

Contribution à la reconnaissance de formes discrètes par passage au continu : applications aux domaines de la biométrie

Résumé : Multiples sont les domaines qui font appel aux techniques de la reconnaissance de formes, cela concerne aussi bien la médecine, la météo ou encore la sécurité. Cette reconnaissance de forme étant confiée à des machines dotées d'une intelligence dite artificielle et de ce fait, toute nouvelle approche fiable pouvant augmenter cette intelligence doit pouvoir trouver des domaines de son application.

Notre travail donc, ambitionne de proposer une nouvelle approche de reconnaissance de formes basée sur le principe du passage du discret au continu des entités à traiter. L'avantage de notre algorithme réside dans la faculté de traiter dans sa globalité l'information concernée, de ce fait, de bon taux de reconnaissance sont obtenus avec des temps de réponse plus réduits.

Comme applications concrètes, nous avons choisis le domaine de biométrie, ce choix étant motivé par l'importance de la sécurité dans notre quotidien. Les applications concerneront la biométrie des empreintes digitales, et celle des signatures numériques. Nous pensons que ces applications ne présentent qu'un des aspects possibles de notre approche dite le passage du discret au continu.

Mots clés : Discrète au continu, reconnaissance de formes, PPM, comparaison.

Abstract:

There are many fields that use techniques of pattern recognition, whether it concerns medicine, weather or safety. Since this form recognition is entrusted to machines equipped with an artificial intelligence, any new reliable approach that can increase this intelligence must be able to find areas of its application.

Our work, therefore, aims to propose a new approach to pattern recognition based on the principle of switching from discrete to continuous. The advantage of our algorithm lies in the ability to handle the information in its entirety, so good recognition rates are obtained with shorter response times.

As practical applications, we chose the field of biometrics, this choice being motivated by the importance of safety in our daily lives. Applications will include fingerprint and digital signatures. We believe that these applications present only one of the possible aspects of our approach, called the transition from discrete to continuous.

Keywords: Discrete to continuous, pattern recognition, PPM, matching.

Contribution à la reconnaissance de formes discrètes par passage au continu : applications aux domaines de la biométrie

Année : 2017 N° thèse : 85/ST2I
Nabil AQILI

Année : 2017



Thèse N° : 85/ST2I

École Nationale Supérieure d'Informatique et d'Analyse des Systèmes
Centre d'Études Doctorales en Sciences des Technologies de l'Information et de l'Ingénieur

THÈSE DE DOCTORAT

CONTRIBUTION A LA RECONNAISSANCE DE FORMES
DISCRETES PAR PASSAGE AU CONTINU : APPLICATIONS
AUX DOMAINES DE LA BIOMETRIE

Présentée par

Nabil AQILI

Le 28/10/2017

Formation doctorale : Informatique

Structure de recherche : LRGE/ERIT (Equipe de recherche en informatique et télécommunications)

JURY

Professeur Rachid OULAD HAJ THAMI PES, ENSIAS, Université Mohammed V de Rabat	Président & Rapporteur
Professeur Ahmed HAMMOUCH PES, ENSET, Université Mohammed V de Rabat	Directeur de thèse
Professeur Mourad RAJI PA, ENSET, Université Mohammed V de Rabat	Invité (Co-Encadrant)
Professeur Najlae IDRISSE PH, EST, Université Sultan Moulay Slimane, Béni Mellal	Rapporteur
Professeur Mohamed Nabil SAIDI PH, INSEA, Rabat	Rapporteur
Professeur Jamal MHAMDI PES, ENSET, Université Mohammed V de Rabat	Examineur
Professeur Abdelilah JILBAB PA, ENSET, Université Mohammed V de Rabat	Invité

*À mon père et ma mère
À mes sœurs et frères
À mes Nièces Salma et Ghita*

Remerciements

En premier lieu, je tiens à remercier mon directeur de thèse, monsieur HAMMOUCH Ahmed, pour la confiance qu'il m'a accordée en m'acceptant dans son équipe de recherche, pour ses multiples conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche.

Je tiens à remercier Prof. OULAD HAJ THAMI Rachid pour avoir accepté de présider mon jury de thèse.

Je remercie également Prof. IDRISSI Najlae, Prof. SAIDI Mohamed Nabil, Prof. CHAFIK Nacir, Prof. MHAMDI Jamal et Prof. JILBAB Abdelilah pour l'honneur qu'ils m'ont fait en faisant partie de mon jury de thèse.

Je tiens à remercier profondément Prof. JILBAB Abdelilah pour avoir participé à l'encadrement de cette thèse. Ses connaissances ont bien favorisé le déroulement de ma thèse.

Je voudrais remercier tout particulièrement Prof. RAJI Mourad qui m'a encadré tout au long de ces années de thèse. Il a toujours été disponible, à l'écoute de mes nombreuses questions, et s'est toujours intéressé à l'avancée de mes travaux. Les nombreuses discussions que nous avons eues ainsi que ses conseils sont pour beaucoup dans le résultat final de ce travail. Sa capacité d'analyse et son enthousiasme m'ont montré que le monde de la recherche pouvait être un univers passionnant. Enfin, ses nombreuses relectures et corrections de cette thèse ont été très appréciables. Cette thèse lui doit beaucoup. Pour tout cela merci.

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Enfin, je ne pourrais finir ces remerciements sans penser à ma famille dont l'affection, l'amour, le soutien et l'encouragement constants m'ont été d'un grand réconfort et ont contribué à l'aboutissement de ce travail.

Résumé

Multiplés sont les domaines qui font appel aux techniques de la reconnaissance de formes, cela concerne aussi bien la médecine, la météo ou encore la sécurité. Cette reconnaissance de forme étant confiée à des machines dotées d'une intelligence dite artificielle et de ce faite, toute nouvelle approche fiable pouvant augmentée cette intelligence doit pouvoir trouver des domaines de son application.

Notre travail donc, ambitionne de proposer une nouvelle approche de reconnaissance de formes basée sur le principe du passage du discrète au continu des entités à traiter. L'avantage de notre algorithme réside dans la faculté de traiter dans sa globalité l'information concernée, de ce fait, de bon taux de reconnaissance sont obtenus avec des temps de réponse plus réduits.

Comme applications concrètes, nous avons choisis le domaine de biométrie, ce choix étant motivé par l'importance de la sécurité dans notre quotidien. Les applications concerneront la biométrie des empreintes digitales, et celle des signatures numériques. Nous pensons que ces applications ne présentent qu'un des aspects possibles de notre approche dite le passage du discrète au continu.

Mots clés : Discrète au continu, reconnaissance de formes, PPM, comparaison.

Abstract

There are many fields that use techniques of pattern recognition, whether it concerns medicine, weather or safety. Since this form recognition is entrusted to machines equipped with an artificial intelligence, any new reliable approach that can increase this intelligence must be able to find areas of its application.

Our work, therefore, aims to propose a new approach to pattern recognition based on the principle of switching from discrete to continuous. The advantage of our algorithm lies in the ability to handle the information in its entirety, so good recognition rates are obtained with shorter response times.

As practical applications, we chose the field of biometrics, this choice being motivated by the importance of safety in our daily lives. Applications will include fingerprint and digital signatures. We believe that these applications present only one of the possible aspects of our approach, called the transition from discrete to continuous.

Keywords: Discrete to continuous, pattern recognition, PPM, matching.

Table des matières

Introduction générale.....	9
i. Objectif principal.....	10
ii. Problématique.....	11
iii. Contributions.....	11
iv. Organisation de la thèse.....	12
Chapitre 1 L'état d'art.....	14
1.1 Introduction à l'appariement de formes.....	15
1.1.1 Appariement exact ou approximatif de formes.....	17
1.1.2 Appariement global ou partiel de formes.....	17
1.1.3 Algorithmes d'appariement de formes spécialisée et généralistes.....	17
1.1.4 Sources des modèles géométriques.....	17
1.2 Appariement de formes géométriques.....	18
1.2.1 Méthodes et algorithmes de correspondance.....	18
1.2.1.1 Corrélation.....	19
1.2.1.2 Relaxation.....	20
1.2.1.3 Programmation dynamique.....	20
1.2.1.4 Appariement de graphes.....	20
1.2.1.5 Hachage géométrique.....	21
1.2.1.6 Transformée de Hough.....	21
1.2.1.7 RANSAC.....	21
1.2.1.8 ICP.....	22
1.2.2 Méthodes et algorithmes d'alignement.....	23
1.2.2.1 Les méthodes globales.....	23
1.2.2.2 Appariement de formes à base de mesure de similarité.....	24
1.3 Appariement de formes à base de mesure de similarité.....	24
1.3.1 Distances, métriques et similarité.....	25
1.3.1.1 La distance Euclidienne.....	25
1.3.1.2 La distance de Bottleneck.....	26
1.3.1.3 La distance de Hausdorff.....	26
1.3.2 La Recherche des concordances exactes avec une métrique discrète.....	27

1.4	Appariement des nuages de points	28
1.4.1	Problème.....	29
1.4.2	Techniques.....	30
1.4.2.1	Clustering	30
1.4.2.2	Algorithmes basés sur les distances entre les points.....	31
1.4.2.3	Relaxation.....	32
1.4.2.4	Autres techniques.....	33
1.5	PPM appliqué aux objets non rigide	35
1.6	Complexité.....	36
1.7	Conclusion.....	37
Chapitre 2 Algorithme de reconnaissance de formes Discrètes par passage au Continu		38
2.1	Introduction.....	39
2.2	Performance de l'algorithme proposé	40
2.2.1	Définition du problème.....	40
2.2.2	Passage au continu de la forme F par interpolation	42
2.2.3	La recherche de \mathbf{T}'	42
2.2.4	Les propositions de correspondances : isomorphisme et transformation T...	44
2.2.5	Recherche de T.....	45
2.3	Outils.....	48
2.3.1	Interpolation	48
2.3.2	Extrapolation	49
2.3.3	Transformations géométriques	50
2.3.3.1	Translation.....	50
2.3.3.2	Mise à l'échelle	51
2.3.3.3	Rotation.....	51
2.3.3.4	Composition des transformations.....	52
2.3.4	La fonction coût QT.....	52
2.3.4.1	Représentation mathématique du problème de la fonction coût QT par les moindres carrés	53

2.3.4.2	Minimisation de la fonction coût QT	54
2.3.5	RMS.....	60
2.4	Résultats du testes.....	61
2.4.1	Application à un nuage 2D.....	61
2.4.2	Application à un nuage 3D.....	63
2.4.3	Benchmark des complexités.....	68
2.5	Conclusion.....	69
Chapitre 3 Applications à la reconnaissance biométriques		70
3.1	Introduction au biométrie	71
3.1.1	Vérification vs. Identification.....	75
3.1.2	Performance d'un système biométrique	75
3.1.3	Les caractéristiques biométriques	76
3.1.3.1	Visage	77
3.1.3.2	Empreintes digitales	78
3.1.3.3	Géométrie de la main.....	78
3.1.3.4	Cachet de la paume (Palmprint).....	79
3.1.3.5	Iris	79
3.1.3.6	Frappe au clavier	80
3.1.3.7	Signature.....	80
3.1.3.8	Voix.....	81
3.1.3.9	Démarche.....	81
3.2	Application à la reconnaissance d'empreintes digitales	82
3.2.1	Introduction	82
3.2.2	Formulation du problème.....	83
3.2.3	Score de similarité.....	85
3.2.4	Notre méthode: du Discret vers le Continu.....	85
3.2.5	Résultats expérimentaux	86
3.2.6	Conclusion.....	88
3.3	Application à la reconnaissance de la signature	89
3.3.1	Introduction	89
3.3.2	Prétraitement et extraction des caractéristiques	89

3.3.2.1	Prétraitement	89
3.3.2.2	Extraction des caractéristiques	90
3.3.3	Algorithmes de mise en correspondance et décision	91
3.3.3.1	Dynamic Time Warping.....	91
3.3.3.2	Notre méthode: du Discret vers le Continu.....	92
3.3.3.3	Décision	93
3.3.4	Résultats.....	94
3.3.4.1	La base de donnée ATVS-SSig DB 1.....	95
3.3.4.2	Résultats expérimentaux	96
3.3.5	Conclusion.....	98
3.4	Conclusion.....	99
	Conclusion générale et perspectives.....	100
	Références	101

Liste des figures

Figure 1.1 La forme F	16
Figure 1.2 La forme SF.....	16
Figure 1.3 Superposition de SF sur F.....	16
Figure 1.4 La forme F	16
Figure 1.5 La forme SF.....	16
Figure 1.6 Correspondances entre SF et F	16
Figure 1.7 Un ensemble de caractéristiques qui sont compatibles avec un modèle.	19
Figure 2.1 La forme F en 3D (à gauche) et la sous-forme SF (à droite).....	40
Figure 2.2 Schéma du simplex	58
Figure 2.3 Model set F avec n=153	61
Figure 2.4 Exemple du data set SF avec m= 16. Les points sont pris aléatoirement et d'une manière successive	61
Figure 2.5 Exemple du data set SF avec m=16. Les points sont pris d'une manière aléatoire et non successive	62
Figure 2.6 Temps d'exécution pour chaque ensemble de points de SF de taille m	63
Figure 2.7 Exemple des modèles 3D sous forme du nuage de points	64
Figure 2.8 Model set F avec n=3093	65
Figure 2.9 Exemple du data set SF pour le problème d'appariement partiel	65
Figure 2.10 La durée moyenne du temps d'exécution pour chaque taille de SF	66
Figure 2.11 Le temps moyen d'exécution pour huit modèles différents.....	66
Figure 2.12 Le temps moyen d'exécution pour 26 modèles pour le cas d'appariement complète (n = m).....	67
Figure 3.1 Des exemples de traits biométriques qui peuvent être utilisés pour l'authentification d'un individu.....	73
Figure 3.3 Echantillons de signatures de la base de données ATVS	95
Figure 3.4 Les échantillons de signature suivant une variabilité intersession d'une personne	96
Figure 3.5 Influence de la taille de la base de données sur le seuil.....	96
Figure 3.6 Taux de reconnaissance de signatures authentiques et falsifiées	97

Liste des tableaux

Tableau 2:1 Complexité de certains algorithmes	68
Tableau 3:1 : Sensibilité et 1- Spécificité.....	87
Tableau 3:2 : Exemple de caractéristiques communes.....	91
Tableau 3:3 : RMS pour l'utilisateur ID_1	93

Organigramme

Organigramme 2—1 Organigramme de l'algorithme	47
---	----

Introduction générale

La reconnaissance de formes est une activité dans laquelle les humains sont excellents. Nous sommes en mesure d'identifier la source et la nature des informations reçues par nos organes sans avoir fait d'effort perceptible. Nous le faisons presque tout le temps et sans effort. Ce qui est encore plus impressionnant c'est l'exactitude avec laquelle nous pouvons exécuter des tâches de reconnaissance même dans des conditions non-idéales, par exemple, quand les informations que nous avons à traiter, sont vagues, imprécises ou même incomplètes. En fait, la majeure partie de nos activités quotidiennes est basée sur l'exécution de tâches de reconnaissance de formes diverses tel l'identification des visages, la compréhension de la parole même en présence du bruit, la reconnaissance des lettres et des mots, la reconnaissance de l'odeur et etc.

La formalisation algorithmique de cette faculté de la perception humaine est à la base des fondements de la reconnaissance de formes. En effet, dans chacun de ces exemples cités ci-dessus, les organes sensoriels envoient des signaux au cerveau qui les traite très rapidement. Ce traitement consiste à comparer le signal reçu avec un ensemble de modèles déjà stockés en mémoire et de décider à qui il correspond. Ces modèles stockés en mémoire sont des connaissances que chaque individu avait apprises. Un individu ne peut identifier que ce qu'il connaît déjà. Pour l'être humain ce modèle est une représentation mentale de l'objet qui peut être apprise en retenant les caractéristiques les plus discriminantes de l'objet. Ces caractéristiques peuvent être de toutes sortes d'attributs de l'objet : géométrie, couleur, texture, taille, volume, odeur et etc. Il s'agit d'une implémentation neurobiologique de l'appariement de gabarit (Template-matching) [Pal and Amita, 2001].

En informatique, la reconnaissance de formes vise à élaborer un ensemble de techniques et des logistiques capables de reproduire, et si possible, de dépasser les capacités humaines de perception, visuelles et auditives. En effet, les progrès scientifiques et techniques nous permettent aujourd'hui d'imiter certaines de ces facultés à l'aide des machines. Sans aller jusqu'à l'idée absurde de remplacer complètement l'homme par une machine, il y a, malgré tout, beaucoup de cas particuliers où les travaux de reconnaissance simples effectués par l'homme peuvent être confiés à une machine. Par exemple, pour l'être humain il lui faut autant du temps pour trier des lettres, placer les mêmes vis dans les mêmes trous, alors que ces tâches ne prennent qu'une fraction de seconde pour une machine. Aussi au niveau de sécurité, il peut être très dangereux pour un policier de désamorcer une bombe d'attentat ou

pour un technicien de réparer une centrale nucléaire. Une machine ou un robot peut effectuer des tâches similaires. Même sur le plan économique, une machine autonome, est généralement moins coûteuse et plus productive qu'un ouvrier.

Les exemples précédents, sans être exhaustifs, illustrent la diversité des problèmes où intervient la reconnaissance des formes. Cette dernière fait appel à plusieurs disciplines parmi lesquels : les statistiques, la linguistique, la recherche opérationnelle, la théorie des communications, la biologie, l'optique, l'électronique et bien d'autres domaines scientifiques.

Il n'y a pas une théorie unifiée de la reconnaissance de formes. Ceci est principalement dû à deux raisons. La première est la grande variété dans les concepts provenant des disciplines citées, les applications et les moyens techniques. Le deuxième est le fossé creusé entre la théorie et la pratique de la reconnaissance des formes que la recherche n'a pas réussi à combler. Les théories disponibles aujourd'hui ne permettent pas de cerner tous les problèmes pour en déduire des systèmes optimisés et pratiques, même au niveau des produits commercialisés, on se contente des performances raisonnables, acceptables ou utilisables sans insister sur une rigueur absolue.

Les approches de la reconnaissance de forme sont très nombreuses. Les méthodes utilisées pour la reconnaissance de forme comprennent des techniques de réseau de neurones (neural network) [Looney, 1997; Pao, 1989], des méthodes statistiques (statistical methods) [Duda and Hart, 2000; FUKUNAGA, 1990; Jain et al., 2000], des méthodes syntaxiques, dites aussi structurelles auxquelles d'ailleurs notre méthode fait partie (syntactical methods) [Gonzalez and Thomason, 1978] et des méthodes d'indexation (indexing methods) [Brin, 1995; Clarkson, 1999; Portegys, 1995; Yianilos, 1993].

i. Objectif principal

L'objectif de ce travail est de proposer une nouvelle méthode de reconnaissance de formes discrètes basée sur le passage au continu. Dans ce travail, nous allons nous intéresser à répondre à la question de l'inclusion d'un nuage de points dans un autre. Pour se faire, nous allons rechercher la transformation T , composée de rotations, translations et éventuellement changement d'échelle pouvant superposer l'un des objets sur l'autre. Ainsi l'existence ou pas d'une telle transformation confirmerait ou infirmerait la similarité de deux objets.

La recherche directe d'une telle transformation peut s'avérer très coûteuse. Nous allons alors, proposer une autre façon de représenter les entités (passage du discret vers le continu) à traiter afin que la recherche de la dite transformation devient moins coûteuse voir même plus performante.

ii. Problématique

La majorité des approches existantes utilisent la manière la plus classique consistant à comparer les matrices de distances intra-points des objets à reconnaître. Le passage par la dite matrice de distances, amène ces méthodes à traiter les formes point par point. Ce type d'approches conduit inévitablement à des temps de réponses importantes (explosion combinatoires, retour en arrière ...).

Après avoir attiré l'attention sur l'origine de cet inconvénient, la question qui se pose est de savoir s'il existe une alternative à ce genre d'approche, autrement dit, existe-t-il une méthode permettant de ne pas passer par les matrices de distances (pour obtenir les correspondances) d'une part et de considérer les objets à traiter dans leurs globalités (sans le point par point).

Si une telle méthode existe, cela a pour conséquences, comme on le verra, l'obtention de meilleurs résultats tant au niveau temps de réponse (complexité) qu'au niveau des taux de reconnaissance.

iii. Contributions

Notre principale contribution porte sur la présentation d'une nouvelle méthode de reconnaissance de formes discrètes basée sur un passage au continu. Afin de valider notre approche, nous l'avons appliqué sur des objets arbitraires (free-form) obtenus à partir de la base de données SHREC où les objets (animaux, oiseaux chaises, ...) se présentent sous la forme de nuage de points 3D. Dans cette première contribution nous avons démontré que notre approche admet un meilleur temps réponse comparé aux autres approches de sa catégorie. Les tests nous ont permis de définir la complexité de notre algorithme qui est de l'ordre de $O(m \log n)$, et qui est, sans doute, l'un des meilleurs de sa catégorie.

En guise d'applications, nous avons choisis des domaines relevant de la biométrie à savoir la reconnaissance d'empreintes digitales et la reconnaissance de la signature en ligne. Ci-dessous, nous présentons les deux applications.

La reconnaissance d'empreintes digitales

Dans cette application, les données ont été obtenues à partir de la base de données UPEK. Les résultats obtenus, comparés aux autres approches, se sont avérés très intéressants. En effet notre méthode a obtenu un taux de reconnaissance de fausses empreintes de 100% alors celui de vraies empreintes avoisine les 86%. Les tests dans ce cas se sont basés sur 12 points caractéristiques (minuties). Dans ces tests, notre algorithme a été utilisé de façon brute sans aucun filtrage au préalable, il aurait possible d'améliorer davantage ces résultats en utilisant des techniques telle les méthodes dites de triangulation et/ou d'indexation, ainsi une recherche effective utilisant notre approche ne s'opérerait qu'après passage de ces opérations de filtrage.

La reconnaissance signature en ligne

En se basant sur les caractéristiques (position et pression) qui sont utilisées dans l'étape d'extraction des signatures, les expériences ont été réalisées en utilisant la base de données ATVS. Les performances globales du système sont honorables comme les prouvent les résultats suivants : il rejette moins 3 % de vraies signatures et accepte environs 20% de fausses signatures, alors que la meilleure approche, qui utilise la technique dite de « Modèle de Markov caché » intégrant plus de caractéristiques, rejette moins 3 % de vraies signatures et accepte environs 7 à 9 % de fausses signatures .

Au même titre que le traitement des empreintes digitales, l'algorithme a été utilisé avec un minimum d'adaptation faisant appel uniquement aux fonctions de base (position et pression), cela sous-entend qu'en prenant d'autres fonctions telle la vitesse, l'amplitude..., les performances seraient sans doute meilleures.

iv. Organisation de la thèse

Ce document est divisé en trois chapitres:

- Dans le premier chapitre, nous avons donné un rappel des différentes approches relatives à la reconnaissance de formes et en particulier celles traitant des informations sous la forme de nuage de points. Afin d'apprécier les différentes approches, nous avons dressé les complexités algorithmiques relatives à chacune d'entre elles. Il ressort de cette comparaison, que l'une des raisons de la complexité de ces méthodes, réside dans le caractère discret des entités à traiter qui imposent un traitement point par point, induisant inévitablement des risques d'explosions combinatoires.

- Le deuxième chapitre constitue l'essentiel de notre travail. C'est dans ce chapitre que nous présentons notre approche qui se base essentiellement sur le passage du discret vers le continu. De ce fait, la recherche directe de la transformation de superposition des entités à traiter, sans connaissance au préalable des correspondances des points, devient possible. Pour toutes les méthodes qui existent, le temps de réponse est une fonction croissante des tailles des objets à traiter, nous allons voir que dans le cas de notre approche, cette règle fait exception, nous essaierons d'ailleurs de donner une explication à ce comportement.

Pour illustrer le comportement et la complexité de notre algorithme, nous allons utiliser des tests en 2D et 3D. Enfin, nous démontrerons que notre approche peut être étendue à ND.

- Le troisième chapitre est destiné à représenter l'efficacité de notre algorithme dans le cadre des applications réelles tel le domaine de la biométrie. Le recours aux techniques de reconnaissances biométriques demeure une méthode convenable pour reconnaître les individus et essentiellement pour établir et renforcer la sécurité des individus, des établissements et des états. En fait, l'efficacité des techniques de reconnaissances biométriques (entre autres) est essentiellement dépendante de l'algorithme de comparaison utilisé.

Cette partie traite deux aspects de la biométrie à savoir la biométrie des empreintes digitales et celle de la signature en ligne. Enfin nous présentons les résultats expérimentaux pour l'évaluation de l'approche proposée.

Nous terminons cette thèse par une synthèse de nos différentes contributions et nous donnons quelques perspectives qui peuvent donner suite à ces travaux.

Chapitre 1 L'état d'art

Dans ce chapitre, nous donnons un rappel des différentes approches relatives à la reconnaissance de formes et en particulier celles qui traitent des informations sous la forme de nuage de points. Une comparaison de la complexité algorithmique de chacune de ces approches sera donnée. Comme on le verra, une des raisons de la complexité de ces méthodes réside dans le caractère discret des entités à traiter et qui impose un traitement point par point induisant inévitablement des risques d'explosions combinatoires.

1.1 Introduction à l'appariement de formes

L'appariement de formes peut être divisé en deux catégories : appariement de formes combinatoires tel que la reconnaissance de chaînes de caractères [Hall and Dowling, 1980; Navarro, 2001] ou de séquences ADN [Bishop and Thompson, 1986; Hérisson et al., 2007], et l'appariement de formes spatiales. Comme exemples de ce dernier, on peut citer les domaines de la vision par ordinateur qui définissent l'appariement de formes spatiales comme un problème où nous sommes amenés à trouver les correspondances entre deux images (méthodes photométriques [Hel-Or et al., 2014; Seo et al., 2005; Tombari et al., 2007; Torreão and Fernandes, 1998]) ou bien détecter des ressemblances entre des modèles géométriques (méthodes géométriques [Alt and Guibas, 1999; Chew et al., 1995; Chew and Kedem, 1992; Gavrilov et al., 1999; Goodrich et al., 1999; Veltkamp and Hagedoorn, 2001]).

Dans le cas de l'appariement de formes dans le domaine spatial, ce qui concerne notre travail, nous traitons précisément l'appariement des formes géométriques. Un exemple simple serait l'appariement de formes géométriques basé sur la recherche d'une transformation. Cette situation pourrait être présentée par deux modèles d'entrée F et SF , tels représentés sur les figures 1.1 et 1.2, respectivement. Le problème donc, est de trouver une transformation T pour laquelle $T(SF)$ est similaire à F . Ici, la transformation T est autorisée à être une combinaison de mise à l'échelle, de rotation et de translation. Les méthodes utilisées dans ce cas doivent trouver une transformation T telle que celle représentée sur la figure 1.3 où $T(SF)$ est superposable sur F .

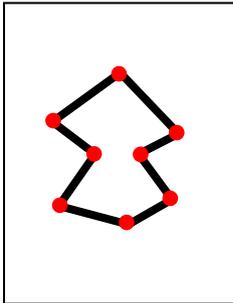


Figure 1.1 La forme F

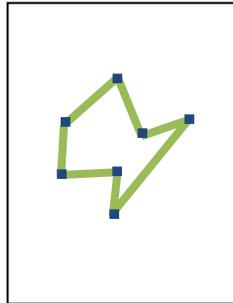


Figure 1.2 La forme SF

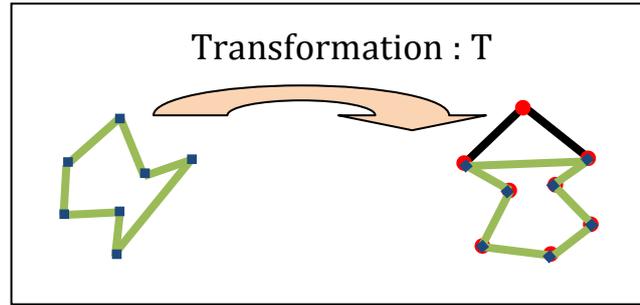


Figure 1.3 Superposition de SF sur F

Un autres exemple de correspondance serait les méthodes basées sur la recherche de correspondances entre les composants de formes SF et F (figure 1.4, 1.5 et 1.6) sans s'occuper de la superposition de l'une des formes sur l'autre et donc sans rechercher une transformation. Une bonne partie de ces méthodes se basent sur les matrices de distances.

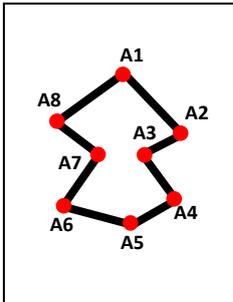


Figure 1.4 La forme F

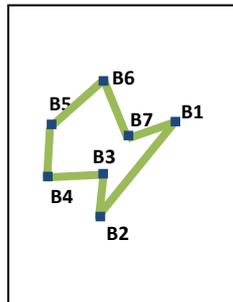


Figure 1.5 La forme SF

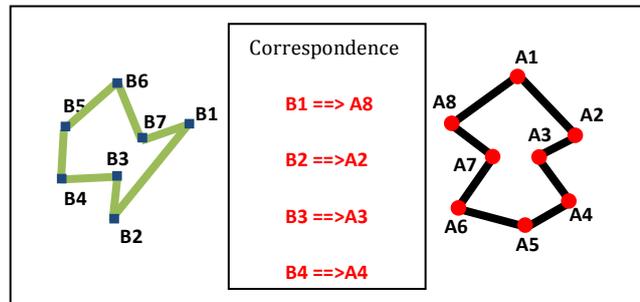


Figure 1.6 Correspondances entre SF et F

En général, l'entrée de ces algorithmes d'appariement de formes, est une paire de modèles et la sortie est une transformation et/ou une liste de correspondances. Le comportement de ce genre d'algorithmes est basé sur trois composantes : la collection des modèles, la classe de transformation et le critère de validation. Les modèles peuvent être des images, des signaux ou des modèles géométriques. On peut citer comme exemples de classes de transformations : les déplacements (translations) qui conservent les distances et les angles orientés, les isométries qui conservent les distances et les angles sans orientation, les similitudes qui conservent les rapports de distance, les transformations affines qui conservent le parallélisme, les transformations homographiques qui conservent les droites et les transformations élastiques qui ne préservent pas nécessairement toutes les distances. Quant au critère de validation, la transformation doit réduire au minimum la valeur d'une mesure de similarité utilisant une métrique adéquate, ou bien maximiser le plus possible le taux de correspondance.

1.1.1 Appariement exact ou approximatif de formes

L'appariement exact de formes est de trouver une transformation en vertu de laquelle les modèles deviennent identiques. Quant à l'appariement approximatif de formes, est de trouver une transformation en vertu de laquelle les deux modèles deviennent semblables mais pas nécessairement identiques. Pour l'appariement de formes à base de correspondance, l'appariement exact vise à trouver pour chaque point de la première forme son homologue dans la deuxième forme. En revanche, dans l'appariement approximatif on s'occupe du taux de vraies correspondances.

