Centre d'Etudes Doctorales
Sciences et Techniques

Mémoire de thèse de Doctorat

Présenté par

Ilyass OUAZZANI TAYBI

Pour obtenir le diplôme de Doctorat National

Formation Doctorale : Mathématique Appliquée et Informatique

Spécialité : ***Informatique***

Sous le thème

Elaboration d'un système de classification et de recherche d'objets 3D basé sur les images de coupe 2D et les réseaux de neurones convolutifs

Soutenue le : 01/07/2021

M. LAKHOUILI Abdellah	Université Hassan 1er, FST Settat	Président
M. MARZAK Abdelaziz	Université Hassan II, FS Casablanca	Rapporteur
M. ABDALI Abdelmounaïm	Université Cadi Ayyad, FST Marrakech	Rapporteur
M. MOUMOUN Lahcen	Université Hassan 1er, ENSA Berrechid	Rapporteur
M. ABDELMOUNIM El Hassane	Université Hassan 1er, FST Settat	Examinateur
M. ALAOUI Rachid	Université Mohamed V, EST Salé	Co-Directeur de thèse
M. GADI Taoufiq	Université Hassan 1er, FST Settat	Directeur de thèse

Dédicaces

A mes chers parents

A ma chère sœur,

A mes chers frères,

A la mémoire de mes chers grands parents,

A toute ma famille.

Résumé

Avec le progrès rapide des instruments de saisie d'objets 3D et de la puissance de calcul, il existe un nombre croissant d'objets 3D dans de différents domaines, tels que la simulation médicale, la vision par ordinateur, la conception assistée par ordinateur, l'infographie et la conception architecturale. Par opposition à la classification et à la recherche basée sur des données en 2D, la classification et la recherche à partir d'informations en 3D est une tâche progressivement viable et judicieuse. À ce titre, la classification et la recherche d'objets 3D est un sujet de recherche pertinent et pratique. En conséquence, il a récemment attiré une attention considérable de la part des chercheurs.

Les premiers travaux de classification et de recherche d'objets 3D sont généralement fondés sur leur forme, pour lesquelles des techniques basées sur des caractéristiques de bas niveau et des techniques basées sur des structures de haut niveau ont été utilisées. Ces approches peuvent généralement être divisées en deux catégories : celles basées sur une correspondance globale et d'autres basées sur une correspondance partielle. Récemment, l'efficacité des approches d'apprentissage approfondi, en particulier les réseaux neuronaux convolutifs, a accéléré le développement de la classification et de la recherche d'objets 3D, et a démontré leur prédominance par rapport aux approches traditionnelles.

Dans cette thèse, nous nous sommes penchés sur des approches de classification et de recherche basées sur la correspondance partielle, représentant la forme à l'aide des images de coupe 2D extraites de l'objet 3D. Ce type de méthodes requiert de mettre en place un ensemble de traitements distincts, notamment : le prétraitement des modèles 3D pour les rendre invariants aux transformations affines, la description pertinente de leur forme, et le processus de mise en correspondance qui doit permettre de répondre aux demandes de l'utilisateur.

Dans un premier temps, nous présentons trois nouvelles méthodes d'indexation et de recherche d'objets 3D basées sur les images de coupe 2D et des algorithmes d'exploration de données. L'idée clé est de représenter l'objet 3D par une série initiale d'images de coupe 2D, et d'en sélectionner les plus représentatives en appliquant des algorithmes d'exploration de données. Ainsi, la correspondance entre les objets 3D soit transformée en mesure de similarité entre leurs images de coupe 2D représentatives. Dans un deuxième temps, nous proposons une nouvelle approche de classification et de recherche d'objet 3D qui tire profit de l'efficacité et de la puissance de l'apprentissage approfondi afin d'extraire les caractéristiques les plus discriminantes d'images de coupe 2D. En particulier, l'approche proposée consiste à utiliser les images de coupe 2D d'objets 3D en vue d'entraîner un réseau de neurones convolutifs 3D à l'extraction des caractéristiques de haut niveau qui vont être utilisées, par la suite, pour concevoir des descripteurs de forme robustes.

Mots clés : Objets 3D, classification et recherche d'objets 3D, descripteur de forme, mesures de similarité partielle, exploration de données, réseaux de neurones convolutifs.

Abstract

With the rapid advancement of 3D object capturing instruments and computing power, there is a growing number of 3D models in different areas, such as medical simulation, computer vision, computer-aided design, computer graphics and architectural design. As opposed to model recognition and retrieval on 2D data, classifying and retrieving models from 3D information is a progressively viable and sensible errand. In that capacity, addressing 3D model classification and retrieval is a pertinent and convenient research subject. As a result, it has lately drawn considerable attention from researchers.

Early works on the classification and retrieval of 3D objects are generally based on their shape, for which techniques based on low-level features and techniques based on high-level structures have been used. These approaches can generally be divided into two categories: those based on global correspondence and those based on partial correspondence. Recently, the effectiveness of deep learning approaches, in particular convolutional neural networks, has accelerated the development of 3D object classification and retrieval, and has demonstrated their predominance over traditional approaches.

In this thesis, we investigated classification and retrieval approaches based on partial correspondence, representing the shape using 2D slices extracted from the 3D object. This type of methods requires the implementation of a set of distinct treatments, in particular: the pre-processing of 3D models to make them invariant to affine transformations, the relevant description of their form, and the matching process that must allow satisfying the user's requests.

First, we present three new methods for indexing and retrieval of 3D objects based on 2D slices and data mining algorithms. The key idea is to represent the 3D object by an initial set of 2D slices, and to select the most representative ones by applying data mining algorithms. In this way, the correspondence between the 3D objects is transformed into a measure of similarity between their representative 2D slices. In a second time, we propose a new 3D object classification and retrieval approach that takes advantage of the efficiency and power of machine learning to extract the most discriminating features from 2D slices. In particular, the proposed approach consists in using 2D slices of 3D objects to train a 3D convolutional neural network to extract high-level features that will then be used to design robust shape descriptors.

Keywords: 3D objects, 3D object classification and retrieval, shape descriptor, partial similarity measuring, data mining, convolutional neural networks.

Table de matières

Liste des Figures

Liste des Tableaux

Liste d'abréviations

AABB	Axis Aligned Bounding Box
ACP	Analyse en Composantes Principales
ACPC	Analyse en Composantes Principales Continue
AE	Auto-Encoders
AGD	Average-Geodesic Distance
ANN	Artificial Neural Networks
ARG	Attributed Relational Graphs
ASC	Adaptive Slices Clustering
BIC	Bayesian Information Criterion
BMU	Best Matching Unit
BoW	Bag-of-Words
CAH	Crease Angle Histograms
CAO	Conception assistée par ordinateur
CDBN	Convolutional Deep Belief Net
CNN	Convolutional Neural Network
DAL	Domain Adaption Layer
DBN	Deep Belief Networks
DCG	Discounted Cumulative Gain
DL	Deep Learning
DM	Distance Map
DS	Description Schemes
ECD	Extraction de Connaissances à partir des Données
ECLat	Equivalent Class Transformation
EGI	Extended Gaussian Image
EM	E-Measure
EMD	Earthmover's Distance
FAO	Fabrication assistée par ordinateur
FC	Fully Connected
FDA	Fisher Discriminative Analysis
FT	First-Tier
FP	Frequent Pattern
GCNN	Graph Convolutional Neural Networks
IAO	Ingénierie assistée par ordinateur
ICMD	Minimum Inter-Cluster Distance
IRM	Magnetic Resonance Imaging
IS	Initial Slices
IPA	Inertial Principal Axes
KNN	k-Nearest Neighbor
KRS	K Representative Slices

LFED	Local Function Energy Distribution
LFD	Light Field Descriptor
LBO	Laplacian Beltrami Operator
mAP	mean Average Precision
MCRBM	Machines Boltzmann à Maillage Convolutionnel Restreint
MICD	Mean Intra-Cluster Distance
MVD-ELM	Multi-View Deep Extreme Learning Machine
MVCNN	Multi-View Convolutional Neural Network
NLP	Natural Language Processing
NN	Nearest Neighbor
NTU	National Taiwan University
N-DCG	Normalized Discounted Cumulative Gain
OBB	Oriented Bounding Box
PAM	Partition Around Medoids
PET	Positron Emission Tomography
PQ	Programmation Quadratique
PSB	Princeton Shape Benchmark
RBF	Radial Basis Functions
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RPN	Region Proposed Network
SGD	Stochastic Gradient Descent
SHD	Spherical Harmonics descriptor
SI-HKS	Scale-Invariant Heat Kernel Signature
SLCAE	Stacked Local Convolutional AutoEncoders
SOM	Self-Organizing Maps
SOFM	Self Organizing Feature Map
STN	Spatial Transformer Network
ST	Second-Tier
SS-BoF	Spatially Sensitive Bag-of-Features
SVM	Support Vector Machines
TFD	Transformée de Fourier Discrète
TFR	Transformée de Fourier Rapide
TFDI	Transformée de Fourier Discrète Inverse
TSDF	Truncated Signed Distance Function
VConv-DAE	Convolutional Volumetric Auto-Encoder
VRN	Voxception-ResNet
3D-CNN	3D Convolutional Neural Network

Introduction

Motivation

Les objets 3D sont utilisés dans de nombreuses applications, allant du rendu dans l'art numérique, les films et les jeux vidéo, à l'analyse du patrimoine culturel, à la simulation et la visualisation physique, à l'assistance en chirurgie médicale. La représentation numérique d'objets sous forme de surfaces ou de volumes tridimensionnels est devenue de plus en plus courante et accessible au cours des dernières décennies et semble s'accroître encore à l'avenir. Les objets 3D ne sortent pas seulement des mains des artistes professionnels. De nombreux utilisateurs peuvent utiliser des systèmes comme Google SketchUp [Gossweiler et al. 2006] pour créer leurs propres objets 3D. De plus, les objets 3D proviennent de la numérisation d'objets réels qui est devenue très accessible.

L'augmentation constante des objets 3D dans le monde présente des avantages évidents pour tous ceux qui travaillent avec ces données, car l'acquisition devient plus facile et moins coûteuse. Cependant, les métadonnées et la sémantique supplémentaire sont rares, lorsque les objets ne sont pas créés par des experts. La grande majorité des nouvelles données est un maillage brut de polygones, ce qui rend extrêmement difficile la recherche, le tri et la comparaison de ces données.

Pour cette raison, des techniques basées sur le contenu ont été développées. « Basé sur le contenu » signifie que la technique n'est pas basée sur des méta-données (par exemple des données textuelles) mais uniquement sur le contenu lui-même, c'est-à-dire les données 3D. L'objectif de toutes les approches est de fournir à l'utilisateur des informations supplémentaires sur les maillages polygonaux bruts. L'utilisateur peut avoir différentes tâches à l'esprit : rechercher un objet 3D spécifique dans une base de données d'objets, rechercher des objets similaires dans une base de données, explorer une base de données sans objet spécifique à l'esprit, rechercher des objets d'un certain type/classe/catégorie, comparer et trier des objets de la même classe ou de classes différentes. Dans tous les cas, l'utilisateur a besoin d'informations supplémentaires sur les objets. Ces informations peuvent consister en une liste triée par similarité avec un objet de référence, ou en une étiquette de classe ou de catégorie. Les informations peuvent également être numériques en quantifiant la similarité des objets ou en quantifiant les caractéristiques des objets.

Recherche et classification d'objets 3D

Toutes les tâches peuvent être généralisées à deux cas : Soit nous avons une requête et nous voulons trouver un objet, soit nous avons déjà un objet et nous voulons recueillir des informations à son sujet. Le premier cas correspond à la tâche de recherche d'objets 3D : nous avons un objet de requête et nous recherchons des objets similaires (voir figure I-1). Le deuxième cas correspond à la classification d'un objet 3D : nous avons un objet inconnu et nous voulons recueillir des informations en le classant (voir figure I-2).

La recherche d'objets 3D couvre le cas d'une recherche dédiée d'un objet spécifique mais elle peut également être utilisée pour une recherche exploratoire. L'objet recherché peut être un nouveau maillage 3D, un exemple de maillage provenant d'une base de données quelconque ou même un objet aléatoire provenant de la même base de données pour lancer une recherche exploratoire. Il est également possible que la requête ne soit pas un véritable objet 3D mais plutôt une image ou un graphe topologique si le système de comparaison permet cette représentation.

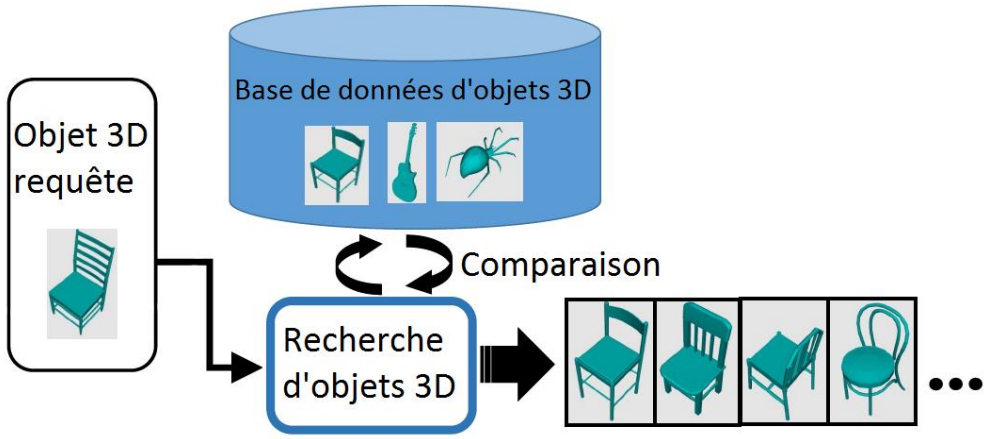


Figure I-1: Recherche d'objets 3D : Une seule requête est donnée. La requête peut être un objet 3D lui-même mais peut aussi avoir une autre forme. La requête est comparée à tous les objets d'une base de données. Le résultat est une liste d'objets triés par similarité. Le premier objet est l'objet le plus similaire.

La classification des objets 3D peut être utilisée pour un seul objet, mais elle peut aussi servir à filtrer ou à regrouper une base de données complète, en classant tous les objets ensemble. Les étiquettes de classe peuvent être exclusives ou se chevaucher, de sorte qu'un objet peut avoir plusieurs étiquettes. Les étiquettes de classe peuvent avoir différents niveaux sémantiques, par exemple un niveau assez large (mammifères, insectes, meubles) ou un niveau plus spécifique (chats à poils courts, chats à poils longs).

La tâche la plus importante dans le processus de recherche et de classification des objets 3D est l'extraction des caractéristiques. En fait, cette étape consiste à transformer l'objet 3D en une représentation réduite et compacte. En général, la forme des objets 3D est caractérisée par un vecteur de caractéristiques qui sert de clé de classification et de recherche dans la base de données. Par conséquent, les techniques de classification et de recherche dépendent fortement de l'approche d'extraction des caractéristiques utilisée. Ces approches peuvent être divisées en approches basées sur une correspondance globale [Gao et al. 2010], et en approches basées sur une correspondance partielles de l'objet 3D [Liu et al. 2015].

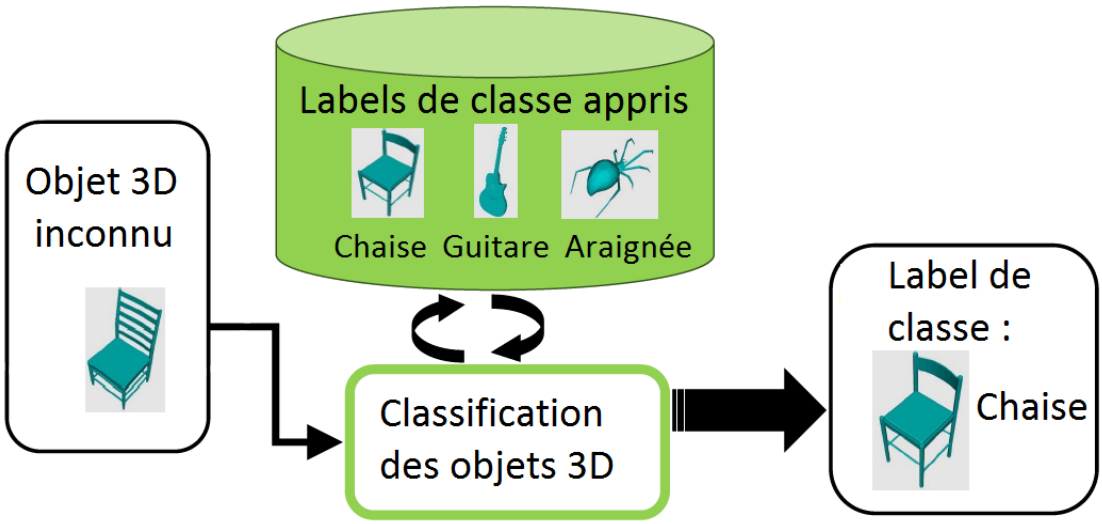


Figure I-2 : Classification des objets 3D : Un objet inconnu est donné. L'objet inconnu est comparé à toutes les étiquettes de classe apprises et il est étiqueté par la classe la plus similaire.

Les premières méthodes de recherche et de classification d'objets 3D appartiennent principalement aux approches basées sur la correspondance globale de la forme d'objets 3D. Cependant, certains objets 3D, comme les modèles artistiques, peuvent être incomplets ou imparfaits ; le processus d'extraction des caractéristiques doit donc traiter les objets 3D en tenant compte de leur similarité partielle. Même si les objets 3D diffèrent visuellement dans l'ensemble, les systèmes de recherche d'objets 3D basés sur la similarité partielle sont prévisibles pour récupérer des modèles partiellement similaires, ce qui n'est pas le cas pour les systèmes qui décrivent les caractéristiques globales de l'objet 3D. Ainsi, les systèmes de recherche d'objets 3D basés sur la similarité partielle sont confrontés à la difficulté de faire correspondre des objets incomplets et imparfaits qui existe encore pour les techniques de recherche et de classification d'objets 3D.

Objectif et structure de la thèse

Le recours aux approches basées sur la correspondance partielle sera primordial dans les systèmes qui devraient faire face au problème des objets incomplets ou imparfaits. Dans la littérature, ces approches sont principalement classées en deux catégories ; celles basées sur la décomposition des objets 3D en parties et celles basées sur les caractéristiques locales. Notre thèse s'inscrit dans la première classification. L'objectif principal de notre thèse est d'introduire de nouvelles méthodes de recherche et de classification d'objets 3D basées sur la similarité partielle, qui peuvent être efficaces à la fois pour des objets 3D complets et aussi incomplets ou imparfaits. Pour ce faire, nous représentons chaque objet 3D par une série d'images de coupe 2D (2D slices) dans certaines directions, de sorte que la correspondance de forme entre les objets 3D soit transformée en mesure de similarité entre leurs images de coupe 2D. Cinque problèmes sont impliqués dans ce processus : la sélection des directions de coupe, la méthode de coupe, le nombre d'images de coupe 2D, la description de ces images 2D, et finalement la mesure de similarité entre leurs descripteurs.

Pour aborder ces problèmes, certaines stratégies et règles sont présentées. Tout d'abord, nous commençons par une étape de normalisation pour nous assurer que les objets 3D similaires seront décomposés de la même manière. Ensuite, nous proposons une méthode de découpe qui peut être utilisée pour obtenir une série d'images de coupe 2D correspondant aux axes déterminés. Par la suite, nous utilisons des descripteurs de forme 2D pour extraire les caractéristiques d'images de coupe 2D. Afin de réduire le stockage et le temps de recherche sans diminuer la performance de nos approches, nous utilisons des algorithmes d'exploration de données, notamment des algorithmes de Clustering et d'extraction des règles d'association afin de choisir parmi les images de coupe 2D extraites celles qui sont les plus représentatives en se basant sur leur descripteurs. Finalement, la similarité entre les descripteurs d'images de coupe représentatives est mesurée en utilisant notre nouvelle métrique basée sur la distance de Hausdorff.

Une autre contribution dans la classification et la recherche par le contenu d'objet 3D est également présentée dans ce travail. En fait, nous proposons une approche qui tire profit de la puissance d'apprentissage machine pour extraire les caractéristiques les plus discriminantes d'images de coupe 2D. Plus précisément, nous utilisons les réseaux de neurones convolutifs 3D (CNN-3D) pour extraire les caractéristiques de haut niveau qui sont utilisées pour concevoir un descripteur de forme profond sur lequel un test de classification et de recherche est effectué.

Le reste de ce manuscrit s'organise comme suit :

Le **premier chapitre** est consacré à la problématique et les concepts associés à la classification et la recherche par le contenu d'objets 3D. Nous commençons par une brève introduction sur les objets 3D, après nous discutons séparément les différentes phases constituant un système de recherche de modèles 3D, notamment : le prétraitement, l'extraction de caractéristiques, la correspondance de similarité. Les critères d'évaluation des approches de recherche et de classification de modèles 3D seront aussi abordés dans ce chapitre.

Dans le **deuxième chapitre**, nous présentons nos trois approches d'indexation et de recherche d'objets 3D basées sur les images de coupe 2D et des algorithmes d'exploration de données, ainsi que les concepts associés. Tout d'abord, nous commençons par une synthèse bibliographique des algorithmes d'extraction de connaissances à partir des données (ECD) et de l'exploration de données, en particulier les algorithmes de partitionnement des données (Clustering) et l'apprentissage des règles d'association. Ensuite, nous introduisons les trois méthodes proposées pour l'indexation et la recherche d'objets 3D. Pour chaque approche proposée, une série de tests expérimentaux est présentée pour démontrer sa performance.

Dans le **troisième chapitre**, nous commençons par une brève présentation de l'apprentissage automatique, et plus particulièrement l'apprentissage supervisé. Après nous fournissons une vue d'ensemble sur les réseaux de neurones, de leurs origines jusqu'aux récentes percées dans le domaine. Nous présentons ensuite les réseaux de neurones convolutifs (CNN). Enfin, nous introduisons notre quatrième approche appelée « 2DSlicesNet », qui profite de la puissance représentative des images de coupe 2D afin d'entraîner un réseau de neurones convolutif 3D (CNN-3D) à la classification et la recherche d'objets 3D.

La dernière partie de ce manuscrit expose une synthèse des travaux réalisés dans cette thèse. Il présente aussi les possibles évolutions futures de nos méthodes ainsi que les perspectives de cette thèse.

Chapitre I : Recherche et classification par le contenu 3D

II.1 Introduction

II.1.1 Problématique de recherche et de classification d'objets 3D basée sur le contenu

Les systèmes de recherche et de classification d'objets 3D basées sur le contenu sont tous deux fondés sur la comparaison d'objets 3D. Dans la recherche d'objets 3D, les objets sont triés par similarité et dans la classification d'objets 3D, la similarité avec une classe est calculée. Le processus de traitement essentiel de ces systèmes peut être décrit approximativement comme suit : les caractéristiques compactes et représentatives, telles que les formes géométriques, les relations spatiales et topologiques, les propriétés statistiques, les textures et les attributs des matériaux, sont d'abord calculées et extraites automatiquement des objets 3D pour construire leurs descripteurs, de sorte que la mesure de similarité peut être calculée en se basant sur ces derniers. Pour les applications de recherche, les valeurs de similarité sont ensuite triées par ordre décroissant, de sorte que les objets ayant les plus grandes valeurs de similarité sont renvoyés en tant que résultats correspondants.

Ici, "basé sur le contenu" signifie que l'utilisateur utilise les caractéristiques visuelles des objets 3D eux-mêmes, plutôt que de se fier aux métadonnées saisies par l'homme, comme les légendes ou les mots-clés. En fait, ces annotations dépendent toujours des connaissances, de la capacité d'expression et de la langue spécifique de l'annotateur. Elles ne sont donc pas fiables. Même la simple représentation linguistique du mappage de formes ou de textures, comme le rond ou le jaune, nécessite des méthodes de formalisation mathématique entièrement différentes, qui ne sont ni intuitives ni uniques ni solides. Par conséquent, les caractéristiques visuelles des modèles 3D doivent être extraites automatiquement ou semi automatiquement et doivent caractériser leur contenu.

II.1.2 Cadre de recherche d'objets 3D basés sur le contenu

Du point de vue conceptuel, un système de recherche d'objets 3D typique, tel qu'illustré à la figure II-1, se compose d'une base de données avec une structure créée hors ligne et un moteur de recherche en ligne. Ce système se compose généralement de quatre éléments principaux :

1) Le module de prétraitement d'objet 3D pour l'enregistrement de la pose, la suppression du bruit, etc ;

2) Le module d'extraction des caractéristiques pour générer à la fois des formes ou des caractéristiques d'apparence 3D de bas niveau et des caractéristiques sémantiques de haut niveau;

3) la phase de correspondance des similarités, c'est-à-dire la procédure de classement de la pertinence en fonction des degrés de similarité calculés ;

4) l'interface de requête, c'est-à-dire une interface utilisateur en ligne pratique conçue pour représenter et traiter les requêtes des utilisateurs.

En général, une procédure de recherche d'objet 3D se déroule en quatre étapes : indexation, interrogation, comparaison et visualisation. À l'exception de la première étape qui est effectuée hors ligne, les trois autres étapes sont effectuées en ligne pour traiter chaque requête utilisateur. En fait, les descripteurs de forme pertinents sont extraits des objets de la base de données pendant la phase hors ligne afin de pouvoir les comparer

efficacement aux requêtes de la phase en ligne. Ces descripteurs de forme fournissent une description pertinente de chaque objet 3D.

Pour effectuer une recherche efficace dans les grands dépôts d'objets 3D en ligne, il faut bien concevoir une structure de données d'indexation et un algorithme de recherche efficace. Le moteur de recherche en ligne calcule le descripteur de la requête, puis quantifie la similarité entre le descripteur de la requête et chaque descripteur de forme dans la base de données sur la base d'une mesure de similarité spécifique. L'ensemble du moteur de recherche de modèles 3D permet à l'utilisateur de rechercher des modèles 3D de manière interactive.

À la différence des systèmes de reconnaissance d'objets 3D conventionnels, qui sont généralement réalisés au coût d'une grande complexité de calcul en établissant des correspondances entre une paire d'objets 3D puis en les comparant, les systèmes de recherche d'objets 3D basés sur le contenu doivent être réalisés "par objet", ce qui signifie que les caractéristiques utilisées pour la correspondance doivent être calculées et stockées indépendamment des objets 3D cibles [Kazhdan et al. 2004]. Cela permet également que le processus d'extraction de caractéristiques soit "hors ligne", car il n'est pas nécessaire d'établir explicitement des correspondances. Ainsi, lors de la phase de recherche "en ligne" proprement dite, la correspondance est effectuée en comparant le descripteur de la requête avec le descripteur de chaque objet 3D dans la base de données. Les caractéristiques de chaque objet 3D de la base de données sont extraites pendant la phase hors ligne pour permettre la comparaison avec les requêtes en ligne par la suite.

Dans le reste de ce chapitre, nous commençons par une brève introduction sur les objets 3D, ensuite nous analysons et discutons plusieurs sujets pour la recherche et la classification d'objets 3D basée sur le contenu, y compris le prétraitement, l'extraction de caractéristiques, la correspondance de similarité et les critères d'évaluation des méthodes de recherche et de classification d'objets 3D.

