descripteurs, non classables dans les précédentes catégories ou bien croisant les caractéristiques de plusieurs catégories, ont été également développés. Parmi ceux-ci, [112] propose de calculer toutes les distances entre deux minuties d'une empreinte. La probabilité d'apparition de chaque longueur peut être estimée sur une base d'apprentissage, et les segments sont triés par probabilité d'apparition. La distance finale entre images peut ensuite être calculé comme une distance d'édition entre deux listes triées. [113] propose un descripteur de type triplet de minuties auquel est combiné une information de texture de l'image de base, en codant chaque triplet par 4 histogrammes de niveaux de gris, calculés sur les 4 cases d'une grille centrée et orientée sur chaque triplet.

3.3.2 Méthodes de comparaison

Comme pour les algorithmes génériques de comparaison d'images, les algorithmes de comparaison entre deux empreintes digitales par descripteurs locaux comportent deux grandes étapes : une étape de comparaison des descripteurs locaux et une étape de vérification de la cohérence globale des appariements. Pour rappel, nous avons fait l'hypothèse dans nos travaux que nous travaillons sur des images à échelle constante acquise sur une surface relativement plane, donc que les similitudes considérées entre images sont des transformations affines. La vérification de la cohérence générale est le plus souvent obtenue par le calcul explicite de la similitude optimale superposant au mieux les descripteurs locaux de l'image requête et de l'image référence, mais elle peut aussi être vérifiée par d'autres méthodes qui seront décrites par la suite.

La vérification de la cohérence globale peut avoir un impact significatif sur le temps de comparaison entre images, et son utilisation peut nécessiter des structures de données particulière. Aussi, suivant l'ordre et la nature des deux étapes décrites de comparaison locale et de cohérence globale, on peut classer les algorithmes de comparaison par descripteurs locaux dans les catégories suivantes :

- les algorithmes utilisant uniquement les appariements locaux
- les algorithmes utilisant une orientation géométrique intrinsèque de l'empreinte digitale
- les algorithmes procédant à un alignement géométrique des deux ensembles de descripteurs locaux à chaque comparaison
- les algorithmes utilisant une représentation par graphes

Méthodes utilisant uniquement les appariements locaux

Quand les descripteurs locaux sont intrinsèquement très discriminants entre les motifs locaux effectivement issus de la même zone du même doigt et les motifs locaux fausses acceptances, ou bien quand ils sont de taille suffisante pour contenir une certaine quantité d'information globale, il n'est pas forcément nécessaire d'ajouter une contrainte de cohérence géométrique globale entre les appariements locaux.

Les scores ou distances agrégés entre image peuvent être directement dérivés des scores et distances génériques décrites au chapitre 2. La distance agrégée de choix est alors la distance de Hausdorff [80]. Si la cohérence globale n'est pas prise en compte, une représentation de type Bag-of-Words peut aussi être envisagée, et dans ce cas les distances utilisables sont les distances entre histogrammes décrites au chapitre 2 également [110] [98] [114] [84] [97].

Méthodes avec orientation intrinsèque

Les méthodes avec alignement intrinsèques peuvent être de deux types :

- les méthodes utilisant un repère intrinsèque à l'empreinte
- les méthodes ramenant la représentation de l'empreinte à une seule dimension

Ces méthodes de pré-alignement peuvent être absolues (un seul repère géométrique intrinsèque par image), telles que développées par [7] qui cherche à orienter les empreintes digitales autour du core, ou par [59] qui cherche à les orienter autour de toutes les singularités. Elles peuvent également être recalculées pour chaque requête, afin d'améliorer la fiabilité de la reconnaissance. Ces méthodes ont été développées notamment par [115] [116], avec orientation autour des singularités, par [117] à partir de la carte d'orientation, ou par [108] à partir des crêtes, indexées par une minutie d'origine pour chaque crête.

Méthodes avec alignement géométrique

Les méthodes d'appariement de points caractéristiques les plus courantes sont les mêmes pour la comparaison d'empreintes digitales et d'images générales : il s'agit essentiellement de la transformée de Hough généralisée avec ses variantes, et de l'algorithme hongrois.

La transformée de Hough généralisée, introduite pour les méthodes de CBIR par [118], a été utilisée en reconnaissance d'empreinte digitale depuis les travaux de [119]. La méthode est restée relativement inchangée depuis, avec des variantes destinées à accélérer le parcours de l'espace des paramètres, en coarse-to-fine [120] ou hiérarchique [121], ou des variantes sur le type de descripteurs utilisés dans la transformée, sur les doublets de minuties plutôt que les minuties elles-mêmes par exemple [122].

Un calcul explicite des paramètres de la rotation affine optimale est envisageable, soit par la méthode générale de CBIR de [15] décrite au chapitre 2, soit par des variantes développées spécifiquement pour l'empreinte digitale : [80] utilise ainsi les deux appariements de minuties présentant le meilleur score d'appariement local, et [123] utilise le meilleur score d'appariement de deux doublets de minuties.

[122] propose une variante de la transformée de Hough généralisée, très utilisée en empreinte digitale. I_1 et I_2 sont deux images à comparer caractérisées par leurs descripteurs locaux respectifs (m_1, \dots, m_M) et (m'_1, \dots, m'_N) .

Chaque descripteur est caractérisée par un jeu de paramètres géométriques (x, y, θ) , correspondant respectivement à sa localisation spatiale en deux dimensions et à son orientation dans le repère image.

$\forall (x_u, y_u, \theta_u), (x_v, y_v, \theta_v) \in I_1 \times I_2$, il existe une rotation affine unique f_{uv} telle que $(x_v, y_v, \theta_v) = f_{ij}(x_u, y_u, \theta_u)$.

A $\{u, v\}$ donné, les descripteurs locaux de paramètres respectifs (x_i, y_i, θ_i) et (x_j, y_j, θ_j) dans $I_1 \times I_2$ forment un appariement compatible avec $\{u, v\}$ si $(x'_i, y'_i, \theta'_i) = f_{ij}(x_i, y_i, \theta_i)$ vérifie $|x'_i - x_j| < \delta_{xy}$, $|y'_i - y_j| < \delta_{xy}$ et $|\theta'_i - \theta_j| < \delta_\theta$, où δ_{xy} et δ_θ sont des seuils de tolérance à la distorsion.

Pour chaque appariement entre un descripteur local requête et un descripteur référence, on calcule la rotation affine associée, et on compte le nombre d'appariements valides entre les deux images à comparer pour cette rotation. La rotation affine obtenant le plus d'appariements valides est alors considérée comme la rotation affine optimale.

En appliquant cet algorithme à l'empreinte digitale, [122] utilise comme descripteurs les minuties, mais préfère utiliser les paramètres de localisation et d'orientation en coordonnées polaires. Les seuils de tolérance à la distorsion sont ainsi calculés en distorsion polaires, accordant une plus grande tolérance à la distorsion aux couples de minuties les plus éloignées les unes des autres, ce qui est plus cohérent avec la nature des empreintes digitales.

Deux types de scores, construits à partir des descripteurs locaux effectivement appariés lors du calcul de la rotation affine optimale, sont couramment utilisés dans les algorithmes de comparaison d'empreintes digitales. En notant k le nombre d'appariements d'une minutie de I_1 à une minutie de I_2 tombant dans le cluster de Hough le plus rempli, et n et m les nombres respectifs de minuties de I_1 et I_2 , on considère les scores suivants :

$$s_1 = \frac{k}{(n + m)/2}$$

$$s_2 = \frac{k^2}{n \times m}$$

Ces scores peuvent être interprétés respectivement comme une moyenne arithmétique et une moyenne géométrique du nombre de minutie de I_1 disposant d'une minutie appariée dans I_2 validée par la cohérence géométrique.

On notera que le fait de normaliser le nombre d'appariements valides par la moyenne arithmétique ou géométrique du nombre total de descripteurs locaux requêtes et référence fait sens pour comparer deux empreintes digitales de même type (enroulé contre enroulé par exemple), mais dans le cas de types dissymétriques (latente contre enroulé, qui est une des situations directement concernées par notre problématique), il peut devenir judicieux de ne pas moyenniser le dénominateur de la normalisation, et de normaliser par le seul nombre de descripteurs requêtes disponibles sur l'image latente, afin de conserver des dynamiques de scores ou de distances comparables entre appariements d'images avec requête latente pouvant contenir un nombre très variable de descripteurs locaux.

Méthodes par appariement de graphes

Plusieurs algorithmes de comparaison d'empreintes digitales proposent une représentation de l'empreinte sous la forme d'un graphe, où les descripteurs locaux forment les sommets, et les relations géométriques entre descripteurs sont représentées par les arêtes. La distance entre deux images est alors calculée comme une distance entre graphes, avec des méthodes proches de celles évoquées en section 2.3.3.

Ce type de méthode a été proposé notamment par [124] qui utilise comme descripteur local les crêtes de l'empreinte digitale, et construit un graphe en reliant par des arêtes les crêtes en voisinage direct. Il a été utilisé également par [91], qui construit un graphe de structures locales de minuties, appariées en utilisant un algorithme de minimisation de flot au coût exponentiel.

Comme décrit dans le chapitre 2 à propos des algorithmes de comparaison d'images génériques, les méthodes utilisant la représentation par graphe, si elles permettent d'enrichir fortement la représentation des voisinages entre descripteurs locaux, impliquent un temps de comparaison qui les rendent non pertinentes pour des algorithmes d'identification rapide dans des très grandes bases de données.

3.3.3 Analyse des approches locales

La mise en perspective des différents descripteurs locaux et des différentes méthodes de comparaison avec les contraintes de notre problématique aboutit à procéder à ne conserver que certains de ces descripteurs et méthodes.

Pour le choix des descripteurs locaux, il est important de noter que les performances en précision de reconnaissance des descripteurs à base de crêtes sont très sensibles au bruit de mesure sur la localisation des crêtes, bruit qui peut s'avérer important sur la majorité des images d'empreintes digitales, y compris des acquisitions non latentes. Nous ne retiendrons donc pas les descripteurs utilisant les crêtes.

Pour les autres types de descripteurs, nous retenons les descripteurs suivants :

- MCC (structure locale de minuties centrée)
- triplets de minuties (structure locale de minuties non centrées)
- patch de texture locale

Parmi les différentes méthodes de comparaison, les méthodes avec orientation intrinsèque sont par nature peu robustes aux acquisitions de type latente, du fait de la nécessité de disposer d'une information globale de qualité suffisante. Les méthodes par appariement de graphes s'avèrent quant à elles trop coûteuses en temps de calcul, et surtout peu adaptées à une accélération potentielle du processus de recherche dans une grande base.

Nous nous concentrons donc sur les méthodes avec appariements locaux, pouvant être enrichies a posteriori avec de l'information de cohérence globale.

3.3.4 Bases de données

Pour évaluer les performances de nos algorithmes, nous utilisons les bases de données publiques mises à disposition par le NIST, ainsi que celles constituées à l'occasion des campagnes

d'évaluation FVC.

Le NIST a fourni quatre bases d'images d'empreintes digitales, composées de scan d'acquisitions papiers de fiches biométriques. Ces acquisitions sont également dites *enroulées*, du fait de la procédure d'acquisition par roulement du doigt sur une surface d'acquisition.

Ces bases sont les suivantes : NIST DB 4, NIST DB 10, NIST DB 9 et NIST DB 14. Parmi ces bases, les plus grandes sont les bases NIST DB 4 et NIST DB 14, contenant chacune deux acquisitions de respectivement 2000 et 27000 doigts. Le NIST a également fourni la base spéciale NIST DB 27, composées de 259 acquisitions latentes et 262 acquisitions enroulées de 259 doigts.

Les bases issues des campagnes de test FVC sont les bases FVC 2000, 2002, 2004 et 2006, chacune composées de 4 sous-bases notées DB 1 à DB 4. Les 4 sous-bases des bases FVC 2000, 2002 et 2004 sont composées chacune de 8 acquisitions de 110 doigts différents (le benchmark FVC utilisant les 10 premiers doigts de chaque sous-base comme base d'apprentissage). La base FVC 2006 est elle composée de 4 sous-bases de 12 acquisitions de 150 doigts chacune, les 10 premiers doigts de chaque base servant également comme bases d'apprentissage.

3.4 Descripteurs choisis

3.4.1 Minutiae Cylinder Code

Le Minutiae Cylinder Code est un descripteur local de type structure locale de minuties centré sur une minutie, à rayon fixe et de taille fixe.

Fondamentalement, le MCC est une carte locale d'un voisinage de minutie centré et orienté sur une minutie centrale. Le MCC est composé d'un ensemble de disques, chaque disque correspondant à une différence d'orientation quantifiée entre les minuties du voisinage et la minutie centrale. Au sein de chaque disque, la minutie est ensuite localisée en 2 dimensions. La présence d'une minutie est manifestée par les valeurs d'une fonction gaussienne centrée sur la position estimée de la minutie du voisinage par rapport à la minutie centrale.

Cette structure permet de stocker dans un unique descripteur de taille fixe à la fois la localisation et l'orientation des minuties d'un voisinage.

L'utilisation de fonctions gaussiennes pour représenter la présence d'une minutie permet également de donner une certaine marge d'erreur sur la localisation et l'orientation des minuties.

La description complète des minuties est fournie en annexe. Le schéma 3.8 tiré de [79] en donne une illustration.

Le détail de la construction des Minutiae Cylinder Code est donné en annexe 4.8

Pour la suite de nos travaux, nous ferons donc usage des Minutiae Cylinder Code tel que construit par le sdk fourni par [125], à partir des informations de localisation et d'orientation des minuties fournies par l'algorithme d'extraction des données (*codeur*) de Safran Morpho.

Nous retenons les paramètres de construction de MCC proposés par [79]. Cependant, ces paramètres impliquent notamment un rayon du MCC de 70 pixels à une résolution de 500 dpi. Cette taille de MCC risque de ne pas s'avérer suffisamment robuste pour des requêtes de type

paramètre	description	valeur
R	rayon du cylindre en pixels à 500 dpi	70
N_s	nombre de cellules sur le diamètre du cylindre	8
N_d	nombre de section sur la hauteur du cylindre	6
σ_s	déviati on standard dans 4.14	28/3
σ_D	déviati on standard dans 4.18	$\frac{2}{9}\pi$
μ_{Psi}, τ_{Psi}	paramètres de sigmoïde pour la fonction de valeur de cellule Ψ	0.01, 400
Ω	offset d'élargissement de l'enveloppe convexe des minuties prises en compte, en pixels	50
min_{VC}	nombre maximum de cellules valides pour valider un MCC	75% du nombre de cellules total
min_M	nombre minimum de minuties couvertes pour valider un MCC	2
min_{ME}	nombre minimum de cellules comparables pour que deux MCC soient comparables	60% du nombre de cellules total par MCC
δ_θ	rotation globale maximale autorisées entre deux MCC	$\pi/2$

TABLE 3.1 – Paramètres des grands MCC

paramètre	description	valeur
R	rayon du cylindre en pixels à 500 dpi	25
N_s	nombre de cellules sur le diamètre du cylindre	3
N_d	nombre de section sur la hauteur du cylindre	6
σ_s	déviatoin standard dans 4.14	28/3
σ_D	déviatoin standard dans 4.18	$\frac{2}{9}\pi$
μ_{Psi}, τ_{Psi}	paramètres de sigmoïde pour la fonction de valeur de cellule Ψ	0.01, 400
Ω	offset d'élargissement de l'enveloppe convexe des minuties prises en compte, en pixels	50
min_{VC}	nombre maximum de cellules valides pour valider un MCC	75% du nombre de cellules total
min_M	nombre minimum de minuties couvertes pour valider un MCC	2
min_{ME}	nombre minimum de cellules comparables pour que deux MCC soient comparables	60% du nombre de cellules total par MCC
δ_θ	rotation globale maximale autorisées entre deux MCC	$\pi/2$

TABLE 3.2 – Paramètres des petits MCC

Dans nos expériences, nous faisons le choix de ne pas tenir compte de l'information de validité de recouvrement dans le calcul d'un score ou d'une distance entre MCC. Ce choix s'explique par deux raisons. D'abord, omettre cette information ne devrait pas réduire considérablement les performances en reconnaissance attendue. [125] propose en effet un algorithme de comparaison d'empreinte digitales par LSH utilisant les MCC comme descripteurs locaux, en négligeant cette information, tout en obtenant de bonnes performances en reconnaissance. Ensuite, la non prise en compte de cette information permet de construire un score entre MCC plus simple en termes de coûts de calcul.

La distance d_{MCC} que nous associerons aux MCC est simplement la distance de Hamming normalisée entre les représentations binaires des MCC, hors information de validité des cellules :

$$d_{MCC}(C_a, C_b) = \frac{d_H(\mathbf{c}_a, \mathbf{c}_b)}{n}$$

3.4.2 Triplets de minuties

Parmi les principaux descripteurs à base de triplets de minuties recensés dans la littérature, les informations les plus couramment utilisées sont les suivantes :

- longueur des côtés du triangle formé par les segments inter-minuties
- angles de ce triangle
- orientation des minuties du triplet, exprimée dans un repère intrinsèque au triangle
- comptage de crêtes sur les segments entre minuties consécutives
- type de minuties du triplet

[90] fait remarquer que les informations de comptage de crête et de typage de minuties sont peu fiables, en particulier sur des empreintes de type latente, et que ces informations ne sont pas forcément disponibles sur toutes les bases recensées. Pour ces raisons, nous ne ferons pas usage de ces informations.

L'utilisation de la totalité des triplets de minuties possibles pour chaque image peut s'avérer problématique : en notant n le nombre de minuties présent sur une image d'empreinte, le nombre de triplets possibles évolue en $O(n^3)$. La localisation exacte des minuties étant sujette à du bruit de mesure, il nous semble nécessaire d'écarter du processus de constructions des triplets les minuties distantes de moins de 10 à 20 pixels (à 500 dpi). Le bruit de mesure rend également peu fiable les localisations relatives des minuties entre elles au-delà d'un certain voisinage, aussi il semble nécessaire d'éliminer aussi les triplets où les minuties seraient distantes de plus de 100 à 200 pixels.

D'une manière générale, il est également nécessaire de ne pas conserver un nombre trop important de triplets par image, d'abord parce que la majorité des scores agrégés entre descripteurs locaux ont des coûts de calcul en $O(nb \text{ minuties requete} \times nb \text{ minuties reference})$, de même que les transformées de Hough généralisées, ensuite pour limiter le coût de stockage de la signature associée aux triplets de minuties.

L'usage d'une triangulation de Delaunay étant répandu pour restreindre le nombre de triplets conservés par image, [96] a estimé le nombre moyen de triplets générés par image, en fonction du nombre de minuties présentes, dans plusieurs cas de figures :

- tous les triplets sont générés
- seuls les triplets dont les segments correspondent aux arêtes d'une triangulation de Delaunay sont générés
- seuls les triplets dont les segments correspondent aux arêtes d'une triangulation de Delaunay d'ordre 0 ou 1 sont générés

L'utilisation d'une triangulation de Delaunay d'ordre supérieur, concept introduit par [126], permet de réduire le problème inhérent à la triangulation simple, qui est la faible robustesse au bruit de localisation et à l'occultation des minuties.

La triangulation de Delaunay d'un ensemble de points P du plan est défini par une triangulation telle qu'aucun cercle inscrit dans l'un des triangle de la triangulation ne peut contenir de points de P autre que ses sommets.

Une triangulation de P est dite triangulation de Delaunay d'ordre k si le cercle inscrit dans les trois sommets de n'importe quel triangle de la triangulation contient au plus k autres sommets. Une triangulation de Delaunay simple correspond donc à une triangulation d'ordre zéro.

Le déplacement ou la suppression de sommets d'un ensemble de points est susceptible de perturber considérablement la triangulation de Delaunay d'ordre zéro.

En incluant dans la triangulation tous les triangles dont les cercles inscrits sont susceptibles de contenir d'autres points que les sommets, une triangulation de Delaunay d'ordre supérieure permet de préserver un plus grand nombre de triangle en cas de perturbation de l'ensemble des points de P .

Les triangulations de Delaunay d'ordres supérieurs constituent donc une heuristique intéressante pour limiter le nombre de triplets de minuties générés, tout en assurant une certaine robustesse au bruit et à la localisation.

Pour restreindre le nombre de triples de minuties, nous utilisons une heuristique approxi- mant une triangulation de Delaunay d'ordre supérieur. Pour chaque minutie, nous récupérons l'ensemble de ses N plus proches voisins en distance euclidienne, avec N défini au préalable. Ne sont alors conservés que les triplets dont chaque minutie appartient aux N plus proches voi- sines des deux autres.

Les triplets que nous utiliserons seront ainsi définis de la manière suivante :

1. Pour chaque minutie de l'empreinte, on construit la liste des N plus proches minuties voisines en distance euclidienne
2. On calcule tous les ensembles possibles de trois minuties sur la totalité de l'empreinte digitale
3. On élimine tous les triplets dont les trois côtés ne sont pas tous compris entre les seuils λ_{min} et λ_{max}
4. Parmi les triplets restants, on élimine tous les triplets dont une minutie ne figurerait pas dans les listes des N plus proches voisins des deux autres.
5. Pour chaque triplet ainsi sélectionné, on calcule les longueurs des trois côtés, triés par longueur décroissante, l_1, l_2 et l_3 .
6. Les minuties m_1, m_2 et m_3 de chaque triplet sont ordonnées également, de telle sorte que m_1 soit localisée entre l_1 et l_2 , et m_2 entre l_2 et l_3
7. Chaque minutie m_i se voit dotée d'une orientation locale α_i invariante aux rotations, α_i étant calculée comme la différence d'angle entre l'orientation θ_i de m_i dans le référentiel image, et le vecteur $\overrightarrow{m_1 m_2}$
8. le sens de parcours, ou orientation, hd (*handedness*) du triplet est donné par $sign(\overrightarrow{m_1 m_2} \wedge \overrightarrow{m_1 m_3})$ et vaut 1 ou -1

Nous conservons les caractéristiques suivantes pour chaque triplet de minuties : les lon- gueurs de côtés par ordre décroissant (l_1, l_2, l_3) , les orientations locales $(\alpha_1, \alpha_2, \alpha_3)$ de chaque minutie par rapport au vecteur $\overrightarrow{m_1 m_2}$, et l'orientation hd .

Pour calculer une dissimilarité entre triplets, nous avons construit la pseudo-distance d entre deux triplets de minuties $Tr^1 = (l_1^1, l_2^1, l_3^1, \alpha_1^1, \alpha_2^1, \alpha_3^1, hd^1)$ et $Tr^2 = (l_1^2, l_2^2, l_3^2, \alpha_1^2, \alpha_2^2, \alpha_3^2, hd^2)$.

$$d(Tr_1, Tr_2) = \sum_{i=1}^3 |hd^1 \times l_i^1 - hd^2 \times l_i^2| + 10 \times \left(\sum_{i=1}^3 (\cos(\alpha_i^1) - \cos(\alpha_i^2))^2 + (\sin(\alpha_i^1) - \sin(\alpha_i^2))^2 \right) \quad (3.3)$$

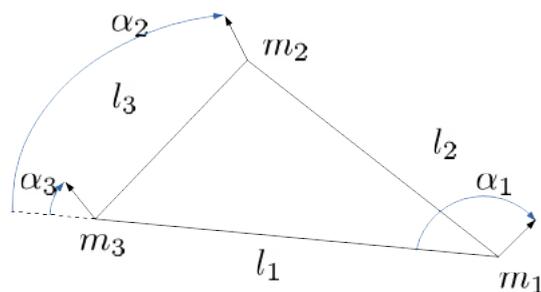


FIGURE 3.9 – schéma du triplet utilisé

Pour faciliter l'implémentation logicielle de d entre triplets, chaque triplet Tr est stocké sous la forme d'un vecteur de dimension 9 : $Tr = (hd \times l_1, hd \times l_2, hd \times l_3, 10 \times \cos(\alpha_1), 10 \times \cos(\alpha_2), 10 \times \cos(\alpha_3), 10 \times \sin(\alpha_1), 10 \times \sin(\alpha_2), 10 \times \sin(\alpha_3))$. Chaque composante du vecteur est stockée sur un octet, ce qui implique une certaine quantification des valeurs des triplets.

nombre de plus proches voisins conservés N	8
longueur maximale @500 dpi λ_{max}	15 pixels
longueur minimale @500 dpi λ_{min}	100 pixels

TABLE 3.3 – Paramètres des triplets de minuties

3.4.3 Patch d'orientation locale

Le descripteur patch d'orientation locale que nous construisons est inspiré du descripteur proposé par [101]. La principale différence tient dans le format de la grille d'échantillonnage utilisée, comme nous allons le décrire.

Les patch d'orientation locale sont construits à partir d'une carte d'orientation fournie par l'algorithme d'extraction d'information de bases (*codeur*) de Safran Morpho. En utilisant ce codeur, on extrait à partir des images d'empreinte des cartes d'orientation. Le processus que nous utilisons divise la résolution par 4 : pour des images de 512x512 pixels à 500 dpi, la carte d'orientation locale de l'ensemble de l'image est une matrice de taille 128x128, chaque coefficient de la matrice encodant une orientation locale, comprise entre 0 et π , que nous quantifions

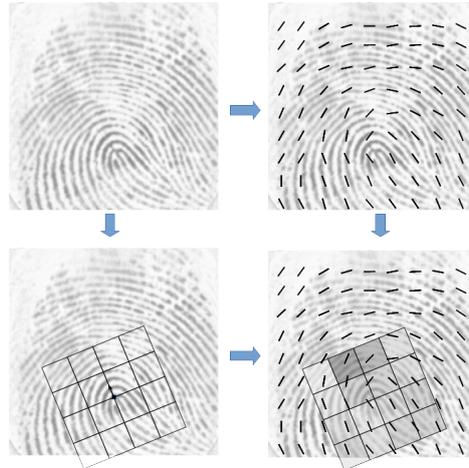


FIGURE 3.10 – Construction des patches de flux de crêtes : à partir de l'image initiale, on construit 1) la grille de flux crête (en haut à droite) 2) le patch centré et orienté sur la minutie (en bas à gauche). En 3), on utilise les valeurs du flux de crête pour remplir les cellules du patch (en bas à droite)

sur 128 niveaux. Les parties de l'image ne comportant pas de surface d'empreinte sont codées sur une valeur distincte.

Pour chaque minutie de l'image d'empreinte, on construit un patch. Chaque patch est une grille carrée de 8x8 cellules centrée et orientée sur sa minutie, dont la résolution est telle qu'à 500 dpi, un patch ayant même orientation que l'image recouvre une surface d'empreinte de 128x128 pixels, ou une surface de carte d'orientation de 32x32 valeurs d'orientation.

Pour remplir les cellules du patch, chaque cellule est projetée sur la carte d'orientation globale. La valeur de la cellule est alors la moyenne (modulo π) des valeurs des orientations locales de la carte d'orientation relevées à l'intérieur de la cellule. Si la cellule recouvre partiellement ou totalement une zone de l'image ne comportant pas de surface d'empreinte, la valeur de la cellule du patch est également codée sur une valeur spéciale ω .

Afin de disposer d'une pseudo-distance aisée à implémenter et manipuler, chaque patch Ptc de n cellules de valeurs d'orientation $[\theta_1, \theta_N]$ est représenté sous la forme d'un vecteur (x_1, \dots, x_{2n}) de $2n$ coefficients ainsi définis :

$$\forall i \in [1, n], x_{2i} = \begin{cases} \cos(2\theta_i) & \text{si } \theta_i \neq \omega \\ 2 & \text{si } \theta_i = \omega \text{ et } Ptc \text{ appartient à une image requête} \\ -2 & \text{si } \theta_i = \omega \text{ et } Ptc \text{ appartient à une image référence} \end{cases}$$

$$x_{2i+1} = \begin{cases} \sin(2\theta_i) & \text{si } \theta_i \neq \omega \\ 2 & \text{si } \theta_i = \omega \text{ et } Ptc \text{ appartient à une image requête} \\ -2 & \text{si } \theta_i = \omega \text{ et } Ptc \text{ appartient à une image référence} \end{cases}$$

Si l'on considère deux patchs $Ptc_1 = [x_{1,1}, x_{1,2n}]$ et $Ptc_2 = [x_{2,1}, x_{2,2n}]$, la pseudo-distance d choisie entre Ptc_1 et Ptc_2 est simplement la norme ℓ_2 entre ces deux vecteurs :

$$d(Ptc_1, Ptc_2) = \sum_{i=1}^{2n} (x_{1,i} - x_{2,i})^2 \quad (3.4)$$

Les coefficients associés aux cellules à valeur ω permettent de pénaliser uniformément les zones de non recouvrement de surface entre deux patchs comparés.

Si un patch présente une faible surface d'empreinte (donc avec beaucoup de cellules à ω), s'il dispose d'un patch matchant il risque d'avoir une forte dissimilarité à ce dernier. En parallèle, un patch présentant une faible surface d'empreinte risque de générer facilement des fausses acceptances avec d'autres patchs à faible surface, la probabilité de présenter par hasard une même configuration de flux de crête sur une plus petite surface étant plus forte.

Pour éviter ces cas de figures, les patchs présentant un nombre de cellules à valeur ω supérieur à 75% du nombre total de cellules ne sont pas référencées.

3.4.4 Patch d'orientation locale - décomposition parcimonieuse

Apprentissage de dictionnaire

Plusieurs algorithmes d'apprentissage de dictionnaires ont été développés pour produire des dictionnaires adaptés à la décomposition parcimonieuse de tout type de données. [127] propose le K-SVD, un algorithme itérant sur deux étapes : une première étape d'optimisation sur les mots du dictionnaire, et une deuxième étape de décomposition de la base d'apprentissage sur le dictionnaire mis à jour.

[128] propose une méthode d'apprentissage en ligne (*Online Learning*) avec une mise à jour du dictionnaire basée sur une descente de coordonnées par bloc (*block-coordinate descent*). La méthode de coordonnées par bloc a été améliorée par [129] pour produire des représentations parcimonieuses structurées hiérarchiquement, permettant d'ajouter de l'information globale à la représentation parcimonieuse.

Décomposition parcimonieuse

Une fois construit un dictionnaire de patchs *ad hoc*, nous résolvons le problème (2.19) pour l'ensemble des descripteurs locaux des empreintes de la base de référence, ainsi que pour les descripteurs locaux des empreintes requêtes, à partir de la représentation des patchs d'orientation comme vecteurs de sinus et cosinus. L'algorithme choisi pour la résolution de (2.19) est le LARS de [130], pour lequel nous utilisons l'implémentation développée par [131]. La figure 3.13 propose une visualisation des mots visuels du dictionnaire de patchs d'orientations appris à partir d'un échantillon de 10 000 patchs sélectionnés aléatoirement parmi les patchs construits sur la base NIST DB14.

L'algorithme LARS de [130] cherche à résoudre le problème 2.20 en plusieurs étapes, par ajouts successifs de coefficients non nuls (x_i) contribuant à la décomposition parcimonieuse

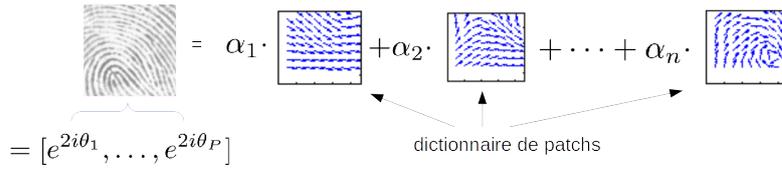


FIGURE 3.11 – Schéma de la décomposition parcimonieuse d'un patch de carte d'orientation sur un dictionnaire de patches

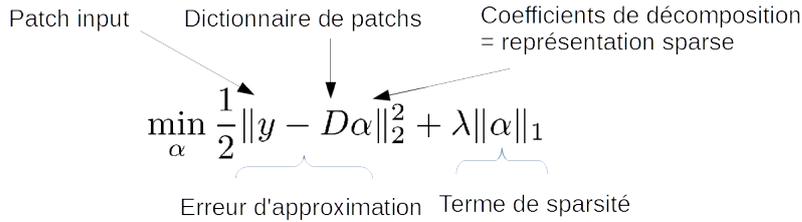


FIGURE 3.12 – Equation de minimisation d'erreur de reconstruction pour la décomposition parcimonieuse

$\tilde{y} = Dx$ d'un signal réel y .

Cet algorithme peut être résumé ainsi :

- Tous les coefficients (x_i) sont initialisés à zéro
- On trouve le mot d_{j_1} le plus fortement corrélé au signal y .
- On cherche le plus grand coefficient x_{j_1} jusqu'à ce que le mot d_{j_2} maximisant la corrélation avec le résidu $y - x_{j_1}d_{j_1}$ soit distinct de d_{j_1} .
- On calcule alors la direction équiangulaire ν_{j_2} aux vecteurs d_{j_1} et d_{j_2} , et on cherche le plus grand coefficient x_{j_2} en valeur absolue dans la direction de ν_{j_2} , jusqu'à ce que le mot d_{j_3} maximisant la corrélation avec le nouveau résidu $(y - x_{j_1}d_{j_1}) - x_{j_2}\nu_{j_2}$ soit indépendant de d_{j_1} et d_{j_2} .
- On réitère l'opération : on calcule une nouvelle direction ν_{j_3} équiangulaire aux vecteurs d_{j_1}, d_{j_2} , on cherche à maximiser $|x_{j_3}|$, jusqu'à ce que le mot d_{j_4} maximisant la corrélation avec $((y - x_{j_1}d_{j_1}) - x_{j_2}\nu_{j_2}) - x_{j_3}\nu_{j_3}$ soit indépendant de d_{j_1}, d_{j_2} et d_{j_3} .

L'opération est répétée jusqu'à ce que le résidu soit négligeable devant une valeur prédéfinie ϵ .

La procédure de décomposition parcimonieuse est illustrée dans les figures 3.11 et 3.12.

Implémentation

Pour la construction du dictionnaire de patch, nous avons utilisé la méthode `mexTrainDL` de [131]. Cette méthode vise à résoudre le problème suivant :

$$\min_{D \in \mathcal{C}} \lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n \min_{\alpha_i} \left(\frac{1}{2} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \right) \quad (3.5)$$

où C est défini par

$$C = \{D = [(d_1), \dots, (d_p)] \in \mathbf{R}^{m \times p} \text{ tel que } \forall j, \|d_j\|_2^2 \leq 1\}$$

Ce problème est résolu avec l'algorithme d'*Online Learning* par *Block-Coordinate Descent* de [129], en utilisant comme valeurs des (x_i) les représentations des patches Ptc_i de l'échantillon d'apprentissage de 10000 patches extraits de NIST DB 14.

La représentation parcimonieuse des patches Ptc pour chaque image d'empreinte est ensuite obtenue en utilisant une version de la méthode **mexLasso** de [131]. Cette méthode vise à résoudre le problème suivant :

$$\min_{y \in \mathbb{R}^p} \frac{1}{2} \|y - Dx\|_2^2 + \lambda \|xa\|_1 \quad (3.6)$$

qui est une formulation du problème 2.20.

Ce problème est résolu avec l'algorithme LARS décrit précédemment. Les y_i correspondent ici aux représentations des patches Ptc_i de l'ensemble des empreintes digitales de la base de référence et de chaque requête à chercher. Les vecteurs $\{x\} = \{(x_i)\}$ obtenus sont utilisés comme représentations parcimonieuses des patches de carte d'orientation.

