

## RÉSUMÉ

Actuellement la vision par ordinateurs, l'archivage et la recherche dans les bases de modèles 3D sont des domaines de recherches très actifs. Plus particulièrement, la recherche d'informations par le contenu est un axe très intéressant et ne cesse de se développer. Néanmoins, il reste toujours des pistes à creuser et ainsi des axes à perfectionner. Dans ce contexte, notre objectif est de reconnaître un objet 3D donné dans une base de donnée, à partir d'une base d'apprentissage contenant quelques vues de cet objet ou bien des objets 3D de forme libre. Notre idée est de formuler une méthodologie locale qui combine les aspects des approches existantes et apporte une amélioration sur la performance de la reconnaissance. Dans cette thèse, deux types d'approches ont été étudiées : l'approche 2D/3D et l'approche 3D/3D.

La première partie de la thèse a été consacrée à l'étude des descripteurs de forme 2D, à savoir, le Moment de Zernike qui est robuste dans la description des images (en particulier les images couleurs), et le descripteur Centrist. Nous avons combiné ces deux descripteurs classiques afin d'augmenter le taux de reconnaissance des objets. En plus, nous avons proposé une approche d'indexation 2D/3D, qui permet de définir un objet 3D à partir d'une image 2D. Cette méthode permet de comparer directement l'image requête à l'objet 3D et s'adapte idéalement à la requête. Concernant la technique de comparaison suivie, elle se base sur la représentation matricielle des vues de l'objet, permettant ainsi d'extraire les caractéristiques et de mettre en place une recherche de la vue requête dans la base traitée.

La deuxième partie de la thèse présente à la fois la théorie et la pratique des approches 3D/3D basées sur le calcul numérique des distances géodésiques sur les variétés riemanniennes. Cette distance géodésique répond à une équation différentielle non linéaire, l'équation Eikonale. Nous avons proposé deux nouvelles approches, la première se base sur la distance géodésique pour calculer la distance minimale entre deux points de deux objets 3D différents. L'autre approche utilise l'équation de la chaleur à la place de l'équation d'Eikonale, tout en préservant le processus de minimisation de la distance. L'idée derrière cette proposition est d'utiliser une équation linéaire facile à résoudre avec les mêmes propriétés que celle de l'équation Eikonale. Les résultats obtenus sont satisfaisants surtout au niveau du temps de calcul.

Nous illustrons les résultats expérimentaux obtenus des approches proposées à l'aide des méthodes de classification les plus connus à savoir les réseaux de Neurones, SVM multi-classes, les arbres de décision telles que (C4.5, Random Forest) et les k plus proches voisins.

**Mots-clés** : Indexation, reconnaissance, objet 3D, RN, KNN, RF, Equation Eikonale, Equation de la chaleur.



UNIVERSITE SULTAN MOULAY SLIMANE  
Faculté des Sciences et Techniques  
Béni-Mellal



Centre d'Études Doctorales : Sciences et Techniques  
Formation Doctorale : Mathématiques et Physique Appliquées

### THÈSE

Présentée par

**DRISS NAJI**

Pour l'obtention du grade de

**DOCTEUR**

*Spécialité : Informatique*

## Contribution à la reconnaissance des objets 3D

Soutenue le 12/04/2017 à 9h devant la commission d'examen composée de :

Président	: Pr. Said SAFI	FH	FP - Béni Mellal
Rapporteurs	: Pr. Cherki DAoui	PES	FST - Béni Mellal
	Pr. Brahim MINAOUI	PH	FST - Béni Mellal
Examineurs	: Pr. Jilali ANTARI	PH	FP - Taroudant
Directeurs de thèse	: Pr. Mohamed FAKIR	PES	FST - Béni Mellal
Co-encadrant	: Pr. Belaid BOUIKHALENE	PH	FP - Béni Mellal

# Dédicaces

---

*J'ai l'honneur de dédier ce modeste travail :*

*A mes très chers parents qui ont consenti d'énormes sacrifices afin de me mener à la réussite de mes études, à travers ce travail qu'ils croient à mes chaleureuses expressions d'amour.*

*A mes chers frères et sœurs pour leurs conseils et leur profond amour.*

*A tous mes amis sans exception, et mes collègues d'étude du Master informatique décisionnel.*

*En fin, que toutes personnes de près ou de loin qui ont participé à ce travail soient remerciées infiniment.*

# Remerciement

---

En tout premier lieu, je tiens à adresser mes grands respects et mes remerciements à mon directeur de thèse, le professeur **Mohamed FAKIR** et mon co-encadrent, le professeur **Belaid BOUIKHALENE** pour l'aide compétente qu'ils m'ont apporté, pour leur patience et leurs encouragements à finir cette thèse. Malgré leurs responsabilités, ils ont toujours répondu présent, me montrant ainsi la voie tout en me laissant une grande liberté de pensée et d'action. Leur renseignements m'ont été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections. De chaque discussion, même brève, j'en suis sortie enrichi. J'ai eu de la chance de travailler sous votre direction pendant ces années. Merci.

Je remercie vivement tous les membres de jury de ma thèse qui ont pris de leurs temps pour lire et juger mon travail ainsi que pour leur déplacement le jour de la soutenance. Egalement, je remercie infiniment les rapporteurs de ma thèse pour leurs corrections et leurs remarques appropriées vis-à-vis de mon manuscrit.

J'adresse également mes sincères remerciements aux différents intervenants et professeurs que j'ai rencontrés tout au long de ma scolarité, en particulier au cours de mon Master Informatique Décisionnelle (MID), qui m'ont inspiré, conseillé et qui m'ont également motivé pour continuer mes études. Je tiens aussi à remercier mes collègues et amis pour tous les bons moments passés ensemble et pour les idées qu'on a échangées.

Enfin, je tiens à remercier ma famille, en particulier mes parents et mes amis pour leur soutien, leurs encouragements et avis sur mon sujet. Ils ont été présents tout au long de ce travail, je ne serais certainement pas là sans eux. Merci !

Pour finir, que toute personne qui ait contribué de près ou de loin à la réalisation de cette thèse, trouve ici l'expression de mes sincères sentiments.

# Résumé

---

Actuellement la vision par ordinateurs, l'archivage et la recherche dans les bases de modèles 3D sont des domaines de recherches très actifs. Plus particulièrement, la recherche d'informations par le contenu est un axe très intéressant et ne cesse de se développer. Néanmoins, il reste toujours des pistes à creuser et ainsi des axes à perfectionner. Dans ce contexte, notre objectif est de reconnaître un objet 3D donné dans une base de donnée, à partir d'une base d'apprentissage contenant quelques vues de cet objet ou bien des objets 3D de forme libre. Notre idée est de formuler une méthodologie locale qui combine les aspects des approches existantes et apporte une amélioration sur la performance de la reconnaissance. Dans cette thèse, deux types d'approches ont été étudiées : l'approche 2D/3D et l'approche 3D/3D.

La première partie de la thèse a été consacrée à l'étude des descripteurs de forme 2D, à savoir, le Moment de Zernike qui est robuste dans la description des images (en particulier les images couleurs), et le descripteur Centrist. Nous avons combiné ces deux descripteurs classiques afin d'augmenter le taux de reconnaissance des objets. En plus, nous avons proposé une approche d'indexation 2D/3D, qui permet de définir un objet 3D à partir d'une image 2D. Cette méthode permet de comparer directement l'image requête à l'objet 3D et s'adapte idéalement à la requête. Concernant la technique de comparaison suivie, elle se base sur la représentation matricielle des vues de l'objet, permettant ainsi d'extraire les caractéristiques et de mettre en place une recherche de la vue requête dans la base traitée.

La deuxième partie de la thèse présente à la fois la théorie et la pratique des approches 3D/3D basées sur le calcul numérique des distances géodésiques sur les variétés riemanniennes. Cette distance géodésique répond à une équation différentielle non linéaire, l'équation Eikonale. Nous avons proposé deux nouvelles approches, la première se base sur la distance géodésique pour calculer la distance minimale entre deux points de deux objets 3D différents. L'autre approche utilise l'équation de la chaleur à la place de l'équation d'Eikonale, tout en préservant le processus de minimisation de la distance. L'idée derrière cette proposition est

d'utiliser une équation linéaire facile à résoudre avec les mêmes propriétés que celle de l'équation Eikonale. Les résultats obtenus sont satisfaisants surtout au niveau du temps de calcul.

Nous illustrons les résultats expérimentaux obtenus des approches proposées à l'aide des méthodes de classification les plus connus à savoir les réseaux de Neurones, SVM multi-classes, les arbres de décision telles que (C4.5, Random Forest) et les k plus proches voisins.

**Mots clés** : Indexation, reconnaissance, objet 3D, RN, KNN, RF, Equation Eikonale, Equation de la chaleur.

# Abstract

---

Currently, computer vision, archiving and searching in 3D model databases are very active research areas. More particularly, the information search by the content is a very interesting axis, and does not cease to develop. Nevertheless, there are always tracks to dig, so axes to perfect. In this context, our objective is to recognize a given 3D object in a database, from a learning base containing some views of this object or 3D objects. Our idea is to formulate a local methodology that combines aspects of existing approaches and brings an improvement on the performance of the recognition. In this thesis, two types of approaches have been studied: the 2D / 3D approach and the 3D / 3D approach.

The first part of the thesis was devoted to study two 2D form descriptors, namely, the Zernike Moment that is robust in describing images (especially color images), and the CENTRIST descriptor. We have combined these two conventional descriptors to increase the recognition rate of objects. In addition, we proposed a 2D / 3D indexing approach, which allows defining a 3D object from a 2D image. This method permits directly comparing the query image to the 3D object and it's ideally adapts to the query. As for the comparison technique, it is based on the views matrix representation of the object, therefore extracting the characteristics and setting up a search of the query view in the processed database.

The second part of the thesis presents both the theory and the practice of 3D / 3D approaches based on the numerical calculation of the geodesic distances on the Riemannian manifolds. This geodesic distance corresponds to a nonlinear differential equation, the Eikonale equation. We proposed two new approaches, the first based on the geodesic distance to calculate the minimum distance between two points of two different 3D objects. The other approach uses the heat equation instead of the Eikonale equation, while preserving the minimization process of distance. The idea behind this proposition is to use a simple linear equation with the same properties as the Eikonale equation. The results obtained are satisfactory especially in terms of computation time.

We illustrate the experimental results obtained from the proposed approaches using the most well-known classification methods, namely Neural networks, multi-class SVM, decision trees such as (C4.5, Random Forest) and k nearest neighbors.

**Key words:** Indexing, recognition, 3D object, RN, KNN, RF, Equation Eikonale, Equation of heat.

أصبحت الرؤية عن طريق الكمبيوتر، والأرشفة والبحث في القواعد الخاصة بالنماذج ثلاثية الأبعاد مجالات بحث حية تعرف إقبالا كبيرا من طرف الباحثين. كما أن البحث عن المعلومات عن طريق المحتوى بصفة خاصة، يمثل محورا مهما لازال في تطور متواصل. وبالرغم من ذلك، فلا تزال هناك مسارات وطرق أخرى للتقيب والبحث. إن الهدف المنشود في هذا الإطار، هو التعرف على جسم ثلاثي الأبعاد معلوم من خلال قاعدة بيانات، انطلاقا من قاعدة تعلم تحتوي على مشاهد لواجهات هذا الجسم أو على أجسام ثلاثية الأبعاد ذات شكل حر. تتمحور فكرتنا ومقترحنا في صياغة طريقة أو خوارزمية محلية التي تدمج مميزات المقاربات الموجودة، مع بعض التطوير فيما يخص الأداء وجودة التعرف على الأجسام. في رسالة الأطروحة هاته، نظرقنا لدراسة مقاربتين أساسيتين: المقاربة 2D/3D والمقاربة 3D/3D:

الجزء الأول، وقد خصص لدراسة الواصفات الثنائية الأبعاد. لقد قمنا، في الواقع، بالجمع بين واصفتين كلاسيكيتين وهما: سنتريست (CENTRIST) وعزم زرنيك (Moments de Zernike)، وذلك من أجل الرفع من معدل التعرف على الأجسام. وتكمن الفكرة وراء اقتراح هذا النموذج الهجين، من جهة، في استخدام قوة عزم زرنيك وقدرته الكبيرة على وصف الصور خصوصا الملونة منها. من جهة أخرى، يمكن استخدام واصفات سنتريست بالإضافة إلى عزم زرنيك من تحديد الواصفات المثالية، وكذا فهرسة قواعد كبيرة خاصة بالصور الملونة. بعد ذلك قمنا باقتراح مقاربة لفهرسة 2D/3D التي تمكن من تعريف جسم ثلاثي الأبعاد انطلاقا من صورة ثنائية الأبعاد. وترتكز طريقة الفهرسة 2D/3D، المقترحة على المقارنة المباشرة بين الصورة والجسم الثلاثي الأبعاد. أما فيما يخص أسلوب المقارنة المعتمد، فإن نظامنا يستند على تمثيل مشاهدات الجسم على شكل مصفوفة مما يضمن استخراج المميزات والخصائص وإنشاء بحث عن المشاهدة المطلوبة في القاعدة التي تتم معالجتها.

في حين أن الجزء الثاني من الأطروحة قد خصص، في نفس الوقت، لتقديم نظريات و تطبيقات مقاربات 3D/3D المرتكزة أساسا على حساب المسافات الجيوديزية على متعدد شعب ريمان (variétés riemanniennes). والشيء المهم هو أن المسافة الجيوديزية من مجموعة نقط انطلاق هو حل معادلة تفاضلية غير خطية، معادلة Eikonale. اقترحنا نهجين جديدين، يستند لأول على حساب الحد الأدنى للمسافة الجيوديزية بين نقطتين على نوعين مختلفين من الاجسام ثلاثية الأبعاد. وترتكز الطريقة الأخرى على حل المعادلة الحرارية بنفس طريقة معادلة Eikonale، مع الحفاظ على

عملية تخفيض المسافة. الفكرة وراء هذا الاقتراح هو استخدام معادلة خطية بسيطة لحلها مع نفس الخصائص كالمعادلة eikonal. وكانت النتائج مرضية خصوصا في وقت الحساب.

من أجل توضيح وتمثيل النتائج التجريبية المحصل عليها استنادا إلى الطرق والمقاربات المقترحة، استخدمنا أساليب التصنيف المعروفة وهي الشبكات العصبية، SVM متعدد الطبقات، وأشجار القرار مثل (C4.5, Random Forest) و k جيران الأكثر قربا.

**كلمات البحث:** الفهرسة، البحث، مشاهدات، أجسام 3D، RN، KNN، RF، معادلة Eikonale ومعادلة الحرارة.

## ***LISTE DES PUBLICATIONS ET COMMUNICATIONS***

### **Publications :**

- 1) D. Naji, M. Fakir, R. Elayachi and B. Bouikhalene; "Recognition of 3D Objects using Heat Diffusion Equations and Random Forests", 978-1-5090-0811-7/16, IEEE DOI 10.1109/CGiV.2016.39
- 2) D. Naji, M. Fakir, B. Bouikhalene and O. Bencharef; "3D Objects Retrieval using Geodesic Distance based on Eikonal equation", TELKOMNIKA Indonesian Journal of Electrical Engineering (ISSN : 2302-4046). Vol 15, No 3, Septembre 2015.
- 3) D. Naji, M. Fakir, B. Bouikhalene and M. Boutaounte; "Recognition of Color Objects Using Hybrids Descriptors". IGI Global, International Journal of Computer Vision and Image Processing Vol 13, Issue 4, 60-68, 2013.

### **Communications :**

- 1) Driss Naji, M. Fakir, R. Elayachi, B. Bouikhalene; "Recognition of 3D Objects using Heat Diffusion Equations and Random Forests", in 13th International Conference Computer Graphics, Imaging and Visualization Béni-Mellal, 2016.
- 2) Driss Naji, M. Fakir, R. Elayachi, B. Bouikhalene; "Recognition of 3D Objects using Heat Diffusion Equations and Random Forests", in Second international Conference on Business Intelligence (CBI' 15) Béni-Mellal, 2015.
- 3) Driss Naji, M. Fakir, B. Bouikhalene et O. Bencharef; "3D Objects Retrieval using Geodesic Distance based on Eikonal equation", in Second international Conference on Business Intelligence (CBI' 15) Béni-Mellal, 2015.
- 4) Driss Naji, M. Fakir, B. Bouikhalene, M. Boutaounte; "Recognition of Color Objects Using Hybrids Descriptors", in third International Symposium on Automatic Amazigh processing SITACAM' 13 Béni-Mellal, 2013.

# Sommaire

<b>CHAPITRE 1 : Généralités .....</b>	<b>17</b>
<b>1.1 Introduction.....</b>	<b>18</b>
<b>1.2. Système de reconnaissance de formes .....</b>	<b>18</b>
1.2.1 Prétraitement.....	19
1.2.2 Extraction de caractéristiques .....	20
1.2.3 Types d'apprentissage.....	20
1.2.4 Classification .....	21
<b>1.3 Descripteurs 2D et 3D.....</b>	<b>22</b>
1.3.1. Approches globales .....	23
1.3.2 Approches locales .....	25
1.3.3 Approches statistique .....	29
1.3.4 Méthodes graphiques .....	30
1.3.5 Méthodes basées sur la vue.....	30
<b>1.4 Conclusion .....</b>	<b>32</b>
<b>CHAPITRE 2 : Extraction des caractéristiques .....</b>	<b>34</b>
<b>2.1 Introduction.....</b>	<b>35</b>
<b>2.2 Descripteurs 2D.....</b>	<b>35</b>
2.2.1 Descripteur CENTRIST .....	36
2.2.2 Moments de Zernike .....	36
2.2.3 Analyse en Composantes Principales (ACP) .....	37
<b>2.3 Equation Eikonale .....</b>	<b>38</b>
2.3.1 Définition de la courbe.....	40
2.3.2 Définition première forme fondamentale .....	40
2.3.3 Surfaces comme étant variété Riemannienne .....	41
2.3.4 La fonction distance géodésique .....	44
2.3.4 Discrétisation de l'équation Eikonale.....	45
<b>2.4 Equation de la chaleur .....</b>	<b>48</b>
2.4.1 Noyau de la chaleur .....	49
2.4.2 Discrétisation par rapport au temps .....	51
<b>2.5 Conclusion .....</b>	<b>54</b>
<b>CHAPITRE 3 : CLASSIFICATION.....</b>	<b>55</b>
<b>3.1 Introduction.....</b>	<b>56</b>
<b>3.2 Réseaux de Neurones .....</b>	<b>56</b>
3.2.1 Apprentissage .....	57
3.2.2 Perceptron .....	57
3.2.3 Perceptrons multicouches .....	59
<b>3.3 Machines à Vecteurs de Support (SVM) .....</b>	<b>60</b>
3.3.1 Principe général .....	61
3.3.2 SVM multi-classes .....	65
<b>3.4 k plus proches voisins (KNN).....</b>	<b>66</b>

3.4.1 Algorithme 1er proche voisin (1-NN) .....	66
3.4.2 Algorithme K proche voisin (KNN) .....	67
3.4.3 Choix de k : .....	68
<b>3.5 K-means.....</b>	<b>68</b>
<b>3.6 Arbres de décision.....</b>	<b>69</b>
3.6.1 Construction d'un arbre de décision .....	70
3.6.2 Mesure d'homogénéité des feuilles.....	71
<b>3.7 Algorithme C4.5 .....</b>	<b>72</b>
3.7.1 Attributs de valeur inconnue .....	72
3.7.2 Attributs à valeur sur intervalle continu .....	72
3.7.3 Elagage de l'arbre de décision .....	73
3.7.4 Algorithme de base .....	73
<b>3.8 Forêts aléatoires .....</b>	<b>75</b>
3.8.1 Bagging.....	75
3.8.2 Extensions des forets aléatoires .....	75
3.8.3 Principe .....	76
<b>6.9 Conclusion .....</b>	<b>77</b>
<b>CHAPITRE 4 : Résultats Expérimentaux .....</b>	<b>78</b>
<b>4.1 Introduction.....</b>	<b>79</b>
<b>4.2 Résultats Approche 2D/3D .....</b>	<b>79</b>
4.2.1 Base d'images utilisée .....	79
4.2.2 Procédure d'indexation.....	80
4.2.3 Méthodologie d'application.....	83
4.2.4 Résultats expérimentaux .....	83
<b>4.3 Approche 3D/3D .....</b>	<b>87</b>
4.3.1 Bases de données.....	88
4.3.2 Résultats expérimentaux (Equation Eikonale) .....	91
4.3.3 Résultats expérimentaux (Equation de la chaleur) .....	95
<b>4.5 Conclusion .....</b>	<b>100</b>
<b>Conclusions et perspectives .....</b>	<b>101</b>
<b>Bibliographie .....</b>	<b>103</b>

# Liste d'acronymes

- RN : Réseaux de neurones
- SVM: Machines à Vecteurs de support
- RF : Random Forest (Forêt Aléatoire)
- KNN: K plus proches voisins
- ACP: Analyse en Composantes Principales
- SV : Saillents Views
- LFD : Descripteur de champ Linière (LFD)

# Liste des tableaux

Tableau 1.1	Résumé des avantages et inconvénients des approches 3D/3D et des approches 2D/3D.....	31
Tableau 4.1	Erreur d'apprentissage en fonction de l'ordre et de la méthode utilisée pour l'extraction .....	71
Tableau 4.2	Taux de reconnaissance en fonction de l'ordre de la base.....	72
Tableau 4.3	Taux de reconnaissance pour RDN et SVM avec (1200 images).....	73
Tableau 4.4	Taux de reconnaissance pour un nombre différent de courbe en utilisant la distance géodésique appliquée à la base données Watertight.....	77
Tableau 4.5	Rconnaissance d'objets 3D en utilisant la distance géodésique et KNN.....	78
Tableau 4.6	Évaluation de la distance géodésique utilisant PCA.....	78
Tableau 4.7	Taux de reconnaissance de la base Watertight.....	79
Tableau 4.8	Taux de reconnaissance de la base donnée Watertight.....	80
Tableau 4.9	Etude comparative entre les méthodes, l'équation de chaleur et équation Eikonale.....	82

# Liste des figures

Figure 1.1	Système de reconnaissance de formes .....	13
Figure 2.1	Plan tangent $\mathcal{T}_{\bar{x}}$ et le dérivée de la courbe sur la surface $S$ .....	40
Figure 2.2	Voisinage $B(x)$ de chaque point $x \in \Omega$ [71].....	43
Figure 2.3	Point $y$ qui réalise l'égalité dans la minimisation (5-27).....	43
Figure 3.1	Perceptron avec seuil [64].....	50
Figure 3.2	Quelques hyperplans linéaires séparateurs valides [72].....	53
Figure 3.3	Hyperplan linéaire ayant la marge maximale [72].....	53
Figure 3.4	Exemple de classification SVM non-linéairement séparable [67].....	54
Figure 3.5	Clustering sur un ensemble de points 2D, 3 clusters.....	58
Figure 4.1	Echantillons des 100 objets de la base d'images COIL-100 de COLUMBIA [68].....	67
Figure 4.2	Schéma de la procédure d'indexation.....	68
Figure 4.3	Erreur d'apprentissage en fonction de l'ordre de la base .....	71
Figure 4.4	Taux de reconnaissance en fonction du nombre d'image dans la base .....	72
Figure 4.5	Taux de reconnaissance pour RDN et SVM avec (1200 images).....	73
Figure 4.6	Organigramme du système de reconnaissance 3D.....	74
Figure 4.7	Exemple d'objets de quatre classes de la base Princeton [85]. (a), (b), (c) et (d) illustrent respectivement les classes "avion", "chaise", "main" et "visage" .....	75
Figure 4.8	Exemple d'objets de quatre classes de la base Watertight [86]. (a), (b), (c) et (d) illustrent respectivement les classes "main", "lunette", "humain", "table".....	76
Figure 4.9	Taux de reconnaissance obtenu pour les trois méthodes. ....	81
Figure 4.10	Etude comparative des taux de reconnaissance pour l'équation de la chaleur et de distance géodésique en utilisant Forêts Aléatoires.....	82

# Introduction

---

## **But :**

La reconnaissance de formes est un moyen d'indexer et de rechercher des objets dans des bases de données bien précises. Généralement, elle classe une forme donnée en catégories définies par d'autres formes classiques. Le premier objet de cette thèse vise à améliorer les techniques d'extraction de l'information significative permettant la reconnaissance des objets 3D. Le deuxième but consiste à développer des systèmes de reconnaissance d'objets 3D.

## **Motivation :**

La reconnaissance des objets 3D à haute définition et à grande échelle ont connus un succès considérable sous le thème de vision par ordinateur. Il est maintenant possible de parcourir facilement un objet avec un téléphone intelligent [1, 2] ou de visiter une ville virtuelle via Google Earth [3]. D'autre part, la conception assistée par ordinateur a également évolué, en effet, les modèles 3D d'objets sont maintenant disponibles à grande quantité dans des bibliothèques publiques ou commerciales [4, 5]. Cette quantité énorme de données nécessite de nouveaux outils de traitement, ainsi le développement de nouvelles applications est inévitable. Une autre motivation majeure est le fait que de nombreuses applications nécessitent les techniques d'indexation. Ces applications peuvent être résumées comme suit :

- Navigation et paramétrage des grandes collections de forme. La création d'un nouveau modèle 3D requiert du temps et des connaissances spécialisées. Au lieu de créer de nouveaux modèles, le concepteur pourrait parcourir, rechercher et manipuler de manière intuitive les modèles 3D existants dans la collection.
- Parcourir les données historiques. Imaginez qu'un ordinateur puisse automatiquement récupérer le point de vue de toutes les images historiques existantes. Cela pourrait changer la façon d'accéder et parcourir les archives d'images historiques. Les utilisateurs peuvent ainsi parcourir les images

intuitivement et comparer des représentations de lieux similaires à des moments différents.

- Édition d'image intelligente. Imaginez qu'un ordinateur puisse identifier des objets dans une image 2D en entrée et récupérer automatiquement les modèles 3D associés.
- Manipulation robotique. Pour qu'un robot manipule un objet, il doit savoir non seulement dans quelle direction l'objet est, mais aussi avoir accès au modèle 3D associé, par exemple, des parties invisibles.

### **Défis :**

Bien qu'il existe plusieurs applications intéressantes pour la reconnaissance d'objets 3D, trouver de bonnes représentations et des algorithmes de reconnaissance appropriés n'est pas une tâche facile. Un bon descripteur d'une forme 3D doit vérifier les conditions suivantes :

- Etre discriminatif pour distinguer entre deux modèles dans différentes poses les uns des autres.
- Etre robuste par rapport aux perturbations comme les déformations, le bruit ou les changements topologiques du modèle étudié.