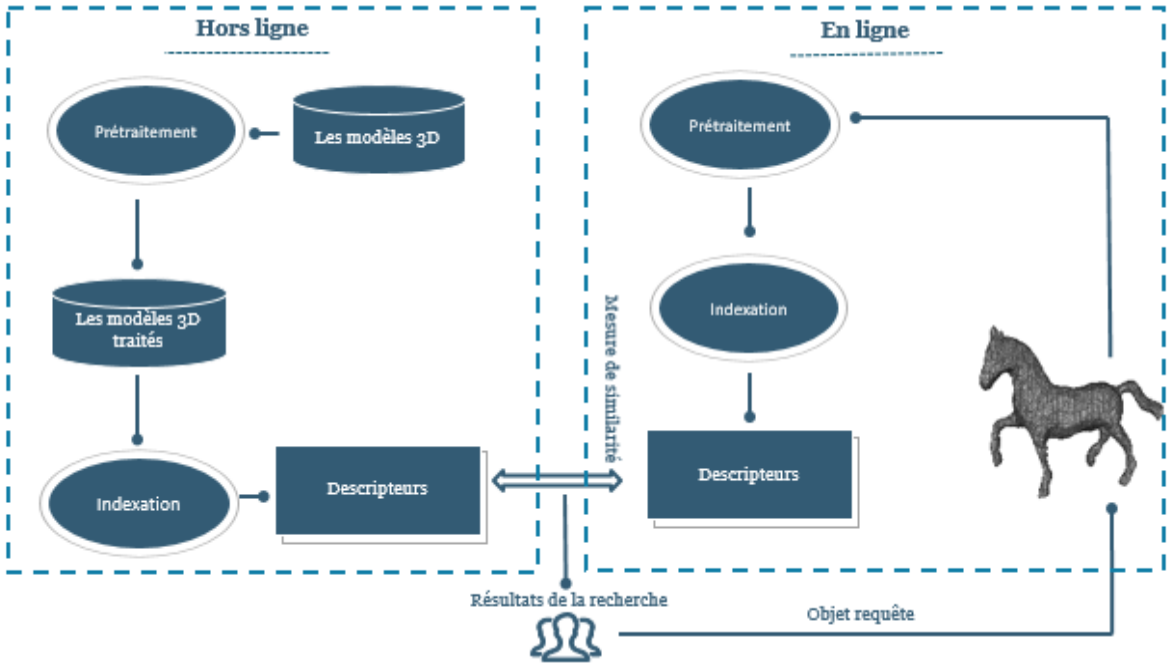


Figure II-1: schéma représentant le processus standard de l'indexation et la recherche par le contenu

II.2 Les objets tridimensionnels

II.2.1 La représentation des objets 3D

Aujourd'hui, les objets 3D sont plus complexes à manipuler que les autres données multimédia, comme les signaux audio ou les images 2D, car il existe de nombreuses représentations différentes de ces objets.

Par exemple, une image 2D a une représentation unique et assez simple : une grille 2D ($n \times n$) composée de n^2 éléments (appelés pixels) contenant chacun une valeur de couleur ou un niveau de gris. Les différents appareils et techniques qui produisent des images numériques (appareils photo numériques, scanners, etc.) fournissent tous la même représentation.

Pour les modèles 3D, il existe différents types de représentation : un objet peut être représenté sur une grille 3D comme une image numérique, ou dans un espace euclidien 3D. Dans ce dernier cas, l'objet peut être exprimé par une seule équation (comme les surfaces algébriques implicites), par un ensemble de facettes représentant sa surface limite ou par un ensemble de surfaces mathématiques.

Les principales difficultés d'un objet 3D sont les suivantes :

- Les différentes sources de données 3D (tomographie, balayage laser) ne produisent pas les mêmes représentations.

- Les différentes applications (conception assistée par ordinateur, médical) ne considèrent pas les mêmes représentations.

- Le passage d'une représentation à une autre est assez complexe et constitue souvent un problème ouvert.

La figure II-2 illustre plusieurs représentations de l'objet 3D Bunny.

A partir d'un objet du monde réel, un scanner laser produit un ensemble de points dans l'espace 3D, chacun étant défini par ses coordonnées x, y et z. La figure II-2-b illustre de points représentant l'objet Bunny. Cette représentation ponctuelle, fournie par les scanners, n'est pas efficace pour calculer des propriétés physiques ou géométriques ou pour les afficher sur un écran. En effet, il n'y a pas de relations de voisinage entre les points 3D, ni d'informations de surface ou volumétriques. C'est pourquoi ces représentations sont souvent converties en maillages polygonaux et surtout en maillages triangulaires (voir figure II-2-c). Avec ce modèle, l'objet est représenté par sa surface limite qui est composée d'un ensemble de faces planes (souvent des triangles). Plus précisément, un maillage polygonal contient un ensemble de points 3D (les sommets) qui sont reliés par des arêtes pour former un ensemble de facettes polygonales.

Les maillages polygonaux peuvent représenter des surfaces ouvertes ou fermées à partir d'une topologie arbitraire, avec une précision qui dépend du nombre de sommets et de facettes. Les algorithmes d'intersection, de détection de collision ou de rendu sont simples et rapides avec ce modèle, car la manipulation des faces planes (et notamment des triangles) est également simple (algèbre linéaire). Cette rapidité est particulièrement utile pour les jeux vidéo. Ces avantages font de ce modèle la représentation la plus répandue pour les objets 3D.

Cependant, les maillages polygonaux présentent certaines limites. Ce modèle est intrinsèquement discret puisque le nombre de sommets et de facettes dépend de la précision attendue, ainsi une grande précision peut conduire à une énorme quantité de données. De plus, la définition de la forme est très locale et il est donc assez difficile d'appliquer une déformation globale ou de créer manuellement une forme facette par facette.

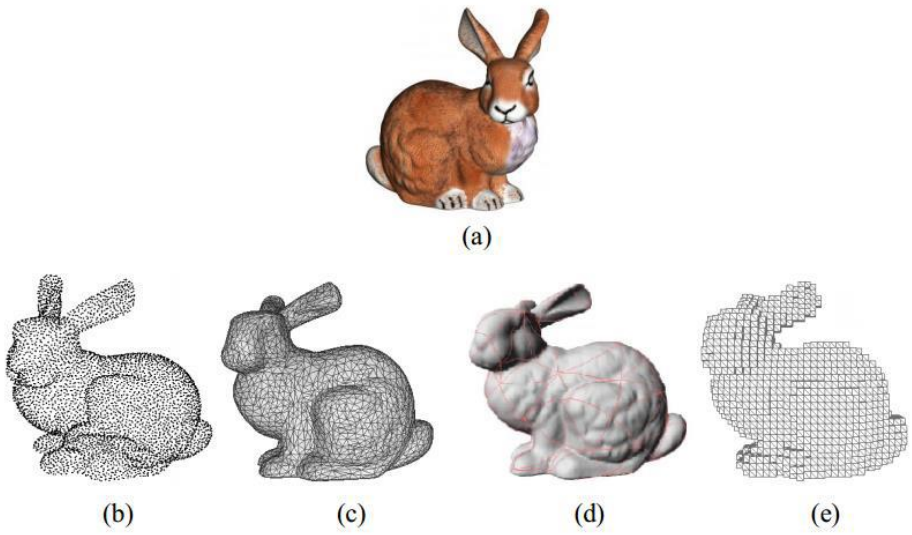


Figure II-2: Différentes représentations de l'objet Bunny (a), nuage de points (b), maillage triangulaire (c), ensemble de surfaces paramétriques(d), ensemble de voxels (e).

Le besoin d'un modèle 3D adapté pour la modélisation et la conception a conduit à l'apparition de surfaces paramétriques. Cette famille de surfaces 3D est particulièrement utilisée pour la conception assistée par ordinateur (CAO), la fabrication (FAO) et l'ingénierie (IAO). Ce modèle permet de définir des surfaces mathématiquement exactes, contrairement aux maillages polygonaux qui ne représentent qu'une approximation. Comme ces surfaces sont définies sur un domaine paramétrique, elles ne peuvent pas représenter une forme avec une topologie arbitraire, d'où un objet 3D souvent représenté par un patch de surfaces paramétriques. Par exemple, Bunny de la figure II-2-d est modélisé avec 153 patchs bicubes B-spline.

Ces surfaces ont de très fortes propriétés de tangence et de continuité de courbure qui les rendent très utiles pour la conception. De plus, elles sont mathématiquement complètes et permettent de modéliser une grande variété de formes. Comme elles ne sont définies que par quelques ensembles de points de contrôle, au lieu d'un ensemble dense de sommets, elles sont également beaucoup plus compactes que les maillages polygonaux. Le principal inconvénient est qu'ils sont plus complexes à manipuler qu'un ensemble de triangles.

Les maillages polygonaux et les surfaces paramétriques sont des représentations de limites puisqu'ils ne modélisent un objet que par sa limite. Plusieurs domaines (en particulier l'imagerie médicale) ont besoin des données intérieures d'un objet 3D et considèrent donc un modèle volumétrique et en particulier la représentation discrète. Ce modèle ne représente pas l'objet 3D dans l'espace euclidien, mais dans une grille 3D similaire à la représentation de l'image 2D. Chaque élément de la grille est un voxel (abréviation de "pixel volumétrique"). L'objet est donc représenté par l'ensemble des voxels constituant son volume. Un voxel peut contenir une valeur booléenne (dans l'objet ou en dehors de l'objet) ou d'autres informations comme les densités locales. Les dispositifs médicaux (RMI par exemple) produisent souvent de telles données volumétriques pour décrire l'intérieur d'un organe. La figure II-2-e illustre l'objet Bunny voxélisé, dans une grille de 50 × 50 × 50.

II.2.2 **Bases de données 3D**

Nous présentons ci-dessous un aperçu des bases de données 3D les plus récentes. Il existe deux grandes catégories de données utilisées par la communauté des chercheurs : les ensembles de données du monde réel et les données synthétiques rendues à partir de

modèles CAO. Il est préférable d'utiliser les données du monde réel ; cependant, les données réelles sont coûteuses à collecter et souffrent généralement de problèmes de bruit et d'occlusion. En revanche, les données synthétiques peuvent produire une énorme quantité de données propres avec des problèmes de modélisation limités. Bien que cela puisse être considéré comme avantageux, cela limite considérablement la capacité de généralisation du modèle appris aux données de test du monde réel. Il est également important de noter que la plupart des bases de données 3D sont plus petites que les grandes bases de données 2D tels qu'ImageNet [Deng et al. 2009]. Cependant, il y a quelques exceptions récentes comme décrit ci-dessous.

ModelNet [Wu et al. 2015] est l'ensemble de données le plus couramment utilisé pour la reconnaissance et la classification des objets 3D. Il contient environ 130 000 objets 3D annotés sur 662 catégories distinctes. Cet ensemble de données a été collecté à l'aide de moteurs de recherche en ligne en interrogeant chacune des catégories. Ensuite, les données ont été annotées manuellement. ModelNet fournit la géométrie 3D de la forme sans aucune information sur la texture. Il comporte deux sous-ensembles : ModelNet10 et ModelNet40. Ces sous-ensembles sont utilisés dans la plupart des travaux récemment publiés. La base de données "Princeton Shape Benchmark" [Shilane et al. 2004], qui a été créée distinctement pour comparer les approches de recherche d'objets 3D avec une mise à disposition de 1.814 objets 3D, au format OFF ou VRML.

SUNCG [Song et al. 2017] contient environ 400K de modèles de pièces complètes. Cet ensemble de données est synthétique, cependant, chacun de ces modèles a été validé pour être réaliste et il a été traité pour être annoté avec des modèles d'objets étiquetés. Cet ensemble de données est important pour apprendre la relation "scène-objet" et pour affiner les données du monde réel pour les tâches de compréhension de la scène. SceneNet [Handa et al. 2016] est également un ensemble de données RVB-D qui utilise des pièces intérieures synthétiques. Cet ensemble de données contient environ 5 millions de scènes qui sont échantillonnées au hasard à partir d'une distribution pour refléter le monde réel. Cependant, toutes les scènes générées ne sont pas réalistes et, en pratique, certaines d'entre elles sont très irréalistes. Néanmoins, cet ensemble de données peut être utilisé pour la mise au point et l'entrainement préalable. En revanche, ScanNet [Dai et al. 2017] est un ensemble de données très riche pour les scènes du monde réel. Il s'agit d'un ensemble de données annotées qui est étiqueté avec une certaine segmentation sémantique, l'orientation de la caméra et les informations 3D qui sont recueillies à partir de séquences vidéo 3D de scènes réelles d'intérieur. Il comprend 2,5 millions de vues, ce qui permet de s'entraîner directement sans apprentissage préalable sur d'autres ensembles de données, comme c'est le cas pour les différents ensembles de données.

En outre, des ensembles de données pour les maillages 3D sont disponibles pour la communauté de la vision par ordinateur en 3D. La plupart des ensembles de données de maillages 3D concernent des objets 3D, des modèles corporels ou des données de visage. La base de données TOSCA [Bronstein et al. 2008] fournit des maillages synthétiques 3D à haute résolution pour des formes non rigides. Il contient un total de 80 objets dans différentes poses. Les objets de la même catégorie ont le même nombre de sommets et la même connectivité de triangulation. TOSCA fournit des déformations artistiques sur les maillages pour simuler les déformations du monde réel des scans réels. SHREC [Bronstein et al. 2010] ajoute une variété de bruit artificiel et de déformations artistiques sur les scans TOSCA. Cependant, le bruit artificiel et les déformations ne sont pas réalistes et ne peuvent pas être généralisés à de nouvelles données invisibles du monde réel, ce qui est nécessaire pour trouver des solutions pratiques. L'ensemble de données FAUST [Bogoet al. 2014], cependant, fournit 300 scans réels de 10 personnes dans différentes poses. L'ensemble de données 3DBodyTex [108] est récemment proposé avec 200 scans corporels réels en 3D avec une texture haute résolution. 3DBodyTex est un ensemble de données enregistré avec les points de repère disponibles pour les modèles 3D du corps humain.

La série des ensembles de données BU est très populaire pour les visages 3D sous diverses expressions. BU-3DFE [Yin et al. 2006] est un ensemble de données statiques qui comprend 100 sujets (56 femmes et 44 hommes) d'âges et de races différents. Chaque sujet a, en plus du visage neutre, six expressions (bonheur, tristesse, colère, dégoût, peur et surprise) d'intensité différente. Il y a 25 maillages pour chaque sujet au total, ce qui donne un ensemble de données de 2500 expressions faciales en 3D. Un autre ensemble de données très populaire est BU-4DFE [Yin et al. 2008], qui est un ensemble de données d'expressions faciales dynamiques qui compte 101 sujets au total (58 femmes et 43 hommes). Comme pour le BU-3DFE, chaque sujet a six expressions. D'autres ensembles de données de visages 3D étaient disponibles, comme le BP4D-Spontanous [Zhang et al. 2014] et le BP4D+ [Zhang et al. 2016].

II.3 Prétraitement d'objets 3D

En général, les objets 3D ont des échelles, des orientations et des positions arbitraires dans l'espace 3D. Dans de nombreuses situations, on doit normaliser la taille et l'orientation d'un objet 3D avant l'extraction des caractéristiques afin de le représenter dans un système de coordonnées canonique. L'objectif de l'étape de normalisation est de garantir que la même représentation des caractéristiques peut être correctement extraite du même objet 3D avec une échelle, une position et une orientation différentes. Cela nous permet d'effectuer des tâches de recherche "par objet", sans autre alignement des objets 3D les uns par rapport aux autres. Actuellement, il existe deux schémas pour réaliser une telle normalisation "par objet" :

(1) La technique de normalisation pour trouver un cadre de coordonnées canonique basé sur des méthodes similaires à l'analyse en composantes principales (ACP), également appelée estimation de la pose ou enregistrement de la pose.

(2) La technique basée sur l'invariance pour définir et extraire des descripteurs qui possèdent les caractéristiques d'invariance inhérentes, de manière à ne pas changer sous l'effet de transformations rigides. Les approches basées sur l'invariance ont reçu un poids croissant dans les recherches récentes en raison de leur robustesse et de leur simplicité.

Cependant, les caractéristiques d'invariance ne sont pas toujours complètes et ne permettent pas de représenter un objet 3D sur tous les côtés. De plus, le calcul de ces descripteurs de caractéristiques est nécessairement effectué sur un cadre de coordonnées unitaires. Ainsi, pour garantir le pouvoir descriptif et la robustesse des représentations des caractéristiques, la normalisation canonique des coordonnées, comme l'alignement et la mise à l'échelle, est également une étape nécessaire avant l'extraction des caractéristiques invariantes.

Outre le processus de normalisation, il est également inévitable d'effectuer d'autres étapes de prétraitement [Funkhouser et al. 2003] [Min et al. 2003] sur les objets 3D avant l'extraction des caractéristiques. Ces étapes comprennent la transformation entre les différentes représentations de données 3D (par exemple, pour transformer les maillages de polygones en grilles de voxels), la partition des unités d'objet et le regroupement des sommets, etc. Dans certains systèmes d'extraction d'objets 3D, au stade du prétraitement, un ensemble d'objets de référence est sélectionnées dans la base de données sur la base d'une analyse des clusters, et les distances entre les modèles de la base de données et les modèles de référence sont calculées et stockées. Dans les sections suivantes, nous aimerions introduire deux étapes typiques de prétraitement, c'est-à-dire la normalisation de la pose et la segmentation du maillage.

II.3.1 Normalisation de la pose

En l'absence de connaissances préalables, les objets 3D ont des échelles, des orientations et des positions arbitraires dans l'espace 3D. Par conséquent, une étape de normalisation est nécessaire pour obtenir des descripteurs invariants, ce qui correspond à placer l'objet 3D dans un système de coordonnées canonique. Les attributs suivants

fournissent des données utiles pour normaliser les modèles 3D en fonction des différences de translation, d'échelle et d'orientation :

(1) Centre de masse : les coordonnées moyennes (x, y, z) de tous les points sur les surfaces de tous les polygones. Ces valeurs peuvent être utilisées pour normaliser les modèles en fonction de l'invariance de la translation.

(2) Échelle : la distance moyenne entre tous les points de la surface de tous les polygones et le centre de masse. Cette valeur peut être utilisée pour normaliser les modèles pour l'invariance d'échelle.

(3) Axes principaux : les vecteurs propres et les valeurs propres associées de la matrice de covariance obtenue en intégrant les polynômes quadratiques $v_i \cdot v_j$, avec $v_i \in \{x, y, z\}$, sur tous les points des surfaces de tous les polygones. Ces axes peuvent être utilisés pour normaliser les modèles pour l'invariance de rotation.

Nous présentons ici deux méthodes typiques de normalisation des poses. L'une est la méthode basée sur l'analyse en composantes principales (ACP), qui rend le vecteur d'éléments de forme résultant indépendant des translations et des rotations autant que possible. L'autre consiste à trouver la seule boîte englobante d'un objet 3D.

II.3.1.1 Normalisation de la position basée sur l'ACP

L'analyse en composantes principales implique une procédure mathématique qui transforme un certain nombre de variables éventuellement corrélées en un plus petit nombre de variables non corrélées appelées composantes principales. La première composante principale explique autant que possible la variabilité des données, et chaque composante suivante explique autant que possible la variabilité restante. Selon le domaine d'application, elle est également appelée transformée discrète de Karhunen-Loève, transformée de Hotelling ou décomposition orthogonale correcte. L'ACP a été inventée en 1901 par Karl Pearson [Pearson et al. 1901]. Aujourd'hui, elle est surtout utilisée comme outil d'analyse exploratoire des données et pour la réalisation de modèles prédictifs. L'ACP implique le calcul de la décomposition en valeur propre d'une matrice de covariance de données ou la décomposition en valeur singulière d'une matrice de données, généralement après avoir centré en moyenne les données pour chaque attribut. Les résultats d'une ACP sont généralement examinés en termes de scores et de charges des composantes. L'ACP est le type le plus simple de véritable analyse multivariée basée sur les vecteurs propres. En général, son fonctionnement peut être considéré comme révélant la structure interne des données d'une manière qui explique au mieux la variance des données. Si un ensemble de données multivariées est visualisé comme un ensemble de coordonnées dans un espace de données à haute dimension (1 axe par variable), l'ACP fournit à l'utilisateur une image à plus basse dimension, c'est-à-dire une "ombre" de cet objet lorsqu'il est vu de son point de vue (dans un certain sens) le plus informatif. L'ACP est étroitement liée à l'analyse factorielle et, de fait, certains progiciels statistiques associent délibérément les deux techniques. En fait, la véritable analyse factorielle fait des hypothèses différentes sur la structure sous-jacente et résout les vecteurs propres à partir d'une matrice légèrement différente.

Dans la normalisation des objets 3D, l'objectif de l'ACP est de modifier les axes du système de coordonnées pour qu'ils coïncident avec les directions des trois plus grands écarts de la distribution des points (c'est-à-dire des sommets). Les étapes détaillées peuvent être décrites comme suit :

Étape 1 : Translation. Tout d'abord, le centre de masse d'objet doit être déplacé vers l'origine des coordonnées comme suit :

$$I_1 = I - c = \{u | u = v - c, v \in I\} \quad \text{(II-1)}$$

Où I est le cadre de coordonnées d'objet 3D original, I_1 est le nouveau cadre de coordonnées après translation et c est le centroïde d'objet [Vraníc et al. 2000].

Étape 2 : Rotation basée sur l'ACP. Ensuite, l'ACP est utilisée pour déterminer les axes de coordonnées canoniques d'un objet 3D, en calculant les vecteurs propres correspondants et la matrice diagonale résultante R des valeurs propres, ordonnée de manière décroissante par leurs valeurs. La transformation de la rotation est représentée par :

$$I_2 = R \cdot I_1 = \{x | x = R \cdot u, u \in I_1\} \quad \text{(II-2)}$$

Où I_1 est le cadre de coordonnées de l'objet 3D avant la rotation et I_2 est le nouveau cadre de coordonnées après la rotation, qui sont identiques aux directions ayant les trois plus grandes variances de la distribution des points.

La transformation ACP générale dans la recherche d'objets 3D est définie sur l'ensemble donné de points représentatifs d'un objet 3D, tels que les sommets, les centroïdes de chaque surface, ou même des emplacements choisis au hasard sur chaque surface à l'aide de techniques statistiques, par exemple l'approche de Monte Carlo [Ohbuchi et al. 2002]. En considérant les différentes tailles de triangles ou de mailles d'un objet 3D, il est possible d'adapter certains facteurs de pondération appropriés, proportionnels à leur surface, afin de rendre la transformation plus robuste et d'améliorer la fiabilité et la véracité de la représentation des objets [Vraníc et al. 2000] [Paquet et al. 2000] [Heczko et al. 2001]. Cependant, la transformation ACP basée sur des points peut entraîner un résultat de normalisation inexact qui affectera sérieusement la précision de l'extraction si les sommets choisis n'ont pas une distribution uniforme sur la surface. Par conséquent, une amélioration plus approfondie, appelée ACPC (ACP continue), qui effectue une transformation ACP basée sur l'ensemble du maillage polygonal 3D, est proposée dans [Vraníc et al. 2001]. L'ACPC généralise la transformation ACP en utilisant les sommes des intégrales sur les surfaces au lieu des sommes sur les sommets sélectifs. Supposons que la taille totale de toutes les surfaces d'un objet 3D soit représentée comme :

$$S = \sum_{i=1}^{N_f} S_i = \int_I dv \quad \text{(II-3)}$$

Où $v \in$ I est le point sur la surface, N_f est le nombre de surfaces sur l'objet 3D et I est l'ensemble de points d'objet 3D comme suit :

$$\mathrm{I} = \bigcup_{\mathrm{i}=1}^{\mathrm{N_v}} \mathrm{v_i} \quad \text{(II-4)}$$

Où N_v est le nombre de points, v_i est le $i^{ième}$ point. De même, l'ensemble de triangles T peut être désigné comme :

$$T = \bigcup_{i=1}^{N_t} \Delta_i \,, \Delta_i = (a_i, b_i, c_i) \quad \text{(II-5)}$$

Où N_t est le nombre de triangles, Δ_i signifie le $i^{ième}$ triangle. La matrice de covariance R est alors définie comme :

$$\mathrm{R} = \frac{1}{\mathrm{S}} \cdot \int_{\mathrm{I}} \mathrm{v} \cdot \mathrm{v}^{\mathrm{T}} \mathrm{dv} \quad \text{(II-6)}$$

Après avoir trouvé la matrice de covariance, nous calculons alors la matrice des vecteurs propres qui diagonalise la matrice de covariance. Cette étape implique généralement l'utilisation d'un algorithme informatique pour le calcul des vecteurs propres et

des valeurs propres. Les valeurs propres et les vecteurs propres sont ensuite ordonnés et appariés. La $i^{ième}$ valeur propre correspond au $i^{ième}$ vecteur propre. Nous trions ensuite les colonnes de la matrice des vecteurs propres et de la matrice des valeurs propres dans l'ordre décroissant des valeurs propres. Enfin, nous sélectionnons un sous-ensemble de vecteurs propres comme vecteurs de base.

L'algorithme de l'ACP est assez simple et efficace. Cependant, il peut attribuer à tort les axes principaux et produire des résultats de normalisation inexacts, en particulier lorsque les valeurs propres sont égales ou proches les unes des autres, ce qui arrive généralement à différents objets de la même catégorie [Funkhouser et al. 2003] [Kazhdan et al. 2004]. Un exemple typique d'ACP [Vranıc et al. 2000] est illustré à la figure II-3, où les axes du système de coordonnées d'origine sont désignés par x, y, z, tandis que les composantes principales sont marquées par p_1, p_2 et p_3.

Étape 3 : Réflexion. Une matrice F à retournement diagonal est conçue pour réaliser l'invariance de réflexion, ce qui garantit qu'un objet et sa réflexion auront le même descripteur de caractéristique.

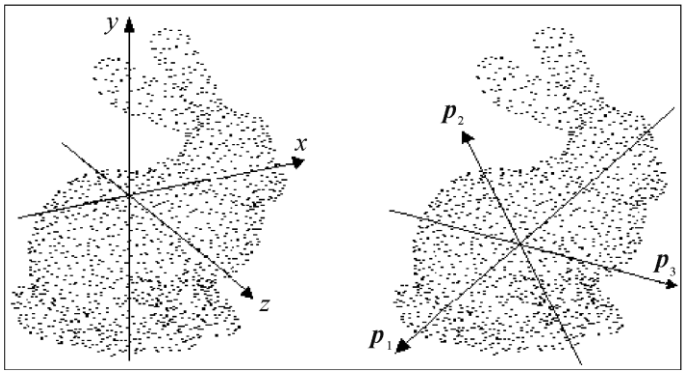


Figure II-3 : Analyse en composantes principales[Vranıc et al. 2000]

Étape 4 : Mise à l'échelle. Enfin, l'objet 3D doit également être mis à l'échelle en multipliant un coefficient d'échelle approprié s par une certaine unité de taille pour garantir l'invariance d'échelle.

La définition de la matrice de renversement et du coefficient d'échelle se trouve dans [Vranıc et al. 2001]. Par conséquent, l'ensemble du processus de normalisation peut être décrit comme suit [Vranıc et al. 2000] :

$$\tau(I) = s^{-1} \cdot F \cdot R \cdot (I - c) \qquad \text{(II-7)}$$

II.3.1.2 Trouver la boîte englobante du modèle 3D

En infographie et en géométrie computationnelle, un volume limite pour un ensemble d'objets est un volume fermé qui contient complètement l'union des objets de l'ensemble. Les volumes limites sont utilisés pour améliorer l'efficacité des opérations géométriques en utilisant des volumes simples pour contenir des objets plus complexes. Normalement, les volumes simples ont des moyens plus simples de tester le chevauchement. Un volume limitant pour un ensemble d'objets est également un volume limitant pour l'objet unique constitué de leur union, et inversement. Il est donc possible de limiter la description au cas d'un seul objet, qui est supposé être non vide et délimité.

Une boîte englobante est un cuboïde, ou en 2D un rectangle, contenant l'objet. Dans la simulation dynamique, les boîtes englobantes sont préférées à d'autres formes de volumes englobants comme les sphères ou les cylindres englobants pour les objets de forme grossièrement cubique lorsque le test d'intersection doit être assez précis. L'avantage est

évident, par exemple, pour les objets qui reposent les uns sur les autres, comme une voiture reposant sur le sol : une sphère délimitante montrerait que la voiture pourrait croiser le sol, ce qui devrait alors être rejeté par un test plus coûteux du modèle réel de la voiture. Une boîte de délimitation indique immédiatement que la voiture ne coupe pas le sol, ce qui permet d'économiser le test le plus coûteux. Dans de nombreuses applications, la boîte englobante est alignée avec les axes du système de coordonnées, et elle est alors connue sous le nom de boîte englobante alignée sur les axes (Axis Aligned Bounding Box, AABB). Pour distinguer le cas général d'un AABB, une boîte englobante arbitraire est parfois appelée boîte englobante orientée (Oriented Bounding Box, OBB). Les AABB sont beaucoup plus simples à tester pour l'intersection que les OBB, mais ont l'inconvénient que lorsque l'objet est tourné, ils ne peuvent pas être simplement tournés avec lui, mais doivent être recalculés.