La valeur de λ que nous utilisons pour l'apprentissage de dictionnaire et pour l'obtention de la décomposition parcimonieuse est $\lambda = 0.2$. Cette valeur est fixée expérimentalement après essai de différentes valeurs, afin d'assurer en moyenne 10% de coefficients x_i non nuls lors de la décomposition.

Nous avons testé plusieurs tailles de dictionnaires pour le calcul de D , un dictionnaire de taille 64 constitue expérimentalement un bon compromis entre précision de la reconnaissance et taille du descripteur obtenu. Une visualisation des mots du dictionnaire, ramenés en orientations en deux dimensions, est présentée en figure 3.13

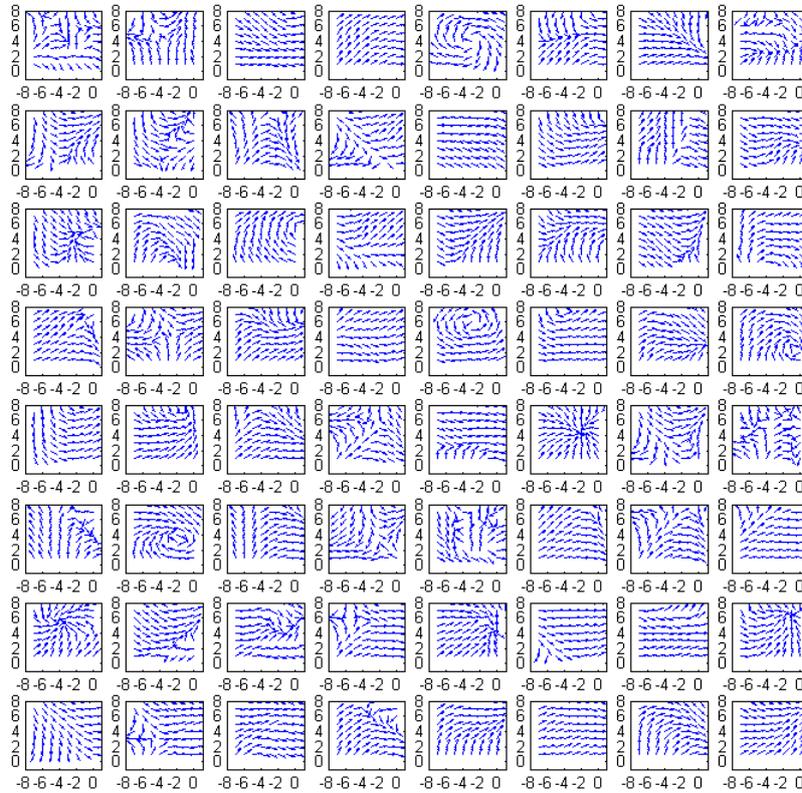


FIGURE 3.13 – visualisation des patches du dictionnaire de 64 mots générés par Online Learning via LARS

Si le dictionnaire de patches comporte n mots visuels, alors chaque patch Ptc de carte d'orientation peut être représenté par un vecteur de coefficients de décomposition parcimonieuse sur ce dictionnaire $Ptc_{sp} = (\alpha_1, \dots, \alpha_n)$

Nous avons utilisé comme pseudo-distances entre représentations parcimonieuses de patches de carte d'orientation la distance de Hamming entre les vecteurs Ptc_{sp} binarisés (coefficient porté à 1 si non nul, 0 sinon).

3.4.5 Algorithmes de comparaison

Pour chacun des descripteurs locaux choisis, on teste donc les distances et scores agrégés suivants, choisis à l'issue de l'analyse du chapitre 2.

Pour deux images $I_1 = (z_{1,i})$ et $I_2 = (z_{2,j})$, les distances et scores agrégés utilisés sont définis ainsi :

– distance agrégée simple :

$$D(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \min_j d(z_{1,i}, z_{2,j})}{n_1} \quad (3.7)$$

– distance agrégée corrigée :

$$\tilde{D}(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \min_j h(d(z_{1,i}, z_{2,j}))}{n_1} \quad (3.8)$$

avec

$$h(d) = \begin{cases} \frac{(1 - \cos(\frac{d}{\alpha} \cdot \pi/2))^2}{2} & \text{si } d < \alpha \\ 1 - \frac{(1 - \cos(\frac{\beta-d}{\beta-\alpha} \cdot \pi/2))^2}{2} & \text{si } \alpha \leq d < \beta \\ 1 & \text{si } d > \beta \end{cases} \quad (3.9)$$

– score agrégé simple

$$S(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \max_j s(z_{1,i}, z_{2,j})}{n_1} \quad (3.10)$$

avec

$$\forall z_{1,i} \in I_1, \forall z_{2,j} \in I_2, s(z_{1,i}, z_{2,j}) = \frac{1}{1 + \alpha \cdot d(z_{1,i}, z_{2,j})} \quad (3.11)$$

– score agrégé *a contrario*

$$S_{ac}(I_1, I_2) = \sum_i \frac{-\log(f(\min_j d(z_{1,i}, z_{2,j})))}{\sum_k m_{ik}} \quad (3.12)$$

Pour les distances ou scores présentant les meilleures performances, on teste les trois versions de transformées de Hough généralisé :

- version 1 : transformée avec seuillage
- version 2 : transformée avec meilleurs appariements
- version 3 : transformée avec N meilleurs appariements

Pour déterminer le degré de quantification nécessaire de l'espace des rotations affines en 2 dimensions, nous avons visualisé sur la base d'apprentissage FVC2000 DB 2 la distribution des 3 composantes des rotations affines (translation en δx , δy , rotation $\delta \alpha$) sur l'ensemble des appariements vérité terrain. Cette visualisation correspond à la figure 3.14.

Nous avons représenté les distributions des écarts des valeurs de rotation et de translation par rapport à la valeur moyenne constatée sur les appariements vérité terrain pour chaque couple d'images matchantes de la base d'apprentissage.

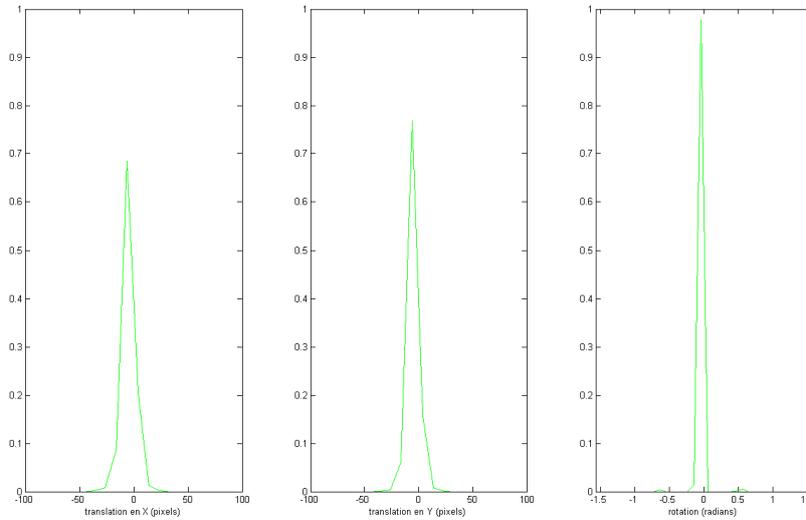
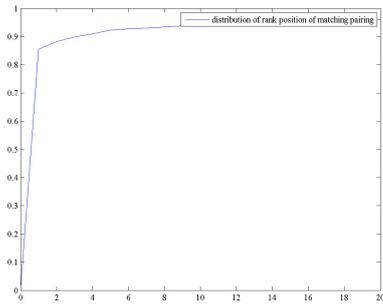


FIGURE 3.14 – Distribution des composantes des rotations affines pour des appariements vérité terrain (vert) et aléatoires (rouge) sur FVC 2000 DB 2

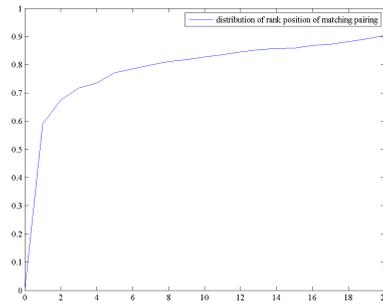
Etant donné les distributions constatées, nous considérons un espace de rotations affines à 500 dpi de $[-256, 256] \times [-256, 256]$ pixels en translation, et $[-\pi, \pi]$ radians en rotation. Chaque dimension est échantillonnée en 20 clusters.

Afin de déterminer le nombre optimal d'appariements n à faire voter dans le cas de la transformée de Hough avec n meilleurs appariements, nous avons calculé, pour les différents descripteurs considérés et pour chaque descripteur local pour lequel on dispose d'un appariement matchant vérité terrain, le rang auquel le descripteur local matchant apparaît dans la liste triée des descripteurs de l'image matchante.

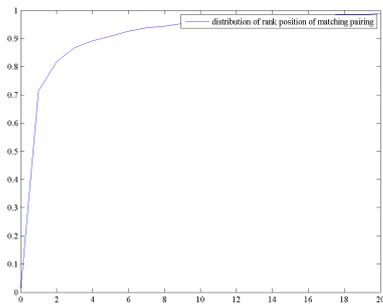
Les figures 3.15 présentent, pour chaque type de descripteur, le ratio, pour l'ensemble des descripteurs requêtes de la base d'apprentissage disposant d'un appariement matchant vérité terrain, de ceux ayant trouvé leur matchant au rang $\leq i$ parmi la liste des descripteurs références de l'image matchante triée par pseudo-distance croissante.



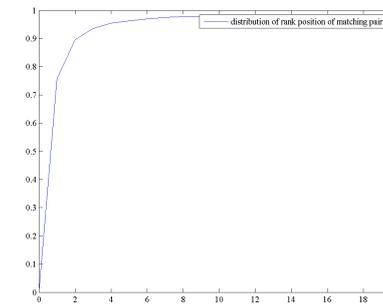
(a) grand MCC



(b) petit MCC



(c) patch de carte d'orientation



(d) triplet de minutie

FIGURE 3.15 – rang d'apparition du descripteur local matchant pour chaque type de descripteur considéré

Pour chaque type de descripteur, dans le cas de la transformée de Hough avec N meilleurs appariements, nous faisons voter les meilleurs appariements jusqu'aux rangs détaillés dans le tableau 3.4 inclus.

Descripteur	rang N maximal
grand MCC	2
petit MCC	4
patch carte d'orientation et patch parcimonieux	4
triplet de minuties	3

TABLE 3.4 – rang N maximal des appariements votant dans la transformée de Hough avec N meilleurs appariements

L'application des transformées de Hough aux distances et scores décrits ci-dessus est réalisée comme suit :

1. On note cl_{Bh} le cluster de l'espace quantifié des similitudes le plus rempli par la transformée de Hough de version h .

2. Afin d'obtenir une robustesse aux effets de bord de la quantification de l'espace des similitudes, on utilise deux offsets différents de quantification pour chaque dimension de cet espace. L'espace étant réduit à trois dimensions (deux dimensions spatiales, une dimension en rotation, pas de facteur d'échelle), on considère donc $2^3 = 8$ grilles de quantification différentes, cl_{Bh} étant défini comme le cluster le plus rempli sur ces 8 grilles.

3. $\forall h \in [1, 3]$

– distance agrégée simple :

$$D_{Hough\ h}(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \min_{j/(z_{1,i}, z_{2,j}) \in cl_{Bh}} d(z_{1,i}, z_{2,j})}{card(cl_{Bh})} \quad (3.13)$$

– distance agrégée corrigée :

$$\tilde{D}_{Hough\ h}(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \min_{j/(z_{1,i}, z_{2,j}) \in cl_{Bh}} h(d(z_{1,i}, z_{2,j}))}{card(cl_{Bh})} \quad (3.14)$$

avec

$$h(d) = \begin{cases} \frac{(1 - \cos(\frac{d}{\alpha} \cdot \pi/2))^2}{2} & \text{si } d < \alpha \\ 1 - \frac{(1 - \cos(\frac{\beta-d}{\beta-\alpha} \cdot \pi/2))^2}{2} & \text{si } \alpha \leq d < \beta \\ 1 & \text{si } d > \beta \end{cases} \quad (3.15)$$

– score agrégé simple

$$S_{Hough\ h}(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \max_{j/(z_{1,i}, z_{2,j}) \in cl_{Bh}} s(z_{1,i}, z_{2,j})}{n_1} \quad (3.16)$$

avec

$$\forall z_{1,i} \in I_1, \forall z_{2,j} \in I_2, s(z_{1,i}, z_{2,j}) = \frac{1}{1 + \alpha \cdot d(z_{1,i}, z_{2,j})} \quad (3.17)$$

– score agrégé *a contrario*

$$S_{ac\ Hough\ h}(I_1, I_2) = \sum_i \frac{-\log(f(\min_{j/(z_{1,i}, z_{2,j}) \in cl_{Bh}} d(z_{1,i}, z_{2,j})))}{\sum_k m_{ik}} \quad (3.18)$$

3.5 Evaluation et analyse

Pour analyser les propriétés des différents descripteurs locaux, on utilise la base FVC 2000 DB 2 décrite en section 3.3.4. Pour chacun des 110 doigts de cette base, on utilise la première acquisition comme requête et les 7 acquisitions suivantes comme références. On utilise un algorithme d'*authentication* commercial de haut niveau pour calculer des appariements de minuties entre chaque requête et toutes ses références correspondantes. Ces appariements sont ensuite considérés comme une *vérité terrain* d'appariements de minuties.

Nombre de minutes extraites dans la base de référence	24000
Nombre de minutes extraites dans la base de requêtes	4000
Nombre de minutes de la base de requêtes disposant d'un appariement matchant vérité terrain	2100

TABLE 3.5 – Données sur les minutes de la base FVC 2000 DB 2

3.5.1 Analyse locale

Pour les figures suivantes portant sur l'analyse des appariements locaux, la légende suivante a été adoptée :

- rouge pointillé = distribution des distances entre appariements locaux non matchants aléatoires
- rouge plein = pour chaque descripteur local d'une image requête, pour chaque image de référence, on cherche le meilleur appariement (la plus petite distance) du descripteur requête aux descripteurs de l'image de référence. On visualise la distribution ainsi obtenue.
- vert plein = ensemble des meilleurs appariements par couples d'images matchantes
- bleu pointillé = ensemble des meilleurs appariements par couples d'images matchantes, vérifiant la vérité terrain
- vert pointillé = ensemble des meilleurs appariements par couples d'images matchantes, fausses acceptances
- bleu plein = ensemble des appariements vérité terrain

La figure 3.16 propose une version simplifiée du schéma qui sera repris successivement pour chacun des descripteurs locaux.

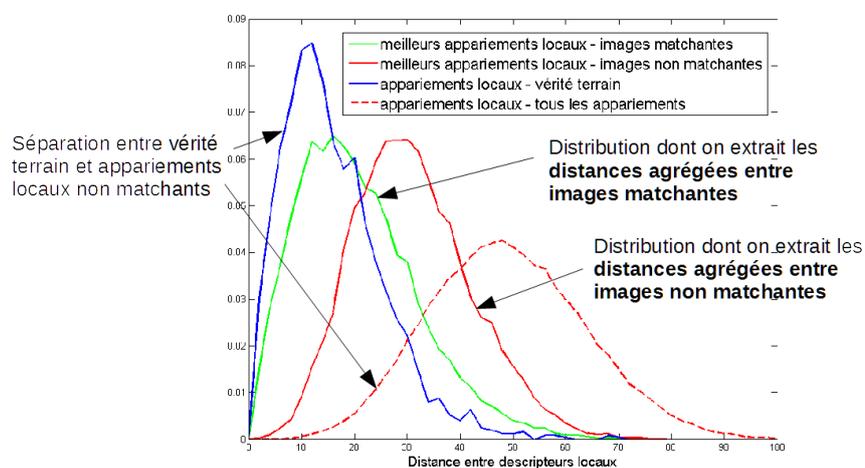


FIGURE 3.16 – distribution des distances entre descripteurs locaux - analyse

On notera que la distance agrégée est calculée, pour les couples d'images matchantes, à partir de distances situées sur la distribution verte, et pour les couples d'images non matchantes

à partir de distances situées sur la distribution rouge.

La qualité d'un descripteur en termes de précision de la reconnaissance peut être évaluée selon les critères suivants :

- séparation des descripteurs matchants des descripteurs non matchants
- minimisation des fausses acceptances

Grands MCC

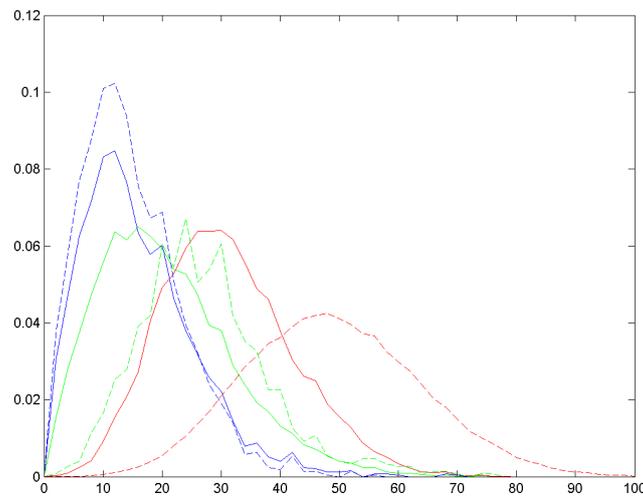


FIGURE 3.17 – distribution des distances entre grands MCC

On observe une nette séparation entre la distribution des appariements de MCC de la vérité terrain et la distribution des appariements non matchants, qui confirme la qualité du descripteur.

On remarque cependant les points suivants :

- un certain nombre d'appariements vérité terrain présentent une distance proche de la distance moyenne observée pour des appariements non matchants

Pour analyser les limites de ce type de descripteur associé à ce type de distance, on procède à la visualisation et à l'analyse d'une part des appariements vérité terrain présentant une distance proche de l'aléatoire, d'autre part des appariements non matchants présentant une distance très faible, et donc les plus susceptibles de générer des fausses acceptances.

On sélectionne un échantillon d'appariements vérité terrain présentant la plus forte distance (avec distance inter-MCC > 40), qu'on classe, selon des critères de visualisation subjectifs, en 4 catégories. Un très grand nombre de ces appariements concerne des appariements de MCC non pris en compte lors de la phase d'agrégation, car l'un des deux MCC au moins présente une surface d'empreinte insuffisante dans son rayon. Ces appariements de MCC ont été placés

heuristiquement à une distance arbitrairement grande. Les 4 catégories sont détaillées dans le tableau 3.6.

Nombre d'appariements MCC	1600
MCC éliminés pour surface insuffisante	1498
non détection/occultation de minuties	25
minuties parasites/fausse détection	16
autre problème	61

TABLE 3.6 – Analyse pires faux rejets grands MCC



FIGURE 3.18 – exemple de faux rejet descripteur grand MCC avec surface insuffisante



FIGURE 3.19 – exemple de faux rejet descripteur grand MCC avec occultation de minuties



FIGURE 3.20 – exemple de faux rejet descripteur grand MCC avec minuties parasites



FIGURE 3.21 – exemple de faux rejet descripteur grand MCC, autre type de problème

On considère à présent un échantillon d'appariements non matchants présentant la plus faible distance (avec distance inter-MCC < 5), qu'on classe en 3 catégories, détaillées dans le tableau 3.7, selon des critères de visualisation subjectifs également.

Nombre d'appariements MCC	350
Trois minuties ou moins	138
minuties en bordure de MCC/effet de bord	105
Structures fortement ressemblantes	36
autre problème	71

TABLE 3.7 – Analyse fausses acceptances grands MCC

On peut remarquer qu'il serait envisageable de réduire le nombre de fausses acceptances en

imposant un critère d'un nombre minimal de minuties dans le MCC de l'ordre de 3 ou 4 pour indexer le descripteur.



FIGURE 3.22 – exemple de fausse acceptance descripteur grand MCC, moins de trois minuties



FIGURE 3.23 – exemple de fausse acceptance descripteur grand MCC, structure fortement ressemblante



FIGURE 3.24 – exemple de fausse acceptance descripteur grand MCC, problème non identifié

Petits MCC

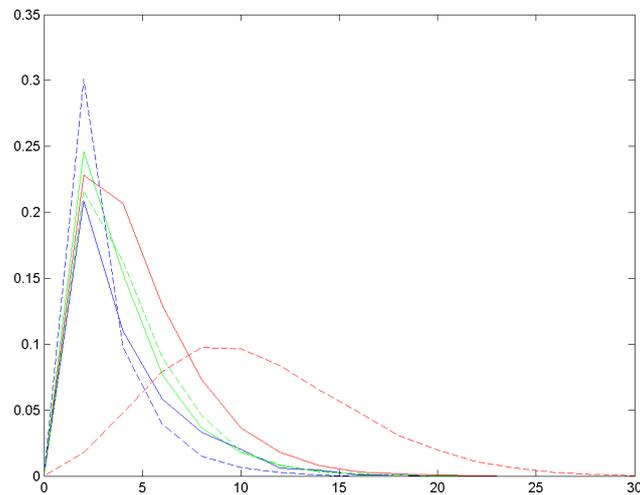


FIGURE 3.25 – distribution des distances entre petits MCC

Par rapport aux grands MCC, on constate que, si la séparation entre la distribution des appariements de MCC de la vérité terrain et la distribution des appariements non matchants reste nette, la probabilité de trouver au sein de chaque image de référence un descripteur MCC proche d'un descripteur requête quelconque est forte

On sélectionne un échantillon d'appariements vérité terrain présentant la plus forte distance (avec distance inter-MCC > 8), qu'on classe en 2 catégories. Comme pour les grands MCC, on observe un grand nombre d'appariements avec l'un des deux MCC au moins qui présente une surface d'empreinte insuffisante dans son rayon. Les catégories sont détaillées dans le tableau 3.8

Nombre d'appariements MCC	480
MCC éliminés pour surface insuffisante	475
déplacement de minuties	5

TABLE 3.8 – Analyse faux rejets petits MCC



FIGURE 3.26 – exemple de faux rejet descripteur petit MCC avec surface insuffisante

FIGURE 3.27 – exemple de faux rejet descripteur petit MCC avec déplacement de minuties

On considère à présent un échantillon d'appariements non matchant présentant la plus faible distance (avec distance inter-MCC < 2). Dans ce cas là, on constate que la totalité des 100 fausses acceptances analysées correspondent à un descripteur MCC vide (pas de minutie autre que la minutie centrale).



FIGURE 3.28 – exemple de fausse acceptation descripteur petit MCC, moins de trois minuties

Patches de carte d'orientation

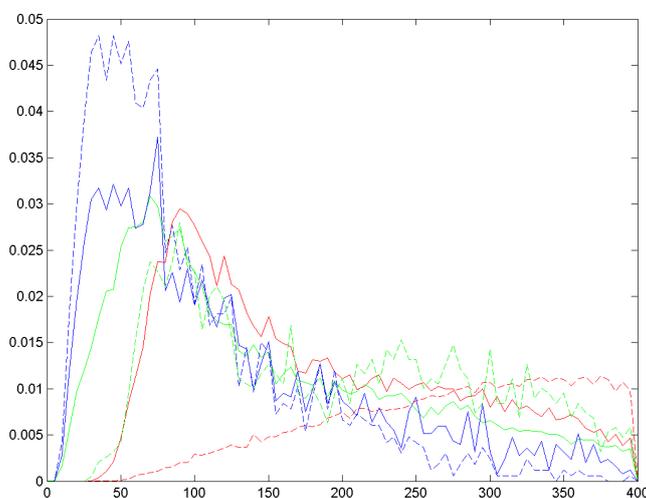


FIGURE 3.29 – distribution des distances entre patches de flux de crêtes

Sur les patches de carte d'orientation, on observe que la séparation entre la distribution des appariements de la vérité terrain et la distribution des appariements non matchants est bonne. La probabilité de trouver au sein de chaque image de référence un descripteur patch proche d'un descripteur requête quelconque apparaît plus élevée que sur les grands MCC (moindre séparation des distributions vertes et rouge pointillé), néanmoins, elle reste plus faible que sur les petits MCC.

On sélectionne un échantillon d'appariements vérité terrain présentant la plus forte distance (avec distance inter-patches > 300 , qu'on classe en 3 catégories, détaillées dans le tableau 3.9. Comme pour les MCC, un certain nombre d'appariements sont éliminés de la phase d'agrégation car l'un des deux patches présente une surface d'empreinte insuffisante, ces appariements ont été mis à une distance arbitrairement grande.

Les résultats sont détaillés dans le tableau 3.9.

Nombre d'appariements patch ridge flow	350
patch éliminés pour surface insuffisante	294
différence d'orientation minutie centrale/champ de crêtes local	32
information trop bruitée	24

TABLE 3.9 – analyse faux rejets patches carte d'orientation

La principale source de faux rejets concerne les patches présentant des surfaces de recouvrement insuffisantes, quand au moins l'un des deux patches se trouve en bordure d'empreinte. Les deux autres types de faux rejets concernent :

- soit les cas de figure où le motif d'orientation n'est pas assez discriminant (en particulier les motifs de crêtes monodirectionnelles). Ces associations sont alors très sensibles au bruit d'acquisition.
- soit les cas de figures où le motif est fortement bruité sur l'une des deux images



FIGURE 3.30 – exemple de faux rejet descripteur patch avec surface insuffisante

FIGURE 3.31 – exemple de faux rejet descripteur patch zone d'orientation complexe

Représentation parcimonieuse des patches de carte d'orientation

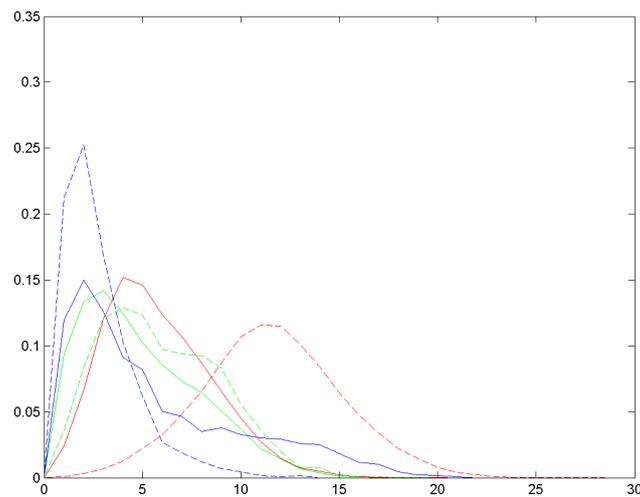


FIGURE 3.32 – distribution des distances entre représentations parcimonieuses de patches de flux de crêtes

Sur les patches de carte d'orientation, on observe que la séparation entre la distribution des appariements de la vérité terrain et la distribution des appariements non matchants est bonne. La probabilité de trouver au sein de chaque image de référence un descripteur patch proche d'un descripteur requête quelconque apparaît plus élevée que sur les grands MCC (moindre séparation des distributions vertes et rouge pointillé), néanmoins, elle reste plus faible que sur les petits MCC.

Nous avons également constaté que l'analyse des appariements locaux matchants à forte distance et des appariements non matchants à faible distance se recoupaient très largement avec ceux de la représentation complète.

Triplets de minuties

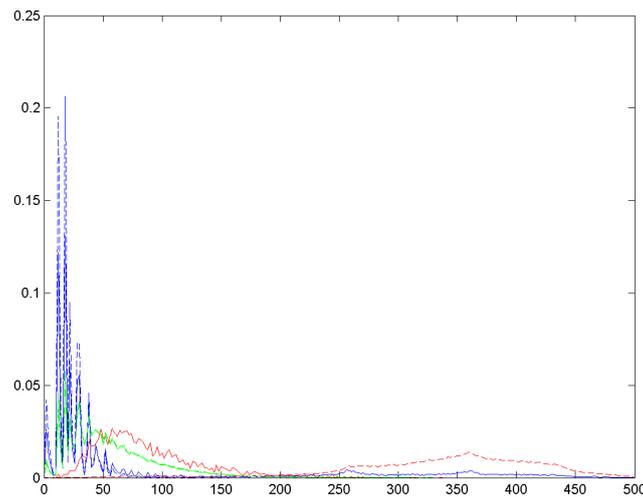


FIGURE 3.33 – distribution des distances entre triplets de minuties

Sur les triplets de minuties, on observe que la séparation entre la distribution des appariements de MCC de la vérité terrain et la distribution des appariements non matchants est bonne. La probabilité de trouver au sein de chaque image de référence un descripteur MCC proche d'un descripteur requête quelconque apparaît plus élevée que sur les grands MCC (moindre séparation des distributions vertes et rouge pointillé), néanmoins, elle reste plus faible que sur les petits MCC.

En visualisant un échantillon d'appariements vérité terrain présentant la plus forte distance (avec distance inter-triplets > 50), on constate que l'erreur en distance peut venir à la fois de l'erreur de positionnement des minuties en deux dimensions, erreur typiquement due à la distorsion de la peau, et de l'erreur d'orientation des minuties.

En visualisant un échantillon d'appariements non matchant présentant la plus faible distance (avec distance inter-triplets < 5), on constate cette fois que les fausses acceptances sont nécessairement liées à des structures de triplets semblables.



FIGURE 3.34 – exemple de faux rejet descripteur triplet de minuties

FIGURE 3.35 – exemple de fausse acceptation descripteur triplet de minuties

La cause essentielle de faux rejets dans le cas des triplets de minuties s'avère donc le bruit d'acquisition : déplacement de minuties, erreur dans le calcul de l'orientation, ou simplement occultation totale. Ces faux rejets peuvent être compensés lors de la phase d'acquisition par le grand nombre de triplets disponibles, permettant de consolider une information peu fiable sur une partie de l'image, par l'information d'une autre partie.

3.5.2 Distance agrégée

Si on revient sur la figure 4.32 correspondant aux distributions des distances entre descripteurs grands MCC, on peut noter que la distance agrégée entre images matchantes est construite à partir de distances locales issues de la distribution verte, et que la distance agrégée entre images non matchantes est construite à partir de distances locales issues de la distribution rouge.

La distance agrégée 3.13 est en effet construite comme la moyenne des meilleurs appariements locaux, c'est-à-dire ceux minimisant les distances entre descripteurs locaux.

La séparation entre la distribution rouge (meilleurs appariements par couple d'image non matchante) et la distribution verte (meilleurs appariements par couple d'image matchante) étant nettement moins marquée qu'entre la distribution bleue (appariements vérité terrain) et la distribution rouge pointillé (appariements aléatoires), on s'attend logiquement à ce que les distributions de distances agrégées ne séparent pas les couples d'images matchantes et non matchantes de manière optimale.

Les distributions de distances agrégées entre couples d'images matchantes et non matchantes ainsi que les performances obtenues en termes de taux de pénétration versus taux de faux rejets sont détaillés dans les figures 3.36 à 3.45.

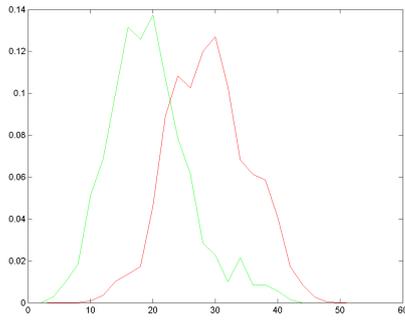


FIGURE 3.36 – distribution des distances agrégées matchantes (vert) et non matchantes (rouge) pour les grands MCC

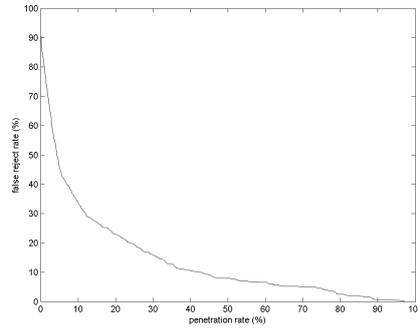


FIGURE 3.37 – taux de pénétration vs taux de faux rejets pour la distance agrégée grands MCC

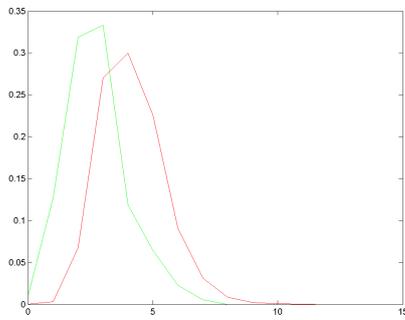


FIGURE 3.38 – distribution des distances agrégées matchantes (vert) et non matchantes (rouge) pour les petits MCC

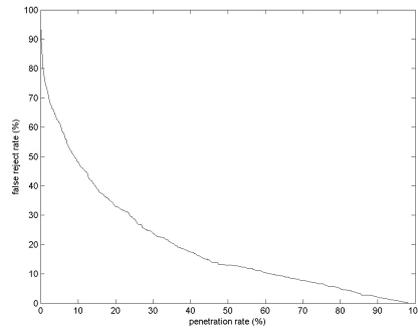


FIGURE 3.39 – taux de pénétration vs taux de faux rejets pour la distance agrégée petits MCC

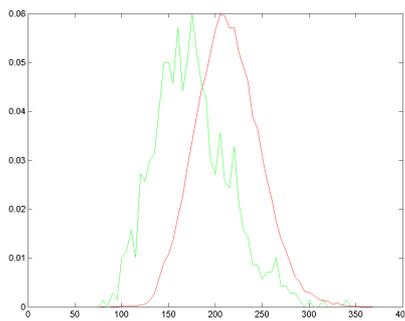


FIGURE 3.40 – distribution des distances agrégées matchantes (vert) et non matchantes (rouge) pour les patches de carte d'orientation

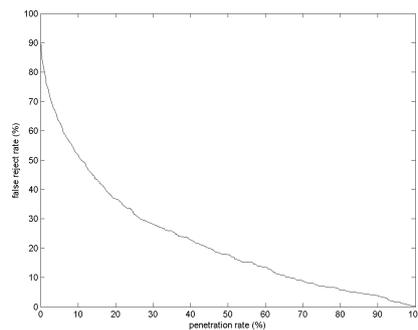


FIGURE 3.41 – taux de pénétration vs taux de faux rejets pour la distance agrégée patches de carte d'orientation

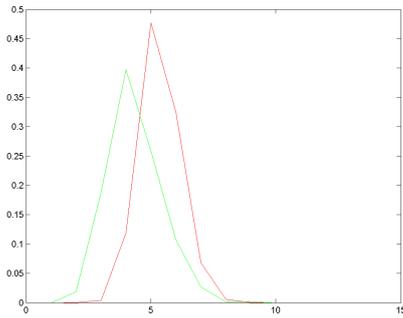


FIGURE 3.42 – distribution des distances agrégées matchantes (vert) et non matchantes (rouge) pour les représentations parcimonieuses de cartes d'orientation

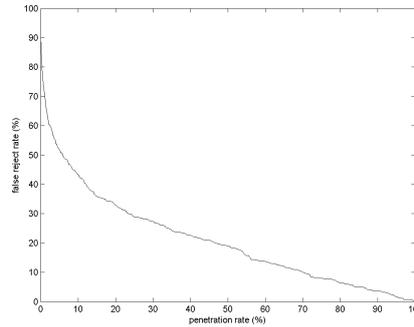


FIGURE 3.43 – taux de pénétration vs taux de faux rejets pour la distance agrégée coefficients parcimonieux patches d'orientation

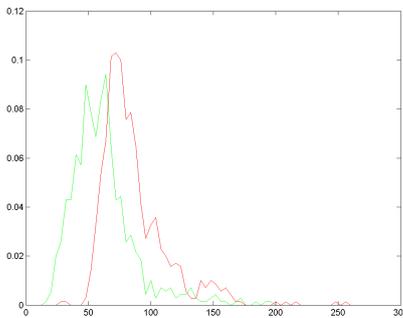


FIGURE 3.44 – distribution des distances agrégées matchantes (vert) et non matchantes (rouge) pour les triplets de minutie

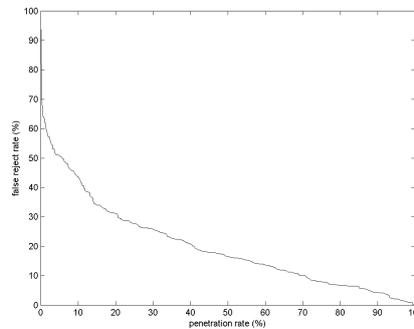


FIGURE 3.45 – taux de pénétration vs taux de faux rejets pour la distance agrégée triplets de minutes

Les performances observées en taux de pénétration vs taux de faux rejets sont effectivement assez faibles pour tous les descripteurs. On peut déjà noter à ce stade que les meilleures performances sont obtenues avec les grands MCC. Ce constat est naturel : les grands MCC sont un descripteur couvrant une surface assez importante d'empreinte, et agrégeant à la fois une information très locale assez précise (position et orientation des minuties), plus une certaine quantité d'information globale, de part sa taille.