En général, il est difficile de prendre en considération ces caractéristiques en même temps car elles sont souvent contradictoires. Cependant, nous nous sommes intéressés particulièrement sur le compromis entre la robustesse et le pouvoir discriminatif, tout en développant des nouveaux descripteurs.

### **Contributions :**

Les principales contributions de cette thèse portent sur les deux problématiques associées à l'indexation et à la recherche d'objets 3D dans des bases de données connues. L'objectif est de développer de nouvelles approches de reconnaissance de modèles 3D à partir d'une requête en deux ou trois dimensions.

Dans un premier temps, nous avons réalisé une étude sur les descripteurs 2D classiques pour connaître leurs principales avantages et défauts. Ensuite, pour augmenter le taux de reconnaissance, nous avons proposé une nouvelle approche

(hybride) qui combine les avantages des moments de Zernike et le descripteur Centrist.

La contribution majeure de ce travail a porté sur la reconnaissance directe des objets 3D. En effet, nous avons présenté deux nouvelles approches : la première utilise la distance géodésique pour calculer la distance minimale entre six points différents de l'objet. Ceci va évidemment discriminer les objets 3D dans la base de données avec des caractéristiques uniques pour chaque classe d'une part et d'autre part diminuer le temps de calcul et ainsi faciliter la reconnaissance d'objets de topologie complexes. Cette distance est calculée via la résolution de l'équation Eikonale sur un maillage régulier triangulaire. Ensuite, dans l'esprit de réduire le temps de calcul et augmenter le taux de reconnaissance, nous avons proposé une deuxième approche qui résoudra l'équation de la chaleur sur la variété riemannienne à la place de l'équation d'Eikonale, tout en préservant le processus de minimisation de la distance. Nous présenterons en détail ces contributions dans les sections qui suivent.

### **Aperçu de la thèse :**

Dans le premier chapitre, nous définissons le système de reconnaissance de forme d'une manière générale. Ensuite nous introduisons un état d'art sur les méthodes d'extraction des caractéristiques en relation avec le thème de cette thèse. Nous présentons en détail, quelques descripteurs de représentation de formes 3D.

Le deuxième chapitre sera consacré à l'étude des trois nouvelles approches proposées d'indexation d'un objet 3D. En effet, nous allons proposer trois nouveaux descripteurs "2D/3D" et "3D/3D" codifiant les caractéristiques de forme intrinsèques à un modèle donné permettant sa comparaison lors d'une requête de l'utilisateur. Une présentation détaillée de ces approches sera élaborée.

Le troisième chapitre est consacré à la présentation des différentes techniques de classification qui vont nous aider à la mise en correspondance des informations stockées, permettant à l'utilisateur d'interroger la base de données.

Le quatrième chapitre fait l'objet des évaluations des méthodes d'indexation proposées. Ensuite, nous effectuons une comparaison de nos approches avec quelques méthodes de la littérature,

Finalement, nous terminons cette thèse par une conclusion générale et quelques perspectives.

# **CHAPITRE 1 :**

# **Généralités**

## 1.1 Introduction

Les systèmes automatisés de reconnaissance de formes utilisent des algorithmes pour traiter des données recueillies : soit électroniquement via des capteurs, soit transcrites par un humain, ce qui permet d'identifier un groupe dont les données sont les plus représentatives.

Les algorithmes utilisés par les systèmes de reconnaissance de formes se font généralement en deux tâches. Premièrement, la description des données collectées à partir de l'environnement en caractéristiques, c'est-à-dire toute valeur qui peut être représentative des données utilisées, ensuite, il y a la tâche de classification pour parvenir à une identification.

Dans ce chapitre, on discutera en détails la reconnaissance d'objets. La première section traitera le système de reconnaissance d'objets en général, définissant ainsi ses différentes composantes. Ensuite, on enchaînera par un état d'art comportant les approches majeures d'extraction des caractéristiques décrites dans la littérature. On définira tout d'abord, les descripteurs globaux et les transformations, puis, l'utilisation des caractéristiques locales.

## 1.2. Système de reconnaissance de formes

La reconnaissance de formes peut être définie comme la classification des objets en un certain nombre de catégories ou de classes par l'extraction d'informations significatives. Généralement, ce processus est basé sur deux étapes : la phase d'apprentissage et la phase de classification.

Dans les systèmes généraux de reconnaissance de formes, il existe deux étapes dans la construction d'un classificateur : l'apprentissage et le test (ou la reconnaissance). Ces étapes peuvent être subdivisées en sous-étapes.

### **Apprentissage :**

1. Prétraitement : Traiter les données afin qu'elles soient sous une forme appropriée.

2. Extraction de caractéristiques : Réduire la quantité de données en extrayant des informations pertinentes, ce qui nous donne généralement un vecteur de valeurs scalaires.
3. Estimation du modèle : A partir de l'ensemble fini de caractéristiques, il faut estimer un modèle pour chaque classe des données de formation.

### Reconnaissance :

1. Prétraitement
2. Extraction de caractéristiques : (les deux étapes sont les mêmes que celle de l'apprentissage).
3. Classification : Comparer les vecteurs caractéristiques aux différents modèles et trouver la correspondance la plus proche. On peut faire correspondre les vecteurs caractéristiques obtenus dans l'ensemble d'apprentissage.

Un système de reconnaissance de formes est présenté comme le montre la figure 1.1.

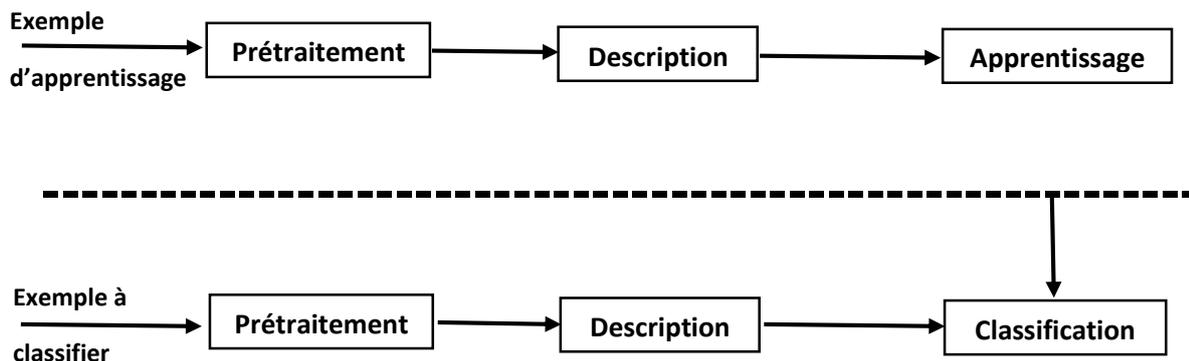


Figure 1.1: Système de reconnaissance de formes.

#### 1.2.1 Prétraitement

Cette étape comprend typiquement des opérations qui améliorent la représentation des objets. Par conséquent, il peut inclure l'enregistrement des données, l'élimination du bruit, la segmentation et la normalisation des données, en fonction de la nature de la tâche de reconnaissance des objets.

Les échantillons de modèles contiennent habituellement du bruit, ce qui nécessite une étape de réduction avant la description. Le terme "bruit" est généralement défini

dans un sens large dans la reconnaissance, tandis que les filtres sont habituellement utilisés pour supprimer le bruit et augmenter l'efficacité des descripteurs.

Certaines tâches de reconnaissance exigent la segmentation de modèles individuels. Pour segmenter des faces dans une image pour créer des objets significatifs pour l'étape d'extraction de caractéristiques.

La normalisation consiste à mettre à l'échelle les caractéristiques des données pour qu'elles tombent dans une petite plage spécifiée. Certains modèles de réseaux neuronaux nécessitent une normalisation des échantillons de données dans la plage de -1 à 1 ou 0 à 1. Toutes ces opérations contribuent à définir une représentation compacte des modèles [6].

### **1.2.2 Extraction de caractéristiques**

L'extraction de caractéristiques est le processus de coder la forme d'un objet sous une description pour l'utiliser dans la tâche de classification. La sélection du meilleur ensemble de caractéristiques pour la réduction des dimensions est l'une des questions les plus importantes de la reconnaissance des formes. Le but de la description est de réduire le nombre de caractéristiques des motifs et, en même temps, de conserver autant que possible leurs informations discriminatoires. Par conséquent, un bon extracteur de caractéristiques choisit des fonctionnalités qui sont similaires pour des motifs dans la même classe et très différentes pour des motifs dans différentes classes. Puisque la dimensionnalité de l'espace échantillon est réduite après l'étape d'extraction de caractéristique, l'extraction entraînera des économies de mémoire et de consommation de temps.

Dans ce qui suit, nous utiliserons de façon interchangeable les termes «ensemble de caractéristiques», «attribut» et «description» pour désigner les éléments utilisés pour caractériser un objet selon une technique spécifique.

### **1.2.3 Types d'apprentissage**

Un système de reconnaissance de formes produit un modèle mathématique qui mappe les objets aux classes correspondantes. Typiquement, il n'est pas possible de déterminer un mappage fiable sans l'aide d'échantillons de données. Trouver ce

modèle est appelé apprentissage ou de formation, et les modèles d'échantillons utilisés au cours de ce processus appelé l'ensemble d'apprentissage. Toute méthode qui incorpore des informations provenant des échantillons de la formation dans la classification des modèles emploie l'apprentissage. Il existe trois types de base de méthodes d'apprentissage en fonction de la nature de la tâche de reconnaissance de formes.

**Apprentissage supervisé :** Dans l'apprentissage supervisé, les étiquettes de classe ou les coûts des échantillons de formation sont connus avant le début de la phase de formation. La phase de formation calcule le modèle qui minimise le coût total pour les modèles d'ensemble d'entraînement. Ce genre d'apprentissage implique le travail humain. Il est généralement la méthode d'apprentissage la plus courante, et il a de nombreuses applications dans la reconnaissance.

**Apprentissage non supervisé :** dans l'apprentissage non supervisé, les échantillons de l'ensemble de formation ne sont pas étiquetés, et l'objectif principal est de décrypter les similitudes sous-jacentes et de regrouper des modèles similaires ensemble.

L'apprentissage non supervisé ne nécessite pas de travail humain pour l'étiquetage, et il a de nombreuses applications en ingénierie, telles que la segmentation d'images et la télédétection multi-spectrale.

#### **1.2.4 Classification**

La tâche de la classification est d'attribuer le vecteur de caractéristiques fourni par l'extracteur de caractéristiques à une classe. La sortie du classificateur est typiquement une sélection discrète d'une des classes prédéfinies. Tous les composants précédents d'un système de reconnaissance de formes sont conçus et accordés pour améliorer les performances du classificateur. Le degré de difficulté de la classification dépend des relations de similarité entre les modèles des différentes classes. Par conséquent, son succès est considérablement affecté par l'étape d'extraction de caractéristiques.

De ces phases, il s'est avéré que l'extraction des caractéristiques est la plus critique du fait que la discrimination influence directement l'efficacité de la tâche de

classification : les caractéristiques qui distinguent véritablement les groupes aident à l'identification, tandis que l'absence de telles caractéristiques peut entraver la tâche de classification. En arrivant à une identification précise.

La sélection de caractéristiques, bien que utile pour minimiser l'effort d'extraction de caractéristiques, est souvent reléguée à la tâche de classification de sorte que l'utilité de chaque entité puisse être évaluée dans le contexte du processus de discrimination. Le résultat final de la tâche de description est un ensemble de caractéristiques, communément appelé vecteur de caractéristique, qui constitue une représentation des données. La tâche de classification utilise un classificateur pour mapper un vecteur de caractéristique à un groupe. Une telle cartographie peut être spécifiée à la main ou, plus communément, une phase d'apprentissage est utilisée pour induire la cartographie à partir d'une collection de vecteurs caractéristiques connus pour être représentatifs des divers groupes parmi lesquels la discrimination est effectuée (c'est-à-dire l'ensemble d'apprentissage). Une fois formulé, le mappage peut être utilisé pour attribuer une identification à chaque vecteur caractéristique non marqué présenté ultérieurement au classificateur.

La généralité de l'architecture de description et de classification associée à la flexibilité qu'offre la phase d'indexations rend les systèmes automatisés de reconnaissance de formes utiles pour résoudre un large éventail de problèmes réels. Différents algorithmes peuvent être utilisés pour les tâches de description et de classification pour mettre en œuvre un système de reconnaissance de motif approprié à un domaine et à une application particuliers. Différentes combinaisons d'algorithmes se sont avérées efficaces.

### **1.3 Descripteurs 2D et 3D**

Le problème fondamental de la reconnaissance des objets 3D et de ces projections est de fournir une représentation symbolique de haut niveau de son contenu. Chaque objet doit être résolu et classé, attribué à une catégorie spécifique. En plus de la structure tridimensionnelle d'un objet, il existe plusieurs autres facteurs qui influencent l'image de l'objet, par exemple l'éclairage, la projection en perspective et le bruit du capteur. L'indexation et la recherche d'objets visent à résoudre le problème malgré ces écarts. Dans ce travail, l'accent sera mis sur la

reconnaissance des objets en fonction de leur forme car la forme est assez indépendante des phénomènes comme l'éclairage, le contraste et la couleur.

Plusieurs approches ont été décrites dans la littérature. Cependant, il est extrêmement difficile de les analyser de manière exhaustive et de les sortir de la portée de ce travail. Nous nous sommes concentrés sur les descripteurs plus fréquents dans la reconnaissance d'objet 3d par projections et sur les descripteurs qui caractérisent directement l'objet 3D.

Les descripteurs d'objets 3Ds basés sur les caractéristiques, qui sont les plus populaires, peuvent être classés en deux approches fondamentales : globales et locales. Chaque approche utilise différentes techniques pour mettre en œuvre la tâche de la description. Les approches hybrides, parfois appelées une approche unifiée combinent à la fois des techniques locales et globales au sein d'un système de la reconnaissance des modèles.

Des efforts ont également été faits pour combiner les propriétés locales et globales de l'objet. Alosaimi et al. [7] ont combiné les informations dans un histogramme 2D et ont utilisé les coefficients PCA des histogrammes concaténés pour former un vecteur caractéristique unique.

Liu et al. [8] a représenté une forme 3D globale comme la configuration spatiale d'un ensemble de traits locaux. La configuration spatiale a été représentée par le calcul des distributions des distances euclidiennes entre des paires de grappes de forme locales, représentées par des images de spin.

### **1.3.1. Approches globales**

Les premiers travaux sur la représentation d'objets 3Ds et son application à la reconnaissance et à la classification se sont concentrés sur les caractéristiques globales et les approches de distribution de fonctionnalités globales.

Les caractéristiques globales calculées pour représenter les objets 3Ds comprennent la zone, le volume et les moments. Elad et al. [9] a calculé les moments de l'objet et utilisé la valeur des vecteurs des moments comme descripteur de l'objet.

Osada et al. [10] ont calculé un certain nombre de distributions de formes globales pour représenter des objets 3D. Les fonctions de forme mesurées comprenaient l'angle entre trois points aléatoires (A3), la distance entre un point et un point aléatoire (D1), la distance entre deux points aléatoires (D2), la zone du triangle entre trois points aléatoires (D3), et le volume entre quatre points aléatoires sur la surface (D4). Ohbuchi et al. [11] ont amélioré la fonction de forme D2 en mesurant non seulement la distance, mais aussi l'orientation mutuelle des surfaces sur lesquelles la paire de points est située. Zaharia et al. [12] a introduit un descripteur de spectre de forme 3D qui a calculé la distribution de l'indice de forme des points sur l'ensemble du maillage.

Des distributions similaires ont également été calculées pour d'autres propriétés de surface telles que la courbure. Certains travaux récents continuent d'utiliser l'approche de distribution de caractéristiques. Mahmoudi et al. [13] ont calculé l'histogramme des distances de diffusion par paires entre tous les points, tandis que Ion et al. [14] ont défini leur descripteur comme l'histogramme de la transformée d'excentricité. L'histogramme utilise la distance géodésique maximale d'un point à tous les autres points de la surface. Les méthodes d'entité globales sont calculées efficacement, car elles réduisent l'espace de calcul de l'objet 3D en décrivant l'objet avec moins de dimension ; cependant ces méthodes ne sont pas suffisamment discriminatoires lorsque les objets ont de petites différences, comme dans le cas de classification d'objets très similaires.

Les représentations de la carte spatiale décrivent l'objet 3D en capturant et en préservant les emplacements physiques. Saupe et al. [15] ont décrit une fonction de mesure sphérique en calculant la mesure maximale d'une forme dans tous les rayons de l'origine. Ils ont comparé deux types différents de représentation de la fonction : en utilisant des harmoniques sphériques et des moments. Leurs résultats ont montré que l'utilisation d'harmoniques sphériques pour représenter la fonction a donné une meilleure performance. Les coefficients d'harmoniques sphériques reconstruisent une approximation de l'objet à différentes résolutions. Kazhdan et al. [16] ont utilisé cette idée pour montrer que les harmoniques sphériques peuvent être utilisés pour transformer des descripteurs de forme dépendant de la rotation en indépendants de la rotation sans avoir besoin de poser normaliser les objets à l'avance. Leurs résultats

ont montré que l'application de la représentation harmonique sphérique améliorait la performance de la plupart des descripteurs de fonctions sphériques. Laga et al. [17, 18] ont échantillonné uniformément des points sur une sphère unitaire et ont utilisé des transformées en ondelettes sphériques pour représenter des objets 3D. Les descripteurs d'ondelettes sphériques sont des extensions naturelles des moments Zernike 3D et des harmoniques sphériques ; Elles offrent une meilleure localisation des caractéristiques et une invariance de rotation puisque l'analyse des harmoniques sphériques présente des singularités à chaque pôle de la sphère.

Les ondelettes sont des fonctions de base qui représentent un signal donné à plusieurs résolutions. Laga et al. ont étudié les deux ondelettes de deuxième génération, y compris les ondelettes linéaires et papillons sphériques avec un système de levage, et les ondelettes d'image avec des règles sphériques d'extension pour construire le descripteur de forme [19, 20]. Il a proposé trois descripteurs basés sur les ondelettes sphériques : en utilisant les coefficients comme vecteurs caractéristiques, en utilisant l'énergie des coefficients. Zhenbao et al. [21] ont comparé leur analyse par ondelettes multi-résolution au descripteur d'ondelettes sphériques et ont montré que leur descripteur était légèrement meilleur. Leur procédé caractérise l'orientation de forme de l'objet en définissant six plans de vue et échantillonne l'orientation de forme de chacun des plans de vue. Ils ont ensuite effectué une analyse par ondelettes multi-résolution sur chacun des plans de vue et ont utilisé les coefficients d'ondelettes pour chacun des plans de vue comme vecteur caractéristique.

Assfalg et al. [22] a capturé la forme d'un objet 3D en utilisant la carte de courbure de la surface de l'objet. Enfin, Tangelder et coll. [23] ont développé une carte spatiale 3D en divisant l'objet 3D en une grille 3D avec des cellules de tailles égales et en mesurant la propriété de courbure dans chaque cellule.

### **1.3.2 Approches locales**

Des recherches récentes commencent à se concentrer davantage sur l'approche locale pour la représentation d'objets 3D, car cette approche a un pouvoir discriminatif plus fort lors de la différenciation d'objets qui sont similaires dans leur forme globale

[24]. Les traits locaux sont souvent des points considérés comme intéressants ou saillants sur l'objet 3D. Ces points sont calculés de différentes façons.

Certaines méthodes sélectionnent au hasard des points sur la surface de l'objet. Forme et al. [25] qui ont développé un contexte de forme 3D et Johnson et al. [26] qui ont conçu des descripteurs d'image de spin, les deux points sélectionnés au hasard comme leurs points de base. Shilane et al. [27, 28] ont utilisé des points aléatoires avec des descripteurs de forme harmonique à quatre échelles différentes. La plupart des autres méthodes utilisent les propriétés géométriques locales de l'objet 3D telles que la courbure ou les normales pour décrire les points sur la surface de l'objet et définissent les différences de niveau extrêmes comme points saillants. Lee et al. [29] ont utilisé des propriétés de courbure moyenne avec le mécanisme de centre-cercle pour identifier les extrêmes comme points saillants finaux.

Une méthode similaire a été adoptée par Li et al. [30, 31] qui ont trouvé les points saillants fiables en considérant un ensemble d'extrema pour une représentation d'échelle d'une surface d'entrée ponctuelle et ont utilisé les emplacements d'extrêmes de différence de niveau comme points caractéristiques saillants. Unnikrishnan et al. [32] a présenté un détecteur de région d'intérêt à plusieurs échelles qui capture la variation de forme en un point par rapport à la taille de son voisinage. Leur méthode a utilisé les extrema de la courbure moyenne pour identifier les points saillants. Watanabe et al. [33] ont utilisé des extrêmes saillants des courbures principales le long des lignes de courbure à la surface. Castellani et al. [34] ont proposé une nouvelle méthodologie pour détecter et faire correspondre les points saillants basés sur la mesure de la quantité d'un sommet est déplacé après le filtrage. Les points saillants sont décrits en utilisant une description locale basée sur un modèle de Markov caché.

Ohbuchi et al. [35] a rendu des vues multiples d'un modèle 3D et extrait des fonctionnalités locales de chaque vue en utilisant l'algorithme SIFT. Les traits locaux ont ensuite été intégrés dans un histogramme en utilisant une approche par sac de caractéristiques pour la reconnaissance. Novatnack et al. [36] [37] a extrait les coins et les bords d'un modèle 3D en paramétrant d'abord la surface d'un modèle de maille 3D sur une carte 2D et en construisant une carte normale de surface dense. Ils ont ensuite construit un espace discret par convolution la carte normale avec les noyaux

gaussiennes d'augmenter l'écart type. Les angles et les bords détectés à des échelles individuelles ont été combinés en une représentation unifiée pour l'objet 3D.

Akagunduz et al. [38] ont utilisé une pyramide gaussienne à plusieurs échelles pour extraire les extrema de surface et ont représenté les points et leurs relations par un modèle graphique. Taati et al. [39] ont généré un descripteur de forme local basé sur des propriétés invariantes extraites de l'espace de composant principal du voisinage local autour d'un point. Les points saillants ont été sélectionnés sur la base de rapports de propriétés de dispersion basiques. D'autres exemples de descripteurs locaux sont présentés dans les sous sections suivantes :

### ➤ **Descripteurs basés sur des histogrammes locaux**

Ces descripteurs ont été développés en premier, suite au succès de méthodes similaires dans l'analyse d'images 2D.

**Spin image** : L'idée de spin images [40, 41] est d'associer à chaque point d'une forme une image qui décrit le contexte local de ce point dans la forme. Des techniques d'appariement d'images peuvent ensuite être utilisées pour correspondre aux descripteurs et ainsi effectuer une adaptation de surface. L'entrée nécessaire pour calculer les images de spin est un nuage de points avec les normales associées à chaque point. Pour chaque point du nuage, un système de coordonnées cylindriques est défini avec le point décrit au centre et sa normale comme axe du cylindre. Cette définition est ambiguë uniquement pour la coordonnée angulaire, et les deux coordonnées de distance sont bien définies. Un histogramme 2D de ces coordonnées pour tous les points de la forme est calculé et utilisé comme descripteur.

**Contexte de la forme** : Le contexte de la forme a été introduit dans [42] pour décrire les lignes d'une image. Semblable aux images de spin, il stocke pour chaque point la distribution des positions relatives des autres points. Dans un contexte de forme, l'histogramme se fait de façon logarithmique et polaire, ce qui présente plusieurs avantages, notamment en accordant plus d'importance aux points de rapprochement et en transformant les rotations et la mise à l'échelle de la forme en une traduction du descripteur. Cet article [42] montre également les résultats de reconnaissance d'objet 3D en utilisant le contexte de forme sur un ensemble de vues du modèle 3D.

L'idée de contexte de forme a été étendue d'une manière plus axée sur les principes aux formes en 3D par [43, 44].

**HOG** : L'idée de HOG de la forme [45] est similaire au contexte de la forme en ce sens qu'elle calcule des histogrammes en coordonnées log-polaires, mais elle vise à décrire une texture sur une forme plutôt que la forme elle-même. En conséquence, il stocke les histogrammes des orientations du gradient dominant de la texture projetée pour chaque case au lieu de la densité des points.

### ➤ **Signatures spectrales**

Dans la plupart des cas, on peut récupérer toutes les informations intrinsèques sur une forme en utilisant les valeurs propres et les fonctions propres de l'opérateur de Laplace-Beltrami sur la forme [46].

Ces fonctions propres sont généralement liées à plusieurs phénomènes physiques, y compris les modes de vibration. Les deux signatures spectrales les plus importantes, la signature de point global et la signature de noyau de chaleur sont présentées dans cette section. La signature du noyau d'onde, se trouve dans la même catégorie de descripteurs spectraux.

**Signature de point global (spg)** : La première signature de point spectral a été développée par [47]. Pour signaler une forme, la signature de point global associe un vecteur. Sa  $k^{\text{ème}}$  composante est la valeur de la  $k^{\text{ème}}$  fonction propre au point décrit divisée par la racine carrée de la norme de  $k^{\text{ème}}$  valeur propre. Cette division donne plus d'importance aux vecteurs propres associés aux basses fréquences. Le principal inconvénient du (spg) est que si une forme est légèrement modifiée, l'ordre des fonctions propres peut être modifié, ce qui donne deux signatures complètement différentes.

**Signature du noyau de chaleur (snc)** : La signature du noyau de chaleur [48] est similaire à la signature de point global, il est défini en utilisant la décomposition propre de l'opérateur de Laplace-Beltrami sur la forme. Cependant, cette signature ne sépare pas les fonctions propres mais les combine d'une manière qui provient naturellement de l'analyse de la diffusion de la chaleur sur la surface de la forme. Il y a eu plusieurs extensions de snc, en particulier, [49] a modifié cette signature pour

être invariant au changement d'échelle. [50] a considéré la diffusion de chaleur dans le volume de forme plutôt que sur sa surface, en définissant un (snc) volumétrique.].

### 1.3.3 Approches statistique

La reconnaissance statistique des objets [51] [52] [53] s'inspire des concepts établis dans la théorie de la décision statistique pour discriminer les données de différents groupes en fonction des caractéristiques quantitatives des données. Il existe une grande variété de techniques statistiques qui peuvent être utilisées dans la tâche de description pour l'extraction de caractéristiques, allant de statistiques descriptives simples à des transformations complexes.