1.1.2 Appariement global ou partiel de formes

Dans l'appariement global de formes (SF et F ont la même taille), un modèle est apparié avec un autre modèle dans son ensemble. Alors que dans l'appariement partiel (taille de SF plus petite que celle de F), un modèle est apparié avec seulement une partie de l'autre modèle. Dans ce dernier cas, le problème devient plus complexe lorsqu'une mise à l'échelle est nécessaire.

1.1.3 Algorithmes d'appariement de formes spécialisée et généralistes

Les méthodes d'appariement de formes sont spécialisées à des degrés divers. Certaines méthodes de reconnaissance de formes sont spécialisées pour détecter ou reconnaître une classe limitée de formes comme par exemple des cercles et des carrés. Les méthodes d'appariement de formes généralistes (indépendants du contexte) ne sont pas spécialisées pour une tâche spécifique et elles s'appliquent pour toutes les classes sans restrictions. Ces méthodes ne tiennent pas compte des hypothèses de dépendance au domaine d'application.

Il est à rappeler que notre travail est fait selon la méthode d'appariement généraliste avec une adaptation dans des domaines spécifiques.

1.1.4 Sources des modèles géométriques

Un modèle géométrique est un sous-ensemble dans l'espace représenté généralement en 2D ou en 3D. En deux dimensions, les modèles sont des unions finies de primitives géométriques telles que des segments de ligne, des segments de courbe et des triangles. Dans la dimension générale, les primitives géométriques sont des points (nuage de points).

Les modèles sont obtenus à partir de :

- Modèles géométriques 2D ou 3D : ils sont représentés sous la forme d'une liste de primitives, ou sous la forme d'une représentation de frontières ou sous la forme d'un nuage de points. La dimension peut être étendue à d'autres caractéristiques (nD) car la représentation d'un point n'est pas limitée aux primitifs géométriques et toutes les autres caractéristiques extraites peuvent aussi définir les objets à traiter.
- Image : le modèle est défini directement par les valeurs d'intensité, ou bien par d'autres caractéristiques telles que la couleur, la texture, la forme, etc.
- Signal vocal, ou tout autre type des signaux.

1.2 Appariement de formes géométriques

Deux principaux paradigmes peuvent être identifiés dans l'appariement de formes géométriques : les méthodes de correspondance et les méthodes d'alignement. Les méthodes de correspondance ajustent les modèles en adaptant les paires de primitives géométriques. Les méthodes d'alignement ajustent les modèles en adaptant les unions des primitives géométriques. Les deux paradigmes ne sont pas exclusivement mutuels. Certaines méthodes peuvent être considérées comme une combinaison de ces deux approches.

1.2.1 Méthodes et algorithmes de correspondance

Les méthodes de correspondance combinent les primitives géométriques qui constituent les modèles d'entrée. Par exemple, la mise en correspondance des polygones dans le plan en jumelant les segments de lignes des deux polygones. Autres exemples des méthodes de correspondance sont l'appariement de graphes (graph matching [Brandstädt et al., 1999; Bunke, 2000; Conte et al., 2004; Isenor and Zaky, 1986; Livi and Rizzi, 2013; Torresani et al., 2008; von der Malsburg, 1988]) et le hachage géométrique (geometric hashing [Bourdon and Medioni, 1990; Grimson and Huttenlocher, 1990; Lamdan and Wolfson, 1988; Wolfson and Rigoutsos, 1997]) qui seront discutés plus tard. Le comportement des méthodes de correspondance dépend plus de la représentation et de la topologie des motifs que de la géométrie de ces derniers. Dans l'exemple ci-dessus, il s'agit de rechercher à correspondre les deux polygones, l'appariement résultant est supposé être bon si les deux polygones ont des distributions similaires au niveau des sommets et des arêtes. Ce comportement est acceptable ou même souhaitable dans les problèmes d'appariement des modèles en CAO (figure 1.7). Cependant, si les modèles sont obtenus avec des méthodes d'extraction de caractéristiques, les hypothèses sur la topologie et la représentation des modèles généralement n'en tiennent pas

compte. L'avantage des méthodes de correspondance réside dans leur pouvoir à produire une relation explicite entre les primitives géométriques des deux modèles.

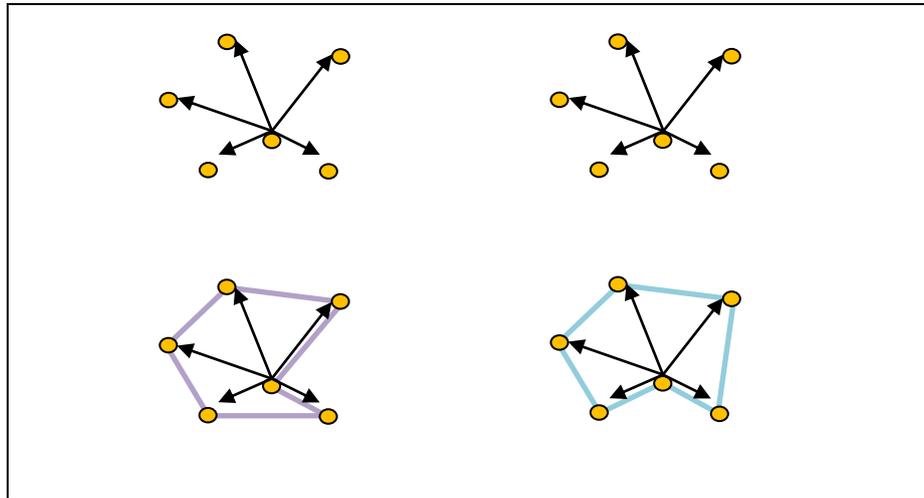


Figure 1.7 Un ensemble de caractéristiques qui sont compatibles avec un modèle.

Différentes stratégies d'utilisation des primitives et des contraintes ont donné lieu à de grandes classes de méthodes de mise en correspondance. Nous proposons de donner ici le principe de quelques unes. Nous ferons aussi un passage sur quelques exemples d'algorithmes qui sont largement utilisés dans le processus de mise en correspondance [Far, 2005; Faugeras, 1993; Hagedoorn, 2001; Horaud and Monga, 1995].

1.2.1.1 Corrélation

Très utilisée en imagerie, cette approche permet de mettre en correspondance des primitives données sous la forme d'un nuage de points. Elle met en correspondance des zones d'intérêt. Cette méthode consiste à comparer le voisinage du point à apparier (une petite région ou fenêtre centrée sur ce point) avec le voisinage des correspondants potentiels situés sur le modèle associé en s'appuyant uniquement sur l'information de l'intensité.

L'avantage de cette méthode est le fait qu'elle n'exige pas beaucoup de prétraitement. En revanche, utilisant uniquement l'information bas-niveau, elle est sensible au bruit et aux variations d'intensité. De plus, elle souffre aussi des problèmes liés à la présence d'occlusion dans la fenêtre de corrélation. Généralement les domaines d'application de la corrélation sont limités à l'imagerie.

1.2.1.2 Relaxation

Le problème de mise en correspondance peut être vu comme un problème d'étiquetage : pour un point donné, on cherche à lui associer un point correspondant ou une disparité. Dans cette approche, les appariements sont amenés à coopérer entre eux.

Dans un premier temps, un ensemble d'appariements initiaux, appelés hypothèses sont définis à partir des comparaisons locales des attributs.

Puis, ces hypothèses sont confrontées les unes aux autres dans le processus de relaxation :

- une hypothèse sera renforcée si les hypothèses voisines sont compatibles avec elle.
- dans l'autre cas, elle sera affaiblie ou abandonnée.

Le processus de relaxation se termine soit après un nombre donné d'itérations, soit après convergence vers un ensemble d'hypothèses compatibles.

1.2.1.3 Programmation dynamique

La mise en correspondance peut aussi se faire par la recherche d'un chemin optimal dans un espace bidimensionnel, généralement représenté par une matrice. Chaque élément de cette matrice représente le coût de la mise en correspondance de deux primitives. Un chemin dans la matrice est composé de transitions verticales ou horizontales lorsqu'une primitive n'a pas de correspondant, et des transitions diagonales lorsqu'il y a appariement.

Le principe de la programmation dynamique est de choisir, à chaque étape, l'appariement permettant d'obtenir le chemin de coût minimum. Le coût d'un appariement est la mesure de la ressemblance des primitives calculées à l'aide de leurs attributs.

1.2.1.4 Appariement de graphes

L'appariement de graphes est une méthode de correspondance. La structure d'un modèle est décrite par un graphe. Dès que les graphes sont construits, l'appariement est effectué entre eux. L'appariement peut être atteint par la recherche des isomorphismes du graphe ou par la recherche des sous-graphes isomorphismes. Remarquons au passage qu'il n'y a aucun algorithme de complexité polynômial pour ces deux problèmes. En outre, il n'est pas démontré que le problème de l'isomorphisme de graphe est NP-complet. Cependant, on sait que le problème d'isomorphisme de sous-graphe est NP-complet.

En raison de sa nature discrète, les graphes sont peu robustes aux erreurs, le succès de la technique dépend de l'extraction correcte des graphes à partir de l'entrée. Une autre limitation des graphes est la discrimination, de grandes classes de modèles partagent le même graphe.

1.2.1.5 Hachage géométrique

Le hachage géométrique [Wolfson, 1990; Wolfson and Rigoutsos, 1997] est une autre classe des méthodes de correspondance pour lequel les primitives géométriques, qui forment un modèle, sont utilisées pour générer une description normalisée de la structure du modèle dans son ensemble. Par exemple, pour un ensemble de points dans le plan, une description invariable est générée. Toutes les descriptions normalisées sont utilisées pour construire une table de hachage.

Pour être plus précis, supposons que les occurrences d'un motif A d'un sous-ensemble de divers modèles B doivent être détectés de manière efficace. Dans ce cas, une structure de hachage est construite pour A et les descriptions normalisées de B sont insérées dans cette structure.

Cette technique est robuste pour les points manquants (missing points) et l'occlusion. Le hachage géométrique est aussi efficace au niveau du temps d'exécution si un ensemble fixe de modèles doit être reconnu. Un inconvénient de cette méthode est que la structure de données, qui dépend de la taille des objets à traiter, peut devenir très grande. La complexité de la transformation en termes du nombre de degré de liberté est aussi une autre source de soucis pour cette méthode.

1.2.1.6 Transformée de Hough

Dans [VC, 1962], Hough a introduit sa transformation pour la détection de lignes et de courbes qui s'étend théoriquement à la détection de n'importe quelle forme géométrique paramétrée (cercle, ellipse, etc.). Par la suite, la transformée de Hough a été généralisée par Ballard [Ballard, 1981] pour la détection de courbes définies arbitrairement.

Le problème posé est celui de la recherche et de la détection de lignes qui seraient éventuellement présentes dans un modèle à analyser. Le principe qui sous-entend la transformée de Hough est qu'il existe un nombre infini de lignes qui passent par un point dont la seule différence est l'orientation (l'angle). Le but de la transformée est de déterminer lesquelles de ces lignes sont les plus fréquentes dans l'image analysée.

1.2.1.7 RANSAC

RANSAC (RANdom SAmple Consensus) est un algorithme proposé par Fischler et Bolles [Fischler and Bolles, 1981]. Son principe est le suivant : Pour un ensemble de données issu d'une unique transformation inconnue, l'objectif est d'estimer les paramètres de la transformation tout en éliminant les données aberrantes.

Le principe de l'algorithme RANSAC est très proche de celui de la transformée de Hough. On cherche à déterminer des consensus à partir d'un échantillon, ainsi les hypothèses sont générées. La différence majeure avec la transformée de Hough est la manière dont ces hypothèses sont exploitées : au lieu de les agréger dans des accumulateurs, RANSAC consiste à les tester successivement en mesurant pour chacune le consensus qu'elle génère.

L'algorithme de RANSAC est caractérisé par sa simplicité de mise en œuvre. En outre, la transformation estimée est en pratique bien plus précise que celle obtenue avec la transformée de Hough.

1.2.1.8 ICP

ICP (Iterative Closest Point) introduit par Besl & McKay [Besl and McKay, 1992], est un algorithme très utilisé pour recaler deux entités géométriques. Chaque entité étant constituée d'un ensemble de points, l'objectif est de minimiser itérativement la distance entre ces deux nuages de points. Nous sommes donc amenés à trouver un ensemble de transformations afin de rapprocher ces deux entités géométriques.

L'algorithme ICP est principalement composé de deux étapes :

- Trouver les correspondances.
- Trouver les paramètres de la transformation.

L'algorithme itère ces deux étapes jusqu'à ce que les deux entités deviennent suffisamment proches.

- Etape 1 "Trouver les correspondances" :

La première étape de mise en correspondance consiste à associer pour tout point de la première entité géométrique le point le plus proche de la deuxième entité. Soit P et X deux entités à recaler, l'objectif est de trouver pour chaque point $p \in P$, un point $x \in X$ qui lui est le plus proche (exemple, en terme de distance Euclidienne). Cette correspondance est décrite par une fonction Y , telle que $Y(i) = j$ implique que le point p_i est le correspondant du point x_j .

- Etape 2 "Trouver les paramètres de la transformation" :

Cette deuxième étape consiste à optimiser, au sens des moindres carrés, la transformation rigide (composée de rotation(R) et de translation(t)) qui recale au mieux ces deux entités.

$$E(R, t) = (1/N) \sum_{i=1}^N \left\| x_{Y(i)} - R * p_i - t \right\|^2$$

avec N est le au nombre de paires de points.

L'algorithme est conceptuellement simple et couramment utilisé. Il révisé itérativement les transformations nécessaires pour minimiser la distance entre les points. Cependant, cet algorithme souffre des problèmes de convergence lorsque les paramètres initiaux sont mal initialisés, ainsi que des problèmes des minima locaux. Il est évident que le temps d'exécution s'en trouverait impacté. La procédure suivie pour définir le point le plus proche et le nombre de points dans les deux entités sont aussi des facteurs influençant l'efficacité de cet algorithme.

1.2.2 Méthodes et algorithmes d'alignement

Les méthodes d'alignement cherchent un appariement entre les deux modèles d'entrée, ignorant la façon dont les modèles sont représentés en termes de leurs primitives géométriques. Des exemples de méthodes d'alignement sont les méthodes globales (global methods) et les méthodes d'appariement de formes basés sur la mesure de la similarité.

1.2.2.1 Les méthodes globales

Les méthodes globales sont les méthodes d'alignement qui utilisent les informations géométriques pour trouver les correspondances. Un exemple est la cartographie de la zone de délimitation d'un motif sur la zone de délimitation d'un autre modèle. Un autre exemple de méthodes globales se traduit par la translation du centre de gravité d'un modèle sur le centre de gravité d'un autre modèle. Les techniques à base des Moments (Moment based techniques) [Chen, 1993; Khotanzad and Hong, 1990] généralisent le deuxième exemple. Elles peuvent être utilisées pour la recherche de la correspondance sous des isométries euclidiennes et des transformations de similitude. Les méthodes globales peuvent être aussi appliquées si les motifs d'entrée sont presque identiques, pour un certain nombre de groupe de transformation. En raison de leur simplicité, ces méthodes sont très rapides. En revanche, elles ne sont pas bien adaptées pour trouver des appariements plausibles entre deux modèles qui diffèrent beaucoup ni pour l'appariement partiel de formes.

1.2.2.2 Appariement de formes à base de mesure de similarité

Une mesure de similarité est une fonction qui assigne un nombre réel positif pour chaque paire de motifs. Un exemple typique de ce genre d'appariement est de trouver une transformation géométrique qui minimise cette mesure de similarité.

Tant que la mesure de similarité utilisée est définie sur les modèles comme étant des ensembles de points, les méthodes d'appariement de formes à base de mesure de similarité sont indépendantes de la représentation. Les performances de ces méthodes dépendent beaucoup du choix de la métrique utilisée pour la mesure de la similarité et de la classe de transformation. Contrairement aux méthodes globales, les méthodes d'appariement de formes à base de mesure de similarité peuvent être utilisées pour trouver des correspondances plausibles si les modèles diffèrent beaucoup. Ainsi elles peuvent être aussi utilisées dans l'appariement partiel de formes.

D'autre part, les méthodes d'appariement de formes à base de mesure de similarité tendent à être plus lentes que la recherche globale. Ces méthodes seront discutées dans la section suivante où plusieurs exemples de mesures de similarité seront donnés.

1.3 Appariement de formes à base de mesure de similarité

La mesure de similarité est une notion fondamentale dans la reconnaissance indépendamment de la technique utilisée. La similarité est utilisée en tant que mesure de ressemblance entre deux objets. La quantification de la similarité repose principalement sur la définition et la sélection des caractéristiques des objets impliqués dans l'estimation de celle-ci, autrement dit la métrique. Ces caractéristiques sont des mesures, des attributs ou des primitives extraites de l'objet et jugées pertinentes par rapport à sa représentation. Elles constituent l'espace des descripteurs dans lequel l'estimation est réalisée. En effet, deux objets A et B sont semblable s'ils ont des caractéristiques très proches.

Cette section présente plusieurs méthodes d'appariement de formes qui sont implicitement ou explicitement basées sur des mesures de similarité. D'abord, le problème fondamental de la recherche de congruence exacte entre deux formes est discuté. Ce type d'appariement de formes est implicitement basé sur une métrique discrète. Les méthodes qui sont explicitement basées sur des mesures de similarité seront aussi discutées. Ces mesures de similarité incluent la distance Euclidienne, la distance de Bottleneck et la distance de Hausdorff.

1.3.1 Distances, métriques et similarité

Les distances et les métriques sont utilisées pour comparer les ressemblances et les différences entre deux vecteurs. En fait, une métrique est une fonction qui décrit la différence entre deux points. Les propriétés désirables dans une mesure sont les suivantes :

- | | |
|--|--------------------|
| 1) $d(A, B) = d(B, A)$ | Symétrie |
| 2) $d(A, B) \geq 0$ | Positive |
| 3) $d(A, A) = 0 \Leftrightarrow A = A$ | Identité |
| 4) $d(A, B) = 0 \Leftrightarrow A = B$ | Unicité |
| 5) $d(A, B) + d(B, C) \geq d(A, C)$ | Inégalité triangle |

Si toutes ces propriétés sont respectées, alors on parle d'une distance métrique. Si l'inégalité triangulaire n'est pas respectée, on parle alors d'une distance semi-métrique.

La similarité est une autre manière pour comparer deux vecteurs. En effet, la distance mesure la différence entre deux vecteurs, alors que la similarité mesure la ressemblance entre deux vecteurs.

1.3.1.1 La distance Euclidienne

La distance euclidienne est la distance la plus couramment utilisée vu sa simplicité du calcul. Il s'agit d'une distance géométrique dans un espace multidimensionnel.

Soit R^k un espace euclidienne à n dimension. La distance euclidienne $d(A, B)$ qui sépare deux points $A\{a_1, a_2, \dots, a_n\}$ et $B\{b_1, b_2, \dots, b_n\}$ est donnée par la formule :

$$d(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (1)$$

Lorsque $n=1$, l'expression (1) donne la distance $d(A, B) = |a_1 - b_1|$ qui sépare deux points sur une même droite.

Lorsque $n=2$, l'expression (1) devient : $d(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$ et qui sépare deux points situés dans un même plan.

1.3.1.2 La distance de Bottleneck

La distance de Bottleneck, entre deux ensembles finis de même cardinalités, est le minimum sur toutes les bijections entre les ensembles sur la distance maximale entre deux points qui sont liés dans une bijection [Efrat and Itai, 1996; Efrat and Katz, 1996]. La définition formelle est la suivante :

Soient A et B deux sous-ensembles finis et m une métrique. Supposons que A et B ont la même cardinalité. Soit $w(A, B)$ l'ensemble de toutes les bijections de A à B . Ainsi, la distance de Bottleneck d_m sera définie comme :

$$d_m(A, B) = \text{Min}_{w \in W(A, B)} \text{Max}_{a \in A} m(w(a), a)$$

Efrat et Itai [Efrat and Itai, 1996] montrent que pour les ensembles de points de cardinalité n , le calcul de la distance de Bottleneck prend du temps $O(n^{\frac{3}{2}} \log n)$ et la minimisation de la distance de Bottleneck sous la translation peut être s'effectuer en temps $O(n^5 \log^2 n)$.

1.3.1.3 La distance de Hausdorff

La métrique de Hausdorff est la mesure de similarité la plus étudiée en géométrie algorithmique. Pour son calcul, il existe plusieurs définitions équivalentes. La définition donnée ci-dessous est souvent utilisée. La distance de Hausdorff directe (unilatéral) h_m sur des sous-ensembles compacts A et B avec une métrique m , est donnée par :

$$h_m(A, B) = \max_{a \in A} \min_{b \in B} m(a, b)$$

La métrique de Hausdorff peut être exprimée comme étant le maximum de deux distances de Hausdorff direct, elle est définie par :

$$h'_m = \max \{h_m(A, B), h_m(B, A)\}$$

Alt, Behrends et Blomer [Alt et al., 1995] montrent que pour deux ensembles de points finis dans \mathcal{R}^2 , la métrique de Hausdorff peut être calculée en $O((n + m) \log(n + m))$. Les résultats obtenus pour la minimisation de la métrique de Hausdorff dans \mathcal{R}^2 sont les suivantes : Huttenlocher, Kedem et Sharir [Huttenlocher et al., 1993a] décrivent un algorithme de complexité $O(nm(n + m) \log(nm))$ pour la minimisation sous translation. Chew et al. [Chew et al. 1997] ont donné un algorithme en temps $O((n + m)^5 \log^2(nm))$, pour la minimisation sous l'isométrie euclidienne.

La distance de Hausdorff direct est le maximum sur tous les points A qui ont des distances minimales à B . Techniquement, elle est le supremum et infimum puisque les ensembles ne doivent pas correspondre 1: 1.

En outre, notons que ceci est la distance minimale la plus grande, et non pas la distance maximale la plus petite comme chez Bottleneck. En revanche, la distance de Hausdorff est sensible aux valeurs aberrantes (outliers).

1.3.2 La Recherche des concordances exactes avec une métrique discrète.

Le problème de concordances (congruence) exactes est le suivant : étant donné deux modèles A et B, on cherche à trouver une transformation $g \in G$ sous laquelle l'image de g (A) est égale B. Typiquement, A et B sont des sous-ensembles finis de R^k et le groupe G consiste en des isométries Euclidiennes dans R^k . La métrique discrète désignée par C, est définie comme suit :

$$\begin{cases} C(A, B) = 0 & \text{si } A = B \\ C(A, B) = 1 & \text{sinon} \end{cases}$$

Le calcul de la concordance exacte entre A et B sous G, est équivalent au calcul du minimum sur $g \in G$ de $C(g(A), B)$. Les algorithmes de recherche de cette concordance, exécutent les pas de normalisation suivants : Les deux modèles sont translatés de telle sorte que leurs centres de gravité se trouvent sur l'origine. L'ensemble résultant est ensuite projeté sur une sphère unité R^k . Chaque point projeté est ensuite étiqueté en fonction de la distance par rapport à l'origine du repère. Les algorithmes de concordance exacte cherchent la correspondance de l'ensemble des points étiquetés. En deux dimensions, le problème de l'adéquation des ensembles de points marqués est réduit à la correspondance de chaîne (string matching) [Atallah, 1984; Hall and Dowling, 1980]. En trois dimensions, il est réduit à un test sur l'isomorphisme de graphe [Alt et al., 1988; Ambühl et al., 2000; Atkinson, 1987]. Dans les dimensions supérieures, les techniques de réduction de la dimension sont utilisées [Akutsu, 1998; Brass and Knauer, 2000]. Les algorithmes de recherche de concordance exacte trouvent la solution d'appariement seulement si les modèles sont égaux et sous un nombre restreint de groupes de transformations. Par conséquent, l'utilisation pratique de ces algorithmes est limitée. Ces algorithmes ne peuvent pas être généralisés à d'autres mesures de similitude que sur une métrique discrète. En revanche, la recherche de cette concordance est un problème fondamental qui peut être formellement réduit au problème de minimisation d'une métrique sur un modèle. Pour la recherche de concordance exacte, aucune limite inférieure rigoureusement étudiée est connue pour le pire des cas au niveau de la complexité algorithmique dans une dimension générale ; ceci est encore un problème ouvert. En raison de la connexion entre l'appariement de formes et les métriques, une limite inférieure pour la

concordance exacte est aussi une limite inférieure pour tous les autres problèmes d'appariement de formes basés sur une métrique.

1.4 Appariement des nuages de points

L'analyse de modèle du nuage de points joue un rôle important dans la reconnaissance de formes. Il est souvent avantageux de représenter l'ensemble des caractéristiques d'une forme par leurs coordonnées qui peuvent être utilisées pour identifier le modèle dans une scène ou dans une base de données.

L'utilisation d'appariement de formes de point (PPM) est plus efficace par rapport aux techniques de traitement de tableaux (Template matching) dans des situations où la taille de l'objet est grande et le nombre de points caractéristiques est faible. La vitesse des techniques basées sur un modèle de points dans telles situations, leurs flexibilités et parfois leurs invariants à certaines transformations ont fait la solution optimale dans de nombreuses applications.

L'importance du PPM est indiquée par la qualité des recherches récentes dans ce domaine. Il y a beaucoup de techniques utilisant des méthodologies très différentes, chacune avec des capacités différentes selon les applications qui ont inspiré leur développement. Cette composante de l'état d'art se concentre sur les techniques qui ont une approche traditionnelle de reconnaissance de formes basée sur la notion du PPM.

En effet, les techniques de reconnaissance des formes ponctuelles sont utiles pour les tâches de reconnaissance d'objets quand un objet est représenté comme un ensemble de points. Ces tâches peuvent être rencontrées dans des domaines divers tels que la vision par ordinateur, la vision robotique et industrielle, la biométrie, la biologie et l'astronomie [Arzoumanian et al., 2005; Brown, 1992; Edgar, 2004; Huttenlocher and Ullman, 1990; Jain et al., 2007, 2004; Matula et al., 2006; Mount et al., 1999; Paumard, 1997] pour ne citer que quelques-uns.

Baumbach et al. [Baumbach and Ortmann, 1999] décrivent la reconnaissance d'objet de type signal en tant qu'appariement de formes à base de nuage de points. Cette approche traite le signal, d'image par exemple, comme un problème d'appariement de deux ensembles de points en fonction de leurs coordonnées. Mo et al. [Mo et al., 2009] proposent une méthode basée sur un modèle de point, pour des applications en temps réel (suivi d'objet). Matula et al.

[Matula et al., 2006] ont développé une méthode basée sur l'appariement de formes basée sur un modèle de points pour l'alignement des cellules vivantes.

De manière générale, un bon algorithme de reconnaissance de formes à base d'un modèle du nuage de points, devrait être en mesure de trouver les correspondances entre deux ensembles de points, de rejeter les valeurs aberrantes et éventuellement de déterminer une transformation qui permet de cartographier chaque point du premier ensemble avec un autre point du deuxième ensemble s'ils sont effectivement homologues.

1.4.1 Problème

Un modèle de point (motif de points) dans R^2 est un ensemble fini non vide de points sur un plan bidimensionnel. Dans le modèle P , chaque point P_i est décrit par les coordonnées (x_i, y_i) . Le nombre de points dans P est donné par $|P|$ (cardinalité du nuage de points). Dans certaines applications, des caractéristiques supplémentaires du point ou de ses voisinages sont ajoutées à la description sous forme d'une étiquette (descripteur f). Dans ce cas, nous avons un modèle de point étiqueté, défini comme suit : $P = \{p_i\} = \{x_i, y_i, f_i\}$ $i = 1 \dots n$ avec $n = |P|$.

PPM est généralement formulée comme un problème de reconnaissance de formes géométriques d'un ensemble de points, dans lequel nous cherchons à identifier les correspondances d'un modèle dans un échantillon défini. Formellement, il est défini comme:

Étant donné deux ensembles de point $P = \{p_i \in R^d\}_{i=1 \dots n}$ (the model set) et $Q = \{q_j \in R^d\}_{j=1 \dots m}$ (the data set) dans l'espace dD . On cherche à déterminer s'il y a une transformation, parmi un groupe de transformations spécifiques, qui dresse (superpose) exactement ou d'une manière satisfaisante, selon un seuil prédéfini, le deuxième ensemble de points sur le premier ensemble de points, pour définir les relations de correspondances entre ces deux ensembles de points.

La recherche de correspondances (correspondance exacte un-à-un ou une correspondance approximative) et de transformation (T) entre deux ensembles de points, est un problème couplé. La résolution de l'un des deux sans informations de l'autre, est assez difficile. Une fois la résolution complète ou partielle de l'une d'entre eux est obtenue, l'autre devient beaucoup plus facile à trouver (transformation à partir de la correspondance ou inversement). Cependant, dans la plupart des applications, nous n'avons aucune connaissance préalable des deux (transformation/correspondance).

Il y a beaucoup de facteurs qui font du PPM un problème compliqué. Les principaux facteurs qui influencent à la fois la précision et l'efficacité du PPM, sont la qualité de données et la nature de la transformation. La qualité des données peut être sujette à la distorsion, à l'incomplétude et aussi au bruit provenant des différents systèmes de mesure. Les types de transformation peuvent aller d'une transformation rigide ou affine simple, à une projective complexe ou élastique avec des applications spatiales qui sont 2D / 2D, 2D / 3D, 3D / 2D ou 3D / 3D [Liu and Rodrigues, 2002; Maintz and Viergever, 1998; Rodrigues and Liu, 1999].

La transformation rigide préserve toutes les intra-distances et les intra-angles d'un ensemble de points dans la transformation. Elle peut être représentée par une matrice de transformation globale. C'est une classe importante de transformations affines qui préservent la colinéarité, les ratios de distances et le parallélisme [Maintz and Viergever, 1998]. Elle est généralement utilisée pour représenter une combinaison linéaire des opérations cartésiennes : translation, rotation, mise à l'échelle, et / ou de cisaillement.

1.4.2 Techniques

Dans la famille des méthodes dites PPM, les techniques sont présentées dans trois catégories principales : le regroupement (Clustering), les algorithmes basés sur les distances entre les points (interpoint distance algorithms) et les méthodes de relaxation. Les techniques qui ne se trouvent pas dans ces catégories seront décrites séparément.

Pour simplifier la description, un modèle sera appelé une scène Q et l'autre un prototype P. En pratique, Q décrit les données acquises et P décrit l'un des objets connus. Il peut être nécessaire de classer Q comme l'un des prototypes, ou de détecter une occurrence de P à l'intérieur de Q ou inversement.

1.4.2.1 Clustering

Ces techniques [Stockman, 1987; Stockman et al., 1982] supposent que les deux modèles de point se différencient par une ou plusieurs transformations. Ces paramètres de transformation sont calculés pour toutes les combinaisons de paires de point des deux modèles. Les groupes les plus forts dans l'espace de paramètres représentent alors les paramètres de transformation les plus probables pour la meilleure correspondance.