3.5.3 Effets de la correction de distance

Une première heuristique a été envisagée pour améliorer la séparation des distributions de distances agrégées 3.14. Cette heuristique consiste à appliquer une fonction de distorsion non linéaire sur les distances locales, de manière à réduire significativement les distances observées entre les appariements locaux ayant une très forte probabilité d'être des appariements matchants,

et de forcer une distance élevée entre les appariements locaux ayant une forte probabilité d'être non matchants.

Les paramètres α et β permettant de définir la fonction de correction de distance définie en 2.29 sont résumés dans le tableau 3.10. Ces paramètres ont été sélectionnés après une phase d'apprentissage en force brute, en cherchant à minimiser le taux de faux rejets à 10% de taux de pénétration.

Descripteurs	α	β
grands MCC	5	30
petits MCC	1.5	4
patchs de carte d'orientation	75	600
coeffs. parcimonieux de patchs	6	20
triplets de minuties	2	40

TABLE 3.10 – Paramètres de correction de distance

Les distributions de distances agrégées entre couples d'images matchantes et non matchantes ainsi que les performances obtenues en termes de taux de pénétration versus taux de faux rejets sont détaillés dans les figures 3.46 à 3.55.

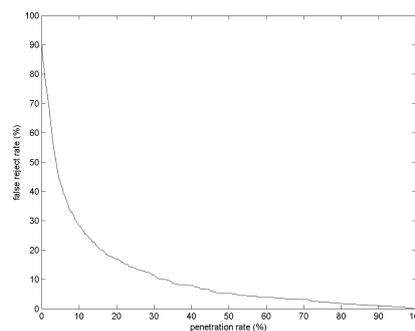
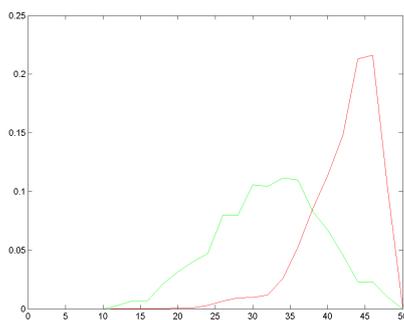


FIGURE 3.46 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les grands MCC

FIGURE 3.47 – taux de pénétration vs taux de faux rejets pour la distance agrégée corrigée grands MCC

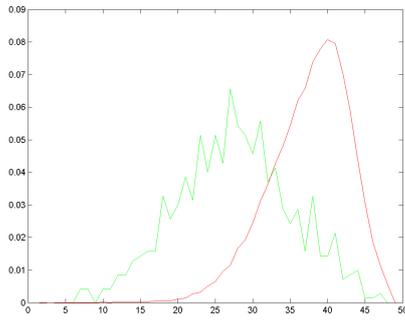


FIGURE 3.48 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les petits MCC

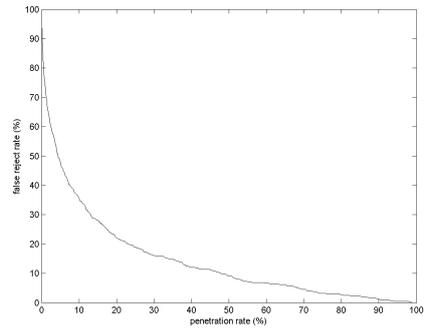


FIGURE 3.49 – taux de pénétration vs taux de faux rejets pour la distance agrégée corrigée petits MCC

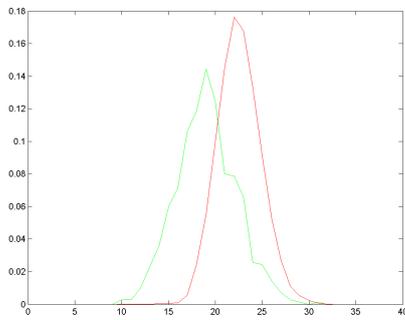


FIGURE 3.50 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les patches de carte d'orientation

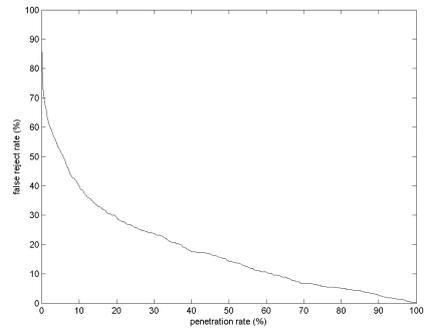


FIGURE 3.51 – taux de pénétration vs taux de faux rejets pour la distance agrégée corrigée patches de carte d'orientation

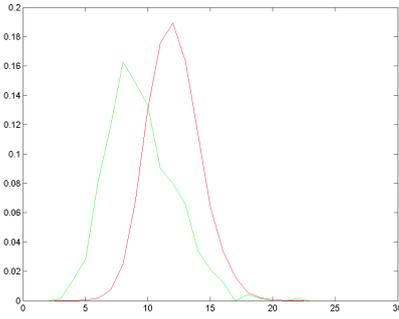


FIGURE 3.52 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les représentations parcimonieuses de cartes d'orientation

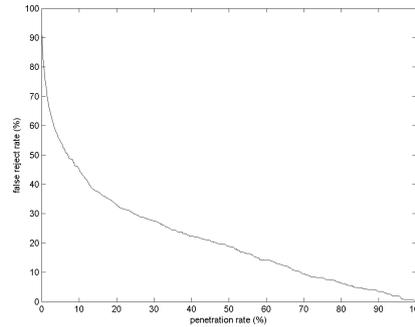


FIGURE 3.53 – taux de pénétration vs taux de faux rejets pour la distance agrégée corrigée co-efficents parcimonieux patches d'orientation

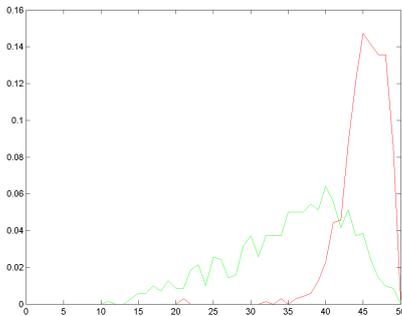


FIGURE 3.54 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les triplets de minutie

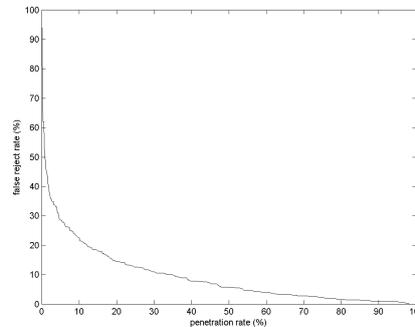


FIGURE 3.55 – distribution des distances agrégées corrigées matchantes (vert) et non matchantes (rouge) pour les triplets de minutie

On constate pour l'ensemble des descripteurs une amélioration générale des performances en reconnaissance, c'est-à-dire une diminution du taux de faux rejets à toute valeur du taux de pénétration. Cette amélioration est due à l'effet de seuillage de la correction de distance : les appariements à faible distance sont "reconnus" comme étant très probablement matchants, et leur distance est réduite, les appariements à forte distance sont "reconnus" comme étant très probablement non matchants, et leur distance est portée à une valeur arbitrairement grande. L'amélioration des performances est essentiellement due à la dégradation des distances agrégées entre image non matchantes, par accroissement des distances locales entre les appariements locaux fausse acceptance.

3.5.4 Score agrégé

Pour rappel, le score agrégé entre deux images $I_1 = (z_{1,i})$ et $I_2 = (z_{2,j})$ est calculé à partir de la distance locale de la façon suivante 3.10 :

$$\forall z_{1,i} \in I_1, \forall z_{2,j} \in I_2, s(z_{1,i}, z_{2,j}) = \frac{1}{1 + \alpha \cdot d(z_{1,i}, z_{2,j})}$$

$$S(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \max_j s(z_{1,i}, z_{2,j})}{n_1}$$

Un des principaux effets du score employé est similaire à celui recherché pour la distance corrigée : les scores des appariements de descripteurs locaux à forte distance sont écrasés vers zéro, les scores des appariements locaux à faible distance peuvent être arbitrairement grands.

Cet effet aboutit à des scores agrégés pouvant être très élevés dès qu'un petit nombre de descripteurs locaux sont à très faible distance (c'est-à-dire très probablement matchants), et à des scores agrégés uniformément faibles dans tous les autres cas (c'est à dire pour des couples d'images très probablement non matchantes).

Les distributions de distances agrégées entre couples d'images matchantes et non matchantes ainsi que les performances obtenues en termes de taux de pénétration versus taux de faux rejets sont détaillés dans les figures 3.56 à 3.65.

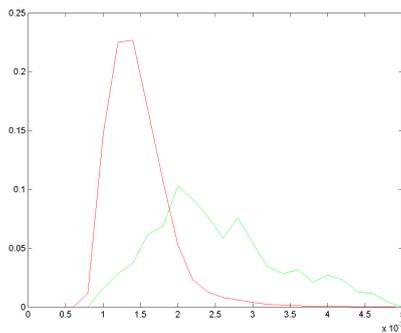


FIGURE 3.56 – distribution des scores agrégés matchants (vert) et non matchants (rouge) pour les grands MCC

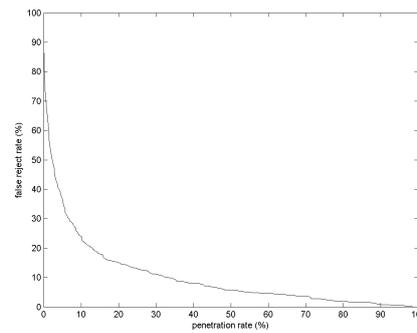


FIGURE 3.57 – distribution des scores agrégés matchants (vert) et non matchants (rouge) pour les grands MCC

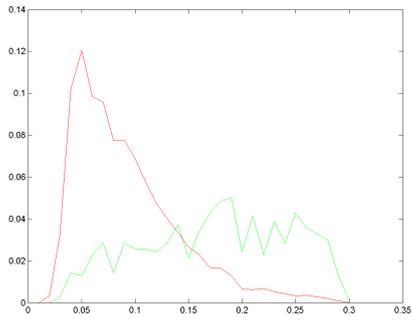


FIGURE 3.58 – distribution des scores agrégés matchants (vert) et non matchants (rouge) pour les petits MCC

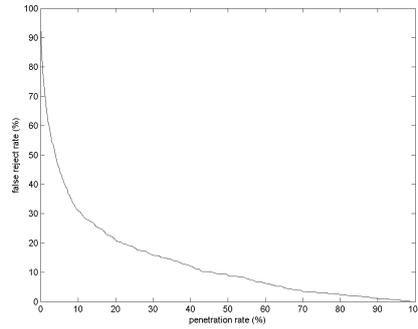


FIGURE 3.59 – taux de pénétration vs taux de faux rejets pour le score agrégé petits MCC

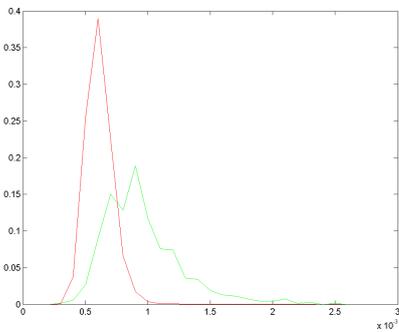


FIGURE 3.60 – distribution des scores agrégés matchants (vert) et non matchants (rouge) pour les patches de carte d'orientation

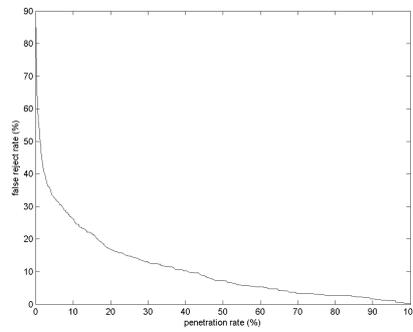


FIGURE 3.61 – taux de pénétration vs taux de faux rejets pour le score agrégé patches de carte d'orientation

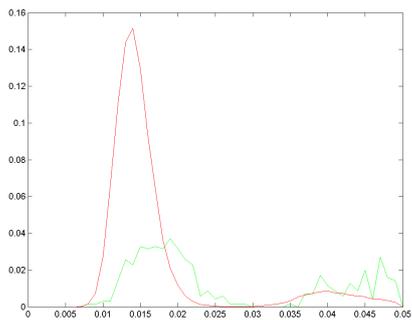


FIGURE 3.62 – distribution des scores agrégés matchants (vert) et non matchants (rouge) pour les représentations parcimonieuses de cartes d'orientation

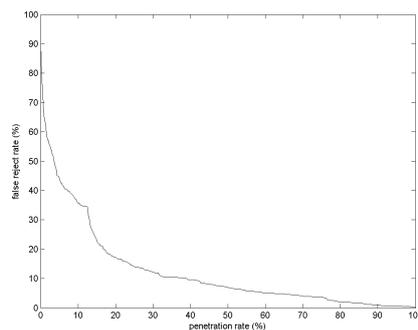


FIGURE 3.63 – taux de pénétration vs taux de faux rejets pour le score agrégé représentations parcimonieuses de cartes d'orientation

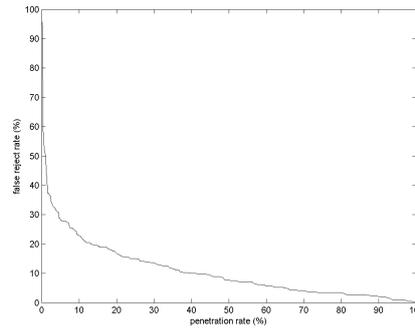
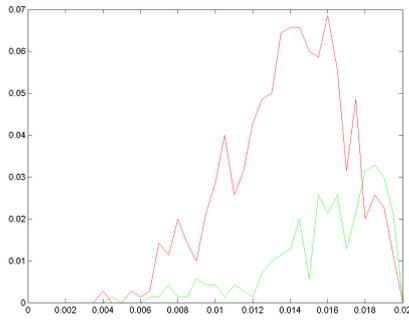


FIGURE 3.64 – distribution des scores agrégés pour faux rejets pour le score agrégé triplets de minutie (vert) et non matchantes (rouge) pour faux rejets pour le score agrégé triplets de minutie

On constate à nouveau une amélioration générale des performances pour l'ensemble des descripteurs, de manière plus nette que pour la distance agrégée corrigée. L'effet de seuillage apporté par l'agrégation en similarité plutôt qu'en dissimilarité sépare mieux les couples d'images matchantes et non matchantes.

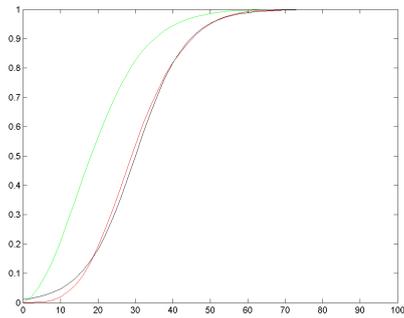
3.5.5 Score *a contrario*

Le score *a contrario* est défini en (3.18) comme suit :

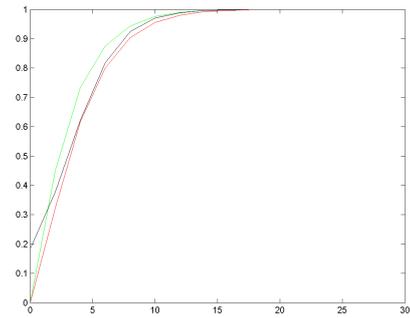
$$S_{ac}(I_1, I_2) = \sum_i \frac{-\log(f(\min_j d(z_{1,1}, z_{2,j})))}{\sum_k m_{ik}}$$

où f est la fonction de répartition de H_0

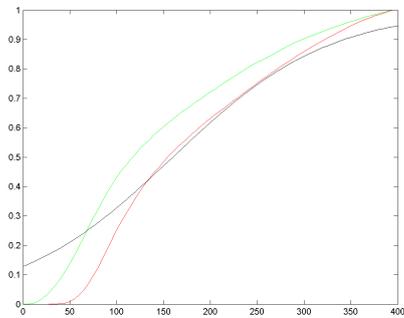
Les fonctions de répartition approchées utilisées pour les différents descripteurs sont détaillées dans les figure 3.66



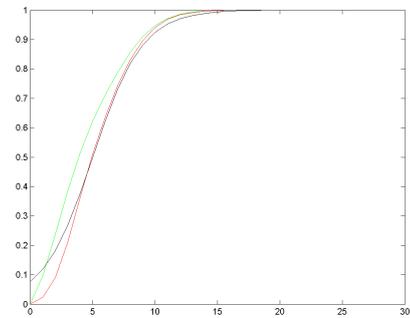
(a) fonction de répartition de la distance entre meilleurs appariements images matchantes (vert), non matchantes (rouge) et fonction f utilisée pour le score a contrario (noir), grands MCC



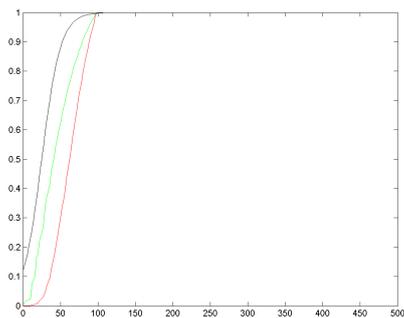
(b) fonction de répartition de la distance entre meilleurs appariements images matchantes (vert), non matchantes (rouge) et fonction f utilisée pour le score a contrario (noir), petits MCC



(c) fonction de répartition de la distance entre meilleurs appariements images matchantes (vert), non matchantes (rouge) et fonction f utilisée pour le score a contrario (noir), patches



(d) fonction de répartition de la distance entre meilleurs appariements images matchantes (vert), non matchantes (rouge) et fonction f utilisée pour le score a contrario (noir), coeffs. parcimonieux



(e) fonction de répartition de la distance entre meilleurs appariements images matchantes (vert), non matchantes (rouge) et fonction f utilisée pour le score a contrario (noir), triplets de minutes

FIGURE 3.66 – détails des fonctions de répartition

Les distributions de distances agrégées entre couples d'images matchantes et non matchantes ainsi que les performances obtenues en termes de taux de pénétration versus taux de faux rejets sont détaillés dans les figures 3.67 à 3.76.

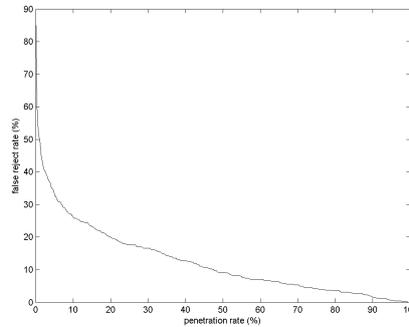
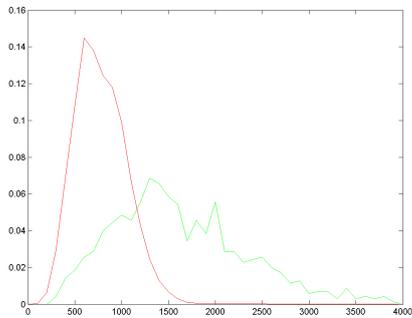


FIGURE 3.67 – distribution des scores a contrario matchants (vert) et non matchants (rouge) pour les grands MCC

FIGURE 3.68 – taux de pénétration vs taux de faux rejets pour le score a contrario grands MCC

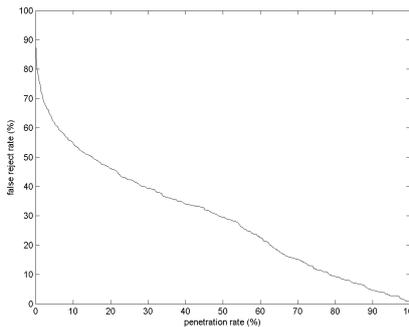
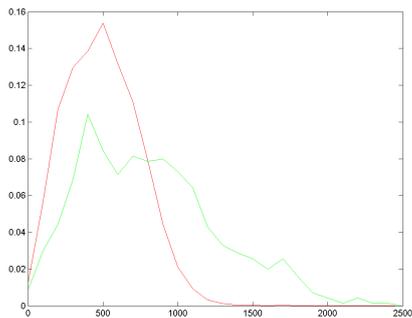


FIGURE 3.69 – distribution des scores a contrario matchants (vert) et non matchants (rouge) pour les petits MCC

FIGURE 3.70 – taux de pénétration vs taux de faux rejets pour le score a contrario petits MCC

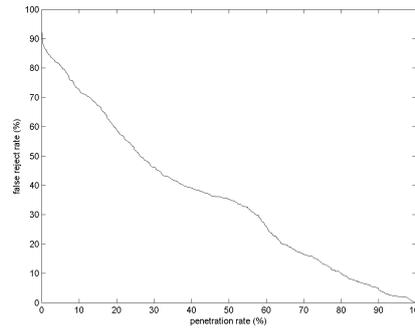
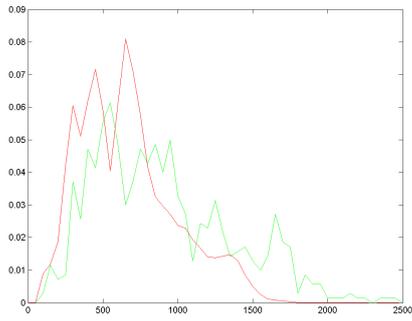


FIGURE 3.71 – distribution des scores a contrario matchants (vert) et non matchants (rouge) pour faux rejets pour le score a contrario patches de les patches de carte d'orientation

FIGURE 3.72 – taux de pénétration vs taux de faux rejets pour le score a contrario patches de carte d'orientation

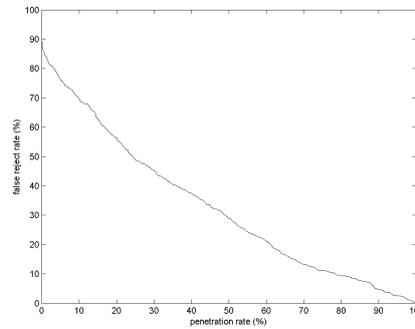
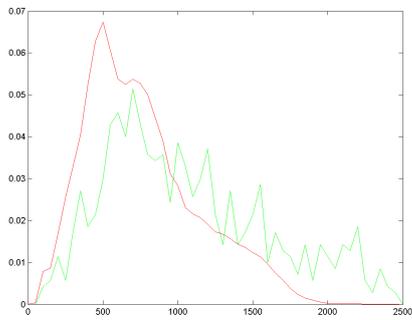


FIGURE 3.73 – distribution des scores a contrario matchants (vert) et non matchants (rouge) pour les représentations parcimonieuses de cartes d'orientation

FIGURE 3.74 – taux de pénétration vs taux de faux rejets pour le score a contrario représentations parcimonieuses de cartes d'orientation

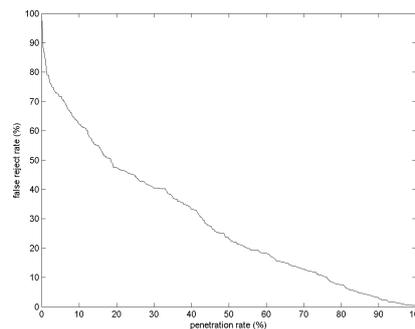
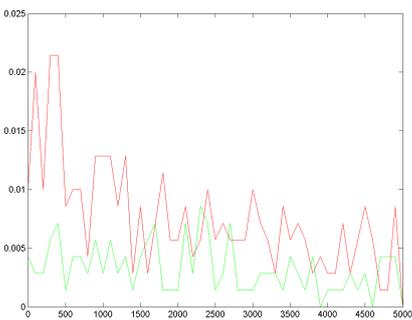


FIGURE 3.75 – distribution des scores a contrario matchants (vert) et non matchants (rouge) pour faux rejets pour le score a contrario triplets de minutie

FIGURE 3.76 – taux de pénétration vs taux de faux rejets pour le score a contrario triplets de minutie

Les performances observées sont médiocres pour l'ensemble des descripteurs à l'exception des grands MCC, surtout comparées au score agrégé. Ces mauvaises performances s'expliquent par un effet non pris en compte dans le score *a contrario* : ce score insiste sur la reconnaissance du caractère non matchant d'un appariement local. Il identifie avec certitude des appariements non matchants, mais identifie mal les appariements matchants, à l'exception de ceux présentant une très forte similarité. Le score agrégé étant fondamentalement une moyenne des similarités locales, les scores entre images matchantes vont être insuffisamment distingués des scores entre images non matchantes.

3.5.6 Effets des transformées de Hough

Un des intérêts de la transformée de Hough généralisée telle que décrite en chapitre 2, est de réduire la liste des appariements de descripteurs locaux parmi lesquels on recherche les plus petites distances.

On s'attend à ce que, dans le cas de couples d'images matchantes, les appariements validant la rotation affine sélectionnée par la transformée de Hough généralisée soient aussi les appariements minimisant les distances entre descripteurs locaux. L'application de la transformée aura donc peu d'impact sur la distribution des distances agrégées matchantes.

A l'inverse, dans le cas de couples d'images non matchantes, le choix de la meilleure rotation affine par la transformée de Hough est supposée aléatoire, et donc les appariements locaux minimisant la distance entre MCC n'auront qu'une probabilité réduite d'être validé par la rotation affine ainsi sélectionnée.

Si on revient sur la figure 4.32, les appariements effectivement sélectionnés pour construire la distance agrégée auront ainsi une distribution de distance plus proche de la distribution rouge pointillé que rouge plein.

Nous avons repris les algorithmes de calcul des trois variantes de la transformée de Hough généralisée présentée en partie 2.3.2.

Les résultats obtenus avec les différentes transformées de Hough sur les grands MCC en distance agrégée sont synthétisés dans les figures 3.77 et 3.78. Les courbes bleues correspondent à la transformée de Hough avec seuillage, les vertes à celle avec meilleur appariement, les rouges avec N meilleurs appariements. Sur les distributions, les courbes pleines correspondent aux appariements d'images matchantes, les pointillés aux appariements d'images non matchantes.

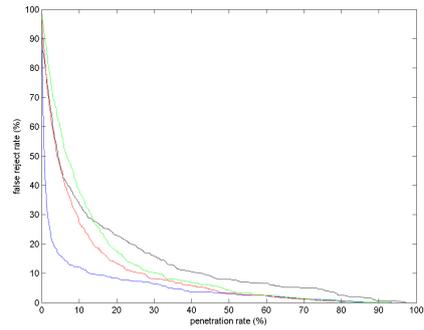
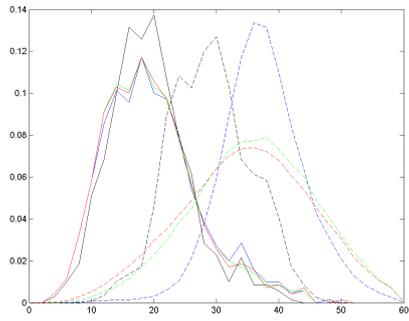
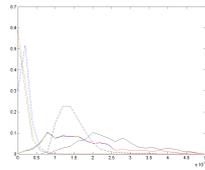


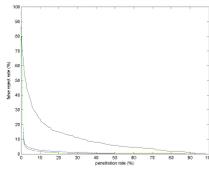
FIGURE 3.77 – distribution des distances agrégées avec grands MCC pour les différentes transformées de Hough

FIGURE 3.78 – performances taux de pénétration vs taux de faux rejets avec grands MCC pour les différentes transformées de Hough

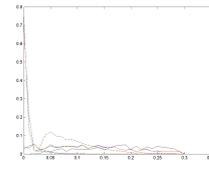
Les résultats en scores agrégés étant globalement ceux présentant les meilleures performances en termes de taux de pénétration vs taux de faux rejets, nous avons synthétisés les résultats obtenus en ajoutant les différentes transformées de Hough aux scores agrégés pour les différents types de descripteurs, dans les figures 3.79a à 3.79j.



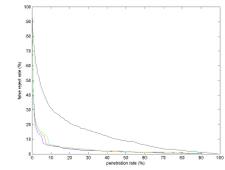
(a) distribution des scores agrégés avec grands MCC pour les différentes transformées de Hough



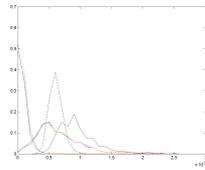
(b) distribution des scores agrégés avec grands MCC pour les différentes transformées de Hough



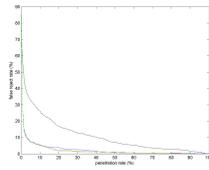
(c) distribution des scores agrégés avec petits MCC pour les différentes transformées de Hough



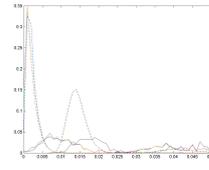
(d) distribution des scores agrégés avec petits MCC pour les différentes transformées de Hough



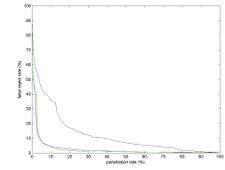
(e) distribution des scores agrégés avec patches de flux de crêtes pour les différentes transformées de Hough



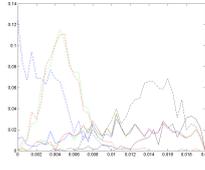
(f) distribution des scores agrégés avec patches de flux de crêtes pour les différentes transformées de Hough



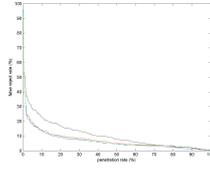
(g) distribution des scores agrégés avec représentation parcimonieuse des patches pour les différentes transformées de Hough



(h) distribution des scores agrégés avec représentation parcimonieuse des patches pour les différentes transformées de Hough



(i) distribution des scores agrégés avec triplets pour les différentes transformées de Hough



(j) distribution des scores agrégés avec triplets pour les différentes transformées de Hough

Sur l'ensemble des descripteurs, on constate que l'ajout de l'information de cohérence globale par la transformée de Hough apporte un net gain global en qualité de reconnaissance. On remarque également que les performances varient peu entre les différents types de transformée de Hough, et que les meilleures performances sont obtenues pour les transformées de Hough avec meilleur appariement ou N meilleurs appariements, plutôt qu'avec la transformée avec seuillage, qui aboutit à faire voter plus d'appariements dans l'espace des similitudes.

Ce résultat s'explique essentiellement par le fait que dans la grande majorité des cas, les appariements de la vérité terrain sont très souvent les meilleurs ou parmi les N meilleurs appa-

riements entre deux images matchantes. Faire voter plus d'appariements n'aboutit qu'à accroître la probabilité de fausses acceptances.

Ce résultat est très intéressant, car il signifie que l'ajout de cohérence globale peut se faire à un coût en temps de calcul en $O(n_1)$, plutôt qu'en $O(n_1 \times n^2)$. Il signifie également que lors d'un processus de recherche accélérée qui ne calculerait explicitement que les distances locales entre les meilleurs appariements, l'utilisation d'information de cohérence globale resterait pertinente, même restreinte aux distances effectivement calculées.

3.5.7 Score en proportion de Hough

Afin de pouvoir évaluer l'apport de la cohérence globale par rapport à la seule information locale et aux effets heuristiques de la correction de distance ou du passage d'une dissimilarité à une similarité, nous avons construit trois derniers scores, simplement définis par la proportion des meilleurs appariements tombant effectivement dans le meilleur cluster de chaque transformée de Hough.

Les résultats obtenus sur les différents descripteurs sont synthétisés dans les figures 3.80 à 3.89.