Les caractéristiques quantitatives extraites de chaque objet pour la reconnaissance statistique de motifs sont organisées en un vecteur caractéristique de longueur fixe où la signification associée à chaque entité est déterminée par sa position dans le vecteur (c'est-à-dire que la première caractéristique décrit une caractéristique particulière des données, Deuxième caractéristique décrit une autre caractéristique, et ainsi de suite). La collection de vecteurs de caractéristiques générés par la tâche de description est transmise à la tâche de classification. (P. Ex., Règle de Bayes), les limites (p. Ex. Les arbres de décision, les réseaux de neurones) et le regroupement (p. Ex. K-moyen, hiérarchique).

Des exemples de techniques d'extraction de caractéristiques statistiques les plus utilisés sont les suivants :

**Transformée de Hough** [53], Le principe de cette transformée s'appuie sur l'accumulation de différentes primitives élémentaires paramétrées selon des variables spécifiques.

Zaharia et al. [54], ont développé un descripteur dérivé de la transformée de Hough 3D. Son principe consiste à accumuler des points sur des plans dans l'espace. En effet, un plan dans l'espace euclidien peut être transformé, dans l'espace de Hough, en un seul point défini par les coordonnées sphériques. Par conséquent, une collection de plans dans l'espace euclidien correspond à une collection de points dans l'espace de Hough.

**La transformée de Fourier** proposée par [55], permet de caractériser l'information spatiale des objets 3D dans l'espace fréquentiel. L'extraction des primitives se fait après une étape de normalisation en utilisant une ACP continue et une voxelisation du modèle 3D. Les composantes du vecteur primitives sont les valeurs absolues des coefficients de la transformée de Fourier Discrète 3D appliquée aux voxels. Ces descripteurs sont invariants à la translation, la rotation, le changement d'échelle et la réflexion.

#### 1.3.4 Méthodes graphiques

Alors que les méthodes basées sur les caractéristiques utilisent uniquement les propriétés géométriques du modèle 3D pour définir la forme de l'objet, les méthodes basées sur les graphiques utilisent les informations topologiques de l'objet 3D pour décrire sa forme. Le graphique qui est construit montre comment les différents composants de forme sont liés ensemble. Les représentations de graphe incluent des graphiques de modèle, des graphiques de Reeb, et des graphiques de squelette. Ces procédés sont connus pour être coûteux en termes de calcul et sensibles à de petits changements topologiques. Sundar et al. [56] ont utilisé le graphique squelettique comme descripteur de forme pour coder à la fois les propriétés géométriques et topologiques de l'objet 3D. Les mesures de similarité entre deux objets ont été approchées à l'aide d'un algorithme gourmand pour l'appariement de graphes bipartites. Hilaga et al. [57] ont introduit l'utilisation de graphiques Reeb pour correspondre aux formes des modèles articulés.

#### 1.3.5 Méthodes basées sur la vue

Le descripteur de forme basé sur la vue le plus efficace est le descripteur Light Field développé par Chen et al. [58]. Un champ lumineux autour d'un objet 3D est une fonction 4D qui représente l'éclat à un point 3D donné dans une direction donnée. Chaque champ lumineux 4D d'un objet 3D est représenté comme une collection d'images 2D rendues à partir d'une matrice 2D de caméras réparties uniformément sur une sphère. Leur méthode place les caméras de lumière sur 20 sommets d'un dodécaèdre régulier et utilise la projection orthogonale pour capturer 10 silhouettes différentes du modèle 3D. Dix rotations différentes sont effectuées

pour capturer un ensemble de descripteurs de champ lumineux pour améliorer la robustesse pour la rotation. Les 100 images rendues sont ensuite décrites en utilisant des moments de Zernike et des descripteurs de Fourier pour décrire la forme de la région et la forme du contour, respectivement du modèle 3D. La reconnaissance des modèles 3D est effectuée par étapes où les objets qui sont très dissemblables par rapport au modèle de requête sont rejetés au début du processus. Ceci est fait en comparant seulement un sous-ensemble des descripteurs de champ de lumière de la requête et des objets de base de données dans les premières étapes du processus de reconnaissance. Le descripteur de champs lumineux a été évalué comme l'un des descripteurs les plus performants dans le cadre du concours SHREC.

Ohbuchi et al. [35] ont utilisé une approche similaire basée sur les vues du descripteur de champ lumineux. Cependant, leur méthode a extrait les caractéristiques locales de chaque image rendue en utilisant l'algorithme SIFT. Wang et al. [59] a amélioré l'efficacité d'utilisation de l'espace du descripteur de LFD en projetant un certain nombre de points aléatoires uniformément échantillonnés le long de six directions pour créer six images qui sont ensuite décrites en utilisant des moments de Zernike. Ils ont également utilisé une méthode d'extraction pour accélérer le processus de reconnaissance. Les résultats expérimentaux sur la base de données de référence de la forme de Princeton ont montré que la performance de leur méthode était comparable au descripteur LFD pour certaines catégories.

Vajramushti et al. [60] ont utilisé une combinaison d'une technique de tampon de profondeur basée sur la vue et d'un descripteur de volume basé sur des caractéristiques pour la correspondance 3D. Leur méthode a utilisé le volume voxel des objets pour réduire l'espace de recherche pour les comparaisons de profondeur-tampon. Vranic [61] a évalué un descripteur composite appelé DESIRE qui a été formé en utilisant des images de profondeur de champ, des silhouettes et des extensions de rayons d'un objet 3D. Ses résultats ont montré que DESIRE a surpassé LFD en reconnaissance des objets de certaines catégories.

D'autres approches ont utilisés les descripteurs 2D sur les vues caractéristiques de l'objet 3D. Parmi ces descripteurs on cite les plus fréquents :

GIST [62], ce descripteur divise l'image en blocs typiquement 4x4 et pour chaque bloc stocke l'énergie associée à différentes orientations (typiquement 8) à différentes échelles (typiquement 3). Il est conçu pour représenter la forme de la scène et évite de regarder des informations très locales. Ainsi, il est robuste à des changements importants dans l'apparence scène locale tels que ceux induits par le changement de style de représentation.

Méthodes basées sur des exemples. Les méthodes basées sur un exemple s'appliquent à l'image associée à l'idée développée pour la reconnaissance des catégories. D'un seul exemple positif, ils apprennent un classificateur [63]. S'il est utilisé avec un descripteur d'une image en entrée, il a été montré qu'il récupère des images de la même instance malgré des changements dans le style de représentation [64]. C'est principalement parce que le classificateur se concentre sur les parties les plus discriminatives, qui sont susceptibles d'être présents dans toutes les représentations de la scène.

## 1.4 Conclusion

Dans ce chapitre, nous avons présenté le contexte général des travaux sur la reconnaissance d'objets 3D. Premièrement nous avons défini le fonctionnement d'un processus complet du système de reconnaissance de forme. Nous avons alors constaté que grâce au choix d'approches (dites "2D-3D", "3D"), nous allons pouvoir offrir de nombreux moyens de requêtes, qui mènent à une extraction efficace et effective des informations ou des formes contenues dans un objet 3D. Ensuite, nous avons élaboré une description plus détaillée sur les méthodes de description de la forme d'un modèle 3D. Grâce à la flexibilité et la robustesse des résultats obtenus par les méthodes 2D pour des images couleurs 2D, une étude plus profonde a été élaborée. Par conséquent, il s'est avéré que ces méthodes sont assez limitées pour traiter des images 3D. Cela nous a poussés à étudier d'autres méthodes plus appropriées, à savoir : les méthodes 3D. Nous présentons en détails quelques avantages et inconvénients des approches 3D et 2D-3D dans le tableau 1.1.

	Avantages	Inconvénients
Approches 3D	<ul style="list-style-type: none"> <li>- Souligne la structure 3D</li> <li>- Invariance aux transformations géométriques (transformations euclidiennes et aux conditions d'acquisition (éclairage))</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité et temps de calcul (exemple : génération du maillage)</li> </ul>
Approches 2D-3D	<ul style="list-style-type: none"> <li>- Réduction de la complexité en temps de calcul</li> <li>- Profiter des approches existantes du domaine 2D</li> </ul>	<ul style="list-style-type: none"> <li>- Sensibilité à la variation de la pose (rotation)</li> <li>- Possibilité de perte d'information (par cause de la structure 2D)</li> </ul>

Tableau 1.1 : Avantages et inconvénients des approches 3D et 2D

# **CHAPITRE 2 :**

# **Extraction des**

# **caractéristiques**

## 2.1 Introduction

L'extraction de paramètres caractéristiques en anglais, « feature extraction » est une étape de grande importance. Si elle est mal conçue, il sera difficile, voire impossible, d'effectuer une reconnaissance efficace.

Dans la première partie de ce chapitre nous allons caractériser la forme des objets couleurs par des paramètres que nous voyons pertinents pour leur forme. Pour cela nous nous sommes principalement intéressés aux méthodes d'analyses de formes basées régions : les moments de Zernike, les descripteurs CENTRIST, ACP, (CENTRIST +M. Zernike, ACP (M. Zernike), ACP (CENTRIST)).

La deuxième partie a pour but de présenter à la fois la théorie des approches 3D/3D basées sur le calcul numérique des distances géodésiques sur les variétés riemanniennes. La notion de variété riemannienne qui est un cadre fédérateur pour tous les problèmes examinés dans le présent document. Cette notion permet de définir une métrique locale qui code les informations sur le problème qu'on veut résoudre, et ensuite intégrer sur l'ensemble du domaine pour obtenir une distance entre des paires de points.

## 2.2 Descripteurs 2D

Les images numériques représentent plusieurs informations. Cependant, reconnaître une image dans une base de données est une tâche difficile. Ce qui conduit donc à l'utilisation d'une représentation de dimension inférieure pour caractériser le contenu de ces images. L'objectif essentiel de l'indexation est de déterminer une représentation compacte et rapidement accessible pour chaque image. Il est alors souhaité d'employer des attributs invariants aux transformations telles que le changement d'éclairage et le changement d'échelle. Ceci permet de prendre en considération aux différentes transformations que peut subir une image d'un objet donné. Toutefois, la caractérisation et l'identification des images restent un grand défi en traitement d'images.

### 2.2.1 Descripteur CENTRIST

Récemment, Wu et Rehg [65] proposent un descripteur visuel appelé CENSus TRansform hISTogramme (CENTRIST), qui est une représentation holistique modélisant la distribution des structures locales. Des informations géométriques approximatives sont capturées en décrivant le CENTRISTE extrait des pyramides spatiales [66]. La transformation de recensement [67] compare la valeur d'intensité d'un pixel à celle de ses huit pixels spatialement voisins. Si l'intensité du pixel central est supérieure à l'un de ses voisins, un bit 1 est positionné dans la position correspondante. Sinon, un bit 0 est défini. De gauche à droite à fond, ces bits sont concaténés pour former une représentation binaire, qui peut être évaluée à un nombre dans la base dix, appelé valeur de transformation de recensement (valeur CT) pour le pixel central.

Après avoir évalué la valeur CT pour chaque pixel, l'histogramme des valeurs CT est construit pour former le descripteur CENTRIST.

### 2.2.2 Moments de Zernike

Les moments de Zernike ont été largement utilisés dans la reconnaissance de forme vu leur propriétés d'orthogonalité et d'invariance en translation, en rotation et en échelle. Ces moments ont été introduits par Teague [68] et qui sont basés sur les fonctions polynomiales de Zernike. Wallin [69] a décrit un algorithme pour la formation d'invariants de rotation de n'importe quel ordre. Les propriétés numériques et les applications possibles des moments de Zernike dans le traitement d'image ont été décrites dans les références [70] et [71].

Les moments de Zernike l'ordre  $p$  et de répétition  $q$  sont définis par l'expression suivante :

$$Z_{pq} = \frac{p+1}{\pi} \iint_{xy} f(x, y) \cdot [V_{pq}(r, \theta)]^* dx dy \quad (2.1)$$

avec  $V_{pq}(r, \theta)$  représentent les polynômes de Zernike en coordonnée polaire de l'ordre  $p$  et de répétition  $q$ .

$$\text{et} \quad V_{pq}(r, \theta) = \mathcal{R}_{pq}(r) e^{-iq\theta} \quad (2.2)$$

Où  $\mathcal{R}(r)$  sont des polynômes radiaux de la forme :

$$\mathcal{R}_{pq}(r) = \sum_{k=|q|}^p r^k \frac{(-1)^{(p-k)/2} \cdot (p+k)!}{\left(\frac{p-k}{2}\right)! \left(\frac{q+k}{2}\right)! \left(\frac{k-q}{2}\right)!} \quad (2.3)$$

Nous avons adopté une formulation discrète de ces moments pour notre application. Cette dernière a été développée dans [72], où l'équation (2.1) est réécrite sous la forme suivante :

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) [V_{pq}(x, y)]^* \quad (2.4)$$

où  $x^2 + y^2 \leq 1$

### 2.2.3 Analyse en Composantes Principales (ACP)

ACP est une méthode de la famille de l'analyse des données et plus généralement de la statistique multivariée, qui consiste à transformer des variables liées entre elles (dites "corrélées" en statistique) en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées "composantes principales", ou axes. Elle permet au praticien de réduire l'information en un nombre de composantes plus limité que le nombre initial de variables.

Il s'agit d'une approche à la fois géométrique (représentation des variables dans un nouvel espace géométrique selon des directions d'inertie maximale) et statistique (recherche d'axes indépendants expliquant au mieux la variabilité – la variance - des données). Lorsqu'on veut alors compresser un ensemble de N variables aléatoires, les n premiers axes de l'ACP sont un meilleur choix, du point de vue de l'inertie ou la variance.

#### a) Objectif de l'Analyse en Composantes Principales (ACP)

Etant donné un ensemble d'observations décrites par des variables exclusivement numériques  $X=\{x_1, x_2, \dots, x_p\}$ , l'ACP a pour objectif de décrire ce même ensemble de données par de nouvelles variables en nombre réduit. Ces

nouvelles variables seront des combinaisons linéaires des variables originales, et porteront le nom de Composantes Principales (CP).

En général, la réduction du nombre de variables utilisées pour décrire un ensemble de données provoque une perte d'information. L'ACP procède de façon à ce que cette perte d'information soit la plus faible possible.

L'Analyse en Composantes principales peut donc être vue comme une technique de réduction de dimensionnalité.

### **b) Description de l'algorithme de l'ACP**

L'algorithme est comme suit :

- ✓ Centrer les données
- ✓ Construire la matrice de covariance  $\Sigma$
- ✓ Décomposer cette matrice en vecteur propres, valeur propres  $\{p_i, \lambda_i\}$
- ✓ Ordonner les valeurs propres par ordre décroissant
- ✓ Le sous-espace de dimension  $q$  qui représente au mieux les données au sens de l'erreur quadratique moyenne est engendré par la matrice :

$$P = (p_1, p_2, \dots, p_{n-1}, p_q) \quad (2.5)$$

Où  $\{p_1, p_2, \dots, p_{n-1}, p_q\}$  sont les  $q$  vecteurs propres associés aux  $q$  plus grandes valeurs propres. L'ensemble des composantes principales s'écrit matriciellement :

$$C = XP = (c_1, c_2, \dots, c_{n-1}, c_q) \quad (2.6)$$

## **2.3 Equation Eikonale**

Les méthodes de points itératifs les plus proches ont été les premières à résoudre des formes rigides en 3D [73]. Elles ont ensuite été étendues pour faire face à certaines déformations non rigides [74] en alignant itérativement les formes et en les déformant utilisant une transformation paramétrique non rigide. Cette idée ne peut toutefois fonctionner qu'avec des déformations limitées en termes de distance euclidienne dans l'espace 3D.

Pour la correspondance de formes, les propriétés intrinsèques de la forme, telles que les distances géodésiques, sont plus significatives. En effet, elles sont le plus souvent

préservées sous des déformations habituelles. Par exemple, la distance géodésique entre les deux mains d'un corps humain restera approximativement la même malgré le changement de leur distance dans l'espace 3D. Cela conduit à l'idée de voir les formes comme des espaces métriques. Si la distance utilisée est la distance euclidienne dans l'espace 3D, la distance naturelle entre les formes est la distance de Hausdorff dans l'espace euclidien 3D. Cependant, d'autres distances telles que la distance géodésique sont plus significatives.

Cette section concerne le rapprochement des distances géodésiques et courbes géodésiques avec schémas numériques rapides. Cela exige une discrétisation de la variété riemannienne en utilisant soit une grille uniforme ou un maillage triangulaire. L'importance est que la distance géodésique à un ensemble de points de départ répond à une équation différentielle non linéaire, l'équation Eikonale, qui est résolue pour calculer numériquement la distance géodésique [75]. Le champ de tenseurs locaux et la distance géodésique sont utilisés pour résoudre le problème d'intérêt pratique tel que la reconnaissance d'objets 3D. Le plus court chemin pour cette distance de Riemann, les soi-disant géodésiques, sont également importants parce qu'ils suivent des structures curvilignes remarquables dans le domaine [75].

Le problème discret non-linéaire peut être résolu par les systèmes itératifs, et dans certains cas en utilisant des méthodes de propagation plus rapide avant. Plus de détails sur la résolution de ce problème seront présentés plus tard dans cette section.

Nous commençons la description de ces algorithmes numériques par un simple paramètre où la distance géodésique est calculée sur une grille régulière pour une métrique isotrope. Ce paramètre restreint est utile car l'équation eikonale est discrétisé en utilisant les différences finies, qui permet d'introduire plusieurs algorithmes importants tels que propagations de Fast Marching.

### **Variétés Riemanniennes**

De manière traditionnelle [75], les méthodes de calcul du chemin géodésique sont classées en deux catégories, selon si on calcule la distance entre un sommet source et l'ensemble des autres sommets ou la distance entre une paire de sommets fixés que l'on appellera point de départ et point d'arrivé. La méthode la plus connue est sans doute l'algorithme de Dijkstra [76] qui permet de manière exhaustive de

trouver le chemin le plus court entre un sommet source et tous les autres sommets du maillage.

Dans ce qui suit on va voir quelques définitions de base de la métrique locale (un champ de tenseur) sur une variété riemannienne et la notion associée de la distance géodésique et chemins minimaux. Le point important est que la distance géodésique à un ensemble de points de départ répond à une équation différentielle non linéaire, l'équation eikonale, qui est résolu pour calculer numériquement la distance géodésique [77].

Soit  $\Omega$  un domaine de  $\mathbb{R}^2$  et  $\varphi: \Omega \rightarrow \mathbb{R}^3$  une application dérivable et injective.  $\mathcal{M} = \varphi(\Omega)$  s'appelle surface paramétrée au-dessus de  $\Omega$  par  $\varphi$ .

### 2.3.1 Définition de la courbe

Soit  $\Omega$  le domaine de la paramétrisation  $\varphi$ .

- Une courbe du domaine  $\Omega$  est définie comme une application :

$$t: [0,1] \rightarrow \gamma(t) \in \Omega \quad (2-10)$$

- La réalisation géométrique de la courbe  $\gamma$  sur la surface  $\varphi(\Omega)$  est donnée par :

$$\bar{\gamma}(t) = \varphi(\gamma(t)) \quad (2-11)$$

Le calcul de la longueur de  $\gamma$  dans l'espace k-dimensionnelle  $\mathbb{R}^k$  suit la définition habituelle, mais de faire le calcul sur le domaine paramétrique, il faut utiliser une métrique locale défini comme suit.

### 2.3.2 Définition première forme fondamentale

Le tenseur métrique est défini sur la surface paramétré  $\varphi(\mathcal{D})$  par:

$$I_\varphi = \left( \left\langle \frac{\partial \varphi}{\partial x_i}, \frac{\partial \varphi}{\partial x_j} \right\rangle \right)_{i,j=1,2} \quad (2-12)$$

où  $\langle , \rangle$  désigne le produit scalaire.

- Le tenseur métrique permet de définir la longueur de la courbe  $\gamma(t)$  comme suit :

$$L(\gamma) = \int_0^1 W(x) \|\bar{\gamma}'(t)\| dt = \int_0^1 W(x) \sqrt{\gamma'(t)^T I_\varphi(\gamma(t)) \gamma'(t)} dt. \quad (2-13)$$

Supposons qu'on a une longueur  $L$  qui dépend du chemin  $\gamma : L(\gamma)$

On cherche le chemin  $\gamma^*$  qui minimise la longueur  $L$  c.-à-d. correspond à la plus petite longueur :

$$\gamma^* = \underset{\gamma \in P(x,y)}{\operatorname{argmin}} L(\gamma) \quad (2-14)$$

où  $P(x,y)$  est l'ensemble des chemins admissibles joignant  $x$  et  $y$  :

$$P(x,y) = \{\gamma: [0,1] \rightarrow \mathcal{M} \text{ tel que } \gamma(0) = x \text{ et } \gamma(1) = y\} \quad (2-15)$$

La longueur  $L$  correspond au chemin  $\gamma^*$  est appelée la distance géodésique

$$\forall (x,y) \in \mathcal{M}^2, d_{\mathcal{M}}(x,y) = L(\gamma^*) \text{ avec } L(\gamma^*) = \min_{\gamma \in P(x,y)} L(\gamma) \quad (2-16)$$

### 2.3.3 Surfaces comme étant variété Riemannienne

Bien que les courbes appartiennent pas aux mêmes espaces, il est possible de formaliser le calcul des géodésiques de la même manière dans les deux cas. Pour ce faire, il faut introduire la variété Riemannienne  $\Omega \subset \mathbb{R}^2$  associée à la surface  $S$ .

Soit  $\Omega \subset \mathbb{R}^2$  variété Riemannienne associée à la surface  $S \subset \mathbb{R}^3$  qui peut être décrite comme une fonction paramétrique

$$\varphi: \begin{array}{l} \Omega \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3 \\ x \mapsto \bar{x} = \varphi(x) \end{array} \quad (2-17)$$

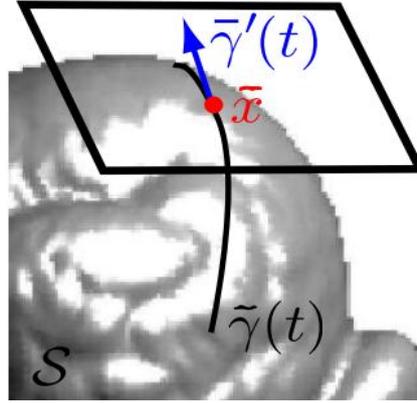


Figure 2.2 : Le plan tangent  $\mathcal{T}_{\bar{x}}$  et le dérivée de la courbe sur la surface  $S$ .

Le plan tangent  $\mathcal{T}_{\bar{x}}$  à un point de la surface  $\bar{x} = \varphi(x)$  est engendré par les deux dérivées partielles de la paramétrisation, qui définissent la matrice dérivée au point  $x = (x_1, x_2)$

$$D\varphi(x) = \left( \frac{\partial \varphi}{\partial x_1}(x), \frac{\partial \varphi}{\partial x_2}(x) \right) \in \mathbb{R}^3 \quad (2-18)$$

Comme le montre la Figure 5.1, le dérivé de  $n'$  importe quelle courbe  $\bar{\gamma}$  à un point  $\bar{x} = \bar{\gamma}(t)$  appartient à la tangente plan  $\mathcal{T}_{\bar{x}}$  de  $S$  à  $\bar{x}$ . La courbe  $\bar{\gamma}(t) \in S \subset \mathbb{R}^3$  définit une courbe de  $\gamma(t)$  comme suit :

$$\gamma(t) = \varphi^{-1}(\bar{\gamma}(t)) \in \Omega \quad (2-19)$$

Depuis  $\bar{\gamma}(t) = \varphi(\gamma(t)) \in \Omega$  les tangentes aux courbes sont liés par l'intermédiaire  $\bar{\gamma}'(t) = D_{\varphi(\gamma(t))}\gamma'(t)$  et  $\bar{\gamma}'(t)$  est dans la tangente plan de  $\mathcal{T}_{\bar{\gamma}(t)}$  qui est engendré par les colonnes de  $\varphi(\gamma(t))$ .

La longueur  $L(\bar{\gamma})$  de la courbe  $\bar{\gamma}$  est calculée comme suite :

$$L(\bar{\gamma}) = \int_0^1 \|\gamma'(t)\|_{T_{\gamma(t)}} dt \quad (2-20)$$

Avec le tenseur métrique  $T_{\gamma(t)}$  est défini par :

$$\forall \gamma(t) \in \Omega, T_{\gamma(t)} = \sqrt{W(\varphi(\gamma(t)))} I_{\varphi}(\gamma(t)) \text{ avec } \bar{\gamma}(t) = \varphi(\gamma(t)) \quad (2-21)$$

et  $I_{\varphi}(x) \in \mathbb{R}^{2 \times 2}$  est la première forme fondamentale de  $S$ .

$$I_{\varphi}(x) = D_{\varphi(x)}^T D_{\varphi(x)} = \left( \left\langle \frac{\partial \varphi}{\partial x_i}, \frac{\partial \varphi}{\partial x_j} \right\rangle \right)_{i,j=1,2} \quad (2-22)$$

Compte tenu que la matrice de  $I_\varphi$  symétrique et positif  $A = (A_{i,j})_{1 \leq i,j \leq d} \in \mathbb{R}^{2 \times 2}$ , nous définissons sa norme associée :

$$\|u\|_A^2 = \langle u, u \rangle_A \quad \text{avec} \quad \langle u, v \rangle_A = \langle Au, v \rangle = \sum_{1 \leq i,j \leq 2} A_{i,j} u_i v_j \quad (2-23)$$

Un domaine  $\Omega$  équipé d'un tel métrique est appelé une variété riemannienne. Pour une dimension arbitraire  $d$ , une variété riemannienne est ainsi décrit localement comme un sous-ensemble de l'espace ambient  $\Omega \in \mathbb{R}^d$ , ayant la topologie d'une sphère ouverte, équipé d'une matrice définie positive  $T_x \in \mathbb{R}^{d \times d}$  pour chaque point  $x \in \Omega$ , que nous appelons le tenseur métrique [77].