Trouver la seule boîte englobante de l'objet 3D est une autre méthode populaire de normalisation de la pose [Gottschalk et al. 1999] [Tomas et al. 2002] [Pu et al. 2004]. De nombreuses méthodes de construction d'une boîte englobante ont été étudiées, telles que la AABB et les axes principaux inertiels (Inertial Principal Axes, IPA) [Gottschalk et al. 1999] [Tomas et al. 2002]. La boîte englobante la plus simple est l'AABB, mais elle n'est pas unique car les directions latérales de la boîte sont déterminées par les axes du système de coordonnées universel. Gottschalk a présenté la méthode IPA pour calculer une boîte englobante bien ajustée, basée sur une méthode statistique. En calculant les vecteurs propres d'une matrice de covariance 3×3, on peut obtenir les vecteurs de direction pour une boîte bien ajustée. Dans la figure II-4, les boîtes englobantes indiquées en (a), (b) et (c) sont des exemples obtenus par cette méthode. La distribution normale maximale (Maximum Normal Distribution, MND), qui est également une méthode puissante pour calculer la seule boîte englobante d'un objet 3D, a été fournie par [Pu et al. 2004]. La normalisation des objets 3D basée sur la distribution normale maximale établit l'orientation des coordonnées d'une boîte englobante selon la distribution normale, et obtient ainsi les coordonnées intrinsèques d'un objet 3D. L'idée principale de la méthode de distribution normale maximale est d'obtenir trois ortho-axes qui coïncident mieux avec le mécanisme de perception visuelle de l'homme. Bien que la méthode IPA puisse obtenir trois ortho-axes de façon unique, ils ne sont toujours pas idéaux pour les trois directions qui ne sont pas en accord avec notre mécanisme de perception visuelle. Par conséquent, Pu et al. ont proposé d'adopter la distribution normale maximale comme l'un des axes principaux. Cette méthode peut être introduite comme suit :

Premièrement, nous devons calculer la direction normale N_d pour chaque triangle Δpqr et la normaliser. Il s'agit du produit croisé de deux bords quelconques, comme :

$$N_d = \frac{pq \times qr}{||pq \times qr||} \tag{II-8}$$

Ensuite, la surface de chaque triangle Δ_i est calculée et les surfaces de tous les triangles ayant des normales identiques ou opposées sont additionnées. Ici, Pu et al. ont pensé que les normales qui sont situées dans la même direction appartiennent à une distribution similaire.

L'étape suivante consiste à déterminer les trois axes principaux. Parmi toutes les distributions normales, la normale avec la surface maximale est sélectionnée comme premier axe principal b_u. Pour obtenir l'axe principal suivant b_v, nous pouvons effectuer une recherche à partir des distributions normales restantes et trouver la normale qui remplit deux conditions : (1) avec l'aire maximale ; (2) orthogonale à la première normale. Naturellement, le troisième axe peut être obtenu en faisant un produit croisé entre b_u et b_v :

$$b_w = b_u \times b_v \tag{II-9}$$

Pour trouver le centre et la demi-longueur de la boîte englobante, Pu et al. ont projeté les points du maillage du polygone sur le vecteur directionnel et ont trouvé le minimum et le maximum le long de chaque direction. Enfin, il faut décider de la direction positive pour chaque axe principal. À cette fin, Pu et al. ont proposé une règle : le côté le plus éloigné du centroïde est la direction positive. Sur la figure II-4, les boîtes indiquées en (d), (e) et (f) sont obtenues par la méthode de la distribution normale maximale, et elles sont beaucoup plus précises que les figures II-4 (a), (b) et (c).

Pour les objets ayant une distribution normale évidente, tels que les objets CAO, la méthode MND est plus performante que la méthode IPA. Toutefois, pour les modèles sans distribution normale évidente, comme le montre la figure II-5, la première méthode échouera parce que la distribution normale a une propriété aléatoire dans ce cas. La figure II-5 montre que la méthode IPA décrit bien la distribution de masse des objets 3D et permet de déterminer les axes de symétrie en fonction des distributions de masse.

Par conséquent, pour surmonter cette limitation et exploiter pleinement les qualités des deux méthodes, Pu et al. ont proposé une règle pour combiner les deux méthodes : sélectionner la boîte englobante de plus petit volume comme boîte finale. Sa validité a été prouvée par un grand nombre de modèles dans leur bibliothèque d'objets 3D comprenant plus de 2 700 modèles.

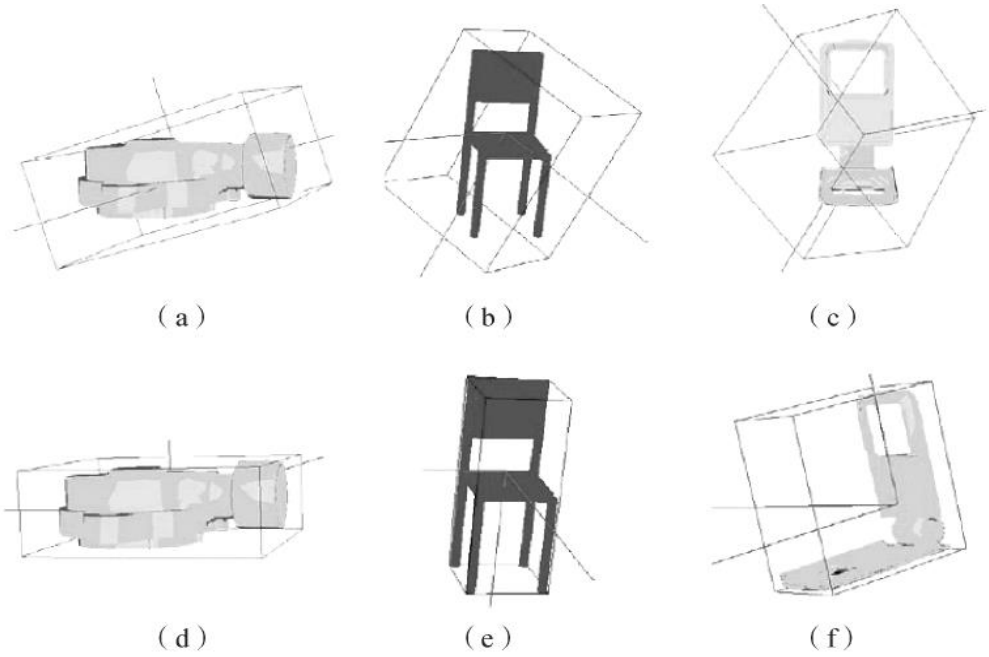


Figure II-4: Exemples de boîtes englobantes [Gottschalk et al. 1999].Les cases limites indiquées en (a), (b) et (c) sont obtenues par la méthode IPA, tandis que les cases indiquées en (d), (e) et (f) sont obtenues par la méthode MND.

II.3.2 **Segmentation des maillages**

La partition des unités de l'objet est également nécessaire si nous extrayons des caractéristiques de diverses parties des modèles 3D. Il s'agit d'un problème de segmentation. La segmentation du maillage est devenue un problème important et difficile en infographie, avec des applications dans des domaines aussi divers que la modélisation, la métamorphose, la compression, la simplification, la recherche d'objets 3D, la détection de collisions, le mappage de texture et l'extraction de squelettes.

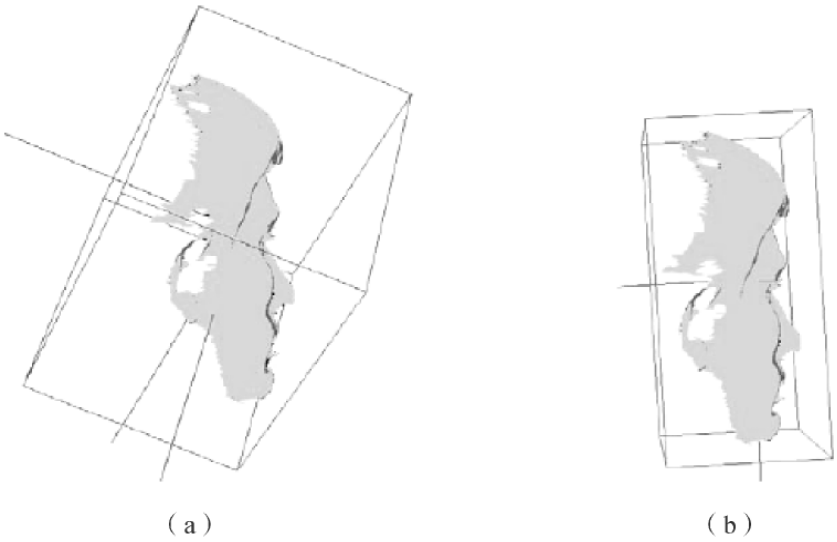


Figure II-5: Un exemple pour la boîte englobante d'un objet maillé, dans lequel la méthode MND échoue [Gottschalk et al. 1999]. (a) La boîte englobante obtenue par la méthode MND ; (b) La boîte englobante obtenue par la méthode IPA.

La segmentation des maillages, et plus généralement des formes, peut être interprétée soit dans un sens purement géométrique, soit dans un sens plus sémantique. Dans le premier cas, le maillage est segmenté en un certain nombre d'éléments qui sont uniformes en ce qui concerne certaines propriétés (par exemple, la courbure ou la distance par rapport à un plan d'ajustement), tandis que dans le second cas, la segmentation vise à identifier les parties qui correspondent aux caractéristiques pertinentes de la forme. Les méthodes qui peuvent être regroupées sous la première catégorie ont été présentées comme une étape de prétraitement pour la reconnaissance de caractéristiques significatives. Les approches sémantiques de la segmentation des formes ont récemment suscité un grand intérêt dans la communauté des chercheurs, car elles peuvent prendre en charge des schémas de paramétrage ou de remaillage, la métamorphose, la recherche d'objets 3D, l'extraction de squelettes ainsi que la modélisation par paradigme de composition qui est basée sur des décompositions naturelles de formes.

Il est cependant assez difficile d'évaluer la performance des différentes méthodes en ce qui concerne leur capacité à segmenter les formes en parties significatives. Cela est dû au fait que la majorité des méthodes utilisées en infographie ne sont pas conçues pour détecter des caractéristiques spécifiques dans un contexte précis. De plus, les classes de formes traitées dans le contexte générique de l'infographie sont une catégorie très variable : des humains virtuels aux artefacts scannés, des formes libres très complexes aux objets très lisses et sans caractéristiques. En outre, il n'est pas facile de définir formellement les caractéristiques significatives de formes complexes dans un contexte non technique et la comparaison des différentes méthodes est donc principalement qualitative. Enfin, les méthodes de segmentation des formes sont généralement conçues pour résoudre un problème d'application spécifique, par exemple la recherche ou le paramétrage, et il n'est donc pas facile de comparer l'efficacité des différentes méthodes pour la segmentation des formes elle-même.

Voici quelques méthodes typiques de segmentation en maillage, et la figure II-6 montre quelques effets de segmentation par ces méthodes.

a) Décomposition du maillage par regroupement flou et coupes [Katz et al. 2003]. L'idée clé de cet algorithme est de trouver d'abord les composants significatifs en utilisant un algorithme de regroupement, tout en gardant les limites entre les

composants floues. Ensuite, l'algorithme se concentre sur les petites zones floues et trouve les limites exactes qui suivent les caractéristiques de l'objet.

b) Segmentation du maillage en utilisant l'extraction de points caractéristiques et de noyaux [Katz et al. 2005]. Cette approche est basée sur trois idées clés. Premièrement, l'échelle multidimensionnelle est utilisée pour transformer les sommets du maillage en une représentation insensible à la pose. Deuxièmement, les points caractéristiques importants sont extraits en utilisant la représentation d'échelle multidimensionnelle. Troisièmement, on trouve l'élément central du maillage. Le noyau, ainsi que les points caractéristiques, fournissent suffisamment d'informations pour une segmentation significative.

c) Tailor : analyse de maillage multi-échelle à l'aide de bulles d'air [Mortara et al. 2004a]. Cette méthode fournit une segmentation d'une forme en groupes de sommets qui ont un comportement uniforme du point de vue de la morphologie de la forme, analysée à différentes échelles. L'idée principale est d'analyser la forme en utilisant un ensemble de sphères de rayon croissant, placées aux sommets du maillage. Le type et la longueur de la courbe d'intersection sphère-maille sont de bons descripteurs de la forme et peuvent être utilisés pour fournir une analyse multi-échelle de la surface.

d) Plumber : segmentation des mailles en parties tubulaires [Mortara et al. 2004-b]. Basée sur l'analyse de la forme du Tailor, la méthode Plumber décompose la forme en éléments tubulaires et en composants corporels et extrait, simultanément, l'axe squelettique des éléments. Les caractéristiques tubulaires capturent les parties allongées de la forme, les saillies ou les puits, et sont bien adaptées aux objets articulés.

e) Segmentation hiérarchique du maillage basée sur des primitives ajustables [Attene et al. 2006]. Basé sur un algorithme de regroupement hiérarchique des visages, le maillage est segmenté en patchs qui s'adaptent le mieux à un ensemble prédéfini de primitives. Dans le prototype actuel, ces primitives sont des plans, des sphères et des cylindres. Au départ, chaque triangle représente un seul groupe. À chaque itération, toutes les paires de groupes adjacents sont prises en compte, et celui qui peut être mieux approché avec l'une des primitives forme un nouveau groupe unique. L'erreur d'approximation est évaluée en utilisant la même métrique pour toutes les primitives, de sorte qu'il est logique de choisir la primitive la plus appropriée pour approximer l'ensemble des triangles d'un groupe.

II.4 Extraction des caractéristiques

II.4.1 Concepts et définitions de base

Nous présentons quelques concepts et définitions de base, tels que les caractéristiques, l'extraction de caractéristiques, le descripteur de forme 3D et les exigences pour l'extraction de caractéristiques 3D.

II.4.1.1 Caractéristiques

Dans la reconnaissance des formes, les caractéristiques sont les propriétés heuristiques individuelles mesurables des phénomènes observés. Dans les objets 3D, la caractéristique est un élément qui peut être utilisé pour identifier l'objectif. Nous pouvons encore la réduire à quelque chose qui peut être facilement compris et traité par les ordinateurs, c'est-à-dire la caractéristique d'une forme géométrique régulière. Le choix de caractéristiques discriminantes et indépendantes est essentiel pour que tout algorithme de reconnaissance de formes réussisse à classifier. Les caractéristiques sont généralement numériques, mais les caractéristiques structurelles telles que les chaînes de caractères et les graphes sont utilisées dans la reconnaissance syntaxique des formes. Bien que les différents domaines de la reconnaissance de formes aient évidemment des caractéristiques différentes, une fois les caractéristiques décidées, elles sont classées par un ensemble d'algorithmes

beaucoup plus restreint. Il s'agit notamment de la classification du plus proche voisin en plusieurs dimensions, des réseaux de neurones ou des techniques statistiques telles que les approches bayésiennes.

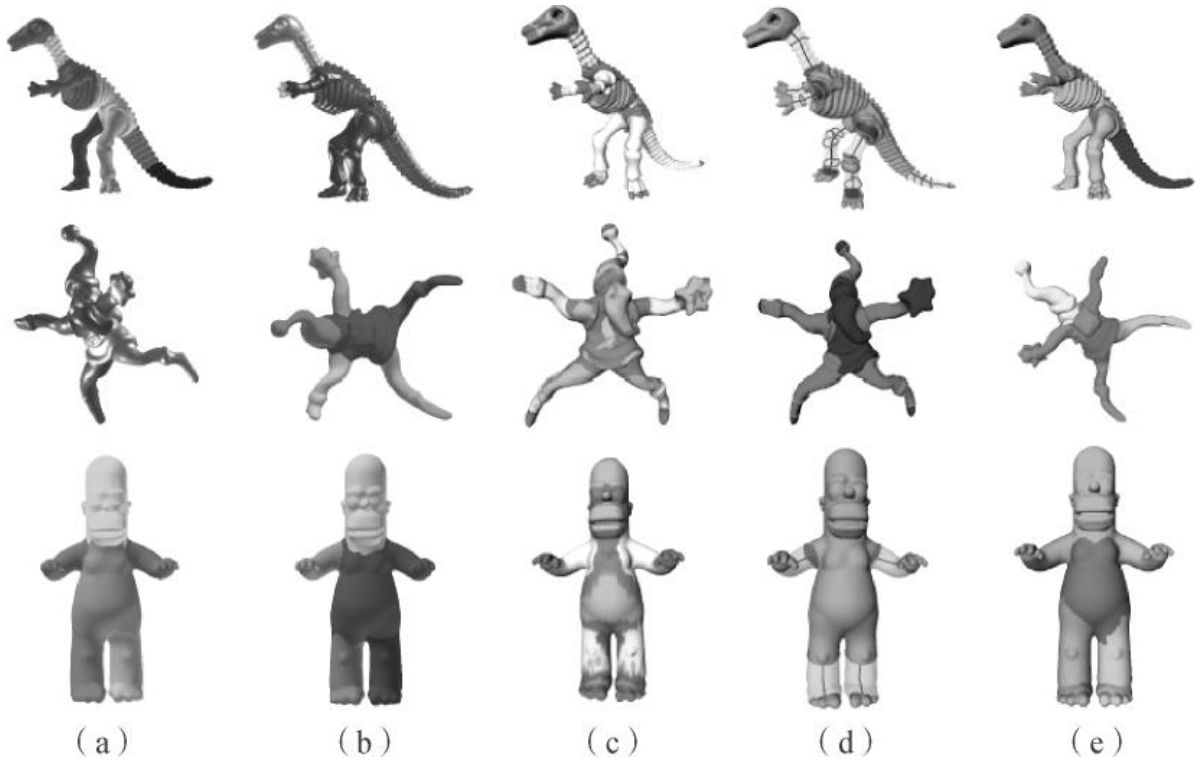


Figure II-6: Segmentations d'objets divers par diverses méthodes [Attene et al. 2006]. (a) Basé sur le regroupement flou et les coupes, (b) Basé sur l'extraction des points caractéristiques et des noyaux, (c) Tailor, (d) Plumber, (e) Basé sur les primitives d'ajustement.

En reconnaissance de caractères, les caractéristiques peuvent inclure les profils horizontaux et verticaux, le nombre de trous internes, la détection des traits et bien d'autres encore. Dans la reconnaissance vocale, les caractéristiques de reconnaissance des phonèmes peuvent inclure les rapports de bruit, la longueur des sons, la puissance relative, les correspondances de filtres et bien d'autres. Dans les algorithmes de détection du spam, les caractéristiques peuvent inclure la présence ou l'absence de certains en-têtes de courrier électronique, leur bonne forme, la langue dans laquelle le courrier électronique semble être, l'exactitude grammaticale du texte, l'analyse des fréquences markoviennes et bien d'autres encore. Dans tous ces cas et bien d'autres, l'extraction de caractéristiques mesurables par un ordinateur est un art et, à l'exception de certaines techniques de réseaux neuronaux et de génétique qui intuitionnent automatiquement les "caractéristiques", la sélection manuelle de bonnes caractéristiques constitue la base de presque tous les algorithmes de classification.

II.4.1.2 Extraction des caractéristiques

Dans la reconnaissance des formes et le traitement multimédia, l'extraction de caractéristiques est une forme particulière de réduction de la dimension. Lorsque les données d'entrée d'un algorithme sont trop volumineuses pour être traitées et qu'on soupçonne qu'elles sont notoirement redondantes (beaucoup de données, mais peu d'informations), les données d'entrée sont alors transformées en un ensemble de caractéristiques à représentation réduite (également appelé vecteur de caractéristiques). La transformation des données d'entrée en un ensemble de caractéristiques est appelée extraction de caractéristiques. Si les caractéristiques extraites sont choisies avec soin, on

s'attend à ce que l'ensemble de caractéristiques extraie les informations pertinentes des données d'entrée afin d'effectuer la tâche souhaitée en utilisant cette représentation réduite au lieu de l'entrée pleine grandeur.

L'extraction de caractéristiques consiste à simplifier la quantité de ressources nécessaires pour décrire avec précision un grand ensemble de données. Lors de l'analyse de données complexes, l'un des principaux problèmes provient du nombre de variables impliquées. Une analyse avec un grand nombre de variables nécessite généralement une grande quantité de mémoire et une grande puissance de calcul ou un algorithme de classification qui surcharge l'échantillon d'entraînement et se généralise mal aux nouveaux échantillons. L'extraction de caractéristiques est un terme général qui désigne les méthodes permettant de construire des combinaisons de variables pour contourner ces problèmes, tout en décrivant les données avec une précision suffisante.

Le meilleur résultat est obtenu lorsqu'un expert construit un ensemble de caractéristiques dépendantes de l'application. Néanmoins, en l'absence de telles connaissances d'expert, des techniques générales de réduction de la dimensionnalité peuvent s'avérer utiles. Celles-ci comprennent l'analyse en composantes principales, l'encastrement semi-défini, la réduction de la dimensionnalité multifactorielle, la réduction de la dimensionnalité non linéaire, l'isomap, l'ACP du noyau, l'analyse sémantique latente, les moindres carrés partiels et l'analyse en composantes indépendantes.

II.4.1.3 Descripteur de forme 3D

Comme nous le savons, la forme est facile à percevoir directement par l'homme. De nombreuses méthodes d'extraction de caractéristiques sont basées sur la forme des objets 3D, qui utilisent souvent les caractéristiques géométriques de surface pour décrire les objets. La forme du modèle est fondamentale et constitue la caractéristique de niveau le plus bas. Il existe donc de nombreuses méthodes qui permettent d'extraire des caractéristiques grâce à l'attribut de la forme de la surface des objets. La distance ou la distance géodésique sur la surface, la surface des pièces, le volume et la direction normale sont toutes des caractéristiques de la forme.

Les représentations utilisées pour la correspondance des formes sont souvent appelées descripteurs de formes 3D et diffèrent généralement de manière substantielle de celles destinées au rendu et à la visualisation d'objets 3D. Les descripteurs de forme visent à coder les propriétés géométriques et topologiques d'un objet de manière discriminante et compacte. La diversité des descripteurs de forme va des moments 3D aux distributions de formes, des harmoniques sphériques à l'échantillonnage par rayons et des nuages de points aux transformations de volume voxélisées.

II.4.1.4 Exigences pour l'extraction de caractéristiques d'objets 3D

La forme d'un objet 3D est décrite par le vecteur de caractéristique qui sert de clé de recherche dans la base de données. Si une méthode d'extraction de caractéristiques inappropriée avait été utilisée, l'ensemble du système de recherche ne serait pas utilisable. Par conséquent, le texte suivant est consacré aux propriétés qu'une méthode idéale d'extraction de caractéristiques devrait avoir [Hlavaty et al. 2003] :

(1) Indépendance des représentations d'objets 3D. Nous devons d'abord réaliser que les objets 3D peuvent être sauvegardés dans de nombreuses représentations telles que les maillages polyédriques, les données volumétriques, les équations paramétriques ou implicites. La méthode d'extraction des caractéristiques devrait accepter ce fait et être indépendante des représentations des données.

(2) Invariance sous transformations. Les valeurs des descripteurs calculées doivent être invariantes sous un ensemble de transformations dépendant de l'application. En général, il s'agit de transformations de similitude, mais certaines applications, comme la recherche d'objets articulés, peuvent également exiger une invariance sous certaines déformations.

C'est peut-être l'exigence la plus importante, car les objets 3D sont généralement enregistrés dans des poses et des échelles différentes.

(3) Insensibilité au bruit. L'objet 3D peut être obtenu soit à partir d'un programme graphique 3D, soit à partir d'un périphérique d'entrée 3D. La deuxième méthode est plus susceptible de comporter certaines erreurs. Ainsi, la méthode d'extraction des caractéristiques devrait également être insensible au bruit.

(4) Pouvoir descriptif. La mesure de similarité basée sur le descripteur doit fournir un ordre de similarité proche de la notion de ressemblance basée sur l'application. Les caractéristiques des différents objets doivent pouvoir être distinguées.

(5) Concision et facilité d'indexation. La base de données peut contenir des milliers d'objets et la flexibilité du système serait également l'une des principales exigences. Le descripteur devrait être compact afin de minimiser les besoins de stockage et d'accélérer la recherche en réduisant la dimensionnalité du problème. Plus important encore, il devrait fournir un moyen d'indexer et donc de structurer la base de données afin d'accélérer encore le processus de recherche.

La méthode d'extraction de caractéristiques qui aurait toutes les exigences mentionnées ci-dessus n'existe probablement pas. Pour autant, il existe des méthodes qui tentent de trouver un compromis entre des propriétés idéales.

II.4.2 **Classification des algorithmes d'extraction de caractéristiques d'objets 3D**

En fonction des différents aspects du contenu qu'ils représentent, les caractéristiques des objets 3D peuvent être classées en deux grandes catégories [Yang et al. 2007] : (1) les caractéristiques de forme, à savoir les caractéristiques géométriques et topologiques et (2) les caractéristiques d'apparence, qui représentent certaines caractéristiques cognitives importantes telles que les couleurs des matériaux, les coefficients de réflexion et la cartographie des textures.

En fonction des différents formats de données de représentation des caractéristiques, les auteurs de [Akgül et al. 2007] ont souligné qu'il existe deux paradigmes pour les opérations et la conception des mesures de similarité des bases de données d'objets 3D, à savoir l'approche vectorielle des caractéristiques et l'approche vectorielle non caractéristique. Le paradigme des vecteurs de caractéristiques vise à obtenir des valeurs numériques de certains descripteurs de forme et à mesurer les distances entre ces vecteurs. D'autre part, un exemple typique de l'approche non basée sur les caractéristiques est de décrire l'objet comme un graphe et d'utiliser ensuite des mesures de similarité de graphe. Du même point de vue, Akgül et al. ont souligné qu'il existe deux principaux paradigmes de la description de formes 3D, à savoir l'approche basée sur les graphes et l'approche basée sur les vecteurs. Les représentations basées sur les graphes, d'une part, sont plus élaborées et complexes, plus difficiles à obtenir, mais représentent les propriétés de la forme de manière plus fidèle et intuitive. Les graphes de choc [Siddiqi et al. 1998], les graphes Reeb multirésolution [Tung et al. 2005] et les graphes squelettiques [Sundar et al. 2003] sont des méthodes qui entrent dans cette catégorie. Cependant, elles ne se généralisent pas facilement et ne sont donc pas très pratiques à utiliser dans le cadre d'un apprentissage non supervisé, par exemple pour rechercher des classes de formes naturelles dans une base de données. Les représentations vectorielles, en revanche, sont plus faciles à calculer. Bien qu'elles ne soient pas nécessairement favorables à des visualisations topologiques plausibles, elles peuvent naturellement être utilisées dans des tâches de classification supervisées et non supervisées. Les représentations vectorielles typiques sont les images gaussiennes étendues [Kang et al. 1993], les histogrammes de cordes et d'angles [Paquet et al. 1997], les histogrammes de formes 3D [Ankerst et al. 1999a], les harmoniques sphériques [Funkhouser et al. 2003] et les distributions de formes [Osada et al. 2002].

Il est nécessaire de rechercher les objets 3D de manière invariable en ce qui concerne la translation, la rotation, la mise à l'échelle et la réflexion. Par conséquent, dans

de nombreux cas, il peut être nécessaire de recourir à des processus supplémentaires de normalisation de l'alignement (enregistrement de la pose) pour aligner les objets 3D sur leur cadre de coordonnées canoniques, ou à des mappages ou transformations plus complexes pour extraire les représentations invariantes des caractéristiques d'un objet 3D avant une correspondance de similarité. De ce point de vue, nous pouvons classer les caractéristiques 3D en deux catégories : les caractéristiques rotation-variante et les caractéristiques rotation-invariante.