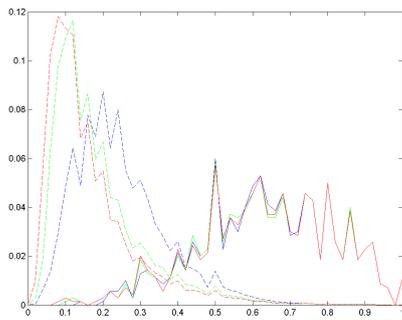


FIGURE 3.80 – distribution des scores en proportion de Hough matchants (vert) et non matchants (rouge) pour les grands MCC

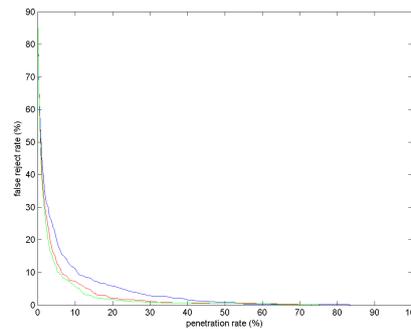


FIGURE 3.81 – taux de pénétration vs taux de faux rejets pour le score en proportion de Hough grands MCC

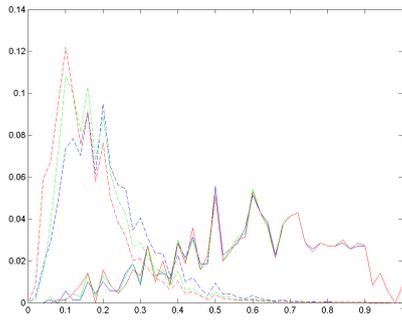


FIGURE 3.82 – distribution des scores en proportion de Hough matchants (vert) et non matchants (rouge) pour les petits MCC

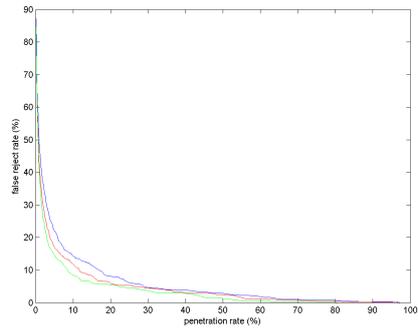


FIGURE 3.83 – taux de pénétration vs taux de faux rejets pour le score en proportion de Hough petits MCC

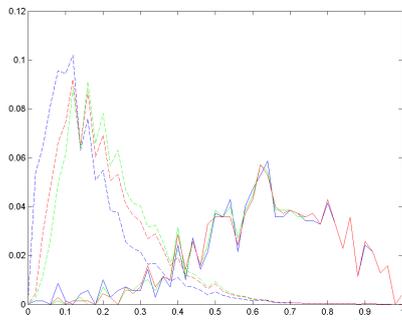


FIGURE 3.84 – distribution des scores en proportion de Hough matchants (vert) et non matchants (rouge) pour les patches de carte d'orientation

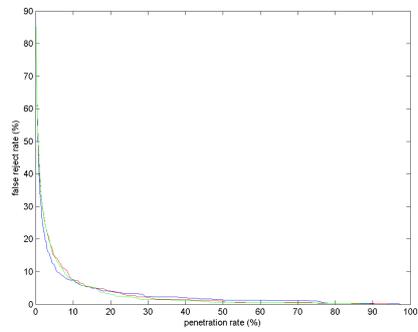


FIGURE 3.85 – taux de pénétration vs taux de faux rejets pour le score en proportion de Hough patches de carte d'orientation

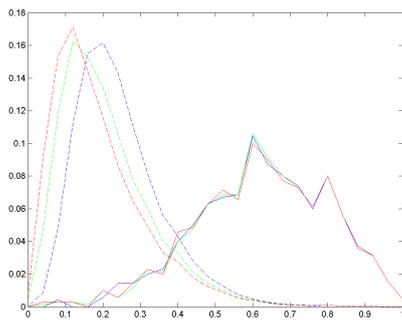


FIGURE 3.86 – distribution des scores en proportion de Hough matchants (vert) et non matchants (rouge) pour les représentations parcimonieuses de cartes d'orientation

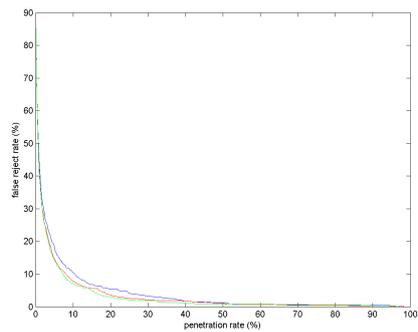


FIGURE 3.87 – taux de pénétration vs taux de faux rejets pour le score en proportion de Hough représentations parcimonieuses de cartes d'orientation

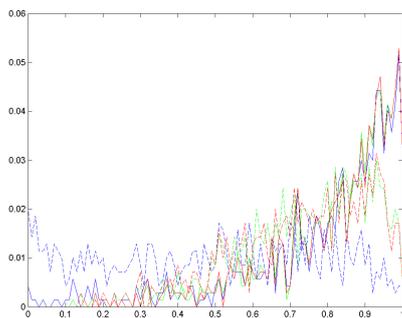


FIGURE 3.88 – distribution des scores en proportion de Hough matchants (vert) et non matchants (rouge) pour les triplets de minutie

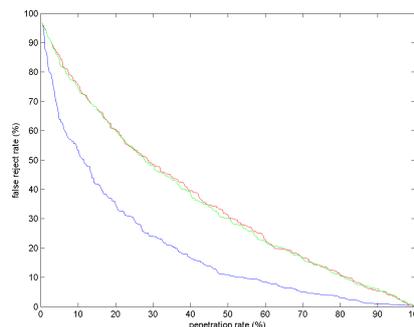


FIGURE 3.89 – taux de pénétration vs taux de faux rejets pour le score en proportion de Hough triplets de minutie

Les courbes précédentes montrent que l'information de cohérence globale ajoutée par les transformées de Hough constitue intrinsèquement une information très pertinente, permettant d'atteindre de bonnes performances en reconnaissance d'empreinte.

3.5.8 Comparaison avec l'état de l'art

Un état de l'art des performances de différents algorithmes de comparaison d'empreintes digitales a été réalisé par [132]. A ce stade de notre travail, nous nous comparons aux méthodes de l'état de l'art évaluées sur la base FVC2000 DB2, que nous avons utilisée pour construire nos algorithmes en force brute. Les méthodes sont celles proposées par :

- Leung and Leung (2011) [133] Ces travaux proposent un algorithme de comparaison par classificateur bayésien, à partir d'une variante du FingerCode.
- Jiang et al. (2006) [76] Ces travaux proposent également une variante de FingerCode, combinée avec un calcul de distance moyenne inter-crêtes, avec une distance *ad hoc* entre descripteurs.
- Liang et al. (2006) [92] Ces travaux présentent une variante de triplets de minuties
- De Boer et al. (2001) [134] Ces travaux proposent des descripteurs de type FingerCode, carte d'orientation et triplets de minuties, ainsi que la fusion de ces descripteurs.
- Cappelli et al. (2011) [125] Ces travaux sont ceux présentant la première implémentation accélérée des MCC par LSH
- Cappelli et al. (2012) [132] Ces travaux présentent la combinaison de descripteurs locaux à base de MCC, avec des descripteurs locaux à base de carte centrée de flux de crêtes et de fréquences de crêtes.

Les travaux faisant usage de descripteurs globaux sont présentés à titre purement indicatifs. En effet, les descripteurs globaux utilisés sont par nature non robustes à des requêtes de type latente.

Les performances des différents algorithmes sont synthétisées dans le graphique 3.90 et le tableau 3.11

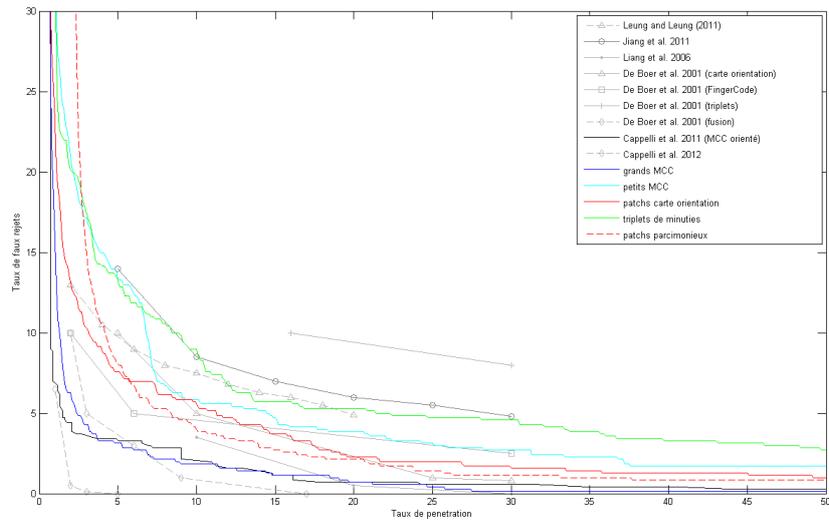


FIGURE 3.90 – Performances en reconnaissance observées sur la base FVC2000 DB2 pour différents algorithmes de l'état de l'art

Méthodes de l'état de l'art	taux d'erreur à 20% de taux de pénétration
Cappelli et al. (2011)	5
Leung and Leung (2011)	2.5
Jiang et al. (2006)	6
Liang et al. (2006)	0.5
De Boer et al. (2001) - directional field	5
De Boer et al. (2001) - FingerCode	9.5
De Boer et al. (2001) - triplets	5
De Boer et al. (2001) - fusion	5
Cappelli et al. (2012)	0
Méthodes présentées	taux d'erreur à 20% de taux de pénétration
grands MCC - score agrégé simple, Hough avec meilleur appariement	0.9
petits MCC - idem	4
patchs de carte d'orientation - idem	2.5
patchs parcimonieux - idem	2
triplets de minuties - idem	5.5

TABLE 3.11 – Comparaison des performances en précision avec l'état de l'art

3.5.9 Conclusion sur les méthodes en force brute

Les conclusions que nous pouvons tirer des résultats obtenus en force brute sont les suivantes.

Il est possible de construire des descripteurs locaux pour empreinte digitale dotés d'un fort pouvoir discriminant à l'échelle locale. Cependant, pour atteindre des performances en reconnaissance suffisante pour la problématique qui nous intéresse, il s'avère nécessaire de consolider l'information locale avec de l'information de cohérence globale via les transformées de Hough généralisées. A ce stade, on peut alors se demander à quel point l'utilisation de l'information globale s'avère utile par rapport à des algorithmes de comparaison qui feraient appel uniquement à la transformée de Hough.

En effet, si l'utilisation de l'information locale au préalable permet de réduire le coût de la transformée à un ordre $O(n_1)$, plutôt que $O(n_1 \times n_2)$, le calcul de l'information locale a lui-même un coût en $O(n_1 \times n_2)$.

L'utilisation de l'information locale se justifie pour deux raisons majeures. D'abord, la robustesse aux requêtes de mauvaise qualité, car les transformées de Hough nécessitent de faire voter un grand nombre d'appariements pour être fiables, donc de disposer d'une grande quantité d'information fiable en requête. L'information propre aux descripteurs locaux accroît la ro-

bustesse au manque d'information. Ensuite, les descripteurs locaux offrent la possibilité d'une indexation purement "locale", c'est-à-dire sans référencement de l'image d'appartenance de chaque descripteur. Or, les méthodes de recherche accélérée dans des bases de descripteurs locaux, que nous allons analyser au chapitre 4, permettent d'atteindre des coûts de recherche inférieurs à ceux d'une transformée de Hough pure.

Chapitre 4

Accélération

Parmi les méthodes de comparaison d'empreintes digitales décrites jusqu'ici, les critères de choix prioritaires étaient la robustesse au bruit et aux occultations des requêtes, mais aussi la possibilité d'accélérer la recherche pour les représentations choisies.

Les algorithmes d'identification d'empreintes digitales sont par essence des algorithmes de recherche des plus proches voisins, et un algorithme d'identification peut être évalué selon sa capacité à faire apparaître l'empreinte ou les empreintes matchantes d'une empreinte requête au sein d'une base de données parmi ses plus proches voisins. Cette évaluation correspond à la courbe *CMC* décrite au chapitre 1.

Dans ce chapitre, nous n'abordons pas la recherche de plus proches voisins à l'échelle des images entières, pour une évaluation de type *CMC*, mais nous nous intéressons à la recherche de plus proches voisins à l'échelle locale. Nous développons les idées présentées dans la section 2.4 afin d'utiliser les méthodes de recherche accélérée, construites notamment pour des descripteurs locaux comme le SIFT, le SURF ou l'ORB pour réduire le temps de parcours de la base de données.

Nous allons donc exposer les différentes méthodes de recherche accélérée de l'état de l'art que nous avons choisies, et comment nous les avons adaptées aux descripteurs locaux pour empreintes digitales que nous avons analysés au chapitre 3.

4.1 Généralités sur les méthodes de recherche de plus proches voisins

[39] définit le problème de recherche des plus proches voisins de la façon suivante : soit $P = \{x_1, x_2, \dots, x_n\}$ un ensemble de points dans un espace métrique M doté d'une distance $d : M \times M \leftarrow \mathbb{R}$ et une requête $q_i \in M$, on cherche $NN(q, P)$ vérifiant :

$$NN(q, P) = \operatorname{argmin}_{x \in P} d(q, x)$$

Le problème des plus proches voisins consiste alors à trouver une méthode de structuration de P permettant de procéder à l'opération de calcul de $\operatorname{argmin}_{x \in P} d(q, x)$ plus efficace qu'un simple parcours exhaustif de P .

Le problème des plus proches voisins peut être généralisé de différentes façons. Le *problème des K plus proches voisins* consiste à trouver non seulement le plus proche voisin de la requête, mais aussi les plus proches suivant. Le *problème des plus proches voisins à distance R* consiste à trouver les voisins de q situés à une distance inférieure à R de q .

Ces deux problèmes peuvent être formalisés de la façon suivante : *problème des K plus proches voisins*

$$KNN(q, P, K) = A$$

où A vérifie

$$|A| = K, A \subseteq P \forall x \in A, y \in PA, d(q, x) \leq d(q, y)$$

problème des plus proches voisins à distance R

$$RNN(q, P, R) = \{p \in P, d(p, q) > R\}$$

Une recherche des plus proches voisins à distance R est susceptible de retourner n'importe quel nombre de points, de zéro à la totalité de la base de données. Pour placer une limite au nombre de points retournés, on peut considérer une *recherche des plus proches voisins à distance R* :

$$RKNN(q, P, K, R) = A,$$

où A vérifie

$$\begin{aligned} |A| &\leq K, A \subseteq P \\ \forall x \in A, y \in PA, d(q, x) &\leq d(q, y) \end{aligned}$$

Comme expliqué dans la section 2.4, les méthodes de recherche approximative de plus proches voisins, destinées à accélérer le processus de recherche via des structures de bases de données *ad hoc*, permettent d'accélérer la recherche par rapport à des méthodes parcourant de manière exhaustive les bases de données.

Dans la section 2.4, les principales méthodes de recherche de plus proches voisins approximatives reposent toutes sur une procédure de clusterisation de la base de données, combinées à des index inversés

Le concept d'index inversé de *mots visuels*, adapté des méthodes d'indexation de texte destinées initialement à référencer les couples (*texte, mot*) par la clé d'indexation du mot plutôt que celle du texte, a été adapté aux algorithmes de CBIR notamment par les travaux de [32], poursuivis par [135] et [136].

L'approche que nous avons choisie pour construire un algorithme de recherche accélérée est inspirée du processus de recherche par descripteurs locaux SIFT pour des images génériques tel que décrit par [14] et [15].

Cette approche est résumée comme suit :

On dispose d'une base d'empreintes digitales de références $I_{r1}, \dots, I_{rj}, \dots, I_{rN}$, avec leurs descripteurs locaux respectifs $(\{z_{r1i}\}_{1 \leq i \leq n_{r1}}, \dots, \{z_{rNi}\}_{1 \leq i \leq n_{rN}})$.

On note $N_T = \sum_{j=1}^N n_{rj}$ le nombre total de descripteurs locaux références.

La requête d'entrée est notée I_s avec ses descripteurs locaux $\{z_{si}\}_{1 \leq i \leq n_s}$

On note $\hat{S} = [\hat{s}_{u,v}]_{1 \leq u \leq n_s, 1 \leq v \leq N_T}$ le tableaux des similarités locales approchées entre les descripteurs locaux requêtes et références.

Le processus de recherche est détaillé dans le pseudo-code suivant 6 et illustré dans les figures 4.1 et 4.2 :

Algorithm 6 Recherche accélérée dans une base d'empreintes digitales

```

1. On initialise le tableau  $\hat{S}$  à zéro
for all  $u \in [1, n_s]$  do
  for all  $v \in [1, N_T]$  do
     $[u, v] \leftarrow 0$ 
  end for
end for
2. On cherche les plus proches voisins de chaque descripteur requête
for all  $u \in [1, n_s]$  do
   $q_1, \dots, q_K$  sont les indices des  $K$  plus proches voisins de  $z_u$  parmi les
   $(\{z_{r1i}\}_{1 \leq i \leq n_{r1}}, \dots, \{z_{rNi}\}_{1 \leq i \leq n_{rN}})$ 
   $\hat{simil}_{u,q_1}, \dots, \hat{simil}_{u,q_K}$  sont leurs similarités approchées respectives estimées par
  l'algorithme de  $K$  nearest neighbors
  for all  $k \in [1, K]$  do
     $s_{i,q_k} \leftarrow \hat{simil}_{u,q_k}$ 
  end for
end for
3. On procède au calcul des scores agrégés locaux, éventuellement consolidés avec
la transformée de Hough généralisée avec meilleurs appariements, avec les similarités
définies en section 3.4.5

```

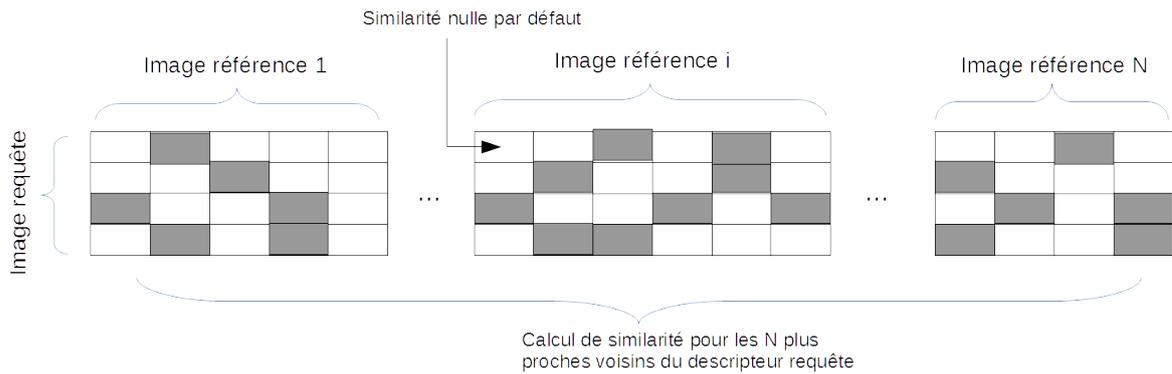


FIGURE 4.1 – Calcul des K plus proches voisins des descripteurs locaux requêtes et remplissage de la matrice des similarités entre descripteurs locaux. Les cases grises représentent par ligne ces K plus proches voisins associés au descripteur local de la ligne

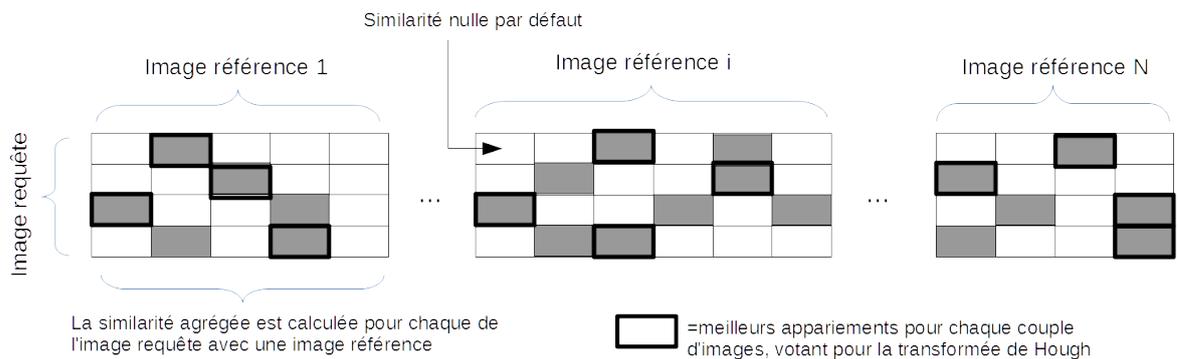


FIGURE 4.2 – A partir de la matrice des similarités locales, calcul des similarités agrégées entre chaque couple d’empreintes, et éventuellement consolidation avec transformée de Hough avec meilleurs appariements pour chaque couple d’empreinte également

4.1.1 Méthodes de recherche accélérée

[39] classe les principales méthodes de recherche de plus proches voisins en trois catégories : les arbres de classification, les méthodes de hachage, et les graphes de voisinages. Parmi les arbres de classification, on trouve notamment les Kd-trees et le Hierarchical K-means, et parmi les méthodes de hachage on trouve notamment le Locality Sensitive Hashing, méthodes qui ont également été analysées en détail par [42]. Comme décrit en section 2.4.2, nous nous sommes focalisés sur ces trois méthodes.

Détails sur le Kd-tree

La structure d’indexation qu’est le Kd-tree a été précédemment définie dans la section 2.4.2. Depuis les travaux originels de [41] et [40], le Kd-tree a été amélioré en termes de coûts de recherche par deux modifications significatives.

Dans l’algorithme de recherche initial du plus proche voisin exact par un kd-tree, le plus proche voisin est très souvent visité avant que l’algorithme n’arrive à terminaison [137], l’algorithme continuant de visiter un certain nombre d’autres noeuds avant de confirmer le plus proche voisin trouvé. Ce temps de recherche supplémentaire est lié à la vérification que le noeud trouvé correspondant est bien le plus proche voisin. Si l’usager est prêt à renoncer à la garantie d’avoir trouvé le plus proche voisin réel, c’est-à-dire dans le cas d’une recherche *approximative* du plus proche voisin, alors le processus de recherche peut être interrompu plus tôt que dans le cas d’une recherche exacte, par exemple en interrompant la recherche après la visite d’un nombre prédéterminé de noeuds.

Pour maximiser les probabilités de trouver effectivement le plus proches voisins parmi les premiers noeuds visités, [137] propose de procéder par *recherche prioritaire (priority search)*, à l’aide d’une *liste de priorité*. Dans cette heuristique, à chaque étape de la recherche dans le kd-tree, le prochain noeud à être visité est le noeud le plus proche de la requête non encore visité.

Le principal défaut de la recherche prioritaire relevé par [138], est que les visites successives de noeuds dans un kd-tree ne sont pas indépendantes. Si la probabilité de trouver le plus proche

voisin parmi les n premiers noeuds visités augmente avec n , la probabilité de trouver le plus proche voisin au n -ème noeud s'il ne figure pas parmi les $n - 1$ précédents décroît fortement avec n .

Pour contourner ce problème et réduire la dépendance entre les noeuds successifs explorés, [138] propose une amélioration de la sélection de la dimension choisie pour la dichotomie de l'ensemble des données associées à chaque noeud lors de la construction du kd-tree. Dans l'algorithme initial, la dimension choisie est celle présentant la plus forte variance des données associées au noeud. [138] propose de choisir la dimension utilisée pour la découpe au hasard parmi les D dimensions présentant la plus grande variabilité des données, construisant ainsi des *kd-trees aléatoires (random kd tree)*.

Les kd-trees que nous utiliserons pour l'indexation des descripteurs locaux d'empreintes digitales correspondent à des random kd-tree avec recherche approximative par priorité. Nous utiliserons pour cela l'implémentation de la librairie FLANN, décrite dans [139] et disponible sur [140].

Les principaux algorithmes de recherche et d'insertion de points dans un kd-tree sont détaillés en annexe 4.8

Hierarchical K-means et LSH

Les HKM et le LSH de type bit-sample que nous utiliserons pour l'indexation de descripteurs locaux d'empreintes digitales sont ceux définis par [139] et [39], implémenté dans la librairie FLANN de [140].

4.1.2 Comparaison des coûts et contraintes théoriques

Le choix de la méthode de recherche des plus proches voisins doit être réalisé selon les critères retenus pour notre problématique, à savoir :

- atteindre de très faibles taux de faux rejets à des taux de pénétration raisonnables
- avoir un coût de comparaison unitaire le plus faible possible
- limiter le coût de stockage des signatures d'empreinte

A ce stade, nous faisons l'hypothèse que l'utilisation de descripteurs locaux comme signatures assure une certaine robustesse au bruit et à l'occultation sur les images requêtes. Cette hypothèse devra être vérifiée par la suite sur des requêtes latentes.

Un comparatif assez exhaustif des coûts de comparaison et de stockage des différentes méthodes de recherche de plus proches voisins a été réalisé par [42], que nous reprenons en partie dans le tableau 4.2, avec les paramètres détaillés dans le tableau 4.1.

Paramètre	Description
l	nombre d'image
s	octets/dimension du descripteur
d	dimension du descripteur
F	nombre de descripteurs par image
T_{kdt}	nombre de kd-trees
L	longueur des listes triées
B_{kdt}	nombre de rappels
T_{lsh}	nombre de tables de hachage
H_{lsh}	nombre de fonctions de hachage LSH
B_{lsh}	nombre de buckets de LSH
D	profondeur de l'arbre HKM
k	facteur de branche de l'arbre HKM
W	nombre de mots du BoW
T_{lsh}	nombre de tables pour le LSH

TABLE 4.1 – Paramètres entrant dans les coûts de la recherche des k plus proches voisins

Méthode	stockage (octets)	calcul
Recherche exhaustive	$(sd + 4)IF$	$F^2I(2d + 1)$
Kd-tree	$IF(sd + 4 + 2T_{kdt} + T_{kdt} \frac{\log_2 IF}{8})$	$B_{kdt}F(2d + 1 + \log_2 FI)$
HKM	$IF(sd + 4) + \frac{k^D - 1}{k - 1}ksd$	$FD(2d + k) + \frac{F^2I}{k^D} \times (2d + 1)$
LSH	$IF(sd + 4 + T_{lsh} \frac{\log_2 IF}{8})$	$FT_{lsh}(H_{lsh}(2d + 2) + \frac{FI}{B_{lsh}}(2d + 1))$

TABLE 4.2 – Coûts théoriques des différentes méthodes de recherche des k plus proches voisins

[42] propose également un état de l'art des performances des principales méthodes d'ANN en termes de performances. Nous complétons ces informations avec les analyses de [139] et [39].

D'une manière générale, les considérations suivantes doivent être prises en compte :

- Les représentations à base de BoW présentent des performances en reconnaissance significativement moindres que les représentations par ensemble de descripteurs locaux
- Les représentations de type BoW présentent l'avantage d'un temps de recherche quasi constant quelle que soit la taille de la base de données à rechercher.
- La majorité des autres méthodes présentent un coût de recherche linéaire avec la taille de la base, à l'exception des kd-trees, qui présentent un coût de recherche logarithmique en la taille de la base de données.

Etat de l'art des méthodes de recherche par plus proches voisins appliquées à l'empreinte digitale

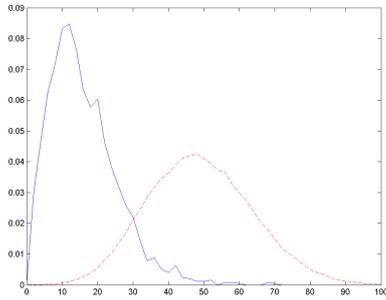
Différentes méthodes de recherche accélérée ont déjà pu être testées dans l'état de l'art des techniques de comparaison d'empreintes digitales. Ainsi, l'utilisation de kd-trees pour l'indexation de triplets de minuties a été proposée par [141], [142] et plus récemment par [143]. Le LSH a été proposé pour l'indexation de descripteurs de type SIFT [106], de cartes locales de minuties [125] ou de triplets de minuties [144].

4.2 Application aux descripteurs d'empreinte digitale choisis

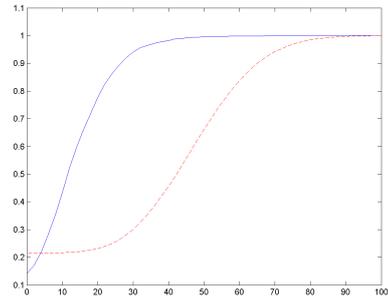
4.2.1 Estimation du nombre de plus proches voisins optimal

Deux paramètres communs à l'ensemble des méthodes de recherche de plus proches voisins, et quel que soit le descripteur local choisi, sont le nombre de plus proches voisins maximum sélectionnés pour chaque descripteur local requête, et le rayon maximal de recherche.

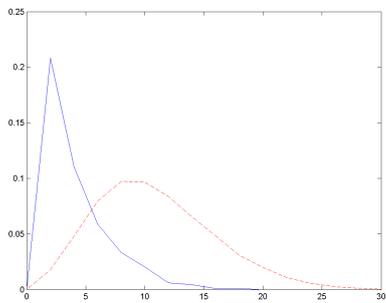
Les pseudo-distances entre descripteurs locaux définies dans le cadre du chapitre 3 pour calculer des distances et scores agrégés, sont les mêmes que celles utilisées par les algorithmes de recherche rapide. Aussi, pour déterminer les seuils de recherche en nombre de voisins, nous pouvons reprendre les distributions de distances entre descripteurs locaux requêtes et références, dans le cas d'appariements vérité terrain d'une part, et d'appariements non matchants aléatoires d'autre part.



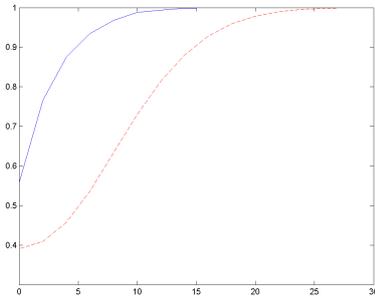
(a) distribution des distances entre grands MCC, appariements vérité terrain (bleu) et non matchants (rouge)



(b) fonctions de répartition associées aux distances entre grands MCC, appariements vérité terrain (bleu) et non matchants (rouge)

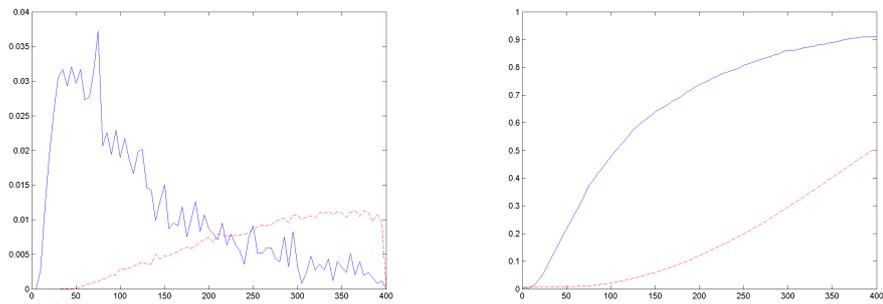


(c) distribution des distances entre petits MCC, appariements vérité terrain (bleu) et non matchants (rouge)



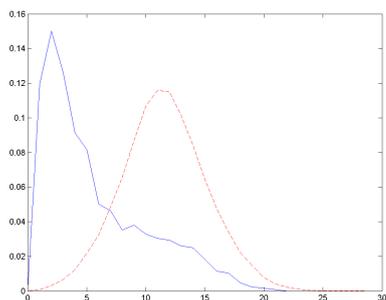
(d) fonctions de répartition associées aux distances entre grands MCC, appariements vérité terrain (bleu) et non matchants (rouge)

FIGURE 4.3 – distribution des distances entre triplets de minuties, appariements vérité terrain (bleu) et non matchants (rouge) - 1e partie

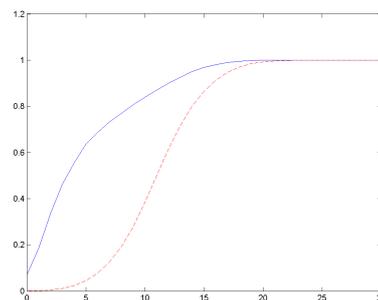


(a) distribution des distances entre patchs de flux de crêtes, appariements vérité terrain (bleu) et non matchants (rouge) (b) fonctions de répartition associées aux distances entre grands MCC, appariements vérité terrain (bleu) et non matchants (rouge)

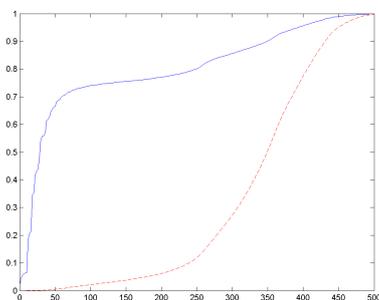
FIGURE 4.4 – distribution des distances entre triplets de minuties, appariements vérité terrain (bleu) et non matchants (rouge) - 2e partie



(a) distribution des distances entre représentations parcimonieuses de patches, appariements vérité terrain (bleu) et non matchants (rouge)



(b) fonctions de répartition associées aux distances entre grands MCC, appariements vérité terrain (bleu) et non matchants (rouge)



(c) distribution des distances entre triplets de minutes, appariements vérité terrain (bleu) et non matchants (rouge) - 3e partie

Nous faisons l'hypothèse que la base d'apprentissage FVC2000 DB 2 est suffisamment représentative de l'ensemble des empreintes digitales pour que les distributions de distances entre appariements matchants et non matchants varient peu par rapport à ceux observés sur cette base.

Pour l'ensemble des descripteurs locaux utilisés, il n'apparaît pas envisageable de fixer des seuils en distance ou en rang permettant de récupérer avec certitude ou quasi certitude la totalité des appariements matchants. En effet, les distributions de distances entre les deux types d'appariements, quel que soit le distributeur choisi, sont insuffisamment séparables, et la récupération de 90% à 99% des appariements matchants nécessiterait de conserver également 20 à 50% des appariements non matchants, ce qui réduirait considérablement l'intérêt d'une recherche accélérée des plus proches voisins par rapport à un calcul explicite de la distance de chaque descripteur requête à tous les descripteurs références.

Cependant, les résultats obtenus au chapitre 3 dans le cadre de distances et scores agrégés en force brute, tendent à montrer que les appariements matchants présentant une forte dissi-

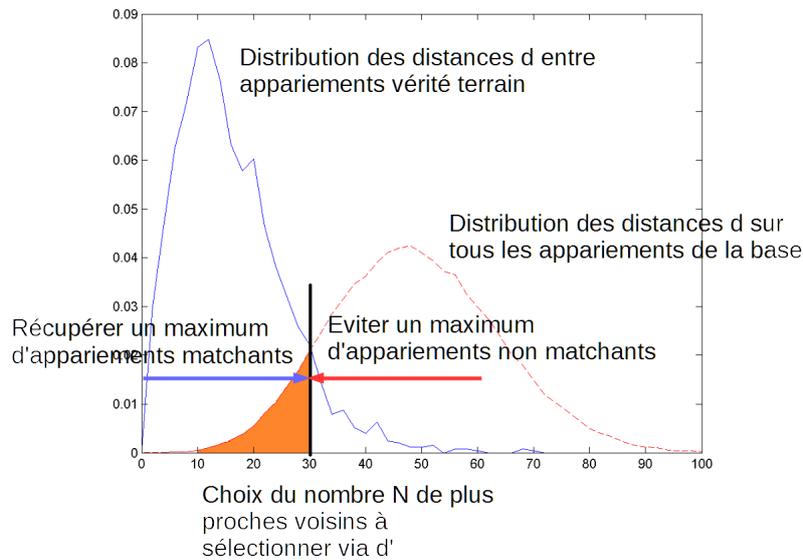


FIGURE 4.6 – Choix du nombre optimal de plus proches voisins à calculer effectivement

milarité contribuent peu à l’obtention de dissimilarités agrégées faibles entre couples d’images d’empreintes matchantes.

Aussi, nous pouvons délibérément choisir des seuils ne conservant qu’une fraction des appariements matchants, afin d’éviter de collecter un trop grand nombre d’appariements non matchants parmi les plus proches voisins, et envisager d’obtenir néanmoins des distances et scores agrégés pertinents, permettant de séparer les couples d’images matchantes des non matchantes.

La problématique du choix optimal de plus proches voisins à collecter est illustré par la figure 4.6.

Pour les différentes tailles de bases, nous avons considéré les nombres suivants de plus proches voisins définis dans le tableau 4.3, valeurs sélectionnées expérimentalement à partir des distributions d’appariements de descripteurs vérité terrain et de descripteurs non matchants

Bases	Nombre de minuties considérées dans les images de référence	Nombre plus proches voisins pour descripteurs minuties-centrés	Nombre plus proches voisins pour triplets de minuties
FVC 2000 DB 2	24 000	100	200
NIST DB4	170 000	200	1000
NIST DB 14	2 470 000	1000	2000

TABLE 4.3 – Nombre de plus proches voisins considérés selon la taille de la base et le type de descripteur local

4.2.2 Comparaison avec l'état de l'art

Pour l'ensemble des résultats qui suivent, nous effectuerons une comparaison avec les performances, en reconnaissance et en temps de calcul, obtenue avec le Sdk développé par [125]. Cet outil présente l'avantage de pouvoir reproduire les résultats de l'état de l'art avec les informations issues du même outil de codage que celui utilisé avec nos propres descripteurs.

Pour rappel, l'algorithme d'indexation de [125] est décrit dans les pseudocodes suivants : 7 (indexation) et 8 (recherche), les paramètres étant décrits dans le tableau 4.4.