De même à (2-23), à chaque point  $x \in \Omega$ , le tenseur  $T_x$  définit la longueur d'un vecteur  $u \in \mathbb{R}^d$

$$\|u\|_{T_x}^2 = \langle u, u \rangle_{T_x} \quad \text{avec} \quad \langle u, v \rangle_{T_x} = \langle T_x u, v \rangle = \sum_{1 \leq i,j \leq d} (T_x)_{i,j} u_i v_j \quad (2-24)$$

Ceci permet de calculer la longueur d'une courbe  $\gamma(t) \in \Omega$  tracé sur la riemannien comme une longueur pondérée lorsque la longueur infinitésimale est mesurée selon  $T_x$ .

$$L(\gamma(t)) = \int_0^1 \|\gamma'(t)\|_{T_{\gamma(t)}} dt \quad (2-25)$$

Preuve :  $(L(\bar{\gamma}) = L(\gamma(t)))$

$$\bar{\gamma}'(t) = \frac{d}{dt} (\varphi(\gamma(t))) = D_{\varphi(\gamma(t))} \gamma'(t)$$

$$\begin{aligned} L(\bar{\gamma}) &= \int_0^1 W(x) \|\bar{\gamma}'(t)\| dt \\ &= \int_0^1 W(x) \sqrt{\bar{\gamma}'^T(t) \bar{\gamma}'(t)} dt \end{aligned} \quad (2-26)$$

$$\text{où} \quad \bar{\gamma}'^T(t) = [D_{\varphi(\gamma(t))} \gamma'(t)]^T = \gamma'^T D_{\varphi(\gamma(t))}^T \quad (2-27)$$

$$\begin{aligned} L(\bar{\gamma}) &= \int_0^1 W(x) \sqrt{\gamma'^T D_{\varphi(\gamma(t))}^T D_{\varphi(\gamma(t))} \gamma'(t)} dt \\ L(\bar{\gamma}) &= \int_0^1 \sqrt{\gamma'^T W^2(x) D_{\varphi(\gamma(t))}^T D_{\varphi(\gamma(t))} \gamma'(t)} dt \end{aligned} \quad (2-28)$$

On pose  $I_\varphi = D_{\varphi(\gamma(t))}^T D_{\varphi(\gamma(t))}$

Donc

$$L(\bar{\gamma}) = \int_0^1 \sqrt{\gamma'^T T_{\gamma(t)} \gamma'} dt$$

Avec

$$\begin{aligned} \sqrt{\gamma'(t)^T T_{\gamma(t)} \gamma'(t)} &= \sqrt{\langle T_{\gamma(t)} \gamma'(t), \gamma'(t) \rangle} \\ &= \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{T_{\gamma(t)}}} = \|\gamma'(t)\|_{T_{\gamma(t)}} \end{aligned}$$

D'où

$$L(\bar{\gamma}) = \int_0^1 \|\gamma'(t)\|_{T_{\gamma(t)}} dt = L(\gamma(t)). \quad (2-29)$$

### 2.3.4 La fonction distance géodésique

La fonction distance  $\mathcal{U}_s$  définie sur  $\Omega$  relativement à l'ensemble de points de départ  $\mathcal{S} = (x_k)_k \subset \Omega$  est donnée par:

$$\forall x \in \mathcal{M}, \mathcal{U}_s(x) = \min_k d_\Omega(x, x_k) \quad (2-30)$$

Si la métrique  $T_{\gamma(t)}$  est continue, alors pour tout ensemble de départ  $\mathcal{S} \subset \Omega$ , la fonction distance  $\mathcal{U}_s$  est l'unique solution de l'équation de Hamilton-Jacobi défini par:

$$\begin{cases} \|\nabla_x \mathcal{U}_s\|_{T_{\gamma(t)}}^{-1} = 1 \\ \mathcal{U}_s(x_k) = 0, \forall k \end{cases} \quad (2-31)$$

La distance géodésique  $\mathcal{U}_s$  entre un point  $x$  et les points de départ  $\mathcal{S}$  calculé en résoudre est l'équation différentielle suivante [77] :

$$\begin{cases} \forall t \geq 0 \frac{d\gamma(t)}{dt} = -\eta_t v(\gamma(t)) \\ \gamma(0) = x \end{cases} \quad (2-32)$$

Avec  $v(\gamma(t))$  est le vecteur tangent donné par :

$$v(\gamma(t)) = T_{\gamma(t)}^{-1} \nabla_x \mathcal{U}_s \quad (2-33)$$

$\eta_t$  est une fonction scalaire qui contrôle la vitesse pour que la vitesse soit unitaire  
 où  $\eta_t > 0$  est une fonction scalaire qui contrôle la vitesse de la paramétrisation géodésique. Pour obtenir une paramétrisation de la vitesse unitaire, on a besoin d'utiliser

$$\|\gamma'(t)\| = 1 \Leftrightarrow \|\eta_t v(\gamma(t))\| = 1 \quad (2-34)$$

avec  $\eta_t > 0$  cela veut que  $\eta_t = \|v(\gamma(t))\|^{-1}$

Alors la courbe géodésique  $\gamma$  qui réalise le minimum entre  $x$  et les points de  $S$  est la solution de l'équation différentielle de Hamilton-Jacobi suivante:

$$\begin{cases} \gamma'(t) = \frac{T_{\gamma(t)}^{-1} \nabla_{\gamma(t)} u_s}{\|T_{\gamma(t)}^{-1} \nabla_{\gamma(t)} u_s\|} \\ \gamma(0) = x \end{cases} \quad (2-35)$$

### 2.3.4 Discrétisation de l'équation Eikonale

On utilisant un maillage triangulaire, le problème discret est reformulé en un problème de point fixe.

Nous considérons un petit voisinage  $B(x)$  de chaque point  $x \in \Omega \setminus S$  tel que  $B(x) \cap S = \emptyset$ .

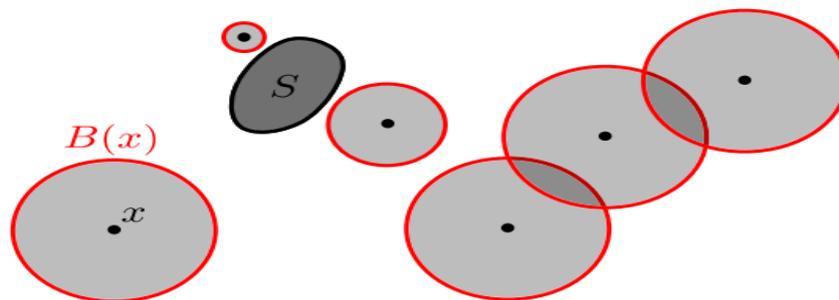


Figure 2.3 : voisinage  $B(x)$  de chaque point  $x \in \Omega$  [77].

$u_s$  est l'unique solution de l'équation :

$$\begin{cases} \forall x \in \Omega, U(x) = \min_{y \in \partial B(x)} U(y) + d(y, x), \\ \forall x \in S, U(x) = 0 \end{cases} \quad (2-36)$$

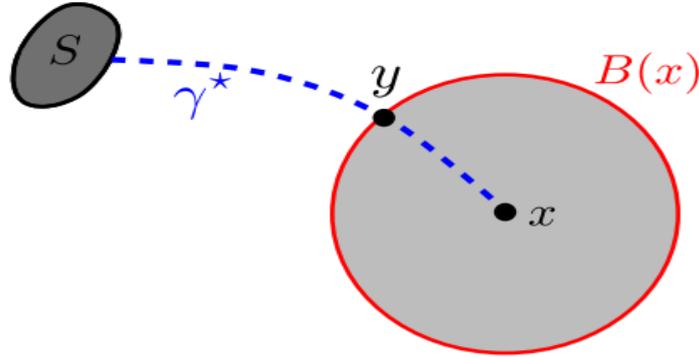


Figure 2.4 : Le point  $y$  qui réalise l'égalité dans la minimisation (2-36)

Où  $d(y, x)$  est la distance géodésique définie précédemment, cette équation peut être vu comme un problème de point fixe :

$$\begin{cases} U = \Gamma(U) \\ U(x) = 0 \text{ si } x \in S \end{cases} \quad (2-37)$$

avec  $\Gamma$  est un opérateur définie comme suit:

$$V = \Gamma(U) \Leftrightarrow V(x) = \min_{y \in \partial B(x)} U(y) + d(y, x). \quad (2-38)$$

La fonction distance  $\mathcal{U}_s(x)$  pour  $x \in \Omega$  peut-être approcher numériquement par un vecteur  $u \in \mathbb{R}^n$  tel que chaque composante  $u_i$  représente une approximation de la valeur de  $\mathcal{U}_s(x_i)$ . Alors l'équation Eikonale discrète est donnée par :

$$\begin{cases} \forall x_i \in \Omega, u_i = \min_{y \in \partial B(x_i)} u(y) + \|y - x_i\|_{T_i^{-1}}, \\ \forall x_i \in S, u_i = 0. \end{cases} \quad (2-39)$$

avec  $T_i$  le tenseur métrique associé au point  $x_i$ .

Pour calculer numériquement la solution de l'équation Eikonale qui est non linéaire on peut utiliser la méthode basé sur les itérations de Jacobi, on donnant un itéré initial  $u^{(0)}$  on peut calculer la valeur de la solution  $u$  comme suit :

$$u^{k+1} = \Gamma(u^k). \quad (2-40)$$

On résume alors cette méthode dans l'algorithme suivant :

---

Entrées :

- Le domaine de l'objet  $\Omega$ .
- La paramétrisation  $\varphi$ .
- La distance initiale  $\mathcal{U}_0$ .
- La courbe  $\gamma$
- L'ensemble des points de départ  $\mathcal{M}$

Début

Etape 1 :

Calcul de la longueur de la courbe  $\gamma(t)$  par le tenseur métrique  $T_x$  utilisant la relation suivante :

$$L(\gamma(t)) = \int_0^1 \|\gamma'(t)\|_{T_{\gamma(t)}} dt.$$

On cherche le chemin  $\gamma^*$  (la distance géodésique) qui minimise la longueur  $L$  :

$$\gamma^* = \operatorname{argmin}_{\gamma \in \mathcal{P}(x,y)} L(\gamma).$$

Etape 2 :

Il faut déterminer un ensemble de points de départ  $\mathcal{S}$  :

$$\mathcal{S} = (x_k)_k \subset \mathcal{M}$$

On calcule ensuite la fonction distance  $\mathcal{U}_{\mathcal{S}}$

$$\mathcal{U}_{\mathcal{S}}(x) = d(x, \mathcal{S}) = \min_{a \in \mathcal{S}} d(x, a)$$

Pour les autres points qui n'appartiennent pas à  $\mathcal{S}$  (c.-à-d.  $x \in \Omega \setminus \mathcal{S}$ ) :

$$\begin{cases} \forall x \in \Omega, \mathcal{U}(x) = \min_{y \in \partial B(x)} \mathcal{U}(y) + d(y, x), \\ \forall x \in \mathcal{S}, \mathcal{U}(x) = 0 \end{cases}$$

Où  $\partial B(x)$  est la frontière du disque  $B$  centré en  $x$ .

Etape 3 :

On résout alors le problème final utilisant une itération de Jacobi :

$$\mathcal{U}^{k+1} = \Gamma(\mathcal{U}^k).$$

Où

$\Gamma$  est l'opérateur définie comme suit :

$$V = \Gamma(U) \Leftrightarrow V(x) = \min_{y \in \partial B(x)} U(y) + d(y, x)$$

Fin

Sortie :

- Les distances  $u^k$ .

---

### Algorithme de l'équation Eikonal

#### Les Avantages de l'Algorithme de l'équation Eikonal

- Utilisation du chemin le plus court pour déterminer les régions de l'objet.
- Le calcul de la distance minimale entre deux régions voisines.
- Approche itérative de la distance calculée par la méthode de Jacobi.
- Le problème de l'eikonal est bien posé (solution existe et elle est unique).
- Le problème est stable et l'algorithme converge vers la solution désirée.

#### Les inconvénients de l'Algorithme de l'équation Eikonal

Le seul inconvénient de cette méthode est qu'elle nécessite un grand temps de calcul dû à sa complexité de l'ordre  $O(N \log(N))$  où  $N$  désigne le nombre de sommet. Il faut donc chercher un autre algorithme plus rapide et assez consistant pour améliorer le temps de calcul. Une solution probable est l'utilisation d'une équation linéaire pour calculer la distance tout en préservant les qualités assurées par l'équation Eikonal. D'où l'intérêt porté à la résolution du noyau de la chaleur sur une variété riemannienne.

## 2.4 Equation de la chaleur

De tous temps, les problèmes de transmission d'énergie, et en particulier de la chaleur ont eu une importance déterminante comme pour l'étude du fonctionnement d'appareils, transformation chimiques, etc. En informatique cette approche est utilisée dans le but de calculer la distance géodésique entre deux points  $x$  et  $y$  d'un objet 3D, ainsi que pour le lissage des objets.

La diffusion de la chaleur dans les milieux, est décrite par une fonction  $K$  appelée le noyau de la chaleur, ce noyau représente l'évolution de la température qui égale à une unité en un

point temps  $t = 0$ . Ce dernier est lié avec la distance géodésique [78] sur une variété Riemannienne via la célèbre formule de S. Varadhan [79]:

$$\frac{\Phi^2(x,y)}{4} = \lim_{t \rightarrow 0} t \log K(t, x, y) \quad (2-41)$$

Avec  $\Phi$  représente la distance géodésique entre le point  $x$  et  $y$  de la variété Riemannienne et  $K(t, x, y)$  représente le noyau de la chaleur. On peut écrire également :

$$\Phi(x, y) = \lim_{t \rightarrow 0} \sqrt{4t \log K_{t,x}(y)} \quad (2-42)$$

avec  $K(t, x, y) = \frac{1}{(4\pi t)^{d/2}} e^{-|x-y|^2}$  et  $x, y \in \mathbb{R}^d$

L'idée derrière cette expression vient du fait que la diffusion de la chaleur, peut être modélisé comme une grande collection des particules chaudes partant d'une façon aléatoire d'un point  $x$  vers un point  $y$ .

L'idée est de calculer la distance géodésique en résolvant l'équation de la chaleur sur une variété Riemannienne.

Le calcul de cette distance peut être fait en résolvant une autre équation dite équation d'Ekionale [80, 81]:

$$|\nabla\Phi| = 1 \quad (2-43)$$

$$\Phi|_{\gamma} = 1, \quad \text{condition au bord } \gamma.$$

Cette équation présente plusieurs difficultés numérique et théorique, car elle s'agit d'une équation hyperbolique non linéaire. Dans la pratique, pour surmonter ces difficultés, la plus part des auteurs cherche une méthode de relaxation itérative comme Gauss- Seidel [82] pour résoudre cette équation.

### 2.4.1 Noyau de la chaleur

Soit  $(M, H)$  une variété riemannienne, et soit  $\Delta$  l'opérateur Laplacien sur  $(M, \gamma)$  opérant les fonctions sur  $M$ . L'opérateur de la chaleur  $L = \frac{\partial}{\partial t} - \Delta$  sur  $(M, H)$ , est défini sur l'espace des fonctions  $M \times \mathbb{R}_+^*$ .

Une fonction  $K$  sur  $M \times M \times \mathbb{R}_+^*$  est appelée solution fondamentale de l'équation de la chaleur ou noyau de la chaleur, si elle vérifie les propriétés suivantes :

- ✓  $K$  est continue, de classe  $C^{k+1}$  pour la deuxième variable et de classe  $C^k$  pour la troisième.
- ✓  $\left(\frac{\partial}{\partial t} - \Delta\right)K = 0$ , où  $\Delta$  est le Laplacien par rapport à la deuxième variable.
- ✓  $\lim_{t \rightarrow 0^+} K(x, \cdot, t) = \delta_x$  et  $\forall x \in M$ , où  $\delta_x$  est la mesure de Dirac au point  $x$ .

La méthode qu'on propose pour calculer la distance géodésique peut être décrite en termes d'opérations et d'opérateur sur les variétés riemanniennes régulières. Cette méthode consiste premièrement à discrétiser l'équation de la chaleur par rapport au temps afin d'aboutir une équation elliptique et ensuite par une discrétisation spatiale de l'équation résultante.

Soit  $\Delta$  l'opérateur Laplacien qui est un opérateur semi définie négative agissant sur les fonctions à valeurs réelles différentiables sur la variété riemannienne  $(M, H)$ , l'algorithme de la méthode se présente comme suit :

---

Entrées :

- La solution initial  $U_0 = U$

Début

Etape 1 :

Choisir un  $t$  dans l'intervalle  $[0,1]$  avec  $t \ll 1$ .

Etape 2 :

Calculer les quantités suivantes :

a) Gradient discret

$$\nabla U = \left(\frac{\partial U}{\partial x}; \frac{\partial U}{\partial y}\right)$$

b) Divergence discret

$$div U = \nabla \cdot U = \frac{\partial U}{\partial x} + \frac{\partial U}{\partial y}$$

c) Laplacien discret

$$\Delta U = div(\nabla U)$$

Etape 3 :

Résoudre l'équation  $U(Id + t_j L) = \delta$  pour chaque  $t_j$ , où

$$L = \frac{\partial}{\partial t} - \Delta$$

Etape 4 :

Déterminer la distance géodésique

$$\Phi(x, y) = \lim_{t \rightarrow 0} \sqrt{4t \log k_{t,x}(y)}$$

Fin

Sortie :

- La distance  $\phi$ .

---

### Algorithme de l'équation de la chaleur

La fonction  $\Phi$  est une approximation de la distance géodésique, tandis que la vraie valeur de  $\Phi$  est obtenue en faisant tendre  $t$  vers 0.

Remarquons que si " $\Phi$ " est une solution de l'équation de Poisson " $\Phi + K$ " est aussi une solution, pour cela on doit ajouter une condition aux limites sur  $\Phi$  au bord pour avoir l'unicité de la solution.

#### 2.4.2 Discrétisation par rapport au temps

On subdivise l'intervalle du temps en  $q$  sous intervalles avec le pas  $\delta t$ , on peut identifier la valeur de  $\frac{\partial u}{\partial t}$  à l'instant  $t_j$  par l'un des formules suivantes :

➤ Schéma progressive  $\frac{\partial u}{\partial t}(t_j) \approx \frac{u(t_j + \delta t) - u(t_j)}{\delta t}$  (2-44)

➤ Schéma rétrograde  $\frac{\partial u}{\partial t}(t_j) \approx \frac{u(t_j) - u(t_j - \delta t)}{\delta t}$  (2-45)

➤ Schéma centré  $\frac{\partial u}{\partial t}(t_j) \approx \frac{u(t_j + \delta t) - u(t_j - \delta t)}{2\delta t}$  (2-46)

Nous nous considérons l'équation de la chaleur décrit à l'étape 1 de l'algorithme, en utilisant un schéma d'Euler rétrograde on peut écrire l'équation à l'instant  $t$  sous la forme suivante :

$$\frac{u(t)-u(t-\delta t)}{\delta t} = \dot{u}(t - \delta t) \quad (2-47)$$

Dans la pratique, cette expression revient à résoudre l'équation linéaire

$$(\text{Id} - t\Delta)u = u_0 \quad (2-48)$$

Ou  $u_0$  est la valeur de  $u$  à l'instant initiale  $t = 0$ , et  $\text{Id}$  représente l'application identité.

### Discrétisation spatiale

Le principe de notre méthode pour résoudre l'équation de la chaleur, peut être appliqué à n'importe quelle variété, il suffit d'avoir l'expression du gradient discret ( $\nabla$ ), de la divergence et de Laplacien. On note que ces opérateurs sont très local et qui dépend en plus de la variété ainsi que le maillage considéré, et qui ne présente pas des grandes erreurs significatives.

On considère un maillage triangulaire et, soit  $u \in \mathbb{R}^{|\mathcal{V}|}$  une fonction linéaire définie par morceaux. Une discrétisation standard du Laplacien au sommet  $i$  est donnée par :

$$(Lu)_i = \frac{1}{2A_i} \sum_j (\cot\alpha_{ij} + \cot\beta_{ij})(u_j - u_i)_i \quad (2-49)$$

Avec  $A_i$  est le un tiers (1/3) de la zone de tous les triangles incidents sur le sommet  $i$ .

La somme est prise sur tous les sommets voisins  $j$ ,  $\alpha_{ij}$  et  $\beta_{ij}$  sont les angles opposées par rapport à l'arête  $ij$ . On peut écrire cette expression sous une forme matricielle

$$L = A^{-1}L_C \quad (2-50)$$

Avec  $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$  est la matrice diagonale représentant les sommets de la zone, et  $L_C \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$  et l'opérateur de Cotan représentant la somme restante.

Le flux de la chaleur peut être alors calculé en résolvant le système symétrique défini positive suivant :

$$(A - tL_C)u = \delta_\gamma \quad (2-51)$$

Avec  $\delta_\gamma$  est l'opérateur de Kronecker  $\delta_\gamma(x) = \begin{cases} 1, & \text{si } x \in \gamma \\ 0, & \text{sinon} \end{cases}$  ou  $\gamma$  est le bord du domaine.

L'expression du gradient discret sur un élément (triangle) du maillage est donnée par :

$$\nabla u = \frac{1}{2A_f} \sum_i u_i (N \times e_i) \quad (2-52)$$

où la constante  $A_f$  est l'aire de la face du triangle,  $N$  est le vecteur normal unitaire sortant,  $e_i$  est le vecteur du  $i^{\text{ème}}$  arête et  $u_i$  représente la valeur de  $u$  sur le sommet opposé. De la même manière on peut associer à ce gradient un opérateur de divergence au sommet  $i$  défini comme suit :

$$\nabla \cdot X = \frac{1}{2} \sum_j \cot \theta_1 (e_1 \cdot X_j) + \cot \theta_2 (e_2 \cdot X_j) \quad (2-53)$$

Sauf cette fois-ci, la somme est prise sur tous les triangles incidents  $j$  chacun avec le vecteur  $X_j$ . Les vecteurs  $e_1$  et  $e_2$  sont les vecteurs des deux arêtes du triangle incident portant le sommet  $i$ ,  $\theta_1$  et  $\theta_2$  sont les deux angles opposés au sommet  $i$ .

Si on considère le vecteur  $b \in \mathbb{R}^{|V|}$ , le vecteur divergence pour le champ normalisé  $X$ , la fonction distance finale est calculée en résolvant le problème de Poisson suivant :

$$L_C \Phi = b. \quad (2-54)$$

#### Les Avantages de l'Algorithme de l'équation de la chaleur

- Utilisation des caractéristiques de l'équation de la chaleur pour approcher la distance
- Le calcul itératif de la distance entre deux régions.
- Le problème est bien posé (solution existe, stable et elle est unique).
- Le problème est stable et l'algorithme converge vers la solution désirée.
- Le temps de calcul assez raisonnable comparé à celui de l'équation Eikonal.

#### Les inconvénients de l'Algorithme de l'équation de la chaleur

L'équation de la chaleur souffre d'un seul inconvénient qui est celui de l'effet régularisant. En effet, pour chercher la distance minimale entre deux régions, si l'une des régions n'est pas homogène (c-à-d des régions discontinues) il se peut qu'on perd des informations dû au lissage de la solution.

## 2.5 Conclusion

Ce chapitre a été consacré à l'ensemble des caractéristiques des objets 3D. Nous nous sommes alors intéressés, dans un premier temps, à la reconnaissance des objets 3D en se basant sur le calcul de la distance géodésique (calculée en résolvant l'équation de la chaleur ou bien à partir de l'équation Eikonale). En choisissant les bonnes méthodes de classification de données, cela a conduit à un taux de reconnaissance assez remarquable. En revanche, les résultats obtenus dépendent continument du choix des classifieurs, ce qui nécessite une étude approfondie de ce choix. Ceci fera l'objet du chapitre suivant.

# **CHAPITRE 3 :**

# **CLASSIFICATION**

### 3.1 Introduction

Les méthodes de classement sont des méthodes basées sur apprentissage supervisé : elles font intervenir une expertise préalable des individus. Leur objectif est de construire un modèle de classement en fonction des individus à classer. A partir d'un ensemble d'individus provenant de chaque classe, elles établissent les frontières de décision dans l'espace des caractéristiques permettant de séparer les individus qui appartiennent à des classes différentes.

La classification est l'étape de décision qui réalise véritablement la reconnaissance : choix de la classe dont la représentation ou le modèle est le plus proche.

Il existe de nombreuses méthodes de classification :

- Réseaux de Neurones
- Machines à Support de Vecteurs (ou SVM)
- Naïve Bayes (ou encore Simple Bayes)
- Arbres de Décisions
- Etc ...

### 3.2 Réseaux de Neurones

Il n'y a pas de définition universellement acceptée de Réseau de Neurones (RdN). On considère généralement qu'un RdN est constitué d'un grand ensemble d'unités (ou neurones), ayant chacune une petite mémoire locale. Ces unités sont reliées par des canaux de communication (les connexions), qui transportent des données numériques. Les unités peuvent uniquement agir sur leurs données locales et sur les entrées qu'elles reçoivent par leurs connexions.

La plupart des RdN ont une certaine capacité d'apprentissage. Cela signifie qu'ils apprennent à partir d'exemples, de même que les enfants apprennent à lire, on lui présente des exemples de lettres et de chiffres, écrits avec des écritures et des fontes différentes. A la fin de l'apprentissage, on attend de lui qu'il ait une capacité de généralisation à partir des exemples qui lui ont été présentés, sans qu'il ne soit

jamais nécessaire de lui fournir une description analytique et discursive de la forme et de la topologie des chiffres et des lettres [82].

### 3.2.1 Apprentissage

L'apprentissage est l'étape essentielle dans un RdN. Lors de l'apprentissage (modification des poids des connexions), on cherche à minimiser une erreur, sur la base d'apprentissage, entre la sortie fournie et la sortie désirée. On espère ainsi minimiser une erreur sur un ensemble de généralisation, c'est à dire un ensemble de données inconnues du réseau. On peut arrêter l'apprentissage :

- Quand l'erreur entre la sortie du réseau et la sortie désirée ne diminue plus ou diminue d'une façon non significative,
- Après un certain nombre d'itérations fixé à priori,
- Pour la classification lorsque le réseau classe correctement toutes les données.

Il y a de très nombreuses sortes de RdN actuellement. Nous sommes intéressés dans le cadre de ce travail par le perceptron et le réseau multicouche.

### 3.2.2 Perceptron

C'est un des premiers réseaux de neurones, conçu en 1958 par Rosenblatt [83]. Il est linéaire et monocouche. Il est inspiré du système visuel. La première couche (d'entrée) représente la rétine. Les neurones de la couche suivante sont les cellules d'association, et la couche finale les cellules de décision (figure 3.1). Les sorties des neurones ne peuvent prendre que deux états (-1 et 1 ou 0 et 1).