Selon les différents types d'objets 3D, les schémas d'extraction de caractéristiques 3D peuvent également être classés en extraction de caractéristiques basée sur un maillage et en extraction de caractéristiques basée sur des points [Daniels et al. 2007]. De nombreuses techniques ont permis d'identifier les bords caractéristiques sur des modèles polygonaux. Cependant, pour les objets basés sur des points, l'hypothèse sous-jacente de connectivité et de normales associées aux sommets du maillage n'est pas disponible. Afin d'extraire les lignes de caractéristiques des nuages de points en utilisant ces techniques, une méthode de construction de la connectivité (reconstruction de surface) doit être appliquée dans une étape de prétraitement. La construction de la connectivité est non triviale, coûteuse en termes de calcul et, de plus, le succès de l'extraction de caractéristiques dépend de la capacité de la procédure de maillage polygonal à construire avec précision les arêtes tranchantes. Pour les méthodes d'extraction d'éléments basées sur des points, l'extraction d'éléments à partir de modèles basés sur des points n'est pas simple en l'absence de connectivité et d'informations normales. Les auteurs de [Pauly et al. 2003] ont utilisé l'analyse de covariance des voisinages locaux en fonction de la distance pour repérer des points caractéristiques potentiels. En faisant varier le rayon des voisinages, ils ont développé un schéma multirésolution capable de traiter des données d'entrée bruitées. Les auteurs de [Gumhold et al. 2001] ont construit un graphique de Riemann sur les voisinages locaux et ont utilisé l'analyse de covariance pour calculer les poids qui signalent les points comme étant des plis, des limites ou des coins potentiels. Les deux techniques, [Pauly et al. 2003] et [Gumhold et al. 2001], relient les points signalés en utilisant un arbre à portée minimale et ajustent les courbes pour obtenir des bords nets approximatifs. Les auteurs de [Demarsin et al. 2006] ont calculé des normales de points en utilisant l'analyse en composantes principales et ont segmenté les points en groupes basés sur la variation normale dans les voisinages locaux. Un arbre à portée minimale est construit entre les points limites des groupes assortis, qui a été utilisé pour construire les courbes de caractéristiques finales. Ces techniques sont capables d'extraire des caractéristiques sur des nuages de points en reliant des points existants. Toutefois, leur précision dépend de la qualité d'échantillonnage de l'objet d'entrée.

Récemment l'apprentissage approfondi (Deep Learning, DL) a remarquablement contribué dans le domaine de la vision par ordinateur en obtenant des résultats de pointe sur plusieurs tâches de vision par ordinateur 2D [Farabet et al. 2013][He et al. 2016][Krizhevsky et al. 2012][Long et al. 2015][Noh et al. 2015][Saito et al. 2016][Sermanet et al. 2013], DL a commencé à gagner en popularité dans le domaine 3D en essayant d'utiliser les riches données 3D disponibles tout en considérant leurs propriétés difficiles. Cependant, l'extension des modèles DL aux données 3D n'est pas simple en raison de la nature géométrique complexe des objets 3D et des grandes variations structurelles qui résultent des différentes représentations 3D.

Dans ce chapitre, selon la technique, nous classons les schémas d'extraction de caractéristiques 3D en six catégories : les algorithmes d'extraction de caractéristiques basés sur les données statistiques, sur l'analyse géométrique globale, sur l'analyse du signal, sur la topologie, sur l'image 2D et sur l'apprentissage approfondi.

II.4.3 **Extraction de caractéristiques statistiques**

À l'heure actuelle, le paramétrage des objets 3D est une tâche très complexe. En outre, comme les surfaces 3D peuvent avoir une topologie arbitraire, certaines méthodes largement utilisées (par exemple, les méthodes basées sur la transformation de Fourier) dans

le traitement des images ne sont pas directement applicables aux objets 3D. Ainsi, il nous est difficile d'acquérir des objets 3D ayant une signification explicite de géométrie ou de formes. Du point de vue des statistiques, les chercheurs montrent une préférence pour la caractéristique statistique avec une grande distinguabilité. Actuellement, les travaux de recherche dans ce domaine adoptent principalement les caractéristiques statistiques suivantes : la relation géométrique entre les sommets (distances, angles, directions normales), la distribution de la courbure des sommets, les moments avec différents ordres de sommets et les coefficients de caractéristiques de diverses transformations, etc.

L'extraction de caractéristiques basée sur des données statistiques permet d'approcher des points d'échantillonnage à la surface d'objets 3D et d'en extraire des caractéristiques. Ces caractéristiques sont généralement organisées sous la forme d'histogrammes ou de distributions représentant la fréquence d'occurrence. La propriété statistique la plus utilisée est celle des "moments", comme les moments de Hu [Hu 1962]. Il existe également d'autres types de propriétés statistiques exprimées sous la forme de différents histogrammes discrets de statistiques géométriques [Ashbrook et al. 1995]. La représentation de la forme est simplifiée en tant que problème de distribution de probabilité par l'utilisation d'histogrammes et évite le processus de normalisation de l'objet.

Par rapport à d'autres méthodes, la plupart des méthodes d'extraction de caractéristiques statistiques sont non seulement rapides et faciles à mettre en œuvre, mais elles présentent également certaines propriétés souhaitées, telles que la robustesse et l'invariance. Dans de nombreux cas, elles sont également robustes contre le bruit, ou les petites fissures et les trous qui existent dans un objet 3D. Malheureusement, comme inconvénient inhérent à la représentation par histogramme, elles ne permettent qu'une discrimination limitée entre les objets : elles ne préservent ni ne construisent d'informations spatiales. Ainsi, ils ne sont souvent pas assez discriminants pour faire de petites différences entre des formes 3D dissemblables, et ne distinguent généralement pas les différentes formes ayant le même histogramme. Dans cette section, nous présentons principalement plusieurs descripteurs de caractéristiques typiques basés sur les moments et sur les histogrammes pour les objets 3D.

II.4.3.1 Moments 3D de la surface

Supposons qu'un objet soit donné en VRML, c'est-à-dire qu'il s'agisse d'un objet 3D représenté par un ensemble de sommets et un ensemble de faces polygonales encastrées en 3D. Les caractéristiques choisies par [Elad et al. 2001] pour représenter les objets sont les moments calculés pour les surfaces des objets, en supposant que le modèle 3D est un modèle creux délimité par ses surfaces. Les moments 3D des surfaces peuvent être calculés comme suit :

$$m_{pqr} = \int_{\partial M} x^p y^q z^r dx dy dz \qquad \text{(II-10)}$$

Où M est le modèle 3D, ∂M est la surface de M, et m_{pqr} est le $(p,q,r)^{ième}$ moment 3D. Pour un modèle 3D, l'ensemble des moments m_{pqr} est unique, de sorte qu'il constitue une description complète de M, et une description partielle de l'objet peut également être obtenue en utilisant un sous-ensemble de ces moments [Duda et al. 1973].

L'essentiel de l'algorithme d'Elad et al. réside dans le calcul d'un sous-ensemble des $(\mathrm{p},\mathrm{q},\mathrm{r})^{\mathrm{ièmes}}$ moments de chaque objet, qui sont utilisés comme ensemble de caractéristiques. Il est donc nécessaire d'effectuer une étape de prétraitement au cours de laquelle les caractéristiques sont calculées pour chaque objet de la base de données. Une façon pratique d'évaluer la totalité des moments définis consiste à calculer analytiquement ces moments pour chaque facette de l'objet, puis à les additionner sur toutes les facettes. Ils utilisent une approche alternative, donnant une approximation des moments. L'algorithme

dessine une séquence de points (x, y, z) répartis uniformément sur la surface de l'objet. Le nombre de points tirés de chacune des facettes de l'objet est proportionnel à sa surface relative. Si on désigne la liste des points pour un objet donné par $\{x_i, y_i, z_i\}, i = 1, 2, \ldots, N$, alors le $(p, q, r)^{ième}$ moment est approximé par :

$$\hat{m}_{pqr} = \frac{1}{N}\sum_{i=1}^{N} x_i^p y_i^q z_i^r \qquad \text{(II-11)}$$

La mesure de similarité doit être invariable en fonction de la position spatiale, de l'échelle et de la rotation des différents objets. Il est donc nécessaire de normaliser les vecteurs de caractéristiques de tous les objets. Les premiers moments m_{100}, m_{010} et m_{001} représentent le centre de masse de l'objet. Ainsi, la normalisation commence par estimer les premiers moments pour chaque objet représenté comme un ensemble de points d'échantillonnage de surface, et les soustrait de chacun de ces points :

$$\forall i = 1, 2, \ldots, N, [x_i, y_i, z_i]^T \leftarrow [x_i - \hat{m}_{100}, y_i - \hat{m}_{010}, z_i - \hat{m}_{001}]^T \qquad \text{(II-12)}$$

Cela revient à positionner tous les objets de manière à ce que leur centre de masse soit en coordonnées $(0,0,0)$, supprimant ainsi toute dépendance à la translation, ou position spatiale. Cela revient également à mettre à 0 chacun des $\hat{m}_{100}, \hat{m}_{010}$ et $\hat{m}_{001}$ pour tous les objets, et donc à les rendre inutiles pour les calculs ultérieurs.

Les seconds moments m_{200}, m_{020}, m_{002}, m_{110}, m_{011} et m_{101} - représentent la rotation et l'échelle de l'objet de la manière suivante. Les seconds moments, calculés pour l'objet recentré à $(0, 0, 0)$, peuvent être ordonnés dans une matrice :

$$Z = \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix} \qquad \text{(II-13)}$$

La décomposition des valeurs singulières (DVS) est ensuite effectuée sur cette matrice, ce qui permet d'obtenir le résultat suivant :

$$U\Delta\Delta^T = DVS(Z) \qquad \text{(II-14)}$$

Où la matrice unitaire U représente la rotation et la matrice diagonale Δ représente l'échelle dans chaque axe, ordonnée en taille décroissante.

La normalisation se poursuit avec une deuxième étape qui consiste à estimer les seconds moments pour chaque objet, en les calculant à partir des ensembles de données de points de surface mis à jour, en utilisant l'équation (III-11) dans $\hat{Z}$. Après avoir effectué la décomposition DVS de la deuxième matrice des moments $\hat{Z}$, on multiplie chaque point par U pour faire pivoter l'objet jusqu'à une position canonique. On divise également chaque point par $\Delta(1,1)$ pour redimensionner l'objet de manière à ce que sa plus grande échelle soit de 1. En résumé, chaque point est remplacé par :

$$[x_i, y_i, z_i]^T \leftarrow \frac{1}{\Delta(1,1)} \cdot U \cdot [x_i, y_i, z_i]^T \qquad \text{(II-15)}$$

Enfin, l'algorithme doit également déterminer l'orientation de chaque objet, par rapport à chaque axe. Pour ce faire, on compte le nombre de points de chaque côté du centre du corps. Afin de normaliser pour que tous les objets aient la même orientation, on retourne chaque objet pour qu'il soit "plus lourd" du côté positif. En comptant le nombre de points et en les retournant en fonction de celui-ci, on force en fait le centre médian à se trouver sur un côté prédéterminé par rapport au centre de masse. Après avoir appliqué toutes les étapes de

normalisation à chaque objet, les moments sont calculés une fois de plus, dans l'ordre prédéfini. Évidemment, le processus de normalisation fixe $\hat{m}_{100}$, $\hat{m}_{010}$, $\hat{m}_{001}$ et $\hat{m}_{200}$ à 0, 0, 0 et 1, respectivement, pour chaque objet. Ils ne sont donc plus utiles en tant que caractéristiques des objets.

II.4.3.2 Moments Zernike 3D

Le principal inconvénient de la méthode des moments 3D décrite précédemment est qu'il faut acquérir un cadre de coordonnées à l'échelle unitaire des objets 3D avant le processus de calcul des caractéristiques. Pour résoudre ce problème, de nouvelles approches d'extraction statistique des caractéristiques sans enregistrement de pose ont été proposées. La caractéristique de forme basée sur les moments de Zernike 3D [Canterakis 1999] en est un exemple. Novotni et al [Novotni et al. 2003] ont démontré que les moments de Zernike 3D sont calculés comme une projection de la fonction définissant l'objet 3D sur un ensemble de fonctions orthonormales dans une sphère unitaire, qui ont une représentation simple mais une bonne performance d'extraction. Ils ont également présenté les invariants 3D de Zernike comme le descripteur de forme 3D. Les étapes nécessaires pour calculer les moments de Zernike 3D et les descripteurs peuvent être exprimées comme suit :

(1) Normalisation. Calculer le centre de gravité de l'objet, le transformer à l'origine, et mettre l'objet à l'échelle pour qu'il soit représenté dans la sphère unitaire.

(2) Calcul du moment géométrique. Calculer tous les moments géométriques

$$m_{pqr} = \int_{|x^2+y^2+z^2|\leq 1} f(x,y,z)\, x^p y^q z^r dxdydz \qquad \text{(II-16)}$$

Pour chaque combinaison d'indices, tels que p, q, r ≥ 0 et p + q + r ≤ N. Notez que le calcul des moments géométriques est d'une importance capitale en ce qui concerne l'efficacité globale du calcul et la précision numérique. Une approche typique du calcul des moments géométriques d'un objet représenté par une grille de voxels 3D est la suivante : 1) Fixer un système de coordonnées avec son origine à un coin de la grille et des axes alignés avec les axes de la grille. Ensuite, échantillonner tous les monomères d'ordre jusqu'à N aux positions des points de la grille. 2) Calculer les moments géométriques selon Eq. (III-16) mais en intégrant sur toute la grille de voxels. 3) Transformer les moments géométriques en fonction de la transformation de normalisation de l'objet. Ceci peut être facilement réalisé, puisque la mise à l'échelle peut être réalisée en mettant à l'échelle les moments, et les moments de l'objet translaté peuvent être représentés en termes de combinaison linéaire des moments originaux d'ordre non supérieur. Les deux premières étapes introduisent des problèmes numériques. Premièrement, l'échantillonnage aux points de grille implique que nous traitons le monomère comme une fonction ayant une valeur constante dans un voxel, qui est déterminée par la valeur du monomère, par exemple, au centre du voxel. Pour les fonctions qui changent rapidement, comme les monomiales d'ordre élevé, il en résulte une imprécision. Deuxièmement, pour une grille de 643 par exemple, la précision du nombre à virgule flottante de double précision est déjà dépassée de l'ordre de 9. Selon l'expérience, des moments jusqu'à l'ordre de 20 sont nécessaires pour fournir un bon descripteur. La première question peut être traitée en calculant les moments géométriques en termes de monomères intégrés sur les voxels. Comme pour les ordres élevés, les descripteurs 3D de Zernike semblent écarter les valeurs des voxels proches de l'origine, l'objet est normalisé avant le calcul des moments, ce qui permet d'obtenir une précision numérique considérablement meilleure et de fournir une solution au second problème. Pour la procédure détaillée, les lecteurs peuvent se référer à [Novotni et al. 2003].

(3) Calcul du moment Zernike 3D. Les invariants de Zernike 3D peuvent être extraits sur la base de ces moments géométriques calculés. Les moments de Zernike peuvent être écrits sous une forme compacte comme une combinaison linéaire de monomères d'ordre jusqu'à n comme suit :

$$\Omega_{nl}^{m} = \frac{3}{4\pi} \sum_{p+q+r \leq n} \bar{\chi}_{nlm}^{pqr} \cdot m_{pqr} \qquad \text{(II-17)}$$

Où $\bar{\chi}_{nlm}^{pqr}$ est le monomère intermédiaire qui peut être trouvé dans [Novotni et al. 2003] pour plus de détails. Notez que la sommation doit être effectuée uniquement pour les coefficients non nuls $\bar{\chi}_{nlm}^{pqr}$. Notez également que pour $m \leq 0$, Ω_{nl}^{m} peut être calculé en utilisant la relation de symétrie $\Omega_{nl}^{-m} = (-1)^{m}\overline{\Omega}_{nl}^{m}$.

(4) Génération du descripteur Zernike 3D. Calculer les descripteurs Zernike 3D invariants en rotation comme normes des vecteurs Ω_{nl} comme suit :

$$F_{nl} = ||\Omega_{nl}|| \qquad \text{(II-18)}$$

Ici, Ω_{nl} est un vecteur à $(2l+1)$-dimensions composé de $2l+1$ moments Ω_{nl}, Ω_{nl}^{l-1}, Ω_{nl}^{l-2},…, Ω_{nl}^{-l}.

Les invariants 3D de Zernike ont été rapportés [Novotni et al. 2003] pour renforcer la robustesse contre les déformations topologiques et géométriques.

II.4.3.3 Histogrammes de formes 3D

La définition d'une fonction de distance appropriée est cruciale pour l'efficacité de tout classificateur du plus proche voisin. Une approche commune pour les modèles de similarité est basée sur le paradigme des vecteurs de caractéristiques. Une transformée de caractéristiques fait correspondre un objet complexe à un vecteur de caractéristiques dans un espace multidimensionnel.

II.4.3.3.1 Histogramme des formes 3D

La similarité de deux objets est alors définie comme le voisinage de leurs vecteurs de caractéristiques dans l'espace des caractéristiques. Les auteurs de [Ankerst et al. 1999b] ont introduit les histogrammes de formes 3D comme vecteurs de caractéristiques intuitifs. En général, les histogrammes sont basés sur un partitionnement de l'espace dans lequel les objets résident, c'est-à-dire une décomposition complète et disjointe en cellules qui correspondent aux cases des histogrammes. L'espace peut être géométrique (2D, 3D), thématique (par exemple, propriétés physiques ou chimiques) ou temporel (modélisation du comportement des objets). Ils ont proposé trois techniques pour décomposer l'espace : un modèle de coquille, un modèle de secteur et un modèle de toile d'araignée comme combinaison des deux premiers, comme le montre la figure II-7. Dans l'étape de prétraitement, un solide 3D est déplacé à l'origine. Ainsi, les objets sont alignés sur le centre de masse du solide.

(1) Modèle de coquille

L'objet 3D est décomposé en coquilles concentriques autour du point central. Cette représentation est particulièrement indépendante d'une rotation des objets, c'est-à-dire que toute rotation d'un objet autour du point central du modèle donne lieu au même histogramme. Les rayons des coquilles sont déterminés à partir des extensions des objets dans la base de données. La coquille la plus extérieure est laissée libre afin de couvrir les objets qui dépassent la taille du plus grand objet connu.

(2) Modèle de secteur

L'objet 3D est décomposé en secteurs qui émergent du point central du modèle. Cette approche est étroitement liée à la méthode de codage des sections en 2D.

Cependant, la définition et le calcul des histogrammes de secteurs 3D sont plus sophistiqués, et ils définissent les secteurs comme suit : Distribuer le nombre de points désirés uniformément sur la surface d'une sphère. Pour cela, on utilise les sommets des

polyèdres réguliers et leurs raffinements récursifs. Une fois les points sont distribués, le diagramme de Voronoï des points définit immédiatement une décomposition appropriée de l'espace. Comme les points sont régulièrement répartis sur la sphère, les cellules de Voronoï se rencontrent au point central de l'objet. Pour le calcul d'histogrammes de formes sectorielles, on n'a pas besoin de matérialiser le diagramme de Voronoï complexe, mais simplement d'appliquer une recherche du plus proche voisin dans l'objet 3D puisque le nombre typique de secteurs n'est pas très grand.

(3) Modèle combiné

Le modèle combiné représente des informations plus détaillées que les modèles de coquilles pures et les modèles de secteurs purs. Une simple combinaison de deux décompositions 3D à grain fin permet d'obtenir une haute dimensionnalité. Cependant, comme la résolution de la décomposition spatiale est un paramètre dans tous les cas, le nombre de dimensions peut facilement être adapté à l'application particulière.

Dans la figure II-8, Ankerst et al. [Ankerst et al. 1999a] ont illustré divers histogrammes de forme pour la protéine d'exemple, 1SER-B, qui est représentée à gauche de la figure. Au milieu, les différentes décompositions spatiales sont indiquées de manière schématique et, à droite, les histogrammes de forme correspondants sont représentés. L'histogramme du haut est purement basé sur les bacs à coquilles, et l'histogramme du bas est défini par 122 bacs à secteurs. Les histogrammes du milieu suivent le modèle combiné, et sont définis par 20 bacs coquille et 6 bacs secteur, et par 6 bacs coquille et 20 bacs secteur, respectivement. Dans cet exemple, tous les différents histogrammes ont approximativement la même dimension, soit environ 120. Notez que les histogrammes ne sont pas construits à partir d'éléments de volume, mais à partir de points de surface uniformément répartis, pris sur les surfaces moléculaires.

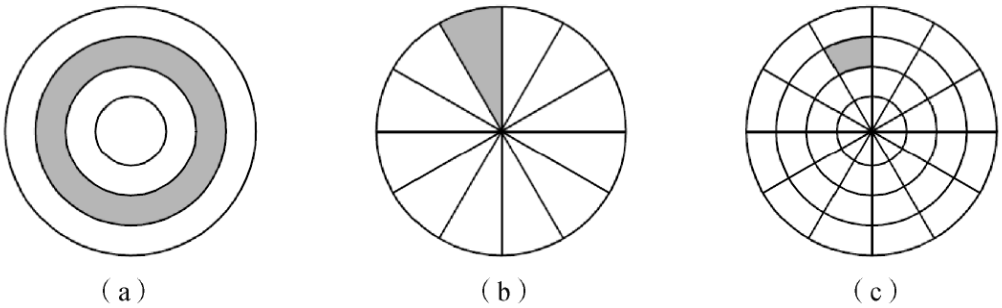


Figure II-7: Les coquilles et les secteurs comme décompositions spatiales de base pour les histogrammes de forme. (a) 4 bacs de coquilles ; (b) 12 bacs de secteurs ; (c) 48 bacs combinés. Dans chacun des exemples en 2D, un seul bac est marqué.

II.4.3.3.2 Histogramme de l'angle de pliage

Besl [Besl 1929] a construit des histogrammes 3D sur les angles de pliage pour tous les bords dans un maillage triangulaire 3D pour correspondre aux formes 3D. La figure II-9 montre les histogrammes des angles de pliage (Crease Angle Histograms, CAH) et les dessins de lignes cachées pour huit formes simples : un bloc, un cylindre, une sphère, un bloc avec canal, une superquadrille "en forme de savon", deux blocs collés ensemble, une superquadrille "double corne" et une superquadrille "en forme de vérin". En travaillant de bas en haut, on voit que le bloc d'histogrammes des angles de pliage est constitué de deux pics simples : un pic à 90 degrés pour les 12 arêtes et un pic à zéro pour les triangles adjacents à l'intérieur d'une face. Les plis du cylindre auront des angles nuls ou petits et positifs ainsi qu'un pic à 90 degrés. Les trois pics idéaux, un pour la planéité, un pour la courbure convexe et un pour les angles de 90 degrés, sont la signature du cylindre. L'histogramme d'un cône idéal sera très, très similaire, sauf que le pic à 90 degrés devrait être deux fois plus petit.

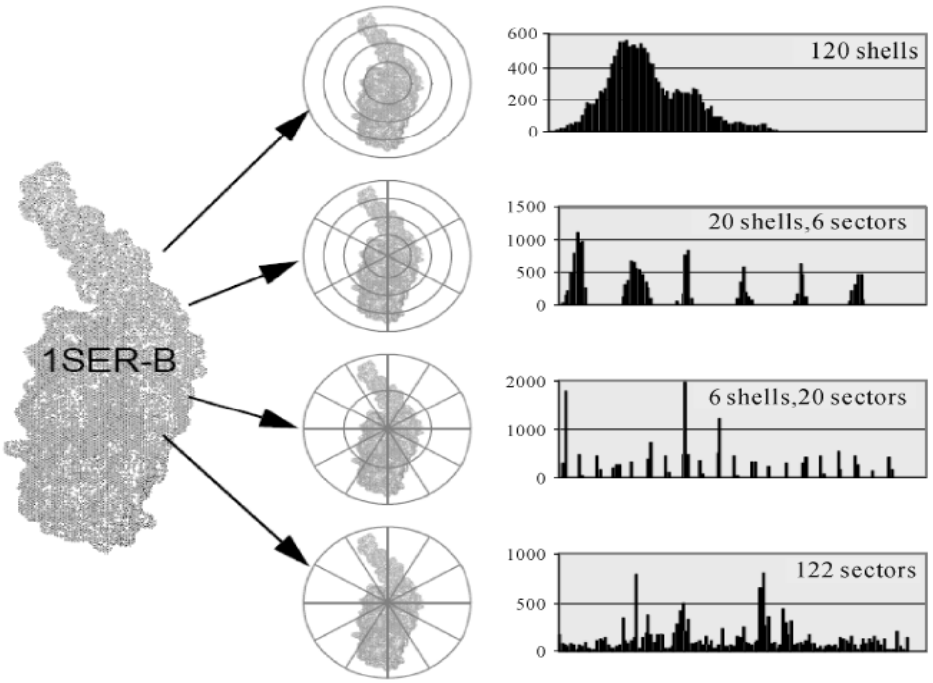


Figure II-8: Plusieurs histogrammes de forme 3D de la protéine 1SER-B. De haut en bas, le nombre de coquilles diminue et le nombre de secteurs augmente [Ankerst et al. 1999a].

II.4.3.3.3 Histogramme de distance

Pour les formes 3D rigides, Novotni et al [Novotni et al. 2001] ont introduit ce que l'on appelle les "histogrammes de distance" comme représentation de base. Leur idée fondamentale est que si deux objets étaient similaires, seule une petite partie du volume de l'un des objets serait en dehors de la limite de l'autre, et la distance moyenne de la limite serait également petite. Ils ont d'abord calculé les coques décalées de chaque objet sur la base d'un champ de distance 3D, puis ont construit les histogrammes de distance pour chaque objet afin d'indiquer quelle partie du volume d'un objet se trouve à l'intérieur de la coque décalée de l'autre.

II.4.3.3.4 Descripteur de forme multirésolution

L'introduction de propriétés géométriques dans l'histogramme rend possible la représentation de formes multirésolution. Les auteurs de [Ohbuchi et al. 2003] ont proposé un descripteur de forme multirésolution, représenté sous la forme d'un ensemble ordonné d'histogrammes. Ils ont d'abord défini une caractéristique de représentation multirésolution, spécifiée comme un ensemble de formes 3D α-shapes [Edelsbrunner et al. 1994], qui a été définie en utilisant un groupe de valeurs α espacées à la puissance de deux intervalles. Les formes α-shapes sont une généralisation de la coque convexe d'un ensemble de points, qui se rétrécit en développant progressivement des cavités jusqu'à ce qu'elle soit identique à la coque convexe lorsque $\alpha = \infty$ [30]. Ensuite, un histogramme 2D a été généré pour chaque représentation multirésolution afin qu'un ensemble ordonné d'histogrammes puisse être produit comme descripteur de forme.