Algorithm 7 Création d'index

Entrée : base de données DB d'empreintes digitales (I_1, \dots, I_m)
ensemble de fonctions de hachage bit-sampling $F = f_1, \dots, f_h, \dots, f_H$
Sortie : tables de hachages $\mathbb{H}_1, \dots, \mathbb{H}_h, \dots, \mathbb{H}_H$

```
for all  $I_i \in DB$  do  
  On construit l'ensemble  $V_i = \{v_{ij}\}_j$  des MCC de  $I_i$   
  for all  $v_{ij} \in V_i$  do  
    for all  $f_h \in F$  do  
       $b = f_h(v_{ij})$   
      Si le nombre de bits à 1 de  $b$  est  $\leq min_{PC}$ , alors on ajoute  $(i, j)$  dans le bucket  
       $b$  de la table  $\mathbb{H}_h$   
    end for  
  end for  
end for
```

Algorithm 8 Requête

Entrée : empreinte digitale requête I_s

base de données DB d'empreintes digitales (I_1, \dots, I_m)

ensemble de fonctions de hachage bit-sampling $F = f_1, \dots, f_h, \dots, f_H$

tables de hachages $\mathbb{H}_1, \dots, \mathbb{H}_h, \dots, \mathbb{H}_H$

Sortie : tableau de score de similarité \mathbb{S} de la requête I_s aux empreintes de la base de données

On initialise les valeurs de \mathbb{S} à zéro

On construit l'ensemble $V = \{v_j\}$ des MCC de I_s

for all $v \in V$ **do**

 On note m la minutie sur laquelle v est centré

 On initialise les valeurs de l'accumulateur de collision \mathbb{A} à zéro

for all $f \in F$ **do**

$b = f(v)$

if $\text{Popcount}(b) \geq \text{min}_{pc}$ **then**

for all couple (i, j) dans le bucket b de la table \mathbb{H} associée à f **do**

 On note m_j la j ème minutie de l'empreinte V_i

if m et m_j sont compatibles **then**

$\mathbb{A}[i, j] = \mathbb{A}[i, j] + 1$

end if

end for

end if

end for

for all V_i de DB présentant au moins une collision dans \mathbb{A} **do**

$CF_{max} = \max_j \{\mathbb{A}[i, j]\}$

$\mathbb{S}[i] = \mathbb{S}[i] + (CF_{max})^{\frac{p}{h}}$

end for

for all V_i de DB **do**

$\mathbb{S}[i] = \mathbb{S}[i] / (\#(V) \cdot l^{\frac{p}{h}})$

end for

end for

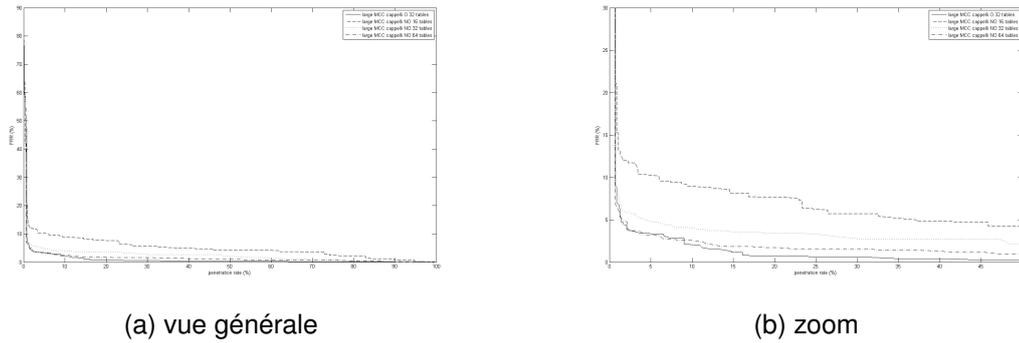


FIGURE 4.7 – performances en taux de pénétration vs taux de faux rejets obtenues avec divers réglages du Sdk MCC-LSH

Les performances présentées ont été obtenues en codant les MCC avec les mêmes paramètres que ceux des grands MCC présentés en 3.1

paramètre	description	valeur
n	nombre de cellules dans un MCC	312
h	nombre de bits sélectionnés par fonction de hachage	24
l	nombre de fonctions de hachage	32
p	paramètre de forme de la fonction de distance	30
min_{pc}	nombre minimum de bits à 1 pour fouiller un bucket	2
δ_{theta}	rotation maximale admise entre deux empreintes (radians)	$\pi/4$
δ_{xy}	translation maximale admise entre deux empreintes (pixels)	256

TABLE 4.4 – Paramètres de la recherche MCC-LSH

On remarquera que les performances obtenues en reconnaissance sont très sensibles à l'utilisation ou non de l'information de préalignement des images : si cette information est utilisée, on ne recherche pas les appariements de MCC excédant une certaine valeur de rotation affine, ce qui réduit significativement à la fois le coût de calcul, et le nombre d'appariements fausse acceptance.

Dans notre optique de construction d'un algorithme de comparaison robuste à des requêtes de type latente, cette information de préalignement ne peut pas être prise en compte. Aussi, la comparaison la plus significative avec l'état de l'art doit être faite avec les réglages n'utilisant pas le préalignement.

4.2.3 Grands MCC

Les grands MCC sont des vecteurs binaires de taille 312. Leur dimension ne permet pas une utilisation optimale des kd-trees. [125] a montré que le LSH binaire fournit de bonnes performances avec ce type de descripteur.

Etant donné ces caractéristiques et les résultats de l'état de l'art, nous avons procédé à l'accélération de la recherche sur les grands MCC par LSH.

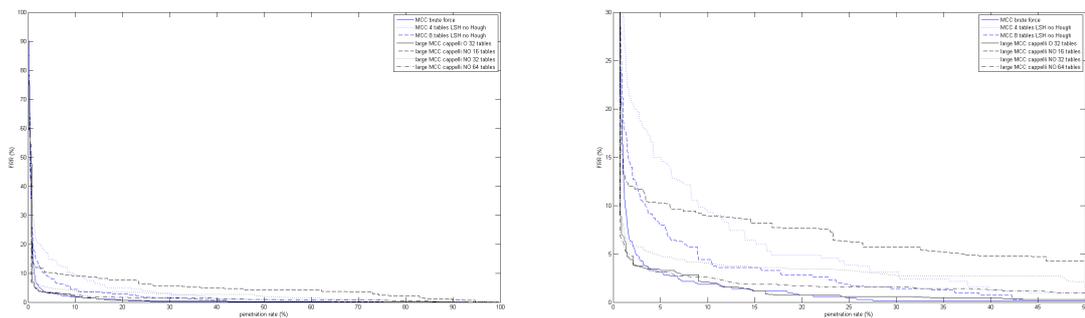
Une différence importante à prendre en compte entre notre évaluation du LSH et celle de [125], est que cette dernière utilise le LSH pour procéder à une clusterisation de l'espace des descripteurs locaux, puis procède à une recherche accélérée par dictionnaire inversé. Dans notre approche, directement dérivée de l'implémentation de la librairie FLANN, il n'y a pas de dictionnaire inversé : les distances de Hamming à la requête sont calculées explicitement pour tous les MCC de référence localisées dans le même bucket de hachage.

Paramétrage	nombre de bits par descripteur local	nombre de tables de hachage	nombre de bits sélectionnés par table	nombre de plus proches voisins sélectionnés
MCC-1	312	2	24	100
MCC-1	312	4	24	100
MCC-1	312	8	24	100

TABLE 4.5 – Paramètres de la recherche rapide par grands MCC

Méthodes	Temps de calcul par comparaison (μs)	FAR @1% FRR	FRR @20% penetration
MCC force brute	> 1000	27	1.2
MCC-LSH 2 tables, pas de Hough	55	60	3.5
MCC-LSH 4 tables, pas de Hough	90	70	3
MCC-LSH 8 tables, pas de Hough	145	50	2.5
MCC-C orienté 32 tables	28	40	2
MCC-C non orienté 16 tables	27	85	9
MCC-C non orienté 32 tables	50	85	7
MCC-C non orienté 64 tables	100	85	5

TABLE 4.6 – Performances de la recherche rapide par grands MCC



(a) vue générale

(b) zoom

FIGURE 4.8 – comparaison des performances en taux de pénétration vs taux de faux rejets pour les grands MCC, comparés à l'état de l'art

Afin d'analyser les limitations de ce type de descripteur associé à ce type de recherche accélérée, nous avons procédé à un examen des 50 couples d'images matchantes de FVC 2000 DB 2 présentant les plus faibles scores d'appariement, pour l'algorithme LSH-MCC à 8 tables.

Les origines de la faiblesse des scores observés sur cet échantillon peuvent être classifiées ainsi :

- Non recouvrement important de zones de l'empreinte riches en minuties. Les descripteurs associés à ces minuties ne trouvent pas de vis à vis dans l'image matchante.
- Présence d'une zone d'empreinte avec très faible qualité d'acquisition (typiquement, les crêtes ne peuvent pas être distinguées). Les minuties présentes dans la même zone de l'image matchante ont des descripteurs qui ne trouvent pas de vis à vis.
- Dans les empreintes présentant un nombre important de minuties, des grappes de "fausses" minuties, liées à du bruit d'acquisition ou des artefacts, ne trouvent pas non plus de vis à vis

Type d'erreur	nombre
non recouvrement	23
zone de faible qualité	9
artefacts importants	15
perte de minutie	3

TABLE 4.7 – Analyse des faux rejets de la recherche rapide par grands MCC



FIGURE 4.9 – exemple de faux rejet mauvais recouvrement



FIGURE 4.10 – exemple de faux rejet zone de faible qualité



FIGURE 4.11 – exemple de faux rejet minuties artefacts

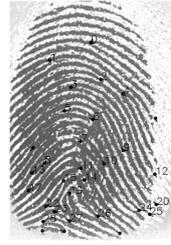


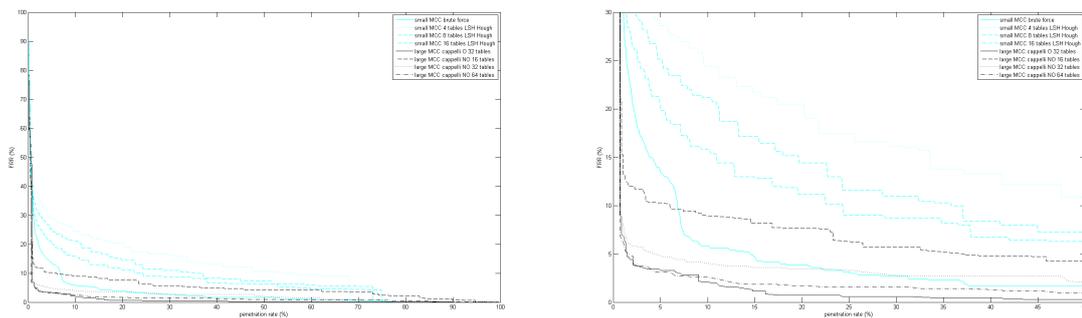
FIGURE 4.12 – exemple de faux rejet minuties bruitées

4.2.4 Petits MCC

Les petits MCC sont des vecteurs binaires de taille 54. Leur dimension est plus réduite que celle des grands MCC mais ne permet pas encore une utilisation optimale des kd-trees. Comme pour les grands MCC, nous avons procédé à l'accélération de la recherche sur les grands MCC par LSH.

Méthodes	Temps de calcul par comparaison (μs)	FAR @1% FRR	FRR @20% penetration
petits MCC force brute	> 1000	27	1.2
MCC-LSH 2 tables, pas de Hough	16	55	36
MCC-LSH 4 tables, pas de Hough	25	NA	35
MCC-LSH 8 tables, pas de Hough	44	NA	33
MCC-C orienté 32 tables	28	40	2
MCC-C non orienté 16 tables	27	85	9
MCC-C non orienté 32 tables	50	85	7
MCC-C non orienté 64 tables	100	85	5

TABLE 4.8 – Paramètres de la recherche rapide par petits MCC



(a) vue générale

(b) zoom

FIGURE 4.13 – comparaison des performances en taux de pénétration vs taux de faux rejets pour les petits MCC, comparés à l'état de l'art

Afin d'analyser les limitations de ce type de descripteur associé à ce type de recherche accélérée, nous avons procédé à un examen des 50 couples d'images matchantes de FVC 2000 DB 2 présentant les plus faibles scores d'appariement, pour l'algorithme LSH-MCC à 8 tables.

Outre les types d'erreurs déjà recensées dans le cas des grands MCC, on identifie un autre type d'erreur : les cas où les minuties de 1; empreinte sont majoritairement espacées d'une distance supérieure au rayon du petit MCC. Dans ce cas, très peu de descripteurs sont non vides, ce qui rend la comparaison très difficile.

Type d'erreur	nombre
non recouvrement	21
zone de faible qualité	7
artefacts importants	13
perte de minutie	2
minuties trop espacées	7

TABLE 4.9 – Performances de la recherche rapide par petits MCC



FIGURE 4.14 – exemple de faux rejet minuties trop espacées

4.2.5 Patches de carte d'orientation

Les patches de carte d'orientation sont des vecteurs à valeurs sur $[0, 255]$, de taille 128. La distance utilisée est la norme ℓ_2 . Leur dimension ne permet pas une utilisation optimale des kd-trees. Nous avons choisi le hierarchical k-means comme méthode de recherche accélérée. Les expériences d'apprentissage de dictionnaire parcimonieux sur l'espace des patches tendent à montrer que celui-ci a une dimensionalité intrinsèque inférieure à 128, c'est-à-dire que les patches sont répartis sur une variété de \mathbf{R}^{128} . Le HKM est plus apte qu'un LSH-L2 à tirer parti d'une telle structure.

Méthodes	Temps de calcul par comparaison (μs)	FAR @1% FRR	FRR @20% penetration
Force brute	> 1000	X	X
HKM facteur branche 16, pas de Hough	5	88	9.5
HKM facteur branche 8, pas de Hough	5	90	11
HKM facteur branche 2, pas de Hough	5	90	13.5
MCC-C orienté 32 tables	28	40	2
MCC-C non orienté 16 tables	27	85	9
MCC-C non orienté 32 tables	50	85	7
MCC-C non orienté 64 tables	100	85	5

TABLE 4.10 – Paramètres de la recherche rapide par patches de carte d'orientation

Afin d'analyser les limitations de ce type de descripteur associé à ce type de recherche accélérée, nous avons procédé à un examen des 50 couples d'images matchantes de FVC 2000 DB 2 présentant les plus faibles scores d'appariement, pour l'algorithme patch de carte d'orientation avec HKM facteur de branche 16.

Les erreurs recensées sont proches en proportion de celles également recensées pour les grands MCC. Cela peut s'expliquer par le fait que les deux descripteurs couvrent une surface d'image proche, et sont tous les deux centrés sur les minuties. Les pires faux rejets étant surtout liés à des problèmes d'appariements de minuties, les minuties problématiques pour un descripteur le sont généralement aussi pour l'autre descripteur. Néanmoins, on observe un turnover entre les 50 pires faux rejets des deux descripteurs, aussi une analyse de la corrélation effective des deux descripteurs est nécessaire, et sera présentée dans la suite des travaux.

Type d'erreur	nombre
non recouvrement	21
zone de faible qualité	8
artefacts importants	17
perte de minutie	4

TABLE 4.11 – Analyse des faux rejets de la recherche rapide par patches de carte d'orientation

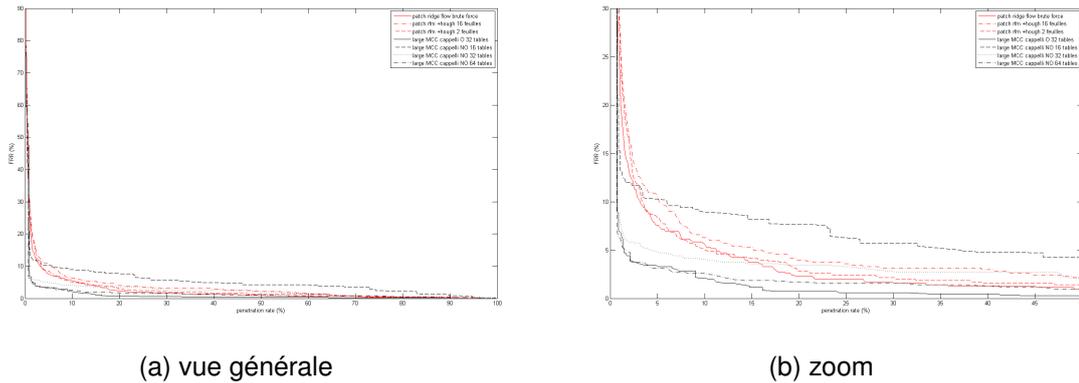


FIGURE 4.15 – comparaison des performances en taux de pénétration vs taux de faux rejets pour les patches de carte d'orientation, comparés à l'état de l'art

4.2.6 Triplets de minuties

Les triplets de minuties sont des vecteurs de dimension 9 à valeur dans $[0255]$. Cette dimension permet l'utilisation des kd-trees, qui présentent l'avantage considérable d'avoir un coût unitaire moyen de comparaison en $O(\log n)$, avec n la taille de la base de données.

Méthodes	Temps de calcul par comparaison (μs)	FAR @1% FRR	FRR @20% penetration
Force brute	> 1000	X	X
random 4 Kdtree, pas de Hough	3	90	45
random 4 Kdtree, Hough	5	90	45
MCC-C orienté 32 tables	28	40	2
MCC-C non orienté 16 tables	27	85	9
MCC-C non orienté 32 tables	50	85	7
MCC-C non orienté 64 tables	100	85	5

TABLE 4.12 – Paramètres de la recherche rapide par triplets de minuties

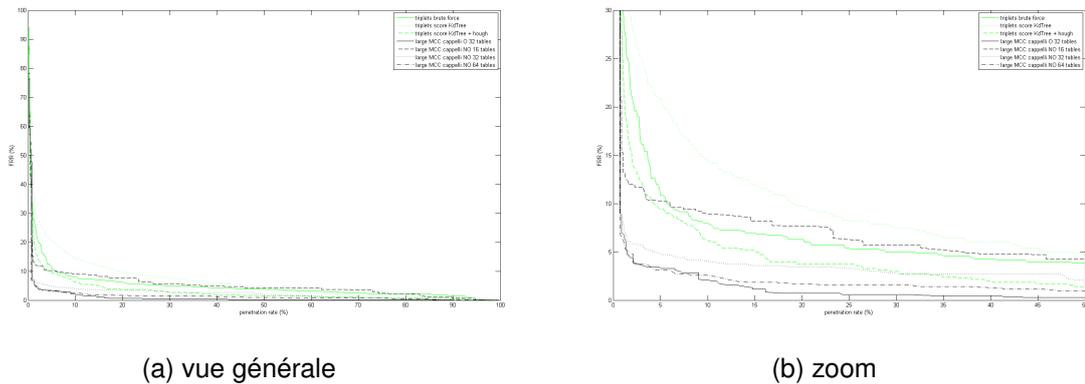


FIGURE 4.16 – comparaison des performances en taux de pénétration vs taux de faux rejets pour les triplets de minuties, comparés à l'état de l'art

Afin d'analyser les limitations de ce type de descripteur associé à ce type de recherche accélérée, nous avons procédé à un examen des 50 couples d'images matchantes de FVC 2000 DB 2 présentant les plus faibles scores d'appariement, pour l'algorithme triplets de minuties avec une forêt de 4 KdTrees aléatoires.

Les types d'erreurs recensées sont similaires à celles déjà observée pour les petits MCC. Cependant, les proportions sont significativement différentes par rapport aux précédents descripteurs analysés. On observe pour les triplets une plus forte proportion d'erreurs liées à l'absence de quelques minuties essentielles sur l'une des deux images. Cet effet s'explique par la nature du descripteur : une seule minutie est susceptible d'appartenir à un grand nombre de triplets, potentiellement significatifs, aussi l'occultation de certaines minuties peut dégrader considérablement les scores d'appariement entre images matchantes.

Type d'erreur	nombre
non recouvrement	18
zone de faible qualité	6
artefacts importants	4
perte de minutie	20
minuties trop espacées	2

TABLE 4.13 – Analyse des faux rejets de la recherche rapide par minuties

4.2.7 Comparaison générale

Les résultats sur la base de test FVC 2000 DB 2 permettent de choisir pour chaque descripteur le jeu de paramètres de recherche accélérée permettant le meilleur arbitrage entre précision de l'algorithme de comparaison et coût en temps de comparaison.

Les résultats en termes de précision de la comparaison sont résumés dans les figures 4.17, ceux en termes de coût de comparaison sont résumés dans le tableau 4.14.

Pour évaluer la robustesse des paramètres choisis sur un changement de base et sur une plus

grande échelle, nous évaluons les algorithmes de recherche accélérée que nous avons construits sur les bases d'enroulés NIST DB 4 et NIST DB 14. Les résultats sont résumés dans les figures 4.17, 4.18 et 4.19 et le tableau 4.14. Les performances en temps de calcul du processus de recherche dans la base de données ont été évaluées avec une implémentation en C, sur un processeur Intel Core i5 à 1.9 GHz.

Nous complétons la comparaison avec l'état de l'art avec les performances obtenues par [133], [145], [76], [88], [125] et [146] sur NIST DB 4, et par [147], [148], [132]. Les performances de [132] sont obtenues par fusion d'un descripteur local et d'un descripteur global.

Performances en précision

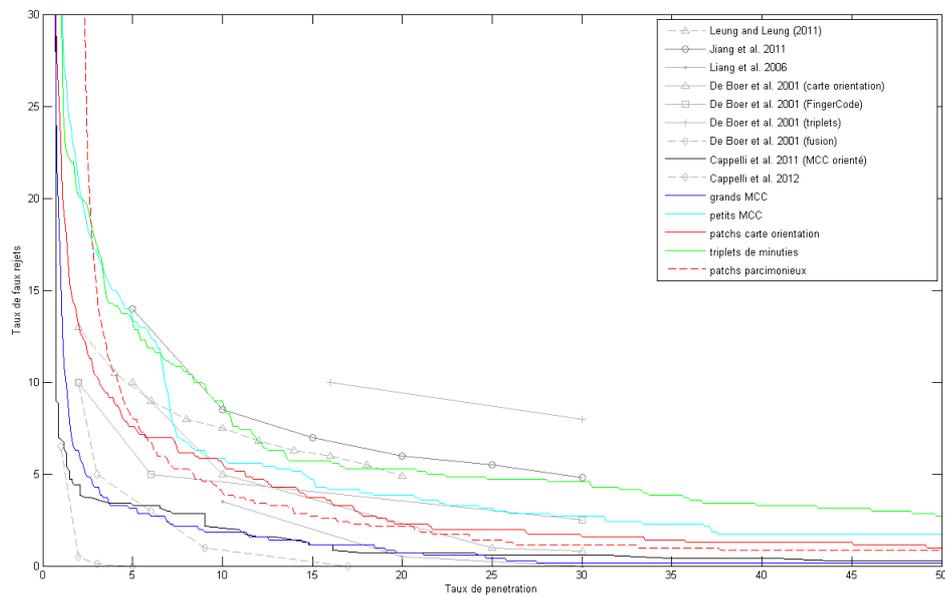


FIGURE 4.17 – performances sur FVC 2000 DB 2

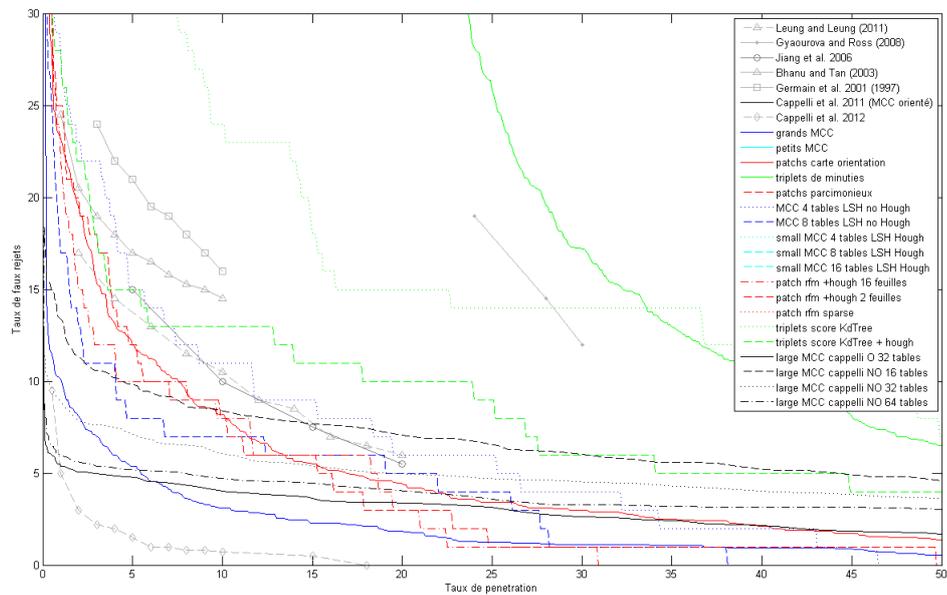


FIGURE 4.18 – performances sur NIST DB 4

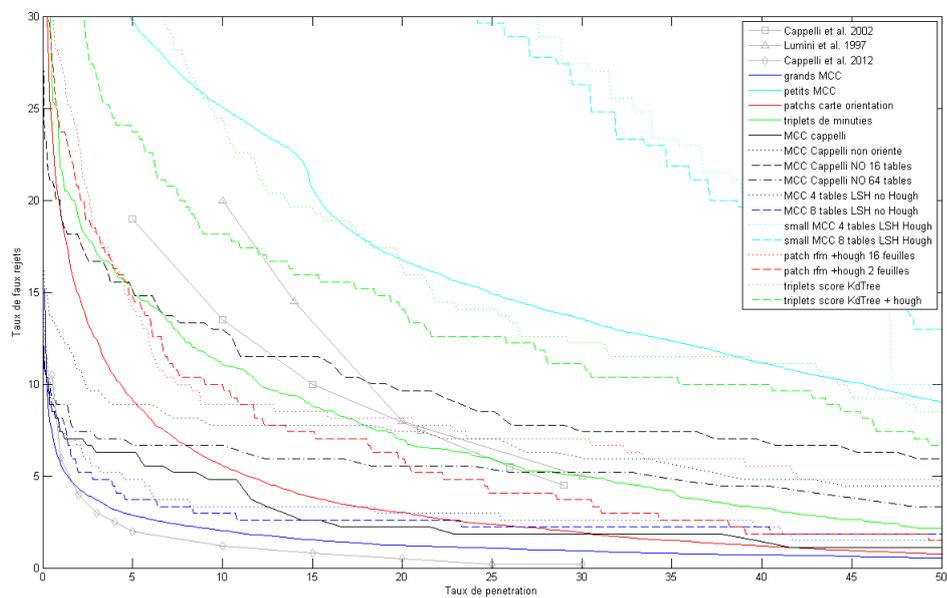


FIGURE 4.19 – performances sur NIST DB 14

Performances en temps de calcul

Méthodes	Temps de calcul par comparaison (μs)	FAR @1% FRR	FRR @20% penetration
grands MCC-LSH 2 tables, pas de Hough	55	60	3.5
grands MCC-LSH 4 tables, pas de Hough	90	70	3
grands MCC-LSH 8 tables, pas de Hough	145	50	2.5
petits MCC-LSH 2 tables, pas de Hough	16	55	36
petits MCC-LSH 4 tables, pas de Hough	25	NA	35
petits MCC-LSH 8 tables, pas de Hough	44	NA	33
HKM facteur branche 16, pas de Hough	5	88	9.5
HKM facteur branche 8, pas de Hough	5	90	11
HKM facteur branche 2, pas de Hough	5	90	13.5
random 4 Kdtree, pas de Hough	3	90	45
random 4 Kdtree, Hough	5	90	45
MCC-C orienté 32 tables	28	40	2
MCC-C non orienté 16 tables	27	85	9
MCC-C non orienté 32 tables	50	85	7
MCC-C non orienté 64 tables	100	85	5

TABLE 4.14 – Performances en temps de calcul et précision sur FVC 2000 DB 2

4.3 Bag of Words

En s'inspirant des travaux de [42], nous considérons deux implémentations possibles de recherche rapide par Bag of Words : le système de dictionnaire inversé et le Min-Hash.

L'approche par Bag of Words a été utilisée à plusieurs reprises pour l'indexation d'empreintes digitales. [110] propose ainsi de construire un score de similarité entre empreintes correspondant au nombre d'occurrences communes des indices de clusters de triplets de minuties entre une image requête et chaque image de référence, ce qui correspond en pratique à un score entre histogrammes de fréquences de mots visuels, correspondants aux clusters de triplets. Ce

type de score par dictionnaire inversé est également utilisé par [149], en combinaison avec une transformée de Hough, ou par [125] sur les MCC.

Comme déjà décrit précédemment au chapitre 2, et conformément aux coûts théoriques estimés tels que présentés dans le tableau 4.2, la recherche accélérée par Bag of Words combinée à un dictionnaire inversé présente les meilleures performances en termes de coûts de comparaison unitaires. Nous avons donc procédé à la construction d'un algorithme de comparaison d'empreintes digitales basé sur un dictionnaire inversé, en utilisant l'implémentation proposée par [42] appliquée aux descripteurs locaux que nous avons construits.

Nous avons procédé à la construction de dictionnaires de mots visuels sur les descripteurs locaux suivants : grands MCC, patches de carte d'orientation, et triplets de minuties.

Pour chacun de ces descripteurs, nous avons appris un dictionnaire en utilisant la base FVC 2000 DB 2 comme base d'apprentissage. Ensuite, le dictionnaire obtenu est utilisé pour indexer une base d'empreintes digitales quelconque, et procéder à une recherche accélérée par dictionnaire inversé. Nous avons procédé à l'évaluation des performances par dictionnaire inversé sur la base FVC 2000 DB 2 (base d'apprentissage), ainsi que sur la base NIST DB 4.

La représentation par Bag-of-Words correspondant à un histogramme de fréquence des mots du dictionnaire, c'est-à-dire à un vecteur où le i -ème coefficient représente la fréquence respective du mot i dans l'image considérée. Nous utilisons comme distance entre vecteurs Bag-of-Words la *distance cosinus* d_{cos} , définie par

$$d_{cos}(a, b) = 1 - \frac{a \cdot b}{\|a\| \|b\|}$$

Nous pondérons également les différents mots du dictionnaire par *tf-idf* (*Term Frequency - Inverse Document Frequency*), concept développé par [150]. La pondération tf-idf consiste, dans un algorithme de recherche de plus proches voisins d'un document textuel ou visuel au sein d'une base de données, à considérer qu'un mot, textuel ou visuel, est d'autant plus discriminant pour trouver des plus proches voisins qu'il est plus fréquent dans la requête, et d'autant moins discriminant qu'il est plus fréquent dans la base de données à chercher.

Si on considère une base de données d'empreintes digitales $DB = (I_1, \dots, I_N)$, avec leurs nombres respectifs de descripteurs locaux n_1, \dots, n_N , et leurs représentations par Bag-of-Words respectives $((x_{1i})_{1 \leq i \leq M}, \dots, (x_{Ni})_{1 \leq i \leq M})$ obtenues sur un dictionnaire de M mots visuels d_1, \dots, d_M , et une empreinte digitale requête I_s de m_s descripteurs locaux et de représentation par BoW $(y_i)_{1 \leq i \leq M}$, alors la distance entre les représentations par BoW de I_s et d'une empreinte de référence $I_r \in DB$ est définie par

$$d_{BoW-tfidf}(I_s, I_r) = 1 - \sum_{i=1}^M tf_{is} idf_i \cdot \frac{y_i x_{ri}}{\|y\| \|x_r\|} \quad (4.1)$$

où

$$tf_{is} = \frac{y_i}{m_s} \quad (4.2)$$

et

$$idf_i = \log\left(\frac{\sum_k n_k}{\#\{I_l/d_i \in I_l\}}\right) \quad (4.3)$$

$tf_{i,s}$ correspond à la fréquence du mot d_i dans l'image requête I_s , et $\{I_l/d_i \in I_l\}$ correspond au nombre d'empreintes de référence comportant au moins une fois le mot d_i .

4.3.1 Dictionnaire inversé sur les grands MCC

Le dictionnaire visuel adapté aux patches de carte d'orientation est construit par algorithme LSH. Ce choix est motivé par les résultats obtenus par LSH en recherche par plus proches voisins, et par le fait que le LSH avec fonction de hachage binaire apparaît bien adapté à la clusterisation de vecteurs binaires comme les MCC.

Les paramètres retenus sont résumés dans le tableau 4.15

Algorithme d'apprentissage du dictionnaire	LSH
Nombre de tables de hachage	16
Nombre de bits par fonction de hachage	24
Distance utilisée entre BoW	<i>cos</i>
Pondération des mots du dictionnaire	tf-idf

TABLE 4.15 – Paramètres du dictionnaire inversé sur les grands MCC

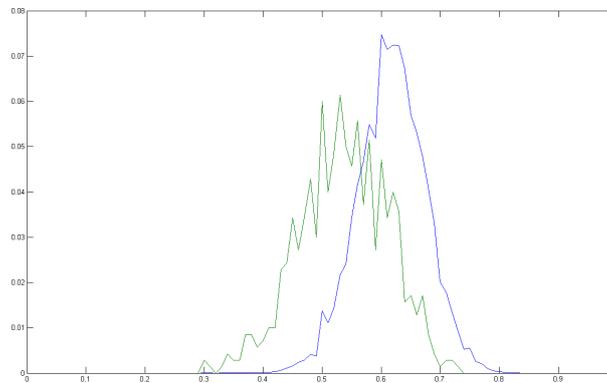
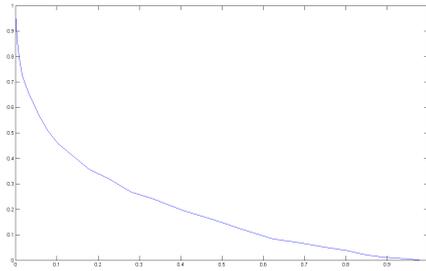
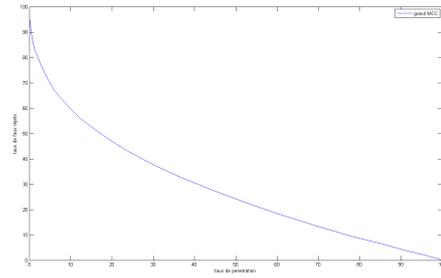


FIGURE 4.20 – distribution des distances BoW de grands MCC pour les couples d'images matchantes et non matchantes sur FVC 2000 DB 2



(a) base FVC 2000 DB 2



(b) base NIST DB 4

FIGURE 4.21 – performances des distances BoW de grands MCC en taux de pénétration vs taux de faux rejets

4.3.2 Dictionnaire inversé sur les patches de carte d'orientation

Le dictionnaire visuel adapté aux patches de carte d'orientation est construit par algorithme k-means. Ce choix est motivé par les résultats obtenus par HKM en recherche par plus proches voisins, et par le fait que le K-means apparaît bien adapté à la clusterisation de ce type de descripteurs, comme expliqué en section 4.2.5. Les paramètres retenus sont résumés dans le tableau 4.16

Nombre de mots du dictionnaire	1024
Algorithme d'apprentissage du dictionnaire	HKM
Nombre d'itérations du K-means	5
Profondeur du HKM	10
Facteur de branche	2
Distance utilisée entre BoW	<i>cos</i>
Pondération des mots du dictionnaire	tf-idf

TABLE 4.16 – Paramètres du dictionnaire inversé sur les patches de carte d'orientation

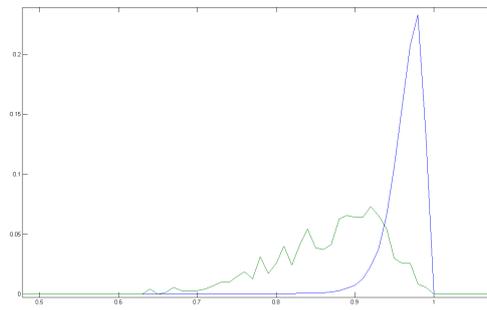
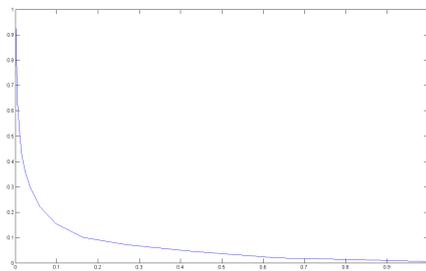
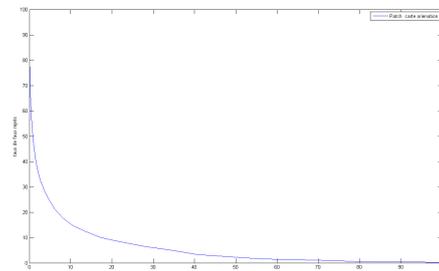


FIGURE 4.22 – distribution des distances BoW de patches pour les couples d'images matchantes et non matchantes sur FVC 2000 DB 2



(a) base FVC 2000 DB 2



(b) base NIST DB 4

FIGURE 4.23 – performances des distances BoW de patches en taux de pénétration vs taux de faux rejets

4.3.3 Dictionnaire inversé sur les triplets de minuties

Le dictionnaire visuel adapté aux triplets de minuties est construit par algorithme k-means. Ce choix est différent de celui utilisé pour la recherche accélérée. Cette différence se justifie par le fait que le calcul du dictionnaire et des histogrammes correspondant étant fait avant la phase de recherche, le critère du coût de recherche associé aux kd-trees n'est plus pertinent. Un k-means est alors plus susceptible de trouver les éventuels variétés de l'espace des triplets. De plus, les travaux de [149] ont montré qu'une telle méthode d'apprentissage de dictionnaire pouvait donner d'excellents résultats sur les triplets de minuties.