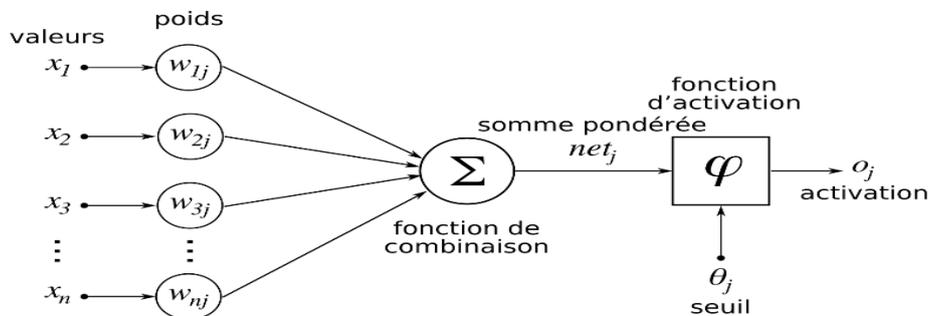


Figure 3.1 : Perceptron avec seuil [83]

Si la sortie du réseau est égale à la sortie désirée, le poids de la connexion n'est pas modifié. Dans le cas contraire le poids est modifié [84].

### Apprentissage du Perceptron

L'apprentissage consiste à déterminer les poids  $w_i$  pour que les exemples soient bien classés. L'algorithme commence avec des poids pris au hasard et on soumet les exemples en entrée du perceptron. On modifie à chaque fois les poids pour que le perceptron classe bien cet exemple :

- ∞ Initialisation des poids et du seuil  $\theta$  à des valeurs (petites) choisies au hasard.
- ∞ Présentation d'une entrée,  $e = (x_1, x_2, \dots, x_{n-1}, x_n)$  de la base d'apprentissage.
- ∞ Calcul de la sortie obtenue  $S$  pour cette entrée :

$$S = \sum(w_i \cdot x_i) - \theta \quad (3-1)$$

où  $\theta = \text{signe}(S)$  (si  $S > 0$  alors  $\theta = +1$  sinon  $S < 0$  alors  $\theta = -1$ ),  $\theta$  : seuil.

- ✓ Si la sortie  $\theta$  du Perceptron est différente de la sortie désirée  $d$  pour cet exemple d'entrée  $E$  alors modification des poids :

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \cdot (d_j - \theta_j) e_i \quad (3-2)$$

où :

- $d_j = +1$  si  $e$  est de la classe 1,
- $d_j = -1$  si  $e$  est de la classe 2,
- $d_j - \theta_j$  est une estimation de l'erreur.
- $w_{ij}(t + 1)$ : poids de la connexion entre les neurones  $i$  et  $j$  à l'instant  $t$ .
- $\eta$ : pas (taux, vitesse) d'apprentissage, comprise entre 0 et 1, on fait le souvent varier au cours de l'apprentissage. Partant d'une valeur élevée (typiquement environ 0,8), on le réduit progressivement quand on approche de la solution.
- ✓ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

Si l'ensemble d'apprentissage est linéairement séparable, l'algorithme du perceptron trouve l'hyperplan séparateur en un nombre fini d'itérations.

### 3.2.3 Perceptrons multicouches

Les perceptrons multicouches (PMC) sont apparus en 1985 [85]. Ce type de réseau est dans la famille générale des réseaux à «propagation vers l'avant », c'est-à-dire qu'en mode normal d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune rétroaction. Son apprentissage est de type supervisé, par correction des erreurs. Dans ce cas uniquement, le signal d'erreur est «rétropropagé » vers les entrées pour mettre à jour les poids des neurones.

Le PMC est un des RdN les plus utilisés pour des problèmes d'approximation, de classification et de prédiction [86].

#### Algorithme d'apprentissage de (rétropropagation)

La *rétropropagation* est une méthode de calcul des poids pour un réseau à apprentissage supervisé qui consiste à minimiser l'erreur quadratique de sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle et la sortie désirée) [82].

L'algorithme de la rétropropagation de gradient, bien que très simple à implanter, nécessite un certain savoir-faire pour une utilisation efficace. En effet, la convergence de l'algorithme n'est pas prouvée et de multiples variables sont à ajuster précisément en fonction du problème traité. Parmi ces variables à fixer, les paramètres apparaissant dans les différentes équations, la sélection des exemples pour l'apprentissage et le test, la structure du réseau (nombre de couches, taille de la couche cachée), la configuration initiale des poids, le nombre d'itérations d'apprentissage [86].

L'algorithme de rétro-propagation est ainsi défini:

$x = (x_1, x_2, \dots, x_{n-1}, x_n)$ , vecteur d'entrée

$o = (o_1, o_2, \dots, o_{n-1}, o_n)$ , vecteur de sortie désiré

$y = (y_1, y_2, \dots, y_{n-1}, y_n)$ , vecteur de sortie obtenu (réel)

- ✓ Appliquer un vecteur d'entrée aux nœuds d'entrées puis initialiser les poids du réseau ;
- ✓ Exécuter l'échantillon d'apprentissage à travers le réseau ;
- ✓ Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées en utilisant respectivement les équations suivante :

$$es = (o - x)f'(y) \quad (3-3)$$

$$ec = f'(y) \sum_{k=1}^m es.w \quad (3-4)$$

où

- $es$ : erreur de la couche de sortie ;
  - $ec$ : erreur de la couche cachée ;
  - $x$  : vecteur d'entrée de la couche de sortie (signal de la couche cachée) ;
  - $f$  : fonction sigmoïde, tel que :  $f'(y) = \frac{1}{1+e^{-x}}$
- ✓ Mise à jour les poids de la couche de sortie et couches cachées en utilisant respectivement les équations suivantes :

$$w(t + 1) = w(t) + \eta . es . x \quad (3-5)$$

$$w(t + 1) = w(t) + \eta . ec . x \quad (3-6)$$

- ✓ Répéter ce processus jusqu'à ce que l'erreur E devienne acceptable

$$\begin{aligned} E &= \frac{1}{2} \sum_{k=1}^m (o_k - x_k)^2 \\ &= \frac{1}{2} \sum_{k=1}^m (o_k - f(y_k))^2 \\ &= \frac{1}{2} \sum_{k=1}^m \left( o_k - f\left(\sum_{j=0}^n w_j x_j\right) \right)^2 \end{aligned} \quad (3-7)$$

### 3.3 Machines à Vecteurs de Support (SVM)

Depuis les années quatre-vingt-dix, les SVMs n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise. La performance de ces classifieurs dépasse celle des méthodes supervisées traditionnelles dans plusieurs applications comme Bi [87], Ritendra [88] et Chen [89]. Dans cette section, le principe de fonctionnement général des SVMs est présenté.

La linéarité et la non-linéarité ainsi que l'extension aux problèmes multi-classes de la méthode SVM sont également exposées.

### 3.3.1 Principe général

Les SVM peuvent être utilisés pour résoudre des problèmes de discrimination, c'est-à-dire décider à quelle classe appartient un échantillon, ou de régression, c'est-à-dire prédire la valeur numérique d'une variable. La résolution de ces deux problèmes passe par la construction d'une fonction  $h$  qui à un vecteur d'entrée  $x$  fait correspondre une sortie  $y$  :

$$y = h(x) \quad (3-9)$$

On se limite pour l'instant à un problème de discrimination à deux classes (discrimination binaire), c'est-à-dire  $y \in \{-1, 1\}$ , le vecteur d'entrée  $x$  étant dans un espace  $X$  muni d'un produit scalaire. On peut prendre par exemple  $X = \mathbb{R}^N$ .

#### a) Discrimination linéaire et hyperplan séparateur

Pour rappeler le cas simple est le cas d'une fonction discriminante linéaire, obtenue par combinaison linéaire du vecteur d'entrée  $= (x_1, x_2, \dots, x_n)^T$ , avec un vecteur de poids  $w = (w_1, w_2, \dots, w_n)$  :

$$h(x) = w^T x + w_0 \quad (3-10)$$

Il est alors décidé que  $x$  est de classe 1 si  $h(x) \geq 0$  et de classe -1 sinon. C'est un classifieur linéaire.

La frontière de décision  $h(x) = 0$  est un hyperplan, appelé hyperplan séparateur, ou séparatrice. Rappelons que le but d'un algorithme d'apprentissage supervisé est d'apprendre la fonction  $h(x)$  par le biais d'un ensemble d'apprentissage :

$$\{(x_1, l_1), \dots, (x_p, l_p)\} \in \mathbb{R}^N \times \{-1, 1\} \quad (3-11)$$

où les  $l_k$  sont les labels,  $p$  est la taille de l'ensemble d'apprentissage,  $N$  la dimension des vecteurs d'entrée. Si le problème est linéairement séparable, on doit alors avoir :

$$l_k h(x_k) \geq 0 \quad 1 \leq k \leq p \quad \text{autrement dit} \quad l_k (w^T x_k + w_0) \geq 0 \quad 1 \leq k \leq p. \quad (3-12)$$

## b) Marge maximale

On se place désormais dans le cas où le problème est linéairement séparable. Même dans ce cas simple, le choix de l'hyperplan séparateur n'est pas évident. Il existe en effet une infinité d'hyperplans séparateurs, dont les performances en apprentissage sont identiques (le risque empirique est le même), mais dont les performances en généralisation peuvent être très différentes. Pour résoudre ce problème, il a été montré [90], qu'il existe un unique hyperplan optimal, défini comme l'hyperplan qui maximise la marge entre les échantillons et l'hyperplan séparateur.

Il existe des raisons théoriques à ce choix. Vapnik a montré [90] que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente.

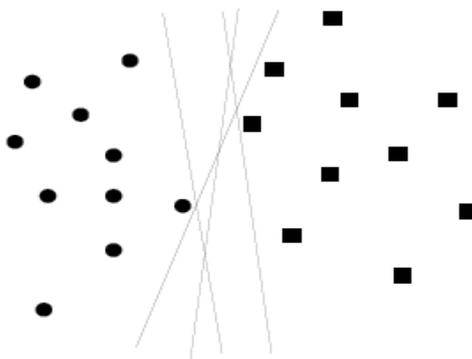


Figure 3.2 : Quelques hyperplans linéaires séparateurs valides [90].

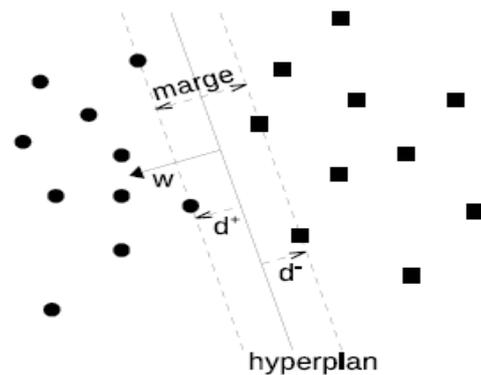


Figure 3.3 : Hyperplan linéaire séparateur ayant la marge maximale [90].

La marge est la distance entre l'hyperplan et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. L'hyperplan qui maximise la marge est donné par :

$$\arg \max_{w_1 w_0} \min_k \{ \|x - x_k\| : x \in \mathbb{R}^N (w^T x + w_0) = 0 \} \quad (3-13)$$

Il s'agit donc de trouver  $w$  et  $w_0$  remplissant ces conditions, afin de déterminer l'équation de l'hyperplan séparateur :

$$h(x) = w^T x + w_0 = 0 \quad (3-14)$$

### c) Cas non séparable : Kernel trick

La notion de marge maximale et la procédure de recherche de l'hyperplan séparateur telles que présentées pour l'instant ne permettent de résoudre que des problèmes de discrimination linéairement séparables [91]. C'est une limitation sèvre qui condamne à ne pouvoir résoudre que des problèmes jouets, ou très particuliers. Afin de remédier au problème de l'absence de séparateur linéaire, l'idée des SVM est de reconsidérer le problème dans un espace de dimension supérieure, éventuellement de dimension infinie.

Dans ce nouvel espace, il est alors probable qu'il existe un séparateur linéaire. Plus formellement, on applique aux vecteurs d'entrée  $x$  une transformation non-linéaire  $\phi$ . L'espace d'arrivée  $\phi(x)$  est appelé espace de redescription. Dans cet espace, on cherche alors l'hyperplan

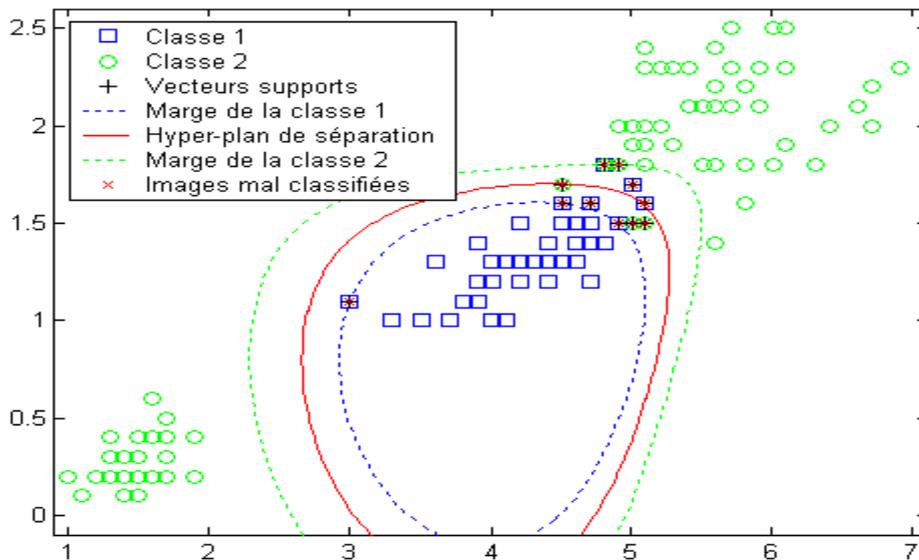


Figure 3.4 : Exemple de classification SVM non-linéairement séparable [91].

$$h(x) = w^T \phi(x) + w_0 \quad (3-19)$$

qui vérifie  $l_k h(x_k) > 0$  pour tous les points  $x_k$  de l'ensemble d'apprentissage, c'est-à-dire l'hyperplan séparateur dans l'espace de redescription.

En utilisant la même procédure que dans le cas sans transformation, on aboutit au problème d'optimisation suivant :

$$\text{Maximiser } L^*(\alpha) = \sum_{k=1}^p \alpha_k \sum_{i,j} \alpha_i \alpha_j l_i l_j \phi(x_i)^T \phi(x_j) \quad (3-20)$$

Sous les contraintes  $\alpha_i \geq 0$  et  $\sum_{k=1}^p \alpha_k l_k = 0$ .

Le problème de cette formulation est qu'elle implique un produit scalaire entre vecteurs dans l'espace de redescription, de dimension élevée, ce qui est coûteux en termes de calcul [91]. Pour résoudre ce problème, on utilise une astuce connue sous le nom de Kernel trick, qui consiste à utiliser une fonction noyau, qui vérifie :

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (3-21)$$

d'où l'expression de l'hyperplan séparateur en fonction de la fonction noyau :

L'intérêt de la fonction noyau est double :

- Le calcul se fait dans l'espace d'origine, ceci est beaucoup moins coûteux qu'un produit scalaire en grande dimension.
- La transformation  $\phi$  n'a pas besoin d'être connue explicitement, seule la fonction noyau intervient dans les calculs. On peut donc envisager des transformations complexes, et même des espaces de redescription de dimension infinie.

#### **d) Choix de la fonction noyau**

En pratique, on ne connaît pas la transformation  $\phi$ , on construit plutôt directement une fonction noyau. Celle-ci doit respecter certaines conditions, elle doit correspondre à un produit scalaire dans un espace de grande dimension. Le théorème de Mercer [91] explicite les conditions que  $K$  doit satisfaire pour être une fonction noyau : elle doit être symétrique, semi-définie positive.

L'exemple le plus simple de fonction noyau est le noyau linéaire :

$$K(x_i, x_j) = x_i^T \cdot x_j \quad (3-22)$$

On se ramène donc au cas d'un classifieur linéaire, sans changement d'espace. L'approche par Kernel trick généralisé ainsi l'approche linéaire. Le noyau linéaire est parfois employé pour évaluer la difficulté d'un problème.

Des noyaux usuels employés avec les SVM sont :

- Intersection d'histogrammes  $K(x, y) = \sum_i \min(|x_i|, |y_i|)$  (3-23)

- Noyau polynomial  $K(x_i, x_j) = (x_i^T \cdot x_j + 1)^d$  (3-24)

- Noyau gaussien  $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$  (3-25)

- Noyau Laplacien  $K(x, y) = \exp\left(-\frac{\|x-y\|}{\sigma}\right)$  (3-26)

- Radial Basis Function (RBF)  $K(x, y) = \exp\left(-\gamma \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}\right)$  (3-27)

- Noyau Sigmoidal  $K(x_i, x_j) = \tanh(x_i^T \cdot x_j + \theta)$  (3-28)

### 3.3.2 SVM multi-classes

Comme nous l'avons déjà mentionné, les machines à vecteurs supports sont réalisables uniquement pour de problème de classification en deux classes.

Pour pouvoir utiliser cette méthode pour les problèmes multi-calasses, on utilise une combinaison de plusieurs SVM binaires. Dans ce sens deux stratégies sont envisageables :

∞ **Un contre tous** : consiste à entraîner k SVM, une classe contre les autres, lors de la reconnaissance, un vecteur inconnu est affecté à la classe associée au SVM qui répond le mieux.

∞ **Un contre un** : entraîne un SVM pour chaque paire de classes. Lors de la reconnaissance, un vecteur inconnu est affecté à la classe identifiée par la majorité des SVM binaires.

### 3.4 k plus proches voisins (KNN)

La méthode des k plus proches voisins est une méthode de classification supervisée offrant des performances très intéressantes dans la classification [92]. Nous présentons le fondement théorique de cette méthode et illustrerons cette technique d'apprentissage statistique au travers du problème diagnostique d'une pathologie complexe.

#### 3.4.1 Algorithme 1er proche voisin (1-NN)

L'algorithme KNN [92] figure parmi les plus simples algorithmes d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation  $x$ , l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de  $x$  est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation  $x$ .

La méthode KNN est donc une méthode à base de voisinage, Ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction  $y = f(x_1, x_2, \dots, x_p)$  qui relie la variable dépendante aux variables indépendantes.

La méthode du plus proche voisin est une méthode où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, au regard des covariables utilisées. La détermination de leur similarité est basée sur des mesures de distance.

Formellement, soit  $L$  l'ensemble de données à disposition ou échantillon d'apprentissage :

$$L = \{(y_i, x_i), i = 1, \dots, n_L\} \quad (3-30)$$

où  $y_i \in \{1, \dots, c\}$  dénote la classe de l'individu  $i$  et le vecteur  $x_i' = (x_{i1}, \dots, x_{ip})$  représente les variables prédictives de l'individu  $i$ . La détermination du plus proche voisin est basée sur une fonction distance arbitraire( $\cdot, \cdot$ ).

La distance euclidienne ou dissimilarité entre deux individus caractérisés par p covariables est définie par :

$$d((x_1, x_2, \dots, x_p), (u_1, u_2, \dots, u_p)) = \sqrt{(x_1 - u_1)^2 + \dots + (x_p - u_p)^2} \quad (3-31)$$

Ainsi, pour une nouvelle observation  $(y, x)$  le plus proche voisin  $(y_{(1)}, x_{(1)})$  dans l'échantillon d'apprentissage est déterminé par :

$$d(x, x_{(1)}) = \min_i(d(x, x_i)) \quad (3-32)$$

et  $y = y_{(1)}$ , la classe du plus proche voisin, est sélectionnée pour la prédiction de  $y$ .

Les notations  $x_{(j)}$  et  $y_{(j)}$  représentent respectivement le jème plus proche voisin de  $x$  et sa classe d'appartenance. Parmi les fonctions distance types, la distance euclidienne est définie comme suit :

$$d(x_i, x_j) = (\sum_{s=1}^p (x_{is} - x_{js})^2)^{\frac{1}{2}} \quad (3-33)$$

et plus généralement la distance de Minkowski :

$$d(x_i, x_j) = (\sum_{s=1}^p |x_{is} - x_{js}|^q)^{\frac{1}{q}} \quad (3-34)$$

La méthode est justifiée par l'occurrence aléatoire de l'échantillon d'apprentissage. La classe  $Y_{(1)}$  du voisin le plus proche  $x_{(1)}$  d'un nouveau cas  $x$  est une variable aléatoire. Ainsi la probabilité de classification de  $x$  dans la classe  $y_{(1)}$  est  $P[Y_{(1)}/x_{(1)}]$ . Pour de grands échantillons d'apprentissage, les individus  $x$  et  $x_{(1)}$  coïncident de très près, si bien que  $P[y_{(1)}/x_{(1)}] \approx P[y/x]$ . Ainsi, la nouvelle observation (individu)  $x$  est prédite comme appartenant à la vraie classe  $y$  avec une probabilité égale approximativement à  $P[y/x]$ .

### 3.4.2 Algorithme K proche voisin (KNN)

Une première extension de cette idée, qui est largement et communément utilisée en pratique, est la méthode des k plus proches voisins. La plus proche observation n'est plus la seule observation utilisée pour la classification. Nous utilisons désormais les k plus proches observations. Ainsi la décision est en faveur de la classe majoritairement représentée par les k voisins. Soit  $k_r$ , le nombre d'observations issues du groupe des plus proches voisins appartenant à la classe  $r$

$$\sum_{r=1}^c k_r = k \quad (3-35)$$

Ainsi une nouvelle observation est prédite dans la classe  $l$  avec :

$$l = \max_r(k_r) \quad (3-36)$$

Ceci évite que la classe prédite ne soit déterminée seulement à partir d'une seule observation. Le degré de localité de cette technique est déterminé par le paramètre  $k$  : pour  $k=1$ , on utilise la méthode du seul plus proche voisin comme technique locale maximale, pour  $k \rightarrow n_L$  on utilise la classe majoritaire sur l'ensemble intégral des observations (ceci impliquant une prédiction constante pour chaque nouvelle observation à classifier).

### 3.4.3 Choix de $k$ :

Le paramètre  $k$  doit être déterminé par l'utilisateur :  $k \in \mathbb{N}$ . En classification binaire, il est utile de choisir  $k$  impair pour éviter les votes égalitaires. Le meilleur choix de  $k$  dépend du jeu de donnée. En général, les grandes valeurs de  $k$  réduisent l'effet du bruit sur la classification et donc le risque de sur-apprentissage, mais rendent les frontières entre classes moins distinctes. Il convient donc de faire un choix de compromis entre la variabilité associée à une faible valeur de  $k$  contre un 'oversmoothing' pour une forte valeur de  $k$ . Un bon  $k$  peut être sélectionné par diverses techniques heuristiques, par exemple, de validation-croisée. Nous choisirons la valeur de  $k$  qui minimise l'erreur de classification.

## 3.5 K-means

La méthode des  $k$ -moyennes (K-means) [93] est une extension des centres mobiles. C'est une méthode de classification automatique (clustering) qui regroupe les individus dans un nombre de classes prédéfini, par rapport à leurs distances avec les barycentres des classes.

K-means est un algorithme de quantification vectorielle (clustering en anglais). K-means est un algorithme de minimisation alternée qui, étant donné un entier  $K$ , va chercher à séparer un ensemble de points en  $K$  clusters (Figure 3.1).

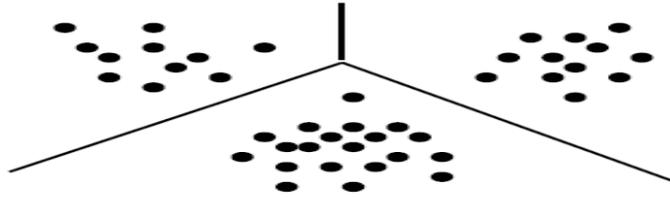


Figure 3.5 : Clustering sur un ensemble de points 2D, 3 clusters.

On utilise les notations suivantes :

- Les  $x_i \in \mathbb{R}^p, i \in \{1, \dots, n\}$  sont les points à séparer.
- Les  $z_i^k$  sont des variables indicatrices associées aux  $x_i$  telles que  $z_i^k = 1$  si  $x_i$  appartient au cluster  $k, z_i^k = 0$  sinon  $z$  est la matrice des  $z_i^k$ .
- $\mu$  est le vecteur des  $\mu_k \in \mathbb{R}^p$ , ou  $\mu_k$  est le centre du cluster  $k$ .

On définit de plus la mesure de distorsion  $J(\mu, z)$  par :

$$J(\mu, z) = \sum_{i=1}^n \sum_{k=1}^n z_i^k \|x_i - \mu_k\|^2 \quad (3-37)$$

### Algorithme

Le but de l'algorithme est de minimiser  $J(\mu, z)$ , il se présente sous la forme d'un algorithme de minimisation alternée :

- Etape 0 : ' choisir le vecteur  $\mu$  '
- Etape 1 : on minimise  $J$  par rapport à  $z : z_i^k = 1$  pour  $k \in \arg \min \|x_i - \mu_k\|$ , i.e. on associe à  $x_i$  le centre  $\mu_k$  le plus proche.
- Etape 2 : on minimise  $J$  par rapport à  $\mu : \mu_k = \frac{\sum_i z_i^k x_i}{\sum_i z_i^k}$ .
- Etape 3 : retour à l'étape 1 jusqu'à convergence.

### 3.6 Arbres de décision

Un Arbre de Décision est un outil d'aide à la décision et à l'exploration de données qui, comme son nom l'indique, prend la forme d'un arbre, qui modélise une hiérarchie de tests sur les valeurs d'un ensemble de variables appelées attributs. À l'issue de ces tests, le prédicteur produit une valeur numérique ou choisit un élément dans un ensemble discret de conclusions. On parle de régression dans le premier cas et de classification dans le second. Leur structure arborescente les rend également

lisibles par un être humain, contrairement à d'autres approches où le prédicteur construit est une « boîte noire ».

Le descriptif que nous proposons ici décrit les bases de leur fonctionnement tout en apportant quelques justifications théoriques. Nous aborderons aussi (brièvement) quelles extensions d'arbre de décision tel que C4.5 [94], CART [95] et Random Forests [96].

### 3.6.1 Construction d'un arbre de décision

L'idée centrale est de Diviser récursivement et le plus efficacement possible les exemples de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtienne des sous-ensembles d'exemples ne contenant (presque) que des exemples appartenant tous à une même classe. Dans toutes les méthodes, on trouve les trois opérateurs suivants :

- Décider si un nœud est terminal, c'est-à-dire décider si un nœud doit être étiqueté comme une feuille. Par exemple : tous les exemples sont dans la même classe, il y a moins d'un certain nombre d'erreurs, ...

- Sélectionner un test à associer à un nœud. Par exemple : aléatoirement, utiliser des critères statistiques, ...