Kahl et al [Kahl et al., 1978] proposent une technique qui définit un espace de paramètres quantifié T, pour toutes les translations possibles entre P et Q. La différence vectorielle est

calculée entre chaque combinaison de paires de point de P et Q. Si l'amplitude de cette différence est inférieure à un seuil ε pour (p_i, q_j) et (p_h, q_k) , alors le mérite de la translation dans T qui mappe p_i à q_k , est incrémenté.

Goshtasby et al [Goshtasby and Stockman, 1985] décrivent une procédure d'appariement qui détermine les paramètres de rotation, de mise à l'échelle et de translation (R_t, S_c, T_r) qui correspondent au plus grand nombre de points dans les deux modèles. Une valeur de mérite est attribuée aux paramètres (R_t, S_c, T_r) de la transformation pour chaque combinaison de paires de points entre P et Q, en fonction du nombre des autres points qui se correspondent à l'intérieur d'un seuil de distance donné D. La meilleure transformation (R_t, S_c, T_r) est ensuite utilisée pour déterminer le nombre de points en jeu au sein de D. La transformation optimale (R_t, S_c, T_r) est calculée en utilisant ces points appariés et en minimisant la somme des carrés des erreurs.

Stockman et al [Stockman et al., 1982] décrivent une technique de reconnaissance d'objet qui utilise un cluster pour l'enregistrement des caractéristiques d'une image à des modèles. Une méthode d'estimation par moindre carrée pour déterminer les paramètres optimaux (R_t, S_c, T_r) qui font correspondre les points des deux modèles, est décrite par Umeyama [Umeyama, 1991].

1.4.2.2 Algorithmes basés sur les distances entre les points

Dans ces techniques, les distances entre les points sont utilisées comme base de comparaison. Lavine et al [Lavine et al., 1983] décrivent trois représentations différentes de la matrice de distance interpoint (IDM), qui sont utilisées pour déterminer une similarité entre deux modèles de points.

Les deux premières représentations sont des vecteurs triés des distances entre points. Le vecteur de distance d'interpoint trié (SIDV) est de calcul coûteux et donne lieu à des problèmes de probabilité géométriques. Le vecteur du plus proche voisin trié (SNNV) est plus efficace en incluant seulement la distance entre un point et son voisin le plus proche. La troisième représentation est l'arbre de recouvrement minimal (the minimal spanning tree) (MST), qui représente le meilleur moyen de relier les nœuds dans une représentation de graphe du modèle de point. Les points définissent les nœuds dans le graphe et les distances entre points définissent les arcs du graphe.

Wong et al [Wong and Salay, 1986] ont construit un arbre à partir de la représentation graphique d'un modèle de points. Cet arbre représente chaque appariement possible de p_i vers q_i de la racine à la feuille. Ils ont défini une fonction de coût pour un couplage de taille n comme la somme des différences de toutes les distances entre les points correspondants P à Q . Une correspondance est trouvée par la recherche de l'arbre pour l'appariement du coût minimal avec le maximum de points qui correspondent dans une tolérance donnée.

1.4.2.3 Relaxation

La technique de relaxation a été utilisée pour enregistrer des images sous la contrainte de translation. Elle est appelée méthode de relaxation, car elle implique une procédure itérative grossière-à-fine (coarse-to-fine) sur la base de paire de données locales pendant la correspondance. Chaque paire de correspondance assumée dans les deux ensembles, définit un déplacement et donne un score de mérite par rapport à quel point les autres paires correspondraient sous ce déplacement. S'il existe un déplacement pour lequel les deux ensembles de points correspondent correctement, donc la paire avec un haut score de mérite indique la concordance la plus raisonnable entre ces deux ensembles de points. Ce score peut alors être recalculé et donne des poids aux autres paires de points en fonction de leurs propres scores. Cette procédure est itérée jusqu'à la convergence selon les critères souhaités.

Ranade et al [Ranade and Rosenfeld, 1980] proposent la relaxation pour l'invariance à la translation, la technique est plus tolérante aux distorsions globales. Chaque appariement des points (p_i, q_j) assigne un score de mérite par rapport à l'appariement des autres points (p_h, q_h) . Les itérations ultérieures utilisent ce score de mérite pour (p_h, q_k) et la distance entre p_h et q_k pour assigner un nouveau score de mérite à la cartographie (p_i, q_j) . L'algorithme converge quand le score de mérite pour l'appariement entre (p_i, q_j) sous la transformation actuel entre P et Q , reste élevé, tandis que ceux des autres appariements deviennent faibles.

Ogawa [Ogawa, 1984] décrit une technique de relaxation (fuzzy relaxation) pour des modèles de points étiquetés. Un ensemble de paires des primitives (p_i, q_j) , est trié pour le modèle P . Les primitives pour la scène Q , sont triées en référence aux primitives de P . Les itérations du processus de relaxation transforment géométriquement les primitives compatibles entre P et Q afin de réduire au minimum le décalage entre les autres paires de primitives.

1.4.2.4 Autres techniques

Hong et al [Hong and Tan, 1988] ont défini une forme canonique pour les modèles de points en vertu de transformation affine et ont utilisé cette représentation comme base de comparaison.

Griffin et al [Griffin and Alexopoulos, 1989] décrivent une technique qui utilise les plus petits cercles de P et Q pour déterminer un facteur d'échelle. Les paramètres des translations t_x et t_y , sont déterminés par la comparaison des centres de gravité des modèles normalisés. Un graphe biparti des mappages possibles est réalisé et l'appariement final de P sur Q est déterminé à partir de cardinalité maximale de ce graphe.

Ogawa [Ogawa, 1986] décrit une technique de correspondance de modèle de points étiquetés. Un sous-ensemble des points dans le modèle P, est choisi en fonction de la similarité entre des points étiquetés dans P et Q. Ce sous-ensemble de points est représenté par un ensemble de triangles en utilisant la triangulation de Delaunay. Pour chacun de ces triangles, il y a des candidats pour un appariement dans la triangulation de Delaunay de Q. Les triangles pariés sont utilisés pour déterminer la transformation affine de P sur Q, et pour fournir des candidats d'appariement à partir de Q pour chaque point de P. Le plus grand ensemble d'appariements des points mutuellement cohérents, est déterminé de la plus grande clique maximale dans le graphe de cohérence des candidats des appariements des points.

1.4.2.4.1 Graphes

Les graphes avec recherche arborescente peuvent donner une représentation naturelle de modèles de points associés aux objets rigides et leurs attributs [Carcassoni and Hancock, 2000; Chui and Rangarajan, 2000; Gaede and Günther, 1998; Grimson and Lozano-Perez, 1987; Lavine et al., 1983; Preparata and Shamos, 2012; Samet, 1990; Sproull, 1991]. L'appariement de formes des points peut être réduit maintenant à la correspondance de ces graphes. Les approches des graphes se définissent généralement par des intra-relations invariantes géométriquement et par d'autres attributs structurels. Toutefois, cette définition du graphe peut être très difficile. Ce schéma pourrait être impossible dans les cas de correspondances pratiques qui utilisent uniquement des informations de localisation de points, en raison de données incomplètes.

1.4.2.4.2 Alignement géométrique et hachage géométrique

L'alignement géométrique [Huttenlocher and Ullman, 1990] et le hachage géométrique [Lamdan and Wolfson, 1988; Wolfson and Rigoutsos, 1997] ont été utilisés dans le contexte de reconnaissance d'objet à base de modèle dans la vision par ordinateur. Ces techniques se concentrent particulièrement sur la correspondance entre les caractéristiques géométriques, qui sont souvent un ensemble de point d'un modèle pré-connu dans une scène. Les méthodes d'alignement géométriques sont généralement exhaustives. En pratique, seulement les caractéristiques les plus discriminantes dans le modèle et les ensembles de données sont utilisées pendant l'alignement. Le hachage géométrique à base d'index est établi sur la base de l'alignement, mais il est plus avantageux en termes de son efficacité et de sa robustesse pour traiter l'occlusion.

1.4.2.4.1 Approches itératives grossière-à-fine

Les approches itératives grossière-à-fine commencent en établissant un ensemble de correspondances partielles afin d'approcher une solution optimale ou approximative globale via la recherche heuristique. Ces méthodes sont généralement basées sur invariance/contrainte-satisfaction géométrique incorporée dans les transformations. Le problème d'optimisation est exprimé et résolu dans l'algèbre matricielle utilisant la régression par moindres carrés [Arun et al., 1987; Eggert et al., 1997], les métriques de Hausdorff [Chew et al., 1997; Hagedoorn and Velkamp, 1997; Huttenlocher et al., 1993b] ou la méthode heuristique ICP (Iterative Closest Point) [Besl and McKay, 1992; Zhang, 1994]. Dans la procédure itérative une première correspondance est généralement supposée par des hypothèses telles que la similitude de pose, le mouvement pré-connu ou un petit mouvement. Pour éviter de telles suppositions, les mesures de similarité entre le modèle et les données, peuvent être utilisées comme les coins [Jung and Lacroix, 2001] ou la table des candidats [Holstein and Li, 2002]. L'hypothèse initiale pourrait être confirmée si la transformation basée sur cette dernière, est capable de mapper les données du modèle à l'ensemble de données et de trouver les correspondances qui satisfont à la fois la taille de l'appariement et les critères de qualité. Une fois un appariement local fiable a été déterminé pour le groupe local, des nouveaux groupes sont rajoutés en propageant la transformation ainsi évaluée. Cette procédure du partiel au global procède itérativement, optimise continuellement et agrandit l'appariement local, à la fin, une transformation globale, est obtenue. Les approches itératives sont généralement implicites, efficaces et pratiques dans la plupart des applications. Elles ont

été largement utilisées dans la littérature avec de nombreuses variantes pour différentes situations. Cependant, les approches implicites itératives sont souvent sensibles à l'initialisation, ce qui peut mettre en péril la convergence vers un optimum global.

1.5 PPM appliqué aux objets non rigide

L'analyse du mouvement des structures non-rigides (articulées, élastiques ou déformables) basée sur les points caractéristiques a eu des intérêts croissants au cours de la dernière décennie, motivé par des applications dans de nombreux domaines [Aggarwal et al., 1994; Chui and Rangarajan, 2003; Pitiot et al., 2003; Rangarajan et al., 1997; Sclaroff and Pentland, 1995]. En comparaison avec le mouvement rigide et affine, le mouvement non rigide exprime une relation mathématique plus complexe. Pour rendre le problème traitable, la plupart des méthodes dépendent de certaines restrictions globales / locales spécifiques et relatives aux mouvements, aux objets et aux propriétés du domaine d'application. Au lieu de chercher la correspondance exacte / identification un-à-un, les correspondances sont produites comme le résultat d'enregistrement non-rigide.

La non-rigidité peut résulter des facteurs divers qui incluent la distorsion non linéaire de capteurs, la déformation d'objet, l'articulation ou autres facteurs spécifiques à un domaine. La formulation approximative du mouvement non-rigide devient un sujet important dans ce cas. Le calcul des transformations locales est généralement nécessaire dans ces situations. Une formulation globale peut être approchée par approximation fragmentaire (piecewise), modèles élastiques / déformables [Aggarwal et al., 1994; Bardinet et al., 1998; Polana and Nelson, 1997; Sclaroff and Pentland, 1995], ou graphe pondéré [Cross and Hancock, 1998; Rangarajan et al., 2001].

Au lieu de calculer une transformation globale uniforme, l'approximation fragmentaire effectue le calcul. L'évaluation et l'optimisation pour chaque fragment se propagent successivement aux autres voisinages. Cette approche est appropriée pour les données segmentées par morceaux. En fait, la méthode par fragments est probablement l'approche la plus populaire dans des applications non-rigides, puisque sa complexité informatique est relativement basse et sa mise en œuvre est relativement facile [Holstein and Li, 2002; Pitiot et al., 2003; Rangarajan et al., 2001].

Un schéma par morceaux permet un bon contrôle de la non-rigidité locale, mais pourrait être moins efficace s'il ignore les informations globales. Les transformations des morceaux locales pourraient être intégrées aux méthodes globales [Feldmar and Ayache, 1996]. Elles sont généralement applicables pour les PPM en cas de morceaux non-rigides. On suppose que les entités ponctuelles sont suffisantes pour établir un ensemble de transformations locales et leur non-rigidité est continue et régulière dans les données comme une chaîne de morceaux connectés. Lorsque l'occlusion se produit, la perte de connectivité pourrait conduire à l'échec. En outre, lorsque les morceaux à apparier sont basés sur une segmentation automatique, la qualité de la correspondance peut être limitée par la précision de cette segmentation.

1.6 Complexité

De manière générale, la tâche qui consiste à établir des appariements est difficile. Prenant, par exemple, le cas où les données sont décrites par des caractéristiques du même type et soient F l'ensemble de caractéristiques objet et SF l'ensemble de caractéristiques données.

Supposant que F et SF ont le même nombre d'éléments et n , le nombre d'appariement possibles est :

$$n!$$

Or F et SF ont rarement la même taille. Si m est la taille de SF et n la taille de F avec $n \geq m$, le nombre d'appariement possible est alors :

$$C_n^m m!$$

Lorsque les caractéristiques d'un objet ne sont pas toutes présentes dans les données, dans ce cas, on espère mettre en correspondance un sous-ensemble de F avec un sous-ensemble de SF . Le nombre d'appariements possibles devient alors beaucoup plus grand :

$$\sum_{i=1}^m C_m^i C_n^i i!$$

La reconnaissance devient assez fiable lorsque le nombre d'appariements est grand. Le problème à résoudre est de trouver le plus grand nombre d'appariements entre F et SF . Un tel appariement maximal ne peut être trouvé facilement. Même si on tient compte des caractéristiques de F et SF , la complexité de ce problème est combinatoire. En pratique, on se contentera d'une solution sous-optimale. Ceci dit, la majorité des méthodes introduit des astuces afin d'éviter de vérifier toutes les combinaisons possibles entre la forme F et la sous-forme SF en réduisant ainsi les temps d'exécutions.

1.7 Conclusion

Au cours de ce chapitre, nous avons présenté différentes approches relatives à la reconnaissance de formes, en particulier celles traitant les données sous la forme de nuage de points.

En effet, les différentes approches admettent des complexités algorithmiques plutôt élevées qui peuvent varier d'une complexité polynomiale à une complexité exponentielle avec des taux de reconnaissance acceptable dans l'ensemble.

Dans leur majorité, ces méthodes sont dédiées à un type de problème plutôt qu'un autre. Certes, des tentatives de généralisations de certaines méthodes ont été tentées mais avec des résultats plutôt timides.

Il est donc clair que vu l'intérêt de cette discipline et de ses champs d'application, il convient de dire que toute nouvelle approche permettant de réduire la complexité et d'augmenter le taux de reconnaissance serait d'une très grande utilité.

De manière générale, on peut confirmer que l'une des raisons de la complexité de ces méthodes réside, entre autre, dans le caractère discret des entités à traiter qui conduisent à une explosion combinatoire (nombre de possibilités à tester). Ce constat constitue la base de notre contribution, qui sera décrit dans le chapitre suivant, à savoir élaborer un moyen transformant le caractère discret de ces entités en une représentation continue.

Chapitre 2 Algorithme de reconnaissance de formes discrètes par passage au continu

Ce chapitre constitue l'essentiel de notre travail. Il présente notre approche qui se base essentiellement sur le passage du discret, qui est la cause des risques d'explosions combinatoires comme évoqué dans le premier chapitre, vers le continu. De ce fait, la recherche directe de la transformation de superposition des entités à traiter, sans connaissance au préalable des correspondances de points, devient possible. Pour toutes les méthodes qui existent, le temps de réponse est une fonction croissante des tailles des objets à traiter. Nous allons voir que dans notre cas, cette règle fait exception. Une explication de ce comportement sera donnée par la suite.

Pour illustrer le comportement et la complexité de notre algorithme, nous allons effectuer des tests en 2D et 3D.

2.1 Introduction

Comme cité dans la partie précédente, l'analyse du modèle du nuage des points joue un rôle important dans la reconnaissance de formes. Il est souvent avantageux de représenter l'ensemble des caractéristiques locales d'une forme par ses coordonnées utilisées pour identifier le modèle dans une scène ou de façon individuelle dans une base de données.

L'utilisation d'appariement de formes de point (PPM) est supposée être plus efficace que les techniques de traitement des tableaux (Template matching) dans des situations où la taille de l'objet est grande alors que le nombre de points caractéristiques est faible. La vitesse d'exécution des techniques basées sur un modèle de nuage de points dans telles situations, leur flexibilité et parfois leurs invariances ont certaines transformations rendant ces techniques comme une solution optimale dans de nombreuses applications.

La PPM, comme indiquée par de nombreuses recherches récentes, offrent beaucoup de techniques utilisant des méthodologies très différentes ; chacune avec des capacités différentes selon les applications qui ont inspiré leur développement.

Dans notre travail, la PPM est utilisée pour formuler un problème de reconnaissance de formes géométriques d'un ensemble de point (nuage de points) dans lequel nous cherchons à identifier une (des) correspondance (s) d'un modèle établi dans un échantillon défini.

2.2 Performance de l'algorithme proposé

2.2.1 Définition du problème

Étant donné deux ensembles de points $F = \{f_i \in \mathbb{R}^d\}_{i=1}^n$ (the model set) et $SF = \{Sf_j \in \mathbb{R}^d\}_{j=1}^{m \leq n}$ (the data set) dans l'espace multidimensionnel ND (figures 2.1).

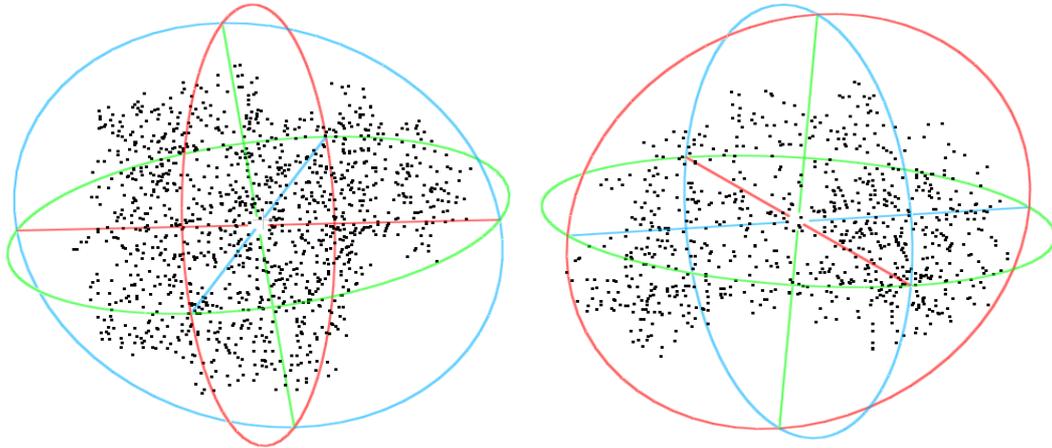


Figure 2.1 La forme F en 3D (à gauche) et la sous-forme SF (à droite)

Le problème consiste alors à savoir si SF est inclus dans F. Traditionnellement, on commence par trouver pour chaque point de SF son correspondant dans F et éventuellement rechercher une transformation qui superposerait SF sur F.

Dans le cas où on serait intéressé par la recherche de la transformation T, une recherche directe de cette dernière, sans aucune connaissance au préalable des correspondances des points de SF par rapport à ceux de F, risque d'être très coûteuse en temps de réponse vu le nombre important de possibilité à tester (explosion combinatoire).

Avant de décrire le principe de notre algorithme, nous avons jugé intéressant de décrire les différentes formes du problème de la PPM liées à la recherche de l'inclusion de SF dans F, on utilise une transformation T :

- EPPM (Exact Point Pattern Matching):
EPPM problème exige que $T(SF) = F$. Autrement dit, chercher une transformation T en vertu de laquelle les deux formes sont supposées être identiques où l'incertitude est égale zéro.
- APPM (Approximate Point Pattern Matching):
Un exemple qui provient souvent du monde réel dont les données ne sont jamais mesurées avec certitude et précision exacte. Dans ce cas, la similarité entre les deux

nuages de points est qualifiée selon un seuil ε prédéfini. Ainsi, l'évaluation est en termes d'une distance D .

Le problème APPM est de chercher s'il existe une telle transformation T en vertu de laquelle $D(T(SF), F) \leq \varepsilon$.

- LCP (Largest Common Point Set) :

LCP problème cherche le plus grand sous-ensemble (la plus grande sous structure commune) de points $L \in SF$ et le plus grand sous-ensemble de points $M \in F$, de telle sorte que $T(L)$ est similaire à M .

- ALCP (Approximate Largest Common Point Set) :

L'ALCP est une variante du APPM et du LCP, est défini par:

$$D(T(L), M) \leq \varepsilon$$

Il est à rappeler que l'existence d'une telle transformation T confirmerait de toutes évidences l'inclusion de SF dans F . Ce dernier point, à savoir la recherche de T pour confirmer cette inclusion, apparait à première vue très coûteuse, constitue, à quelques modifications près, le principe de notre algorithme.

L'origine de la difficulté (explosion combinatoire) provient clairement de la nature discrète du nuage de points à traiter. La question qui se pose serait donc de se dire : est-il possible d'opérer un passage au continu de l'une de ces entités (F ou SF) évitant, de ce fait, les risques d'explosions combinatoires, tout en conservant les informations des formes à traiter ?

Si c'était le cas, avec une représentation continue de F par interpolation (SF serait conservée dans sa représentation discrète), le problème de savoir si le nuage SF est inclus dans le nuage F devient la recherche, non pas de T , mais d'une transformation T' qui ramènerait SF , prise sous sa forme discrète, sur une représentation continue de F , ainsi l'existence de T' pourrait induire l'existence probable de T , et par conséquent, confirmera l'inclusion de SF dans F .

En résumé, notre travail consistera donc à ne pas rechercher T mais plutôt rechercher une transformation T' qui assurerait la superposition de SF sur la représentation continue de F . Il est à noter que l'existence d'une telle transformation T' est nécessaire mais pas suffisante pour confirmer l'inclusion (totale ou partielle) de SF dans F . En effet, la transformation T' doit en plus assurer que pour tout point $P_{sf_i} \in SF$ il existe un point $P_{f_j} \in F$ telle que $T'(P_{sf_i}) = P_{f_j}$ pour confirmer que $T'=T$.

2.2.2 Passage au continu de la forme F par interpolation

Les modèles des nuages seront considérés dans un espace 3D et une généralisation à un espace ND sera proposée par la suite. Il serait tentant de faire la transition de la représentation discrète vers une représentation continue pour inclure les points de F dans une surface continue par une fonction d'interpolation surfacique. Une telle tentative risque d'introduire beaucoup d'informations supplémentaires sur les fragments de surface entre des points réels de F.

Donc l'utilisation d'une surface d'interpolation, semble inutilement complexe et ne permet pas de généraliser l'idée à d'autres dimensions. Nous préférons alors d'assurer le passage du discret vers le continu d'une manière différente tout en gardant le même principe.

Les points de F sont d'abord projetés dans le plan O_{xy} et dans le plan O_{xz} . Ces deux projections assurent la conservation de l'information. La projection sur le plan O_{yz} ne sera pas nécessaire.

Chaque nuage de points 2D ainsi obtenu, est interpolé ; on utilise une interpolation polynomiale.

Posons les polynômes d'interpolation P_{xy} et P_{xz} respectivement dans les plans O_{xy} et O_{xz} . Ainsi, pour chaque point $f_i = \{x_i, y_i, z_i\}$ appartient à F, nous avons la représentation suivante qu'on note R :

$$\begin{cases} P_{xy}(x_i) = y_i \\ P_{xz}(x_i) = z_i \end{cases} \text{ Appelons cette représentation : R}$$

Il existe différentes méthodes d'interpolation pour représenter R. Afin que le degré du polynôme ne dépend pas de la taille du nuage de points F, nous avons choisi pour notre part l'interpolation dite "cubique spline" qui est une succession de polynôme de degré trois (interpolation par morceaux) et qui assure aussi la continuité et la dérivabilité sur l'ensemble de l'intervalle d'interpolation. La continuité et la dérivabilité de ce type d'interpolation nous assurent une large gamme d'outils d'optimisation comme il sera décrit plus tard.

2.2.3 La recherche de T'

Il est à remarquer que la transformation T' recherchée concernerait aussi bien le plan O_{xy} que le plan O_{xz} , il ne s'agit en aucun cas de rechercher un T' relatif à chaque plan.

La transformation T' aurait donc pour but de ramener la forme SF sur l'interpolation cubique de la forme F selon les deux plans (O_{xy}, O_{xz}) .

La transformation T' recherchée, est exprimée en coordonnées homogènes. Cela implique que les points projetés de SF sur la représentation continue R de F seraient exprimés à l'aide de ces deux polynômes.

Si on considère $(x_{SF_j}, y_{SF_j}, z_{SF_j})$ les coordonnées d'un point $j \in SF$ et $(x'_{SF_j}, y'_{SF_j}, z'_{SF_j})$ sa transformation par T' , les points de SF ainsi transformés doivent aussi vérifier R, à savoir :

$$R \begin{cases} P_{xy}(x'_{SF_j}) = y'_{SF_j} \\ P_{xz}(x'_{SF_j}) = z'_{SF_j} \end{cases}$$

Les paramètres de la transformation T' que nous cherchons sont :

- Trois translations : t_x, t_y, t_z
 - ❖ t_x : Translation suivant l'axe O_x .
 - ❖ t_y : Translation suivant l'axe O_y .
 - ❖ t_z : Translation suivant l'axe O_z .
- Trois rotations : $\theta_x, \theta_y, \theta_z$
 - ❖ θ_x : Rotation suivant l'axe O_x .
 - ❖ θ_y : Rotation suivant l'axe O_y .
 - ❖ θ_z : Rotation suivant l'axe O_z .
- Trois facteurs d'échelle : $\lambda_x, \lambda_y, \lambda_z$
 - ❖ λ_x : Facteur d'échelle suivant l'axe O_x .
 - ❖ λ_y : Facteur d'échelle suivant l'axe O_y .
 - ❖ λ_z : Facteur d'échelle suivant l'axe O_z .

Si on considère que SF est inclus dans F, alors il existe certainement plusieurs chemins pour aligner (ramener) SF sur F (ou suffisamment près de F). Ces chemins dépendent de l'ordre dans lequel les opérations sont effectuées (translation, rotation et mise en échelle). Notre but n'étant pas de chercher une transformation optimale, et par conséquent nous avons choisi un ordre arbitraire.

Notons que l'utilisation des paramètres $(\lambda_x, \lambda_y, \lambda_z)$ correspond à un changement d'échelle selon les différents axes. L'utilisation de ce type de transformation est très utile dans le cas de reconnaissance de formes élastiques ou contenant des effets de zoom. Cet aspect ne sera pas traité dans ce travail. Ainsi, pour simplifier la compréhension et la représentation, on considère le cas de mise en échelle uniforme, et par conséquent, on suppose que $\lambda_x = \lambda_y = \lambda_z = \lambda$.

La notation de R peut donc s'écrire comme :

$$R \begin{cases} P_{xy}(x'_{SF_j}) - y'_{SF_j} = 0 \\ P_{xz}(x'_{SF_j}) - z'_{SF_j} = 0 \end{cases}$$

Ou encore

$$(P_{xy}(x'_{SF_j}) - y'_{SF_j})^2 + (P_{xz}(x'_{SF_j}) - z'_{SF_j})^2 = 0$$

Appliquée à l'ensemble des points de SF, on obtient la formule suivante :

$$\sum_{j=1}^m \sqrt{((P_{xy}(x'_{SF_j}) - y'_{SF_j})^2 + (P_{xz}(x'_{SF_j}) - z'_{SF_j})^2)} = 0$$

Appelons QT l'expression suivante:

$$QT_{(\lambda, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z)} \sum_{j=1}^m \sqrt{((P_{xy}(x'_{SF_j}) - y'_{SF_j})^2 + (P_{xz}(x'_{SF_j}) - z'_{SF_j})^2)} = 0$$

A base de cette définition, nous cherchons maintenant les paramètres de la transformation T' qui minimise la fonction QT .

La fonction du coût QT ainsi obtenue, est une équation non-linéaire dont les inconnus sont $(\lambda, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$. QT étant une fonction continue et dérivable, nous disposons donc d'une grande gamme de méthodes numériques qui sont dédiées à résoudre ce problème.

2.2.4 Les propositions de correspondances : isomorphisme et transformation T

Rien ne prouve que la transformation T', sensée ramener les points de SF sur la représentation R va ramener ces derniers sur ceux de F. En fait, la description de la forme par un polynôme introduit des informations supplémentaires : les points de la représentation continue R n'appartiennent pas à F et peuvent en principe s'apparier avec les points de SF.

Cependant, cette possibilité est réduite en fonction de la taille et de la complexité de SF et de F.

À titre d'exemple, nous pouvons citer quelques cas évidents où la représentation R a un risque de ne pas répondre à nos attentes car il y aurait une infinité de solutions possibles:

- Recherche de la position d'un arc de cercle dans un autre arc (de même rayon) plus long.
- Recherche d'un segment de droite dans une droite.

Fort heureusement, les formes F et SF n'ont pas toujours des formes régulières. Il est à remarquer que les complexités géométriques et les tailles de SF et F contribuent à augmenter les contraintes et donc fournissent des solutions de plus en plus proches de ce qui est recherché. Cette même complexité de géométries et de tailles, doit en principe impacter le temps de réponse ; mais vu que les entités sont prises dans leur totalité pour rechercher T', comme on le remarquera plus tard l'augmentation des tailles de SF et F peuvent dans certains cas réduire le temps de réponse, ce qui est contraire à ce qu'on connaît avec les autres méthodes où justement le temps de réponse dépend de la complexité des entités traitées.

2.2.5 Recherche de T

Après l'étape d'ajustement des points de SF sur la représentation continue de F (défini grâce à T'), nous associons à chaque point de SF son isomorphe qui est son voisin le plus proche dans F selon un type de distance et un seuil prédéterminé.