Les paramètres retenus sont résumés dans le tableau 4.17

Nombre de mots du dictionnaire	1024
Algorithme d'apprentissage du dictionnaire	HKM
Nombre d'itérations du K-means	5
Profondeur du HKM	10
Facteur de branche	2
Distance entre BoW	<i>cos</i>
Pondération des mots du dictionnaire	tf-idf

TABLE 4.17 – Paramètres du dictionnaire inversé sur les triplets de minuties

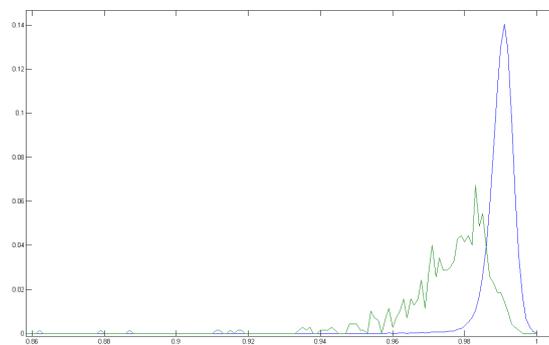
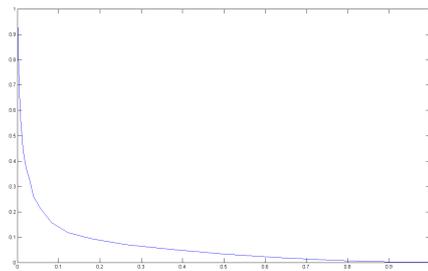
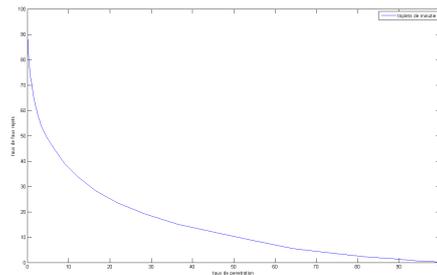


FIGURE 4.24 – distribution des distances BoW de triplets pour les couples d'images matchantes et non matchantes sur FVC 2000 DB 2 - abscisses = distances, ordonnées = densité de probabilité, en vert distances matchantes, en bleu distances non matchantes



(a) sur FVC 2000 DB 2



(b) sur NIST DB 4

FIGURE 4.25 – performances des distances BoW de triplets en taux de pénétration (abscisse) vs taux de faux rejets (ordonnées), en pourcentage

4.4 Fusion des données

La combinaison des informations issues de descripteurs différents pour optimiser les performances en reconnaissance est utilisée depuis plusieurs années dans les algorithmes de comparaison d'empreintes digitales. Parmi les travaux fondateurs sur ce sujet, on trouve ceux de [134], qui proposait la fusion des scores de similarité obtenus à partir de triplets de minuties, de FingerCode et de cartes d'orientation.

L'amélioration des performances en reconnaissance par fusion de descripteurs tient fondamentalement au fait que les informations issues de descripteurs de natures différentes sont décorréélées, sauf dans le cas où les descripteurs sont des appariements matchants.

Deux grands types de fusion de descripteurs peuvent être utilisés, éventuellement de façon complémentaire : la fusion à l'échelle locale et la fusion à l'échelle image.

La fusion à l'échelle locale suppose que les différents types de descripteurs locaux peuvent être liés entre eux (par exemple les différents types de descripteurs centrés sur une même minutie). Un descripteur local agrégé, combinaison de plusieurs descripteurs locaux, est alors construit en une zone locale de l'image (typiquement le voisinage d'une minutie), et la similarité entre descripteurs agrégés locaux est construite par combinaison des similarités entre les descripteurs locaux associés à cette zone.

La fusion à l'échelle image ne nécessite pas de construire des relations entre descripteurs locaux de natures différentes. Dans ce cas, une similarité agrégée locale est construite directement par combinaison des similarités agrégées associées au couple d'images considéré.

La méthode de fusion proposée par [134] consiste, pour chaque image requête, à combiner les rangs, après tri par score de similarité entre images, de chaque image de la base référence, pour chaque type de descripteurs. Il s'agit ici d'une fusion à l'échelle image : la combinaison proposée est la méthode Borda, une forme de vote pondéré par le rang.

D'autres approches par fusion ont été proposées plus récemment. Ainsi, [151] propose de fusionner les similarités entre images obtenues à partir de descripteurs utilisant les informations de niveau 1 et 2 des empreintes digitales (crêtes et minuties).

Les descripteurs choisis sont les suivants : cartes d'orientation centrées sur les minuties, triplets de minuties, fréquences de crête autour du core de l'empreinte, points singuliers (cores et deltas), MCC binaires. Ces descripteurs recoupent en grande partie les descripteurs locaux que nous avons retenus, à des différences d'implémentation près pour les cartes d'orientations locales et les triplets de minuties. La méthode de fusion est encore une méthode à l'échelle image, combinaison linéaire et produit de scores agrégés pour chaque type de descripteur.

[132] propose également une méthode de fusion de descripteurs, cette fois en fusionnant directement un descripteur global, de type carte d'orientation et carte de fréquence de crête centrées sur le core de l'empreinte, avec des descripteurs locaux MCC. La fusion est également faite à l'échelle image, et abouti à d'excellentes performances en reconnaissance à un coût de comparaison unitaire proche de celui du MCC-LSH original ([125]).

4.4.1 Décorrélation des descripteurs utilisés

Niveau local

Afin de vérifier que la fusion des descripteurs locaux que nous avons construits est effectivement susceptible d'améliorer les performances en précision de la reconnaissance, nous avons procédé au calcul du degré de corrélation entre les pseudo-distances locales d'appariements locaux non matchants pour chacun des descripteurs centrés sur les minuties, à savoir les grands et petits MCC, les patches de carte d'orientation et leur représentation parcimonieuse.

Cette corrélation est vérifiée sur un large échantillon d'appariements locaux non matchants de la base d'apprentissage FVC2000 DB 2. Les résultats sont présentés dans le graphique 4.26.

Ce graphique présente sous forme matricielle les corrélations entre les différents descripteurs, chaque ligne et chaque colonne correspondant à un type de descripteur. Les schémas sur la diagonale présente la distribution des distances du type de descripteur présent simultanément en abscisse et en ordonnée. Les schémas hors diagonale présente les distributions de distances selon deux types de descripteurs différents en abscisse et en ordonnée, ainsi que la droite de régression linéaire entre les deux distances. Les schémas de la matrice sont symétriques de part et d'autre de la diagonale.

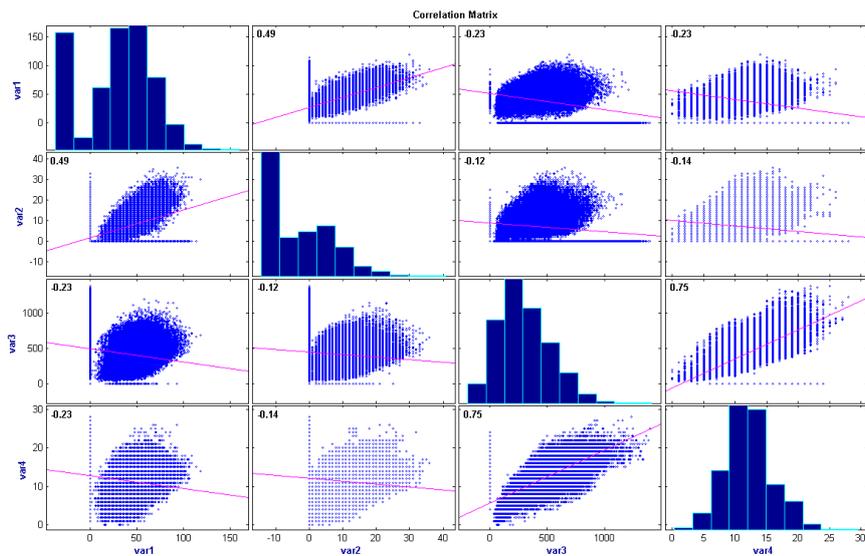


FIGURE 4.26 – corrélations linéaires entre les différents descripteurs minuties-centrés (de gauche à droite par colonne : grands MCC, petits MCC, patches de carte d'orientation, représentation parcimonieuse de patches. Même ordre de haut en bas par ligne)

Nous constatons une corrélation importante entre les deux types de MCC, corrélation qui se justifie par le fait que les deux descripteurs utilisent la même information, à un degré de quantification différent. Les patches de cartes d'orientation sont également fortement corrélés à leur représentation parcimonieuse, ce qui apparaît naturel.

D'une manière générale, la corrélation entre l'information de carte d'orientation et de carte locale de minuties (les MCC) apparaît faible.

Parmi les constats moins évidents, on observe que la corrélation est néanmoins sensiblement plus forte entre les patchs et les grands MCC qu'entre les patchs et les petits MCC. Cela peut s'expliquer par la taille des grands MCC, et le fait qu'il existe une corrélation importante entre l'orientation des minuties et l'orientation des crêtes. Sur une grande surface d'empreinte, la corrélation observable entre une partie de l'information des MCC (l'orientation des minuties) et celle des patchs (l'orientation des crêtes) sera donc accrue par rapport à une petite surface.

Niveau global

Nous avons également procédé à la vérification de la corrélation au niveau global, en comparant les scores agrégés en force brute obtenus par 3.16, avec utilisation de la transformée de Hough avec meilleurs appariements, similarité sur laquelle nous observons les meilleures performances générales en force brute.

Cette corrélation est vérifiée sur un large échantillon de scores d'appariements agrégés entre images non matchantes de la base d'apprentissage FVC2000 DB 2. Les résultats sont présentés dans le graphique 4.27. Ce graphique est conçu de la même façon que le graphique 4.26, en illustrant cette fois les scores agrégés à l'échelle image, et en incluant les triplets de minuties.

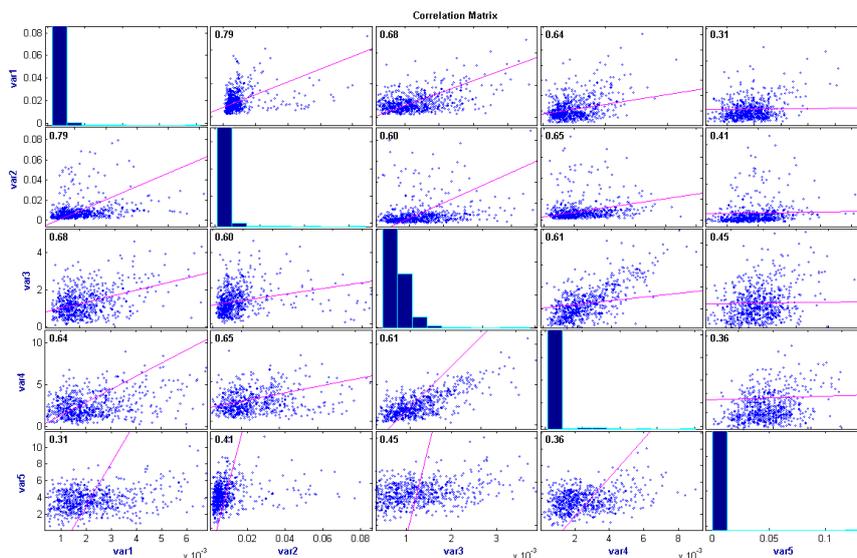


FIGURE 4.27 – corrélations linéaires entre les différents scores agrégés (de gauche à droite par colonne : grands MCC, petits MCC, patchs de carte d'orientation, représentation parcimonieuse de patchs, triplets de minutie. Même ordre de haut en bas par ligne)

On observe cette fois une corrélation significative entre l'ensemble des descripteurs minutie-

centrés. Cette corrélation peut s'expliquer par la cohérence globale ajoutée au score utilisée, cette cohérence s'appuyant sur la même distribution spatiale de minuties pour un couple d'images donné, quels que soient les descripteurs considérés.

On observe par contre que la corrélation reste faible entre les triplets de minuties et l'ensemble des autres descripteurs, ce qui là encore peut s'expliquer par le support différent de la cohérence globale, les triplets utilisant une distribution spatiale sensiblement différente de celle des simples minuties.

4.5 Fusion des LSH MCC et coefficients parcimonieux

Nous avons procédé à l'expérimentation de la fusion des descripteurs entre les grands MCC et les représentations parcimonieuses de patch de carte d'orientation. Pour cela, nous avons modifié le calcul de similarité entre deux empreintes, en nous rapprochant de l'implémentation utilisée par [146].

Pour un type de descripteur local donné, le score agrégé utilisé entre deux empreintes I_1 et I_2 , avec leurs ensembles de descripteurs locaux respectifs $\{z_{1,i}\}_{1 \leq i \leq n_1}$ et $\{z_{2,j}\}_{1 \leq j \leq n_2}$, est le suivant :

$$S_{Hough\ LSH}(I_1, I_2) = \frac{\sum_{i=1}^{n_1} \max_{j/(z_{1,i}, z_{2,j}) \in cl_B} s_{LSH}(z_{1,i}, z_{2,j})}{n_1} \quad (4.4)$$

où cl_B désigne le cluster de l'espace quantifié des similitudes le plus rempli par la transformée de Hough avec meilleurs appariements, et $s_{LSH}(z_{1,i}, z_{2,j})$ désigne le nombre de collisions sur l'ensemble des tables de LSH entre $z_{1,i}$ et $z_{2,j}$

Pour comparer une empreinte requête I_s dotées des descripteurs MCC $\{z_{MCC,s,i}\}_{1 \leq i \leq n_s}$ et patches parcimonieux $\{z_{patch,s,i}\}_{1 \leq i \leq n_s}$ à une empreinte de référence I_r dotées des descripteurs MCC $\{z_{MCC,r,i}\}_{1 \leq i \leq n_r}$ et patches parcimonieux $\{z_{patch,r,i}\}_{1 \leq i \leq n_r}$, les méthodes de fusion que nous avons implémentées sont les suivantes :

- Fusion à l'échelle image Nous calculons un score de fusion par simple sommation des deux scores agrégés calculés en parallèle pour les grands MCC et les patches parcimonieux. On calcule ainsi un score $S_{Hough\ LSH\ MCC}(I_1, I_2)$ en appliquant l'équation 4.6 aux seuls descripteurs grands MCC, puis un score $S_{Hough\ LSH\ patch}(I_1, I_2)$ en appliquant l'équation 4.6 aux seuls descripteurs patches parcimonieux. On calcule enfin un score fusionné à l'échelle image $S_{LSH\ fusion\ image} = S_{Hough\ LSH\ MCC} + S_{Hough\ LSH\ patch}$
- On construit une similarité $s_{fusion}((z_{MCC,s,i}, z_{Patch,s,i}), (z_{MCC,r,j}, z_{Patch,r,j}))$ entre minuties par somme des similarités $s_{MCC}(z_{MCC,s,i}, z_{MCC,r,j})$ et $s_{Patch}(z_{Patch,s,i}, z_{Patch,r,j})$. $\forall i$, on note $z_{combi,i}$ le couple de descripteurs locaux $(z_{MCC,i}, z_{Patch,i})$ associés à la minutie i . On définit alors un score agrégé $S_{LSH\ fusion\ local}(I_1, I_2)$ par

$$S_{LSH\ fusion\ local}(I_s, I_r) = \frac{\sum_{i=1}^{n_s} \max_{j/(z_{combi,s,i}, z_{combi,r,j}) \in cl_B} s_{fusion}(z_{combi,s,i}, z_{combi,r,j})}{n_s} \quad (4.5)$$

- On construit un nouveau descripteur z_{concat} par concaténation directe des coefficients binaires qui forment les descripteurs grands MCC et patches parcimonieux. Le score fu-

sionné $S_{concat}(I_s, I_r)$ est alors construit par application directe de l'équation 4.6 sur les descripteurs concaténés :

$$S_{LSH \text{ fusion concat}}(I_s, I_r) = \frac{\sum_{i=1}^{n_s} \max_{j/(z_{concat,s,i}), (z_{concat,r,j}) \in cl_B} S_{concat}((z_{concat,s,i}), (z_{concat,r,j}))}{n_s} \quad (4.6)$$

Afin d'équilibrer le pouvoir discriminant respectif des deux descripteurs, lors du hachage du descripteur concaténé, nous imposons un nombre prédéfini de bits choisis dans la partie MCC du descripteur, et un autre nombre de bits dans la partie patch.

Les paramètres utilisés pour les différents types de fusion sont décrits dans les tableaux 4.18 et 4.19.

Nombre de tables de hachage	32
Nombre de bits sélectionnés par table de hachage	24
Nombre minimum de bits à 1 pour indexer le MCC	3

TABLE 4.18 – Paramètres de LSH pour le descripteur patch de carte d'orientation représentation parcimonieuse

Nombre de tables de hachage	32
Nombre de bits sélectionnés par table de hachage dans le MCC	24
Nombre de bits sélectionnés par table de hachage dans le patch parcimonieux	4
Nombre minimum de bits à 1 pour indexer le MCC	2

TABLE 4.19 – Paramètres de LSH pour le descripteur concaténé

Les figures 4.28, 4.29 et 4.30 synthétisent les résultats obtenus en termes de performances sur FVC 2000 DB 2m NIST DB 4 et NIST DB 14, tirés de nos travaux de [152]

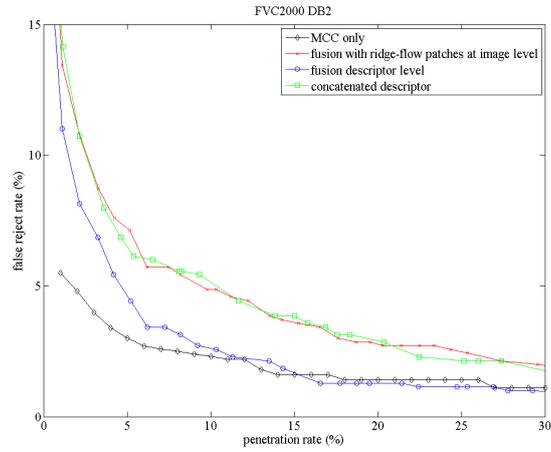


FIGURE 4.28 – Performances sur FVC2000 DB2 du MCC-LSH seul et des différents types de fusion

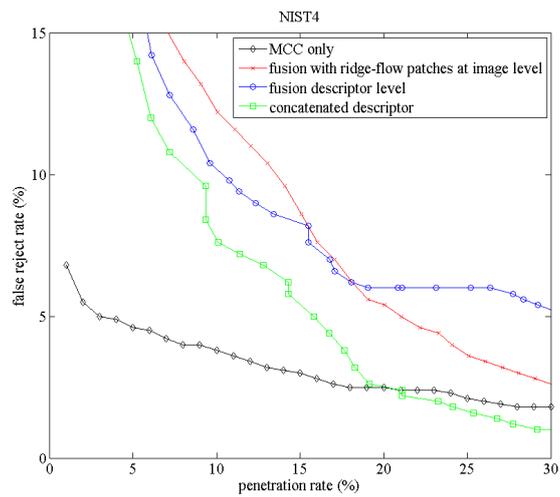


FIGURE 4.29 – Performances sur NIST DB 4 du MCC-LSH seul et des différents types de fusion

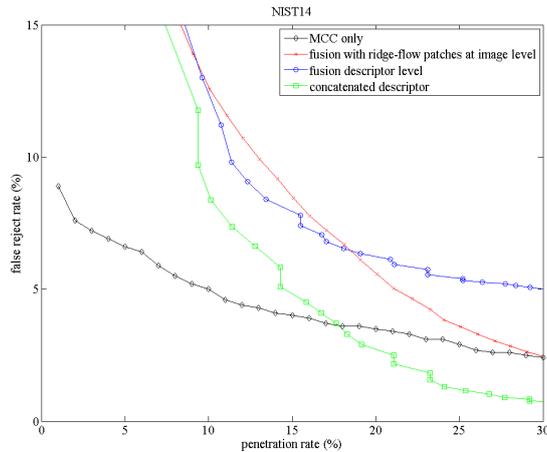


FIGURE 4.30 – Performances sur NIST DB 14 du MCC-LSH seul et des différents types de fusion

Le tableau 4.20 synthétise les performances en temps de calcul des différentes méthodes de fusion, avec une implémentation en C, sur un processeur 3.40 GHz Intel Core i5.

Méthode	Taille de la base de données 27 000 images références
temps de requête pour une empreinte digitale via index MCC-LSH	68 ms
temps de requête pour une empreinte digitale via patch parcimonieux	57 ms
temps de requête pour la fusion au niveau image	91 ms
temps de requête pour la fusion au niveau descripteur	40 ms
temps de requête pour la fusion par concaténation	21 ms

TABLE 4.20 – temps de recherche moyen sur évalué sur NIST DB 14 sur un processeur 3.40 GHz Intel Core i5 (monocoeur)

4.6 Fusion des distances BoW

Etant donnée la décorrélation observée entre les distances entre triplets de minuties et patches de carte d'orientation, nous avons procédé à la fusion des distances de Bag of Words associées à ces deux descripteurs. La distance fusion obtenue est une combinaison linéaire des deux distances correspondantes, les coefficients de la combinaison linéaire étant obtenus en procédant à une SVM entre les distances matchantes et non matchantes, telle qu'illustrée sur la figure 4.31

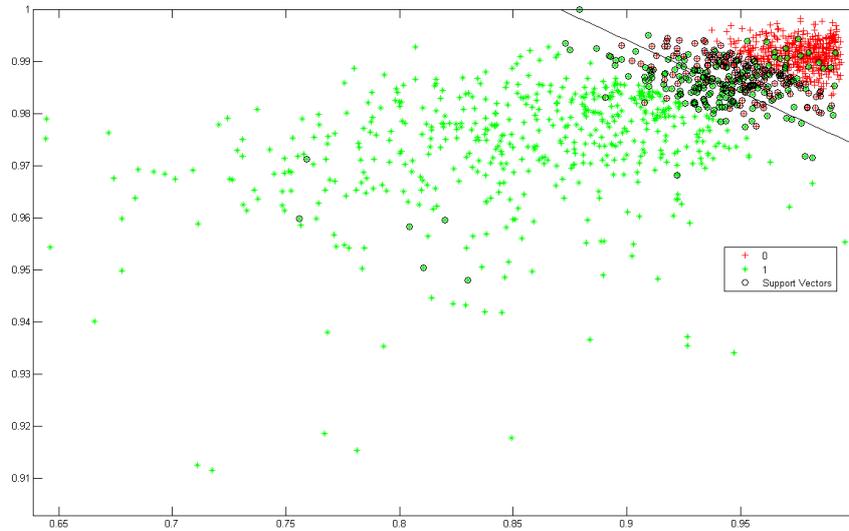


FIGURE 4.31 – visualisation de la corrélation des distances BoW patch de carte d'orientation et BoW triplets de minuties et de l'hyperplan de séparation entre les appariements matchants et non matchants par SVM - abscisse = distances BoW patches, ordonnée = distances BoW triplets - vert = couples d'images matchantes, rouge = couples d'images non matchantes

La SVM cherchant à séparer un échantillon de distances entre images matchantes et un échantillon de distances entre images non matchantes, avec mêmes effectifs entre les deux échantillons, fournit une droite de séparation d'équation $\alpha d_{BoW\ triplet} + \beta d_{BoW\ patch} + \gamma = 0$. Nous utilisons alors les coefficients α et β obtenus pour construire une distance fusion

$$d_{BoW\ fusion} = \alpha d_{BoW\ triplet} + \beta d_{BoW\ patch}$$

Le calcul de SVM nous donne $\alpha = 1.9$ et $\beta = 1.7$.

Les performances obtenues en précision sont détaillées dans la figure 4.33, les performances en temps sont détaillées dans le tableau 4.21, avec une implémentation en Matlab sur un processeur intel Core i5 1.9 GHz (monocoeur).

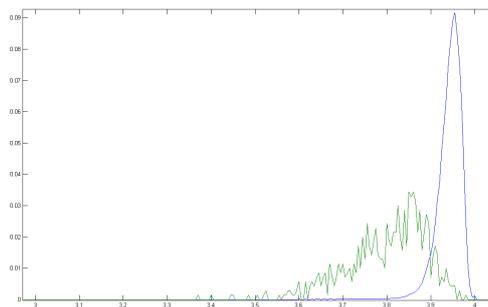
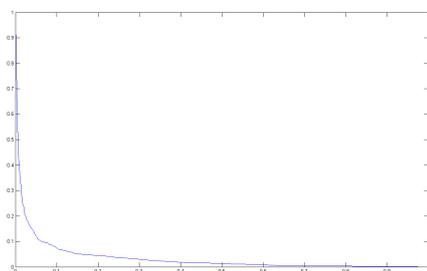
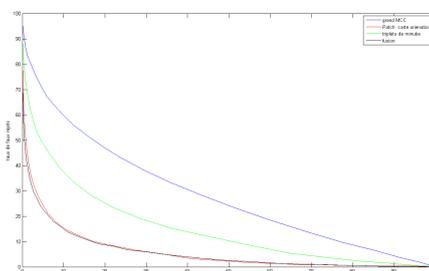


FIGURE 4.32 – distribution des distances BoW fusion pour les couples d'images matchantes (vert) et non matchantes (bleu) sur FVC 2000 DB 2



(a) sur FVC 2000 DB 2



(b) sur NIST DB 4

FIGURE 4.33 – performances des distances BoW fusion en taux de pénétration vs taux de faux rejets

Descripteur	FVC2000 DB 2	NIST DB 4
grand MCC	$0.7\mu s$	$1.1\mu s$
patch carte d'orientation	$0.4\mu s$	$6.2\mu s$
triplets de minuties	$1.5\mu s$	$3.45\mu s$

TABLE 4.21 – Performances en temps de calcul des algorithmes Bag of Words, en temps de comparaison 1 image requête vs 1 image référence moyen

4.7 Application aux requêtes latentes

[151] recense les évaluations disponibles d'algorithmes de comparaison d'empreintes digitales appliqués à des requêtes de type latente. Ces évaluations sont faites en utilisant les requêtes latentes de la base NIST DB 27, sur des bases incorporant les références matchantes de bonne qualité de NIST DB 27, plongées dans des bases d'empreintes digitales de grande taille destinées à fournir un bruit de fond.

[153] annonce ainsi un taux de faux rejets de 0.7% à un taux de pénétration de 39%, obtenu en utilisant un sous-échantillon de NIST de 10 000 images de NIST DB 14 comme base de bruit.

[144] annonce un taux de faux rejets de 19.3% à un taux de pénétration de 10%, obtenu en utilisant 240 000 images issues de NIST DB 14 et d'une base de données privée comme base de bruit.

[151] annonce des taux de faux rejets de 18.2% et 4.3% à des taux de pénétration respectifs de 10% et 39%, obtenus en utilisant 267 000 images issues de NIST DB 14 et d'une base de données privée comme base de bruit. Ces travaux annoncent également un temps de comparaison de 28 667 comparaison d'images par seconde, soit un temps moyen de comparaison d'environ $35\mu s$ par couple requête/référence, obtenues sur un Opteron 6176 @2.3 Ghz (sur un seul coeur), à comparer aux temps de calcul que nous annonçons 4.14 pour nos propres algorithmes (de l'ordre de 5 à $50\mu s$ par comparaison).

La comparaison avec des évaluations utilisant une sélection particulière d'une base publique, ou une base privée de nature inconnue, étant difficile, pour évaluer la robustesse de nos algorithmes à des requêtes de mauvaise qualité, nous avons procédé à une évaluation sur la base de *slap* FVC 2000 DB 2, en supprimant la totalité des informations issues de la moitié droite des images requêtes avant calcul des descripteurs locaux. Cette mesure est destinée essentiellement à simuler une forte occultation des requêtes.

Les descripteurs et algorithmes testés sont listés dans le tableau 4.22

Descripteur	transformée de Hough	paramètres
grand MCC	non	8 tables de hachage
patchs de carte d'orientation	oui	facteur de branche 16
triplets de minuties	oui	nombre d'arbres 4

TABLE 4.22 – Algorithmes utilisés pour la recherche sur requêtes latentes

Les résultats en courbes ROC sont présentés dans la figure 4.34

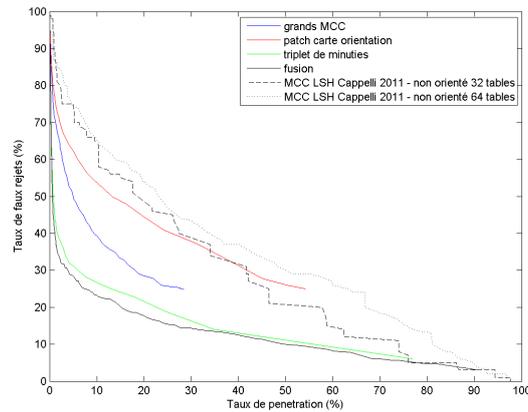


FIGURE 4.34 – Performances obtenues sur FVC 2000 DB 2 avec requêtes latentes de synthèse

Pour comparaison, nous rappelons les performances obtenues avec les requêtes non amputées, en utilisant les mêmes algorithmes, dans la figure 4.35.

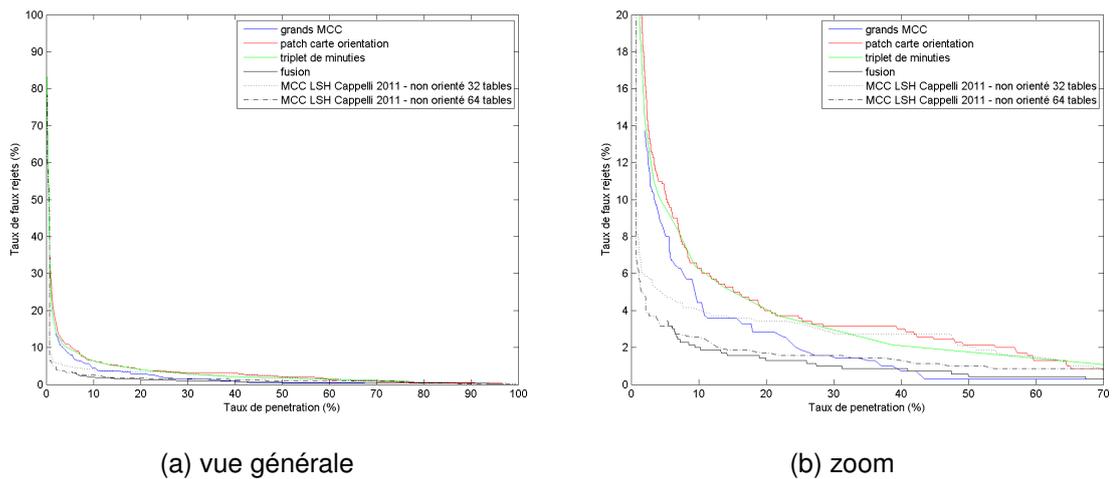


FIGURE 4.35 – performances sur FVC 2000 DB 2, requêtes complètes - la terminaison des courbes avant 100% de pénétration correspond à un plancher de faux rejets incompressibles

On constate que les performances en taux de faux rejets (24%) obtenues en fusion avec nos algorithmes à 10% de taux de pénétration sont du même ordre que celles annoncées par [151], [151], pour des temps de calcul inférieurs. Nos algorithmes sont également plus robustes à l'occultation que les MCC-LSH de [125].

4.8 Conclusion

La principale conclusion que nous pouvons tirer des méthodes d'accélération par k -nn que nous avons testées et évaluées sur les descripteurs locaux que nous avons choisies, est qu'elles permettent une accélération significative du processus de recherche, sans sacrifier pour autant les performances en précision.

Ce résultat était prévisible dès les conclusions du chapitre précédent, car nous avons constaté que d'une part les descripteurs locaux utilisés et leurs pseudo-distances associées séparent raisonnablement bien des voisinages locaux matchants de voisinages non matchants, et que d'autre part une accumulation partielle de quelques voisinages avec très forte similarité entre deux images d'empreintes suffit dans la très grande majorité des cas à les distinguer comme couple d'images matchantes.

Nous avons constaté que l'ensemble des descripteurs locaux choisis et de leurs méthodes de recherche accélérée associées donnent des résultats satisfaisants en précision, à l'exception des petits MCC, trop peu discriminants et trop peu adaptés à l'accélération choisie.

Les performances obtenues en accélération avec les triplets de minuties dépendent également du type d'empreintes utilisées : les performances en précision apparaissent meilleures avec des empreintes de type *slap* telles que présentes dans la base FVC 2000 DB 2. Une optimisation du descripteur pour des empreintes de type *rolled* serait une amélioration intéressante de nos travaux, étant donné que les triplets se prêtent à un processus de recherche en coût logarithmique en la taille de la base de données, et que plusieurs résultats de l'état de l'art en force brute suggèrent que des améliorations en termes de précision de la recherche sont possibles.

Les résultats obtenus en fusionnant les descripteurs locaux par différentes méthodes sont significatifs également. Du fait de la nature différente des informations portées par les grandes catégories de descripteurs locaux utilisés, leur fusion permet des gains en précision tout en contenant les coûts dus à l'exécution de plusieurs recherches différentes en parallèle à un niveau raisonnable.