- Affecter une classe à une feuille. On attribue la classe majoritaire sauf dans le cas où l'on utilise des fonctions coût ou risque.

Le schéma général de l'algorithme est le suivant :

- 
- 1: Entrées : langage de description ; échantillon S
  - 2: Début
  - 3: Initialiser à l'arbre vide ; // la racine est le nœud courant
  - 4: Répéter
  - 5: Décider si le nœud courant est terminal ;
  - 6: Si le nœud est terminal
  - 7: Alors affecter une classe ;
  - 8: Sinon sélectionner un test et créer le sous arbre ;

- 9: Passer au nœud suivant non exploré s'il existe ;
  - 10: Jusqu'à obtenir un arbre de décision ;
  - 11: Fin
- 

### 3.6.2 Mesure d'homogénéité des feuilles

On souhaite alors définir une mesure de l'hétérogénéité d'une feuille. Soient  $p_1, \dots, p_n$  les fréquences relatives des classes 1, ..., n dans P. Voici les mesures fréquemment rencontrées dans la littérature :

- ✓ Entropie :

En général, si on nous donne une distribution de probabilité  $P = (p_1, p_2, \dots, p_n)$  et un échantillon S alors l'Information portée par cette distribution, aussi appelée l'Entropie de P, est :

$$Entropie(P) = - \sum_{i=1}^n p_i \times \log(p_i) \quad (3-37)$$

C'est la mesure d'erreur utilisée dans les arbres ID3 et C4.5 [3, 1].

- ✓ Gain d'information :

On définit le gain pour un test T et une position p

$$Gain(p, T) = Entropie(p) - \sum_{j=1}^n (p_j \times Entropie(p_j)) \quad (3-38)$$

Où n est l'arité du test T et  $p_j$  la proportion d'éléments de S à la position p qui vont en position  $p_j$  (qui satisfont la i ème branche du Test)

- ✓ Critère de Gini :

Soit  $S_1, \dots, S_K$  des partitions d'un échantillon S selon les classes de C.

$$Gini(S) = \sum_{i=1}^K \frac{|S_i|}{|S|} \left( 1 - \frac{|S_i|}{|S|} \right) = \sum_{i \neq j} \frac{|S_i| \times |S_j|}{|S|^2} \quad (3-39)$$

CART utilise l'indice de Gini pour déterminer dans quel attribut de la branche doit être généré. La stratégie est de choisir l'attribut dont indice de Gini est minimum après la séparation [94].

### 3.7 Algorithme C4.5

Cet algorithme a été proposé en 1993, toujours par Ross Quinlan [97], pour pallier les limites de l'algorithme ID3 [98]. Le C4.5 repose complètement sur l'algorithme ID3. Nous nous focaliserons donc davantage ici sur les limites de l'algorithme ID3 et les améliorations apportées par C4.5.

#### 3.7.1 Attributs de valeur inconnue

Durant la construction de l'arbre de décision il est possible de gérer les données pour lesquels certains attributs ont une valeur inconnue en évaluant le gain, ou le gain ratio pour un tel attribut en considérant seulement les enregistrements pour lesquels cet attribut est défini.

En utilisant un arbre de décision, il est possible de classer les enregistrements qui ont des valeurs inconnus en estimant la probabilité des différents résultats possibles.

Le nouveau critère de gain sera de la forme :

$$\text{Gain}(x) = F (\text{Infos}(T) - \text{Info}(X, T)) \quad (3-40)$$

$F$  = nombre d'échantillons dans la base de données avec la valeur connue pour un attribut donné / nombre total d'échantillons dans un ensemble de données.

#### 3.7.2 Attributs à valeur sur intervalle continu

Il est maintenant également de gérer le cas d'attributs à valeur dans des intervalles continus, de la façon suivante. Disons que l'attribut  $C_i$  a un intervalle continu de valeurs. On examine les valeurs de cet attribut dans les données d'apprentissage. Disons que ces valeurs sont en ordre croissant,  $A_1, A_2, \dots, A_m$ . Ensuite pour chacune de ces valeurs, on partitionne les enregistrements entre ceux qui ont des valeurs de  $C_i$  inférieures ou égales à  $A_j$  et celles qui ont des valeurs supérieures à  $A_j$ . Pour chacune de ces partitions on calcule le gain, ou le gain ratio et on choisit la partition qui maximise le gain.

### 3.7.3 Elagage de l'arbre de décision

L'élagage de l'arbre de décision s'effectue en remplaçant un sous arbre entier par une feuille. Cette substitution a lieu si une règle de décision établit que le taux d'erreur attendu dans le sous arbre est supérieur que celui d'une simple feuille.

L'élagage, dans les algorithmes ID.3, C4.5, CART, consiste à calculer « le pourcentage des pixels bien classifiés dans une feuille » afin d'éliminer les nœuds pour lesquels les pourcentages de pixels sont inférieurs ou égaux à un seuil donné ; certaines branches de l'arbre seront alors supprimées [97]; [94]; [95].

Le pourcentage des pixels bien classifiés  $i$  contenus dans une feuille est donné par :

$$P_i = \frac{N_i}{N} \quad (3-41)$$

Où  $P_i$  est le pourcentage des pixels bien classifiés dans une feuille ;

$N_i$  est le nombre de pixels bien classifiés ;

$N$  est le nombre total de pixels appartenant dans cette feuille.

C4.5 utilise l'ensemble d'apprentissage pour élaguer l'arbre obtenu. Le critère d'élagage est basé sur une heuristique permettant d'estimer l'erreur réelle sur un sous-arbre donné. Bien qu'il semble peu pertinent d'estimer l'erreur réelle sur l'ensemble d'apprentissage, il semble que la méthode donne des résultats corrects.

### 3.7.4 Algorithme de base

On suppose toujours que le langage de représentation est constitué d'un certain nombre d'attributs. Ces attributs peuvent être binaires, qualitatifs (à valeurs dans un ensemble fini de modalités) ou continus (à valeurs réelles). On utilise pour cela des critères statistiques qui permettent d'atteindre les deux objectifs suivants : un nombre de classes pas trop important et une bonne répartition entre les différentes classes.

Nous supposons prédéfini un ensemble de tests  $n$ -aires. Pour définir l'algorithme, nous allons définir les trois opérateurs utilisés par l'algorithme C4.5 pour calculer un bon arbre de décision (phase d'expansion on ne va pas la traiter parce qu'elle ressemble à celle d'ID3), puis nous verrons la phase d'élagage. On dispose en entrée

d'un ensemble d'apprentissage  $A$ . On utilise la fonction entropie définie précédemment.

Soit  $p$  une position, on choisit alors le test qui maximise le gain en utilisant la fonction entropie pour mesurer le degré du mélange. La fonction Gain, ainsi définie, privilégie les attributs ayant un grand nombre de valeurs. Elle est donc pondérée par une fonction qui pénalise les tests qui répartissent les éléments en un trop grand nombre de sous-classes. Cette mesure de la répartition est nommée 'SplitInfo' et est définie par :

$$SplitInfo(p, test) = - \sum_{j=1}^n P' \left( \frac{j}{p} \right) \times \log(P' \left( \frac{j}{p} \right)) \quad (3-42)$$

Dans laquelle  $n$  est l'arité de test, et  $P'(j/p)$  est la proportion des éléments présents à la position  $p$  prenant la jème valeur de test. Il faut remarquer que, contrairement à l'entropie, la définition précédente est indépendante de la répartition des exemples à l'intérieur des différentes classes.

La valeur de 'Splitinfo' ne dépend que de la répartition entre les différentes valeurs possibles pour le test. Cette fonction a des valeurs grandes lorsque le test a un grand nombre de valeurs possibles avec peu d'éléments pour chacune des valeurs. En effet, considérons le cas extrême d'un attribut  $n$ -aire avec un exemple par classe, la fonction vaut alors  $\log n$ . À l'inverse, considérons le cas d'un attribut binaire pour lequel les exemples sont répartis uniformément entre ces deux valeurs, la fonction vaut alors 1.

La nouvelle fonction de gain, appelée ratio de gain et notée 'GainRatio', est alors définie par :

$$GainRatio(p, T) = \frac{Gain(p, T)}{SplitInfo(p, T)} \quad (3-43)$$

En position  $p$  (non maximale), on choisit le test qui maximise le GainRatio (option par défaut de C4.5). On peut modifier les options pour utiliser le Gain : Affecter une classe à une feuille. On attribue la classe majoritaire. S'il n'y a aucun exemple on attribue la classe majoritaire du nœud père.

### 3.8 Forêts aléatoires

Au cours de ces dernières années, les chercheurs qui travaillent sur les problématiques de Reconnaissance de Formes, se sont de plus en plus intéressés aux méthodes de classification qui utilisent des combinaisons de classifieurs. Certaines de ces méthodes ont montré qu'elles étaient particulièrement performantes. L'une des plus efficaces, et des plus récentes par ailleurs, est la famille de méthodes appelée Forêts Aléatoires (Random Forests) [96].

Random Forest est une méthode généralement efficace et facile à comprendre, introduite par Leo Breiman 2001 [96] qui s'appuie sur la combinaison parallèle d'un nombre arbitraire de classifieurs élémentaires de type Arbres de Décision.

#### 3.8.1 Bagging

Le « bagging », acronyme pour « bootstrap aggregating », est un méta-algorithme qui combine les deux techniques sus-acronymées :

↪ Bootstrapping :

Un bootstrap d'un ensemble  $T$  est l'ensemble obtenu en tirant  $|T|$  fois des éléments de  $T$  uniformément au hasard et avec remise. Le bootstrapping d'un ensemble d'entraînement  $T$  produit un nouvel ensemble  $T'$  qui présente en moyenne  $1 - e^{-1} \approx 63\%$  instances uniques différentes de  $T$  quand  $|T| \gg 1$ .

↪ Aggrégation :

On produit plusieurs bootstraps  $T'_1, \dots, T'_m$ , chaque bootstrap  $T'_i$  étant utilisé pour entraîner un prédicteur  $t_i$  (penser ici à un arbre de régression, mais la technique s'applique à n'importe quelle famille de prédicteurs). Étant donnée une instance  $(x, y)$ , on fait régresser chaque arbre, ce qui nous donne un ensemble de valeurs  $y_1, \dots, y_m$  prédites. Celles-ci sont alors agrégées en calculant leur moyenne  $\bar{y} = 1/m \sum_i y_i$ .

#### 3.8.2 Extensions des forêts aléatoires

De nombreux modèles de forêts aléatoires ont été créés qui correspondent à autant de manière d'incorporer de l'aléatoire dans les arbres. À titre d'exemple, on peut citer:

- le Tree Bagging [99] introduit de l'aléatoire dans l'échantillon initial sélectionnant certains points plutôt que d'autres et laisse grandir l'arbre jusqu'à ce que chaque nœud comporte un unique élément
- le random subspace consiste à sélectionner à chaque nœud  $K$  variables de manière aléatoire et, parmi celles-ci, à choisir celle qui minimise un certain critère.
- la Random Forest [96] qui consiste à mélanger le CART, le bagging et le random subspace : pour chaque arbre, on tire un échantillon à partir de l'échantillon initial ; à chaque nœud, on choisit aléatoire  $K$  variables et on prend, parmi celles-ci, celle qui minimise le critère de l'algorithme CART. On laisse grandir l'arbre jusqu'à ce qu'il n'y ait plus qu'un seul élément dans chaque nœud.
- le Random Select Split [99] qui sélectionne les  $K$  meilleures séparation et qui en choisit une parmi celles-ci de manière aléatoire. La position de la coupure est également calculée de manière aléatoire.

### 3.8.3 Principe

L'algorithme de forêts aléatoires (Leo Breiman, 2001) [96] crée un ensemble d'arbres de décision afin de réduire l'erreur de biais et d'assurer une faible corrélation entre les arbres. Les arbres sont construits sans élagage pour assurer un faible biais. Dans ce qui suit on va voir le principe des forêts aléatoires que nous avons utilisées.

Un arbre de la forêt est construit à partir d'un échantillon bootstrap tiré avec remise depuis l'ensemble d'apprentissage et la recherche de la meilleure coupe est fondée sur un sous-ensemble d'attributs tiré aléatoirement. Ces deux dernières propositions ont pour objectif de maintenir à un bas niveau la corrélation entre les arbres afin d'assurer leur diversité.

Considérons une tâche de classification de  $m$  individus  $x_i$  ( $i = 1, m$ ) et  $n$  attributs. Un arbre de décision (noté DT) dans la forêt de  $k$  arbres (noté  $RF = \{DT_i\}_{i=1, \dots, k}$ ) est créé de la façon suivante :

- tirage avec remise depuis l'ensemble d'apprentissage d'un échantillon bootstrap (noté  $B_i(i = 1, k)$ ) qui est utilisé pour la construction de l'arbre ;
- recherche d'une meilleure coupe pour chaque nœud de décision à partir d'un sous ensemble aléatoire de  $n_0$  attributs ( $n_0 < n, n_0 = \sqrt{n}$ ) ;
- construction de l'arbre le plus profond possible (sans élagage).

De cette façon on peut obtenir un ensemble d'Arbres de Décision adaptés au problème, mais différents les uns des autres et pouvant prendre des décisions différentes. S'agissant alors de classer un individu quelconque de la population de données étudiée, tous les Arbres de Décision fournissent un « vote » pour une classe, que l'on combine ensuite, généralement à l'aide d'un vote à la majorité, dans le cas de la classification pour obtenir la décision finale ou la moyenne des prédictions des arbres dans le cas d'une régression..

## 6.9 Conclusion

Pour résoudre un problème de classification, on est confronté à des formes susceptibles d'appartenir à des catégories, ou classes différentes. Un classifieur est capable d'attribuer une classe à une forme inconnue qui lui est présentée. Comme conclusion, on a constaté que les performances d'un classifieur dépendent essentiellement de la représentation des formes utilisées, c'est-à-dire du prétraitement effectué sur les données brutes.

# **CHAPITRE 4 :**

## **Résultats**

### **Expérimentaux**

### **4.1 Introduction**

Dans ce chapitre, nous présentons les résultats d'indexation d'objets 3D obtenus par les approches 3Det 2D-3D. Grâce aux étapes d'extraction des caractéristiques et la classification, nous sommes en mesure de comparer la requête fournie avec chaque objet de la base de données. Ce chapitre est divisé en deux parties qui élaborent les résultats expérimentaux obtenus par les deux approches.

### **4.2 Résultats Approche 2D/3D**

Cette partie présente le système de navigation dans une base d'objets 3D que nous avons développé. Nous montrerons que la navigation dans une collection d'objets à l'aide d'une interface peut apporter à l'utilisateur un aide significative dans la visualisation de la collection dans son ensemble. En effet, en plus de la représentation d'objets, l'ajout d'une dimension supplémentaire permet d'afficher un plus grand nombre d'informations. Nous concluons en présentant les résultats obtenus.

#### **4.2.1 Base d'images utilisée**

Pour évaluer et valider les différentes approches étudiées et développées dans ce travail, une base d'images est utilisée, à savoir, la base « Columbia Object Image Library (COIL-100)» [100]. Cette base constitue des concepts bien définis où chaque image ne peut appartenir qu'à une seule classe.

La base d'images COLUMBIA contient des images tirées d'objets 3D pris en différentes positions. La collection COIL-100 est utilisée surtout dans le domaine de reconnaissance d'objets. Elle est constituée par des images en couleurs prises à partir de 100 objets différents. L'ensemble de ces objets est affiché dans la figure 4.1.



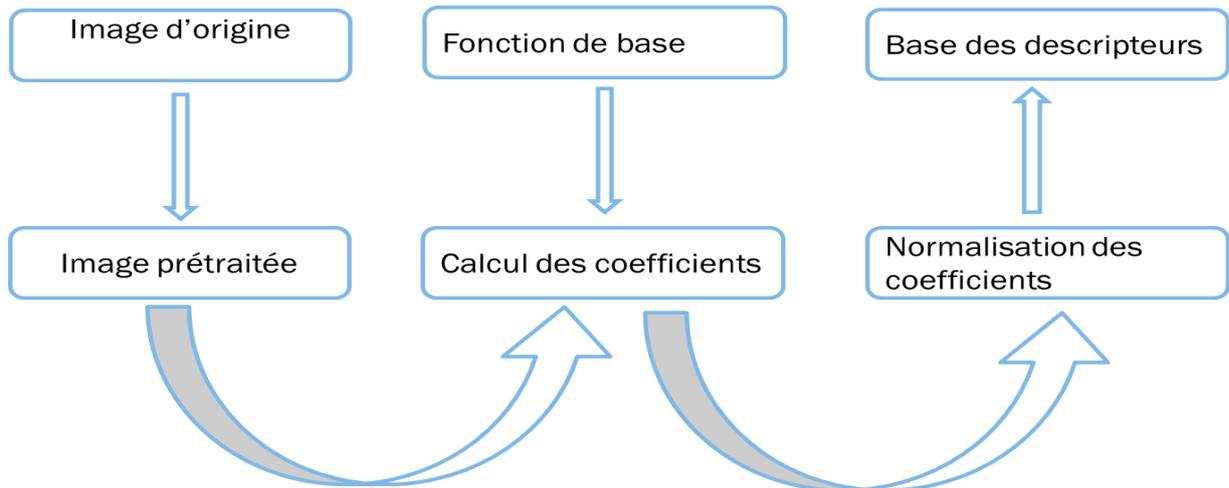


Figure 4.2 : Schéma de la procédure d'indexation.

#### ✚ Prétraitements

Cette étape se compose de deux parties : l'extraction de la région d'intérêt et l'interpolation. Tandis que la première partie élimine le problème du décalage dans l'image, la deuxième partie standardise la taille de l'image à transformer, dans le but de simplifier la mise en correspondance avec les fonctions de base.

Pour extraire la région d'intérêt, le centre de gravité de l'objet et la dimension de l'image qui contient l'objet sont calculés. La dimension vaut deux fois la distance du centre de gravité au point de l'objet le plus éloigné [65].

Une grande partie des méthodes d'indexation d'objets 3D que nous venons dénoncer sont robustes aux principales déformations usuelles par le fait que le prétraitement normalise les objets. Les déformations qui peuvent déformer un objet 3D sont de trois types :

- la translation, qui déplace un objet par rapport au centre du repère ;
- le changement d'échelle, qui modifie la taille de l'objet ;
- la rotation : qui le fait tourner.

Certaines méthodes sont naturellement invariantes à certaines de ces déformations (Le descripteur ne change pas). On parle de méthode invariante aux rotations, aux changements d'échelle et aux translations. A l'opposé, on qualifie des méthodes de robustes à ces déformations, si elles possèdent un prétraitement, qui normalise les objets avant de les indexer.

La normalisation porte sur trois points, qui sont :

- le centrage, qui place le centre de gravité des objets au centre du repère pour rendre les méthodes robustes aux translations,
- la mise à l'échelle, qui modifie la taille de l'objet pour rendre les méthodes robustes aux changements d'échelles,
- l'alignement, qui aligne les objets par rapport aux axes pour rendre les méthodes robustes aux rotations [65].

#### ✚ Calcul des coefficients

Dans cette étape, les moments de Zernike et le descripteur CENTRIST et (CENTRIST + M. Zernike), ACP (M. Zernike), et ACP(CENTRIST), sont appliquées sur les images prétraitées.

#### ✚ Normalisation des coefficients

Comme les coefficients de  $Z_{n,m}$  sont complexes, on se sert des modules des coefficients  $\|Z_{n,m}\|$  qui sont normalisés par  $\|Z_{0,0}\|$ . Ceci est appelé la normalisation de surface, puisque le premier coefficient  $Z_{0,0}$  est égal à la somme des pixels non nuls dans l'image. Nous disposons ainsi de  $n \times m - 1$  coefficient, puisque  $Z_{0,0}$  est utilisé pour la normalisation.

#### ✚ Base des descripteurs

Cette étape sert aux stockages des coefficients normalisés de chaque image dans la base de données des signatures.

Afin d'évaluer le pouvoir discriminant individuel de chacune de ces descripteurs, on les a appliqués sur la base données comme suivant :

- Moments de Zernike
- ACP (Zernike, 14) : on applique la méthode ACP sur la base après une extraction par M. Zernike [67].
- CENTRIST
- ACP (CENTRIST, 100) : on applique la méthode ACP sur la base après une extraction par CENTRIST [65].
- Exécution de l'analyse de composants principaux (ACP) :

- HYBRIDE : On combine les descripteurs CENTRIST et Zernike par une concaténation horizontale.

### 4.2.3 Méthodologie d'application

Nous appliquons les phases classiques d'apprentissage et de décision de la manière suivante [24]:

- ∞ L'image est redimensionnée au format 128x128 pixels.
- ∞ Le descripteur CENTRIST est calculé pour les images.
- ∞ Les moments de Zernike sont calculés à partir de l'image.
- ∞ On a utilisé le réseau de neurone comme méthode de classification pour l'évaluation de l'approche 2D. On a adapté la configuration suivante pour tous les tests fait :
  - Erreur sur performance = 0,0001
  - Erreur minimale du gardien = 0,00001
  - Nombre d'itération = 5000
  - Nombre de couches cachées = 1
  - Nombre de neurones de la couche cachée = 50
  - Fonction d'activation utilise pour toutes les couches est « sigmoïde »
  - Nombre de neurone dans la couche d'entrée et la couche de sortie dépend de nombre d'objets et d'images par objet.
- ∞ On a utilisé les SVM comme deuxième méthode de classification pour faire une comparaison avec les résultats que nous avons obtenus avec le réseau de neurone.

### 4.2.4 Résultats expérimentaux

La première étape c'est la phase d'apprentissage. Pour ce travail, on fait un apprentissage supervisé on se basant sur les réseaux de neurones. Les résultats d'apprentissage sont décrits dans le tableau 4.1.

### ✓ Taux de reconnaissance

Les résultats de la phase de reconnaissance sont résumés dans le tableau suivant :

	Zernike	ACP(Zer,14)	CENTRIST	ACP (CENT,100)	CENTRIST +Zernike	ACP
200 images	100,00%	98,00%	98,50%	99,50%	99,00%	99,50%
400 images	98,75%	93%	99,25%	98,75%	99%	90%
800 images	92,13%	24,25%	94,25%	97,88%	99,12%	80,40%
1200 images	92%	18,37%	99,50%	93,42%	99,25%	79,01%
1600 images	69,31%	3,13%	86,81%	84,50%	91,56%	68,12%
2000 images	17,05%	2,40%	13,05%	4,35%	78%	12,23%
2400 images	1,20%	0,63%	2,42%	3,67%	6,30%	0,56%
2800 images	0,06%	0,00%	2,54%	2,36%	3,18%	0,00%

Tableau 4.1: Taux de reconnaissance en fonction de l'ordre de la base

Lorsqu'on augmente l'ordre des images, c'est-à-dire le nombre de classes dans la base, les taux de reconnaissance diminuent pour toutes les méthodes présentes dans le tableau 4.1. Cela vient du fait que nous avons travaillé avec une seule couche cachée pour le réseau de neurones durant toutes les expériences faites. En revanche, on peut remarquer que le descripteur hybride (CENTRIST+ Zernike) marque toujours le meilleur taux comparant aux autres méthodes. Ceci nous montre la robustesse du modèle proposé. Par contre on remarque qu'en augmentant le nombre des images le descripteur Hybride perd son efficacité, ceci est dû à l'utilisation de la même configuration du réseau de neurones. Pour remédier à ce genre de problème il suffit d'ajouter d'autres couches cachées. Ce qui reste un point faible de la méthode de classification utilisée.

Lorsqu'on applique l'ACP sur les descripteurs, on constate :

- Une diminution de temps d'exécution et de taux de reconnaissance (ACP et Zernike).
- Une diminution de temps d'exécution, mais une stabilité de taux de reconnaissance (ACP et CENTRIST).

Pour visualiser l'évolution du taux de reconnaissance pour les méthodes présentées dans le tableau 4.1 en fonction de nombre d'images, on présente les courbes sur la figure 4.3.

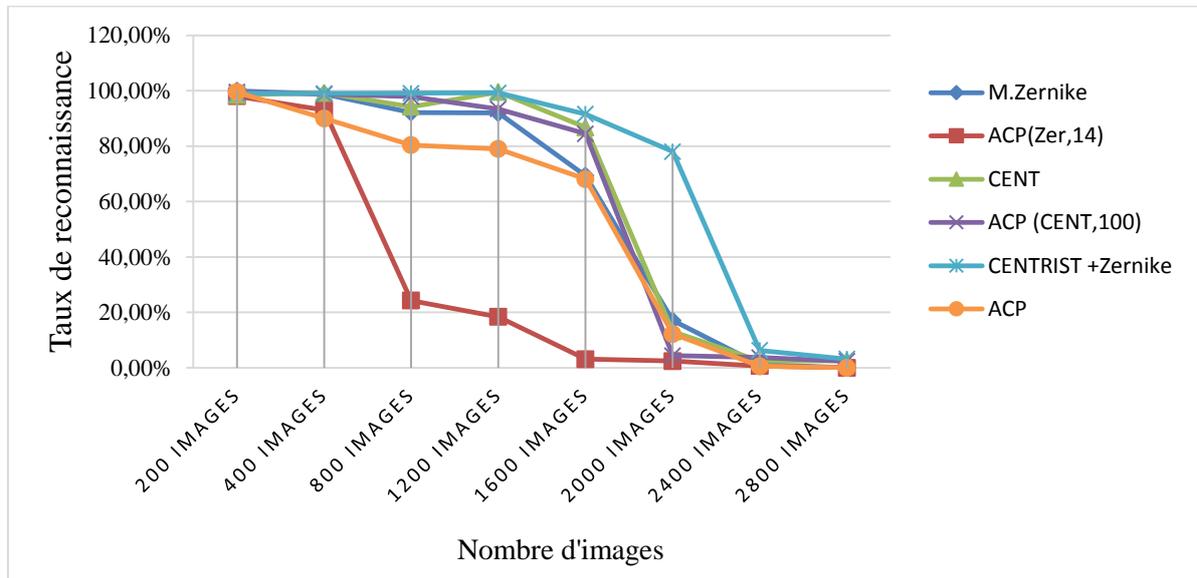


Figure 4.3 : Taux de reconnaissance en fonction du nombre d'image dans la base

#### ✓ L'erreur sur l'apprentissage

D'après le tableau 4.2, on présente l'erreur quadratique concernant l'apprentissage utilisé par les réseaux de neurones. On peut voir clairement que l'erreur associée au descripteur Hybride (CENTRIST+ Zernike) est le plus faible comparativement aux autres méthodes. En effet, on obtient un taux d'erreur faible même si on augmente le nombre d'images. En revanche, avec les moments de Zernike par exemple on obtient un taux d'erreur considérable lorsqu'on augmente le nombre des images, de même pour les autres descripteurs. Cela vient du fait que l'utilisation des deux descripteurs en même temps donne une capacité descriptive suffisante de l'image. Le graphe de la figure 4.4 montre le comportement de chaque descripteur en fonction de nombre d'images et de l'erreur.