Posant pour un point $P_j \in SF$: $T(x_{SF_j}, y_{SF_j}, z_{SF_j}) = (x''_{SF_j}, y''_{SF_j}, z''_{SF_j})$

Si $P_i(x_{F_i}, y_{F_i}, z_{F_i})$ est un point de F qui est isomorphe du point P_j de SF défini grâce à T' ; si $Distance(P_j, P_i)$ est la distance utilisée, et si la RMS est utilisée pour mesurer la précision globale de l'inclusion de SF dans F, on a donc:

$$RMS(\lambda', \theta'_x, \theta'_y, \theta'_z, t'_x, t'_y, t'_z) = \sum Distance(P_i, P_j)$$

La RMS est un indice standard pour mesurer la qualité de la superposition. Si la distance utilisée est la distance euclidienne, alors RMS devient :

$$RMS = \frac{1}{m} \sqrt{\sum_{j=1}^m (x''_{SF_j} - x_{F_i})^2 + (y''_{SF_j} - y_{F_i})^2 + (z''_{SF_j} - z_{F_i})^2}$$

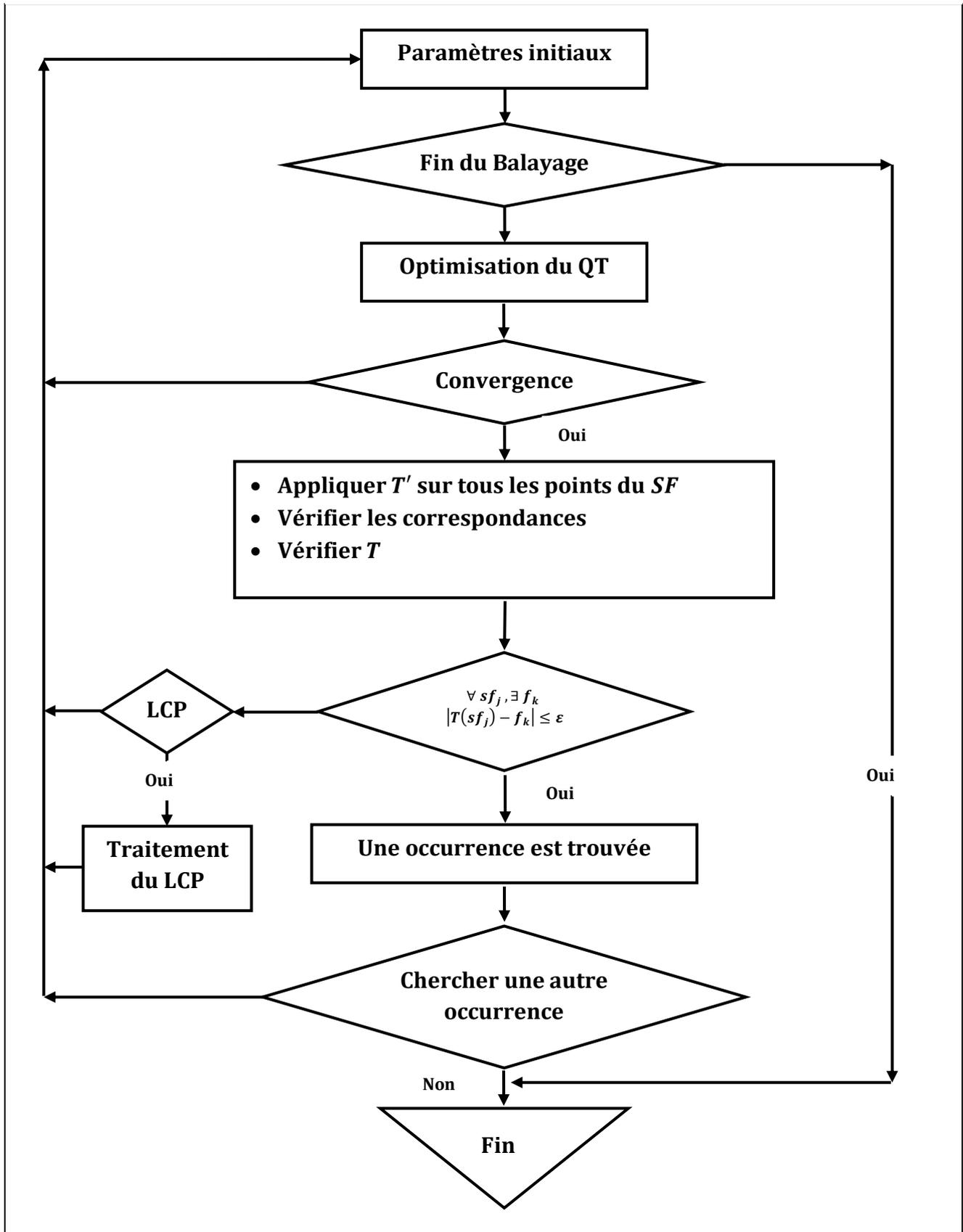
Une fois la forme canonique de la RMS est définie, nous allons procéder à la détermination de ses paramètres $(\lambda', \theta'_x, \theta'_y, \theta'_z, t'_x, t'_z)$ minimisant cette dernière. Les valeurs $(\lambda, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$ définies au cours du calcul de T' seraient utilisées comme état initial pour définir les paramètres $(\lambda', \theta'_x, \theta'_y, \theta'_z, t'_x, t'_y, t'_z)$.

La RMS est aussi formulée sous forme d'une fonction non-linéaire. Généralement, la transformation T' ainsi trouvée, fournit directement les paramètres $(\lambda', \theta'_x, \theta'_y, \theta'_z, t'_x, t'_y, t'_z)$, et donc la solution de la RMS est déjà trouvée. Néanmoins, certains affinages peuvent être opérés selon un seuil de RMS prédéfini.

Après détermination des paramètres de la RMS $(\lambda', \theta'_x, \theta'_y, \theta'_z, t'_x, t'_y, t'_z)$, seuls seront retenus les points de SF dont la distance à leurs isomorphes dans F , est inférieur à un seuil ϵ fixé au préalable.

- Si toutes les distances entre les points de SF à leurs isomorphes dans F , ne dépassent pas ϵ , alors SF est déclarée incluse dans F .
- Si SF est déclarée non-incluse dans F , il pourrait être intéressant dans certaines applications de traiter la plus grande sous-structure commune (LCP) entre les deux nuages de points. En effet, les couples isomorphes qui ne respectent pas le seuil ϵ prédéfini seront éliminés et une relance d'un affinage de la RMS aura lieu pour une meilleure superposition des points restants de SF sur les points de F .

Organigramme 2—1 Organigramme de l'algorithme



2.3 Outils

2.3.1 Interpolation

De manière générale, partant d'un nuage de point, l'interpolation consiste à chercher des fonctions $P(x)$ (polynômes, polynômes par morceaux, polynômes trigonométriques) passant par (ou proche) des points donnés.

Cette situation intervient dans plusieurs cas, parmi lesquels on peut citer:

La fonction $P(x)$ n'est pas connue, on ne connaît que les valeurs de certains points x_i . Les données peuvent être, par exemple, des mesures expérimentales. Le but est alors de trouver une représentation synthétique (analytique) de ces données expérimentales.

Etant donné $n + 1$ couples $(x_i, y_i)_{i=0, \dots, n}$, le but de l'interpolation est alors de trouver un polynôme $P(x)$ qui appartient à une certaine classe et qui prend dans les nœuds d'interpolation x_i les valeurs y_i . Autrement dit ; $P(x_i) = y_i \forall i \in [0, n]$. On dit que $P(x)$ interpole y_i aux nœuds x_i . Contrairement à l'interpolation, l'approximation d'une fonction ne demande pas que la courbe recherchée passe par les points (x_i, y_i) , mais plutôt un critère d'approximation qui doit être satisfait, comme par exemple le critère de minimax, le critère des moindres carrés, etc.

Pour encadrer l'intervalle de la recherche, nous nous intéressons spécialement aux méthodes d'interpolation dont l'ordre du polynôme ne dépend pas du nombre des points à interpoler. En d'autre terme, la méthode ne doit pas être influencée par le phénomène du Runge Kota qui intervient lorsqu'on tente d'approcher une fonction à l'aide de polynômes d'interpolation. En effet, lorsque le degré du polynôme interpolateur augmente, les oscillations de ce dernier au bord du domaine s'intensifient. Ce comportement induit, par exemple, des valeurs trop élevées des dérivées premières et secondes.

En raison de ces deux facteurs mentionnés ci-dessus et en raison d'autres facteurs qui seraient décrits plus tard, les méthodes d'interpolation de Lagrange et de Newton sont exclues dans notre cas. Parmi les méthodes d'interpolations échappant au phénomène du Runge Kota, on peut citer: les courbes d'Hermite, les courbes de Bézier, les courbes B-Splines, les cubiques splines, etc.

Suite à ces constats, il est évident que l'on devra utiliser une méthode d'interpolation dont les degrés seraient limités et indépendants de la taille du nuage de points. Dans ce travail, nous avons opté la méthode dite "cubique spline" dont les propriétés sont :

- $P(x)$ est une fonction polynomiale définie par morceaux qui interpole tous les points donnés, ainsi l'ordre du polynôme ne dépend pas du nombre des points à interpoler.
- $P(x)$ est continue sur l'intervalle $[x_1, x_n]$.
- La dérivée première $P'(x)$ est continue sur l'intervalle $[x_1, x_n]$.
- La dérivée seconde $P''(x)$ est continue sur l'intervalle $[x_1, x_n]$.
- $P(x)$ interpole tous les points de données, et par conséquent, nous pouvons conclure que $P(x_i) = y_i \forall i \in [0, n]$.
- Extrapolable.

La régularité d'une fonction est appréciée à travers ses dérivées. En effet, plus une fonction est différentiable, plus la courbe qui lui est associée est lisse et plus régulière (proche de la réalité, autrement dit même représentation que l'œil humaine).

Le choix de l'interpolation par splines cubiques est donc justifié par le fait que cette méthode représente un compromis très intéressant entre la régularité de la courbe et le degré des polynômes utilisés. Aussi, elle passe par tous les points de données.

2.3.2 Extrapolation

Comme nous l'avons introduit au début de ce chapitre, l'interpolation sert à évaluer la fonction coût QT à l'intérieur de l'intervalle $[0, N]$. Or, la minimisation de QT doit prendre en compte les valeurs de celle-ci en dehors de ce champ, d'où la nécessité de prévoir la possibilité de traiter l'extrapolation. Pour les deux intervalles à droite et à gauche de l'intervalle de l'interpolation dont les deux premiers et les deux derniers points peuvent être utilisés dans le processus de l'extrapolation. D'autres points supplémentaires peuvent être pris aussi en compte dans ce processus conformément à la méthode utilisée. Généralement, les méthodes sont les mêmes que pour l'interpolation.

À noter que le processus de l'extrapolation dont on parle, n'a pas de relation avec la prédiction. D'autre part, la méthode utilisée ne doit pas affecter la convergence de notre approche. Cependant, à travers plusieurs travaux de recherche existents, nous avons constaté que l'extrapolation linéaire peut bien satisfaire nos besoins.

2.3.3 Transformations géométriques

Les coordonnées cartésiennes usuelles permettent de représenter un point (x, y) par une matrice ligne $[x, y]$. En coordonnées homogènes, un point est toujours représenté par une matrice ligne avec une coordonnée supplémentaire. Les coordonnées homogènes reposent sur une notation dans laquelle les vecteurs en N dimensions, sont représentés par un vecteur en $N + 1$ dimensions. La valeur selon le $(N + 1)^{ième}$ axe est toujours fixée à l'unité 1.

Les coordonnées homogènes fournissent une notation uniforme pour les transformations. Pour cette raison, le recours aux coordonnées homogènes est expliqué par le fait que la représentation matricielle des translations est impossible en coordonnées cartésiennes, ce qui devient possible en coordonnées homogènes.

Généralement les transformations géométriques sont caractérisées par :

- Ce sont des applications d'un système de coordonnées vers un autre ou vers lui-même.
- Le modèle géométrique subit un changement par rapport à son modèle initial.
- Les transformations sont appliquées à un objet représenté par des points.

Par la suite, nous donnons quelques exemples de différentes formes canoniques d'une transformation géométrique en coordonnées homogènes :

2.3.3.1 Translation

Soit $\text{Trans}(t_x, t_y) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$ la matrice de translation en 2D dans le plan O_x et O_y

Par conséquent : $\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + t_x \\ y + t_y \\ 1 \end{pmatrix}$

Cette dernière vérifie que le point (x, y) est translaté en $(x + t_x, y + t_y)$.

Une translation en 3D définie par t_x, t_y, t_z dans la direction O_x, O_y et O_z respectivement, a la matrice suivante:

$$\text{Trans}(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le point $p(x, y, z, 1)$ est translaté au point $p'(x', y', z', 1)$ comme suit :

$$\begin{aligned}
 p' &= \text{Trans}(t_x, t_y, t_z) p \\
 &= \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \\
 &= \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{pmatrix}
 \end{aligned}$$

Par conséquent, le point (x, y, z) est translaté au $(x + t_x, y + t_y, z + t_z)$.

2.3.3.2 Mise à l'échelle

La matrice du changement d'échelle en 2D est définie par :

$$\text{Scale}(S_x, S_y) = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Par conséquent : } \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x * S_x \\ y * S_y \\ 1 \end{pmatrix}$$

Ce qui implique $\text{Scale}(x, y) = (x * S_x, y * S_y)$.

Une mise à l'échelle en 3D par rapport à l'origine définie par les facteurs S_x , S_y et S_z dans les plans O_x , O_y et O_z respectivement, est :

$$\text{Scale}(S_x, S_y, S_z) = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Donc : $\text{Scale}(x, y, z) = (x * S_x, y * S_y, z * S_z)$

2.3.3.3 Rotation

La matrice de transformation pour une rotation plane en 2D autour de l'axe O_z d'un angle Θ est :

$$\text{Rot}(\Theta) = \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Le point de coordonnée (x, y) subit une rotation autour de l'axe O_z , et par conséquent, les nouvelles coordonnées de ce point sont :

$$\begin{cases} x' = x * \cos(\theta) - y * \sin(\theta) \\ y' = x * \sin(\theta) + y * \cos(\theta) \end{cases}$$

Comme dans une rotation dans le plan, une rotation dans l'espace autour d'une ligne prend dénommé son axe de rotation. Toutes les rotations en 3D peuvent être exprimées en trois rotations primaires sur les axes O_x, O_y et O_z :

$$\text{Rot}_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Rot}_y(\theta_y) = \begin{pmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Rot}_z(\theta_z) = \begin{pmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.3.3.4 Composition des transformations

Beaucoup de transformations peuvent s'exprimer à l'aide de transformations élémentaires. Le produit de matrice étant associatif ($ABC = (AB)C = A(BC)$), cette relation peut être simplifiée en effectuant le produit des matrices et on obtient ainsi la matrice de la transformation globale.

Le produit des matrices n'est pas commutatif (AB n'est pas égale à BA en général) et donc on ne peut pas permuter les matrices dans le produit sans risquer de changer le résultat. Il faut bien faire attention à l'ordre des transformations. Comme nous l'avons décrit auparavant, il existe plusieurs chemins pour aligner (ramener) SF sur F . Ces chemins dépendent de l'ordre dans lequel les opérations sont effectuées (translations, rotations et mises en échelle).

2.3.4 La fonction coût QT

La représentation mathématique de la fonction coût à minimiser, peut se faire selon différents modèles. Dans ce qui suit, nous allons présenter quelques méthodes pouvant être utilisées dans ce type de situations.

2.3.4.1 Représentation par la méthode des moindres carrés

Comme son nom l'indique, cette méthode consiste à minimiser (moindre valeur) une addition de termes au carré. Elle permet de comparer des données expérimentales (généralement entachées d'erreurs de mesure) à un modèle mathématique censé décrire ces données. Dans le cas le plus courant, le modèle théorique est une famille de fonctions $P(x, \delta)$ d'une ou plusieurs variables x indexées par un ou plusieurs paramètres δ inconnus. La méthode des moindres carrés permet de sélectionner parmi ces fonctions celle qui reproduit le mieux des données expérimentales. On parle dans ce cas d'ajustement par la méthode des moindres carrés.

Cette méthode consiste en une prescription (initialement empirique), qui est la fonction $P(x; \delta)$ qui décrit efficacement les données, est celle qui minimise la somme quadratique des déviations des mesures aux prédictions de $P(x; \delta)$. Si par exemple, nous disposons de $N_{\text{mesures}}, y_{i(i=1, \dots, N)}$, les paramètres δ optimaux, au sens de la méthode des moindres carrés, sont ceux qui minimisent la quantité :

$$QT(\delta) = \sum_{i=1}^N (y_i - P(x_i; \delta))^2 = \sum_{i=1}^N r_i^2(\delta)$$

Où les $r_i(\delta)$ sont les résidus du modèle (les écarts entre les points de mesure y_i et le modèle $P(x; \delta)$).

La quantité $QT(\delta)$ peut être considérée comme une mesure de la distance quadratique entre les données expérimentales et le modèle théorique qui prédit ces données. La prescription des moindres carrés exige que cette distance soit minimale.

Sa grande simplicité fait que cette méthode est très couramment utilisée de nos jours en sciences expérimentales. Une des applications courantes est le lissage des données expérimentales par une fonction empirique (fonction linéaire, polynômes ou splines). Cependant, son usage le plus important est probablement la mesure des quantités physiques à partir de données expérimentales.

Dans de nombreux cas, la quantité qu'on cherche à mesurer, n'est pas observable et n'apparaît qu'indirectement comme paramètre δ d'un modèle théorique $P(x; \delta)$. Dans ce cas de figure, il est possible de montrer que la méthode des moindres carrés permet de construire un estimateur de δ qui vérifie certaines conditions d'optimalité.

Par ailleurs, dans tous les cas, les estimateurs obtenus sont extrêmement sensibles aux points aberrants, de ce fait, ils ne sont pas robustes. Plusieurs techniques permettent cependant d'améliorer la dite méthode: LMS (Least Median of squares) et LTS (Least Trimmed Squares) [Leroy and Rousseeuw, 1987; Rousseeuw, 1984], M-estimateurs, L-Estimateurs ou R-estimateurs [Huber, 2011]. Ces techniques visent à redéfinir la fonction coût à minimiser afin que le minimum global ne soit pas affecté par les données aberrantes [Malis and Marchand, 2005].

2.3.4.2 Minimisation de la fonction coût QT

Généralement, notre fonction coût est représentée sous la forme des moindres carrés non linéaires, et par conséquent elle est positive. La résolution de celle-ci peut donc faire appel aux méthodes qui passent par la recherche des paramètres du zéro de la dérivée correspondante à un minimum global ou à un minimum local dans le cas plutôt défavorable. Les valeurs correspondantes à ce minimum du dérivé, définissent aussi le minimum (local ou global) de notre fonction coût (QT égale à zéro dans le cas idéal). Ces méthodes appartiennent aux catégories dites de résolution (minimisation) d'une équation non-linéaire à un ou plusieurs inconnus. Parmi les méthodes les plus utilisées, on peut citer à titre d'exemple la méthode de NEWTON-RAPHSON, l'algorithme de Levenberg-Marquardt et la méthode de Nelder- MEAD (SIMPLEX) qui seront décrites par la suite.

2.3.4.2.1 La méthode de NEWTON-RAPHSON

Rappelons que le point optimum (optimal) est la racine de la dérivée de la fonction. Si la fonction à minimiser $f(\alpha_1, \alpha_2, \dots, \alpha_n)$, les dérivées première ∇_f et seconde (hessien) $[H]_f$, ainsi que le vecteur de différence indépendante γ de la fonction peuvent être écrites comme suivant:

$$\nabla_f = \begin{Bmatrix} \frac{D_f}{D\alpha_1} \\ \frac{D_f}{D\alpha_2} \\ \vdots \\ \frac{D_f}{D\alpha_n} \end{Bmatrix} \quad [H]_f = \begin{bmatrix} \frac{D^2_f}{D\alpha_1^2} & \dots & \frac{D^2_f}{D\alpha_1 D\alpha_n} \\ \vdots & \ddots & \vdots \\ \frac{D^2_f}{D\alpha_1 D\alpha_n} & \dots & \frac{D^2_f}{D\alpha_n^2} \end{bmatrix} \quad \gamma^{m+1} = \begin{Bmatrix} (\alpha_1^{m+1} - \alpha_1^m) \\ (\alpha_2^{m+1} - \alpha_2^m) \\ \vdots \\ (\alpha_n^{m+1} - \alpha_n^m) \end{Bmatrix}$$

La $(m + 1)^{\text{ième}}$ itération pour la méthode de Newton-Paphson est définie par :

$$\{\gamma^{m+1}\} = -\{\nabla_f^m\} [H_f^m]^{-1}$$

L'équation peut être résolue en utilisant plusieurs méthodes (telle que l'élimination de gauss). Une estimation initiale pour toutes les valeurs de α est exigée pour commencer la procédure itérative. Les dérivées premières et secondes, peuvent être décrites analytiquement si la fonction à minimiser est bien définie, ou encore elles peuvent être calculées numériquement mais cela augmentera les erreurs du nombre de pas itératifs ainsi que le temps de calcul.

2.3.4.2.2 L'algorithme de Levenberg-Marquardt (LM)

L'algorithme de Levenberg-Marquardt (LM) est l'algorithme d'optimisation le plus largement utilisé. Il surpasse la méthode de la descente de gradient simple et d'autres méthodes de gradient conjugué pour une grande variété de problèmes. Cet algorithme combine l'algorithme de Gauss-Newton et l'algorithme du gradient. Plus stable que celui de Gauss-Newton, il trouve une solution même s'il démarre très loin du minimum, ce qui est l'un des points faibles de Gauss-Newton. Cependant, pour certaines fonctions très régulières, il peut converger légèrement moins vite.

Le problème pour lequel l'algorithme de LM propose une solution est appelé minimisation des moindres carrés non linéaires. Cela signifie que la fonction à minimiser est de la forme suivante:

$$f(\alpha) = \frac{1}{2} \sum_{j=1}^m r_j^2(\alpha)$$

Où $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ est le vecteur des paramètres et les r_j sont les résidus.

La dérivée de f peut être décrite en utilisant la matrice Jacobéenne J définie par :

$$J(\alpha) = \frac{Dr_j}{D\alpha_i} \quad 1 \leq j \leq m, \quad 1 \leq i \leq n$$

La descente de gradient est la plus simple technique et la plus intuitive pour trouver le minimum d'une fonction. Des mises à jour des paramètres sont effectuées à savoir :

$$\alpha_{i+1} = \alpha_i - \lambda \nabla f$$

Cette méthode de descente de gradient simple connaît divers problèmes de convergence. En effet, si nous aimerions prendre de grandes marches à descendre la pente dans des endroits où elle est faible (la pente est douce) et inversement, nous faisons de petits pas lorsque le gradient est grand, afin de ne pas ébranler sur les minimas ; ce qui est contradictoire avec la

règle de mise à jour ci-dessus. Un autre problème c'est que la courbure de la surface d'erreur ne peut pas être la même dans toutes les directions.

Cette situation peut être améliorée, en utilisant la courbure ainsi que des informations du gradient, à savoir les dérivées secondes. Une façon de le faire est d'utiliser la méthode de Newton pour résoudre l'équation $\nabla f(\alpha) = 0$.

La méthode de Newton utilise implicitement une hypothèse quadratique sur f , le Hessien n'a pas besoin d'être évalué précisément. Au contraire, une approximation peut être utilisée. Le principal avantage de cette technique est la rapidité de convergence. Cependant, la vitesse de cette convergence est sensible au point de départ.

On peut voir que la méthode de descente de gradient simple et la méthode de Gauss-Newton, sont complémentaires dans les avantages qu'elles offrent. Levenberg propose un algorithme basé sur cette observation dont la règle de mise à jour est un mélange des algorithmes mentionnés ci-dessus et elle est donnée comme suit :

$$\alpha_{i+1} = \alpha_i - (H + \lambda I)^{-1} \nabla f(\alpha_i)$$

H est la matrice de Hessien évaluée à α . Cette règle de mise à jour est utilisée comme suit : Si l'erreur va vers le bas, suite à une mise à jour, cela signifie que notre hypothèse quadratique sur $f(x)$ sert à réduire λ (généralement par un facteur de 10) afin de réduire l'influence de la descente de gradient. D'autre part, si l'erreur augmente, nous aimerions suivre le gradient de telle sorte que λ augmente par le même facteur. L'algorithme de Levenberg peut donc être résumé comme suit :

1. Effectuer une mise à jour selon les directives de la règle ci-dessus.
2. Évaluer l'erreur dans le nouveau vecteur de paramètres.
3. Rétracter l'étape, si l'erreur a augmenté suite à cette mise à jour, rétablir les poids à leurs valeurs précédentes et augmenter λ par un facteur de 10. Ensuite, allez à (1) et essayer une mise à jour à nouveau.
4. Accepter l'étape, si l'erreur a diminué suite à cette mise à jour, maintenir les poids à leurs nouvelles valeurs et diminuer λ par un facteur de 10 ou plus.

L'algorithme ci-dessus présente un inconvénient : si la valeur de λ est grande, la matrice de Hesse calculée est inutilisable. Nous pouvons tirer un avantage de la dérivée seconde, même dans de tels cas, par une multiplication de chacune des composantes du gradient en fonction de la courbure. Il devrait en résulter un mouvement plus large le long des directions où le

gradient est plus petit de telle sorte que le problème "erreur de la vallée" classique ne se produise pas. Cette idée essentielle a été fournie par Marquardt. Il a remplacé la matrice d'identité avec la diagonale de la Hesse qui donne la règle de la mise à jour de Levenberg-Marquardt suivante :

$$\alpha_{i+1} = \alpha_i - (H + \lambda \text{diag}[H])^{-1} \nabla f(\alpha_i)$$

Puisque la Hesse est proportionnelle à la courbure de f , cette dernière règle de mise à jour entraînera un grand pas dans la direction à faible courbure et un petit pas dans la direction de courbure élevée.

Bien que la méthode de LM soit en aucune façon optimale, mais juste une heuristique, elle fonctionne très bien dans la pratique. L'un de ses inconvénients est son besoin de la matrice inversée de dans le cadre de la mise à jour. En plus, comme tous les algorithmes d'optimisation, l'algorithme de LM nécessite une initialisation pour les paramètres à estimer.

2.3.4.2.3 Nelder- MEAD SIMPLEX

Le problème classique d'optimisation est de trouver un point de minimum (ou maximum) x^* d'une fonction donnée $f: \mathcal{R}_n \rightarrow \mathcal{R}$. Il y a une grande variété de méthodes dans la littérature à notre disposition pour résoudre ce problème, en fonction de la quantité d'informations disponibles concernant f . Dans certaines applications nous pouvons appliquer cet algorithme dans les cas où :

- f est continue mais pas lisse.
- f n'est pas continue.

Pour trouver la solution x^* , la méthode d'optimisation ne devrait utiliser que les valeurs de la fonction f sans l'estimation des dérivées de f car elles peuvent ne pas exister à un moment donné. Ce procédé appartient à la catégorie des méthodes dites de recherche directe.

La méthode du simplexe de Nelder-Mead est un procédé de recherche multidimensionnel. Le simplexe (Figure 2.2) est une entité géométrique en n -dimensions contient $(n + 1)$ points dans un espace de dimension n . Le procédé utilise le concept d'un simplexe qui est un polytope spécial de $N + 1$ sommets à N dimensions.

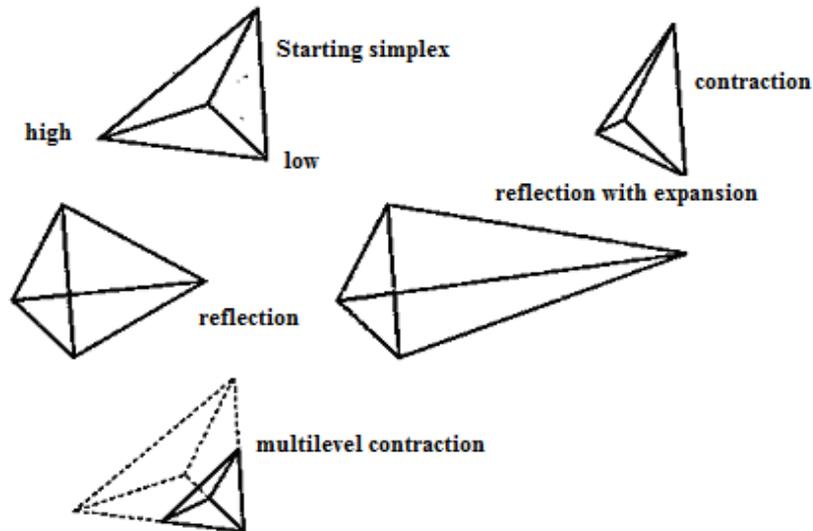


Figure 2.2 Schéma du simplex

La méthode du simplex peut donc être définie de la manière suivante :

- 1) Ordonner f en fonction des valeurs des sommets:

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$$

- 2) Calculer x_0 , le centre de gravité de tous les points sauf x_{n+1} .
- 3) Réflexion :

- Calculer le point de reflété $x_r = x_0 + \alpha(x_0 - x_{n+1})$
- Si le point réfléchi est mieux que le deuxième, mais pas mieux que le meilleur :

$$f(x_1) \leq f(x_r) < f(x_n)$$

- Pour obtenir un nouveau simplex, remplacer le mauvais moment x_{n+1} avec le point x_r réfléchi, et revenir à l'étape 1.

- 4) Expansion :

- Si le point réfléchi est le meilleur point obtenu jusqu'à cet instant $f(x_r) < f(x_1)$, on calcule le point élargi x_e par : $x_e = x_0 + \gamma(x_0 + x_{n+1})$
- Si ce point élargi est mieux que le point réfléchi $f(x_e) < f(x_r)$, remplacer le mauvais moment x_{n+1} avec le point x_e élargi, et revenir à l'étape 1.
- Sinon, remplacer le mauvais moment x_{n+1} avec le point x_r réfléchi, et reprendre l'étape 1.

5) Contraction :

À ce niveau, il est certain que $f(x_r) \geq f(x_n)$

- Calculer alors le point contracté x_c par : $x_c = x_{n+1} + \rho(x_0 - x_{n+1})$
- Si le point contracté est mieux que le point le plus mauvais, à savoir $f(x_c) \leq f(x_{n+1})$, remplacer le moment x_{n+1} par le point x_c , et revenir à l'étape 1.

6) Réduction :

Sauf le meilleur point, remplacer tous les points x_i avec $x_i = x_1 + \sigma(x_i - x_1) \forall i \in \{2, \dots, n + 1\}$ et reprendre l'étape 1.

Remarques :

- α, γ, ρ et σ sont respectivement les coefficients de la réflexion, de l'expansion, de la contraction et du retrait. Ses valeurs normalisées sont $\alpha = 1, \gamma = 2, \rho = \frac{1}{2}$ et $\sigma = \frac{1}{2}$.
- Pour la réflexion, puisque x_{n+1} est le sommet de valeur plus élevée, nous attendons à trouver une valeur inférieure à la réflexion du x_{n+1} dans la face opposée formée par tous les sommets x_i sauf x_{n+1} .
- Pour l'expansion, si le point de réflexion x_r est le nouveau minimum le long des sommets, nous attendons à trouver des valeurs intéressantes dans la direction depuis x_0 à x_r .
- Pour la contraction, si $f(x_r) > f(x_n)$, nous attendons qu'une meilleure valeur soit à l'intérieur du simplexe formé par tous les sommets x_i .
- Le simplexe initial est important ; en effet, un trop petit simplexe initial peut conduire à une recherche locale et par conséquent, le NM peut facilement se retrouver dans ce qu'on peut appeler un puits de potentiel. Ce simplexe devrait, par obligation, dépendre de la nature du problème.