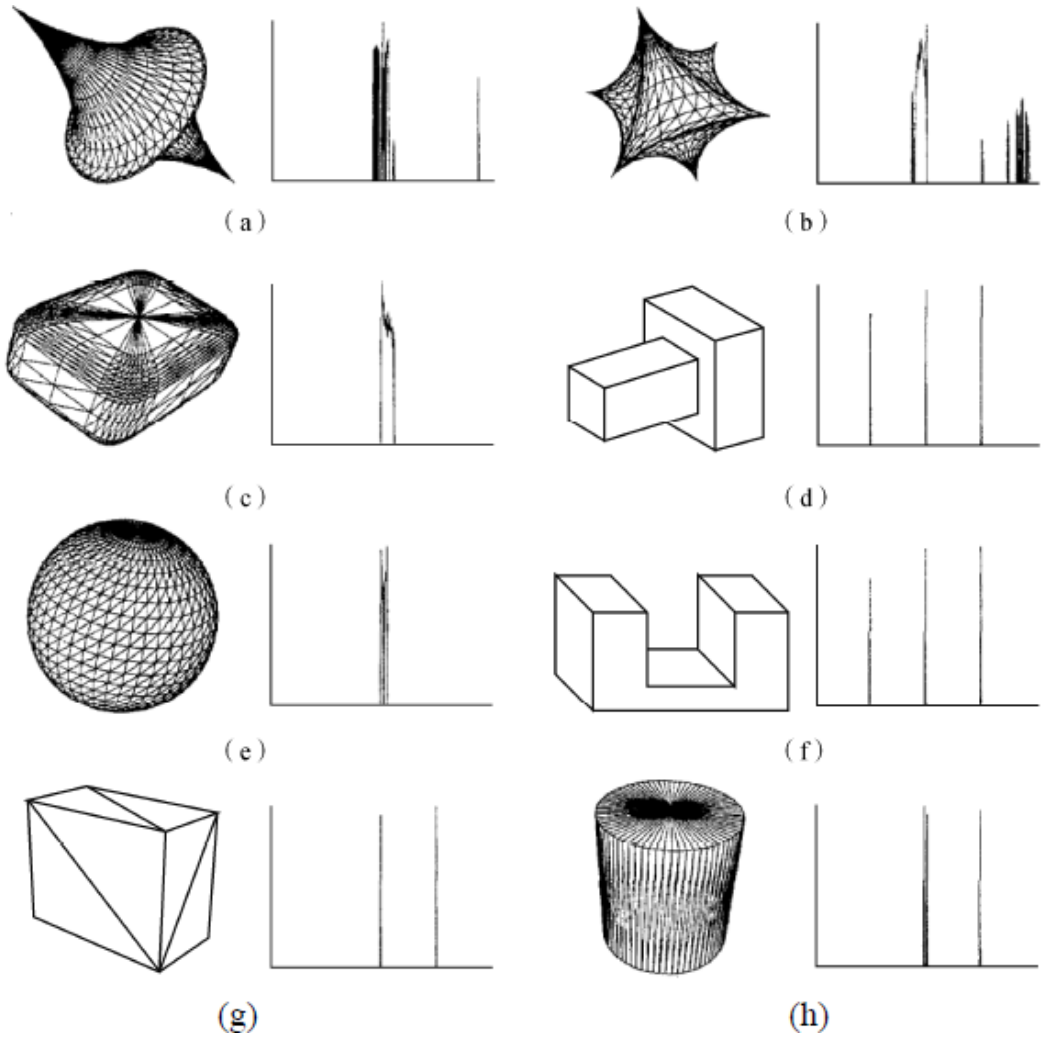


Figure II-9: Histogrammes des angles de pliage pour les formes simples. (a) Superquadrique à double corne ; (b) Superquadrique en forme de jack ; Superquadrique en savon (c) ; (d) Deux blocs collés ; (e) Sphère ; (f) Bloc avec canal ; (g) Bloc ; (h) Cylindre [Besl 1929].

II.4.3.3.5 Autres histogrammes

Paquet et al [Paquet et al. 1998] ont présenté les caractéristiques des histogrammes, notamment l'histogramme couleur, l'histogramme vectoriel normal et l'histogramme matériel pour représenter les formes 3D. Paquet et al. ont également souligné qu'un histogramme peut représenter les distributions de données 3D, basées sur des voxels, et est invariant de la transformation. Dans la norme MPEG-7, il existe également un descripteur d'histogramme de forme pour un objet 3D connu sous le nom de descripteur de spectre de forme 3D.

II.4.3.4 Densité de points

Suzuki et al [Suzuki et al. 2000] ont suggéré que plusieurs étapes sont nécessaires pour créer des descripteurs de caractéristiques invariants de rotation : (1) Les informations associées aux caractéristiques de forme doivent être extraites des fichiers de données; (2) Les informations extraites sont converties en vecteurs de caractéristiques sous forme d'indices de la base de données; (3) Les vecteurs de caractéristiques sont regroupés en classes d'équivalence, de sorte que ces vecteurs puissent être convertis en vecteurs de caractéristiques invariants à la rotation. Dans leur papier, seules les formes des objets 3D sont concernées, donc seules les informations relatives aux sommets sont utilisées.

Lorsqu'un objet 3D est affiché, un ensemble de points est utilisé pour représenter la forme. Cet ensemble de points est relié par des lignes pour former un cadre filaire. Ce cadre montre un ensemble de polygones. Une fois que les polygones ont été créés, l'algorithme de rendu peut ombrager les polygones individuels pour produire un objet solide. Suzuki et al ont utilisé la densité des nuages de points comme vecteurs de caractéristiques. Chaque objet 3D est placé dans le cube unitaire, puis le cube unitaire est divisé en grilles grossières. Le nombre de points est compté dans chaque cellule de la grille pour calculer la densité des nuages de points. Dans leur papier, seule la densité des nuages de points est utilisée. Cependant, d'autres caractéristiques peuvent également être utilisées, comme les vecteurs normaux des faces de polygones.

Comme la distribution des nuages de points dépend de la façon dont l'objet 3D est généré, ils ont normalisé les positions des points en utilisant des programmes de triangulation de polygones. La densité des nuages de points nous donne des descripteurs de forme grossière des objets 3D qui comprennent la courbure, la hauteur, la largeur et les positions. Ces descripteurs de caractéristiques ne sont pas invariants en termes de rotation, car les orientations des objets 3D sont définies par ceux qui ont conçu les modèles 3D. Les orientations peuvent être normalisées par des règles. Les règles appropriées pour définir les orientations des objets 3D dépendent de l'objectif des applications.

II.4.3.5 Fonctions de distribution des formes

Osada et al [Osada et al. 2002] ont décrit et analysé une méthode de calcul des signatures de formes 3D et des mesures de dissimilarité pour des objets arbitraires décrits par des modèles polygonaux 3D éventuellement dégénérés. L'idée clé est de représenter la signature d'un objet comme une distribution de forme échantillonnée à partir d'une fonction de forme mesurant les propriétés géométriques globales de l'objet. La principale motivation de cette approche est que le problème de la correspondance des formes est réduit à la comparaison de deux distributions de probabilité, ce qui est un problème relativement simple par rapport aux problèmes plus difficiles rencontrés par les méthodes traditionnelles de correspondance des formes, comme l'enregistrement des poses, le paramétrage, la correspondance des caractéristiques et l'ajustement des modèles. Les défis de cette approche consistent à sélectionner des fonctions de forme discriminantes, à développer des méthodes efficaces pour les échantillonner et à calculer de manière robuste la dissimilitude des distributions de probabilité.

Sélection d'une fonction de forme

La première et la plus intéressante question est de sélectionner une fonction dont la distribution fournit une bonne signature pour la forme d'un modèle polygonal 3D. Idéalement, la distribution devrait être invariante sous les transformations de similarité, et elle devrait être insensible au bruit, aux fissures, à la tessellation et à l'insertion/suppression de petits polygones.

En général, toute fonction peut être échantillonnée pour former une distribution de forme, y compris celles qui incorporent des connaissances spécifiques à un domaine, des informations de visibilité (par exemple, la distance entre des points aléatoires mais mutuellement visibles), et/ou des attributs de surface (par exemple, la couleur, les coordonnées de texture, les normales et la courbure). Cependant, dans un souci de clarté, Osada et al. se sont concentrés sur un petit ensemble de fonctions de forme basées sur des mesures géométriques (par exemple, angles, distances, surfaces et volumes). Plus précisément, dans leur étude initiale, ils ont expérimenté les fonctions de forme suivantes (voir la figure II-10) :

(1) A3 : mesure l'angle entre trois points aléatoires sur la surface d'un modèle 3D.

(2) D1 : Mesure la distance entre un point fixe et un point aléatoire sur la surface. On utilise le centroïde de la limite du modèle comme point fixe.

(3) D2 : mesure la distance entre deux points aléatoires de la surface.

(4) D3 : Mesure la racine carrée de l'aire du triangle entre trois points aléatoires sur la surface.

(5) D4 : mesure la racine cubique du volume du tétraèdre entre quatre points aléatoires de la surface.

Ces cinq fonctions de forme ont été choisies principalement pour leur simplicité et leur invariance. En particulier, elles sont rapides à calculer, faciles à comprendre et produisent des distributions qui sont invariantes aux mouvements rigides (translations et rotations). Elles sont invariantes à la tessellation du modèle polygonal 3D, puisque les points sont choisis au hasard sur la surface. Elles sont insensibles aux petites perturbations dues au bruit, aux fissures, et à l'insertion/suppression de polygones, puisque l'échantillonnage est pondéré en fonction de la surface. En outre, la fonction de forme A3 est invariante à l'échelle, tandis que les autres doivent être normalisées pour permettre des comparaisons. Enfin, les fonctions de forme D2, D3 et D4 permettent une comparaison intéressante des mesures géométriques 1D, 2D et 3D.

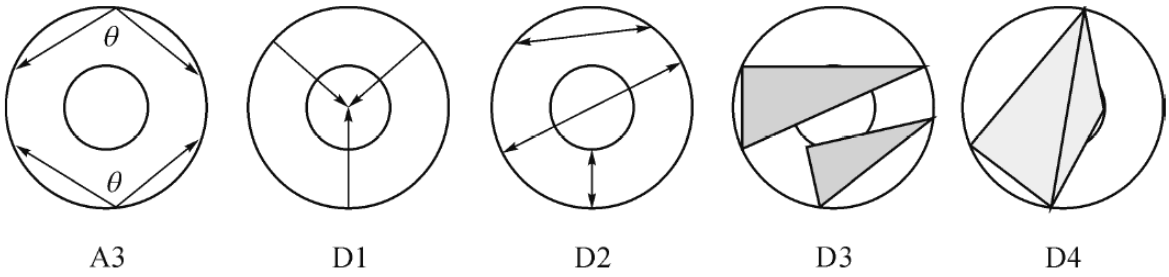


Figure II-10: Cinq fonctions de forme simples basées sur les angles (A3), les longueurs (D1, D2), les surfaces (D3) et les volumes (D4)

Malgré leur simplicité, Osada et al. ont trouvé que ces fonctions de forme polyvalentes se distinguaient assez bien comme signatures de la forme 3D, car des changements significatifs des structures rigides dans le modèle 3D affectent les relations géométriques entre les points de leurs surfaces. Par exemple, on peut remarquer que les distributions de la fonction de forme D2 sont indiquées pour quelques formes canoniques dans les figures II-11(a)-(f). Chaque distribution est distincte. Et les changements continus du modèle 3D affectent les distributions D2. Par exemple, la figure II-11(g) montre les distributions de distance pour des ellipsoïdes de différentes longueurs de demi-axe superposés sur le même graphique. La courbe la plus à gauche représente la distribution D2 pour un segment de ligne-ellipsoïde (0, 0, 1) ; la courbe la plus à droite représente la distribution D2 pour une sphère-ellipsoïde (1, 1, 1) ; et les autres courbes montrent la distribution D2 pour les ellipsoïdes entre-ellipsoïdes (r, r, 1) avec $0 < r < 1$. Notez que le changement de sphère à segment de droite est continu. De même, les figures II-11(h)-(i) montrent les distributions D2 de deux sphères unitaires lorsqu'elles s'éloignent de 0, 1, 2, 3 et 4 unités. Dans chaque distribution, la première bosse ressemble à la distribution linéaire d'une sphère, tandis que la seconde bosse est le croisement des distances entre les deux sphères. Au fur et à mesure que les sphères s'éloignent l'une de l'autre, la distribution de D2 change continuellement.

Construire des distributions de formes

Une fonction de forme ayant été choisie, la question suivante est de calculer et de stocker une représentation de sa distribution. Le calcul analytique de la distribution n'est possible que pour certaines combinaisons de fonctions de forme et de modèles (par exemple, la fonction D2 pour une sphère ou une droite). Ainsi, en général, Osada et al. ont utilisé des méthodes stochastiques. Plus précisément, Osada et al. ont évalué N échantillons de la

distribution des formes et ont construit un histogramme en comptant combien d'échantillons tombent dans chacun des B bacs de taille fixe. À partir de l'histogramme, Osada et al. ont reconstruit une fonction linéaire par morceaux avec V (≤ B) sommets équidistants, qui constitue la représentation de la distribution des formes. Osada et al. ont calculé la distribution des formes une fois pour chaque modèle et l'ont stockée sous la forme d'une séquence de V entiers.

Une question dont nous devons nous préoccuper est celle de la densité d'échantillonnage. D'une part, plus nous prélevons d'échantillons, plus nous pouvons reconstruire avec précision et exactitude la distribution des formes. D'autre part, le temps nécessaire pour échantillonner une distribution de forme est linéairement proportionnel au nombre d'échantillons, de sorte qu'il y a un compromis précision/temps dans le choix de N. De même, un plus grand nombre de sommets donne des distributions de plus haute résolution, tout en augmentant les coûts de stockage et de comparaison de la signature de forme. Dans leurs expériences, Osada et al. ont choisi de privilégier la robustesse, en prélevant un grand nombre d'échantillons pour chaque case de l'histogramme. Empiriquement, ils ont constaté qu'en utilisant N = 1 024^2 échantillons, B = 1 024 bacs et V = 64 sommets, on obtient des distributions de formes avec une variance suffisamment faible et une résolution suffisamment élevée pour être utiles à leurs premières expériences. Des méthodes d'échantillonnage adaptatives pourraient être utilisées dans les travaux futurs pour rendre plus efficace la construction robuste des distributions de formes.

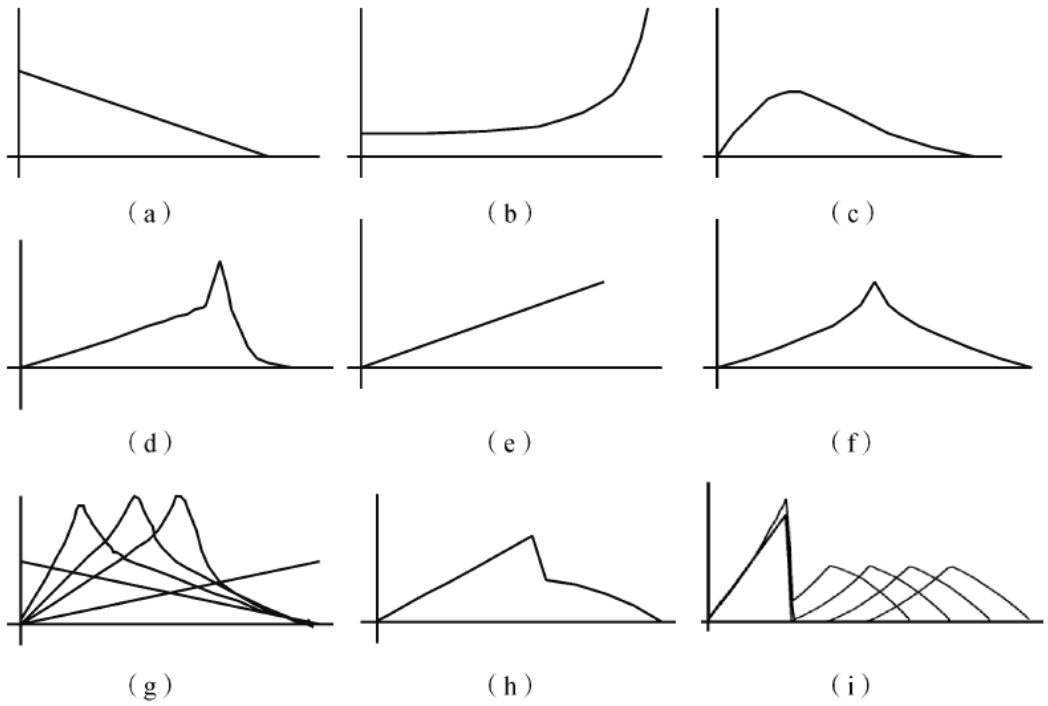


Figure II-11 : Exemple de distributions de formes D2. Dans chaque graphique, l'axe horizontal représente la distance, et l'axe vertical représente la probabilité de cette distance entre deux points de la surface. (a) Segment de droite ; (b) Cercle (périmètre seulement) ; (c) Triangle ; (d) Cube ; (e) Sphère ; (f) Cylindre (sans calottes) ; (g) Ellipsoïdes de rayons différents ; (h) Deux sphères unitaires adjacentes ; (i) Deux sphères unitaires séparées par 1, 2, 3 et 4 unités

Un deuxième problème est la génération d'échantillons. Bien qu'il soit plus simple d'échantillonner directement les sommets du modèle 3D, les distributions de formes résultantes seraient biaisées et sensibles aux changements de tessellation. Au lieu de cela, les fonctions de forme d'Osada et al. sont échantillonnées à partir de points aléatoires à la

surface d'un modèle 3D. La méthode de génération de points aléatoires non biaisés par rapport à la surface d'un modèle polygonal se déroule comme suit. Tout d'abord, Osada et al. ont itéré à travers tous les polygones, les divisant en triangles si nécessaire. Ensuite, pour chaque triangle, Osada et al. ont calculé sa surface et l'ont stockée dans un tableau avec la surface cumulée des triangles visités jusqu'à présent. Ensuite, Osada et al. ont sélectionné un triangle avec une probabilité proportionnelle à sa surface en générant un nombre aléatoire entre 0 et la surface cumulée totale et ont effectué une recherche binaire sur le tableau des surfaces cumulées. Pour chaque triangle sélectionné ayant des sommets (A, B, C), Osada et al. ont construit un point sur sa surface en générant deux nombres aléatoires, r_1 et r_1, entre 0 et 1, et ont évalué l'équation suivante :

$$P = \left(1 - \sqrt{r_1}\right)A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C \qquad \text{(II-19)}$$

Intuitivement, $\sqrt{r_1}$ fixe le pourcentage du sommet A au bord opposé, tandis que r_2 représente le pourcentage le long de ce bord (voir figure II-12). En prenant la racine carrée de r_1, on obtient un point aléatoire uniforme par rapport à la surface.

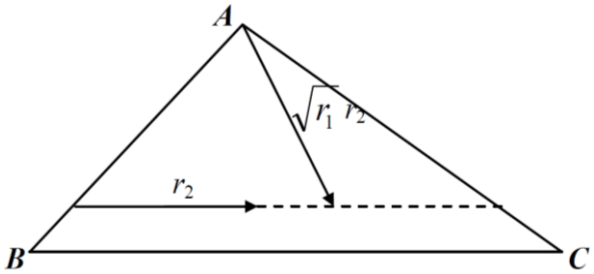


Figure II-12: Echantillonnage d'un point aléatoire dans un triangle.

Les résultats expérimentaux d'Osada et al. ont démontré que les distributions de formes peuvent être assez efficaces pour discriminer entre des groupes de modèles 3D. Dans l'ensemble, ils ont atteint une précision de 66 % dans leurs expériences de classification avec une base de données diversifiée de modèles 3D dégénérés affectés à des groupes fonctionnels. La distribution des formes D2 a été plus efficace que les moments pendant leurs tests de classification. Malheureusement, il est difficile d'évaluer la qualité de ce résultat par rapport à d'autres méthodes, car il dépend en grande partie des détails de la base de données de test. Cependant, ils estiment que leur méthode s'est avérée utile pour la discrimination des formes 3D, au moins pour la pré-classification avant des comparaisons de similarité plus exactes avec des méthodes plus coûteuses.

Méthodes améliorées

Osada et al. ont montré que le D2 est la meilleure caractéristique parmi leurs cinq caractéristiques. Il représente la distribution des distances entre deux points aléatoires. Cette caractéristique est invariante à la tessellation des modèles polygonaux 3D, puisque les points sont choisis au hasard sur la surface de l'objet. Cependant, il est sensible aux petites déformations dues au bruit, aux fissures, ou à l'insertion/suppression de polygones, puisque l'échantillonnage est pondéré en fonction de la surface. Pour représenter finement les composants complexes d'un objet 3D, un modèle 3D nécessite souvent de nombreux polygones. L'échantillonnage aléatoire d'un modèle 3D serait dominé par ces composants complexes. Ainsi, une nouvelle caractéristique, appelée grille D2, est proposée par Shih et al. [Shih et al. 2005] pour améliorer les performances de la grille D2 traditionnelle. Tout d'abord, le modèle 3D est décomposé par une grille de voxels. Un voxel est considéré comme valide si une surface polygonale y est située, et invalide dans le cas contraire. Ensuite, la distribution des distances entre deux voxels valides au lieu de deux points sur la surface est calculée. Par conséquent, le défaut pondéré de la surface dans le processus

d'échantillonnage sera grandement réduit puisque chaque voxel valide est pondéré de manière égale, quel que soit le nombre de points situés à l'intérieur de ce voxel. Les principales étapes du calcul de la grille D2 sont décrites ci-dessous :

(1) Tout d'abord, un modèle 3D est segmenté en une grille de voxels $2R \times 2R \times 2R$. Pour être invariant à la translation et à la mise à l'échelle, le centre de masse de l'objet est déplacé à l'emplacement (R, R, R) et la distance moyenne des voxels valides au centre de masse est mise à l'échelle pour être $R/2$. R est fixé à 32, ce qui offre une résolution adéquate pour discriminer les objets tout en filtrant les surfaces polygonales à haute fréquence dans les composants complexes d'un objet 3D.

(2) Deux voxels valides sont choisis au hasard et leur distance est mesurée. Un total de U distances est évalué à partir de l'ensemble des voxels valides. Un histogramme contenant 256 bacs est construit : $H = \{B_1, B_2, \ldots, B_{256}\}$, où B_i désigne le nombre de distances dans la plage du i-ième bac. Pour normaliser la distribution, la grille D2 (GD2) est définie comme suit :

$$GD2 = \left\{\frac{B_1}{U}, \frac{B_2}{U}, \frac{B_3}{U}, \ldots, \frac{B_{256}}{U}\right\} \quad \text{(II-20)}$$

Où U est fixé à 643. La figure II-13 montre que les distributions D2 sont clairement différentes, tandis que les distributions GD2 sont similaires pour ces deux avions similaires. Les résultats expérimentaux montrent que la méthode de Shih et al. est supérieure aux autres, et le nouveau descripteur de forme est à la fois discriminant et robuste.

En outre, Song et al. [Song et al. 2003] ont également adopté une représentation par histogramme, basée sur des fonctions de forme pour faire correspondre des formes 3D en générant des histogrammes utilisant la courbure gaussienne discrète et la courbure moyenne discrète de chaque sommet d'un maillage triangulaire 3D.

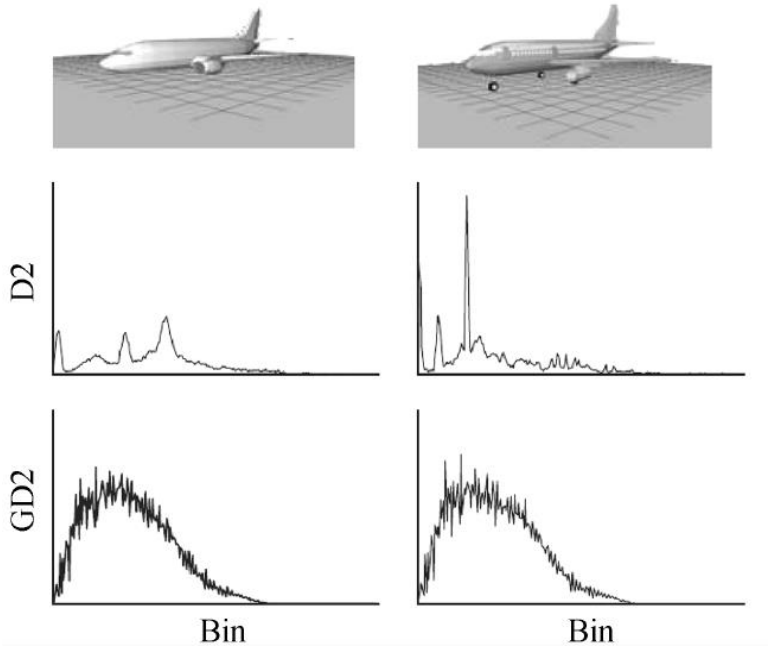


Figure II-13: Distributions de D2 et GD2 pour deux avions similaires [Shih et al. 2005].

II.4.3.6 Image gaussienne étendue

Dans [Horn 1986], Horn a défini l'image gaussienne étendue (Extended Gaussian Image, EGI), a discuté de ses propriétés et a donné des exemples. Des méthodes pour

déterminer les images gaussiennes étendues des polyèdres, des solides de révolution et des objets à courbure douce en général ont été présentées. L'histogramme d'orientation, une approximation discrète de l'image gaussienne étendue, a été décrit ainsi que diverses façons de tesseler la sphère. Les concepts et propriétés détaillés de l'EGI peuvent être décrits comme suit.

Définitions de l'image gaussienne étendue pour les polyèdres convexes

Minkowski a montré en 1897 qu'un polyèdre convexe est entièrement spécifié par la surface et l'orientation de ses faces. Les informations vectorielles normales de surface pour tout objet peuvent être cartographiées sur une sphère unitaire, appelée sphère gaussienne. On peut représenter l'aire et l'orientation des faces de façon pratique par des masses ponctuelles sur cette sphère. Un poids est attribué à chaque point de la sphère gaussienne, égal à l'aire de la surface ayant la normale donnée. Les poids sont représentés par des vecteurs parallèles aux normales de la surface, dont la longueur est égale au poids. Imaginez que l'on déplace la normale de surface unitaire de chaque face de sorte que sa queue se trouve au centre d'une sphère unitaire. La tête de la normale unitaire se trouve alors sur la surface de la sphère unitaire. Chaque point de la sphère gaussienne correspond à une orientation particulière de la surface. L'image gaussienne étendue du polyèdre est obtenue en plaçant en chaque point une masse égale à la surface de la face correspondante.

Il semble à première vue que certaines informations soient perdues dans cette cartographie, puisque la position des normales de surface est écartée. Vu sous un autre angle, aucune note n'est faite sur la forme des faces ou sur leurs relations de proximité. On peut néanmoins montrer que l'image gaussienne étendue définit uniquement un polyèdre convexe. Des algorithmes itératifs peuvent être utilisés pour récupérer un polyèdre convexe à partir de son image gaussienne étendue.

Image gaussienne pour les surfaces à courbure douce

On peut associer un point sur la sphère gaussienne à un point donné sur une surface en trouvant le point sur la sphère qui a la même normale de surface. Il est donc possible de faire correspondre des informations associées à des points de la surface à des points de la sphère gaussienne. Dans le cas d'un objet convexe ayant une courbure gaussienne positive partout, il n'y a pas deux points qui ont la même normale de surface. Dans ce cas, la correspondance entre l'objet et la sphère gaussienne est inversible : A chaque point de la sphère gaussienne correspond un point unique sur la surface. Si la surface convexe comporte des taches de courbure gaussienne nulle, les courbes ou même les zones qui s'y trouvent peuvent correspondre à un seul point de la sphère gaussienne.

Une propriété utile de l'image gaussienne est qu'elle tourne avec l'objet. Considérons deux normales de surface parallèles, l'une sur l'objet et l'autre sur la sphère gaussienne. Les deux normales resteront parallèles si l'objet et la sphère gaussienne tournent de la même manière. Une rotation de l'objet correspond donc à une rotation égale de la sphère gaussienne.

Courbure gaussienne pour des surfaces à courbure douce

Envisagez un petit patch δO sur l'objet. Chaque point de ce patch correspond à un point particulier de la sphère gaussienne. Le patch δO sur l'objet se transforme en un patch, δS par exemple, sur la sphère gaussienne. D'une part, si la surface est fortement courbée, les normales des points de la pièce pointeront dans un large éventail de directions. Les points correspondants sur la sphère gaussienne seront étalés. D'autre part, si la surface est plane, les normales de la surface sont parallèles et s'inscrivent en un seul point.