	ZERNIKE	ACP(ZER)	CENTRIST	ACP(CENTRIST)	ACP	CENTRIST+Zernike
400 images	9,92E-06	3,26E-04	9,85E-05	5,69E-06	3,87E-03	9,68E-06
800 images	1,55E-05	4,02E-04	9,85E-05	9,91E-05	4,38E-04	9,87E-06
1200 images	5,59E-04	4,18E-04	6,18E-04	9,65E-04	9,95E-04	9,99E-06
1600 images	2,84E-04	4,17E-04	8,33E-04	6,81E-04	9,79E-04	9,61E-06
2000 images	4,98E-04	4,27E-04	8,72E-03	2,94E-04	9,28E-04	9,93E-06
2400 images	6,11E-03	4,58E-04	7,90E-04	1,57E-03	9,85E-04	9,78E-06
2800 images	1,70E-03	2,32E-04	1,16E-03	1,41E-03	9,80E-04	9,14E-06

Tableau 4.2: Erreur d'apprentissage en fonction de l'ordre et de la méthode utilisée pour l'extraction

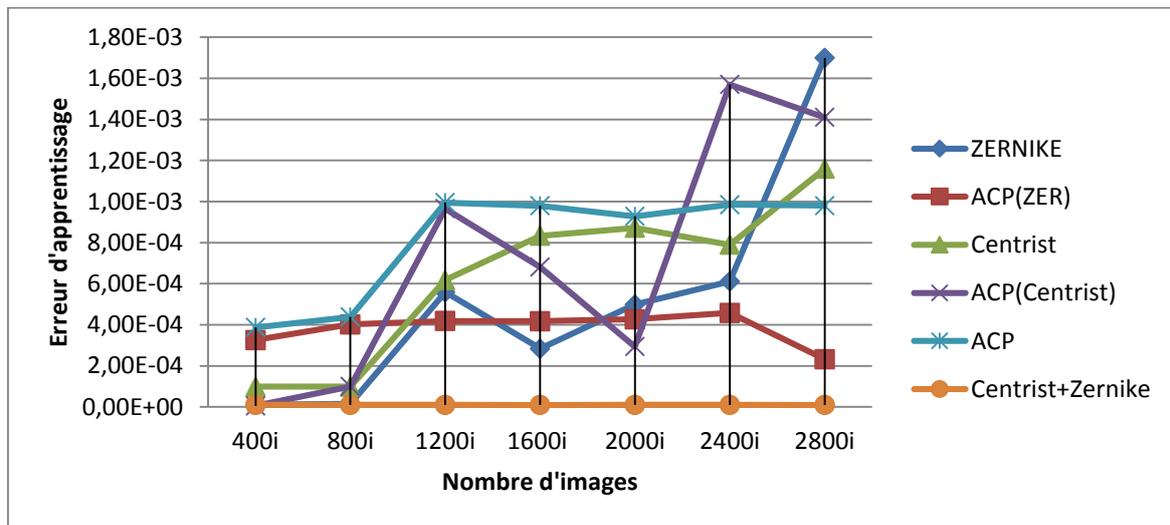


Figure 4.4: Erreur d'apprentissage en fonction de l'ordre de la base

Pour finir on va présenter une étude comparative entre deux méthodes de classification, à savoir SVM et les réseaux de neurones, en calculant les taux de reconnaissance. D'après le tableau 4.3 et la figure 4.5 on constate que les résultats obtenus par SVM sont encourageants par rapport à ceux obtenus par les réseaux de neurones.

	RDN	SVM
M. Zernike	92%	97,65%
ACP (Zernike, 14)	18,37%	67,33%
D. CENTRIST	99,50%	100,00%
ACP(CENTRIST, 100)	93,42%	96,49%
CENTRIST + Zernike	99,25%	100,00%
ACP	79,01%	82,60%

Tableau 4.3: Taux de reconnaissance pour RDN et SVM avec (1200 images)

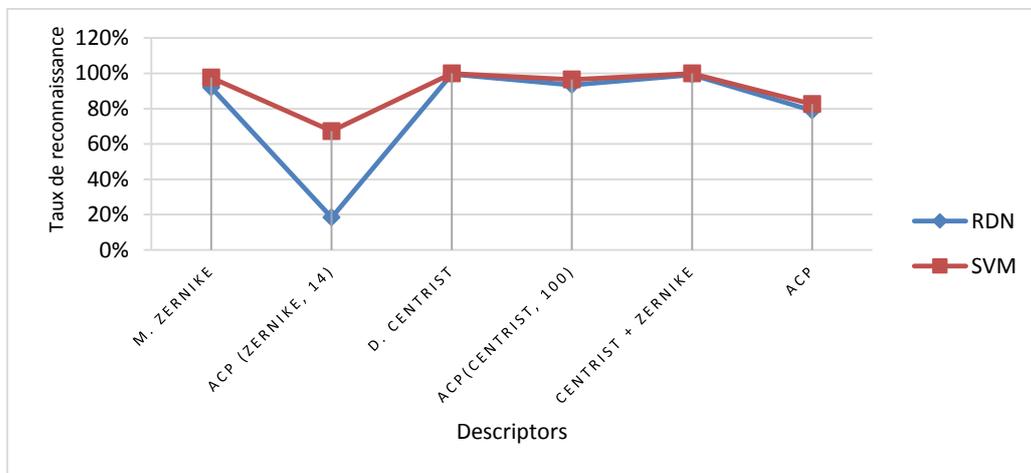


Figure 4.5 : Taux de reconnaissance pour RDN et SVM avec (1200 images)

### 4.3 Approche 3D/3D

Dans cette partie, nous présentons les résultats obtenus par l'approche 3D/3D d'indexation d'objets 3D utilisant la distance géodésique. D'après les méthodes présentées dans la section (2.3, 2.4), nous sommes en mesure de comparer la requête fournie par l'utilisateur avec chaque objet de la base de données.

En appliquant la méthode d'extraction basée sur le calcul de la distance géodésique, nous obtenons une distance unique pour chaque objet. En utilisant les méthodes de classification il est alors possible de retourner à l'utilisateur les modèles 3D qu'il cherche. La (Figure 1) donne une vue du système adopté à l'approche 3D. Les résultats expérimentaux sont présentés ci-dessous montrent l'efficacité de nos méthodes.

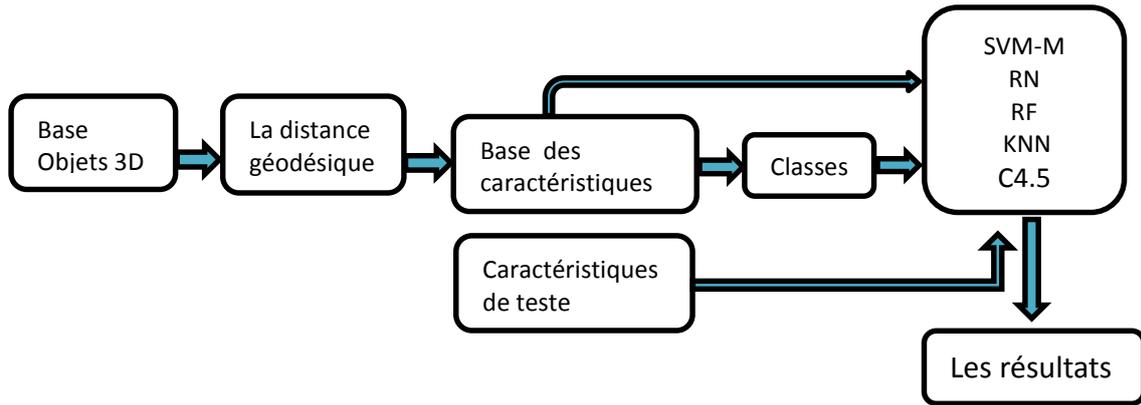


Figure 4.6 : Organigramme du système de reconnaissance 3D

### 4.3.1 Bases de données

Les bases d'objets 3D sont les briques essentielles pour évaluer un processus d'indexation et de recherche de modèles 3D. En particulier, nous utilisons deux bases pour évaluer notre processus d'indexation et de recherche. Il s'agit de deux bases de données qui sont : Princeton Shape Benchmark [101] et la base SHREC'07 Watertight [102]. Dans nos recherches, nous allons travailler sur des maillages de 200 à 10000 sommets, et que les descripteurs que nous proposons dans la suite de cette thèse sont adaptés aux trois dimensions. Cependant, nous ne travaillerons qu'avec des représentations polygonales.

#### a) Princeton

La base de données Princeton Shape Benchmark [101] est une base d'objets 3D mise à disposition par l'équipe du Princeton. Elle est décomposée en deux parties, l'une pour l'entraînement et l'autre pour l'évaluation. Chacune d'entre elles comporte 907 objets 3D généralement répartis, inégalement, dans 90 classes pour la première et 92 pour la seconde.

Il est intéressant de noter que ce regroupement en classes a été fait sur la sémantique des objets 3D plus que sur leur forme. En effet, il est possible de retrouver dans la classe "Escalier" des escaliers droits ou en colimaçons. Ce choix peut être handicapant lorsque l'on veut évaluer une méthode d'indexation par le contenu. Néanmoins, cette base offre une grande variété d'objets répartis dans un grand nombre de classes permettant une évaluation satisfaisante. D'autre part, sa

grande notoriété dans le monde de la recherche permet une comparaison par rapport aux autres équipes du domaine. La Figure 4.7 montre des exemples d'objets qui la composent.



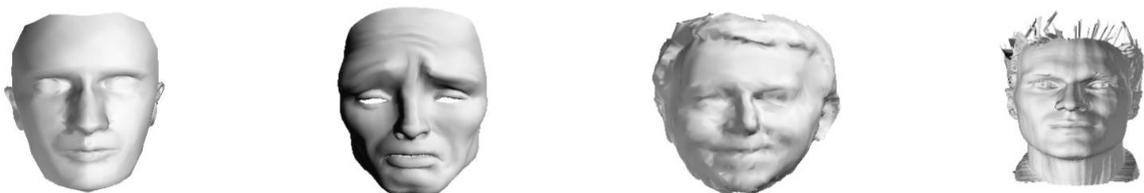
(a) Classe "avion commerciaux"



(b) Classe "chaise de bureau"



(c) Classe "main"



(d) Classe "visage"

Figure 4.7 : Exemple d'objets de quatre classes de la base Princeton [101]. (a), (b), (c) et (d) illustrent respectivement les classes "avion", "chaise", "main" et "visage".

## b) Watertight datasets

La base de données SHREC'07 Watertight [102] provient du concours annuel "SHREC" et regroupe les objets "articulés". Elle est composée de 400 modèles répartis équitablement dans 20 classes permettant ainsi une évaluation juste pour

chaque classe. Chacune d'entre elles comporte, en général, un même objet déformé au niveau des articulations. Il existe cependant des classes non articulées.

À cause du caractère spécialisé de cette base, il est indispensable de mettre en place une méthode prenant en compte, lors de la description des objets, les déformations articulaires. Cette particularité est généralement obtenue par des approches par squelette ou par la mise en œuvre d'une élasticité dans la mesure de dissimilarité. C'est la motivation qui nous a motivés à travailler sur cette base de données pour montrer l'efficacité de notre approche. Cette base est idéale pour tester et évaluer un processus complet d'indexation et de recherche d'objets 3D. La Figure 4.8 montre des exemples d'objets de cette base.



(a) Classe "main" (articulée)



(b) Classe "lunette" (articulée)



(c) Classe "humain" (articulée)



(d) Classe “table” (non articulée)

Figure 4.8 : Exemple d’objets de quatre classes de la base Watertight [102]. (a), (b), (c) et (d) illustrent respectivement les classes “main”, “lunette”, “humain”, “table”.

### 4.3.2 Résultats expérimentaux (Equation Eikonale)

Le premier objectif de cette expérience est de détecter le nombre minimal de courbes nécessaires pour catégoriser une famille d’objets 3D. Après cela, nous allons évaluer le nombre sélectionné à l’aide de différentes techniques.

Pour identifier le nombre des courbes géodésiques nécessaires pour catégoriser une famille, nous avons testé les combinaisons suivantes :

- 2 points de départ et 3 points d’arrivée, (6 courbes).
- 3 points de départ et 3 points d’arrivée, (9 courbes).
- 3 points de départ et 4 points d’arrivée, (12 courbes).
- 4 points de départ et 4 points d’arrivée, (16 courbes).
- 4 points de départ et 5 points d’arrivée, (20 courbes).

Nous avons choisi la méthode SVM multi-class pour évaluer les combinaisons proposées. Le Tableau 4.4 montre les taux de reconnaissance obtenu pour les différents noyaux de SVM. Il faut noter que le temps d’exécution et la performance du système dépendent essentiellement de la complexité des objets 3D. En général, ils sont liés au nombre de sommets et de faces de chaque maillage.

Le noyau est effectivement une mesure de similarité, afin de choisir un noyau selon la connaissance préalable est une bonne idée. En absence d’expertise, nous choisissons d’effectuer notre test en utilisant les quatre noyaux.

A part le noyau polynomial, tous les noyaux présentent de bons résultats. En ce qui concerne le nombre de courbes nécessaires pour classer la famille d’objets 3D

les meilleurs résultats ont été trouvés pour les deux ensembles composés de 9 et 12 courbes. La diminution du taux de classification pour les objets représentés par 20 courbes géodésiques peut être expliquée par le fait que plus le nombre de courbe augmente, les descripteurs deviennent plus spécifiques. En conséquence, il peut être utilisé pour classer un seul objet ou pour séparer l'objet de la même famille.

M-SVM ( kernel)	6 curves	9 curves	12 curves	16 curves	20 curves
Rbf	69,18%	98,46%	96,46%	94,47%	91,46%
Polynomial	41,13%	72,31%	73,31%	71,21%	68,31%
Quadratic	65,20%	98,23%	98,78%	95,13%	87,43%
Linear	68,36%	96,46%	97,16%	94,76%	89,55%

Tableau 4.4 : Les taux de reconnaissance pour un nombre différent de courbe en utilisant la distance géodésique appliquée à la base données Watertight.

Dans le second test, nous essayons d'évaluer l'efficacité de 9 courbes de détection et de similitude. Le tableau 4.5 montre des résultats obtenus en utilisant le KNN avec  $K = 4$ .

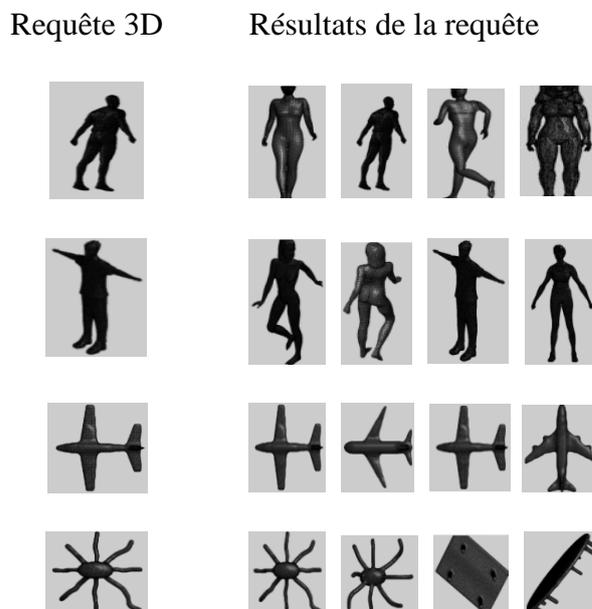


Tableau 4.5 : Rconnaissance d'objets 3D en utilisant la distance géodésique et KNN.

D'après le tableau 4.5, on peut voir qu'en utilisant la distance géodésique on arrive à reconnaître les trois premiers requêtes 3D. Par contre, la dernière requête n'a pas été reconnue entièrement, cela peut être expliqué par la complexité des formes semblables des objets. Ce qui montre l'efficacité de la distance géodésique comparant aux autres requêtes.

Dans la prochaine partie de l'expérience, nous avons évalué 9 courbes définies à l'aide de différentes techniques. D'abord par l'évaluation de l'intra-connexion des objets de la même classe et la seconde par la qualité d'objet distinct de la classe différente en utilisant l'ensemble de données.

a. Évaluation de classes inter-connexion

Pour évaluer la classe inter connexion de la base Watertight, nous commençons par l'analyse en composantes principales (ACP) pour analyser la matrice représentant l'ensemble des caractéristiques. Le tableau 4.6 montre les cinq premières composantes principales.

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	107.977441	97.909587	59.00 %		59.00 %
2	10.067854	0.340559	5.50 %		64.51 %
3	9.727296	2.395702	5.32 %		69.82 %
4	7.331594	1.973111	4.01 %		73.83 %
5	5.358483	0.359001	2.93 %		76.76 %

Tableau 4.6 : évaluation de la distance géodésique utilisant PCA

D'après le tableau 4.6 on peut clairement voir que la majorité de l'information est concentré dans la première valeur propre qui correspond au premier axe factoriel. Maintenant ce qui il faut faire c'est projeté la base des caractéristiques sur le premier axe et un autre axe pour avoir une projection visuelle classes d'appartenance.

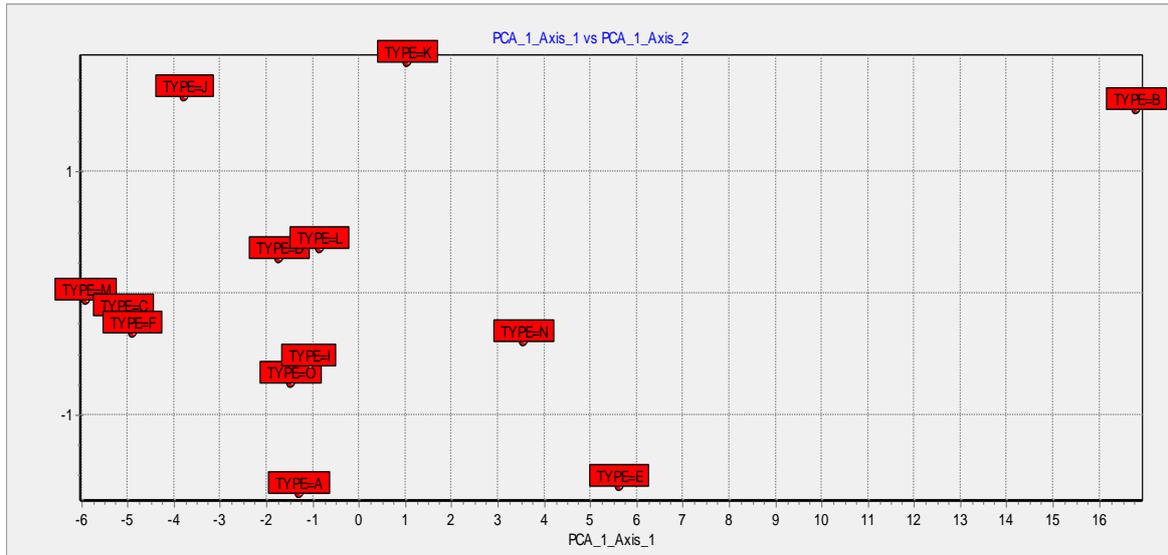


Figure x projection des caractéristiques sur les axes factoriels

La figure x montre que les classes obtenues sur les axes sont discernable. Cela donne une idée générale de capacité discriminatif des caractéristiques basées sur la distance géodésique.

b. Évaluation en utilisant des algorithmes de Data Mining

L'ensemble des modèles de données a été divisé en une base de données d'apprentissage et de test. La méthode 3D doit être formée sur la base de données d'apprentissage (sans influence de la base de données de test). Puis, après toute exploration est terminée et tous les paramètres algorithmiques ont été gelés, les résultats doivent être déclarés pour la base de données de test.

Nous évaluons le descripteur avec 9 courbes en utilisant des techniques de classification multi-classes comme ANN, KNN, C4.5 et forêt aléatoire (tableau 4.7).

	KNN	ANN	C4.5	RF
Class 1	0.3158	0.7895	0.7368	0.99
Class 2	0.45	0.85	0.7	0.98
Class 3	0.5	0.15	0.85	0.98
Class 4	0.25	0.1	0.5	0.97
Class 5	0.55	0.7	0.6	0.97
Class 6	0.4	0.8	0.5	0.96
Class 7	0.5	0.65	0.6	0.96
Class 8	0.4	0.3	0.6	0.95
Class 9	0.15	0.8	0.5	0.95
Class 10	0.2	0.45	0.75	0.94
Class 11	0.8	0.35	0.85	0.94
Class 12	0.45	0	0.6	0.93
Class 13	0.55	0.1	0.5	0.93
Computing times	936ms	4618ms	1888ms	172ms

Tableau 4.7 : Taux de reconnaissance de la base Watertight

D'après le tableau 4.7 on remarque que la méthode Random Forest donne toujours un taux de reconnaissance meilleur que celui obtenu par les autres méthodes de classifications. Cela vient des raisons suivant : concernant les réseaux de neurones pour augmenter les taux il faut changer la configuration c.-à-d. ajouté d'autres couches cachées, tandis que pour le KPPV et C 4.5 sont limités.

### 4.3.3 Résultats expérimentaux (Equation de la chaleur)

Pour évaluer l'efficacité de cette méthode, l'utilisation d'une base de données de référence est nécessaire (comme "référence Princeton") pour permettre des comparaisons avec d'autres travaux internationales. Cela nous a conduits à chercher une méthode efficace pour la classification des données, par exemple la méthode des Forêts Aléatoires. Cette méthode est basée sur plusieurs arbres de décision. Chaque arbre dépend des valeurs d'un vecteur aléatoire échantillonné indépendamment mais avec la même distribution. La méthode Random Forest vient comme une combinaison de plusieurs facteurs prédictifs de tous les arbres.

Avant de présenter les résultats expérimentaux, nous présentons les méthodes avec lesquelles nous avons comparé notre approche :

- En commençant par le descripteur de champ Linière (LFD) développé par (Chen, 2003) [59] leur méthode extrait les caractéristiques de 100 2D images et qui mesure la similarité entre deux objets 3D en trouvant la meilleure correspondance entre l'ensemble des vues 2D pour les deux objets.
- La méthode (Indriyati Atmosukarto & al, 2013) [103] est basée sur les vues saillants (SV) extraits sélectionnés uniquement sur la base des faits saillants de chaque objet.

Notre approche proposée pour la reconnaissance d'objets 3D est basée sur l'équation de la chaleur pour calculer la distance géodésique entre deux points à l'aide d'une transformation du noyau de la chaleur. La Figure 4.6 montre le schéma adopté dans notre système de reconnaissance.

L'ensemble des modèles de données a été divisé en une base de formation et une base de données de test. L'algorithme doit être formé sur la base de données de formation (sans influence de la base de données de test). Puis, après toute exploration est terminée et tous les paramètres algorithmiques ont été gelés, les résultats doivent être déclarés pour la base de données de test.

#### a. Evaluation 1

Le premier objectif de cette étude est d'évaluer la performance de la méthode de diffusion de la chaleur pour cette raison, nous nous sommes basés sur la classification en utilisant la validation croisée pour tester notre efficacité de la méthode.

Nous évaluons les descripteurs proposées en utilisant des techniques de classification multiclassées comme C4.5, Forêt aléatoire (RF) (Tableau 4.8).

	<b>C4.5</b>	<b>KNN</b>	<b>RF</b>
Classe 1	42%	5%	100%
Classe 2	80%	25%	100%
Classe 3	90%	90%	100%
Classe 4	65%	65%	100%
Classe 5	70%	20%	100%
Classe 6	50%	20%	100%
Classe 7	25%	40%	100%
Classe 8	65%	25%	100%
Classe 9	40%	30%	100%
Classe 10	65%	30%	100%
Classe 11	70%	15%	100%
Classe 12	85%	35%	100%
Classe 13	40%	45%	100%
Computing times	765 ms	281 ms.	62 ms

Tableau 4.8 : Taux de reconnaissance de la base donnée Watertight

En se basant sur le tableau 4.8 on remarque que les résultats obtenus par la méthode de Random Forest sont performantes au niveau des taux de reconnaissance par rapport aux autres méthodes de classification pour toutes les catégories.

#### b. Evaluation 2

Nous présentons ci-après les résultats obtenus à partir des méthodes décrites ci-dessus (Figure 4.9). Les tests ont été effectués sur 70 classes de la base de données Princeton. La performance a été systématiquement évaluée par le système de la figure 4.9.

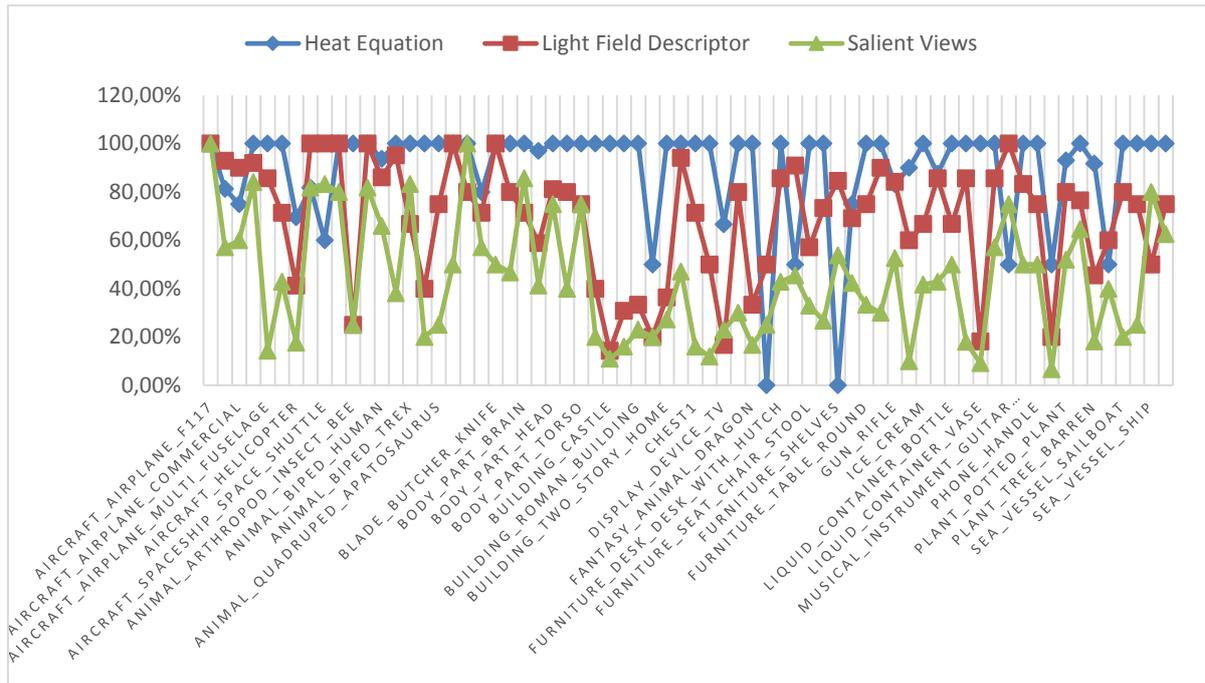


Figure 4.9 : Taux de reconnaissance obtenu pour les trois méthodes

La figure 4.9 présente les résultats expérimentaux pour les trois approches. La courbe (avec sommet sous forme losange) montre les résultats obtenus par l'application de notre méthode (EDC) et les deux autres montrent les résultats calculés (Indriyati Atmosukarto & al, 2013) [103].