Le BoW, s'il n'offre pas encore des performances en précision suffisantes pour une application industrielle, présente des performances en temps de calcul très intéressantes, parmi les meilleures de l'état de l'art, tout en restant une implémentation ne nécessitant pas par nature d'une information globale dense. L'enrichissement de cette représentation par de la cohérence globale via transformée de Hough pourrait notamment permettre d'améliorer la précision de la recherche à un coût en temps de calcul modique.

Conclusion

Rappel de la problématique

La problématique principale de la thèse consiste à développer un algorithme de filtrage adapté à de très grandes bases de données d’empreinte digitale, et robuste à des requêtes de type latente. Cette problématique revient à la conception d’un algorithme de type *recherche d’image basée sur le contenu* appliqué à des images d’empreintes digitales.

Les contraintes imposées par le sujet sont les suivantes :

- coût moyen en temps de calcul d’une image requête contre une image de référence très faible
- bonne précision de l’algorithme, permettant d’éviter au maximum de rater les images matchantes dans une base de données lors de la sélection d’un sous-ensemble restreint de candidats à l’image matchante. En termes techniques, cela se traduisait par un taux de faux rejets très faible, atteint à un taux de pénétration raisonnable. En ordre de grandeur, cela correspondait à 1% de faux rejets pour des taux de pénétration de l’ordre de 50% maximum
- coût de stockage raisonnable

Synthèse

La construction d’un algorithme de reconnaissance d’images par le contenu se fait typiquement en plusieurs grandes étapes :

- l’acquisition des images au format digital (*écart d’acquisition*). Cette étape ne nous concernait pas dans nos travaux : nous utilisons des données préétablies à ce niveau.
- l’extraction de données de base à partir de l’image brute (étape *encodage*. Cette phase forme la première partie de *l’écart sémantique*)
- la construction d’une représentation mathématique formelle (*signature*) à partir des données de base, permettant le calcul de similarités (ou dissimilarités) entre images. Cette phase forme la deuxième partie de *l’écart sémantique*)

La revue de l’état de l’art en CBIR a montré qu’il existait deux principaux types de signatures d’images, à savoir les représentations globales et les représentations locales.

Du point de vue du traitement d'image général, la représentation globale est théoriquement peu robuste à des requêtes bruitées, ce qui tendait à la disqualifier pour notre problématique. Nous nous sommes donc intéressés plus particulièrement aux représentations locales.

Les représentations locales présentent comme principal avantage une robustesse intrinsèque aux requêtes bruitées, du fait de l'extraction localisée d'information. Un deuxième avantage de ces représentations, est la possibilité de les enrichir à différents degrés avec de l'information globale dans la mesure de disponibilité de cette dernière.

Dans un premier temps, nous avons évalué les descripteurs choisis uniquement en termes de précision de la reconnaissance. Dans cette optique, nous avons construits des algorithmes de comparaisons sans rechercher d'optimisation en temps de calcul, quitte à obtenir des temps de comparaison moyens très importants (approche en *force brute*).

Les résultats obtenus avec des algorithmes de comparaison de type force brute ont permis d'évaluer la pertinence de l'utilisation des descripteurs choisis et de leurs distances associées en termes de précision de la reconnaissance.

Les descripteurs choisis étaient les suivants :

- le Minutiae Cylinder Code (MCC), descripteur tirés de l'état de l'art récent, que nous avons utilisés avec deux jeux de paramètres
- les patches de carte d'orientation, dont le concept existait dans l'état de l'art, et dont nous avons proposé une implémentation personnelle. Sur ce descripteur, nous avons également proposé une représentation originale, utilisant les coefficients de décomposition parcimonieux.
- les triplets de minuties, type de descripteur également répandu dans l'état de l'art, dont nous avons proposé une implémentation personnelle

Sur les différents descripteurs, on peut faire les constats suivants :

- Concernant les grands MCC : L'information locale contenue par les grands MCC est la plus riche de tous les descripteurs : c'est sur ce descripteur que l'on obtient les meilleures performances hors utilisation de la cohérence globale. Cela s'explique essentiellement par la surface d'empreinte considérée (70 pixels de rayon à dpi), pouvant recouvrir jusqu'à la moitié d'une empreinte digitale de type posé (*slap*) . Une autre raison est la quantité d'information stockée par ce descripteur (312 bits), relativement importante et condensant une grande quantité d'information concernant la localisation et l'orientation de voisinages de minuties.

Un corollaire attendu de la surface importante du MCC est sa robustesse moindre à des acquisitions de mauvaise qualité, notamment par rapport aux triplets de minuties.

Enfin, du fait de sa taille, le grand MCC incorpore déjà au sein même du descripteur une information de cohérence semi-globale, qui explique le moindre apport que constitue pour lui l'ajout de l'information globale par transformée de Hough.

- Concernant les petits MCC : Les petits MCC ont été construits afin de remédier au problème de non-robustesse aux latentes attendu sur les grands MCC. Leur rayon réduit est prévu pour assurer cette robustesse sur des surfaces d'empreintes de taille réduite. En pratique, nous avons constaté que ce rayon peut souvent s'avérer trop petit pour contenir une information pertinente sur les minuties. Typiquement, l'information des MCC n'est utile

que sur les grappes concentrées de minuties, un cas de figure assez répandu sur les empreintes digitales, mais pas présent de manière systématique, générant de ce fait des faux rejets importants à l'échelle des images.

- Concernant les patches de carte d'orientation : Les patches de carte d'orientation constitue un autre type de descripteur riche en information, bien que moins riche que les grands MCC. C'est également un descripteur couvrant une surface significative (64 pixels de rayon à 500 dpi) . Sa quantité d'information est significative également : 64 valeurs d'orientation continues. Le descripteur patch sépare moins bien les appariements matchants et non matchants que les MCC, essentiellement car certains motifs de patches sont fortement récurrents, en particulier les zones d'orientation rectiligne. Les zones à forte variabilité de l'orientation, en particulier au voisinage des points caractéristiques (cores et deltas) s'avèrent par contre fortement discriminante.
- Concernant les triplets de minuties : Les triplets de minuties sont un descripteur très répandu dans la littérature depuis deux décennies, sur lequel de nombreuses expérimentations ont été faites. Son intérêt principal est sa compacité : de 3 à 10 dimensions ou informations mono-dimensionnelles en général. Son deuxième intérêt est la compacité de la surface d'empreinte nécessaire à la construction d'un triplet complet, ce qui se traduit notamment par la plus grande robustesse de ce descripteur à des latentes. Cette compacité a un défaut intrinsèque, la nécessité de construire un très grand nombre de descripteurs, et de procéder à d'autant plus de comparaisons locales, pour atteindre des performances en précision de reconnaissance intéressantes. La génération de la totalité des triplets possibles aboutit à des quantités trop importantes pour des temps de calculs raisonnables, même avec des procédures de recherche accélérée, aussi l'utilisation d'heuristiques bien choisies pour sélectionner des sous-ensembles de triplets riches en informations discriminantes est nécessaire. Une heuristique répandue et efficace testée ces dernières années est l'utilisation des triangulation de Delaunay d'ordre n pour générer les arêtes les triplets. L'heuristique que nous avons choisie, la génération d'arêtes restreintes aux k plus proches voisins, en constitue une approximation plus rapide et plus simple à calculer, avec des performances en précision comparables aux méthodes de l'état de l'art.

Pour chacun des descripteurs ainsi sélectionnés et construits, nous avons évalué différentes méthodes d'agrégation de l'information locale en une similarité ou dissimilarité à l'échelle image. Les méthodes d'agrégation à base de graphes ont été éliminées de notre analyse du fait de coût de comparaison intrinsèquement importants. En effet, les distances entre graphes sont généralement complexes et coûteuses en temps de calcul, et se prêtent difficilement à des méthodes de recherche accélérée. Nous avons privilégié les méthodes d'agrégation susceptibles de permettre une accélération de la procédure de recherche. Pour ces méthodes, l'information globale peut être utilisée pour enrichir l'agrégation, mais la disponibilité de cette information ne doit pas être une condition nécessaire à l'agrégation.

L'ensemble des algorithmes d'agrégation que nous avons évalués peuvent être interprétés comme des moyennes des meilleurs distances ou scores d'appariements locaux. L'ajout d'information supplémentaire lors de l'agrégation est faite selon deux approches :

- la correction de la dynamique des distances ou des scores de similarités entre descripteurs

locaux. Cette approche consiste à modifier la distribution des distances locales en utilisant deux informations : en premier lieu les distributions de distances entre appariements locaux matchants et non matchants sont significativement éloignées, même si en pratique ces deux distributions ne sont séparées pour aucun des descripteurs que nous avons considérés, et en second lieu, un couple d'images matchantes est supposé contenir un nombre significatif d'appariements locaux matchants à distance faible voire très faible. Il est alors envisageable d'apprendre une distorsion de la distance locale de manière à séparer nettement les appariements locaux ayant une très forte probabilité d'être matchants de tous les autres appariements. Sous l'hypothèse que chaque couple d'images matchantes contient au moins un tel appariement, on peut ainsi accroître la similarité moyenne de ces couples, et améliorer la séparabilité des scores agrégés entre couples d'images matchantes et non matchantes.

Un tel résultat est obtenu avec la fonction de transformation des distances en scores de similarité, qui écrase à des valeurs proches de zéro les appariements au delà d'une certaine distance, et accroît à des valeurs pouvant être arbitrairement élevées les appariements à distance nulle ou très faible.

L'utilisation d'un score original, le score a contrario, est également destiné à corriger cette dynamique des distances locales. Ce score consiste à pondérer les distances entre appariements locaux selon leur probabilité plus ou moins élevée d'apparaître spontanément entre deux images non matchantes aléatoires.

- l'ajout de cohérence géométrique globale par transformée de Hough généralisée. Dans son principe, la transformée de Hough généralisée, appliquée à notre problématique d'empreintes digitales, revient à faire voter dans l'espace des rotations affines chaque appariement possible entre descripteurs locaux d'une image requête et d'une image référence.

Dans le cas où les images sont non matchantes, on attend une répartition des votes aléatoires, dans le cas où les images sont matchantes, on s'attend à observer une accumulation de votes des appariements locaux matchants dans la région de la rotation affine effective entre les deux images.

La proportion de descripteurs locaux votant pour ce meilleur cluster est un score couramment utilisé dans l'état de l'art des algorithmes de comparaison d'empreintes digitales. Un tel score se passe complètement de l'information locale, ou se restreint à la seule orientation locale de la minutie. En notant n le nombre de moyen de descripteurs locaux par image, le coût d'implémentation de ce type de score est en $O(K_{Hough}N^2)$ là où un score faisant usage des distances d'appariements locaux sera en $O(K_{local}n^2)$, K_{Hough} et K_{local} représentant des constantes de temps de calculs vérifiant $K_{local} \gg K_{Hough}$.

Cette approche présente cependant trois défauts.

D'abord, en notant n_s le nombre de descripteurs de l'image requête et n_r le nombre de descripteurs de l'image référence, comme $O(n_s \times n_r)$ appariements votent et que seuls $O(n_s)$ au maximum sont matchants, le ratio signal/bruit de l'espace de Hough est en $\frac{1}{n_r}$, ce qui rend la détection de la rotation affine matchante plus difficile quand le nombre de descripteurs locaux augmente. C'est notamment le cas pour les triplets de minuties, qui sont un ordre de magnitude plus nombreux que les minuties.

Ce ratio peut être amélioré en affinant la quantification de l'espace des rotations affines, réduisant d'autant le remplissage aléatoire des cellules, cependant cette technique a ses limites, du fait de l'étalement naturel des votes matchants autour de la rotation matchante, empêchant d'utiliser des cellules trop fines.

Ce défaut reste néanmoins minime pour des descripteurs locaux peu nombreux, par exemple les descripteurs centrés sur une minutie. Dans le cas des triplets de minuties, plus nombreux, il peut se montrer plus complexe à gérer.

Le deuxième défaut de cette approche est sa faible robustesse à une réduction significative du nombre de descripteurs locaux requêtes, cas de figure susceptible d'arriver fréquemment avec des requêtes latentes. Il devient alors très difficile de localiser la rotation affine matchante.

Le troisième défaut est la difficulté de l'accélération d'une telle technique. En effet, même si le coût de calcul unitaire d'une rotation affine entre deux descripteurs locaux est plus faible que le calcul de la distance locale, la méthode nécessite le calcul de tous les appariements locaux possibles, et le coût reste en $O(n \times m)$. Une procédure d'accélération est difficile à envisager avec les algorithmes connus de recherche accélérée.

Pour tenir compte de ces limitations, nous avons combiné l'information globale et l'information locale, en procédant à une transformée de Hough généralisée sur les appariements locaux après calcul de l'ensemble des distances locales sur tous les appariements possibles. Les distances locales ainsi calculées permettent de filtrer les appariements qui vont effectivement voter dans l'espace des rotations affines, selon trois méthodes : vote des appariements en deçà d'une certaine distance locale, vote des meilleurs appariements par descripteur local requête, vote des N meilleurs appariements par descripteur local requête.

Les expérimentations faites montrent un gain significatif en précision de reconnaissance par ajout de la transformée de Hough. Cette amélioration de performances est essentiellement obtenue en éliminant du calcul des scores agrégés un grand nombre d'appariement locaux non matchants mais présentant de fortes similarités. Cet effet, que corrige la transformée de Hough, génère beaucoup de similarités agrégées élevées entre couples d'images matchantes.

Les expérimentations faites en force brute ont permis d'évaluer le potentiel en termes de qualité de reconnaissance des différents descripteurs considérés. Ces expérimentations ont montré que des performances suffisantes pour notre problématique pouvaient être atteinte en utilisant l'information locale consolidée *a posteriori* avec de la cohérence globale obtenue par transformée de Hough.

Un autre constat important que l'on peut tirer des expériences en force brute, est que les performances obtenues avec un score agrégé de type moyenne des meilleurs appariements locaux sont optimisées en surpondérant les appariements à très faible distance, ceux qui sont une très forte probabilité d'appartenir à un couple d'images matchantes, et en sous-pondérant tous les

autres.

Ce constat nous amène à envisager d'accélérer le processus de comparaison par recherche des n plus proches voisins appliquée aux descripteurs locaux. En effet, puisque les appariements locaux qui assurent le potentiel discriminant du score agrégé sont ceux présentant la plus faible distance, il est intéressant, lors de la recherche dans une base d'images d'empreintes de grande taille, de récupérer la liste des descripteurs de référence les plus proches des descripteurs requêtes, et de restreindre le calcul explicite des scores locaux à ces seuls descripteurs. Les scores locaux aux descripteurs de références peuvent être fixés à une valeur nulle par défaut sans réduire la qualité de reconnaissance du score agrégé.

Pour procéder à la recherche rapide des n plus proches voisins, nous pouvons mettre à profit un important état de l'art dans ce domaine, développé à des fins de CBIR générique plutôt qu'appliqué explicitement à l'empreinte digitale.

Les principales méthodes de recherche rapide de plus proches voisins reposent toutes sur un principe de clusterisation des éléments de la base de données. Une clusterisation, généralement à plusieurs niveaux, est calculée sur une base d'apprentissage, formant une partition de l'espace des données. La base de données à référencer est ensuite insérée dans cette partition de l'espace. Les descripteurs requêtes se voient associer une clé de cluster dans l'index de l'espace de données

Le coût de recherche est ainsi restreint au calcul de la clé de cluster, puis aux calculs des distances aux descripteurs de référence situé dans le cluster correspondant, aboutissant à un coût global nettement réduit par rapport à une recherche des plus proches par calcul exhaustif des distances exactes à tous les descripteurs références.

Les techniques de clusterisation sont nombreuses, et nous nous sommes efforcés de choisir chaque technique et d'en adapter les paramètres pour chaque type de descripteur :

- Le Locality Sensitive Hashing repose sur la projection de l'espace des données sur un grand nombre de directions aléatoires. Les clusters correspondent à une quantification régulières de ces directions aléatoires. Le nombre de collisions entre deux points de données sur un grand nombre de directions aléatoires permet d'approcher, sous certaines conditions, la distance euclidienne effective entre ces points. En stockant en mémoire les listes de points de données associés à chaque indice de cluster, cette méthode permet donc d'échanger du temps de comparaison contre de l'espace de stockage.

Le LSH s'est avéré adapté aux MCC, qui sont des descripteurs d'empreintes digitales binaires de grande taille.

- Les indexations par k-means sont adaptables à de nombreux types de données. Elles présentent entre autres qualités la capacité de s'adapter à l'existence de variétés au sein de l'espace de données, qualité non présente dans les clusterisations par LSH.
- Les indexations par kd-trees présentent comme qualité fondamentale un coût en temps de recherche logarithmique en la taille de la base de données. La contrepartie de cet avantage, est que ses performances s'effondrent pour des descripteurs à forte dimension (typiquement > 10). Le coût de stockage de la structure d'indexation est aussi nettement plus élevé que pour d'autres méthodes.

L'application des méthodes de recherche accélérée aux différents descripteurs locaux construits aboutit à des performances en précision et en vitesse de calcul du même ordre de grandeur que l'algorithme le plus usité de l'état de l'art, utilisant le MCC-LSH. Les performances obtenues avec cet algorithme dépendent en partie de l'hypothèse d'alignement des images d'empreintes digitales. Cette hypothèse impose une majoration de la rotation affine entre empreintes digitales, qui réduit significativement le nombre d'appariements entre descripteurs MCC à considérer. En éliminant un grand nombre de fausses acceptances à considérer, et donc autant de distances locales à calculer, l'hypothèse d'alignement permet à la fois de réduire significativement le coût de calcul de l'algorithme, et de réduire le nombre de fausses acceptances à faux rejets constants, donc d'améliorer les performances en précision de reconnaissance.

Notre problématique impliquant des requêtes latentes, l'hypothèse d'alignement n'est pas recevable dans nos travaux. La comparaison de nos algorithmes avec ceux de l'état de l'art doit donc se faire hors alignement. Dans cette perspective, les performances obtenues deviennent comparables ou meilleures que celles de l'état de l'art, aussi bien en précision qu'en coût de calculs.

Toutes les techniques d'accélération de la recherche ne sont pas également intéressantes dans le cas de l'indexation de bases de données de très grande taille. Les méthodes analysées permettent de passer d'un coût de comparaison unitaire entre une image requête et une image référence en $O(n_s \text{ times } n_r / K_{ann})$ avec $\frac{n_r}{K_{ann}} \ll 1$ plutôt qu'en $O(n \times m)$, le coût de comparaison ainsi envisagé reste linéaire en la taille de la base de descripteurs locaux, en ce qui concerne le Hierarchical K-Means, les différents LSH. En effet, pour ces algorithmes, l'augmentation de la base de données aboutit nécessairement à un remplissage moyen des clusters accru. Accroître le nombre de clusters pour contrebalancer cet effet aboutira à des performances réduites en précision, qui devront être compensées par plus de backtracking dans les structures d'indexation.

La recherche accélérée par Kd-trees sur les triplets de minuties est la plus prometteuse en termes de passage à l'échelle sur des bases de données de plusieurs millions d'images. En effet, ce type d'indexation a un coût de recherche moyen logarithmique en la taille de la base de descripteurs, et la faible dimensionalité des triplets de minuties que nous avons construit permet d'utiliser le Kd-tree sur ce type de descripteur avec des performances correctes en termes d'arbitrage précision/temps de calcul.

Nous avons cependant pu constater que l'information des triplets de minuties seuls ne suffit pas à atteindre des performances en précision suffisantes pour notre problématique. Cette information nécessite d'être complétée par de l'information de cohérence globale par transformée de Hough *a posteriori*.

Ensuite, l'utilisation de Kd-trees présente comme défaut persistant la taille considérable de la structure d'indexation. Ce défaut a pu être partiellement réduit par les avancées récentes en compression de structures, mais peut rester rédhibitoire dans de nombreuses applications pratiques.

Afin de dépasser ces limitations, nous avons envisagé deux autres axes de travaux : la fusion des informations issues des différents descripteurs locaux, et la représentation par Bag-of-Words.

La fusion d'informations issues de descripteurs locaux ou globaux d'empreinte digitales a été testée à de nombreuses reprises depuis deux à trois décennies. L'intérêt de la fusion repose sur l'utilisation de descripteurs et d'informations non corrélés, impliquant que des similarités entre images d'empreinte utilisant des informations de nature différentes, auront des valeurs élevées pour des couples d'images matchantes quel que soit la nature de l'information, mais auront des valeurs fortement décorrélées pour des couples d'images non matchantes.

La fusion de descripteurs locaux décorrélés peut ainsi permettre d'améliorer la précision de la reconnaissance d'empreinte de deux manières :

- il est peu probable qu'un couple d'image non matchant présentant une forte similarité pour un descripteur (fausse acceptance), présente aussi une forte similarité pour le deuxième. La combinaison des deux similarités peut permettre d'éliminer des fausses acceptances.
- il est peu probable qu'un couple d'image matchante présentant une faible similarité pour un descripteur (faux rejet), présente aussi une faible similarité pour le deuxième, à moins d'une très faible qualité intrinsèque d'une des deux images. La combinaison des deux similarités peut permettre d'éviter des faux rejets.

S'il existe des correspondances directes entre descripteurs locaux (par exemple, tous les descripteurs centrés sur une minutie donnée), la fusion des descripteurs peut être envisagée à l'échelle locale. Sinon, l'état de l'art propose de nombreuses méthodes de fusion entre algorithmes calculant des similarités entre empreintes digitales.

La majorité de ces méthodes sont empiriques : somme, produit, combinaison linéaire, maximum ou minimum des similarités observées. Nous avons retenus comme méthode de fusion la combinaison linéaire, dont les coefficients sont calculés via une SVM entre les similarités des appariements matchants et non matchants. Les coefficients de l'hyperplan de séparation calculés par SVM fournissent les coefficients de combinaison linéaires recherchés. Cette approche de fusion par SVM, si elle a été déjà utilisée en empreinte digitales, n'a pas encore été utilisée de manière systématique pour des descripteurs locaux.

Nous avons constaté expérimentalement que les descripteurs associés respectivement aux minuties (grands et petits MCC) et à la carte d'orientation (patches et représentation parcimonieuse de patches) étaient fortement décorrélés. Cette observation confirme l'hypothèse d'une décorrélation entre la localisation spatiale des minuties, proche de l'aléatoire, et l'orientation des crêtes.

Ce constat fait, nous avons appliqué une représentation de type Bag-of-Words aux images d'empreinte digitale, d'abord pour les différents descripteurs séparément, puis en fusionnant les différents descripteurs à l'échelle image. Les performances obtenues deviennent alors proches de celles obtenues précédemment avec les méthodes de recherche accélérée sur les descripteurs séparés, avec un gain d'un à deux ordre de magnitude en coûts de calcul, dus à la représentation BoWE extrêmement compacte.

La représentation par BoW telle qu'implémentée ne tient pas compte de l'information de cohérence globale des images d'empreinte, car l'information de localisation spatiale des descripteurs locaux est perdue. Deux approches étaient envisageables pour ajouter cette information à

un coût raisonnable : la pyramide spatiale et une transformée de Hough généralisée, nécessitant donc le recalcul des rotations affines associées aux appariements locaux. La pyramide spatiale a été écartée des méthodes envisagées, à cause des dimensions particulières des empreintes digitales. La transformée de Hough généralisée a effectivement permis d'atteindre d'excellentes performances en précision, à un coût supplémentaire significatif en temps de calcul.

Enfin, nous avons procédé au test de la recherche rapide avec les descripteurs locaux grands MCC, patchs de carte d'orientation et triplets de minuties avec des latentes de synthèse, conçues en occultant une moitié des images requêtes de la base FVC2000 DB 2. Les résultats obtenus montrent des performances comparables à l'état de l'art à des taux de pénétration faibles, pour des temps de comparaison légèrement inférieur. Les résultats obtenus sont meilleurs que ceux obtenus par application directe du descripteur MCC de l'état de l'art avec recherche rapide par LSH.

Contributions

Sur l'ensemble de notre problématique, nous avons donc pu faire les apports suivants :

- Nous avons proposé un état de l'art systématique des méthodes de CBIR adaptables à des algorithmes de comparaison d'empreintes digitales robustes aux requêtes de mauvaise qualité. Un tel état de l'art n'a pas encore été proposé à notre connaissance.
- Nous avons construit, en nous inspirant des descripteurs existant dans l'état de l'art, nos propres versions des descripteurs locaux patch de carte d'orientation et triplets de minuties. En particulier, nous avons fait en sorte que les distances courantes de type euclidienne ou normes de Minkovski soient utilisables telles quelles sur ces différents descripteurs.
- Nous avons constaté que l'espace des cartes d'orientation se prête bien à une décomposition parcimonieuse sur des dictionnaires de mots visuels bien choisis. Cette propriété nous a permis de construire un nouveau type de descripteur, composé des coefficients de décomposition parcimonieuse des patchs de carte d'orientation. Ce descripteur présente des performances intéressante quand combiné à de l'information de cohérence globale, comme l'ensemble des autres descripteurs étudiés. Cette représentation présente l'avantage d'une grande compacité de l'information, utilisable avec des distances de Hamming, très peu coûteuses en temps de calcul.
- Nous avons quantifié les apports respectifs de la cohérence globale par transformée de Hough et de l'information purement locale sur un ensemble de descripteurs locaux. Ces expériences nous ont permis de constater que l'utilisation de l'information locale en plus de l'information globale permettait de réduire le nombre d'opérations théoriques nécessaires à une comparaison entre deux images.
- Nous avons expérimenté un ensemble de méthodes de recherche accélérée issue de l'état de l'art le plus récent en CBIR, appliquées à des descripteurs locaux d'empreinte digitales. Les performances obtenues sont du même ordre ou légèrement meilleures que les algorithmes de recherche rapide pour empreintes digitales de l'état de l'art robustes à des requêtes latentes.
- Nous avons identifié les problèmes et limitations intrinsèques des différents types de des-

- cripteurs locaux utilisés.
- Nous avons expérimenté des représentations de type BoW avec les descripteurs locaux que nous avons étudiés. Ces représentations présentent les meilleures performances en temps de recherche observées pour des descripteurs locaux d’empreinte digitale à notre connaissance. Les temps de comparaison sont du même ordre que les descripteurs globaux les plus performants de l’état de l’art.
 - Nous avons implémenté une fusion des descripteurs locaux en représentation BoW, permettant d’atteindre des performances intéressantes en précision, à un coût limité en temps de calcul.

Perspectives

La principale conclusion de nos travaux est que l’utilisation de l’information issue de descripteurs locaux d’empreintes digitales permet l’utilisation systématique des méthodes d’indexation et de recherche rapide développées pour des descripteurs images génériques en CBIR, tout en atteignant des performances intéressantes en précision de reconnaissance.

Les perspectives liées à ces résultats sont nombreuses :

- Une implémentation optimisée des Kd-trees ([154]) a été développée, permettant de réduire l’espace de stockage nécessaire à la structure d’indexation. Cette implémentation peut être applicable aux kd-trees de triplets de minuties.
- Des variantes des descripteurs proposés peuvent être construits et analysés, notamment une plus grande variété de surface couverte. La surface des descripteurs est en effet un critère important de leur précision, en sachant qu’une surface importante réduit la robustesse aux latentes. Un large choix de surface permettrait d’affiner l’arbitrage entre robustesse au bruit et richesse de l’information contenue dans le descripteur. En particulier, des MCC d’une taille intermédiaire à ceux que nous avons évalués devraient permettre un meilleur arbitrage entre robustesse aux occultations et précision de la recherche
- L’utilisation de plusieurs surfaces et tailles de descripteurs devrait permettre d’envisager des approches de type *coarse-to-fine*, par exemple la pyramide spatiale : quand l’information disponible est suffisamment importante, une mini-cascade de descripteurs, de semi-globaux à très locaux, pourrait permettre d’accélérer encore les temps de calcul
- Enfin, l’utilisation des descripteurs locaux conjuguée à la transformée de Hough généralisée permet de calculer la rotation affine optimale entre deux images. Quand cette rotation affine est suffisamment fiable, il serait alors intéressant de consolider les scores agrégés à partir de descripteurs locaux, avec des descripteurs globaux nécessitant le calcul explicite préalable de cette rotation affine pour être utilisable

Annexes

Minutiae Cylinder Code

La structure locale associée à une minutie donnée $m = (x_m, y_m, \theta_m)$ est représentée par un cylindre de rayon R et de hauteur 2π dont la base est centré sur la minutie de coordonnées (x_m, y_m) . Le cylindre est inscrit dans un cube dont la base est alignée sur la direction θ de la minutie. Le cuboïde est discrétisé en $N_C = N_S \times N_S \times N_D$ cellules. Chaque cellule est un petit cube de base $\Delta_S \times \Delta_S$ et de hauteur Δ_D , où $\Delta_S = \frac{2R}{N_S}$ et $\Delta_D = \frac{2\pi}{N_D}$. Chaque cellule peut être identifiée de manière unique par trois indices (i, j, k) qui correspondent à sa position dans le cube inscrit dans le cylindre, avec $i, j \in I_S = \{n \in \mathbb{N}, 1 \leq n \leq N_S\}$ et $k \in I_D = \{n \in \mathbb{N}, 1 \leq n \leq N_D\}$.

Soit

$$d\phi_k = -\pi + \left(k - \frac{1}{2}\right) \cdot \Delta_D \quad (4.7)$$

l'angle associé à chaque cellule à hauteur k dans le cylindre et soit

$$p_{i,j}^m = \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \Delta_s \cdot \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \cdot \begin{bmatrix} i - \frac{N_s+1}{2} \\ j - \frac{N_s+1}{2} \end{bmatrix} \quad (4.8)$$

le point bidimensionnel correspondant au centre des cellules d'indices i, j projetées sur la base du cylindre, exprimées en coordonnées spatiale du template de minuties. Ces points étant projetés sur la base, l'indice k n'est pas nécessaire

Pour chaque cellule (i, j, k) , une valeur numérique $C_m(i, j, k)$ est calculée en accumulant les contributions de chaque minutie située dans le voisinage $N_{p_{ij}^m}$ de p_{ij}

$$N_{p_{ij}^m} = \{m_t \in T, m_t \neq m, d_S(m_i, p_{ij}^m) \leq 3\sigma\} \quad (4.9)$$

où 3σ est le rayon du voisinage et $D_S(m, p)$ est la distance euclidienne entre la minutie m et le point p

La fonction $C_m : I_S \times I_S \times I_D \leftarrow V$ est définie par

$$C_m(i, j, k) = \begin{cases} \psi \left(\sum_{m_t \in N_{p_{ij}^m}} \left(C_m^S(m_t, p_{ij}^m) \right) \cdot C_m^D(m_t, d\phi_k) \right) & \text{si } \xi_m(p_{ij}^m) \\ \text{invalide} & \text{sinon} \end{cases} \quad (4.10)$$

où

- $V = [0, 1] \cup \{\text{invalide}\}$ est le codomaine de la fonction
- $C_m^S(m_t p_{ij}^m)$ et $C_m^D(m_t, d\phi_k)$ désignent respectivement les contributions spatiales et directionnelles de la minutie m_t

$$\xi_m(p_{ij}^m) = \begin{cases} ccvalide & \text{si } d_S(m, p_{ij}^m) \leq R \text{ et } p_{ij}^m \in Conv_{Hull}(T, \Omega) \\ invalide & \text{sinon} \end{cases} \quad (4.11)$$

où $Conv_{Hull}(T, \Omega)$ est l'enveloppe convexe des minuties de T , élargie d'un offset de Ω pixels

- $\Psi(v) = Z(v, \mu_\psi, \tau_\psi)$ est une fonction sigmoïde, contrôlée par deux paramètres μ_ψ, τ_ψ qui limitent la contribution de clusters denses de minuties et ramènent la valeur finale entre 0 et 1. La fonction sigmoïde est définie par

$$Z(v, \mu, \tau) = \frac{1}{1 + e^{-\tau(v-\mu)}} \quad (4.12)$$

Heuristiquement, $C_m(i, j, k)$ pour une cellule valide, représente la probabilité de trouver une minutie proche de p_{ij}^m avec une différence d'orientation à l'orientation de m proche de $d\phi_k$. $C_m^S(m_t, p_{ij}^m)$ est la contribution spatiale que la minutie m_t donne à la cellule (i, j, k) , définie comme suit :

$$C_m^S(m_t p_{ij}^m) = G_S(d_s(m_t, p_{ij})) \quad (4.13)$$

où

$$G_S(t) = \frac{1}{\sigma_S \sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma_S^2}\right) \quad (4.14)$$

$C_m^D(m_t, d\phi_k)$ est la contribution directionnelle de m_t , définie comme suit :

$$C_m^D(m_t, d\phi_k) = G_D(d\phi(d\phi_k, d_\theta(m, m_t))) \quad (4.15)$$

où $d\phi(\theta_1, \theta_2)$ est la différence entre les angles θ_1 et θ_2

$$d\phi(\theta_1, \theta_2) = \begin{cases} cc\theta_1 - \theta_2 & \text{si } -\pi \leq \theta_1 - \theta_2 < \pi \\ 2\pi + \theta_1 - \theta_2 & \text{si } \theta_1 - \theta_2 < -\pi \\ -2\pi + \theta_1 - \theta_2 & \text{si } \theta_1 - \theta_2 \geq \pi \end{cases} \quad (4.16)$$

et $d_\theta(m_1, m_2)$ est la différence directionnelle entre deux minuties

$$d_\theta(m_1, m_2) = d\phi(\theta_{m_1}, \theta_{m_2}) \quad (4.17)$$

$G_D(\alpha)$ est l'intégrale de la fonction gaussienne sur l'intervalle $[\alpha - \frac{\Delta_D}{2}, \alpha + \frac{\Delta_D}{2}]$

$$G_D(\alpha) = \frac{1}{\sigma_D \sqrt{2\pi}} \int_{\alpha - \frac{\Delta_D}{2}}^{\alpha + \frac{\Delta_D}{2}} e^{-\frac{t^2}{2\sigma_D^2}} dt \quad (4.18)$$

Pour comparer deux MCC C_a et C_b , on calcule d'abord une fonction de validité d'un MCC. Un MCC C_m est considéré comme *invalide* si l'une des conditions est remplie :

- il y a strictement moins de min_{vc} cellules valides dans le MCC
- il y a strictement moins de min_M minutes contribuant au remplissage du MCC

Dans tous les autres cas, un MCC est considéré comme *valide*.