Ces considérations suggèrent une définition appropriée de la courbure. La courbure gaussienne est définie comme étant égale à la limite du rapport entre les deux surfaces car elles tendent vers zéro. C'est-à-dire,

$$K = \lim_{\delta O \to 0} \frac{\delta S}{\delta O} = \frac{\delta O}{\delta O} \tag{II-21}$$

De cette relation différentielle, nous pouvons obtenir deux intégrales utiles. Considérons d'abord l'intégration de K sur une tache finie O sur l'objet :

$$\iint_O K dO = \iint_S dS = A_s \tag{II-22}$$

Où A_s est la zone de la tache correspondante sur la sphère gaussienne. L'expression de gauche est appelée la courbure intégrale. Cette relation permet de traiter les surfaces qui présentent des discontinuités dans la normale à la surface.

Envisagez maintenant d'intégrer $1/K$ sur un patch S sur la sphère gaussienne

$$\iint_S (1/K)\, dS = \iint_O dO = A_O \tag{II-23}$$

Où A_O est la zone du patch correspondant sur l'objet. Cette relation suggère l'utilisation de l'inverse de la courbure gaussienne dans la définition de l'image gaussienne étendue d'un objet à courbure douce, comme nous le verrons. Elle montre également, en passant, que l'intégrale de $1/K$ sur toute la sphère gaussienne est égale à la surface totale de l'objet.

Définition de l'image gaussienne étendue pour les surfaces à courbure douce

On peut définir une cartographie qui associe l'inverse de la courbure gaussienne en un point de la surface de l'objet avec le point correspondant de la sphère gaussienne. Soient u et v des paramètres utilisés pour identifier des points sur la surface d'origine. De la même façon, que ξ et η soient des paramètres utilisés pour identifier des points sur la sphère gaussienne. Il peut s'agir de la longitude et de la latitude, par exemple. Ensuite, on définit l'image gaussienne étendue comme :

$$G(\xi, \eta) = \frac{1}{K(u, v)} \tag{II-24}$$

Où (ξ, η) est le point sur la sphère gaussienne qui a la même normale que le point (u, v) sur la surface d'origine. On peut montrer que cette cartographie est unique pour les objets convexes. C'est-à-dire qu'il n'y a qu'un seul objet convexe correspondant à une image gaussienne étendue particulière. La preuve est malheureusement non constructive et aucune méthode directe de récupération de l'objet n'est connue.

Propriétés de l'image gaussienne étendue pour les polyèdres convexes

L'image gaussienne étendue n'est pas affectée par la translation de l'objet. La rotation de l'objet entraîne une rotation égale de l'image gaussienne étendue, puisque les normales de surface unitaires tournent avec l'objet.

Les distributions de masse, qui se situent entièrement dans un hémisphère, sont nulles dans l'hémisphère complémentaire et ne correspondent pas à des objets fermés. On peut démontrer que le centre de masse d'une image gaussienne étendue doit se situer à l'origine. Ceci est clairement impossible si l'hémisphère entier est vide. De plus, une distribution de masse non nulle uniquement sur un grand cercle de la sphère correspond à la limite d'une séquence d'objets cylindriques de longueur croissante et de diamètre décroissant. Ici, de tels cas pathologiques sont exclus et l'attention est limitée aux objets fermés et délimités.

Certaines propriétés de l'image gaussienne étendue sont importantes. Premièrement, la masse totale de l'image gaussienne étendue est évidemment juste égale à la surface totale du polyèdre. Si le polyèdre est fermé, il aura la même surface projetée lorsqu'il est vu de n'importe quelle paire de directions opposées. Cela permet de calculer l'emplacement du centre de masse de l'image gaussienne étendue.

Une représentation équivalente, appelée modèle en épi, est un ensemble de vecteurs dont chacun est parallèle à l'une des normales de surface et de longueur égale à la surface de la face correspondante. Le résultat concernant le centre de masse est équivalent à l'affirmation que ces vecteurs doivent former une chaîne fermée lorsqu'ils sont placés bout à bout.

II.4.4 Extraction de caractéristiques de la géométrie globale

La géométrie globale d'un modèle 3D est analysée en échantillonnant directement l'ensemble des sommets, l'ensemble des mailles des polygones ou l'ensemble des voxels dans le domaine spatial. Le rapport d'aspect, le bitmap binaire de voxels 3D et les angles 3D des sommets ou des arêtes peuvent être considérés comme les caractéristiques les plus simples et les plus directes [Kolonias et al. 2001], bien que leur pouvoir discriminant soient limitées. Ces types d'analyses utilisent généralement des méthodes de type ACP pour aligner le modèle dans un cadre de coordonnées canonique dans un premier temps, puis définir la représentation de la forme sur cette orientation normalisée.

La caractéristique commune de ces méthodes est qu'elles sont presque toutes dérivées directement de l'unité élémentaire d'un modèle 3D, c'est-à-dire le sommet, le polygone ou le voxel, et un modèle 3D est visualisé et traité comme un ensemble de sommets, un ensemble de mailles de polygones ou un ensemble de voxels. Leurs avantages résident dans leur dérivation facile et directe à partir de structures de données 3D, ainsi que dans leur pouvoir de représentation relativement bon. Toutefois, les processus de calcul sont généralement trop longs et trop sensibles pour les petites caractéristiques. De plus, les exigences de stockage sont trop élevées en raison des difficultés à mettre en place un mécanisme d'indexation concis et efficace pour ces données dans les grandes bases de données de modèles.

II.4.4.1 Représentation des caractéristiques géométriques basées sur les rayons

Vranić et al. [Vranić et al. 2000] ont proposé une représentation des caractéristiques géométriques basée sur les rayons. Ils ont échantillonné un modèle 3D dans son cadre de coordonnées canoniques comme un ensemble de vecteurs de direction régulièrement espacés et ont placé des rayons le long de chaque vecteur de direction à partir de l'origine des coordonnées, qui a coupé le maillage triangulaire d'un polyèdre entourant le modèle 3D. Pour chaque direction, la distance maximale entre le maillage triangulaire coupé et les coordonnées d'origine a été calculée et tous les échantillons de distance ont composé un vecteur caractéristique. Le processus détaillé peut être exprimé comme suit.

Prétraitement avec la technologie ACP modifiée

Vranić et al. ont incorporé une modification de l'analyse en composantes principales (ACP) dans le module d'extraction des caractéristiques géométriques. Cette transformation transforme les axes du système de coordonnées en de nouveaux axes qui coïncident avec les directions des trois plus grands écarts de la distribution des points (c'est-à-dire des sommets). Un objet 3D représentant un maillage triangulaire est constitué de géométrie, de topologie et d'attributs. La géométrie est déterminée par les coordonnées des sommets, les informations sur la façon dont les sommets sont connectés afin de former des triangles sont appelées topologie et les attributs sont la couleur, la texture, etc. Dans leur système, les attributs ne sont pas encore pris en compte car l'accent est mis sur la représentation des relations spatiales dans un modèle 3D, c'est-à-dire la géométrie et la topologie.

L'objectif de l'analyse en composantes principales appliquée au modèle 3D est de rendre le vecteur des caractéristiques de la forme résultante aussi indépendant que possible de la translation et de la rotation. L'ACP sera basée sur la collecte de vecteurs verticaux. Pour tenir compte des différentes tailles des triangles correspondants, Vranić et al. ont introduit des facteurs de pondération proportionnels à la surface correspondante.

Extraction des caractéristiques

Supposons que nous ayons un ensemble donné de L vecteurs directionnels $\{u_1, u_2, \ldots, u_L\}$, comme le montre la figure II-14. Le maillage triangulaire est alors coupé avec le rayon émanant de l'origine du système de coordonnées de l'ACP et se déplaçant dans la direction u_i $(i \in \{1, \ldots, L\})$. La distance à l'intersection la plus éloignée est prise comme la i^{ième} composante du vecteur caractéristique qui est mis à l'échelle de la longueur de l'unité euclidienne pour assurer l'invariance de l'échelle. Dans l'expérience de Vranić et al., L est fixé à 20. Les sommets d'un dodécaèdre, avec le centre dans les coordonnées d'origine, sont pris comme directions. Cette caractéristique est invariante en ce qui concerne la rotation et la translation en raison du fait que les axes des coordonnées initiales sont transformés. L'invariance d'échelle est réalisée en normalisant le vecteur de la caractéristique.

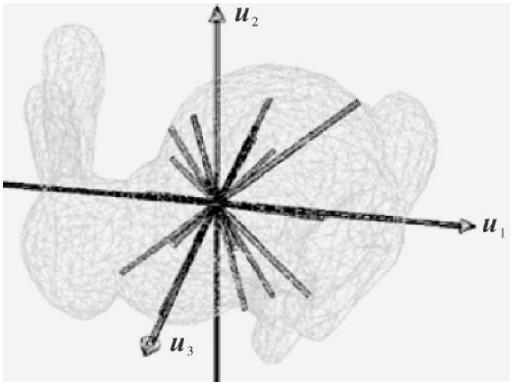


Figure II-14: Illustration du descripteur de forme basé sur les rayons [Vranić et al. 2001].

Description des caractéristiques

Après l'extraction des caractéristiques, l'étape suivante est leur description formelle. Comme nous le savons, la norme MPEG-7 fournit un riche ensemble de mécanismes et de moyens normalisés visant à décrire le contenu multimédia. La terminologie MPEG-7 a été adoptée et la relation mutuelle entre un descripteur et une caractéristique est expliquée dans la définition suivante : Un descripteur est une représentation d'une caractéristique. Un descripteur est utilisé pour définir la syntaxe et la sémantique de la représentation d'une caractéristique [MPEG Requirements Group 1999].

Par conséquent, le descripteur du vecteur de caractéristique ci-dessus est déterminé avec 20 nombres réels non négatifs, où la i-ième composante est l'extension de l'objet dans la direction du i-ième sommet du dodécaèdre mentionné, qui est défini (les coordonnées du sommet et la numérotation) en interne. Ceci définit la sémantique du descripteur. La syntaxe est définie par des schémas de description (Description Schemes, DS) pour les vecteurs réels.

Le MPEG-7 n'est pas un système restrictif pour la description du contenu audiovisuel. Il s'agit d'un cadre flexible et extensible pour la description de données multimédia avec un ensemble de méthodes et d'outils développés. Comme mentionné dans le MPEG-7, le DS du modèle 3D doit prendre en charge "la représentation hiérarchique de différents descripteurs afin que les requêtes puissent être traitées plus efficacement aux niveaux successifs (où les descripteurs de niveau N complètent les descripteurs de niveau (N-1))". Il convient donc de prendre en compte différentes caractéristiques à différents

niveaux de détail. Vranić et al. ont été encouragés par le réflecteur du groupe DS MPEG-7 à mettre en œuvre leur propre DS pour les modèles 3D. Ce DS doit être conforme à la spécification MPEG-7.

Autres méthodes

En utilisant une idée similaire, les auteurs de [Yu et al. 2003] ont extrait la géométrie globale 3D comme une carte de distance et des caractéristiques de la carte de pénétration de surface. Ces deux cartes de caractéristiques spatiales décrivent la géométrie et la topologie des patchs de surface sur l'objet, tout en préservant les informations spatiales des patchs dans les cartes. Les cartes de caractéristiques saisissent la quantité d'effort nécessaire pour transformer un objet 3D en une sphère canonique, sans effectuer de transformation 3D explicite. Lorsqu'un objet 3D est donné, il est d'abord mis à l'échelle et intégré dans une sphère de rayon unitaire de telle sorte que le centre de la sphère coïncide avec le centroïde de l'objet. Ensuite, un rayon est tiré depuis le centre de la sphère à travers chaque point de l'objet jusqu'à la surface de la sphère, comme le montre la figure II-15. La distance parcourue par le rayon depuis un point de l'objet jusqu'à la surface de la sphère est enregistrée dans la carte des distances (Distance Map, DM). Les transformées de Fourier des cartes de caractéristiques sont utilisées pour la comparaison des objets afin d'obtenir une récupération invariante en cas de rotation arbitraire, de réflexion et de mise à l'échelle non uniforme des objets. Les résultats expérimentaux montrent que leur méthode de récupération de modèles 3D est très précise, atteignant une précision supérieure à 0,86, même à un taux de rappel de 1,0.

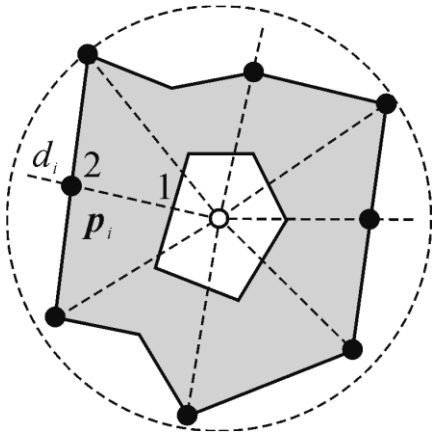


Figure II-15: Calculer des cartes de caractéristiques. Les lignes pointillées sont tracées à partir du centre (point blanc) d'une sphère délimitée (cercle pointillé), en passant par les points de l'objet (points noirs), jusqu'à la surface de la sphère. La distance d_i parcourue par le rayon depuis un point p_i jusqu'à la surface de la sphère et le nombre de surfaces d'objets (lignes pleines ; 2, dans ce cas) pénétrées par le rayon depuis qu'il quitte le centre de la sphère sont enregistrés dans les cartes de caractéristiques [Yu et al. 2003].

II.4.4.2 Ensembles de points pondérés

Tangelder et al. ont proposé une méthode utilisant des ensembles de points pondérés comme descripteur de forme pour un maillage polygonal 3D [Tangelder et al. 2003]. Ils ont supposé qu'une forme 3D est représentée par un maillage polyédrique. Ils n'exigent pas que le maillage polyédrique soit fermé. Par conséquent, leur méthode peut également traiter des modèles polyédriques qui peuvent contenir des lacunes. Ils ont également enveloppé l'objet dans une grille de voxels 3D et ont représenté la forme comme un ensemble de points pondérés en sélectionnant un point représentatif pour chaque cellule de la grille non vide. Ils ont ensuite sélectionné le sommet ayant la courbure gaussienne la plus élevée ou la moyenne pondérée par la surface de tous les sommets d'une cellule de grille, pour représenter les caractéristiques géométriques du modèle.

De nombreuses méthodes mentionnées dans les sections précédentes ne tiennent pas compte de la position spatiale relative globale, mais rejettent certaines de ces informations, afin de traiter des données de moindre complexité, par exemple des vues en 2D ou des histogrammes en 1D. Ce qui est nouveau dans la méthode de Tangelder et al. est qu'ils utilisent la position spatiale relative globale en représentant la forme 3D comme un ensemble de points pondérés, sans tenir compte des relations de connectivité. Les ensembles de points pondérés, qui peuvent être considérés comme des distributions de probabilités 3D, sont comparés en utilisant une nouvelle distance de transport qui est une variante de la distance de déplacement de la Terre (Earth Mover's Distance) [Rubner et al. 1998]. En revanche, les approches basées sur des histogrammes peuvent être considérées comme des méthodes comparant des distributions de probabilités 1D. Contrairement à la distance de déplacement, la distance de transport dans l'approche de Tangelder et al. satisfait l'inégalité triangulaire, et donc leur méthode peut être utilisée dans les schémas d'indexation qui utilisent cette propriété. Leurs expériences démontrent que la performance de récupération de leur méthode se compare favorablement à celle d'autres méthodes d'appariement de formes.

Pour comparer deux objets indépendamment de leur orientation, de leur position et de leur échelle, Tangelder et al. ont d'abord appliqué l'analyse des composantes principales pour amener les objets dans une pose standard définie par les principaux axes d'inertie. De plus, dans l'étape de prétraitement, ils entourent chaque objet par une grille 3D et génèrent pour chaque objet une signature représentant un ensemble de points pondérés, qui contient pour chaque cellule de grille non vide un point saillant. Ci-dessous, ils comparent trois méthodes pour obtenir dans chaque cellule de la grille un point saillant. Ces trois méthodes n'utilisent que les sommets et les facettes adjacentes aux sommets pour obtenir un point saillant. Elles peuvent donc traiter les modèles qui contiennent des lacunes. Notez que les modèles contenant des polygones mal orientés ne sont traités correctement que par la troisième méthode.

(1) Méthode basée sur la courbure gaussienne. Pour une surface lisse, la courbure gaussienne en un point est le produit de la courbure principale minimale et maximale en ce point. Le sommet de la cellule ayant la courbure gaussienne la plus élevée peut être choisi comme point saillant.

(2) Méthode basée sur la variation normale. Une autre approche pour obtenir une mesure liée à la courbure est la méthode de la variation normale. Dans cette approche, on estime la courbure dans une cellule de la grille par la variation normale dans la cellule de la grille. On choisit comme point saillant la moyenne des sommets de la cellule de la grille, pondérée par la surface.

(3) Méthode basée sur le point médian. Les deux méthodes décrites ci-dessus peuvent échouer si les modèles 3D contiennent des polygones mal orientés. C'est le cas des modèles qui sont représentés par des "soupes polygonales", c'est-à-dire des ensembles de polygones non organisés et dégénérés. Pour traiter de tels modèles dégénérés, on peut adopter une approche simple appelée méthode du point médian qui est similaire à l'algorithme de simplification des polygones de Rossignac [Rossignac et al. 1993]. La méthode du point médian permet d'obtenir une signature S en ajoutant pour chaque cellule de la grille le centre de masse de tous les sommets de la cellule ayant un poids unitaire à la signature S.

Enfin, ils calculent la similarité entre deux formes en comparant leurs signatures à l'aide d'une mesure de similarité de forme qui est une nouvelle variation de la distance du Earth Mover. Les résultats expérimentaux donnés par Tangelder et al. sont très prometteurs, mais leur principal défaut est le temps qu'il a fallu pour calculer les descripteurs.

II.4.4.3 Autres méthodes

Heczko et al. [Heczko et al. 2001] ont mis en œuvre une méthode basée sur la structure octrée pour représenter les caractéristiques de forme des modèles volumétriques 3D en réalisant une subdivision multirésolution de l'espace du modèle 3D. Pour chaque cellule de la grille, ils ont pris la somme des maillages délimités par la cellule de la grille comme composants des caractéristiques, qui ont formé un descripteur de caractéristiques de dimensions $2r \times 2r \times 2r$, où r est la résolution de la représentation en octree.

Comme pour les modèles solides industriels 3D, Cicirello et al. [Cicirello et al. 2001] et McWherter et al. [McWherter et al. 2001] ont tous deux comparé des formes 3D en extrayant les caractéristiques géométriques et techniques des modèles 3D dans des domaines spatiaux.

En ce qui concerne les modèles solides industriels 3D, Cicirello et al. et McWherter et al. ont tous deux comparé des formes 3D en extrayant les caractéristiques géométriques et techniques de modèles 3D dans des domaines spatiaux.

Afin d'améliorer les performances globales, la stratégie "diviser pour mieux régner" peut être adoptée dans le processus d'extraction des caractéristiques. Dans certains cas, la faible efficacité est principalement due au fait que certaines des représentations des caractéristiques ne peuvent pas être calculées directement à partir des maillages 3D, qui doivent d'abord être transformés en un espace de voxels 3D. Ce processus prend du temps et nécessite une grande quantité d'espace de stockage. Pour résoudre ce problème, Zhang et al [Zhang et al. 2001] ont proposé un algorithme d'analyse géométrique globale utilisant la stratégie "diviser pour mieux régner" sans transformation volumétrique. Ils ont d'abord calculé les caractéristiques de chaque surface élémentaire (un triangle ou un tétraèdre) d'un modèle de maillage 3D, puis les ont additionnées pour former le vecteur de caractéristiques global.

II.4.5 Extraction de caractéristiques basée sur l'analyse du signal

Les méthodes d'extraction de caractéristiques basées sur l'analyse du signal analysent les modèles 3D du point de vue du domaine fréquentiel. Cependant, comme le modèle 3D n'est pas un signal régulièrement échantillonné, le processus de prétraitement avant l'extraction de caractéristiques est généralement compliqué. Dans cette section, nous aimerions présenter trois descripteurs de forme typiques basés sur les domaines de transformation.

II.4.5.1 Descripteur de Fourier

Nous introduisons la transformée de Fourier discrète, le système de Vranić et Soupe et d'autres systèmes.

Transformation de Fourier discrète

En mathématiques, la transformée de Fourier discrète (TFD) est un type spécifique de transformée de Fourier, utilisée en analyse de Fourier. Elle transforme une fonction en une autre, ce qu'on appelle la représentation du domaine fréquentiel, ou simplement la TFD, de la fonction originale (qui est souvent une fonction dans le domaine temporel). Mais la TFD nécessite une fonction d'entrée qui est discrète et dont les valeurs non nulles ont une durée limitée (finie). Ces entrées sont souvent créées par l'échantillonnage d'une fonction continue, comme la voix d'une personne. Et contrairement à la transformée de Fourier en temps discret, elle n'évalue qu'un nombre suffisant de composantes de fréquence pour reconstruire le segment fini qui a été analysé. Sa transformation inverse ne peut pas reproduire l'ensemble du domaine temporel, à moins que l'entrée ne soit périodique (pour toujours). Par conséquent, on dit souvent que la TFD est une transformée pour l'analyse de Fourier des fonctions en temps discret dans le domaine fini. Les fonctions de base sinusoïdales de la décomposition ont les mêmes propriétés. Comme la fonction d'entrée est une séquence finie de nombres réels ou complexes, la TFD est idéale pour le traitement des

informations stockées dans les ordinateurs. En particulier, la TFD est largement utilisée dans le traitement du signal et les domaines connexes pour analyser les fréquences contenues dans un signal échantillonné, pour résoudre des équations différentielles partielles et pour effectuer d'autres opérations telles que des convolutions. En pratique, la TFD peut être calculée efficacement à l'aide d'un algorithme de transformée de Fourier rapide (TFR).

La séquence de N nombres complexes $x_0,\dots,x_{N-1}$ est transformée en la séquence de N nombres complexes $X_0,\dots,X_{N-1}$ par la TFD selon la formule :

$$X_k = \sum_{n=0}^{N-1} X_n e^{-\frac{2\pi j}{N}kn}, k = 0,\dots,N-1 \quad \text{(II-25)}$$

Où $e^{-\frac{2\pi j}{N}}$ est une racine $N^{ième}$ primitive de l'unité. La transformée de Fourier discrète inverse (TFDI) est donnée par :

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} x_k e^{\frac{2\pi j}{N}kn}, n = 0,\dots,N-1 \quad \text{(II-26)}$$

Système de Vranić et Soupe

Dans l'analyse d'un modèle 3D, le descripteur de Fourier décompose le modèle 3D en composantes de fréquence et extrait les caractéristiques des coefficients de TFD. Vranić et Soupe [Vranić et al. 2001] ont appliqué le TFD-3D pour extraire les caractéristiques. Les étapes comprennent la normalisation de la pose, la voxélisation et la TFD-3D. Après avoir trouvé la position canonique et l'orientation d'un modèle, l'extraction des caractéristiques est effectuée en deux étapes : (1) voxélisation en utilisant le cube de délimitation ; (2) application du TFD-3D.

Le cube délimitant d'un modèle 3D est défini comme étant le cube le plus serré dans le cadre de coordonnées canonique qui entoure le modèle, avec le centre dans l'origine et les bords parallèles aux axes de coordonnées. Après avoir déterminé le cube délimitant, la voxélisation est effectuée de la manière suivante : le cube délimitant est subdivisé en N^3 (N est une puissance de 2) cubes de taille égale et calcule la proportion de la surface totale du maillage à l'intérieur de chacun des nouveaux cubes (cellules). La cellule ayant la valeur attribuée est considérée comme le voxel à la position donnée. Évidemment, avec l'augmentation de N, la fraction de tous les voxels à l'intérieur du cube délimitant ayant des valeurs supérieures à zéro diminue. Par conséquent, une structure en octree est un moyen approprié de stocker un vecteur de caractéristiques basé sur les voxels. Ainsi, une représentation hiérarchique efficace des caractéristiques peut être obtenue.

Les informations contenues dans cet octree peuvent être utilisées de plusieurs manières. Vranić et Soupe anciennement dans [Heczko et al. 2001] ont utilisé une voxélisation similaire comme caractéristique dans le domaine spatial avec un N raisonnablement petit. Le vecteur de la caractéristique avait des composantes N^3et les normes L_1 ou L_2 étaient engagées pour le calcul des distances. Alors qu'en [Vranić et al. 2001], leur modification est la suivante : Une valeur supérieure de N est sélectionnée et l'élément est représenté dans le domaine des fréquences en appliquant la TFD-3D au modèle voxélisé (c'est-à-dire les valeurs calculées dans les cellules N^3).

Soit $Q = \{q_{ikl} | q_{ikl} \epsilon R, -\frac{N}{2} \le i,k,l < \frac{N}{2}\}$ l'ensemble de tous les voxels. L'ensemble Q est transformé en l'ensemble $G = \{g_{uvw} | g_{uvw} \epsilon C, -\frac{N}{2} \le u,v,w < \frac{N}{2}\}$ par :

$$g_{uvw} = \sum_{i=-N/2}^{N/2-1} \sum_{k=-N/2}^{N/2-1} \sum_{l=-N/2}^{N/2-1} q_{ikl} e^{\frac{2\pi j}{N}(iu+kv+lw)} \quad \text{(II-27)}$$

Enfin, on trouve les valeurs absolues des coefficients g_{uvw}avec les indices $-K \leq u, v, w \leq K$ (les fréquences les plus basses). À l'exception du coefficient g_{000}, tous les nombres complexes sélectionnés sont conjugués par paires. Par conséquent, le vecteur de caractéristique est constitué de $((2K+1)^3+1)/2$ composantes à valeur réelle. Dans les expériences de Vranić et de Soupe, ils sélectionnent $K = 1, 2, 3$, c'est-à-dire que les descripteurs possèdent respectivement 14, 63 et 172 composantes.

La valeur du paramètre N (la résolution de la voxélisation) doit être suffisamment grande pour que les propriétés spatiales d'un modèle soient saisies par la TFD-3D. En pratique, Vranić et Soupe ont sélectionné $N = 128$ et en moyenne environ 20000 voxels (sur 1283 éléments de l'ensemble Q) ont des valeurs supérieures à zéro. Cela rend la représentation de l'octree très efficace. Lors du TFD-3D, ils n'ont calculé que les éléments de l'ensemble G qui sont utilisés dans le vecteur de caractéristiques (14, 63, ou 172 sur 1283). Le descripteur proposé présente une meilleure performance de récupération que la caractéristique basée sur les voxels présentée dans [Heczko et al. 2001]. En gardant à l'esprit que le descripteur basé sur les rayons [Heczko et al. 2001] a été amélioré en incorporant des harmoniques sphériques [Vranić et al. 2001-b], ils ont déduit que si la norme L_1 ou L_2 est engagée, la représentation d'une caractéristique dans le domaine fréquentiel est plus efficace que la représentation de la même caractéristique dans le domaine spatial.

Autres descripteurs

Dans [Arbter et al. 1990], le descripteur de Fourier est étendu pour produire un ensemble de coefficients normalisés qui sont invariants sous toute transformation affine (translation, rotation, mise à l'échelle et cisaillement). La méthode est basée sur une description de frontière paramétrée qui est transformée dans le domaine de Fourier et normalisée pour éliminer les dépendances de la transformation affine et du point de départ. Richard et Hemani [Richard et al. 1974] ont utilisé le descripteur de Fourier pour calculer la courbure de frontière de l'objet 3D et obtenir sa caractéristique. Zhang et Fiume [Zhang et al. 2002] ont adopté le descripteur de Fourier pour décrire les contours 3D fermés. En outre, Sijbers et al. [Sijbers et al. 2002] ont proposé une méthode efficace pour calculer le descripteur de Fourier 3D.

II.4.5.2 Analyse sphérique harmonique

Vranić [Vranić et al. 2001] a introduit pour la première fois l'analyse harmonique dans le domaine de l'extraction des caractéristiques d'un objet 3D, qui est un descripteur de caractéristique pertinent pour la rotation. Kazhdan et al [Kazhdan et al. 2003] ont amélioré ce schéma, le rendant non pertinent en termes de rotation. L'idée clé de cette approche est de décrire une fonction sphérique en termes de quantité d'énergie qu'elle contient à différentes fréquences. Comme ces valeurs ne changent pas lorsque la fonction est tournée, le descripteur résultant est invariant par rapport à la rotation. Cette approche peut être considérée comme une généralisation de la méthode du descripteur de Fourier au cas des fonctions sphériques. La procédure détaillée peut être décrite comme suit.