2.3.5 RMS

L'erreur quadratique moyenne (Root Mean Square Error RMSE, ou Root Mean Square Deviation RMSD) est un critère puissant et largement utilisé pour estimer la similarité entre deux ensembles ordonnés de points. En effet, la RMS est utilisée pour calculer les différences entre les valeurs prévues par un modèle ou un estimateur et les valeurs réellement observées. Ces différences individuelles sont appelées résidus lorsque les calculs sont effectués sur l'échantillon de données qui ont été utilisées pour l'estimation et sont appelées erreurs de prédiction lorsque elles sont calculées hors échantillon.

La RMS sert à regrouper les résidus en une seule mesure positive de puissance prédictive. Cette mesure de similarité est utilisée dans notre approche pour juger l'inclusion de SF dans F. Cependant, la RMS n'est pas la seule mesure qui peut être utilisée à ce niveau. À titre d'exemple, on peut remplacer la RMS par le coefficient de corrélation de Pearson ou par le coefficient de Nash-Sutcliffe :

❖ Root Mean Square :

$$RMS = \sqrt{\frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{n}}$$

❖ Root Mean Square Normalisé:

$$NRMS = \frac{RMS}{X_{obs,max} - X_{obs,min}}$$

$$NRMS = \frac{RMS}{\bar{X}_{obs}}$$

❖ Le coefficient de corrélation de Pearson

$$R = \frac{\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 * \sum_{i=1}^n (y_i - \bar{y})^2}}$$

❖ Le coefficient de Nash-Sutcliffe

$$E = 1 - \frac{\sum_{i=1}^n (X_{obs,i} - X_{model,i})^2}{\sum_{i=1}^n (X_{obs,i} - \bar{X}_{model})^2}$$

2.4 Résultats expérimentaux

Cette section présente les résultats expérimentaux qui visent à démontrer la robustesse de notre approche pour l'appariement de formes à base d'un modèle de points. Les performances de l'algorithme sont analysées par plusieurs tests effectués sur des ensembles de données en 2D et en 3D.

2.4.1 Application à un nuage 2D

Afin de tester les performances de l'algorithme en 2D, nous générons un ensemble de données synthétiques. Un exemple simple est montré sur la figure 2.3 dont l'ensemble de modèle (the model set F) est composé de 153 points ($n = 153$). Les figures 2.4 et 2.5 donnent un exemple de l'ensemble de données (SF) composé de 16 points ($m = 16$) sélectionnés parmi les points de F.

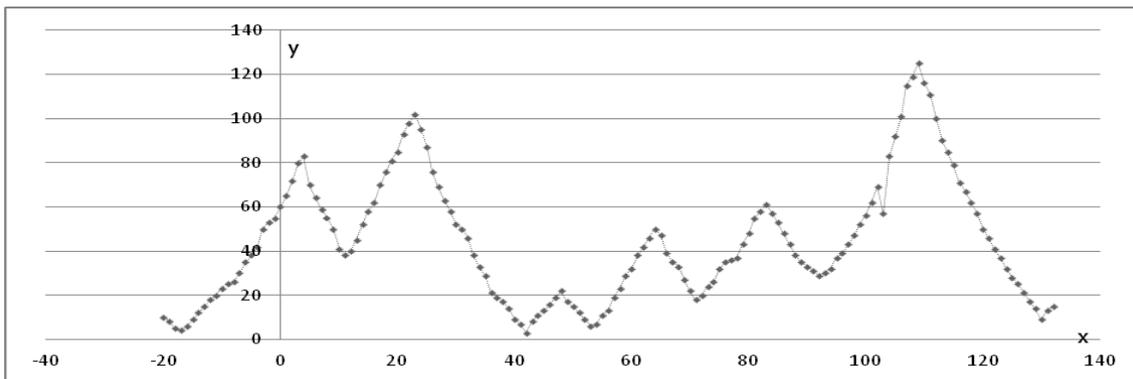


Figure 2.3 Model set F avec $n=153$

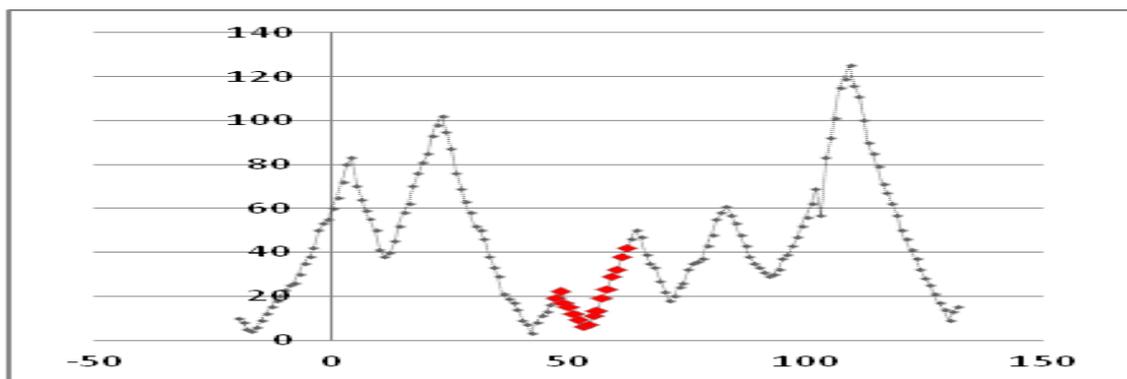


Figure 2.4 Exemple du data set SF avec $m= 16$. Les points sont pris aléatoirement et d'une manière successive

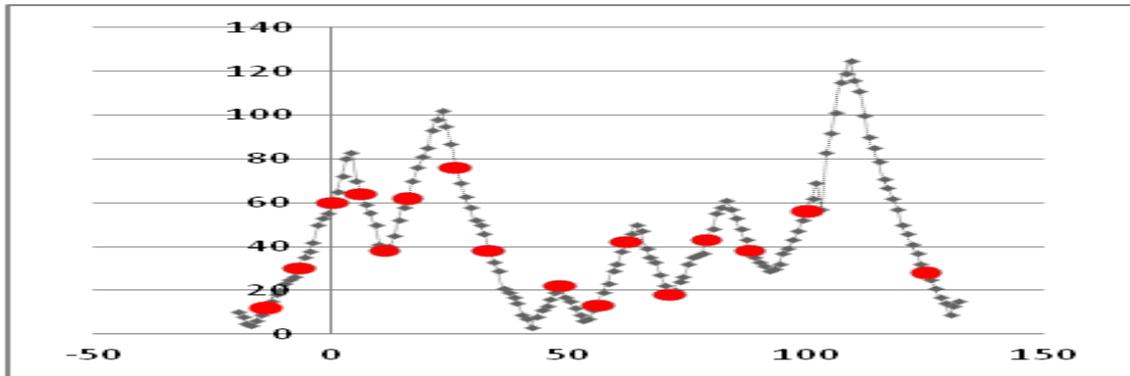


Figure 2.5 Exemple du data set SF avec $m=16$. Les points sont pris d'une manière aléatoire et non successive

Après avoir sélectionné aléatoirement (successivement ou non) des points à partir de F pour construire SF (figures 2.4 et 2.5). L'étape suivante consiste à appliquer une transformation aléatoire affine G sur ces points de SF telle que :

$$G = \begin{cases} t_x = 100 * \beta \\ t_y = 100 * \beta \\ \theta_z = 2 * \pi * \beta \\ \lambda = 0.1 + (5 * \beta) \end{cases} \quad \text{Avec } \beta \text{ choisi aléatoirement dans } [-1,1]$$

Remarquons que le facteur λ peut, selon β , correspondre à un zoom en avant ou un zoom en arrière. Le traitement du problème d'image miroir de figure est obtenu avec un β négatif. Ce dernier est sans intérêt dans notre situation et donc ne sera pas pris en compte.

À chacun des types de jeux de données de synthèse, nous générons 100 fois et à chaque fois une nouvelle transformation aléatoire G est appliquée sur tous les points du SF . La figure 2.6 illustre les résultats de tous les cas où pour chaque test, la taille (nombre de points) de l'ensemble des points SF est augmentée d'un point.

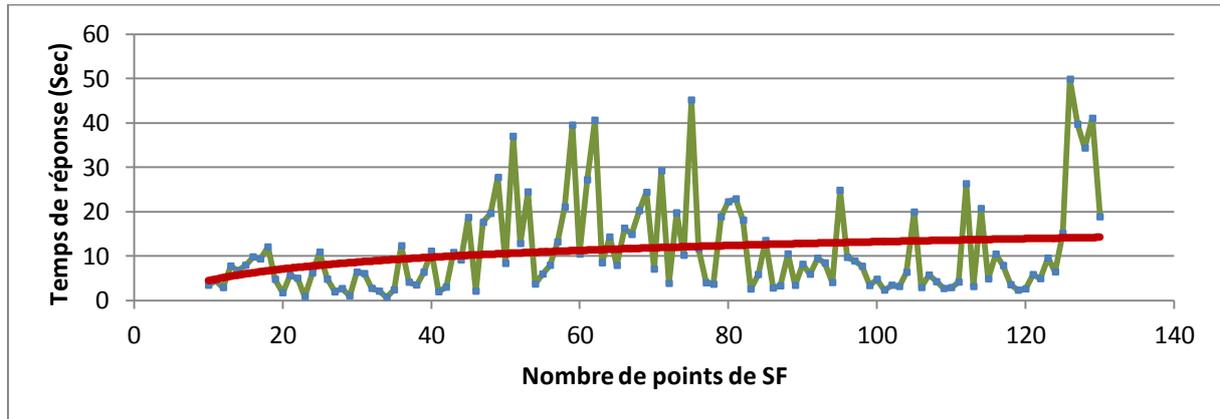


Figure 2.6 Temps d'exécution pour chaque ensemble de points de SF de taille m

Les tendances générales du temps d'exécution dépendent des tailles (n et m) des deux motifs de points à apparier. Dans notre expérience, il est augmenté en fonction log. Son non-uniformité est due au fait que la question est traitée dans son intégralité. En effet, l'augmentation de la taille de SF est normalement accompagnée d'une augmentation du temps de réponse or sur la base de notre algorithme, tout article supplémentaire (point-clé) ajoute une contrainte supplémentaire minimisant le nombre de concurrent (minima locaux), et par conséquent plus d'informations géométriques pour calculer la transformation T' , et donc devrait en principe conduire à une réduction du temps de réponse. En général, tous nos résultats expérimentaux ont montré que les performances de l'algorithme proposé sont satisfaisantes.

2.4.2 Application à un nuage 3D

Pour nos expériences en 3D, nous utilisons la base de données de référence générique SHREC 2009 [Godil et al., 2009]. Cette référence contient 720 formes organisées en 40 classes avec 18 formes par classe. La figure 2.7 montre quelques exemples de modèles 3D représentés sous forme d'un nuage de points pris à partir de cette base de données.

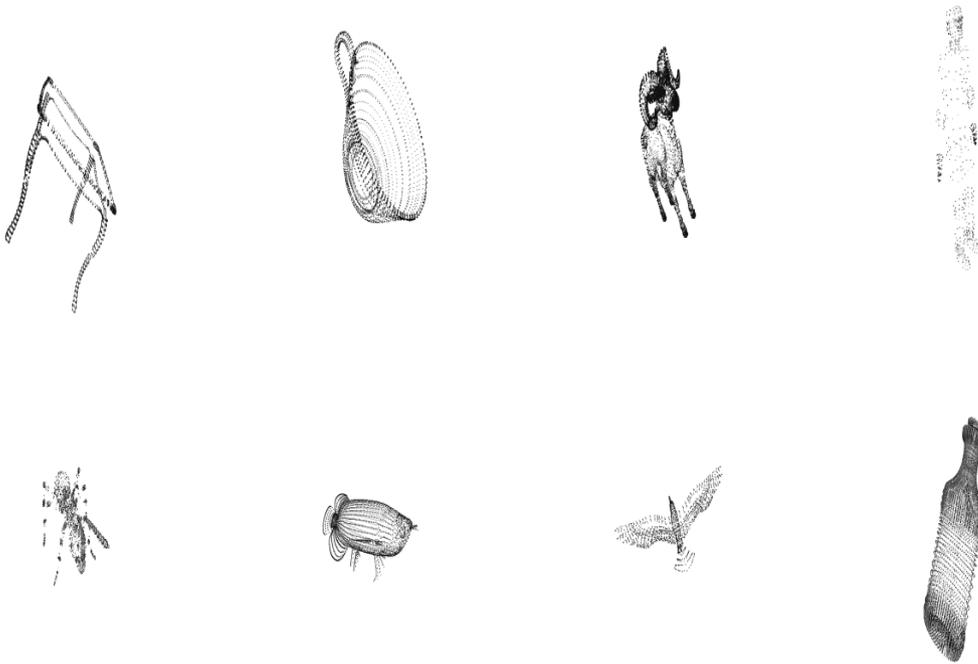


Figure 2.7 Exemple des modèles 3D sous forme du nuage de points

Les expériences ont été réalisées dont le but de chercher l'appariement partiel et complet. Tout au long de ces expériences, le même principe de génération du jeux de données a été adopté, avec un changement de la transformation affine pour qu'elle devienne:

$$G = \begin{cases} t_x = 100 * \beta & \theta_x = 2 * \pi * \beta \\ t_y = 100 * \beta & \theta_y = 2 * \pi * \beta \\ t_z = 100 * \beta & \theta_z = 2 * \pi * \beta \\ \lambda = 0.2 + (5 * \beta) \end{cases} \quad \text{avec } \beta \in [-1,1]$$

Dans la première expérience, un modèle de vache a été pris comme ensemble de modèle (the model set avec n = 3093 points). La figure 2.8 illustre ce modèle et la figure 2.9 illustre quelques exemples d'ensemble de données (data set SF).

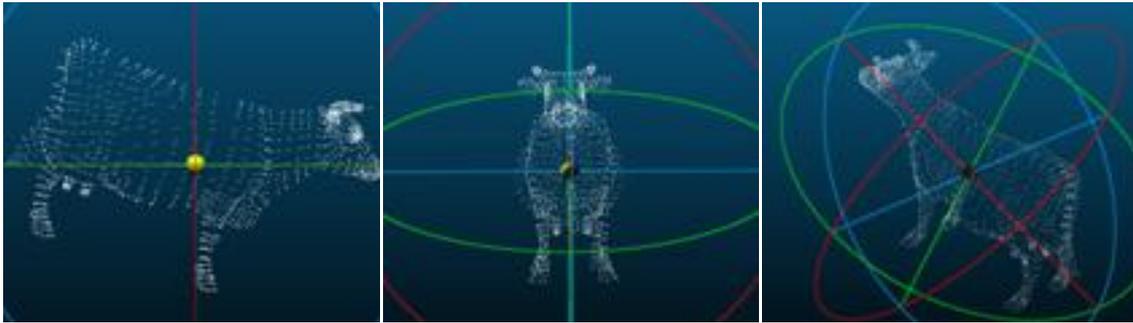


Figure 2.8 Model set F avec $n=3093$

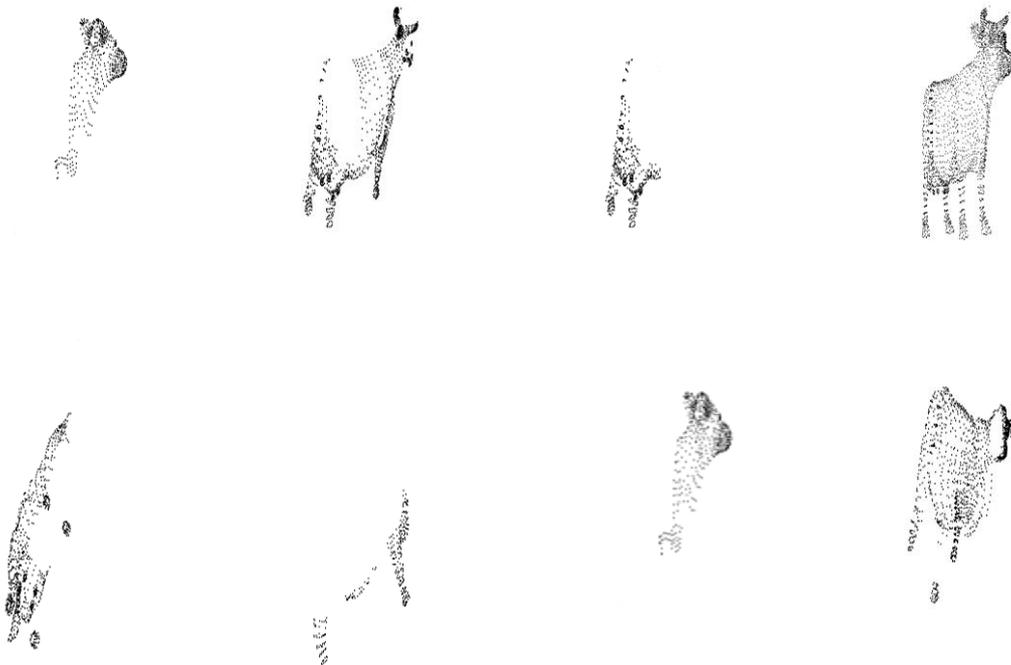


Figure 2.9 Exemple du data set SF pour le problème d'appariement partiel

La taille de SF est augmentée à chaque fois de 100 points. Pour chacun des types de jeux de données, nous générons 100 fois et à chaque fois avec une nouvelle transformation aléatoire G qui est appliquée sur tous les points de SF. La figure 2.10 montre la durée moyenne du temps d'exécution pour chaque taille de SF.

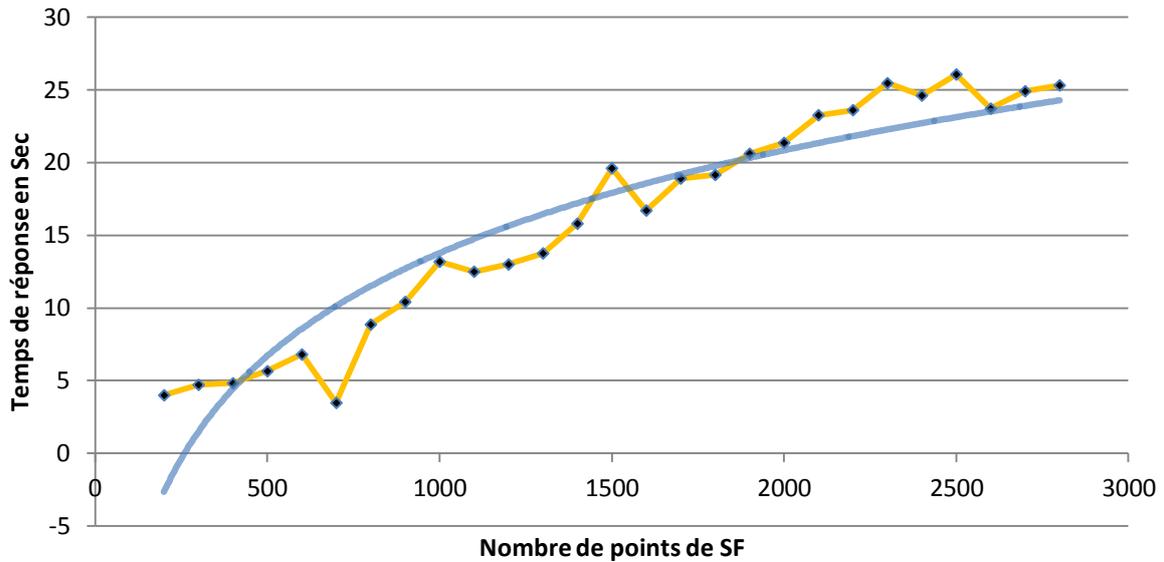


Figure 2.10 La durée moyenne du temps d'exécution pour chaque taille de SF

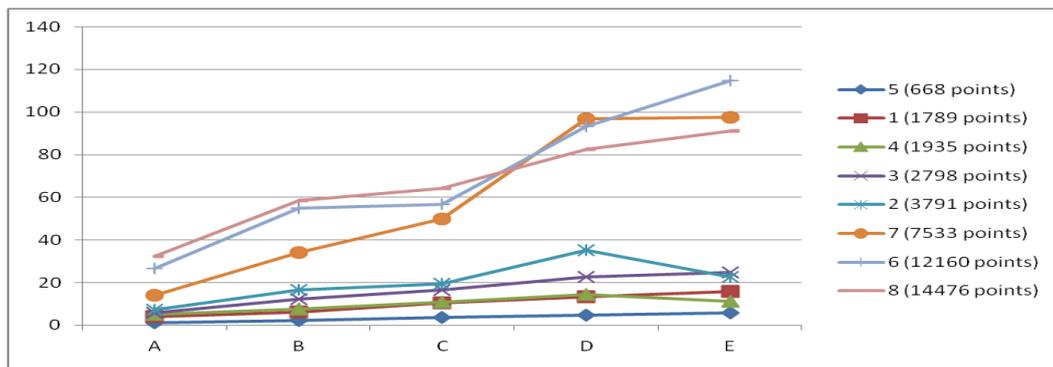


Figure 2.11 Le temps moyen d'exécution pour huit modèles différents

La figure 2.11 montre le temps moyen d'exécution pour huit modèles différents, où le test A signifie que $m = 1/5 n$, le test B signifie que $m = 2/5 n$, et ainsi de suite. Les modèles utilisés dans cette expérience sont sélectionnés aléatoirement de la même base de données. Les nombres de points sont distribués de 668 points jusqu'à 14476 points. Le temps de réponse est relatif à la taille des nuages de points à apparier ainsi qu'à la topologie des modèles.

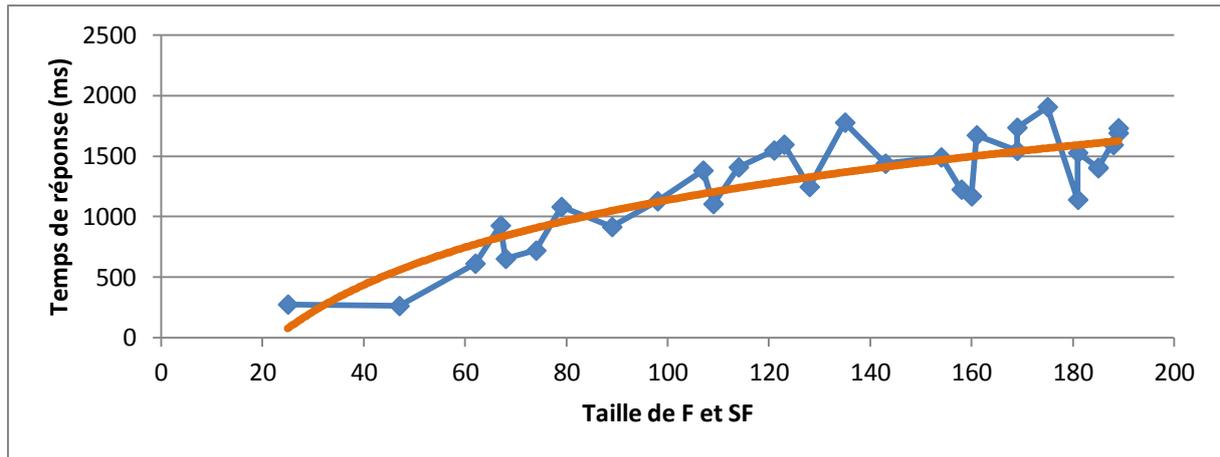


Figure 2.12 Le temps moyen d'exécution pour 26 modèles pour le cas d'appariement complète ($n = m$)

La figure 2.12 illustre le temps moyen d'exécution pour 26 modèles dans le cas d'appariement complet ($n = m$), où l'axe des abscisses représente les tailles des modèles et l'axe des ordonnées représente le temps d'exécution.

Ces tests nous amènent à l'objectif de la projection / interpolation de ces modèles 3D sur des plans 2D séparés. En effet, ce procédé de projection/interpolation nous permet de généraliser notre approche à d-dimensions avec une variation logarithmique de la fonction du temps. Cependant, il peut être utile pour la recherche 2D-3D (SF est en 2D et F est en 3D et vis versa). Tout au long de nos expériences en 2D et en 3D, notre algorithme nous permet d'obtenir de très bons résultats pour la reconnaissance de formes. Il peut être utilisé comme un algorithme de reconnaissance si on a un accès direct aux formes à appairer (modèles sous forme de nuages de points) ou comme une composante (matching algorithm) dans la chaîne de reconnaissance pour appairer les points d'intérêt.

Enfin, les résultats présentés dans ce chapitre ont fait l'objet de publications [Aqili et al., 2016c] dans laquelle nous avons prouvé que la complexité de notre algorithme est logarithmique ($m \log n$), chose qui nous permet de dire que :

- d'une part, notre algorithme est le meilleur dans sa catégorie (tableau 2.1), et cela revient du fait qu'il traite les formes à appairer dans leurs totalités et non pas point par point.
- d'autre part, notre algorithme fait partie des algorithmes généralistes, du fait qu'il peut être utilisé à D-dimension ainsi il n'est pas dédié à une seule application spécifique et par conséquent, il est ouvert à toutes les applications possibles pouvant être traitées par PPM.

2.4.3 Benchmark des complexités

Tableau 2:1 Complexité de certains algorithmes

Auteurs	Technique	Propriétés	Complexité
Notre approche	Discret to Continuos	Translation, rotation, mise à l'échelle	$O(m \log n)$
Ranade et Rosenfeld	Relaxation	Translation	$O(n^4)$
Ogawa	Fuzzy Relaxation	Translation, rotation, mise à l'échelle	$O(n^6)$
Huttenlocher et Ullman	Alignment	Translation, rotation, mise à l'échelle	$O(m^3 n^2 \log n)$
Vinod et Ghose	Asymmetric Neural Network	Translation, rotation, distorsion et bruit	
Irani et Raghavan	Randomized alignment	Translation, rotation, mise à l'échelle	$O(mn^2 \log n)$
Chang et al	2D Cluster	Translation, rotation, mise à l'échelle, distorsion local, extra et/ou missing point	$O(n^4)$
Boxer	Un séquentiel algorithme	Translation et rotation dans 3D	$O(n^2 (\frac{\lambda_6(n)}{n})^{1/2}) \log n$
Chang et al	Nearest Neighbors Search	Translation, rotation, mise à l'échelle, distorsion local, extra et/ou missing point	$O(k^2 n^2)$
Van Wamelen et al	Probabilistic Sorted Nearest Neighbors	Translation, rotation, mise à l'échelle, distorsion local, extra et/ou missing point	$O(n(\log m)^{3/2})$

On constate clairement que notre approche présente une complexité meilleure comparée à ses concurrents. Les taux de reconnaissances seront discutés selon le domaine d'application dans le chapitre suivant.

2.5 Conclusion

Comme nous l'avons annoncé dans la conclusion du premier chapitre, le caractère discret des entités traitées est l'une des raisons des risques d'explosion combinatoire. Notre algorithme, qui propose une interpolation du nuage de point de l'une des entités à traiter, permet transformer le problème de reconnaissance en superposant la sous forme sur une partie ou sur la totalité de la première forme en fonction des données disponibles. Cette nouvelle approche permet de traiter la question dans sa totalité au lieu de point par point évitant ainsi le retour en arrière (backtracking) comme avec pratiquement toutes les méthodes.

Contrairement aux autres méthodes, la complexité de la topologie de la forme et l'augmentation de la taille introduisent de nouvelles contraintes de superposition de la question sur la forme, ce qui permet de réduire le temps de réponse dans notre approche.

Il est à signaler que notre algorithme sait distinguer la forme et son image miroir. Cette caractéristique est dans certains domaines très importants.

Alors que toutes les méthodes existantes commencent dans un premier temps par rechercher la correspondance de points avant de procéder à une superposition de la question sur la forme afin d'apprécier le résultat obtenu, notre approche commence par la recherche de la meilleure superposition (sur une représentation continue) pour en déduire les meilleurs correspondances.

Et enfin, il ne serait pas trop prétentieux d'annoncer de notre part que notre approche pourrait être considéré comme généraliste pour le domaine des reconnaissances de formes discret et ce par la définition d'une métrique adéquate qui servira à optimiser la superposition de la sous-forme sur la forme.

Chapitre 3 Applications à la reconnaissance biométriques

Ce chapitre est destiné à représenter l'efficacité de notre algorithme dans des applications réelles tel le domaine de la biométrie. Le recours aux techniques de reconnaissances biométriques demeure une méthode convenable pour reconnaître les individus et essentiellement pour établir et renforcer la sécurité des individus, des établissements et des états. En fait, l'efficacité des techniques de reconnaissances biométriques (entre autres) est essentiellement dépendante de l'algorithme de comparaison utilisé. Cette partie traite deux aspects de la biométrie à savoir la biométrie des empreintes digitales et celle de la signature en ligne. Enfin nous présentons les résultats expérimentaux obtenus l'approche proposée.

3.1 Introduction au biométrie

La biométrie est la science de reconnaître l'identité d'une personne basée sur les attributs physiques, chimiques ou comportementaux de l'individu comme le visage, les empreintes digitales, la voix et l'iris. Elle se positionne comme une technologie fiable qui peut être intégrée dans des systèmes de gestion d'identité à grande échelle. Vu la nécessité augmentée pour des techniques de reconnaissance robustes dans des applications critiques telles que la surveillance, le contrôle d'accès sécurisé, le passage de la frontière internationale, le renforcement de la loi, le partage des ressources informatiques en réseau, l'accès aux installations nucléaires, le recours aux transactions financières à distance et aussi dans des applications militaires.

Les systèmes biométriques sont basés sur la notion que la plupart des caractéristiques physiques et comportementales des êtres humains sont distinctive, et qu'elles peuvent être acquises de façon fiable via des capteurs convenables et représentées sous un format numérique.

La croissance rapide dans le déploiement des systèmes biométriques, a clairement mis en évidence des défis associés à la conception et l'intégration de ces systèmes. Ces dernières années ont vu une croissance significative dans la recherche biométrique aboutissant au développement des capteurs innovants, des nouvelles techniques d'extraction des caractéristiques et des algorithmes de comparaisons et de mise en correspondances.

La tâche primordiale d'un système de gestion d'identité est la détermination (ou la vérification) de l'identité d'un individu (ou l'identité revendiquée). Une telle action peut être nécessaire pour une variété de raisons, mais l'intention primaire, dans la plupart des

applications, est d'empêcher les imposteurs d'avoir accès aux ressources protégées. Les méthodes traditionnelles d'établir l'identité d'une personne incluent des mécanismes à base de connaissances (les mots de passe) et des mécanismes à base de jetons (les cartes d'identité). Cependant, ces représentations de substitut d'identité peuvent facilement être perdues, partagées, manipulées ou volées ainsi mettant en péril la sécurité souhaitée.

La biométrie offre une solution naturelle et fiable à certains aspects de la gestion d'identité, en utilisant des systèmes entièrement automatisés ou semi-automatisés pour reconnaître des individus en fonction de leurs caractéristiques biologiques [Jain et al., 2004]. En utilisant la biométrie, il est possible d'établir une identité fondée sur qui vous êtes, plutôt que de ce que vous possédez. Dans certaines applications, la biométrie peut être utilisée pour compléter les cartes d'identité et les mots de passe offrant ainsi un niveau de sécurité supplémentaire. Un tel arrangement est souvent appelé un système d'authentification de facteur doublé.