On calcule également une fonction de validité croisée des cellules : en notant \mathbf{t} le triplet d'indices (i, j, k) définissant la position d'une cellule du MCC, deux éléments $\mathbf{c}_a[t]$ et $\mathbf{c}_b[t]$ sont considérés comme *comparables* si et seulement si $\mathbf{c}_a[t] \wedge \neq invalide \mathbf{c}_b[t] \neq invalide$. On définit alors deux vecteurs $\mathbf{c}_{a|b}$ et $\mathbf{c}_{b|a}$ par

$$\mathbf{c}_{a|b}[t] = \begin{cases} \mathbf{c}_a[t] & \text{si } \mathbf{c}_a[t] \text{ et } \mathbf{c}_b[t] \text{ sont comparables} \\ 0 & \text{sinon} \end{cases} \quad (4.19)$$

et

$$\mathbf{c}_{b|a}[t] = \begin{cases} \mathbf{c}_b[t] & \text{si } \mathbf{c}_a[t] \text{ et } \mathbf{c}_b[t] \text{ sont comparables} \\ 0 & \text{sinon} \end{cases} \quad (4.20)$$

Deux cylindres C_a, C_b sont définis comme comparables si et seulement si les conditions suivantes sont remplies :

- la différence d'orientation entre les deux minutes est inférieure ou égale à un seuil prédéfini δ_{theta} ($d_\theta(a, b) \leq \delta_\theta$)
 - au moins min_{ME} cellules correspondantes entre les deux cylindres sont comparables
 - $\|\mathbf{c}_{a|b} + \mathbf{c}_{b|a}\| \neq 0$ (l'intersection des cellules comparables est non nulle)
- [79] définit alors la similarité $\gamma(a, b)$ entre les deux cylindres C_a, C_b par

$$\gamma(a, b) = \begin{cases} 1 - \frac{\|\mathbf{c}_{a|b} - \mathbf{c}_{b|a}\|}{\|\mathbf{c}_{a|b} + \mathbf{c}_{b|a}\|} & \text{si } C_a \text{ et } C_b \text{ sont comparables} \\ 0 & \text{sinon} \end{cases} \quad (4.21)$$

La similarité ainsi définie est un score continu à valeur dans $[0, 1]$. Pour optimiser le temps de comparaison, [79] propose également une implémentation d'une représentation binaire des des cylindres. La fonction Ψ définie en 4.10 est corrigée en une fonction de seuillage Ψ_{bit} :

$$\Psi_{bit}(v) = \begin{cases} 1 & \text{si } v \leq \mu_\Psi \\ 0 & \text{sinon} \end{cases} \quad (4.22)$$

L'information d'un cylindre C_a peut alors être représentée par deux vecteurs, codant respectivement la valeur et la validité de chaque cellule :

$$\mathbf{c}_a[t] = \begin{cases} 1 & \text{si } C_a[t] = 1 \\ 0 & \text{sinon} \end{cases} \quad (4.23)$$

$$\hat{\mathbf{c}}_a[t] = \begin{cases} 1 & \text{si } C_a[t] \neq invalide \\ 0 & \text{sinon} \end{cases} \quad (4.24)$$

Les vecteurs $\mathbf{c}_{a|b}$ et $\mathbf{c}_{b|a}$ peuvent alors être définis ainsi :

$$\mathbf{c}_{a|b} = \mathbf{c}_a \wedge (\mathbf{c}_{\hat{a}} \wedge \mathbf{c}_{\hat{b}}) \quad (4.25)$$

$$\mathbf{c}_{b|a} = \mathbf{c}_b \wedge (\mathbf{c}_{\hat{a}} \wedge \mathbf{c}_{\hat{b}}) \quad (4.26)$$

La similarité corrigée $\gamma_{bit}(a, b)$ entre les deux cylindres C_a, C_b est alors définie ainsi :

$$\gamma(a, b) = \begin{cases} 1 - \frac{\|\mathbf{c}_{a|b} \text{ XOR } \mathbf{c}_{b|a}\|}{\|\mathbf{c}_{a|b} + \mathbf{c}_{b|a}\|} & \text{si } C_a \text{ et } C_b \text{ sont comparables} \\ 0 & \text{sinon} \end{cases} \quad (4.27)$$

[125] propose une implémentation plus simple encore de la similarité entre deux cylindres avec représentation binarisée. En notant d_H la distance de Hamming entre deux vecteurs binaires, le score γ_H est défini par

$$\gamma_H = \left(1 - \frac{d_H(\mathbf{c}_a, \mathbf{c}_b)}{n}\right)^p \quad (4.28)$$

où p est un paramètre fixé au préalable déterminant la forme de la fonction de similarité. Une valeur de p élevée fait converger le score γ_H plus rapidement vers 0 quand la distance de Hamming d_H entre les cylindres augmentent.

On remarque que l'information de validité croisée des cylindres n'est pas utilisée dans ce score, ce qui n'empêche néanmoins pas [125] de présenter de très bonnes performances en reconnaissance d'empreintes digitales.

Kd-trees

Rappels sur le formalisme :

Structure	<i>kdNode</i>
valeur de l'enregistrement indexé par le noeud	<i>kdNode.Data</i>
indice du noeud dans la base de données	<i>kdNode.Index</i>
1 si le noeud est une feuille, 0 sinon	<i>kdNode.nodeType</i>
dimension choisie pour le discriminant	<i>kdNode.Discr</i>
valeur du noeud séparant fils gauche et droit	<i>kdNode.Value</i>
fils gauche	<i>kdNode.LeftSon</i>
fils droit	<i>kdNode.RightSon</i>

En utilisant ce formalisme, un kd-tree peut être défini par son noeud racine de type *kdNode*, avec l'ensemble de ses noeuds fils.

On décrit l'algorithme d'insertion d'un point dans un kd tree 9.

On décrit l'algorithme d'insertion d'un point dans un kd tree 10.

Algorithm 9 Insertion d'un nouvel enregistrement dans un kd-tree

Soit kd un kd-tree
Soit $Root$ la racine du kd-tree considéré
Soit P un nouvel enregistrement de type $kdNode$
On définit le noeud $CurrentNode$ par $CurrentNode \leftarrow Root$
 $insert(P, kd)$
if $CurrentNode == NULL$ **then**
 $CurrentNode \leftarrow P$
 break
end if
if $CurrentNode.Data == P.data$ **then**
 break
else
 if $CurrentNode.Value > P.data[CurrentNode.Disc]$ **then**
 $CurrentNode = CurrentNode.LeftSon$
 else
 $CurrentNode = CurrentNode.RightSon$
 end if
end if

Algorithm 10 Construction statique d'un kd-tree

Soit S un ensemble d'enregistrements de type $kdNode.Data$
Soit kd le kd-tree à construire
 $build(S, kd, discr)$
if $S == \emptyset$ **then**
 $kd \leftarrow \emptyset$
 return kd
end if
if $card(S) == 1$ **then**
 $kd \leftarrow newCurrentNode(S, discr)$
 return kd
else
 $med = median(S, discr)$ (enregistrement de valeur médiane de S sur la $discr$ -ème dimension)
 $leftSons = P \in S/P \leq med$
 $rightSons = P \in S/P > med$
 $kd \leftarrow newCurrentNode(med, discr)$
 $kd.LeftSon = build(leftSons, med, next(discr))$
 $kd.RightSon = build(rightSons, med, next(discr))$
 return kd
end if

L'algorithme d'insertion d'enregistrements dans l'arbre peut également être interprété comme un algorithme de recherche de plus proche voisin.

Une description exhaustive de l'algorithme de recherche exact d'un kd-tree, ainsi que de l'algorithme de recherche par *priority search*, est donnée par [137].

On décrit l'algorithme de recherche standard d'un point dans un kd tree 11, 12, avec les paramètres 4.23.

N	nombre d'enregistrements
d	dimensions de l'espace de données
q	requête
$offsets[d]$	liste des offsets
nn_{dist}	meilleure distance au carré courante

TABLE 4.23 – notations utilisées

Algorithm 11 Algorithme de recherche standard, avec calcul de distance incrémental

```

kdStandard( $q, root$ )
 $nn_{dist} = HUGE$ 
for  $i = 0; i < d; i++$  do
     $offset[i] = 0$ 
end for
 $rkd(root, 0.0)$ 
return  $nn_{dist}$ 

```

Algorithm 12 Procédure de recherche

```
rkd(currentNode, dist)
if currentNode.nodeType == 1 then
    nndist = min(nndist,  $\|q - \text{currentNode.data}\|_2^2$ )
else
    cutDim  $\leftarrow$  currentNode.discr
    prevOffset = offset[currentNode.discr]
    nextOffset = q[currentNode.discr] - currentNode.value
    if nextOffset < 0 then
        rkd(currentNode.leftSon, dist)
        dist + = -prevOffset2 + nextOffset2
        if dist < nndist then
            off[cutDim] = nextOffset
            rkd(currentNode.rightSon, dist)
            off[cutDim] = prevOffset
        end if
    else
        if nextOffset < 0 then
            rkd(currentNode.rightSon, dist)
            dist + = -prevOffset2 + nextOffset2
            if dist < nndist then
                off[cutDim] = nextOffset
                rkd(currentNode.leftSon, dist)
                off[cutDim] = prevOffset
            end if
        end if
    end if
end if
end if
```

Bibliographie

- [1] Francis Galton, “Finger prints,” *MacMillan and Co*, 1892.
- [2] E.R. Henry, “Classification and uses of finger prints,” *George Routledge and sons*, 1900.
- [3] Sharath Pankanti, Salil Prabhakar, and Anil K Jain, “On the individuality of fingerprints,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 8, pp. 1010–1025, 2002.
- [4] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar, *Handbook of fingerprint recognition*, Springer, 2009.
- [5] Jonathan D Stosz and Lisa A Alyea, “Automated system for fingerprint authentication using pores and ridge structure,” in *SPIE’s 1994 International Symposium on Optics, Imaging, and Instrumentation*. International Society for Optics and Photonics, 1994, pp. 210–223.
- [6] ANSI, “Fingerprint identification - data format for information interchange,” *ANSI/NBS-ICTS*, 1986.
- [7] Joseph H Wegstein, *An automated fingerprint identification system*, US Department of Commerce, National Bureau of Standards, 1982.
- [8] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang, “Image retrieval : Ideas, influences, and trends of the new age,” *ACM Computing Surveys (CSUR)*, vol. 40, no. 2, pp. 5, 2008.
- [9] SR Surya and G Sasikala, “Survey on content based image retrieval,” *Indian Journal of Computer Science and Engineering (IJCSE)*, 2011.
- [10] I Felci Rajam and S Valli, “A survey on content based image retrieval,” *Life Science Journal*, vol. 10, no. 2, pp. 2475–2487, 2013.
- [11] Gholamreza Rafiee, Satnam Singh Dlay, and Wai Lok Woo, “A review of content-based image retrieval,” in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*. IEEE, 2010, pp. 775–779.
- [12] Aude Oliva and Antonio Torralba, “Building the gist of a scene : The role of global image features in recognition,” *Progress in brain research*, vol. 155, pp. 23–36, 2006.
- [13] Vaibhavkumar J Mistry and Mahesh M Goyani, “A literature survey on facial expression recognition using global features,” *Int. J. Eng. Adv. Technol*, vol. 2, pp. 653–657, 2013.
- [14] David G Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157.

- [15] David G Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [16] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid, “Local features and kernels for classification of texture and object categories : A comprehensive study,” *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.
- [17] Tinne Tuytelaars and Krystian Mikolajczyk, “Local invariant feature detectors : a survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.
- [18] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, “Orb : an efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.
- [19] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst, “Freak : Fast retina keypoint,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee, 2012, pp. 510–517.
- [20] James Ze Wang, Jia Li, Robert M Gray, and Gio Wiederhold, “Unsupervised multiresolution segmentation for images with low depth of field,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 1, pp. 85–90, 2001.
- [21] CL Mallows, “A note on asymptotic joint normality,” *The Annals of Mathematical Statistics*, pp. 508–515, 1972.
- [22] ByoungChul Ko and Hyeran Byun, “Integrated region-based image retrieval using region’s spatial relationships,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. IEEE, 2002, vol. 1, pp. 196–199.
- [23] Pablo Musé, Frédéric Sur, Frédéric Cao, Yann Gousseau, and Jean-Michel Morel, “An a contrario decision method for shape element recognition,” *International Journal of Computer Vision*, vol. 69, no. 3, pp. 295–315, 2006.
- [24] Richard O Duda and Peter E Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [25] Dana H Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [26] Euripides G. M. Petrakis and Christos Faloutsos, “Similarity searching in medical image databases,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 9, no. 3, pp. 435–447, 1997.
- [27] Stefano Berretti, Alberto Del Bimbo, and Enrico Vicario, “Efficient matching and indexing of graph models in content-based retrieval,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 10, pp. 1089–1105, 2001.
- [28] Andrés Almansa and Laurent Cohen, “Fingerprint image matching by minimization of a thin-plate energy using a two-step algorithm with auxiliary variables,” in *Applications of Computer Vision, 2000, Fifth IEEE Workshop on*. IEEE, 2000, pp. 35–40.
- [29] Arun Ross, Sarat Dass, and Anil Jain, “A deformable model for fingerprint matching,” *Pattern Recognition*, vol. 38, no. 1, pp. 95–103, 2005.

- [30] Arun Ross, Sarat C Dass, and Anil K Jain, “Fingerprint warping using ridge curve correspondences,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 19–30, 2006.
- [31] Y-L Boureau, Nicolas Le Roux, Francis Bach, Jean Ponce, and Yann LeCun, “Ask the locals : multi-way local pooling for image recognition,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2651–2658.
- [32] Josef Sivic and Andrew Zisserman, “Video google : A text retrieval approach to object matching in videos,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.
- [33] Bruno A Olshausen and David J Field, “Sparse coding with an overcomplete basis set : A strategy employed by v1 ?,” *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [34] Edoardo Amaldi and Viggo Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,” *Theoretical Computer Science*, vol. 209, no. 1, pp. 237–260, 1998.
- [35] Stéphane G Mallat and Zhifeng Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [36] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, “Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2169–2178.
- [37] Kristen Grauman and Trevor Darrell, “The pyramid match kernel : Discriminative classification with sets of image features,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. IEEE, 2005, vol. 2, pp. 1458–1465.
- [38] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas, “The earth mover’s distance as a metric for image retrieval,” *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [39] Marius Muja and David Lowe, “Scalable nearest neighbour algorithms for high dimensional data,” 2014.
- [40] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel, “An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.
- [41] Jon Louis Bentley, “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [42] Mohamed Aly, Mario Munich, and Pietro Perona, “Indexing in large scale image collections : Scaling properties and benchmark,” in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 418–425.
- [43] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al., “Similarity search in high dimensions via hashing,” in *VLDB*, 1999, vol. 99, pp. 518–529.
- [44] RM McCabe and EM Newton, “Data format for the interchange of fingerprint, facial, and other biometric information,” *ANSI/NIST*, 2007.

- [45] R Michael McCabe, "Fingerprint interoperability standards," in *Automatic Fingerprint Recognition Systems*, pp. 433–451. Springer, 2004.
- [46] RM Stock and CW Swonger, "Development and evaluation of a reader of fingerprint minutiae," *Cornell Aeronautical Laboratory, Technical Report CAL no. XM-2478-X-1*, pp. 13–17, 1969.
- [47] Bijan Moayer and King-Sun Fu, "A tree system approach for fingerprint pattern recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 3, pp. 376–387, 1986.
- [48] MR Verma, Arun K Majumdar, and B Chatterjee, "Edge detection in fingerprints," *Pattern Recognition*, vol. 20, no. 5, pp. 513–523, 1987.
- [49] Nalini K Ratha, Shaoyun Chen, and Anil K Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, no. 11, pp. 1657–1672, 1995.
- [50] Carlotta Domeniconi, Sibel Tari, and Ping Liang, "Direct gray scale ridge reconstruction in fingerprint images," in *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*. Citeseer, 1998, vol. 5, pp. V–2941.
- [51] Marius Tico and Pauli Kuosmanen, "A topographic method for fingerprint segmentation," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. IEEE, 1999, vol. 1, pp. 36–40.
- [52] Dario Maio and Davide Maltoni, "Direct gray-scale minutiae detection in fingerprints," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 1, pp. 27–40, 1997.
- [53] Xudong Jiang, Wei Yun Yau, and Wee Ser, "Minutiae extraction by adaptive tracing the gray level ridge of the fingerprint image," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. IEEE, 1999, vol. 2, pp. 852–856.
- [54] Xudong Jiang, Wei-Yun Yau, and Wee Ser, "Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge," *Pattern Recognition*, vol. 34, no. 5, pp. 999–1013, 2001.
- [55] Jinxiang Liu, Zhongyang Huang, and Kap Luk Chan, "Direct minutiae extraction from gray-level fingerprint image by relationship examination," in *Image Processing, 2000. Proceedings. 2000 International Conference on*. IEEE, 2000, vol. 2, pp. 427–430.
- [56] Jeng-Horng Chang and Kuo-Chin Fan, "Fingerprint ridge allocation in direct gray-scale domain," *Pattern Recognition*, vol. 34, no. 10, pp. 1907–1925, 2001.
- [57] Nicolau Canyellas, Enrique Cantó, Giuseppe Forte, and Mariano López, "Hardware-software co-design of a fingerprint identification algorithm," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2005, pp. 683–692.
- [58] Antonio Grasselli, "On the automatic classification of fingerprints," *Methodologies of Pattern Recognition*, pp. 253–273, 1969.
- [59] Asker M Bazen and Sabih H Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 905–919, 2002.
- [60] Rafael C Gonzalez and Richard E Woods, "Digital image processing," 2007.

- [61] Michael Kass and Andrew Witkin, "Analyzing oriented patterns," *Computer vision, graphics, and image processing*, vol. 37, no. 3, pp. 362–385, 1987.
- [62] MA Oliveira and Neucimar Jerônimo Leite, "A multiscale directional operator and morphological tools for reconnecting broken ridges in fingerprint images," *Pattern Recognition*, vol. 41, no. 1, pp. 367–377, 2008.
- [63] BG Sherlock, "Computer enhancement and modeling of fingerprint images," in *Automatic Fingerprint Recognition Systems*, pp. 87–112. Springer, 2004.
- [64] Takahiro Nakamura, Miwako Hirooka, Hideto Fujiwara, and Kazuhiko Sumi, "Fingerprint image enhancement using a parallel ridge filter," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 1, pp. 536–539.
- [65] Toshio Kamei, "Image filter design for fingerprint enhancement," in *Automatic Fingerprint Recognition Systems*, pp. 113–126. Springer, 2004.
- [66] Sharat Chikkerur, Alexander N Cartwright, and Venu Govindaraju, "Fingerprint enhancement using stft analysis," *Pattern Recognition*, vol. 40, no. 1, pp. 198–211, 2007.
- [67] Masahiro Kawagoe and Akio Tojo, "Fingerprint pattern classification," *Pattern Recognition*, vol. 17, no. 3, pp. 295–303, 1984.
- [68] Louis Coetzee and Elizabeth C Botha, "Fingerprint recognition in low quality images," *Pattern Recognition*, vol. 26, no. 10, pp. 1441–1460, 1993.
- [69] Hiroshi NAKAJIMA, Koji KOBAYASHI, and Tatsuo HIGUCHI, "A fingerprint matching algorithm using phase-only correlation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 87, no. 3, pp. 682–691, 2004.
- [70] Koichi Ito, Ayumi Morita, Takafumi Aoki, Hiroshi Nakajima, Koji Kobayashi, and Tatsuo Higuchi, "A fingerprint recognition algorithm combining phase-based image matching and feature-based matching," in *Advances in Biometrics*, pp. 316–325. Springer, 2005.
- [71] Vivek A Sujan and Michael P Mulqueen, "Fingerprint identification using space invariant transforms," *Pattern Recognition Letters*, vol. 23, no. 5, pp. 609–619, 2002.
- [72] Zhengyu Ouyang, Jianjiang Feng, Fei Su, and Anni Cai, "Fingerprint matching with rotation-descriptor texture features," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. IEEE, 2006, vol. 4, pp. 417–420.
- [73] Koichi Ito, Ayumi Morita, Takafumi Aoki, Tatsuo Higuchi, Hiroshi Nakajima, and Koji Kobayashi, "A fingerprint recognition algorithm using phase-based image matching for low-quality fingerprints," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. IEEE, 2005, vol. 2, pp. II–33.
- [74] Xin Shuai, Chao Zhang, and Pengwei Hao, "The optimal ros-based symmetric phase-only filter for fingerprint verification," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. IEEE, 2007, vol. 2, pp. II–381.
- [75] Anil K Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti, "Filterbank-based fingerprint matching," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 846–859, 2000.

- [76] Xudong Jiang, Manhua Liu, and Alex C Kot, "Fingerprint retrieval for identification," *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 4, pp. 532–542, 2006.
- [77] Zin Mar Win and Myint Myint Sein, "Robust fingerprint recognition system using orientation and texture features," 2012.
- [78] Asker M Bazen and Sabih H Gerez, "An intrinsic coordinate system for fingerprint matching," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 198–204.
- [79] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni, "Minutia cylinder-code : A new representation and matching technique for fingerprint recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2128–2141, 2010.
- [80] Xudong Jiang and Wei-Yun Yau, "Fingerprint minutiae matching based on the local and global structures," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. IEEE, 2000, vol. 2, pp. 1038–1041.
- [81] Asker M Bazen and Sabih H Gerez, "Fingerprint matching by thin-plate spline modelling of elastic deformations," *Pattern Recognition*, vol. 36, no. 8, pp. 1859–1867, 2003.
- [82] Sharat Chikkerur, Alexander N Cartwright, and Venu Govindaraju, "K-plet and coupled bfs : a graph based fingerprint representation and matching algorithm," in *Advances in Biometrics*, pp. 309–315. Springer, 2005.
- [83] Xinjian Chen, Jie Tian, and Xin Yang, "A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure," *Image Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 767–776, 2006.
- [84] Akhil Vij and Anoop Namboodiri, "Fingerprint indexing based on local arrangements of minutiae neighborhoods," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 71–76.
- [85] Huimin Deng and Qiang Huo, "Minutiae matching based fingerprint verification using delaunay triangulation and aligned-edge-guided triangle matching," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2005, pp. 270–278.
- [86] Nalini K Ratha, Ruud M Bolle, Vinayaka D Pandit, and Vaibhav Vaish, "Robust fingerprint authentication using local structural similarity," in *Applications of Computer Vision, 2000, Fifth IEEE Workshop on*. IEEE, 2000, pp. 29–34.
- [87] Julien Bringer and Vincent Despiegel, "Binary feature vector fingerprint representation from minutiae vicinities," in *Biometrics : Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, 2010, pp. 1–6.
- [88] Robert S Germain, Andrea Califano, and Scott Colville, "Fingerprint matching using transformation parameter clustering," *Computing in Science and Engineering*, vol. 4, no. 4, pp. 42–49, 1997.
- [89] George Bebis, Taisa Deaconu, and Michael Georgiopoulos, "Fingerprint identification using delaunay triangulation," in *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*. IEEE, 1999, pp. 452–459.

- [90] Bir Bhanu and Xuejun Tan, “Fingerprint indexing based on novel features of minutiae triplets,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 616–622, 2003.
- [91] Tsai-Yang Jea and Venu Govindaraju, “A minutia-based partial fingerprint recognition system,” *Pattern Recognition*, vol. 38, no. 10, pp. 1672–1684, 2005.
- [92] Xuefeng Liang, Tetsuo Asano, and Arijit Bishnu, “Distorted fingerprint indexing using minutia detail and delaunay triangle,” in *Voronoi Diagrams in Science and Engineering, 2006. ISVD’06. 3rd International Symposium on*. IEEE, 2006, pp. 217–223.
- [93] Jia Jia, Lianhong Cai, Pinyan Lu, and Xuhui Liu, “Fingerprint matching based on weighting method and the svm,” *Neurocomputing*, vol. 70, no. 4, pp. 849–858, 2007.
- [94] Hongjun Wang, Jianhuai Qi, Weifan Zheng, and Mingwen Wang, “Balance k-means algorithm,” in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. IEEE, 2009, pp. 1–3.
- [95] Miguel Angel Medina-Pérez, Milton García-Borroto, Andres Eduardo Gutierrez-Rodríguez, and Leopoldo Altamirano-Robles, “Improving fingerprint verification using minutiae triplets,” *Sensors*, vol. 12, no. 3, pp. 3418–3437, 2012.
- [96] Xuefeng Liang, Arijit Bishnu, and Tetsuo Asano, “A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles,” *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 721–733, 2007.
- [97] M Uysal and S Gorgunoglu, “Ridge pattern representation for fingerprint indexing,” *Elektronika ir Elektrotechnika*, vol. 20, no. 7, pp. 65–68, 2014.
- [98] Ogechukwu Iloanusì, Aglika Gyaourova, and Arun Ross, “Indexing fingerprints using minutiae quadruplets,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE, 2011, pp. 127–133.
- [99] Asker M Bazen, Gerben TB Verwaaijen, Sabih H Gerez, Leo PJ Veelenturf, and Berend Jan van der Zwaag, “A correlation-based fingerprint verification system,” 2000.
- [100] Zsolt Miklos Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1266–1276, 2000.
- [101] Marius Tico and Pauli Kuosmanen, “Fingerprint matching using an orientation-based minutia descriptor,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 8, pp. 1009–1014, 2003.
- [102] Geok See Ng, Xifeng Tong, Xianglong Tang, and Daming Shi, “Adjacent orientation vector based fingerprint minutiae matching system,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 1, pp. 528–531.
- [103] Jin Qi and Yangsheng Wang, “A robust fingerprint matching method,” *Pattern Recognition*, vol. 38, no. 10, pp. 1665–1671, 2005.
- [104] Xuchu Wang, Jianwei Li, and Yanmin Niu, “Fingerprint matching using orientationcodes and polylines,” *Pattern Recognition*, vol. 40, no. 11, pp. 3164–3177, 2007.

- [105] Farid Benhammadi, MN Amirouche, Hamid Hentous, Kadda Bey Beghdad, and Mohamed Aisani, "Fingerprint matching from minutiae texture maps," *Pattern Recognition*, vol. 40, no. 1, pp. 189–197, 2007.
- [106] Xin Shuai, Chao Zhang, and Pengwei Hao, "Fingerprint indexing based on composite set of reduced sift features," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.
- [107] Ru Zhou, Dexing Zhong, and Jiuqiang Han, "Fingerprint identification using sift-based minutia descriptors and improved all descriptor-pair matching," *Sensors*, vol. 13, no. 3, pp. 3142–3156, 2013.
- [108] Anil Jain, Lin Hong, and Ruud Bolle, "On-line fingerprint verification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 4, pp. 302–314, 1997.
- [109] Xiaoguang He, Jie Tian, Liang Li, Yuliang He, and Xin Yang, "Modeling and analysis of local comprehensive minutia relation for fingerprint matching," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 5, pp. 1204–1211, 2007.
- [110] Arun Ross and Rajiv Mukherjee, "Augmenting ridge curves with minutiae triplets for fingerprint indexing," in *Defense and Security Symposium*. International Society for Optics and Photonics, 2007, pp. 65390C–65390C.
- [111] Xiaohui Xie, Fei Su, and Anni Cai, "Ridge-based fingerprint recognition," in *Advances in Biometrics*, pp. 273–279. Springer, 2005.
- [112] Cristiano Carvalho and Hani Yehia, "Fingerprint alignment using line segments," in *Biometric Authentication*, pp. 380–386. Springer, 2004.
- [113] Alejandro Chau Chau and Carlos Pon Soto, "Hybrid algorithm for fingerprint matching using delaunay triangulation and local binary patterns," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 692–700. Springer, 2011.
- [114] Julien Bringer, Vincent Despiegel, and Mélanie Favre, "Adding localization information in a fingerprint binary feature vector representation," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2011, pp. 80291O–80291O.
- [115] Sharat Chikkerur and Nalini Ratha, "Impact of singular point detection on fingerprint matching performance," in *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*. IEEE, 2005, pp. 207–212.
- [116] Ying Jie, Zhang Renjie, Song Qifa, et al., "Fingerprint minutiae matching algorithm for real time system," *Pattern Recognition*, vol. 39, no. 1, pp. 143–146, 2006.
- [117] Almudena Lindoso, Luis Entrena, Judith Liu-Jimenez, and Enrique San Millan, "Correlation-based fingerprint matching with orientation field alignment," in *Advances in Biometrics*, pp. 713–721. Springer, 2007.
- [118] George Stockman, Steven Kopstein, and Sanford Benett, "Matching images to models for registration and object detection via clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 3, pp. 229–241, 1982.

- [119] Nalini K Ratha, Kalle Karu, Shaoyun Chen, and Anil K Jain, “A real-time matching system for large fingerprint databases,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 799–813, 1996.
- [120] Yuliang He, Jie Tian, Qun Ren, and Xin Yang, “Maximum-likelihood deformation analysis of different-sized fingerprints,” in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 421–428.
- [121] Chaoqiang Liu, Tao Xia, and Hui Li, “A hierarchical hough transform for fingerprint matching,” in *Biometric Authentication*, pp. 373–379. Springer, 2004.
- [122] Shih-Hsu Chang, Fang-Hsuan Cheng, Wen-Hsing Hsu, and Guo-Zua Wu, “Fast algorithm for point pattern matching : invariant to translations, rotations and scale changes,” *Pattern recognition*, vol. 30, no. 2, pp. 311–320, 1997.
- [123] Xifeng Tong, Jianhua Huang, Xianglong Tang, and Daming Shi, “Fingerprint minutiae matching using the adjacent feature vector,” *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1337–1345, 2005.
- [124] DK Isenor and Safwat G Zaky, “Fingerprint identification using graph matching,” *Pattern Recognition*, vol. 19, no. 2, pp. 113–122, 1986.
- [125] Raffaele Cappelli, “Fast and accurate fingerprint indexing based on ridge orientation and frequency,” *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 41, no. 6, pp. 1511–1521, 2011.
- [126] Joachim Gudmundsson, Mikael Hammar, and Marc van Kreveld, “Higher order delaunay triangulations,” *Computational Geometry*, vol. 23, no. 1, pp. 85–98, 2002.
- [127] Michal Aharon, Michael Elad, and Alfred M Bruckstein, “K-svd and its non-negative variant for dictionary design,” in *Optics & Photonics 2005*. International Society for Optics and Photonics, 2005, pp. 591411–591411.
- [128] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 689–696.
- [129] Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach, “Learning hierarchical and topographic dictionaries with structured sparsity,” in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2011, pp. 81381P–81381P.
- [130] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., “Least angle regression,” *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [131] Julien Mairal, “Sparse modeling software,” <http://spams-devel.gforge.inria.fr>.
- [132] Raffaele Cappelli and Matteo Ferrara, “A fingerprint retrieval system based on level-1 and level-2 features,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10465–10478, 2012.
- [133] KC Leung and CH Leung, “Improvement of fingerprint retrieval by a statistical classifier,” *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 1, pp. 59–69, 2011.

- [134] J. de Boer, A. M. Bazen, and S. H. Gerez, "Indexing fingerprint databases based on multiple features," *Proceedings of the 12th ProRISC Workshop (Program for Research on Integrated Systems and Circuits)*, pp. 300–306, November 2001.
- [135] David Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 2, pp. 2161–2168.
- [136] Justin Zobel and Alistair Moffat, "Inverted files for text search engines," *ACM computing surveys (CSUR)*, vol. 38, no. 2, pp. 6, 2006.
- [137] Sunil Arya and David M Mount, "Algorithms for fast vector quantization," in *Data Compression Conference, 1993. DCC'93*. IEEE, 1993, pp. 381–390.
- [138] Chanop Silpa-Anan and Richard Hartley, "Optimised kd-trees for fast image descriptor matching," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [139] Marius Muja and David G Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," in *VISAPP (1)*, 2009, pp. 331–340.
- [140] Marius Muja and David Lowe, "Fast library for approximate nearest neighbors," <http://www.cs.ubc.ca/research/flann/>.
- [141] Partha Bhowmick and Bhargab B Bhattacharya, "Approximate fingerprint matching using kd-tree," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 1, pp. 544–547.
- [142] Vandana Dixit, Deepti Singh, Parul Raj, M Swathi, and P Gupta, "kd-tree based fingerprint identification system," in *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*. IEEE, 2008, pp. 5–10.
- [143] Om Prakash Singh, Somnath Dey, and Debasis Samanta, "Fingerprint indexing using minutiae-based invariable set of multidimensional features," *International Journal of Biometrics*, vol. 6, no. 3, pp. 272–303, 2014.
- [144] Baoxi Yuan, Fei Su, and Anni Cai, "Fingerprint retrieval approach based on novel minutiae triplet features," in *Biometrics : Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*. IEEE, 2012, pp. 170–175.
- [145] Aglika Gyaourova and Arun Ross, "A novel coding scheme for indexing fingerprint patterns," in *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 755–764. Springer, 2008.
- [146] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni, "Fingerprint indexing based on minutia cylinder-code," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 1051–1057, 2011.
- [147] Raffaele Cappelli, Dario Maio, and Davide Maltoni, "A multi-classifier approach to fingerprint classification," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 136–144, 2002.
- [148] Alessandra Lumini, Dario Maio, and Davide Maltoni, "Continuous versus exclusive classification for fingerprint retrieval," *Pattern Recognition Letters*, vol. 18, no. 10, pp. 1027–1034, 1997.

- [149] Alfredo Muñoz-Briseño, Andrés Gago-Alonso, and José Hernández-Palancar, “Fingerprint indexing with bad quality areas,” *Expert Systems with Applications*, vol. 40, no. 5, pp. 1839–1846, 2013.
- [150] Karen Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.
- [151] Alessandra A Paulino, Eryun Liu, Kai Cao, and Anil K Jain, “Latent fingerprint indexing : Fusion of level 1 and level 2 features,” in *Biometrics : Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 1–8.
- [152] Antoine Deblonde, “Fingerprint indexing through sparse decomposition of ridge flow patches,” in *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2014 IEEE Symposium on*. IEEE, 2014, pp. 202–208.
- [153] Jianjiang Feng and Anil K Jain, “Filtering large fingerprint database for latent matching,” in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.
- [154] Mohamed Aly, Mario Munich, and Pietro Perona, “Compactkdt : Compact signatures for accurate large scale object recognition,” in *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*. IEEE, 2012, pp. 505–512.

Algorithmes rapides et fiables de recherche dans une base d'images d'empreintes digitales

Antoine DEBLONDE

RESUME : Dans cette thèse, nous proposons la construction d'algorithmes de comparaison d'empreintes digitales susceptibles de pouvoir parcourir rapidement de très grandes bases de données tout en conservant une forte précision dans la qualité de la reconnaissance. Nous avons comme principale contrainte à l'exécution des algorithmes la robustesse à des requêtes de mauvaise qualité, bruitées et occultées. Cette thèse s'articule en trois grandes étapes. Dans un premier temps, nous avons procédé à une analyse de l'état de l'art des méthodes de recherche d'images dans des bases de données génériques, et nous avons sélectionné parmi ces méthodes celles susceptibles de répondre favorablement à nos contraintes. Dans un second temps, nous nous sommes appuyé sur l'état de l'art des représentations d'empreintes digitales pour développer des représentations adaptées aux méthodes génériques que nous avons sélectionnées, afin de construire des algorithmes de comparaison assurant une bonne précision de la reconnaissance. Dans un troisième temps, nous avons procédé à l'accélération des algorithmes de comparaison ainsi construit, en s'efforçant d'optimiser l'arbitrage entre vitesse et précision du processus de recherche.

MOTS-CLEFS : empreinte digitale, biométrie, recherche d'image par le contenu, recherche de plus proches voisins, parcimonie

ABSTRACT : In this thesis, we propose to build fingerprint comparison algorithms that allow to browse rapidly large database, while keeping a good accuracy in recognition. The major constraint is the robustness of the algorithm to poor quality requests, such as occulted or noisy images. This thesis goes in three main steps. As a first step, we proceed to the analysis of state-of-the-art content-based image retrieval methods applied to generic image databases, and we select among these methods those which best fit our constraints. As a second step, we made use of the state of the art in fingerprint representation to develop representations that fit the generic methods we chose, in order to build comparison algorithms that ensure a good accuracy level. As a third step, we proceed to the acceleration of these algorithms through optimization of the arbitration between speed and accuracy.

KEY-WORDS : fingerprint, biometrics, content-based image retrieval, nearest neighbor search, sparsity