Harmoniques sphériques

En mathématiques, les harmoniques sphériques sont la partie angulaire d'un ensemble de solutions à l'équation de Laplace. Représentées dans un système de coordonnées sphériques, les harmoniques sphériques de Laplace sont un ensemble spécifique d'harmoniques sphériques qui forment un système orthogonal, introduit pour la première fois par Laplace. Les harmoniques sphériques sont importantes dans de nombreuses applications théoriques et pratiques, notamment dans le calcul des

configurations électroniques des orbites atomiques, la représentation des champs gravitationnels, des géoïdes et des champs magnétiques des corps planétaires et des étoiles, et la caractérisation du rayonnement de fond des micro-ondes cosmiques. Dans l'infographie 3D, les harmoniques sphériques jouent un rôle particulier dans un grand nombre de sujets, notamment l'éclairage indirect (occlusion ambiante, éclairage global, transfert de radiance pré-calculé, etc.) et dans la reconnaissance des formes 3D.

Afin de représenter une fonction sur une sphère de manière invariante à la rotation, Kazhdan et al [Kazhdan et al. 2003] ont utilisé la notion mathématique d'harmoniques sphériques pour décrire la manière dont les rotations agissent sur une fonction sphérique. La théorie des harmoniques sphériques dit que toute fonction sphérique $f(\theta, \phi)$ peut être décomposée comme la somme de ses harmoniques :

$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_{lm} Y_l^m(\theta,\phi) \qquad \textit{(II-28)}$$

Les harmoniques sont visualisées sur la figure II-16. La propriété clé de cette décomposition est que si nous la limitons à une certaine fréquence l, et définissons le sous-espace des fonctions :

$$V_l = Span(Y_l^{-l}, Y_l^{-l+1}, \ldots., Y_l^{l-1}, Y_l^l \qquad \text{(II-29)}$$

Nous avons alors les deux propriétés suivantes : (1) V_l est une représentation pour le groupe de rotation : Pour toute fonction $f \in V_l$ et toute rotation R, nous avons $R(f) \in V_l$. Cela peut également être exprimé de la manière suivante : si π_l est la projection sur le sous-espace V_l, alors π_l commute avec les rotations :

$$\pi_l\big(R(f)\big) = R(\pi_l(f)) \qquad \text{(II-30)}$$

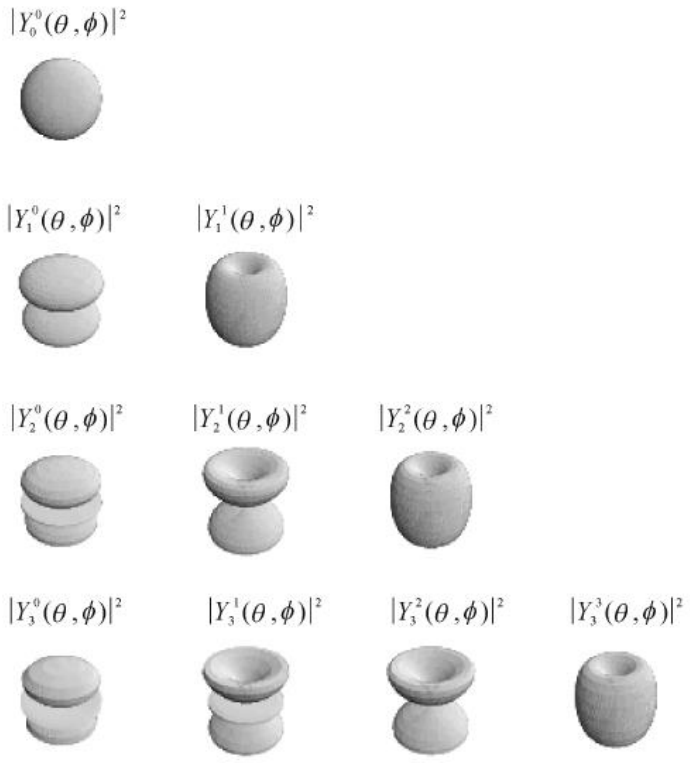


Figure II-16: Harmoniques sphériques

(2) V_l est irréductible : V_l ne peut pas être décomposé davantage comme la somme directe $V_l = V_l' \oplus V_l''$, où V_l' et V_l'' sont également des représentations (non négligeables) du groupe de rotation.

La première propriété présente une façon de décomposer les fonctions sphériques en composantes invariantes de rotation, tandis que la seconde propriété garantit que, dans un sens linéaire, cette décomposition est optimale.

Descripteurs invariants en rotation

En utilisant les propriétés des harmoniques sphériques et en observant que la rotation d'une fonction sphérique ne change pas sa norme $L2$, nous représentons les énergies d'une fonction sphérique $f(\theta, \phi)$ comme :

$$SH(f) = \{||f_0(\theta, \phi)||, ||f_1(\theta, \phi)||, \dots\} \quad \textit{(II-31)}$$

Où f_l est la composante de fréquence de f :

$$f_l(\theta, \phi) = \pi_l(f) = \sum_{m=-l}^{l} a_{lm} Y_l^m(\theta, \phi) \quad \textit{(II-32)}$$

Cette représentation a la propriété d'être indépendante de l'orientation de la fonction sphérique. Pour voir cette propriété, nous laissons R être n'importe quelle rotation et nous avons :

$$\begin{aligned} SH(R(f)) &= \left\{\left|\left|\pi_0(R(f))\right|\right|, \left|\left|\pi_1(R(f))\right|\right|, \dots\right\} \\ &= \left\{\left|\left|R(\pi_0(f))\right|\right|, \left|\left|R(\pi_1(f))\right|\right|, \dots\right\} \\ &= \{||\pi_0(f)||, ||\pi_1(f)||, \dots\} = SH(f) \end{aligned} \quad \text{(II-33)}$$

De sorte que l'application d'une rotation à une fonction sphérique f ne modifie pas sa représentation énergétique.

II.4.6 Extraction de caractéristiques basées sur des images 2D

Les méthodes basées sur l'image visuelle établissent une correspondance fonctionnelle entre l'objet 3D d'origine et un domaine prédéfini, généralement plusieurs vues planes 2D représentatives de dimensions réduites. Cette méthode a longtemps été étudiée dans les communautés de conception technique et de CAO en 3D, et est devenue l'un des moyens les plus populaires pour extraire les signatures de formes 3D. Les projections d'un objet 3D dans toutes les directions de visualisation sont importantes dans l'analyse 3D. Les méthodes d'extraction de caractéristiques basées sur l'image visuelle transforment les problèmes 3D compliqués en techniques de traitement d'images relativement matures pour réduire la difficulté. En même temps, ce type de méthode est en accord avec le système visuel humain, ce qui fait que les performances d'extraction sont meilleures que celles des autres types de méthodes. Cependant, pour tout objet 3D, il est nécessaire d'extraire des caractéristiques de plusieurs images 2D, ce qui nécessite beaucoup d'espace de stockage et de temps d'exécution, et donc une efficacité de récupération moindre. Actuellement, de nombreuses méthodes d'extraction de caractéristiques d'objets 3D basées sur des projections ont été proposées dans la littérature, où plusieurs projections 2D d'un modèle 3D ou des vues planes 2D de différentes perspectives sont générées et combinées en tant que descripteurs de caractéristiques de formes ou de silhouettes. Dans cette section, nous les présenterons dans les deux catégories suivantes.

II.4.6.1 Méthodes basées sur la projection fonctionnelle en 2D

La projection fonctionnelle 2D réduit le problème de correspondance 3D en un cas 2D sans avoir à calculer de multiples vues de l'objet. Voici quelques méthodes typiques dans cette catégorie.

Spin images

Johnson et al [Johnson et al. 1999] ont proposé une représentation d'image de spin, c'est-à-dire une image descriptive 2D associée à un sommet d'échantillonnage placé sur une surface 3D, pour laquelle les informations de position et de direction sont toutes deux impliquées. Les valeurs des coordonnées x et y de l'image de spin 2D sont définies comme les valeurs cumulées de deux fonctions de distance différentes des sommets 3D, et le coefficient de corrélation entre deux images de spin est calculé comme mesure de similarité. Cependant, étant donné qu'un objet 3D est généralement constitué de nombreuses surfaces, un grand nombre d'images de spin sont générées pour chaque modèle 3D. Pour obtenir une représentation plus concise et plus compacte des caractéristiques, l'ensemble original d'images de spin est compressé par la méthode ACP.

Image de coupe 2D

Pu et al [Pu et al. 2004] ont présenté une approche basée sur des images de coupe 2D pour mesurer les similitudes entre les objets 3D. L'idée clé est de représenter l'objet 3D par une série de coupes 2D, comme le montre la figure II-17, dans certaines directions, de sorte que le problème de correspondance des formes entre les modèles 3D soit transformé en mesure de similarité entre les coupes 2D. Cependant, les trois problèmes suivants doivent être résolus : sélection des directions de coupe, méthodes de coupe et mesure de similarité. Pour résoudre ces problèmes, certaines stratégies et règles sont proposées dans [Pu et al. 2004]. Tout d'abord, une méthode de distribution normale maximale est présentée pour obtenir trois orthoaxes qui coïncident mieux avec le mécanisme de perception visuelle humaine. Deuxièmement, une méthode de coupe est présentée qui peut être utilisée pour obtenir une série de coupes composées d'un ensemble de polygones fermés. Troisièmement, une méthode de distribution de formes 2D est développée pour mesurer la similarité entre les coupes 2D. Ce schéma découle d'un fait tel que décrit dans la figure II-17 : puisque les objets 3D peuvent être coupés en une série de coupes aux contours polygonaux, pourquoi pas l'inverse? La procédure inverse pourrait-elle être utilisée pour effectuer la correspondance des formes? Dans cette figure, la forme du milieu est constituée de 33 images de coupe 2D, tandis que celle de droite est constituée de 100 images de coupe 2D. Il est montré qu'un objet 3D peut être reconstruit avec précision en superposant une série d'images de coupe 2D qui représentent le contour local d'un modèle 3D. Plus le nombre d'images de coupe 2D est important, plus l'objet 3D final sera précis. Pour le processus détaillé, le lecteur peut se référer à [Pu et al. 2004].

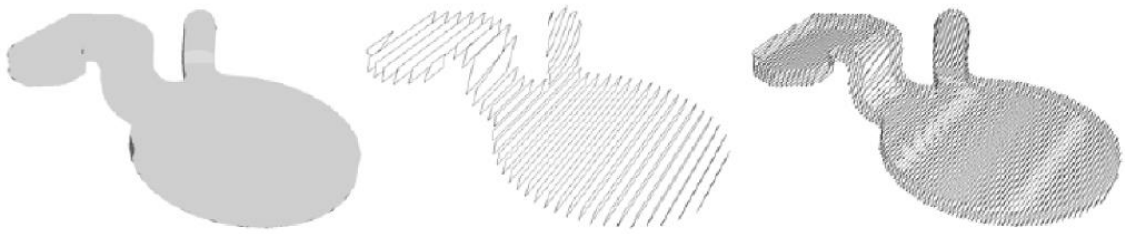


Figure II-17: Représentation de la forme en images de coupe 2D, où la forme de droite est reconstruite avec plus d'images de coupe que celle du milieu [Pu et al. 2004].

II.4.6.2 Méthodes basées sur la représentation en vue plane 2D

Par rapport aux méthodes de la sous-section précédente, les méthodes de mappage 2D qui établissent des mappages à partir d'une vue 3D vers un ensemble de vues planes 2D spécifiques sous différents angles sont beaucoup plus naturelles et simples. L'idée de base est que si deux formes 3D sont similaires, elles doivent l'être à partir de nombreuses vues différentes. Ainsi, des formes en 2D, telles que des silhouettes en 2D, peuvent être extraites et adoptées pour la correspondance de formes en 3d. Il existe une vaste littérature sur ces techniques particulières.

Informations sur les contours 2D

Vranić [Vranić 2004] a présenté une représentation des objets basée sur des informations de contours 2D, après avoir projeté l'objet 3D sur trois plans de coordonnées standard, c'est-à-dire les plans XY, XZ et YZ. Pour chaque projection sur un plan donné, une silhouette est acquise en sélectionnant des points de contour, équidistants ou équiangulaires ; puis le spectre de puissance de Fourier est calculé. Les n premiers coefficients du spectre de puissance sont finalement extraits en tant que caractéristique. L'inconvénient est l'incapacité à refléter correctement les informations spatiales 3D, car l'objet 3D n'est considéré que comme une simple combinaison de trois projections 2D standard, ce qui entraîne une perte trop importante d'informations sur la structure. Pour résoudre ce problème, Vranić a ajouté des informations de profondeur, qui codaient la différence de distance spatiale des surfaces 3D en différentes valeurs de gris de leurs images de projection 2D [Vranić 2004]. Ils ont également remplacé la correspondance de forme 2D basée sur les contours par une correspondance basée sur les régions, ce qui a également augmenté la précision de l'extraction.

Aspect Graph

Cyr et al [Cyr et al. 2001] ont proposé une approche, aspect-graph, pour représenter les formes 3D, comme le montre la figure II-18. Tout d'abord, les vues de projection 2D sont calculées en fonction des angles de vue obtenus après avoir divisé la sphère de vision tous les 5°. Ensuite, les vues de projection 2D similaires sont regroupées dans le même groupe de manière à générer un certain nombre de groupes appelés "aspect", à partir desquels la représentation de la forme est créée en sélectionnant une vue représentative pour chaque "aspect".

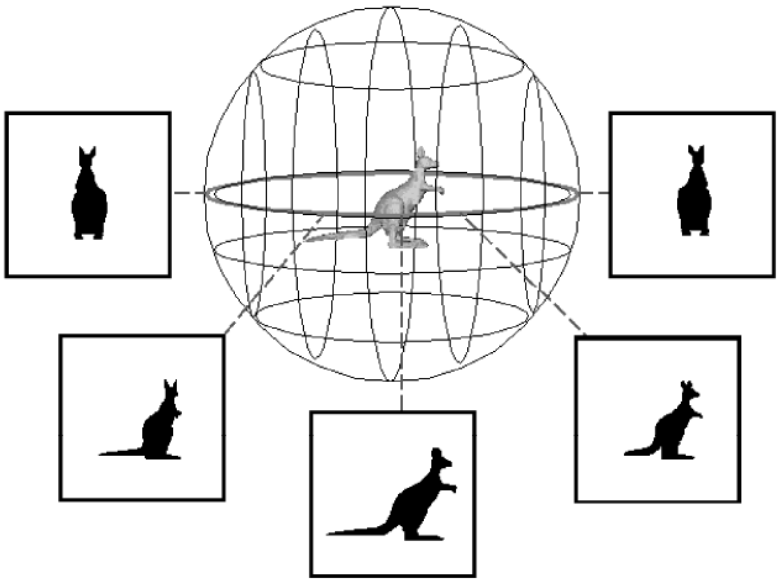


Figure II-18: Aspect-graph [Cyr et al. 2001]

De même, Min et al. [Min et al. 2003] ont projeté chaque objet 3D en plusieurs images de silhouettes 2D à partir de m points de vue différents, puis ont fait correspondre toutes leurs combinaisons avec $n\,(m > n)$ croquis 2D dessinés par l'utilisateur ou les combinaisons correspondantes d'autres modèles 3D. La similarité est mesurée comme la somme minimale de tous les scores similaires de croquis à l'image (ou d'image à image) par paire.

Descripteur Light Field

Chen et al. [Chen et al. 2003] ont proposé un descripteur représentant le champ lumineux 4D d'un objet 3D avec une collection d'images 2D, qui sont capturées par un ensemble de caméras uniformément réparties en utilisant le concept de "Light Field" du rendering basé sur l'image. Les caméras sont commandées pour tourner plusieurs fois lorsqu'elles mesurent la similitude entre les descripteurs de deux objets 3D, comme le montre la figure II-19, de manière à être commutées sur leurs différents sommets. Les résultats finaux de l'extraction des objets 3D sont combinés à partir des résultats correspondants de toutes les images 2D acquises en intégrant les descripteurs 2D du moment de Zernike et de Fourier.

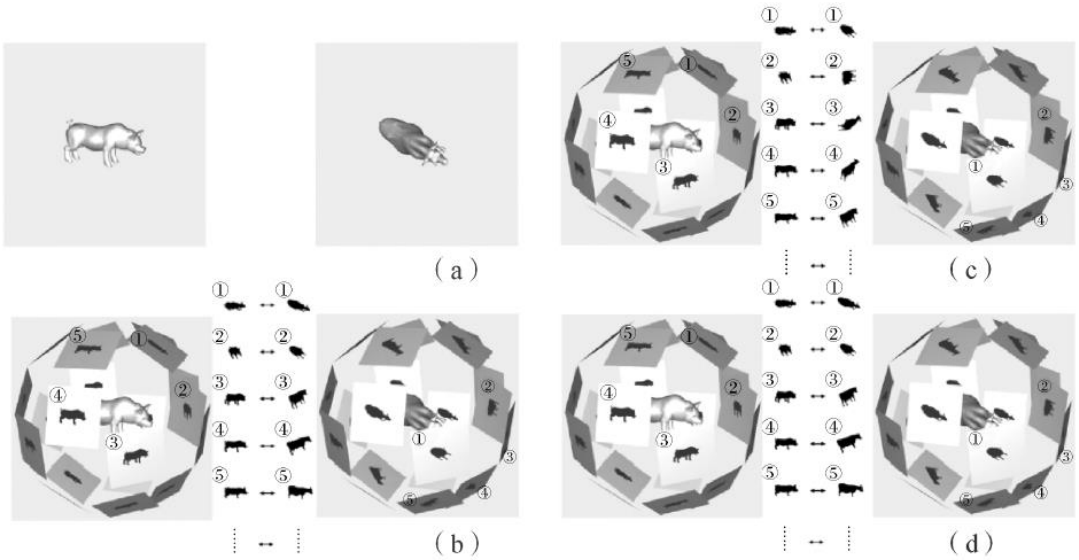


Figure II-19 : (a)-(d) montrant la rotation et la comparaison d'un ensemble de vues représentant deux objets 3D [Chen et al. 2003]

Image de profondeur

Ohbuchi et al [Ohbuchi et al. 2003] ont présenté une méthode similaire. Ils ont généré une image de profondeur ou de valeur z d'un objet 3D à partir de plusieurs points de vue qui sont également espacés sur la sphère unitaire. L'appariement des objets 3D est ensuite effectué en adoptant un descripteur de Fourier 2D pour l'appariement par similarité des images 2D. La principale différence est que l'image 2D de Chen ne contient que des silhouettes alors que celle d'Ohbuchi contient des informations sur la profondeur. La figure II-20 illustre le processus d'extraction des caractéristiques d'Ohbuchi. L'image en profondeur est d'abord mappée de la coordonnée cartésienne à la coordonnée polaire pour effectuer une transformation de Fourier avant que les descripteurs de Fourier ne soient calculés.

Comme il est possible d'extraire beaucoup plus de caractéristiques pour une forme 2D, les méthodes de mappage des fonctions rendent le processus d'extraction plus souple. Elles peuvent également réduire la complexité du calcul des caractéristiques et rendre le descripteur des caractéristiques plus compact. Toutefois, cela entraîne inévitablement une perte importante d'informations 3D, car le processus de mappage des fonctions est limité par

différentes contraintes. De plus, pour la représentation de vues planes en 2D, la manière de décider le nombre nécessaire de vues de projection 2D est un autre problème dans la pratique [Cyr et al. 2001].

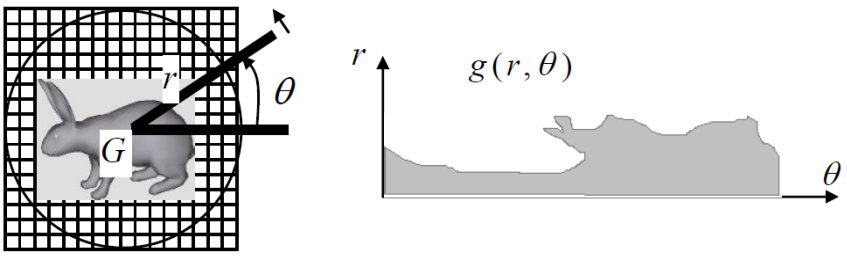


Figure II-20: Image de profondeur

II.4.7 **Extraction de caractéristiques basées sur la topologie**

La topologie est une représentation relativement de haut niveau. Elle décrit l'organisation et le rangement spatial des informations : comment les sommets sont reliés pour composer des surfaces avec des arêtes. Une structure de données de graphe et un algorithme de graphe bien conçus peuvent être adoptés pour représenter la topologie et le squelette caractéristiques des objets 3D. Par conséquent, ce type de méthode produit généralement une structure de type graphique, plutôt que des descripteurs de caractéristiques numériques.

Bardinet et al [Bardinet et al. 2000] ont présenté une représentation structurée de la forme 3D basée sur un squelette 3D et des axes médians, comme extension du concept de transformation 2D de l'axe médian [Blum 1973]. Tout d'abord, des graphes relationnels attribués adéquats, consistant en un ensemble de nœuds avec des attributs et un ensemble de liens, sont générés et les caractéristiques topologiques sont ensuite extraites des structures de nœuds et de liens de ces graphes. Hilaga et al [Hilaga et al. 2001] ont représenté la topologie d'un objet 3D sous la forme d'un graphe de Reeb en utilisant une fonction de "distance géodésique" [Sharir et al. 1986] entre les points du maillage. Le graphique de Reeb est une représentation squelettique utilisant une fonction scalaire continue définie sur un objet de dimensions arbitraires [Reeb 1946].

L'analyse topologique peut également être effectuée en décomposant un objet 3D en un modèle paramétrique d'un ensemble de formes régulières élémentaires simples. La topologie est représentée par les relations et les dispositions spatiales de ces formes de base, telles que les cylindres généralisés [Binford 1971], les régions déformables [Basri et al. 1998], l'échafaudage de choc [Leymarie et al. 2001] et les superquadrilles [Zhang et al. 2003]. Ma et al [Ma et al. 2003] ont même présenté une approche pratique, utilisant un objet basé sur des fonctions de base radiales (Radial Basis Functions (RBF)) pour extraire des squelettes en 3D. Pour un objet polygonal 3D, les sommets sont traités comme des centres pour la construction d'ensembles de niveaux RBF et un algorithme de descente de gradient est employé sur chaque sommet pour localiser les maxima locaux dans la RBF. Enfin, toutes les paires de maxima connectées sont traitées à l'aide de la méthode Snake et les positions finales des séquences Snake sont extraites en tant que caractéristiques du squelette. Tal et al [Tal et al. 2006] ont d'abord décomposé un maillage en éléments appelés "bassins versants" en utilisant un algorithme de décomposition des bassins versants [Serra 1982], puis les ont ajustés et classés en quatre types de formes de base : surfaces sphériques, surfaces cylindriques, surfaces coniques et surfaces planes. Ensuite, la signature de la forme, un graphique de décomposition attribué, est construite.

Les caractéristiques topologiques et de forme squelettique sont intéressantes pour la recherche en 3D car elles permettent de capturer les structures de forme significatives d'un objet 3D. En même temps, elles sont relativement élevées et proches de la perception intuitive de l'homme, ce qui les rend utiles pour définir une représentation plus naturelle des requêtes 3D. Ils peuvent également effectuer des tâches de correspondance partielle pour contenir des propriétés structurelles locales et globales. Cependant, les objets 3D ne sont pas toujours suffisamment bien définis pour être facilement et naturellement décomposés en un ensemble canonique de caractéristiques ou de formes de base. En outre, le processus de décomposition est généralement coûteux en termes de calcul. En outre, les processus de décomposition des objets sont assez sensibles au bruit pour les petites perturbations du modèle. Ainsi, un effort supplémentaire est, à son tour, nécessaire pour les gérer. Enfin, par rapport aux algorithmes d'indexation et de recherche des similarités relativement simples basés sur des vecteurs de caractéristiques numériques, les algorithmes d'indexation et de recherche des représentations sous forme de graphes sont relativement complexes et longs, en raison des processus de recherche de graphes nécessaires. Et, comme il n'existe actuellement aucune solution universelle de correspondance de graphes à usage général, différents algorithmes de correspondance de graphes doivent être conçus pour s'adapter à différentes représentations de graphes.

II.4.8 Extraction de caractéristiques basées sur l'apprentissage approfondi

Au cours de la dernière décennie, divers groupes de recherche ont fait des progrès considérables dans la mise au point d'approches d'apprentissage qui permettent l'extraction des caractéristiques d'objet 3D. Le fait de disposer de différentes représentations de données 3D a conduit les chercheurs à suivre différentes voies d'apprentissage approfondi pour adapter le processus d'apprentissage aux propriétés des données. Dans cette section, nous donnons un aperçu des différents paradigmes de DL appliqués aux différentes représentations de données 3D.

II.4.8.1 Architectures d'apprentissage approfondi sur les descripteurs de données 3D.

Les descripteurs de bas niveau ont été utilisés comme une partie importante du processus d'apprentissage des données 3D. Bien que de nombreux descripteurs de bas niveau aient été proposés dans la littérature, tels que [Sun et al. 2009] et [Liu et al. 2014], ils souffrent de plusieurs limitations importantes, car ils ne peuvent pas apprendre les caractéristiques distinctives des formes 3D. Par conséquent, la structure globale et locale des formes 3D ne peut pas être préservée. Heureusement, les modèles DL sont efficaces pour apprendre les caractéristiques discriminantes hiérarchiques qui peuvent être généralisées à d'autres données invisibles. C'est pourquoi des descripteurs de bas niveau ont été combinés avec les architectures DL pour apprendre des caractéristiques de haut niveau plus informatives de l'objet 3D. Cette pratique courante a été suivie par Liu et al [Liu et al. 2013] dans le but d'apprendre des caractéristiques de haut niveau afin de les utiliser pour des tâches de classification et de recherche. Les "Bag-of-Words (BoWs)" visuels encodent les représentations des caractéristiques de bas niveau, qui sont ensuite introduites dans les "Deep Belief Networks (DBNs)" pour l'apprentissage des caractéristiques sémantiques de haut niveau de l'entrée. Les résultats des expériences de recherche et de classification en 3D ont montré que les caractéristiques apprises sont discriminatoires à l'égard des variations entre classes, ce qui permet d'obtenir de meilleurs résultats que les caractéristiques de bas niveau classiques des BoW. Dans [Bu et al. 2014], Bu et al. ont proposé un pipeline en trois étapes pour apprendre les propriétés géométriques des formes 3D. Leur idée principale était d'utiliser les caractéristiques de bas niveau extraites pour construire des caractéristiques géométriques de niveau moyen indépendantes de la position sur lesquelles un modèle DL peut être utilisé pour apprendre les caractéristiques hiérarchiques de haut niveau de la forme 3D. Dans cette méthode, la « Scale-Invariant Heat Kernel Signature (SI-HKS) » [Bronstein et al. 2010] et la « Average-Geodesic Distance (AGD) » ont été utilisées pour apprendre les caractéristiques de bas niveau. Ensuite, le « Spatially Sensitive Bag-of-Features (SS-BoF) »,

sensible à l'espace, a été utilisé pour apprendre les mots spatialement proches et la relation entre eux. Enfin, les DBN ont été utilisés sur les SS-BoW pour apprendre des fonctionnalités de haut niveau. Les expériences sur la recherche et la reconnaissance 3D ont montré des améliorations significatives par rapport à l'utilisation de descripteurs de bas niveau uniquement. Dans [Bu et al. 2015], Bu et al. ont étendu leurs travaux dans [Bu et al. 2014] à une implémentation basée sur le GPU pour accélérer les calculs et l'ont utilisé pour les tâches de détection de correspondance et de symétrie où le modèle proposé s'est avéré plus performant.