L'efficacité d'un authentifiant (biométrique ou non biométrique) est basée sur sa capacité d'adaptation pour une application particulière, ainsi que sa robustesse à divers types d'attaques malveillantes. Bien que certaines de ces attaques puissent être évitées par l'incorporation des mécanismes de défense appropriés. En revanche il reste impossible de traiter tous les problèmes associés à l'utilisation des mots de passe et des jetons.

Les systèmes biométriques utilisent une variété de caractéristiques physiques ou comportementales (Figure 3.1), y compris l'empreinte digitale, le visage, la géométrie de main/doigt, l'iris, la rétine, la signature, la démarche, l'empreinte de la paume (palmier), l'empreinte vocale, l'oreille, la veine, l'odeur ou les informations d'ADN d'un individu pour établir l'identité [Jain et al., 2006, 2005]. Tandis que les systèmes biométriques ont leurs propres limitations [O'Gorman, 2002] ils ont un bord sur les méthodes de sécurité traditionnelles. Ils ne peuvent pas être facilement volés ou partagés. Outre le renforcement de la sécurité, ces systèmes biométriques améliorent aussi la commodité de son utilisateur, en soulageant le besoin de concevoir et de se rappeler des mots de passe.

Un système biométrique est essentiellement un système de reconnaissance de formes qui acquit des données biométriques d'un individu, extrait un ensemble de caractéristiques saillantes à partir de ces données, compare cet ensemble de caractéristiques contre l'ensemble (s) de caractéristiques stockés dans la base de données et exécute une action basée sur le résultat de la comparaison. Donc, un système biométrique peut être vu comme un système

composé de quatre modules principaux : un module de capteur, un module d'extraction des caractéristiques, un module de comparaison et un module de base de données. Seuls le module d'extractions de caractéristique et le module de comparaison seront décrits ci-dessous.

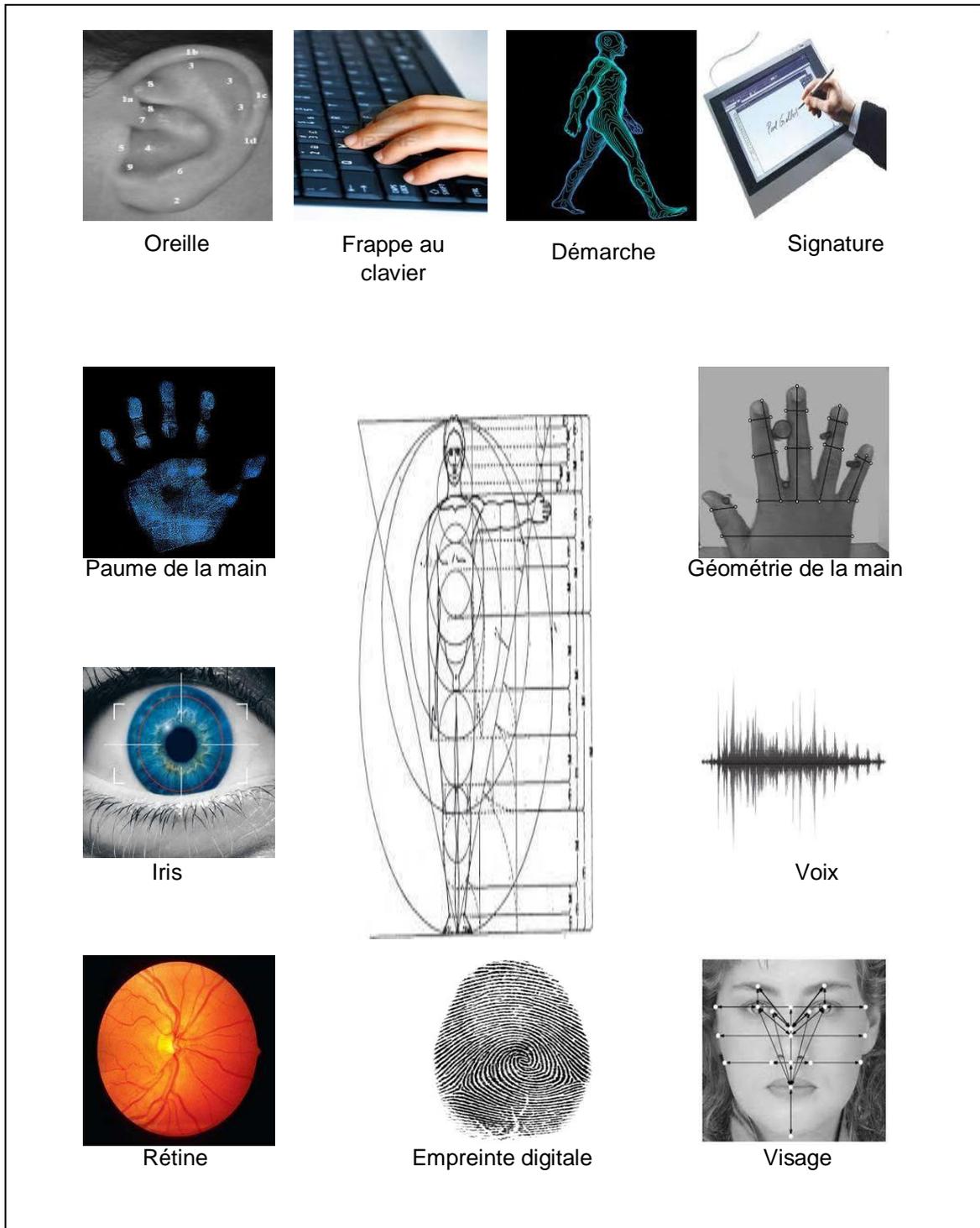


Figure 3.1 Des exemples de traits biométriques qui peuvent être utilisés pour l'authentification d'un individu

- **Module d'extraction de caractéristiques:** Les données biométriques sont alors traitées et un ensemble de caractéristiques discriminatoires saillantes, sont extraites pour représenter le trait sous-jacent. Par exemple, la position et l'orientation des minuties dans une image d'empreinte digitale sont extraites par le module d'extraction de caractéristique dans un système biométrique à base d'empreinte digital. Pendant l'inscription, cet ensemble de caractéristiques est stocké dans la base de données et généralement mentionné comme un modèle (gabarit ou Template). Dans la plus part des systèmes de reconnaissance commercialisés, une évaluation de la qualité des données biométriques acquises par le capteur est nécessaire pour déterminer sa pertinence à un traitement ultérieur. Typiquement les données acquises sont soumises à un algorithme d'amélioration de signal pour améliorer sa qualité. Cependant, dans certains cas, la qualité des données peut être si pauvre que l'utilisateur est invité à présenter à nouveau les données biométriques.
- **Le module de correspondance et de prise de décisions :** les caractéristiques extraites sont comparées contre les modèles stockés pour produire le grand nombre de correspondance. Dans un système biométrique à base d'empreintes digitales, le nombre de minuties correspondant entre l'entrée et les ensembles de caractéristiques de modèle, est déterminé et un score de correspondance est rapporté. Ce score peut être influencé par la qualité des données biométriques extraites. Ce module de comparaison et de mise en correspondance encapsule aussi un module de prise de décisions, dans lequel les scores de correspondance sont utilisés soit pour valider une identité revendiquée ou de fournir un classement des identités inscrites afin d'identifier un individu.

Le modèle d'un utilisateur peut être extrait à partir d'un seul échantillon biométrique, ou produit par le traitement de plusieurs échantillons. Ainsi, le modèle de points caractéristiques d'un doigt peut être extrait après l'adaptation de plusieurs échantillons du même doigt. Certains systèmes stockent plusieurs modèles afin de tenir compte des variations intra-classe associées à un utilisateur. Les systèmes de reconnaissance faciale, par exemple, peuvent stocker plusieurs modèles d'un individu, avec chaque modèle correspondant à une pose du visage par rapport à la caméra. Selon l'application, le modèle peut être stocké dans la base de données centrale du système biométrique ou être enregistré sur un jeton (par exemple, la carte à puce) délivré à l'individu.

3.1.1 Vérification vs. Identification

Selon le contexte de l'application, un système biométrique peut fonctionner soit en mode de vérification, soit en mode d'identification ou les deux en même temps. Dans le mode de vérification, le système valide l'identité d'une personne en comparant les données biométriques capturées avec son propre modèle (s) biométrique stocké dans la base de données du système. Dans un tel système, un individu, qui désire être reconnu, revendique une identité, d'habitude via un PIN, un nom d'utilisateur ou une carte à puce et le système conduit une comparaison un-à-un (one-to-one) pour déterminer si la réclamation est vraie ou non (par exemple, "est-ce que ces données biométriques appartiennent à Nabil ? "). La vérification est typiquement utilisée pour empêcher que plusieurs personnes utilisent la même identité.

Dans le mode d'identification, le système reconnaît un individu, par la recherche des modèles de tous les utilisateurs de la base de données pour un appariement. Par conséquent, le système effectue une comparaison un-à-plusieurs (one-to-many) pour établir l'identité d'un individu ou échouer si le sujet ne soit pas inscrit dans la base de données du système. Le but de l'identification est d'empêcher une seule personne d'utiliser des identités multiples. L'identification peut également être utilisée pour la commodité, où l'utilisateur n'est pas censé de réclamer son identité.

3.1.2 Performance d'un système biométrique

Contrairement aux systèmes à base de mot de passe, où une correspondance parfaite entre deux chaînes alphanumériques est nécessaire pour valider l'identité d'un utilisateur, un système biométrique rencontre rarement deux échantillons du trait biométrique d'un utilisateur qui aboutissent exactement au même ensemble de caractéristiques. Ceci est dû aux conditions de détection imparfaite (par exemple, l'empreinte digitale brouillent en raison de la défaillance du capteur), aux changements dans la caractéristique biométrique de l'utilisateur (par exemple, les affections respiratoires ayant un impact sur la reconnaissance du locuteur), aux changements de conditions ambiantes (par exemple, des niveaux d'éclairage contradictoires dans la reconnaissance faciale) et aux variations de l'interaction de l'utilisateur avec le capteur (par exemple, l'iris ocluse ou des empreintes digitales partielles).

Le degré de ressemblance entre deux ensembles de caractéristiques biométriques, est indiqué par un score de similarité. Un score de correspondance de similitude est connu comme un

score authentique s'il est le résultat d'appariement de deux échantillons du même trait biométrique d'un utilisateur. Il est connu comme un score d'imposteur s'il provenait de différents utilisateurs. Un score d'imposteur, qui excède le seuil, aboutit à un faux accepte (un faux match, ou une fausse correspondance). Alors qu'un véritable score, qui tombe au-dessous du seuil, aboutit à un faux rejet (un faux non-match, ou une fausse non-correspondance). Le taux de faux Accepter (FAR : False Accepte Rate) (ou le taux de faux match (RMF)) d'un système biométrique peut donc être défini comme la fraction des scores imposteurs dépassants le seuil. De même, le taux de faux rejet (FRR) (ou, le Taux de faux non-correspondance (FNMR)) d'un système peut être défini comme la fraction des scores authentiques tombant au-dessous du seuil. La régulation de la valeur des changements du FRR et les valeurs de FAR pour un système biométrique donné, est contradictoire, il n'est pas possible de diminuer ces deux erreurs simultanément. Le FAR et FRR à diverses valeurs peuvent être récapitulés en utilisant une courbe (DET) [Martin et al., 1997] qui trace le FRR contre les FAR à différents seuils sur une échelle normale. Le graphe résultant est connu comme une courbe Receiver Operating Characteristic (ROC) [Egan, 1975].

3.1.3 Les caractéristiques biométriques

Un certain nombre de caractéristiques biométriques est utilisés dans des applications diverses. Chaque biométrie a ses avantages et ses inconvénients, par conséquent, le choix d'un trait biométrique pour une application particulière dépend d'une variété de contraintes relative au domaine d'application. Jain et al. [Jain et al., 2006] ont identifié sept facteurs qui déterminent la pertinence physique ou comportementale d'un trait à utiliser dans une application biométrique.

1. Universalité : Chaque individu, ayant accès à l'application, devrait posséder le trait.
2. Unicité: Le caractère utilisé doit être suffisamment différent entre les individus constituant la population.
3. Permanence : La caractéristique biométrique d'un individu doit être suffisamment invariante au cours d'une période de temps par rapport à l'algorithme de correspondance. Un trait, qui change de manière significative au fil du temps, ne constitue pas un élément biométrique utile.
4. mesurabilité: Il devrait être possible d'acquérir et de numériser le trait biométrique à l'aide de dispositifs appropriés qui ne causent pas de gêne notable à l'individu. En

outre, les données brutes acquises doivent se prêter à un traitement afin d'en extraire des ensembles de caractéristiques représentatives.

5. Performance: La précision de la reconnaissance et les ressources nécessaires pour réaliser cette exactitude devraient respecter les contraintes imposées par l'application.
6. Acceptabilité: Les individus dans la population ciblée, qui utiliseront l'application, doivent être prêts à présenter leurs caractéristiques biométriques au système.
7. Mise en échec ou neutralisation : Ceci se réfère à la facilité avec laquelle le trait d'un individu peut être imité.

Aucune unique biométrie ne peut répondre efficacement à toutes les exigences (l'exactitude, l'aspect pratique, le prix) imposées par toutes les applications. Autrement-dit, aucune biométrie n'est idéale, mais un certain nombre d'entre eux est admissible. La pertinence des données biométriques spécifiques à une application est établie en fonction de la nature et des exigences de l'application. Une brève introduction à certaines des caractéristiques biométriques couramment utilisées est donnée ci-dessous:

3.1.3.1 Visage

La reconnaissance faciale est une méthode non-intrusive, les attributs du visage sont probablement les caractéristiques biométriques les plus couramment utilisés par les humains pour reconnaître les uns les autres. Les applications de reconnaissance faciale s'étendent d'un contrôle d'authentification statique à une identification de visage dynamique non contrôlée dans un contexte encombré. Les approches les plus populaires pour la reconnaissance faciale [Jain and Li, 2011] sont basées sur (i) l'emplacement et la forme des attributs du visage, comme les yeux, les sourcils, le nez, les lèvres le menton et leurs relations spatiales, ou (ii) l'analyse globale de l'image de visage qui représente un visage comme une combinaison pondérée d'un certain nombre de visages canoniques. Tandis que la performance d'authentification des systèmes de reconnaissance faciale, qui sont disponibles dans le commerce, est raisonnable [Phillips et al., 2003]. Ces Systèmes imposent un certain nombre de restrictions sur la façon dont les images faciales sont obtenues, exigent souvent un fond fixe et simple avec éclairage contrôlé. Ces systèmes ont aussi des difficultés à apparier les images de visages capturées à partir de deux points de vue différents, dans différentes conditions d'éclairage, et à des moments différents. Il est douteux que le visage lui-même, sans aucune information contextuelle, est une base suffisante pour reconnaître une personne à partir d'un grand nombre d'identités avec un niveau de confiance très élevé. Pour qu'un

système de reconnaissance faciale fonctionne bien dans la pratique, il devrait automatiquement (i) détecter si un visage est présent dans l'image acquise; (ii) localiser le visage s'il y a un; et (iii) reconnaître le visage d'un point de vue général (à savoir, de n'importe quelle pose) dans des conditions ambiantes différentes.

3.1.3.2 Empreintes digitales

Les humains ont utilisé les empreintes digitales pour l'identification personnelle depuis de nombreuses décennies. La précision de cette biométrie reste très efficace [Wilson et al., 2004]. Une empreinte digitale est un motif de crêtes et de vallées sur la surface du bout d'un doigt dont la formation est déterminée au cours des sept premiers mois de développement du fœtus. Il a été empiriquement déterminé que les empreintes digitales de vrais jumeaux sont différentes, ainsi que les impressions sur chaque doigt de la même personne [Maltoni et al., 2009]. La précision des systèmes de reconnaissance d'empreintes digitales actuellement disponibles, est suffisante pour les systèmes d'authentification dans plusieurs applications, particulièrement en médecine légale. Les empreintes digitales multiples d'une personne fournissent des informations supplémentaires pour permettre l'identification à grande échelle impliquant des millions d'identités. Un des problèmes avec les systèmes de reconnaissance d'empreintes digitales à grande échelle est qu'ils nécessitent une grande quantité de ressources informatiques, impliquées particulièrement, dans le mode d'identification. Finalement, les empreintes digitales d'une petite fraction de la population, peuvent être peu convenables pour l'identification automatique à cause des facteurs génétiques, du vieillissement, des raisons environnementales ou professionnelles (par exemple, les travailleurs manuels peuvent avoir un grand nombre de coupures et de contusions sur leurs empreintes digitales qui ne cessent de changer).

3.1.3.3 Géométrie de la main

Les systèmes de reconnaissance de la géométrie de la main sont basés sur un certain nombre de mesures prises à partir de la main humaine, y compris la forme, la taille de la paume, les longueurs et les largeurs des doigts [Zunkel, n.d.]. Les systèmes d'authentification commerciaux basés sur la géométrie de la main ont été installés dans de nombreux endroits à travers le monde. La technique est très simple, relativement facile à utiliser et peu coûteuse. Les facteurs environnementaux tels que la sécheresse ou les anomalies individuelles telles que la peau sèche, ne semblent pas nuire la précision de l'authentification de ces systèmes. Cependant, cette biométrie n'est pas connue être très distinctive, et par conséquent, ces systèmes de reconnaissance ne peuvent pas être mis en œuvre dans des applications de

reconnaitances à grande échelle. En outre, les informations de la géométrie de la main ne peuvent pas être invariantes pendant la période de croissance des enfants. De plus, les bijoux d'un individu (des anneaux) ou des limitations dans la dextérité (l'arthrite), peuvent constituer des défis dans l'extraction de l'information correcte de la géométrie de la main. D'un autre côté, la taille physique d'un système basé sur la géométrie de la main est grande, et elle ne peut pas être intégrée dans certains appareils comme les ordinateurs portables. Cependant, il existe des systèmes d'authentification disponibles qui sont basés sur des mesures de seulement de quelques doigts (typiquement, l'index et le milieu) au lieu de la main entière. Ces dispositifs sont plus petits que ceux utilisés pour la géométrie de la main, mais encore beaucoup plus grande que ceux utilisés pour se procurer certains autres biométries (les empreintes digitales, le visage, la voix).

3.1.3.4 Cachet de la paume (Palmpoint)

La paume de la main humaine est un modèle de crêtes et de vallées tout comme les empreintes digitales. La superficie de la paume est beaucoup plus grande que celle d'un doigt et, par conséquent, les palmprints devraient être encore plus distinctifs que les empreintes digitales [Zhang et al., 2003]. Puisque les scanners des palmprint doivent capturer une grande surface, ils sont plus volumineux et plus coûteux que les capteurs d'empreintes digitales. Les paumes humaines contiennent aussi des traits distinctifs supplémentaires tels que les lignes principales (principal lines) et les rides (wrinkles) qui peuvent être capturés, même avec un scanner de résolution inférieure, ce qui serait moins cher. Finalement, en utilisant un scanner de cachet de la paume de haute résolution, toutes les caractéristiques de la main comme la géométrie, les caractéristiques de vallée et de crêtes (les minuties et les points singuliers tels que deltas), les lignes principales, et les rides peuvent être combinés pour construire un système biométrique fortement précis.

3.1.3.5 Iris

L'iris est la région annulaire de l'œil délimité par la pupille et la sclérotique (blanche de l'œil) de chaque côté. La texture visuelle de l'iris est formée au cours du développement fœtal et se stabilise au cours des deux premières années de la vie. La texture de l'iris est complètement complexe et comporte des informations très distinctives et utiles pour la reconnaissance personnelle [Daugman, 2004a]. Grâce à sa précision et sa vitesse, ces systèmes de reconnaissances à base d'iris actuellement déployés, sont prometteurs et supportent la faisabilité d'identification à grande échelle. Chaque iris est distinctif et même ceux de vrais jumeaux sont différents. Il est possible de détecter des lentilles de contact imprimées avec un

faux iris [Daugman, 2004b]. Bien que les systèmes de reconnaissance basés sur les iris, exigent la participation des utilisateurs et ils sont chers, des nouveaux systèmes plus récents sont devenus plus conviviaux et rentables [Fancourt et al., 2005; Negin et al., 2000]. Tandis que les systèmes d'iris ont une très faible FAR (False Accepter Rate) par rapport à d'autres traits biométriques, le taux de faux rejet (FRR) de ces systèmes peut être assez élevé [Bowyer et al., 2008].

3.1.3.6 Frappe au clavier

On suppose que chaque personne tape sur un clavier d'une manière caractéristique. Cette biométrie ne devrait pas être unique à chaque individu, mais il peut être prévu offrir des informations discriminatoires et suffisantes pour permettre la vérification de l'identité [Monrose and Rubin, 1997]. La dynamique de frappe est une biométrie comportementale; on peut attendre observer de grandes variations intra-classe dans les habitudes de frappe d'une personne en raison de changements dans l'état émotionnel, la position de l'utilisateur par rapport au clavier, le type de clavier utilisé, etc. Les frappes d'une personne peuvent être surveillées discrètement. Ceci permet un contrôle continu de l'identité d'un individu au cours d'une session après que la personne se connecte à l'aide d'un fort identifiant (mot de passe, adresse email).

3.1.3.7 Signature

La façon dont une personne signe son nom, est connue être une caractéristique de cette personne [Lee et al., 1996; Nalwa, 1997]. Bien que les signatures nécessitent un contact avec l'instrument d'écriture et un effort de la part de l'utilisateur, ils ont été acceptés au sein du gouvernement juridique et les transactions commerciales en tant que méthode d'authentification. Avec l'apparition des PDA et Tablet PC, la signature en ligne peut être un choix promoteur. La signature est une biométrie comportementale qui change sur une période de temps et elle est influencée par les conditions physiques et émotionnelles des signataires. Les signatures de certaines personnes varient considérablement: les impressions successives de la même signature sont sensiblement différentes. En outre, les faussaires professionnels peuvent être en mesure de reproduire des signatures qui trompent le système de vérification de la signature.

3.1.3.8 Voix

La voix est une combinaison de caractéristiques biométriques physiques et comportementales [Campbell, 1997]. Les caractéristiques physiques de la voix d'un individu sont basées sur la forme et la taille des appendices (par exemple, des tracts vocaux, la bouche, les cavités nasales et les lèvres qui sont utilisés dans la synthèse du son). Ces caractéristiques physiques de la parole humaine sont invariantes pour un individu, mais l'aspect comportemental a des influences sur la parole au fil du temps en raison de l'âge, des conditions médicales (comme le rhume), d'état émotionnel, etc. La voix n'est pas aussi très distinctive et ne peut pas être appropriée à l'identification à grande échelle. Un système de reconnaissance vocale peut être dépendant du texte (basé sur l'énonciation d'une expression prédéterminée) comme il peut être indépendant du texte où le système reconnaît le locuteur indépendamment du texte. Un système indépendant du texte est plus difficile à concevoir qu'un système dépendant du texte, mais offre plus de protection contre la fraude. Un des inconvénients de la reconnaissance à base de la voix, est que les caractéristiques de la parole sont sensibles à un certain nombre de facteurs comme le bruit de fond. La reconnaissance du locuteur est plus appropriée dans des applications à base de téléphone.

3.1.3.9 Démarche

La démarche (Gait) se réfère à la façon dont une personne marche et l'un des rares traits biométriques qui peuvent être utilisés pour reconnaître les gens à distance. Par conséquent, ce trait est très approprié dans les scénarios de surveillance où l'identité d'un individu peut être subrepticement établie. La plupart des algorithmes de reconnaissance de la démarche tente d'extraire la silhouette humaine afin d'en tirer les attributs temporo-spatiaux d'une personne en mouvement. Par conséquent, la sélection d'un modèle pour représenter le corps humain est essentiel au bon fonctionnement d'un système de reconnaissance de la démarche. Certains algorithmes utilisent le flux optique associé à un ensemble de points mobiles extraits dynamiquement sur le corps humain pour décrire la démarche d'un individu [Nixon et al., 1996]. Ces systèmes offrent également la possibilité de suivre une personne sur une longue période de temps. Cependant, la démarche d'un individu est affectée par plusieurs facteurs, y compris le choix des chaussures, la nature de l'habillement, la surface de marche... etc.

3.2 Application à la reconnaissance d'empreintes digitales

3.2.1 Introduction

La reconnaissance des empreintes digitales est la méthode la plus populaire et la technique biométrique la plus utilisée avec succès pour l'identification personnelle. Son utilisation est due à sa faisabilité, sa distinctivité (même des vrais jumeaux ont des empreintes digitales différentes), sa permanence, son exactitude, sa fiabilité, son acceptabilité et aussi elle est un système biomédical à faible coût [Moses et al., 2011].

L'appariement des empreintes digitales est le processus utilisé pour déterminer si deux ensembles de détails de crêtes d'empreintes digitales proviennent du même doigt. Il existe plusieurs algorithmes qui font l'appariement des empreintes digitales par différentes manières. Certaines impliquent la correspondance entre les points caractéristiques des deux images, tandis que d'autres cherchent des ressemblances dans la structure la plus grande de l'empreinte digitale.

Un des plus grands défis de la reconnaissance des empreintes digitales, est la grande variabilité généralement trouvée entre les différentes impressions du même doigt (grandes variations intra-classe). Les principaux facteurs responsables de ces variations intra-classe sont:

Le déplacement ou la rotation entre les différentes acquisitions, le recouvrement partiel (en particulier dans les capteurs de petite surface), l'état de la peau, le bruit, la distorsion non linéaire en raison de l'élasticité de la peau, la différence de pression sur le capteur et les erreurs d'extraction.

La correspondance d'empreintes digitales reste un problème difficile de reconnaissance de formes en raison de la difficulté d'apparier des empreintes digitales qui sont affectées par un ou plusieurs des facteurs mentionnés ci-dessus [Maltoni et al., 2009].

Une empreinte digitale est modélisée comme un ensemble de minuties. Après une extraction réussie, les minuties sont stockées dans un modèle (Template) qui peut contenir les positions des minuties (x,y) , les directions des minuties (θ) , le type de minuties (bifurcation ou terminaison), et dans certains cas, la qualité des minuties peuvent être envisagée. Les algorithmes d'appariement d'empreintes digitales procèdent en deux étapes; d'abord, ils alignent les empreintes digitales, puis trouvent les correspondances entre les deux empreintes digitales.

L'appariement basé sur les minuties est essentiellement un problème d'appariement de formes de point qui est généralement difficile parce qu'il traite le problème de correspondance. Il peut être tout à fait difficile d'obtenir la correspondance des minuties parce que la nouvelle image peut être soumise à une certaines transformations comme translation, rotation ou même déformation. L'emplacement et les erreurs de direction de minuties détectées, aussi la présence de fausses minuties ou l'absence de véritables minuties, peuvent causer beaucoup d'incongruité dans la correspondance des minuties.

3.2.2 Formulation du problème

Soient A et B représentant successivement le modèle stocké et le modèle d'entrée

$$A = \{m_1, m_2, m_3, \dots, m_m\} \quad \text{Avec } m_i = \{x_i, y_i, \theta_i, \dots\} \text{ et } i = 1, \dots, m$$

$$B = \{m'_1, m'_2, m'_3, \dots, m'_n\} \quad \text{Avec } m'_j = \{x'_j, y'_j, \theta'_j, \dots\} \text{ et } j = 1, \dots, n$$

Où m et n désignent respectivement le nombre de minuties dans A et B. Deux points de minuties m_i et m'_j sont appariés si leurs positions et leurs orientations sont proches, en utilisant la distance spatiale D_s et la différence de direction D_d .

$$D_s(m_i, m'_j) = \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2} \leq r_0$$

$$D_d(m_i, m'_j) = \min(|\theta_i - \theta'_j|, 360 - |\theta_i - \theta'_j|) \leq \theta_0$$

Où r_0 et θ_0 sont les deux paramètres de tolérance qui sont tenus d'accueillir les erreurs inévitables faites par des algorithmes d'extraction de caractéristiques [Noor, 2012].

Soit $\text{map}()$ une fonction qui fait transformée une minutie $m'_j = \{x'_j, y'_j, \theta'_j\}$ à $m''_j = \{x''_j, y''_j, \theta'_j + \alpha\}$ (de B vers B) utilisant une transformation géométrique donnée.

Considérons un déplacement $[\Delta x, \Delta y]$ et une rotation $[\alpha]$ autour de l'origine:

$$\text{map}_{\Delta x, \Delta y, \alpha}(m'_j) = m''_j$$

$$\text{Avec } \begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

Soit $mm()$ une fonction d'indicateur qui renvoie 1 dans le cas où la minutie m''_j et m_i sont bien appariés, et 0 dans le cas contraire. Alors, le problème de mise en correspondance peut être formulé comme suit:

$$\text{maximise}_{\Delta x, \Delta y, \alpha, \epsilon} \sum_{j=1}^n mm(\text{map}_{\Delta x, \Delta y, \alpha}(m'_{p(i)}), m_i)$$

Et qui peut être réduit à : $\text{maximise}_{\Delta x, \Delta y, \alpha, \epsilon} \sum_{j=1}^n mm(m''_j, m_i)$

Où $p(i)$ est une fonction qui détermine l'appariement entre les minuties de A et B, autrement dit, chaque minutie a exactement un homologue dans l'autre empreinte digitale ou n'a pas un homologue du tout.

Les contraintes sont:

- $p(i) = j$: indique que le homologue de $m_i \in A$ est la minutie $m'_j \in B$.
- $p(i) = null$: indique que la minutie $m_i \in A$ n'a pas de homologue dans B.
- Chaque minutie dans A est associé au maximum avec une seule minutie dans B.

En général, $p(i) = j$ ne signifie pas nécessairement que les minuties m_i et m'_j sont appariées d'une manière définitive, mais seulement qu'elles sont la paire la plus probable dans la transformation actuelle. En outre, chaque minutie déjà accouplée doit être marquée pour éviter le ré-couplement.

Malheureusement, en pratique, ni les paramètres d'alignement ni la fonction de correspondance $p()$ ne sont connus à l'avance, par conséquent, la résolution du problème d'adaptation est difficile. Une approche de force brute (évaluer toutes les solutions possibles), est inadmissible parce que le nombre de solutions possibles est exponentiel au nombre de minuties (la fonction $p()$ est plus qu'une permutation en raison des valeurs nulles possibles).

Quelques approches de force brute ont été également proposées dans la littérature :

Huvanandana et al [Huvanandana et al., 2000] ont proposé de quantifier grossièrement l'emplacement des minuties et d'effectuer une recherche exhaustive pour trouver l'alignement optimal.

He et al [He et al., 2003] ont proposé une recherche grossière-à-fine de l'espace des paramètres discrétisés pour déterminer l'alignement, et ont utilisé la distance de Hausdorff à fin d'évaluer la correspondance des minuties.