Pour la plupart des classes, la comparaison des résultats montre clairement que les taux de reconnaissance calculée pour la méthode de l'équation de chaleur sont plus élevés que ceux trouvés pour les autres méthodes (LFD et SV). Cependant, il y a des classes (13 classes pour LFD et 5 classes SV) où les taux de reconnaissance sont plus faibles. Ce problème est dû à la complexité de la forme de l'objet. Les résultats de cette comparaison montrent que le procédé qui utilise l'équation de diffusion de chaleur pour caractériser l'objet 3D donne des résultats remarquablement supérieurs aux autres méthodes. De façon générale, le taux de reconnaissance globale se trouve dans l'ordre de 90%, contrairement aux autres approches dont les taux de reconnaissance globaux ne dépassent pas 76%.

### c. Evaluation 3

Nous avons aussi fait la comparaison avec la méthode qui extrait les informations directement à partir des objets 3D [78]. Cette étude nous permet d'évaluer la

performance de notre méthode [80] par rapport à la méthode qui calcule la distance géodésique sur des objets 3D à l'aide de la résolution de l'équation eikonale. Cette comparaison est effectuée en utilisant la base de données Watertight et la classification par Random Forest.

La méthode basée sur l'équation de la chaleur donne le même taux de reconnaissance pour les 13 classes avec une valeur égale à 100%. D'autre part la méthode géodésique donne un taux de reconnaissance plus faible diminue linéairement de la première classe à la dernière.

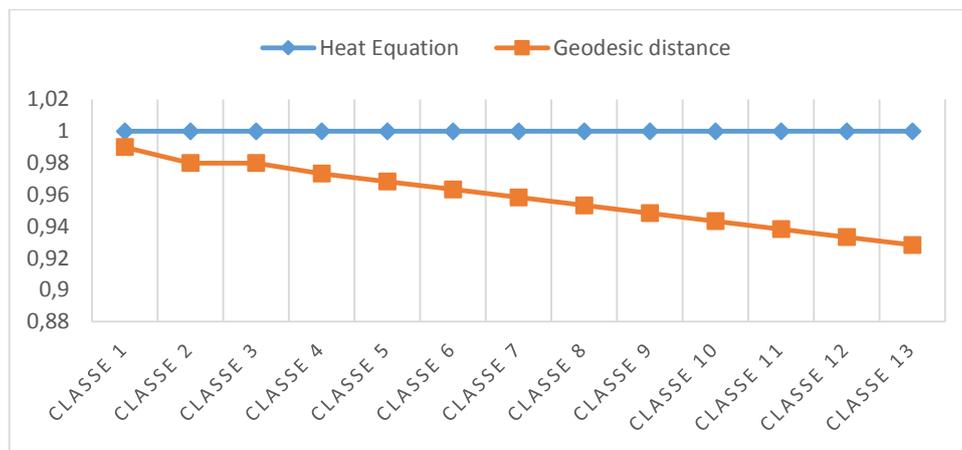


Figure 4.10 : étude comparative des taux de reconnaissance pour l'équation de la chaleur et de distance géodésique en utilisant Forêts Aléatoires

Alternativement, on peut dire que la méthode de l'équation de la chaleur est meilleure à tous les niveaux de classification par validation croisée, comme indiqué dans le tableau suivant :

	KNN	C4.5	RF	Complexité
Equation de la chaleur	281ms	765ms	62ms	$O(N)$
Equation Eikonale	936ms	1888ms	172ms	$O(N \log(N))$

Tableau 4.9 : Etude comparative du temps et complexité entre les méthodes, l'équation de chaleur et équation Eikonale.

## 4.5 Conclusion

Dans la première partie (Approche 2D/3D), on a mis en œuvre les résultats expérimentaux en adoptant plusieurs méthodes pour bien discriminer les objets couleurs. Après une comparaison entre les résultats obtenus par les descripteurs hybride et celles obtenus par les descripteurs (individuellement), on a constaté que les méthodes hybrides sont plus performantes au niveau du taux de reconnaissance. D'une autre part, l'utilisation du SVM comme classificateur rend les résultats plus pertinents. En revanche, différentes voies peuvent être envisagées en ce qui concerne l'amélioration de l'approche (hybride) qui a été développée.

- L'optimisation d'algorithme de recherche pour minimiser le temps d'exécution.
- L'application de cette approche sur l'indexation 3D/3D.

Dans la deuxième partie, les éléments fondamentaux d'un système de reconnaissance d'objets par le contenu ont été présentés. Une grande attention a été dédiée à la classification et à la description d'objets 3D dans la base de données. D'une part, nous avons présenté les résultats de l'approche proposée pour la reconnaissance des objets 3D basée sur le calcul de la distance géodésique, et sur l'étude du nombre de courbes géodésiques nécessaires pour catégoriser une famille d'objets 3D. Les résultats de cette évaluation sont basés sur les différentes techniques de classification et de classement. Nous pouvons conclure qu'un petit nombre de courbes géodésiques (entre 9 et 12) est suffisant pour caractériser une famille d'objets 3D. D'une autre part, le calcul de la distance géodésique basé sur la résolution de l'équation de la chaleur offre des résultats encourageants aux niveaux du temps de recherche ainsi que le taux de reconnaissance.

## Conclusions et perspectives

### Travail réalisé :

Dans cette thèse, nous nous sommes intéressés aux problématiques associées à l'indexation et à la recherche d'objets 3D dans des bases de données bien précises. Nos principaux objectifs ont porté sur le développement de nouvelles approches permettant l'indexation d'une bibliothèque de modèles 3D à partir d'une requête en deux, ou en trois dimensions.

Dans un premier temps, nous avons testé des descripteurs 2D classiques pour identifier les projections des objets 3D. Pour le cas de 2D, les résultats obtenus n'étaient pas encourageantes, cela nous a poussés à chercher d'autres pistes pour augmenter les taux de reconnaissances. Pour cela nous avons élaboré une nouvelle approche (approche hybride) où on a combiné entre les moments de Zernike et le descripteur Centrist. Ce qui a donné de bons résultats de reconnaissance comparant aux approches classiques.

La grande partie de ce travail a porté sur la reconnaissance directe des objets 3D. En effet, nous avons proposé deux nouvelles approches. La première est l'utilisation de la distance géodésique pour calculer la distance minimale entre deux points de deux objets différents. Cette distance est calculée via la résolution de l'équation Eikonale. Les résultats de reconnaissance obtenus ont été assez prometteuses, en revanche le temps de calcul était très grand provenant de la complexité de la résolution de l'équation d'Eikonale (équation non linéaire). Pour remédier au problème de temps de calcul, nous avons pensé à utiliser l'équation de la chaleur à la place de l'équation d'Eikonale, tout en préservant le processus de minimisation de la distance. L'idée derrière cette proposition est d'utiliser une équation linéaire facile à résoudre avec les mêmes propriétés que celle de l'équation Eikonale. Les résultats ont été pertinents surtout au niveau du temps de calcul. Malgré les bons résultats obtenus, cette méthode souffre du problème de lissage des objets qui contiennent des discontinuités dans les bords des objets, ce qui perturbe parfois la reconnaissance des objets avec des structures complexes. En effet, cela est dû à l'effet régularisant de cette équation qui rend les données discontinues infiniment continues.

Finalement, on a proposé une étude comparative entre les différentes méthodes proposées avec les approches classiques. Il s'est avéré que dans la plupart des exemples traités, les approches avec l'équation Eikonale et l'équation de la chaleur sont meilleurs. La seule différence entre ces deux approches est le temps de calcul qui favorise le choix de l'équation de la chaleur en dépend de l'Eikonale.

### **Perspectives :**

Le travail élaboré tout au long de cette thèse a conduit à de bons résultats concernant l'amélioration du taux de reconnaissances des objets 3D. Néanmoins, il reste toujours d'autres pistes à creuser, en particulier, comment reconnaître des objets 3D dont la forme est complexe ? Pour répondre à cette question plusieurs perspectives s'ouvrent. Nous proposons d'énumérer les plus importantes :

- Approfondir l'analyse et le développement de l'approche hybride proposée.
- Optimiser le temps de calcul.
- Raffiner l'étape de prétraitement pour des données bruitées.
- Formuler un descripteur tridimensionnel en rajoutant une information sur la texture
- amélioré les résultats de reconnaissance à l'aide d'un système distribué.

D'une autre part, l'amélioration du temps de reconnaissance est toujours un point important surtout au niveau de commercialisation. En effet, pour vendre un algorithme de reconnaissance à une société de renommée, le temps de calcul joue un rôle primordial pour avoir un agrément mutuel. On peut donner comme exemple les applications sur les smartphones qui exigent un temps de calcul ne dépassent pas les 5 secondes. Donc pour améliorer les algorithmes proposés dans cette thèse au niveau du temps de reconnaissance, d'autres schémas de résolution numérique des équations étudiées doivent être perfectionnés. Une autre voie consiste à utiliser les techniques d'apprentissage automatique pour accélérer la reconnaissance des objets 3D, ce qui fera l'objet des futurs travaux.

## Bibliographie

- [1] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. 'Turning mobile phones into 3D scanners'. In Proceedings of the conference on Computer Vision and Pattern Recognition, 2014.
- [2] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. 'Live metric 3D reconstruction on mobile phones'. In Proceedings of International Conference on Computer Vision, 2013.
- [3] Google earth. <https://www.google.com/earth/>.
- [4] ShapeNet. <https://shapenet.cs.stanford.edu>.
- [5] Trimble 3D warehouse. <https://3dwarehouse.sketchup.com/>.
- [6] R. O. Duda, P. E. Hart, and D. G. Stork, 'Pattern Classification'. 2nd edition, John Wiley & Sons, Inc, 2001.
- [7] F.R. Al-Osaimi, M. Bennamoun, and A. Mian. 'Integration of local and global geometrical cues for 3D face recognition'. Pattern Recognition, 41:1030–1040, 2008.
- [8] Yi Liu, Hongbin Zha, and Hong Qin. 'The generalized shape distributions for shape matching and analysis'. In Shape Modeling and Applications, 2006.
- [9] M. Elad, A. Tal, and S. Ar. 'Content based retrieval of vrmf objects: an iterative and interactive approach'. In Euro graphics workshop on Multimedia, pages 107–118, 2001.
- [10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. 'Shape distributions'. ACM Transactions on Graphics, 21:807–832, 2002.
- [11] R. Ohbuchi, T. Minamitani, and T. Takei. 'Shape-similarity search of 3D models by using enhanced shape functions'. International Journal of Computer Applications in Technology, 23(2):70–85, 2005.
- [12] T. Zaharia and F. Preteux. '3D shape based retrieval within the mpeg-7 framework'. SPIE Applications, 4304, 2001.

- [13] M. Mahmoudi and G. Sapiro. 'Three-dimensional point cloud recognition via distributions of geometric distances'. In CVPR, 2008.
- [14] A. Ion, N. Artner, G. Peyre, S. Marmol, W. Kropatsch, and L. Cohen. '3D shape matching by geodesic eccentricity'. In CVPR, 2008.
- [15] D. Saupe and D. Vranic. '3D model retrieval with spherical harmonics and moments'. In DAGM Symposium on Pattern Recognition, pages 392–397, 2001.
- [16] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. 'Rotation invariant spherical harmonic representation of 3D shape descriptors'. In Eurographics, pages 156–164, 2003.
- [17] Hamid Laga, Hirko Takahashi, and Masayuki Nakajima. 'Spherical wavelet descriptors for content-based 3D model retrieval'. In Shape Modeling and Applications, pages 15–23, 2006.
- [18] H. Laga and M. Nakajima. 'A boosting approach to content-based 3D model retrieval'. In GRAPHITE, pages 227–234, 2007.
- [19] Peter Schroder and Win Sweldens. 'Spherical wavelets: Efficiently representing functions on the sphere'. In SIGGRAPH, pages 161–172, 1995.
- [20] E. Praun and H. Hoppe. 'Spherical parametrization and remeshing'. In SIGGRAPH, pages 340–349, 2003.
- [21] Zhenbao Liu, Jun Mitani, Yukio Fukui, and Seiichi Nishihara. 'Multiresolution wavelet analysis of shape orientation for 3D shape retrieval'. In ACM Multimedia Information Retrieval, pages 403–410, 2008.
- [22] J. Assfalg, A. Del Bimbo, P. Pala, 'Curvature Maps for 3D CBR'. International Conference on Multimedia and Expo (ICME '03), volume 1, pages 301–304. Baltimore, MD, Etat-Unis. 2003.
- [23] J. Tangelder and R. Veltkamp. 'Polyhedral model retrieval using weighted point sets'. In SMI, 2003.

- [24] Georgios Passalis and Theoharis Theoharis. 'Intraclass retrieval of nonrigid 3D objects: Application to face recognition'. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):218– 229, 2007.
- [25] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bulow, and Jitendra Malik. 'Recognizing objects in range data using regional point descriptors'. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–237, 2004.
- [26] Andrew Johnson and Martial Hebert. 'Using spin images for efficient object recognition in cluttered 3D scenes'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433 – 449, May 1999.
- [27] P. Shilane and T. Funkhouser. 'Selecting distinctive 3D shape descriptors for similarity retrieval'. In *Shape Modeling International*, pages 18–25, 2006.
- [28] P. Shilane and T. Funkhouser. 'Distinctive regions of 3D surface'. *ACM Transactions on Graphics*, 26(2), 2007.
- [29] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. 'Mesh saliency'. *ACM Trans. Graph.*, 24(3):659–666, 2005.
- [30] Xinju Li and Igor Guskov. 'Multi-scale features for approximate alignment of point based surfaces'. In *Euro graphics Symposium on Geometry Processing*, 2005.
- [31] Xinju Li and Igor Guskov. '3D object recognition from range images using pyramid matching'. In *ICCV Workshop on 3dRR*, 2007.
- [32] Ranjith Unnikrishnan and Martial Hebert. 'Multi-scale interest regions from unorganized point clouds'. In *CVPR*, 2008.
- [33] Kouki Watanabe and Alexander G. Belyaev. 'Detection of salient curvature features on polygonal surfaces'. *Computer Graphics Forum*, 20(3), 2001.
- [34] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. 'Sparse point matching by combining 3D mesh saliency with statistical descriptors'. *Computer Graphics Forum*, 27(2):643–652, 2008.

- [35] Ryutarou Ohbuchi, Kunio Osada, Takahiko Furuya, and Tomohisa Banno. 'Salient local visual features for shape-based 3D model retrieval'. In *Shape Modeling International*, pages 93–102, 2008.
- [36] John Novatnack, Ko Nishino, and Ali Shokoufandeh. 'Extracting 3D shape features in discrete scale space'. In *3D Data Processing, Visualization, and Transmission*, pages 946–953, 2006.
- [37] J. Novatnack et K. Nishino. 'Scale-dependent 3D geometric features'. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [38] E. Akagunduz et Ilkay Ulusoy. '3D object representation using transform and scale invariant 3D features'. In *ICCV 3dRR Workshop*, pages 1–8, 2007.
- [39] B. Taati, M. Bondy, P. Jasiobedzki, and M. Greenspan. 'Variable dimensional local shape descriptors for object recognition in range data'. In *ICCV*, 2007.
- [40] M. Andreetto, N. Brusco, and G. M. Cortelazzo. 'Automatic 3d modeling of textured cultural heritage objects'. *Image Processing, IEEE Transactions on*, 13(3):354–369, 2004.
- [41] A. E. Johnson and M. Hebert. 'Using spin images for efficient object recognition in cluttered 3D scenes'. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.
- [42] S. Belongie, J. Malik, and J. Puzicha. 'Shape matching and object recognition using shape contexts'. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [43] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein. 'Intrinsic shape context descriptors for deformable shapes'. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, pages 159–166, 2012.
- [44] M. Kortgen, G.-J. Park, M. Novotni, and R. Klein. '3d shape matching with 3d shape contexts'. In *The 7th central European seminar on computer graphics*, volume 3, pages 5–17, 2003.

- [45] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. 'Surface feature detection and description with applications to mesh matching'. In Proceedings of the conference on Computer Vision and Pattern Recognition, pages 373–380, 2009.
- [46] M. Kac. 'Can one hear the shape of a drum?' *American Mathematical Monthly*, pages 1–23, 1966.
- [47] R. M. Rustamov. Laplace-beltrami 'eigen functions for deformation invariant shape representation'. In Proceedings of the fifth Euro graphics symposium on Geometry processing, pages 225–233. Euro graphics Association, 2007.
- [48] J. Sun, M. Ovsjanikov, and L. Guibas. 'A concise and provably informative multiscale signature based on heat diffusion'. In *Computer Graphics Forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [49] M. M. Bronstein and I. Kokkinos. 'Scale-invariant heat kernel signatures for non-rigid shape recognition'. In Proceedings of the conference on Computer Vision and Pattern Recognition, pages 1704–1711, 2010.
- [50] D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel. 'Volumetric heat kernel signatures'. In Proceedings of the ACM workshop on 3D object retrieval, pages 39–44, 2010.
- [51] Richard O. Duda, Peter E. Hart, and David E. Stork. 'Pattern Classification'. Wiley, New York, second edition, 2001.
- [52] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. 'Statistical Pattern Recognition: A Review'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.
- [53] P. Bhattacharya, H. Liu, A. Rosenfeld, S. Thompson, 'Hough- transform detection of lines in 3D space'. *Pattern Recognition Letters*, 21(9):843–849. 2000.
- [54] T. Zaharia, F. Prêteux, 'Indexation de maillages 3D par descripteurs de forme'. 13ème Congrès Francophone de Reconnaissance de Forme et Intelligence Artificielle (RFIA '02), volume 1, pages 48–57. Angers, France. 2002.

- [55] Vranic D.V. et Saupe D. '3D shape descriptor based on 3D Fourier transform' Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001). - 2001. - pp. 271-274.
- [56] H. Sundar, D. Silver, N. Gagvani, and S. Dickenson. 'Skeleton-based shape matching and retrieval'. In Shape Modeling International, pages 130–138, 2004.
- [57] M. Hilaga, Y. Shinagawa, and T. Kohmura. 'Topology matching for fully automatic similarity estimation of 3D shapes'. In SIGGRAPH, pages 203–212, 2001.
- [58] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. 'On visual similarity based 3D model retrieval'. Computer Graphics Forum, 22(3):223–232, 2003.
- [59] Yuehong Wang, Rujie Liu, Takayuki Baba, Yusuke Uehara, Daiki Masumoto, and Shigemi Nagata. 'An images-based 3D model retrieval approach'. In Advances in Multimedia Modeling, pages 90–100, 2008.
- [60] N. Vajramushti, I. A. Kakadiaris, T. Theoharis, and G. Papaioannaou. 'Efficient 3D object retrieval using depth images'. In MIR, 2004.
- [61] D. Vranic. 'Desire= a composite 3D shape descriptor'. In ICME, 2005.
- [62] A. Oliva and A. Torralba. 'Modeling the shape of the scene: A holistic representation of the spatial envelope'. International Journal of Computer Vision, 42(3):145–175, 2001.
- [63] T. Malisiewicz, A. Gupta, and A. A. Efros. 'Ensemble of exemplar-svms for object detection and beyond'. In Proceedings of International Conference on Computer Vision, 2011.
- [64] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. 'Data-driven visual similarity for cross-domain image matching'. ACM Transactions on Graphics (Proc. SIGGRAPH Asia), 30(6), 2011.
- [65] Wu, J. and Rehg, J.M. 2011. 'CENTRIST: a visual descriptor for scene categorization'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1489- 1501.

- [66] Lazebnik, S., Schmid, C., and Ponce, J. 2006. 'Beyond bags of features: spatial pyramid matching for recognizing natural scene categories'. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169-2178.
- [67] Zabih, R. and Woodfill, J. 1994. 'Non-parametric local transforms for computing visual correspondence'. Proceedings of European Conference on Computer Vision, vol. 2, pp. 151-158.
- [68] M. R. Teague, "Image analysis via the general theory of moments," Journal of the Optical Society of America, vol. 70, no. 8, pp. 920-30, 1980.
- [69] A. Wallin, and O. Kübler, "Complete sets of complex Zernike moment invariants and the role of the pseudo invariants," IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 17, no. 11, pp. 1106-10, 1995.
- [70] M. Pawlak, 'Image Analysis by Moments: Reconstruction and Computational Aspects'. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2006.
- [71] S.-K. Hwang, and W.-Y. Kim, "A novel approach to the fast computation of Zernike moments," Pattern Recognition, vol. 39, no. 11, pp. 2065-76, 2006.
- [72] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4-37, January 2000.
- [73] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. Image and vision computing, 10(3):145-155, 1992.
- [74] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In Proceedings of the conference on Computer Vision and Pattern Recognition, pages 1-8, 2007.
- [75] J. M. ITCHELL 'Geometric shortest paths and network optimization'. In Handbook of Computational Geometry. Elsevier Science, Amsterdam, 1998.

- [76] E. W. DIJKSTRA, T. H. Eindhoven, 'A short introduction to the art of programming', Ewd316 The Netherlands. August 1971.
- [77] G. Peyré, M. Péchaud, R. Keriven and L. D. Cohen "Geodesic Methods" Computer Vision and Graphics. vol. 5, Nos. 3-4 197-397 2009.
- [78] Driss Naji, M. Fakir, O. Bencharef. "3D Objects Retrieval using Geodesic Distance Based on Eikonal equation". TELKOMNIKA Indonesian Journal of Electrical Engineering (ISSN: 2302-4046). Vol 15, No 3, Septembre 2015.
- [79] S. R. S. Varadhan. (1967) 'On the behavior of the fundamental solution of the heat equation with variable coefficients'. Communications on Pure and Applied Mathematics 20, 2, 431-455.
- [80] Driss Naji, M. Fakir, O. Bencharef. "3D Objects Retrieval using Geodesic Distance based on Eikonal equation". Second international Conference on Business Intelligence (CBI'15) Béni-Mellal, 2015.
- [81] Driss Naji, M. Fakir, R. Elayachi "Recognition of 3D Objects using Heat Diffusion Equations and Random Forests". Second international Conference on Business Intelligence (CBI'15) Béni-Mellal, 2015.
- [82] G. Dreyfus, "Réseaux de neurones, méthodes et applications", Eyrolles, 2004.
- [83] I. Rivals, L. Personnaz, G. Dreyfus, " Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception et illustrations industrielles", Ecole supérieure de physique et de chimie industrielles de la ville de paris laboratoire d'électronique, 1995.
- [84] C. Timothée, 'Reconnaissance de formes par réseau de neurones', Cours école polytechnique, 2002.
- [85] Claude Touzet, 'Les réseaux de neurones artificiels'. Cours, 1992.
- [86] M. Parizeau, "Réseaux de neurones", université Laval, cours, 2004

- [87] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, "Dimensionality reduction via sparse support vector machines". *Journal of Machine Learning Research*, 3:1229–1243, 2003.
- [88] D. Ritendra, J. Dhiraj, L. Jia, J. Wang, "Studying aesthetics in photographic images using a computational approach". *Computer Science, Proceedings of the European Conference on Computer Vision*, vol 3, p. 288–301, 2006.
- [89] Y. Chen, J. Bi, J. Wang, "Miles: Multiple-instance learning via embedded instance selection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947. 2006.
- [90] V. Vapnik, S. Kotz, "Estimation of Dependences Based on Empirical Data", *Springer Series in Statistics*, ISBN: 978-0387907338, 1982.
- [91] Bernhard Schölkopf, Alexander J. Smola, "Learning with Kernels: Support Vector Machines", *Regularization, Optimization and Beyond*, MIT Press, p.190, 2002.
- [92] Eve Mathieu-Dupas. 'Algorithme des k plus proches voisins pondérés et application en diagnostic'. 42èmes Journées de Statistique, Marseille, France, inria-00494814, 2010.
- [93] T. Kaungo, D. M. Mount, et al. 'An efficient k-means clustering algorithm: Analysis and implementation'. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, vol. 24, 881–892, 2002.
- [94] Ron Kohavi and Ross Quinlan. 'Decision tree discovery'. In *Handbook of Data Mining and Knowledge Discovery*, pages 267–276. University Press, 1999.
- [95] Olshen Stone Breiman, Friedman. 'Classification and regression tree'. 2011.

- [96] Leo Breiman. 'Random forests'. *Machine Learning*, 45(1):5–32, 2001.
- [97] J. Ross Quinlan. 'Induction of decision trees'. *Machine Learning*, 1(1):81–106, 1986.
- [98] L. Breiman. 'Bagging predictors'. *Machine Learning*, 24(2), 1996.
- [99] T. Dietterich. 'An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization'. *Machine Learning*, 40(2), 2000.
- [100] la base Coil-100 est sur le site <http://www.cs.columbia.edu/CAVE/>
- [101] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser, 'The Princeton Shape Benchmark', *Proceedings of SMI: International Conference on Shape Modeling and Applications* 167–178, 2004.
- [102] Daniela Giorgi, Silvia Biasotti, Laura Paraboschi, 'Watertight Models Track' *Shape Retrieval Contest 2007: CNR – IMATI Via De Marini 6, 16149, Genoa, Italy. 2007.*
- [103] Indriyati Atmosukarto & Linda G. Shapiro.' 3D object retrieval using salient views'. *International Journal of Multimedia Information Retrieval* June 2013, Volume 2, Issue 2, pp 103-115 Springer-Verlag, 2013.