Par ailleurs, il existe d'autres méthodes plus avancées que les approches par force brute : la géométrie algébrique [Bishnu et al., 2006; Udup et al., 2001], la transformé de Hough [Ratha et al., 1996] et la relaxation [Rosenfeld and Kak, 2014; Ton and Jain, 1989].

3.2.3 Score de similarité

Le score de similarité est calculé en normalisant le nombre de minuties (k) qui sont appariées entre les deux empreintes digitales (correspondance):

$$\text{Score} = \frac{2k}{m + n}$$

Un facteur quantitatif peut être utilisé et qui nécessite un minimum de minuties (12 minuties) qui doivent être trouvées. Dans ce travail, le facteur quantitatif est utilisé dans le calcul du score de similarité.

3.2.4 Notre méthode: du Discret vers le Continu

Notre méthode est une technique d'alignement et de mise en correspondance de points. Dans ce travail, elle est appliquée pour la reconnaissance des empreintes digitales, en considérant celle ci comme un problème de recherche de la plus grande sous-structure commune (LCP). Il s'agit donc de trouver une transformation T , telle que : $T(B) \subset A$.

L'origine de cette difficulté provient clairement de la nature discrète des points à appairier (minuties). Pour surmonter cette difficulté, un passage de cette représentation discrète vers une représentation continue, est adopté dans ce travail. En premier lieu, nous ne essayons pas d'appairier directement les deux nuages de points, nous essayons seulement de surimposer un nuage de points sur la représentation continue du deuxième nuage de points. Après ce prétraitement, des isomorphismes sont alors offertes.

Nous avons vu que le calcul direct de la transformation T est un problème assez compliqué s'il est effectué sur une représentation discrète sans aucune information à priori sur les relations de correspondance entre les deux nuages des points. De notre part, nous proposons une étape intermédiaire, dans laquelle nous calculons une transformation T' dont le rôle est de ramener le nuage des points discret (data set B) sur la représentation continue de l'autre nuage des points (model set A).

L'algorithme propose d'aligner les deux empreintes digitales et de trouver les minuties qui sont couplées (correspondance) entre ces deux empreintes digitales.

Dans un premier temps, les points (minuties) de A sont interpolés en utilisant un polynôme $P(\forall i_{(x,y)} / P(x_i) = y_i)$; si les points de B sont inclus dans A, donc après avoir appliqué la transformation T' sur ces points (les points de B), ils doivent être inclus dans la représentation polynomial de A : $\forall j_{(x',y')} / P(x'_j) = y'_j$.

Les paramètres de la transformation T' sont atteints par la minimisation de la fonction objective QT :

$$QT_{(\Delta x, \Delta y, \alpha)} = \sum_{j=1}^m (P(x'_j) - y'_j)^2$$

Δx et Δy sont des déplacements : horizontal suivant l'axe O_x et vertical suivant l'axe O_y . α est le facteur de rotation autour de l'axe O_z .

Après cette étape d'ajustement des points de B sur la représentation continue de A en utilisant la transformation T' , nous associons chaque minutie de B à la minutie la plus proche dans A en fonction des contraintes mentionnées.

La deuxième étape consiste à calculer les distances et enfin évaluer le score de similarité.

3.2.5 Résultats expérimentaux

La base de données des images d'empreintes digitales a été obtenue à partir de la base UPEK [Upek, 2013]. Ces images ont été prises à partir de 16 personnes (16 classes), chaque classe est composée de 8 versions d'image; ainsi le nombre total des images dans la base de données, est de 128. Dans notre expérience, nous comparons chaque modèle d'empreinte digitale contre toute la base de données. Le cas idéal est quand chaque modèle a un maximum de 8 images similaires dans la base de données (y compris lui-même). Cela fait 1024 tentatives authentiques et 15360 tentatives imposteurs.

Afin d'évaluer notre algorithme de reconnaissance d'empreinte digitale dans une comparaison expérimentale, les algorithmes MJY [Medina-Pérez et al., 2012a], MTK [Medina-Pérez et al., 2009] et MQYW [Medina-Pérez et al., 2012a] qui sont des versions améliorées de JY [Jiang and Yau, 2000], TK [Tico and Kuosmanen, 2003] et QYW [Qi et al., 2005] respectivement, sont inclus et évalués avec leurs paramètres par défaut [Medina-Pérez et al., 2014].

À noter que notre algorithme est basé sur une représentation populaire des minuties, telle que la position, l'orientation et le type des minuties. Tandis que les autres algorithmes utilisent

une représentation assez avancée et spécifique des minuties, telle que les triplets, les angles entre les minuties et d'autres.

Tableau 3.1 montre les différentes valeurs de sensibilité et 1- spécificité pour chaque algorithme utilisé dans les comparaisons expérimentales pour un seuil fixe (12 minuties). A ce stade, nous pouvons constater que notre algorithme obtient des meilleurs résultats par rapport aux autres.

Tableau 3:1 : Sensibilité et 1- Spécificité

Algorithmes	FAR (%)	TAR (%)
DC	0	85,54
MQYW	0 ,004361	85,25
MTK	0	79,10
MJY	0	77,83

Les expériences montrent que l'utilisation du facteur quantitatif (12 minuties), d'une part, diminue légèrement l'efficacité de l'algorithme proposé et d'autre part il a un FAR = 0. Quand au TAR (le système répond correctement à un véritable utilisateur) est de 85,54%.

Enfin, l'architecture de notre algorithme est conçue de telle manière à gérer plus des informations liées à la description des minuties autant que la position et l'orientation. Si les informations pertinentes à la description des minuties sont fournies, alors l'algorithme proposé devient plus efficace et aussi plus rapide par rapport à la version discutée. En fait, toute information supplémentaire est une contrainte supplémentaire qui mènera à minimiser le nombre de concurrents et par conséquent plus d'informations géométriques discriminantes entre les différentes empreintes digitales. D'autre part, l'algorithme utilisé est un outil efficace pour l'étape de la mise en correspondance dans le processus de la reconnaissance d'empreintes digitales (identification / vérification). Ainsi, l'algorithme peut gérer la déformation d'échelle (sous réserve de la poursuite des travaux). Par ailleurs, on remarque que le passage de la représentation discrète vers une représentation continue nous permet de surmonter le problème de minuties manquantes (l'extraction de minuties ne parviennent pas à détecter toutes les minuties existantes), le problème de minuties parasites (l'extracteur de minuties identifie faussement des minuties) et les erreurs de l'emplacement et de l'orientation des minuties.

3.2.6 Conclusion

Dans ce travail [Aqili et al., 2016a] nous avons proposé un algorithme de comparaison d'empreinte digitale à l'aide de l'approche du passage du discret au continu. Les résultats expérimentaux ont été obtenus en utilisant 1024 tentatives authentiques et 15360 tentatives imposteurs. L'algorithme proposé atteint FAR (faux acceptés) = 0% et TAR (vrai acceptés) = 85,54% respectivement à 12 minutes comme seuil.

Au total, les résultats des expériences réalisées à ce seuil (12 minutes), ont démontré que notre algorithme, utilisé de façon brut sans aucun filtrage au préalable, obtient des résultats comparables par rapport à d'autres algorithmes qui sont les plus utilisés dans ce domaine. il aurait possible d'améliorer davantage ces résultats en utilisant des filtrages comme les méthodes dite de triangulation et d'indexation, ainsi une recherche effective utilisant notre approche ne s'opérerait qu'après passage de ces opérations de filtrage.

3.3 Application à la reconnaissance de la signature

3.3.1 Introduction

La vérification des signatures peut être divisée en deux principales catégories en fonction de la méthode d'acquisition de données utilisée: la vérification de la signature en ligne ou la vérification de la signature hors-ligne. Les systèmes en ligne reposent sur des informations dynamiques d'une signature extraite au moment où la signature est effectuée. Les systèmes hors-ligne utilisent les images numérisées d'une signature. Les systèmes en ligne ont des informations supplémentaires telles que le temps, la pression, le stylet en haut et en bas, l'altitude et etc.

Au niveau de la falsification des signatures, il y a trois aspects de falsifications: la falsification aléatoire (pas de connaissances sur la forme de la signature ou le nom du signataire), la falsification simple (sachant que le nom du signataire) et la falsification qualifiée (en regardant l'échantillon de signature originale). La plupart des systèmes développés se concentrent sur la vérification plus que l'identification en raison de l'utilisation quotidienne des systèmes de vérification de la signature [Rashidi et al., 2012].

Les méthodes de vérification peuvent être décrites selon trois catégories:

- i) Template Matching, où la signature d'entrée est comparée à des modèles de signatures stockées en se basant sur des techniques telles que DTW (Dynamics Time Wrapping) [Faundez-Zanuy, 2007].
- ii) Approche statistique, où la décision est fondée sur les classificateurs de réseaux de neurones [Jain and Gangrade, 2013], de modèle de Markov caché (HMM) [Fierrez et al., 2007], etc.
- iii) Approches structurelles, qui sont liées à la représentation structurelle de la signature et qui sont comparées en utilisant des graphes [Barkoula et al., 2013].

3.3.2 Prétraitement et extraction des caractéristiques

3.3.2.1 Prétraitement

Généralement les systèmes de reconnaissance de la signature en ligne utilisent des tablettes avec des stylets numériques, et qui fournissent des séries de données tridimensionnelles (3D) pour représenter une signature : la position du stylet ($x(t)$, $y(t)$) et la pression du stylet $p(t)$.

Sachant qu'il y a d'autres données fournies par la tablette pour représenter une signature, dans ce travail nous utilisons seulement les trois données citées ci-dessus.

Pour résoudre le problème de différence des tailles des signatures, les signaux $x(t)$ et $y(t)$ doivent être normalisés. Une normalisation géométrique composée d'une normalisation de la position et d'une normalisation de la rotation. Pour celle de la position, soit on aligne le centre de masse des différentes signatures, soit on normalise les composantes horizontale et verticale à une taille standard. Tandis que la normalisation de la rotation, elle consiste à aligner l'angle tangent moyenne à la trajectoire des différentes signatures.

Un autre type de prétraitement consiste à supprimer les virtuels pen-up (stylet en bas ou en haut). La principale raison de ces virtuels pen-up, est la pression tout au long du processus de signature.

Si les deux signatures qui doivent être jumelées, ont été normalisées à une fenêtre fixe, certaines informations importantes seront perdues en particulier dans les comparaisons intra-session. Inversement, la normalisation de la rotation pourrait augmenter la rapidité d'adaptation. Cependant, l'algorithme de comparaison utilisé dans ce travail peut gérer les paramètres de translation et de rotation. Pour cette raison, aucune des techniques de prétraitements mentionnées ont été appliquées sur les signaux. D'autre part, la raison la plus particulière de ne pas utiliser la normalisation des virtuels pen-up, est que les segments de signature, sont utilisés dans l'étude des déformations des formes.

3.3.2.2 Extraction des caractéristiques

L'extraction des caractéristiques est une étape importante dans les systèmes de reconnaissance. Deux types de caractéristiques peuvent être extraits à partir d'une signature:

- i) selon les fonctions : la signature est caractérisée par une fonction du temps dont la valeur est composée de la position, de la vitesse, de la pression, etc.
- ii) sur la base des paramètres : la signature est représentée en tant que vecteur d'éléments. Ces paramètres peuvent être locaux ou globaux. Les paramètres locaux dépendent de chaque point de la signature. Tandis que les paramètres globaux présentent l'ensemble de la signature et du processus de signature, comme la durée totale, la pression moyenne, la vitesse moyenne, etc.

Tableau 3:2 : Exemple de caractéristiques communes

	Fonction de base	fonction étendue	
Local	coordonnées	Vitesse	accélération
	X(t)	V _x (t)	A _x (t)
	Y(t)	V _y (t)	A _y (t)
	P(t) pression --- Ψ altitude --- Φ azimuth		
Global	position absolue --- vitesse absolue ---- accélération absolue		

Dans ce travail, la représentation de la signature considérée est basée sur les trois séquences de temps suivantes : la position horizontale x_n , la position verticale y_n et le signal de pression p_n . La valeur $n = 1, \dots, N$ est l'indice temporel discret obtenu dans la phase d'acquisition et N est la durée de la signature dans les unités d'échantillonnage. En conséquent, le jeu de fonctions de base, se compose de x_n, y_n et p_n .

Les dérivées (première et seconde) de la fonction du temps, sont des paramètres discriminatoires en cas de vérification. En fait, l'ensemble de la fonction de base peut être étendu à d'autres séquences telles que le chemin de la tangente des angles, le chemin de la magnitude de la vitesse, le log de la courbure et la magnitude de l'accélération totale. Dans ce travail, seules les fonctions de base sont utilisées. Alors, si ces fonctions sont bien comportées, certainement ses fonctions étendues (les drives première et seconde) aussi doivent l'être. Dans tous les cas, les dérivés de temps discrets peuvent être calculés en utilisant la régression du second ordre. L'algorithme utilisé dans ce travail utilise une représentation continue de ces données discrètes, et par conséquent le calcul direct des dérivées est bien assuré.

3.3.3 Algorithmes de mise en correspondance et décision

3.3.3.1 Dynamic Time Warping

L'ensemble de données est une séquence discrète de la fonction temporelle. En général, les signaux à correspondre ont des durées différentes, la longueur de temps peut être unifiée et l'algorithme DTW peut être utilisé pour trouver les correspondances entre les deux signaux.

Soit A et B deux signaux de longueurs N et M, respectivement. Pour trouver les points correspondants des deux signaux utilisant DTW, une matrice de taille $N \times M$ est nécessaire. Après cela, un trajet de déformation au quel la somme des éléments de cette matrice par rapport au premier élément jusqu'au dernier élément, doit être calculé, puis le chemin minimal est choisi. Ce dernier représente les détails de la mise en correspondance entre les deux signaux.

Tout d'abord, si les deux signaux sont très longs (c'est un cas fréquent dans le processus de la signature), la matrice nécessaire de DTW augmente avec la longueur des deux signaux, ce qui a une grande influence sur la mémoire et la rapidité. En outre, la nécessité de l'unification de la longueur des deux signaux, diminue la capacité de l'algorithme de pouvoir différencier entre les signatures authentiques et falsifiées. En effet, l'unification des longueurs est un ré-échantillonnage de la signature de telle sorte que les points redondants seront enlevés. Toutefois, cette étape peut entraîner une perte significative des points caractéristiques.

3.3.3.2 Notre méthode: du Discret vers le Continu

Pour éviter les contraintes et les inconvénients mentionnés ci-dessus, l'approche par passage du discret au continu a été utilisée pour la vérification des signatures.

La puissance de cet algorithme réside dans la transformation du problème de l'appariement de deux formes discrètes à un problème de superposition d'un signal de points discret sur une représentation continue. Cet objectif est atteint par l'interpolation du signal de formation (signal de modèle stocké dans la base de données). Dans l'étape suivante; il surimpose l'autre signal (signal de test/ les données discrètes) dans la représentation polynomiale continue du premier signal. En d'autres termes, supposons que T est une matrice de transformation en 2D, composée d'une translation et d'une rotation. Le problème de mise en correspondance de deux signaux A et B est réduit à un problème de trouver les meilleurs paramètres de la matrice de transformation T telle que $T(B) \subset A$.

C'est un algorithme itératif basé sur la méthode de programmation dynamique qui comporte une minimisation d'une fonction de coût. Au stade final de cet algorithme la RMS est calculée, elle représente le degré de similitude entre les deux signaux A et B. En fait, cette valeur pourrait être utilisée comme une métrique.

Dans les méthodes fonctionnelles, un motif de signature est défini en fonction du temps, ses fonctions de base et ainsi ses fonctions étendues sont comparées au niveau local. Ces comparaisons peuvent être effectuées point par point ou segment par segment. L'algorithme utilisé calcule la valeur de RMS et le nombre de points mis en correspondance entre les deux signaux à l'aide d'un seuil prédéfini.

L'avantage de mise en correspondance du point-à-point est clairement démontré lorsque la fonction prolongée est utilisée, parce que les points semblables dans la fonction de base devront avoir un comportement plus similaire à l'étape prolongée.

3.3.3.3 Décision

Au stade de la formation (apprentissage) du système, un modèle de signature est dérivé ou l'ensemble des échantillons d'apprentissage sont utilisées dans l'étape d'appariement. Dans le présent document, le but de l'apprentissage est de déterminer un seuil de décision. Pour ce fait, cinq véritables signatures de chaque personne sont utilisées et les similarités entre eux sont calculées.

Les RMS moyennes sont utilisées pour déterminer le seuil de décision. En effet, pour confirmer ou rejeter une signature qui est prétendue être de la personne ID_j , si la condition $RMS_{ID_j} \leq RMS_{moy_{ID_j}}$ est remplie, alors la signature d'entrée sera confirmée. Sinon, elle sera rejetée.

Dans le procédé d'apprentissage, les cinq premières signatures (S_1, S_2, S_3, S_4 et S_5) de chaque client sont utilisées. Les valeurs des RMS sont calculées entre eux. Le tableau 3.3 donne un exemple de la procédure du calcul des RMS concernant la première personne (ID_1)

Tableau 3:3 : RMS pour l'utilisateur ID_1

	S_1	S_2	S_3	S_4	S_5	RMS
S_1	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	RMS_{r_1}
S_2	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}	RMS_{r_2}
S_3	r_{31}	r_{32}	r_{33}	r_{34}	r_{35}	RMS_{r_3}
S_4	r_{41}	r_{42}	r_{43}	r_{44}	r_{45}	RMS_{r_4}
S_5	r_{51}	r_{52}	r_{53}	r_{54}	r_{55}	RMS_{r_5}

Où r_{ij} est la valeur de RMS entre la signature S_i et la signature S_j :

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Avec x_i et y_i sont les coordonnées cartésiennes.

$$RMS_{r_1} = (r_{11}, r_{12}, r_{13}, r_{14}, r_{15})/5$$

$$RMS_m = (RMS_{r_1}, RMS_{r_2}, RMS_{r_3}, RMS_{r_4}, RMS_{r_5})/5$$

Pour chaque utilisateur ID_i , la valeur RMS_m est calculée et sa signature d'entraînement, ayant la RMS_r la plus proche de RMS_m , est utilisée comme signature de référence dans la phase de test et de décision, la RMS_m est utilisée comme un seuil dans l'étape de décision. Cette étape a pour objective la sélection d'une seule signature représentative de chaque utilisateur dans le but d'éviter la comparaison de la signature du test avec toutes les signatures de référence. Dans la littérature les protocoles d'évaluations exigent l'utilisation de cinq signatures comme référence.

3.3.4 Résultats

Les bases de données réelles ont des avantages, comme la représentation de la signature réelle de l'homme, mais elles ont des restrictions comme la quantité limitée de données, les problèmes de confidentialité, etc. Par contre, les bases de données de signatures de synthèse n'ont pas les restrictions mentionnées ci-dessus, en revanche, elles n'ont pas l'avantage d'avoir des situations du monde réel et de vraies signatures.

En général, ces bases de données synthétiques ont une bonne approche pour simuler des véritables signatures. La base de données ATVS-SSig DB 1 est l'une des bases de données synthétiques utilisées dans la vérification. La figure 3.3 illustre quelques échantillons de signature, la première rangée de la figure représente les signatures, la deuxième, troisième et quatrième ligne représentent les signaux $x(t)$, $y(t)$ et $p(t)$ respectivement.

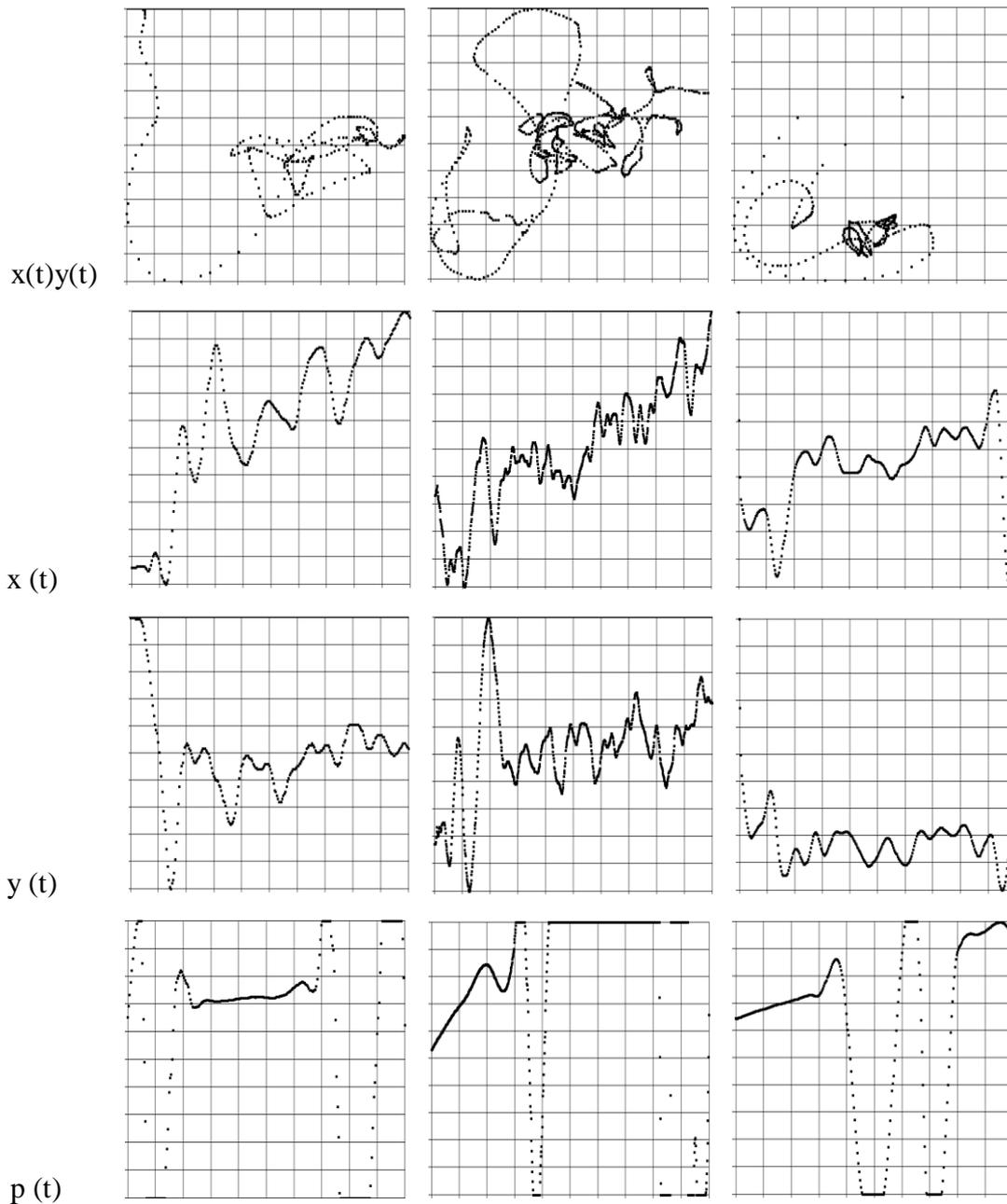


Figure 3.2 Echantillons de signatures de la base de données ATVS

3.3.4.1 La base de donnée ATVS-SSig DB 1

Les échantillons sont générés en modifiant directement les fonctions de temps de la signature maître selon le procédé décrit dans [Galbally et al., 2012]. Elle contient 25 signatures de 350 utilisateurs. Les 5 premières signatures suivent une variation intra-session et les 20 signatures restantes suivent une variation intersession. La figure 3.4 présente un échantillon de trois signatures suivant une variabilité intra-session. Ces signatures représentent une seule personne.

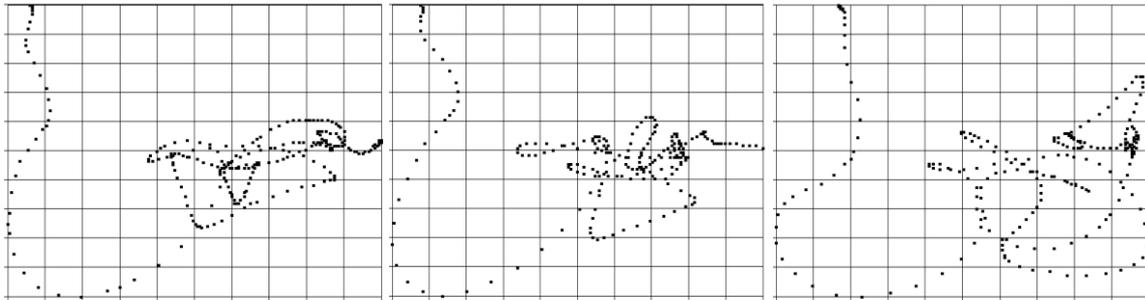


Figure 3.3 Les échantillons de signature suivant une variabilité intersession d'une personne

3.3.4.2 Résultats expérimentaux

Notre premier objectif était de trouver une combinaison appropriée entre les signatures d'apprentissages pour effectuer un bon régime à base de référence dans la phase du test. Ce choix a été fait dans le but d'améliorer le système pour reconnaître les contrefaçons, tolérer les variances interpersonnelles et aussi identifier la signature dans les domaines d'application qui exigent des systèmes de reconnaissance de signature plus rapide.

Afin d'évaluer l'efficacité du système, deux expériences ont été réalisées :

La première expérience montre un processus de vérification. La figure 3.5 illustre les résultats de deux ensembles. Le premier ensemble contient 50 personnes et le deuxième contient 100 personnes. L'objectif de cette expérience est de quantifier l'influence de la taille de la base de données sur le seuil.

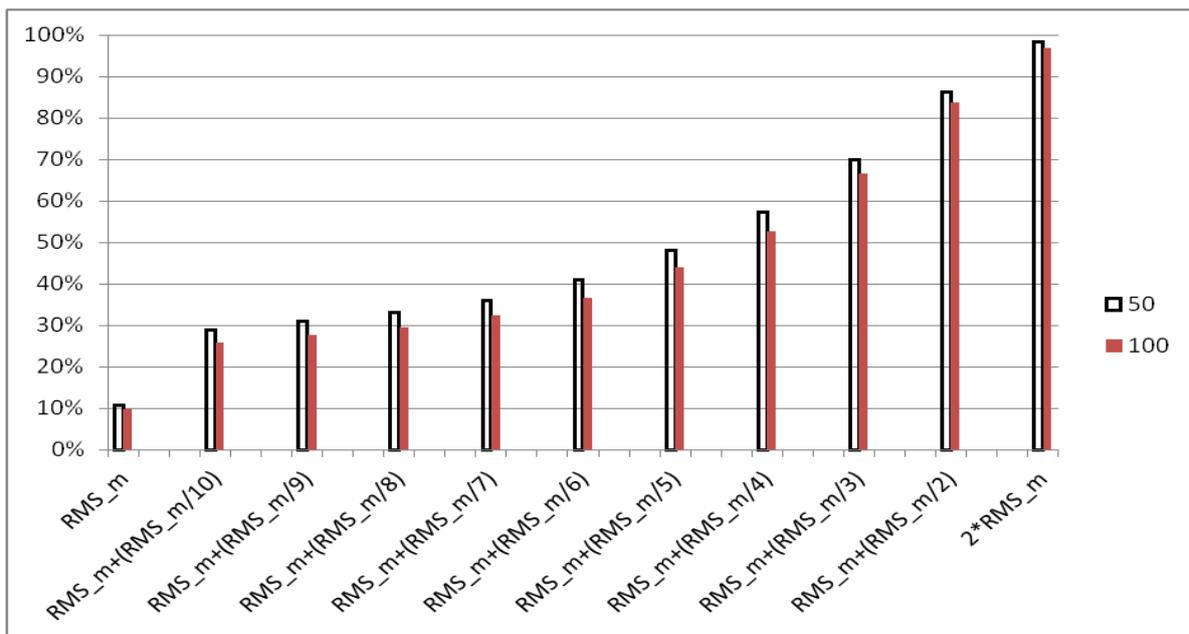


Figure 3.4 Influence de la taille de la base de données sur le seuil

La deuxième expérience a été réalisée sur un ensemble de 100 personnes. L'ensemble d'apprentissage contient cinq signatures réelles de chaque utilisateur, une seule d'entre eux est utilisée comme référence dans la phase du test. L'ensemble de test se compose de 20 échantillons, conduisant ainsi à 2000 scores authentiques. Le score des imposteurs est obtenu en utilisant une signature de chacun des utilisateurs restants, ce qui fait 9900 scores imposteurs. Ces scores sont utilisés pour calculer la FAR (les imposteurs acceptés) et FRR (les vrais utilisateurs refusés) du système. La figure 3.6 montre les résultats relatifs à la deuxième expérience.

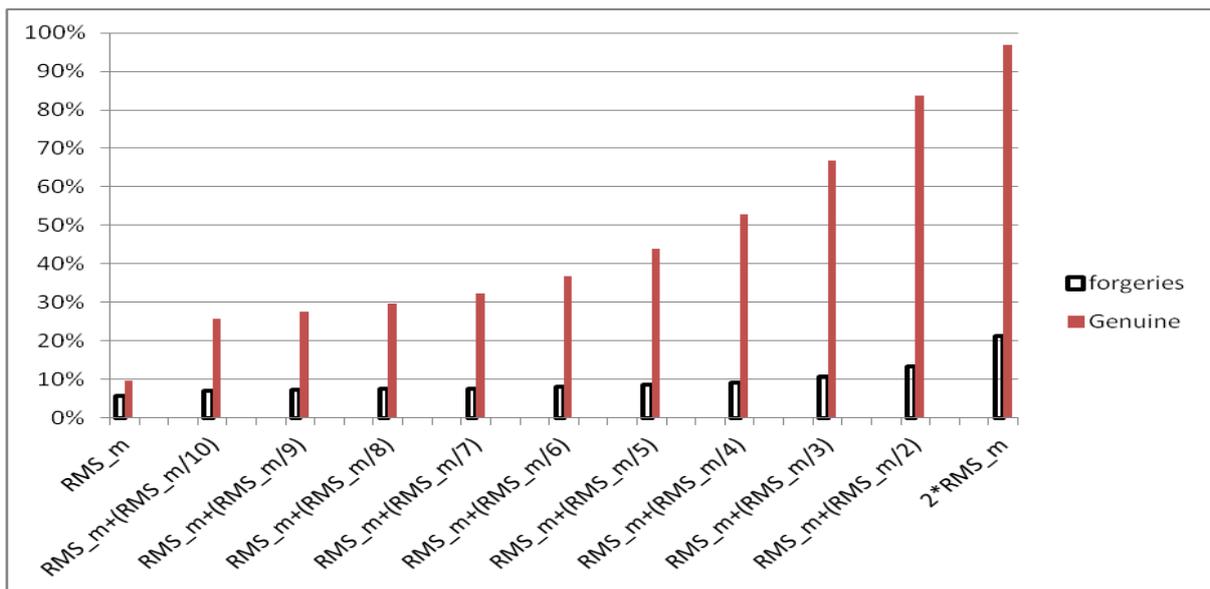


Figure 3.5 Taux de reconnaissance de signatures authentiques et falsifiées

La moyenne quadratique (RMS) est une règle qui mesure l'amplitude moyenne de l'erreur. Dans ce cas, la performance globale du système est acceptable. Il a tendance à rejeter moins de 3% de vraies signatures et accepter environ 20% de fausses signatures.

La RMS est utilisée dans ce travail pour sélectionner des points très similaires entre les deux signatures. Il est évident que ces points potentiels utilisés dans la première étape d'apprentissage, méritent une étude rigoureuse appliquée sur ses fonctions étendues. Il est communément admis que l'information dynamique de signature est difficile à falsifier et les contrefaçons peuvent être distinguées.

3.3.5 Conclusion

Dans notre système de reconnaissance [Aqili et al., 2016b], nous avons utilisé l'approche du discret au continu comme une nouvelle alternative pour identifier les personnes en fonction de leurs signatures en ligne. En se basant sur les caractéristiques (positions et pression) qui sont utilisées dans l'étape d'extraction. L'algorithme du discret au continu est appliqué sur les caractéristiques dynamiques basées sur la RMS pour fixer un seuil utilisé durant la phase de décision. Les expériences ont été réalisées en utilisant la base de données ATVS. Les résultats présentent un bon taux d'appariement.

Finalement, nous notons que si les fonctions étendues de la fonction de base sont incluses dans l'algorithme d'appariement, le taux de reconnaissance sera considérablement augmenté.

3.4 Conclusion

Ce chapitre avait pour objectif de présenter l'efficacité de notre algorithme dans des applications réelles dans le domaine de la biométrie. L'efficacité des techniques de reconnaissances biométriques (entre autres) est essentiellement dépendante de l'algorithme de comparaison utilisé. Comme notre algorithme propose une stratégie incluant toutes les informations géométriques possibles, ceci lui procure son indépendance au domaine d'application, son efficacité en termes de résultats et un temps d'exécution réduit. C'est un outil puissant pour accomplir la tâche de comparaison et de mise en correspondance dans le processus de la reconnaissance et l'identification des individus.

En résumé nous pouvons annoncer que les performances, de notre algorithme utilisé dans son états "brute", aussi bien dans le domaine de la reconnaissance d'empreintes digitales que celui de la reconnaissance de la signature, sont plus qu'honorables. Ces résultats pourraient être améliorées par des adaptations plus spécifiques dans le choix des outils numériques ainsi que l'introduction des méthodes de filtrage et de prétraitements comme ceux préconisés par la triangulation dans le traitement d'empreintes digitales ou encore comme l'utilisation des fonctions étendues (drivés première et seconde) dans le cas des signatures.

Conclusion générale et perspectives

La reconnaissance de formes est un domaine relevant du champ de l'intelligence artificiel qui ne cesse de trouver, tous les jours, de nouvelles applications (biométrie, vision par ordinateur, etc.). Contrairement à ce que nous pouvions penser, en termes d'algorithme, beaucoup de choses restent à améliorer.

Notre travail ne prétend en aucun cas, proposer une solution définitive à ces problèmes. Néanmoins, nous pensons pouvoir offrir d'une certaine manière un outil capable d'être adapté à différentes situations dans le domaine de la reconnaissance de formes. Les temps de réponse générés par notre algorithme restent parmi les meilleurs et cela est dû essentiellement au traitement de la globalité de la sous-forme contrairement à la majorité des méthodes qui traitent la question point par point.

L'adaptation de notre algorithme se base essentiellement sur la définition quadratique de la transformation et de la métrique. Il a été aussi utilisé dans le domaine de la reconnaissance de la parole en choisissant une transformation rigide (2D) et une métrique de type « Cosine » [A. Maazouzi et al., 2016; A.-E. Maazouzi et al., 2016; Maazouzi et al., 2015]. Il a donné de bons résultats.

Les résultats obtenus selon les différentes applications sont intéressantes malgré un minimum d'adaptations. Cela nous permet de penser que des performances meilleures pourraient être obtenues avec de meilleures adaptations tant au niveau approche (triangulation dans le cas des empreintes [Medina-Pérez et al., 2012b]) qu'au niveau outils (l'algorithme de résolution de QT [Es-Sadek, 2009]).

Néanmoins, nous ne pouvons nier qu'une difficulté majeure réside d'une part dans la compréhension de l'algorithme et dans sa mise en œuvre d'autre part.

Enfin, nous ne pouvons-nous empêcher de penser qu'en dehors du domaine de la reconnaissance de forme, notre approche pourrait trouver des utilisations dans d'autres disciplines (économie, industrie...) dont la problématique consiste à faire passer un système d'un état « A » défini dans un espace nD vers un autre état « B » moyennant un procédé donné. Pour se faire, il serait nécessaire de définir la représentation quadratique de la transformation paramétrée « T » ainsi que la métrique à utiliser. La fonction économique QT obtenue pourrait, dans sa minimisation, prendre en considération des contraintes sur les paramètres de « T » à définir.

Références

- Aggarwal, J.K., Cai, Q., Liao, W., Sabata, B., 1994. Articulated and elastic non-rigid motion: A review, in: *Motion of Non-Rigid and Articulated Objects*, 1994., Proceedings of the 1994 IEEE Workshop on. IEEE, pp. 2–14.
- Akutsu, T., 1998. On determining the congruence of point sets in d dimensions. *Comput. Geom.* 9, 247–256.
- Alt, H., Behrends, B., Blömer, J., 1995. Approximate matching of polygonal shapes. *Ann. Math. Artif. Intell.* 13, 251–265.
- Alt, H., Guibas, L.J., 1999. Discrete geometric shapes: Matching, interpolation, and approximation. *Handb. Comput. Geom.* 1, 121–153.
- Alt, H., Mehlhorn, K., Wagener, H., Welzl, E., 1988. Congruence, similarity, and symmetries of geometric objects. *Discrete Comput. Geom.* 3, 237–256.
- Ambühl, C., Chakraborty, S., Gärtner, B., 2000. Computing largest common point sets under approximate congruence, in: *European Symposium on Algorithms*. Springer, pp. 52–64.
- Aqili, N., Maazouzi, A., Raji, M., Jilbab, A., Chaouki, S., Hammouch, A., 2016a. Fingerprint matching algorithm based on discrete to continuous approach, in: *2016 International Conference on Electrical and Information Technologies (ICEIT)*. IEEE, pp. 414–417.
- Aqili, N., Maazouzi, A., Raji, M., Jilbab, A., Chaouki, S., Hammouch, A., 2016b. On-line signature verification using point pattern matching algorithm, in: *2016 International Conference on Electrical and Information Technologies (ICEIT)*. IEEE, pp. 410–413.
- Aqili, N., Raji, M., Jilbab, A., Chaouki, S., Hammouch, A., Aqili, N., Raji, M., Jilbab, A., Chaouki, S., Hammouch, A., 2016c. PPM Translation, Rotation and Scale in D -Dimensional Space by the Discrete to Continuous Approach. *Int. Rev. Comput. Softw. IRECOS* 11, 270–276. doi:10.15866/irecos.v11i3.8746
- Arun, K.S., Huang, T.S., Blostein, S.D., 1987. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* 698–700.

- Arzoumanian, Z., Holmberg, J., Norman, B., 2005. An astronomical pattern-matching algorithm for computer-aided identification of whale sharks *Rhincodon typus*. *J. Appl. Ecol.* 42, 999–1011.
- Atallah, M.J., 1984. Checking similarity of planar figures. *Int. J. Comput. Inf. Sci.* 13, 279–290.
- Atkinson, M.D., 1987. An optimal algorithm for geometrical congruence. *J. Algorithms* 8, 159–172.
- Ballard, D.H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* 13, 111–122.
- Bardinet, E., Cohen, L.D., Ayache, N., 1998. A parametric deformable model to fit unstructured 3D data. *Comput. Vis. Image Underst.* 71, 39–54.
- Barkoula, K., Economou, G., Fotopoulos, S., 2013. Online signature verification based on signatures turning angle representation using longest common subsequence matching. *Int. J. Doc. Anal. Recognit. IJDAR* 16, 261–272.
- Baumbach, T., Ortmann, W., 1999. Shift detection by restoration-demonstrated by signal based point pattern matching, in: *Image Analysis and Processing, 1999. Proceedings. International Conference on. IEEE*, pp. 310–315.
- Besl, P.J., McKay, N.D., 1992. Method for registration of 3-D shapes, in: *Robotics-DL Tentative. International Society for Optics and Photonics*, pp. 586–606.
- Bishnu, A., Das, S., Nandy, S.C., Bhattacharya, B.B., 2006. Simple algorithms for partial point set pattern matching under rigid motion. *Pattern Recognit.* 39, 1662–1671.
- Bishop, M.J., Thompson, E.A., 1986. Maximum likelihood alignment of DNA sequences. *J. Mol. Biol.* 190, 159–165.
- Bourdon, O., Medioni, G., 1990. Object recognition using geometric hashing on the connection machine, in: *Pattern Recognition, 1990. Proceedings., 10th International Conference on. IEEE*, pp. 596–600.
- Bowyer, K.W., Hollingsworth, K., Flynn, P.J., 2008. Image understanding for iris biometrics: A survey. *Comput. Vis. Image Underst.* 110, 281–307.
- Brandstädt, A., Spinrad, J.P., others, 1999. *Graph classes: a survey.* Siam.

- Bra\ss, P., Knauer, C., 2000. Testing the congruence of d-dimensional point sets, in: Proceedings of the Sixteenth Annual Symposium on Computational Geometry. ACM, pp. 310–314.
- Brin, S., 1995. Near neighbor search in large metric spaces.
- Brown, L.G., 1992. A survey of image registration techniques. ACM Comput. Surv. CSUR 24, 325–376.
- Bunke, H., 2000. Graph matching: Theoretical foundations, algorithms, and applications, in: Proc. Vision Interface. pp. 82–88.
- Campbell, J.P., 1997. Speaker recognition: a tutorial. Proc. IEEE 85, 1437–1462.
- Carcassoni, M., Hancock, E.R., 2000. Point pattern matching with robust spectral correspondence, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. IEEE, pp. 649–655.
- Chen, C.-C., 1993. Improved moment invariants for shape discrimination. Pattern Recognit. 26, 683–686.
- Chew, L.P., Dor, D., Efrat, A., Kedem, K., 1995. Geometric pattern matching in d-dimensional space, in: European Symposium on Algorithms. Springer, pp. 264–279.
- Chew, L.P., Goodrich, M.T., Huttenlocher, D.P., Kedem, K., Kleinberg, J.M., Kravets, D., 1997. Geometric pattern matching under Euclidean motion. Comput. Geom. 7, 113–124.
- Chew, L.P., Kedem, K., 1992. Improvements on geometric pattern matching problems, in: Scandinavian Workshop on Algorithm Theory. Springer, pp. 318–325.
- Chui, H., Rangarajan, A., 2003. A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. 89, 114–141.
- Chui, H., Rangarajan, A., 2000. A new algorithm for non-rigid point matching, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. IEEE, pp. 44–51.
- Clarkson, K.L., 1999. Nearest neighbor queries in metric spaces. Discrete Comput. Geom. 22, 63–93.

- Conte, D., Foggia, P., Sansone, C., Vento, M., 2004. Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.* 18, 265–298.
- Cross, A.D., Hancock, E.R., 1998. Graph matching with a dual-step EM algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 1236–1253.
- Daugman, J., 2004a. How iris recognition works. *IEEE Trans. Circuits Syst. Video Technol.* 14, 21–30.
- Daugman, J., 2004b. Recognising persons by their iris patterns, in: *Advances in Biometric Person Authentication*. Springer, pp. 5–25.
- Duda, R., Hart, P., 2000. and D. Stork, *Pattern classification*. Wiley-interscience.
- Edgar, R.C., 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32, 1792–1797.
- Efrat, A., Itai, A., 1996. Improvements on bottleneck matching and related problems using geometry, in: *Proceedings of the Twelfth Annual Symposium on Computational Geometry*. ACM, pp. 301–310.
- Efrat, A., Katz, M.J., 1996. Computing fair and bottleneck matchings in geometric graphs, in: *International Symposium on Algorithms and Computation*. Springer, pp. 115–125.
- Egan, J.P., 1975. Signal detection theory and $\text{\$}\{\text{\$ROC}\}\text{\$}$ analysis.
- Eggert, D.W., Lorusso, A., Fisher, R.B., 1997. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vis. Appl.* 9, 272–290.
- Es-Sadek, M.Z., 2009. Contribution à l’optimisation globale: approche déterministe et stochastique et application. Rouen, INSA.
- Fancourt, C., Bogoni, L., Hanna, K., Guo, Y., Wildes, R., Takahashi, N., Jain, U., 2005. Iris recognition at a distance, in: *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, pp. 1–13.
- Far, A.B., 2005. Analyse multi-images: Application à l’extraction contrôlés d’indices images et à la détermination de descriptions scéniques. Université Louis Pasteur (Strasbourg).

- Faugeras, O., 1993. Three-dimensional computer vision: a geometric viewpoint. MIT press.
- Faundez-Zanuy, M., 2007. On-line signature recognition based on VQ-DTW. *Pattern Recognit.* 40, 981–992.
- Feldmar, J., Ayache, N., 1996. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. Comput. Vis.* 18, 99–119.
- Fierrez, J., Ortega-Garcia, J., Ramos, D., Gonzalez-Rodriguez, J., 2007. HMM-based on-line signature verification: Feature extraction and signature modeling. *Pattern Recognit. Lett.* 28, 2325–2334.
- Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395.
- FUKUNAGA, R., 1990. *Statistical pattern recognition.*
- Gaede, V., Günther, O., 1998. Multidimensional access methods. *ACM Comput. Surv. CSUR* 30, 170–231.
- Galbally, J., Plamondon, R., Fierrez, J., Ortega-Garcia, J., 2012. Synthetic on-line signature generation. Part I: Methodology and algorithms. *Pattern Recognit.* 45, 2610–2621.
- Gavrilov, M., Indyk, P., Motwani, R., Venkatasubramanian, S., 1999. Geometric pattern matching: A performance study, in: *Proceedings of the Fifteenth Annual Symposium on Computational Geometry.* ACM, pp. 79–85.
- Godil, A., Dutagaci, H., Akgül, C.B., Axenopoulos, A., Bustos, B., Chaouch, M., Daras, P., Furuya, T., Kreft, S., Lian, Z., others, 2009. SHREC'09 Track: Generic shape retrieval, in: *3DOR.* pp. 61–68.
- Gonzalez, R.C., Thomason, M.G., 1978. *Syntactic pattern recognition: An introduction.*
- Goodrich, M.T., Mitchell, J.S., Orletsky, M.W., 1999. Approximate geometric pattern matching under rigid motions. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 371–379.
- Goshtasby, A., Stockman, G.C., 1985. Point pattern matching using convex hull edges. *IEEE Trans. Syst. Man Cybern.* 631–637.

- Griffin, P.M., Alexopoulos, C., 1989. Point pattern matching using centroid bounding. *IEEE Trans. Syst. Man Cybern.* 19, 1274–1276.
- Grimson, W.E.L., Huttenlocher, D.P., 1990. On the sensitivity of geometric hashing, in: *Computer Vision, 1990. Proceedings, Third International Conference on.* IEEE, pp. 334–338.
- Grimson, W.E.L., Lozano-Perez, T., 1987. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Anal. Mach. Intell.* 469–482.
- Hagedoorn, M., 2001. Pattern matching using similarity measures.
- Hagedoorn, M., Veltkamp, R., 1997. A general method for partial point set matching, in: *Proceedings of the Thirteenth Annual Symposium on Computational Geometry.* ACM, pp. 406–408.
- Hall, P.A., Dowling, G.R., 1980. Approximate string matching. *ACM Comput. Surv. CSUR* 12, 381–402.
- He, Y., Tian, J., Ren, Q., Yang, X., 2003. Maximum-likelihood deformation analysis of different-sized fingerprints, in: *International Conference on Audio-and Video-Based Biometric Person Authentication.* Springer, pp. 421–428.
- Hel-Or, Y., Hel-Or, H., David, E., 2014. Matching by tone mapping: Photometric invariant template matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 317–330.
- Hérisson, J., Payen, G., Gherbi, R., 2007. A 3D pattern matching algorithm for DNA sequences. *Bioinformatics* 23, 680–686.
- Holstein, H., Li, B., 2002. Low density feature point matching for articulated pose identification.
- Hong, J., Tan, X., 1988. A new approach to point pattern matching, in: *Pattern Recognition, 1988., 9th International Conference on.* IEEE, pp. 82–84.
- Horaud, R., Monga, O., 1995. *Vision par ordinateur: outils fondamentaux.* Editions Hermès.
- Huber, P.J., 2011. *Robust statistics.* Springer.

- Huttenlocher, D.P., Kedem, K., Sharir, M., 1993a. The upper envelope of Voronoi surfaces and its applications. *Discrete Comput. Geom.* 9, 267–291.
- Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1993b. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 850–863.
- Huttenlocher, D.P., Ullman, S., 1990. Recognizing solid objects by alignment with an image. *Int. J. Comput. Vis.* 5, 195–212.
- Huvanandana, S., Kim, C., Hwang, J.-N., 2000. Reliable and fast fingerprint identification for security applications, in: *Image Processing, 2000. Proceedings. 2000 International Conference on.* IEEE, pp. 503–506.
- Isenor, D.K., Zaky, S.G., 1986. Fingerprint identification using graph matching. *Pattern Recognit.* 19, 113–122.
- Jain, A., Bolle, R., Pankanti, S., 2006. *Biometrics: personal identification in networked society.* Springer Science & Business Media.
- Jain, A., Flynn, P., Ross, A.A., 2007. *Handbook of biometrics.* Springer Science & Business Media.
- Jain, A., Maltoni, D., Maio, D., Wayman, J., 2005. *Biometric Systems Technology, Design and Performance Evaluation.* London: Spring Verlag.
- Jain, A.K., Duin, R.P.W., Mao, J., 2000. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 4–37.
- Jain, A.K., Li, S.Z., 2011. *Handbook of face recognition.* Springer.
- Jain, A.K., Ross, A., Prabhakar, S., 2004. An introduction to biometric recognition. *IEEE Trans. Circuits Syst. Video Technol.* 14, 4–20.
- Jain, P., Gangrade, J., 2013. Online Signature Verification Using Energy, Angle and Directional Gradient Feature with Neural Network. *Int. J. Innov. Res. Sci. Eng. Technol.* 2.
- Jiang, X., Yau, W.-Y., 2000. Fingerprint minutiae matching based on the local and global structures, in: *Pattern Recognition, 2000. Proceedings. 15th International Conference on.* IEEE, pp. 1038–1041.

- Jung, I.-K., Lacroix, S., 2001. A robust interest points matching algorithm, in: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, pp. 538–543.
- Kahl, D.J., Rosenfeld, A., Danker, A., 1978. Some experiments in point pattern matching. DTIC Document.
- Khotanzad, A., Hong, Y.H., 1990. Invariant image recognition by Zernike moments. IEEE Trans. Pattern Anal. Mach. Intell. 12, 489–497.
- Lamdan, Y., Wolfson, H.J., 1988. Geometric hashing: A general and efficient model-based recognition scheme.
- Lavine, D., Lambird, B.A., Kanai, L.N., 1983. Recognition of spatial point patterns. Pattern Recognit. 16, 289–295.
- Lee, L.L., Berger, T., Aviczer, E., 1996. Reliable online human signature verification systems. IEEE Trans. Pattern Anal. Mach. Intell. 18, 643–647.
- Leroy, A.M., Rousseeuw, P.J., 1987. Robust regression and outlier detection. Wiley Ser. Probab. Math. Stat. N. Y. Wiley 1987 1.
- Liu, Y., Rodrigues, M.A., 2002. Geometrical analysis of two sets of 3D correspondence data patterns for the registration of free-form shapes. J. Intell. Robot. Syst. 33, 409–436.
- Livi, L., Rizzi, A., 2013. The graph matching problem. Pattern Anal. Appl. 16, 253–283.
- Looney, C.G., 1997. Pattern recognition using neural networks: theory and algorithms for engineers and scientists. Oxford University Press, Inc.
- Maazouzi, A., Laaroussi, A., Aqili, N., Raji, M., Hammouch, A., 2016. Speech recognition system using burg method and Discrete Wavelet Transform, in: Electrical and Information Technologies (ICEIT), 2016 International Conference on. IEEE, pp. 250–254.
- Maazouzi, A.-E., Aqili, N., Hammouch, A., Raji, M., Maazouzi, A.-E., Aqili, N., Hammouch, A., Raji, M., 2016. Spoken Digits Recognition Using Wavelet Transform and Power Spectrum Density Estimation. Int. Rev. Comput. Softw. IRECOs 11, 35–41. doi:10.15866/irecos.v11i1.8261
- Maazouzi, A.-E., Aqili, N., Raji, M., Hammouch, A., 2015. A speaker recognition system using power spectrum density and similarity measurements, in:

- Complex Systems (WCCS), 2015 Third World Conference on. IEEE, pp. 1–5.
- Maintz, J.A., Viergever, M.A., 1998. A survey of medical image registration. *Med. Image Anal.* 2, 1–36.
- Malis, E., Marchand, E., 2005. Méthodes robustes d'estimation pour la vision robotique, in: *Journées Nationales de La Recherche En Robotique, JNRR'05*.
- Maltoni, D., Maio, D., Jain, A., Prabhakar, S., 2009. *Handbook of fingerprint recognition*. Springer Science & Business Media.
- Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., 1997. The DET curve in assessment of detection task performance. DTIC Document.
- Matula, P., Kozubek, M., Dvorak, V., 2006. Fast point-based 3-D alignment of live cells. *IEEE Trans. Image Process.* 15, 2388–2396.
- Medina-Pérez, M.A., García-Borroto, M., Gutierrez-Rodríguez, A.E., Altamirano-Robles, L., 2012a. Improving the multiple alignments strategy for fingerprint verification, in: *Mexican Conference on Pattern Recognition*. Springer, pp. 147–154.
- Medina-Pérez, M.A., García-Borroto, M., Gutierrez-Rodríguez, A.E., Altamirano-Robles, L., 2012b. Improving fingerprint verification using minutiae triplets. *Sensors* 12, 3418–3437.
- Medina-Pérez, M.A., Gutiérrez-Rodríguez, A., García-Borroto, M., 2009. Improving fingerprint matching using an orientation-based minutia descriptor, in: *Iberoamerican Congress on Pattern Recognition*. Springer, pp. 121–128.
- Medina-Pérez, M.A., Loyola-González, O., Gutierrez-Rodríguez, A.E., García-Borroto, M., Altamirano-Robles, L., 2014. Introducing an experimental framework in C# for fingerprint recognition, in: *Mexican Conference on Pattern Recognition*. Springer, pp. 132–141.
- Mo, F., Pei, H., Li, S., Li, D., Xie, B., Zhou, W., 2009. A Novel Object Tracking Algorithm Based on Point Pattern Matching, in: *2009 Second International Symposium on Information Science and Engineering*. IEEE, pp. 455–459.

- Monrose, F., Rubin, A., 1997. Authentication via keystroke dynamics, in: Proceedings of the 4th ACM Conference on Computer and Communications Security. ACM, pp. 48–56.
- Moses, K.R., Higgins, P., McCabe, M., Prabhakar, S., Swann, S., 2011. Automated fingerprint identification system (AFIS). Sci. Work. Group Frict. Ridge Anal. Study Technol. Natl. Inst. Justice Eds SWGFAST-Fingerpr. Sourceb. 1–33.
- Mount, D.M., Netanyahu, N.S., Le Moigne, J., 1999. Efficient algorithms for robust feature matching. *Pattern Recognit.* 32, 17–38.
- Nalwa, V.S., 1997. Automatic on-line signature verification. *Proc. IEEE* 85, 215–239.
- Navarro, G., 2001. A guided tour to approximate string matching. *ACM Comput. Surv. CSUR* 33, 31–88.
- Negin, M., Chmielewski, T.A., Salganicoff, M., von Seelen, U.M., Venetainer, P.L., Zhang, G.G., 2000. An iris biometric system for public and personal use. *Computer* 33, 70–75.
- Nixon, M.S., Carter, J.N., Cunado, D., Huang, P.S., Stevenage, S.V., 1996. Automatic gait recognition, in: *Biometrics*. Springer, pp. 231–249.
- Noor, A., 2012. A new algorithm for minutiae extraction and matching in fingerprint. Brunel University, UK.
- Ogawa, H., 1986. Labeled point pattern matching by Delaunay triangulation and maximal cliques. *Pattern Recognit.* 19, 35–40.
- Ogawa, H., 1984. Labeled point pattern matching by fuzzy relaxation. *Pattern Recognit.* 17, 569–573.
- O’Gorman, L., 2002. Seven issues with human authentication technologies, in: *Proc. Workshop Automatic Identification Advanced Technologies (AutoID)*. pp. 185–186.
- Pal, S.K., Amita, P., 2001. *Pattern recognition: from classical to modern approaches*. World Scientific.
- Pao, Y., 1989. *Adaptive pattern recognition and neural networks*.

- Paumard, J., 1997. Reconnaissance de formes sur des images comprimées, in: 16° Colloque Sur Le Traitement Du Signal et Des Images, FRA, 1997. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images.
- Phillips, P.J., Grother, P., Micheals, R.J., Blackburn, D.M., Tabassi, E., Bone, J.M., 2003. Face recognition vendor test (FRVT) 2002: Overview and summary. Online [Httpwww Frvt OrgFRVT2002documents Htm](http://www.frvt.org/FRVT2002/documents/Htm).
- Pitiot, A., Malandain, G., Bardinet, E., Thompson, P.M., 2003. Piecewise affine registration of biological images, in: International Workshop on Biomedical Image Registration. Springer, pp. 91–101.
- Polana, R., Nelson, R.C., 1997. Detection and recognition of periodic, nonrigid motion. *Int. J. Comput. Vis.* 23, 261–282.
- Portegys, T.E., 1995. A search technique for pattern recognition using relative distances. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 910–914.
- Preparata, F.P., Shamos, M., 2012. Computational geometry: an introduction. Springer Science & Business Media.
- Qi, J., Yang, S., Wang, Y., 2005. Fingerprint matching combining the global orientation field with minutia. *Pattern Recognit. Lett.* 26, 2424–2430.
- Ranade, S., Rosenfeld, A., 1980. Point pattern matching by relaxation. *Pattern Recognit.* 12, 269–275.
- Rangarajan, A., Chui, H., Mjolsness, E., 2001. A relationship between spline-based deformable models and weighted graphs in non-rigid matching, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, p. I–897.
- Rangarajan, A., Chui, H., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P., Duncan, J., 1997. A robust point-matching algorithm for autoradiograph alignment. *Med. Image Anal.* 1, 379–398.
- Rashidi, S., Fallah, A., Towhidkhah, F., 2012. Authentication based on signature verification using position, velocity, acceleration and Jerk signals, in: Information Security and Cryptology (ISCISC), 2012 9th International ISC Conference on. IEEE, pp. 26–31.
- Ratha, N.K., Karu, K., Chen, S., Jain, A.K., 1996. A real-time matching system for large fingerprint databases. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 799–813.

- Rodrigues, M.A., Liu, Y., 1999. Geometrical analysis of two sets of 3D correspondence data patterns, in: Shape Modeling and Applications, 1999. Proceedings. Shape Modeling International'99. International Conference on. IEEE, pp. 211–218.
- Rosenfeld, A., Kak, A.C., 2014. Digital picture processing. Elsevier.
- Rousseeuw, P.J., 1984. Least median of squares regression. *J. Am. Stat. Assoc.* 79, 871–880.
- Samet, H., 1990. The design and analysis of spatial data structures. Addison-Wesley Reading, MA.
- Sclaroff, S., Pentland, A.P., 1995. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 545–561.
- Seo, J.K., Sharp, G.C., Lee, S.W., 2005. Range data registration using photometric features, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, pp. 1140–1145.
- Sproull, R.F., 1991. Refinements to nearest-neighbor searching ink-dimensional trees. *Algorithmica* 6, 579–589.
- Stockman, G., 1987. Object recognition and localization via pose clustering. *Comput. Vis. Graph. Image Process.* 40, 361–387.
- Stockman, G., Kopstein, S., Bennett, S., 1982. Matching images to models for registration and object detection via clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 229–241.
- Tico, M., Kuosmanen, P., 2003. Fingerprint matching using an orientation-based minutia descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1009–1014.
- Tombari, F., Di Stefano, L., Mattocchia, S., 2007. A robust measure for visual correspondence, in: Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on. IEEE, pp. 376–381.
- Ton, J., Jain, A.K., 1989. Registering Landsat images by point matching. *IEEE Trans. Geosci. Remote Sens.* 27, 642–651.
- Torreão, J.R., Fernandes, J.L., 1998. Matching photometric-stereo images. *JOSA A* 15, 2966–2975.

- Torresani, L., Kolmogorov, V., Rother, C., 2008. Feature correspondence via graph matching: Models and global optimization, in: European Conference on Computer Vision. Springer, pp. 596–609.
- Udup, R., Garg, G., Sharma, P., 2001. Fast and accurate fingerprint verification, in: International Conference on Audio-and Video-Based Biometric Person Authentication. Springer, pp. 192–197.
- Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 376–380.
- Upek, 2013. <http://www.advancedsourcecode.com/PNGfingerprint.rar> [WWW Document]. URL <http://www.advancedsourcecode.com/> (accessed 4.5.17).
- VC, H.P., 1962. Method and means for recognizing complex patterns. Google Patents.
- Veltkamp, R.C., Hagedoorn, M., 2001. State of the art in shape matching, in: *Principles of Visual Information Retrieval*. Springer, pp. 87–119.
- von der Malsburg, C., 1988. Pattern recognition by labeled graph matching. *Neural Netw.* 1, 141–148.
- Wilson, C., Hicklin, R.A., Korves, H., Ulery, B., Zoepfl, M., Bone, M., Grother, P., Micheals, R., Otto, S., Watson, C., others, 2004. Fingerprint vendor technology evaluation 2003: Summary of results and analysis report. NIST Res. Rep. 7123.
- Wolfson, H.J., 1990. Model-based object recognition by geometric hashing, in: European Conference on Computer Vision. Springer, pp. 526–536.
- Wolfson, H.J., Rigoutsos, I., 1997. Geometric hashing: An overview. *IEEE Comput. Sci. Eng.* 4, 10–21.
- Wong, A.K., Salay, R., 1986. An algorithm for constellation matching, in: *Proceedings of 8th International Conference on Pattern Recognition*. pp. 546–549.
- Yianilos, P.N., 1993. Data structures and algorithms for nearest neighbor search in general metric spaces, in: *SODA*. pp. 311–21.

- Zhang, D., Kong, W.-K., You, J., Wong, M., 2003. Online palmprint identification. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1041–1050.
- Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* 13, 119–152.
- Zunkel, R., n.d. Hand geometry based authentication. *Biom. Pers. Identif. Networked Soc.* 87–102.