Motivés par les performances du HKS en matière d'extraction de caractéristiques de bas niveau, Xie et al. [Xie et al. 2015] ont utilisé le HKS comme descripteur de bas niveau à plusieurs échelles. Le résultat a été utilisé par les « Auto-Encoders, AE » pour apprendre les caractéristiques discriminantes pour la recherche 3D. La "Fisher Discriminative Analysis (FDA)" a également été utilisée pour améliorer les représentations des caractéristiques. Les expériences ont prouvé la robustesse de ce modèle contre les déformations. Dans [Han et al. 2017], Han et al. ont proposé des "Machines Boltzmann à Maillage Convolutionnel Restreint (MCRBM)" pour apprendre les caractéristiques discriminantes hiérarchiques des maillages 3D. Le modèle proposé a permis d'apprendre les caractéristiques globales et locales des objets 3D. La structure des caractéristiques locales a été préservée en utilisant la "Local Function Energy Distribution (LFED)". Une extension, un modèle plus profond composé de plusieurs MCRBM empilés, a été testée dans le contexte de la recherche de formes et de la correspondance. Ce modèle a surpassé les techniques de référence actuelles telles que [Wu et al. 2015] et [Bronstein et al. 2010].

La plupart des modèles de DL utilisés dans les méthodes précédentes appartiennent à la catégorie des méthodes d'apprentissage non supervisées, car les méthodes supervisées ont tendance à apprendre des abstractions hiérarchiques sur les données brutes. Cependant, présenter des données 3D avec des descripteurs est en effet une forme d'abstraction. C'est pourquoi les méthodes supervisées peuvent ne pas produire de caractéristiques informatives, car elles apprennent des abstractions qui peuvent conduire à une perte des propriétés réelles des formes si la représentation des descripteurs est très simple/abstraite. C'est pourquoi les méthodes non supervisées sont plus adaptées à une telle représentation pour apprendre les motifs cachés ou le regroupement dans les données d'entrée. Cependant, dans certains cas, les descripteurs peuvent fournir de riches informations sur lesquelles l'opération de convolution peut être efficace pour apprendre les caractéristiques hiérarchiques des représentations d'entrée telles que [Han et al. 2017]. Ces méthodes peuvent toujours être combinées avec des modèles non supervisés. En bref, le choix du modèle DL sur les représentations des descripteurs dépend de la richesse du descripteur.

II.4.8.2 Architectures d'apprentissage approfondi basées sur des projections d'objets 3D

L'une des premières tentatives d'apprentissage des caractéristiques des données 3D en les projetant dans des plans 2D a été présentée par Zhu et al. dans [Zhu et al. 2016]. Le pipeline proposé a commencé par un prétraitement des données où la translation, la mise à l'échelle et la normalisation de la pose ont été appliquées sur chaque modèle 3D. Ensuite, diverses projections 2D ont été appliquées sur chaque modèle 3D traité pour l'alimenter en une pile de RBM afin d'extraire les caractéristiques des différentes projections. Afin d'apprendre une représentation globale des objets 3D à utiliser pour la tâche de recherche, un AE a été utilisé. Les expériences ont montré que ce système était plus performant que les techniques basées sur des descripteurs globaux. Les performances ont été améliorées en combinant les représentations locales avec les représentations globales apprises. Dans ce contexte, Shi et al [Shi et al. 2015] ont proposé DeepPano. DeepPano se réfère à l'extraction de vues panoramiques 2D à partir d'objets 3D en utilisant une projection cylindrique autour de l'axe principal de l'objet 3D. L'architecture classique 2D de CNN a été utilisée pour entraîner le modèle. Afin d'obtenir une invariance de rotation autour de l'axe principal, une couche de " max-pooling " par rangée a été utilisée entre la couche " Convolution (Conv) "

et la couche "Fully Connected (FC)". Le réseau proposé était composé de quatre couches Conv, une couche max-pooling par ligne, deux couches FC et une couche softmax insérée à la toute fin du réseau. Le modèle proposé a été testé sur des tâches de reconnaissance et de recherche d'objets 3D où il a prouvé son efficacité par rapport aux modèles précédents. Sinha et al. dans [Sinha et al. 2016] ont proposé des images de géométrie où les objets 3D étaient projetés dans une grille 2D afin que les CNN 2D classiques puissent être utilisés. La méthode proposée a créé un paramétrage de planificateur pour les objets 3D en utilisant un paramétrage authentique (conservation de zone) sur un domaine sphérique pour apprendre les surfaces des formes 3D. Ensuite, les images de géométrie construite ont servi d'entrées à l'architecture classique des CNN pour apprendre les caractéristiques géométriques des objets 3D. Comme étape de prétraitement, des opérations d'augmentation, de mise à l'échelle, de rotation et de translation des données ont été effectuées pour augmenter la taille des données d'apprentissage et pour fournir une certaine variété. Les images géométriques ont été testées sur différents ensembles de données : ModelNet10, ModelNet40, SHREC1, SHREC2, McGill11 et McGill2 pour les tâches de classification et de recherche. Les résultats ont montré que les images géométriques peuvent produire des résultats comparables par rapport aux méthodes les plus récentes [Shi et al. 2015] [Su et al. 2015] [Wu et al. 2015].

Motivés par les résultats obtenus par les méthodes de projection, Cao et al. [Cao et al. 2017] ont proposé d'utiliser une projection à domaine sphérique pour projeter des objets 3D autour de leur barycentre, produisant un ensemble de patchs cylindriques. Le modèle proposé utilise les patchs cylindriques projetés comme entrée pour des CNN pré-entraînés. Deux projections complémentaires ont également été utilisées pour mieux saisir les caractéristiques 3D. La première projection complémentaire saisit les variations de profondeur des formes 3D tandis que la seconde apprend les informations de contour intégrées dans différentes projections sous différents angles. Le modèle proposé a été utilisé pour la tâche de classification des objets 3D où il a été testé sur plusieurs ensembles de données produisant des résultats comparables aux méthodes précédentes. Comme pour les travaux précédents, Sfikas et al. [Sfikas et al. 2017] ont proposé de représenter les objets 3D sous forme de vues panoramiques extraites d'objets 3D normalisés. Dans un premier temps, les objets 3D sont prétraités pour normaliser leurs poses en utilisant le système "Pose Normalization of 3D Models via Reflective Symmetry on Panoramic Views (SymPan)" [Sfikas et al. 2018]. Les vues panoramiques sont ensuite extraites pour être combinées et transmises à CNN pour effectuer des tâches de classification et de recherche. Tout en étant similaire aux modèles précédents, cette méthode a amélioré la précision lors des tests sur les ensembles de données ModelNet10 et ModelNet40. Une extension de ce modèle a été introduite dans [Sfikas et al. 2018] où un ensemble de CNN a été utilisé pour le processus d'apprentissage. Cette extension a produit des résultats très élevés lors des tests sur les ensembles de données susmentionnés.

Les représentations par projection sont simples mais efficaces pour l'apprentissage d'objets 3D à l'aide des méthodes de DL 3D. Les propriétés géométriques de la forme sont perdues à cause des projections ; c'est pourquoi, dans les travaux précédents, les chercheurs ont essayé de combiner plus d'une projection pour compenser les informations manquantes. Bien que les modèles DL 2D puissent être directement appliqués sur cette représentation, les réseaux nécessitent généralement un réglage plus fin que d'opérer directement sur la représentation de données 3D brutes, qui pourrait être adaptée aux données d'apprentissage de manière spécifique et provoquer une pénurie lorsqu'elle est testée sur de nouvelles données non vues.

II.4.8.3 Architectures d'apprentissage approfondi basées sur les données RGBD

En raison de la disponibilité des données des capteurs RGB-D, de nombreux efforts de recherche ont été consacrés à l'exploitation des données disponibles afin de les utiliser pour plusieurs tâches. Une des premières approches dans cette direction a été proposée par Socher et al. dans [Socher et al. 2012], où les auteurs ont présenté un pipeline de convolution et de réseaux neuronaux récursifs pour traiter à la fois les images couleur et la

profondeur des données RGB-D. Deux CNN monocouches ont été utilisés pour apprendre les représentations des caractéristiques de l'entrée RGB-D. Le descripteur résultant a été transmis à plusieurs « Recurrent Neural Network, RNN » avec des poids aléatoires. Ensuite, les résultats des RNN ont été combinés et fusionnés pour être utilisés comme entrée dans un classificateur softmax. Ce système a été utilisé pour la classification des objets ménagers, où il s'est avéré très performant. Les chercheurs ont continué à utiliser la puissance des RNN pour apprendre les caractéristiques des données RGB-D. Dans [Couprie et al. 2013], Couprie et al. ont proposé une segmentation sémantique multi-échelle des scènes RGB-D d'intérieur des CNN. Le réseau proposé a appris les données RGB-D à plusieurs échelles (trois échelles différentes). Le réseau est principalement composé de deux CNN parallèles où le premier CNN est responsable de la classification des objets dans la scène et les résultats du second CNN sont transmis à un classificateur pour calculer le score de prédiction de l'étiquette de classe. Le label de classe final a été décidé en utilisant les résultats du classificateur ainsi que les super-pixels segmentés de la scène. Dans le cadre du prétraitement de ce modèle, les canaux des images RGB-D sont normalisés à une moyenne nulle et les informations de profondeur sont ajoutées en tant que quatrième pixel aux images RGB et introduites dans les CNN. Cette méthode a produit une meilleure précision de 6 % que les modèles précédents et est très efficace sur le plan du calcul. L'avantage de cette méthode est que, malgré la simplicité d'utilisation des CNN combinée à la profondeur, le résultat est bien meilleur par rapport aux caractéristiques obtenues à la main. De plus, cette méthode démontre l'importance considérable des informations sur la profondeur pour l'application de la segmentation, car il s'agit d'un facteur clé pour la séparation entre les objets. Cependant, le CNN semble ici n'apprendre que l'objet de classe sans apprendre la géométrie de la forme.

Inspirés par les résultats du réseau à deux flux proposé dans [Couprie et al. 2013], les chercheurs ont commencé à exploiter le même concept en y apportant quelques modifications inédites. Au lieu d'utiliser les réseaux pour deux tâches d'apprentissage différentes (classification et segmentation), les chercheurs ont commencé à traiter séparément les informations de profondeur et les informations de couleur en utilisant un réseau différent, ce qui a commencé à être une pratique courante dans le traitement des données RGB-D. Dans [Eitel et al. 2015], Eitel et al. ont proposé d'utiliser un réseau CNN à deux flux sur les données RGB-D pour la reconnaissance d'objets 3D, comme le montre la figure II-21. Un flux CNN traite les informations de couleur RGB et l'autre flux sert à traiter la profondeur. Chacun des deux flux CNN possède cinq couches Conv et deux couches FC. Chaque réseau a été formé séparément, puis les résultats ont été fusionnés dans les couches FC et softmax pour décider de la classe de l'objet. Cette méthode a surpassé les précédentes méthodes existantes et a démontré une performance prometteuse pour la reconnaissance d'objets dans des environnements bruyants du monde réel. Feng et al. [Feng et al. 2016] ont proposé un ensemble d'AE utilisant un seul modèle RGB-D pour la recherche d'objets 3D. Chaque AE a été formé à l'aide de l'algorithme « Stochastic Gradient Descent, SGD » sur un ensemble de données de différents modèles d'objets CAO. En raison de la différence entre les données d'entraînement et les données de test, les scores de sortie des AE ont ensuite été transmis à ce qu'ils ont appelé la couche d'adaptation de domaine (Domain Adaption Layer, DAL) pour classer les scores récupérés. Cette méthode a permis d'améliorer les performances par rapport à d'autres méthodes connexes.

Alexandre [Alexandre 2016] a combiné le concept d'apprentissage par transfert et les CNNs pour entraîner quatre CNNs indépendamment. Dans ce modèle, chaque canal des quatre canaux dans les données RGB-D était traité en utilisant un CNN séparé et les poids étaient transférés de chaque réseau à un autre. Les expériences ont montré que cette méthode a permis d'augmenter les performances, ce qui implique que les informations de profondeur contiennent des informations précieuses sur la forme 3D, ce qui a poussé Schwarz et al. dans [Schwarz et al. 2015] à explorer le concept d'apprentissage par transfert sur les données RGB-D pour la tâche de classification des objets. Dans ce modèle, les données RGB-D ont été représentées dans une perspective canonique et la profondeur

obtenue a été colorée en fonction de la distance par rapport au centre de l'objet. Le CNN utilisé dans ce modèle était un CNN pré-entrainé pour la catégorisation des objets. La sortie des deux dernières couches du réseau a été utilisée comme descripteur d'objet, qui a été transmis aux SVM pour apprendre la classe de l'objet. L'étape de pré-entraînement permet au modèle d'extraire de meilleures caractéristiques, ce qui contribue à améliorer les performances.

L'apprentissage approfondi s'est avéré efficace pour apprendre les données RGB-D malgré la simplicité des modèles. De plus, le traitement séparé du canal de profondeur permet de savoir que la profondeur contient des informations précieuses sur l'objet 3D qui contribuent à l'ensemble du processus d'apprentissage. Cependant, ces méthodes n'apprennent pas la géométrie complète de l'objet 3D et ne peuvent déduire que certaines des propriétés 3D en fonction de la profondeur. Les travaux ultérieurs nous donnent une vue d'ensemble, examinent la représentation volumétrique complète de la forme 3D plutôt que d'utiliser les images 2D plates de couleur et les informations de profondeur.

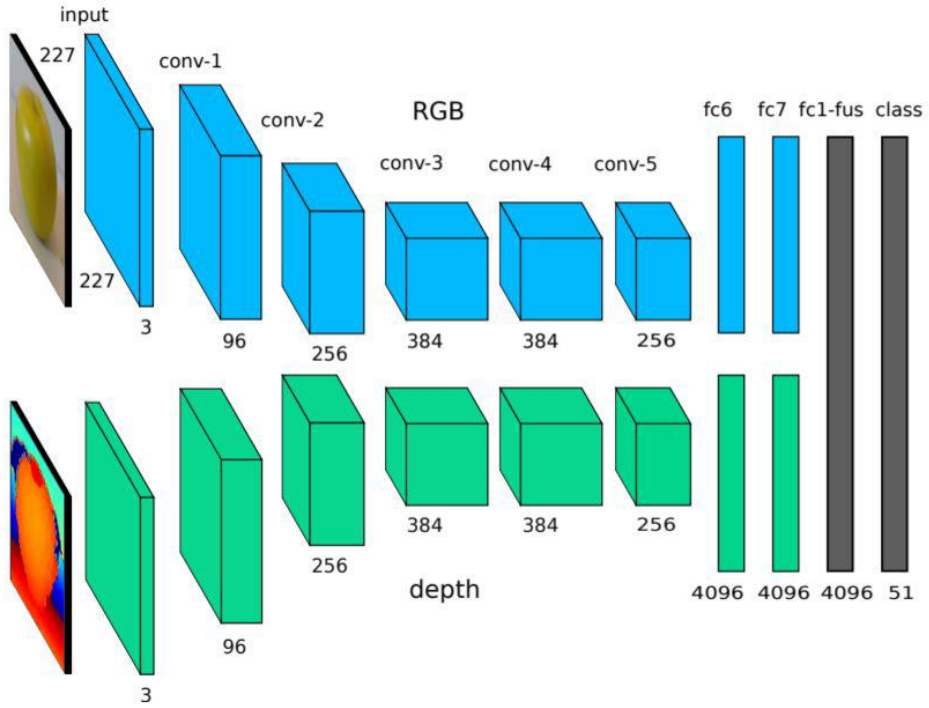


Figure II-21: Deux flux de CNN sur des données RGB-D pour la tâche de reconnaissance d'objets 3D [Eitel et al. 2015].

II.4.8.4 Architectures d'apprentissage approfondi basées sur les données volumétriques

Certains efforts ont été consacrés au traitement des représentations volumétriques 3D des objets 3D afin d'exploiter la géométrie complète de l'objet. ShapeNet [Wu et al. 2015] est le premier modèle DL qui exploite la géométrie des objets 3D représentés sous forme de voxels. L'objet d'entrée est un tenseur binaire 30x30x30 indiquant si le voxel fait partie de l'objet 3D ou non. Un concept de "Convolutional Deep Belief Net (CDBN)" [Lee et al. 2009] a été adapté du DL 2D pour modéliser des objets 3D. Les CDBN peuvent être considérés comme une variation des DBN où la convolution est également utilisée pour bénéficier de la propriété de partage de poids afin de réduire le nombre de paramètres. C'est pourquoi les CDBN sont utilisés pour modéliser et apprendre la distribution de probabilité commune sur des voxels représentant différentes catégories d'objets avec un petit nombre de paramètres. ShapeNet se compose de cinq couches (la couche d'entrée, trois couches de

convolution et la couche de sortie). Le réseau proposé a été initialement pré-entraîné. Pendant la phase de pré-entraînement, le réseau a été entraîné par couche, les quatre premières couches étant entraînées selon la méthode de la "Contrastive Divergence" et la dernière couche étant entraînée selon la méthode de la "Fast Persistent Contrastive Divergence". Pendant la phase de test, les données sont fournies sous forme de cartes de profondeur unique pour l'objet 3D, qui est ensuite converti en une représentation en grille de voxels.

ShapeNet a été testé pour trois tâches différentes : Classification des formes 3D, reconnaissance basée sur la vue et prédiction de la meilleure vue suivante. Bien que ShapeNet ait été le premier réseau à exploiter directement les données volumétriques 3D en apprentissage approfondi, il impose de nombreuses contraintes. La dimension supplémentaire du noyau de convolution donne lieu à un modèle extrêmement difficile à calculer qui peut difficilement traiter des données de grande taille ou à haute résolution. De plus, le réseau est formé sur la vue isolée de voxels de taille fixe sans aucune information supplémentaire ou encombrement de fond qui rend le processus d'apprentissage difficile. Malgré ces limitations, ce réseau produit des résultats impressionnants étant donné qu'il fonctionne sur des voxels de faible résolution. De plus, en plus de présenter ShapeNet dans [Wu et al. 2015], les auteurs ont également présenté l'ensemble de données ModelNet que nous décrivons en détail dans la section 2.3 du chapitre 2. La disponibilité des objets 3D étiquetés a ouvert la porte à d'autres expériences et a renforcé la recherche dans ce domaine.

Dans [Maturana et al. 2015], Maturana et Scherer ont exploité le concept de convolution 3D et ont proposé à VoxNet d'effectuer la reconnaissance d'objets 3D sur différentes représentations de données 3D : données RVB-D, nuages de points LIDAR et objets 3D CAO. La convolution dans VoxNet a suivi la convolution 2D, sauf pour le filtre où un filtre 3D a été utilisé au lieu d'un filtre 2D. L'architecture du réseau est composée de (la couche d'entrée, deux couches Conv, une couche de Pooling et deux couches FC). Les données d'entrée ont été construites comme une grille d'occupation volumétrique de 32x32x32 voxels et elles ont été transmises au réseau qui a été formé en utilisant SGD avec momentum. Les expériences ont montré que VoxNet est plus performant que ShapeNet lorsqu'il est testé sur les ensembles de données ModelNet10, ModelNet40 et NYUv2 pour la tâche de classification lorsque les réseaux sont entraînés à partir de zéro. Cependant, ShapeNet surpasse VoxNet lorsqu'il est testé sur NYUv2 avec le modèle pré-entraîné pour ModelNet10. Motivés par les performances prometteuses de VoxNet, Seaghat et al [Seaghat et al. 2016] ont modifié l'architecture de VoxNet pour intégrer l'orientation de l'objet 3D dans le processus d'apprentissage. Cela a permis d'améliorer les résultats de la classification sur l'ensemble de données ModelNet10. Pour apprendre les représentations de données 3D à l'aide de techniques non supervisées, Sharma et al [Sharma et al. 2016] ont proposé d'utiliser un "Convolutional Volumetric Auto-Encoder (VConv-DAE)" pour apprendre l'intégration d'objets 3D de manière non supervisée. VConv-DAE apprend la représentation volumétrique à partir de données bruitées en estimant la grille d'occupation des données de voxels. Dans [Wu et al. 2016], Wu et al. ont présenté le 3D-GAN pour apprendre implicitement les caractéristiques des objets 3D de l'espace latent probabiliste en utilisant le discriminateur adversatif. Le discriminateur adversatif apprend à capturer la structure des objets 3D avec la capacité d'identifier si elle est réelle ou synthétisée. Il obtient ainsi une information sémantique discriminante sur les objets 3D qui est efficace pour modéliser les objets 3D et générer des données synthétiques comme le montrent les expériences.

Les grandes avancées dans les architectures 2D très profondes ont motivé Brock et al [Brock et al. 2016] à adopter de tels modèles pour la classification des objets 3D sur les ensembles de données ModelNet10 et ModelNet40. Les auteurs ont proposé un modèle très profond de Voxception-ResNet (VRN). Comme son nom l'indique, VRN s'appuie sur les architectures Inception [He et al. 2016] [Szegedy et al. 2017]. En plus d'adopter les méthodes de normalisation par lots [He et al. 2016] [Ioffe et al. 2015] et les techniques stochastiques de profondeur de réseau [Huang et al. 2016], VRN est composé de 45 couches

profondes, ce qui a nécessité une augmentation des données pour le processus d'entrainement. VRN est similaire à VoxNet dans le sens où ils adoptent tous deux ConvNet avec des filtres 3D, mais VRN est très profond par rapport à VoxNet, qui a réalisé une amélioration significative de 51,5% dans la tâche de classification sur les ensembles de données ModelNet, ce qui marque l'état de l'art des performances sur cet ensemble de données. Malgré les performances remarquables de cette méthode, elle a une architecture complexe et nécessite une augmentation importante de données pour éviter le problème de surajustement qui peut résulter de l'architecture profonde d'un petit ensemble de données. Ces contraintes sont limitatives et ne peuvent pas être facilement réalisées. Xu et Todorovic ont proposé le modèle de recherche de faisceau pour apprendre l'architecture optimale de CNN 3D afin d'effectuer la classification sur l'ensemble de données ModelNet40 [Xu et al. 2016]. Le modèle proposé identifie le nombre de nœuds de CNN 3D, le nombre de couches, la connectivité et les paramètres d'entraînement également. Le modèle commence par un réseau assez simple (deux couches Conv et une couche FC). La méthode de recherche de faisceau part de cette architecture et l'étend pour construire le modèle CNN 3D optimal, soit en ajoutant un nouveau filtre Conv, soit en ajoutant une nouvelle couche Conv. Le modèle de recherche de faisceau est formé par couche, la méthode standard de "Contrastive Divergence" étant utilisée pour former les couches Conv, tandis que la méthode de "Fast Persistent Contrastive Divergence" est utilisée pour former la couche FC. Une fois qu'une couche est entraînée, les poids sont alors fixés et les paramètres d'activation sont transférés à la couche suivante. La méthode proposée a produit des résultats significatifs dans la tâche de classification sur l'ensemble de données ModelNet40.

Dans une tentative d'apprentissage des caractéristiques 3D à différentes échelles en utilisant des CNN 3D, Song et Xiao dans [Songet al. 2016] ont présenté un modèle de "Deep Sliding Shapes" pour effectuer la reconnaissance et la classification d'objets 3D sur le jeu de données ModelNet. Les auteurs ont converti les cartes de profondeur des scènes RGB-D en voxels 3D en utilisant une "Truncated Signed Distance Function, TSDF" directionnelle. De plus, le "Region Proposed Network, RPN" 3D a été proposé pour traiter l'objet 3D à deux échelles différentes et générer deux boîtes de délimitation 3D autour de l'objet 3D. Cela permet de traiter des données 3D d'échelles et de tailles différentes. La scène est prétraitée pour obtenir des informations sur l'orientation de l'objet afin d'éviter toute ambiguïté dans les orientations des boîtes englobantes. La puissance de ce modèle provient de la représentation TSDF qui donne une représentation informative sur la géoémétrie de l'objet 3D au lieu d'utiliser la carte de profondeur brute. De plus, les valeurs RGB peuvent être ajoutées à la TSDF, ce qui donne une représentation compacte. Ce modèle a produit des résultats comparables sur le jeu de données NYUv2 pour les tâches de détection d'objets sur diverses classes d'objets.

Malgré l'efficacité des modèles volumétriques d'objets 3D, la plupart des architectures actuelles nécessitent une énorme puissance de calcul en raison de l'opération de convolution et du grand nombre de paramètres. C'est ce qui a motivé Zhi et al [Zhi et al. 2018] à proposer LightNet. LightNet est un CNN volumétrique en temps réel conçu pour la reconnaissance d'objets 3D. La principale puissance de LightNet est double. LightNet exploite la puissance du multitâche pour apprendre plusieurs fonctions en même temps. De plus, pour obtenir une convergence plus rapide avec moins de paramètres, l'opération de normalisation par lots est utilisée entre l'opération de convolution et l'activation. LightNet comprend deux tâches d'apprentissage principales : la première consiste à apprendre les étiquettes de classe pour chaque voxel 3D et la seconde à apprendre l'orientation. LightNet a été testé sur les ensembles de données ModelNet pour la tâche de classification où il a surpassé VoxNet d'environ 24,25% sur ModelNet10 et 24,25% sur ModelNet40 avec des paramètres inférieurs à 67% de VoxNet. Cela prouve la force et les capacités du modèle proposé. D'autres travaux ont tenté d'étudier davantage la représentation multi-vues afin d'incorporer toutes les informations géométriques de la scène provenant de plusieurs vues 2D tout en utilisant des modèles DL 2D pour le traitement, ce qui est plus plausible sur le plan des calculs.

II.4.8.5 Architectures d'apprentissage approfondi basées sur les données multi-vues

Malgré l'efficacité des méthodes volumétriques d'apprentissage en profondeur, la plupart de ces approches sont coûteuses en termes de calcul en raison de la nature volumétrique des filtres convolutifs qui permettent d'extraire les caractéristiques augmentant la complexité du calcul de manière cubique par rapport à la résolution des voxels, ce qui limite l'utilisation des modèles volumétriques DL en 3D. C'est pourquoi l'exploitation de vues multiples d'objets 3D est pratique. En effet, elle permet d'exploiter les paradigmes DL 2D déjà établis sans qu'il soit nécessaire de concevoir un modèle sur mesure pour les données volumétriques 3D à haute complexité de calcul. L'une des premières tentatives d'exploitation de modèles DL 2D pour l'apprentissage de données 3D multi-vues a été présentée par Leng et al. dans [Leng et al. 2014a], où le DBN a été utilisé sur diverses images de profondeur basées sur des vues pour extraire des caractéristiques de haut niveau de l'objet 3D. Une méthode d'apprentissage ultérieure a été utilisée pour entraîner le DBN en utilisant la méthode de "Contractive Divergence". Le modèle proposé a donné de meilleurs résultats que l'approche des descripteurs composites utilisée dans [Daras et al. 2010]. Xie et al [Xie et al. 2015] ont proposé une "Multi-View Deep Extreme Learning Machine, MVD-ELM". Le MVD-ELM proposé a été utilisé sur 20 images de profondeur multi-vues qui ont été uniformément capturées avec une sphère au centre de l'objet 3D. La MVD-ELM proposée contenait des couches Conv qui avaient des poids communs à toutes les vues. Les poids d'activation de sortie ont été optimisés en fonction des cartes de caractéristiques extraites. Ce travail a été étendu pour être entièrement convolutif, ce qui a donné (FC-MVD-ELM). FC-MVD-ELM a été entraîné en utilisant les images de profondeur multi-vues à tester pour la segmentation 3D. Les étiquettes prédites lors de l'apprentissage ont ensuite été projetées sur l'objet 3D où le résultat final a été lissé par la méthode d'optimisation de la coupe du graphique. Le MVD-ELM et le FC-MVD-ELM ont tous deux été testés sur des tâches de classification et de segmentation de formes 3D et ont surpassé les travaux précédents [Wu et al. 2015] et ont réduit le temps de traitement de manière significative.

D'autres recherches ont été menées par Leng et al. pour utiliser les paradigmes DL sur les données 3D multi-vues. Dans [Leng et al. 2015], Leng et al. ont proposé une extension des AE classiques de manière similaire à l'architecture CNN. Le cadre qu'ils proposent est appelé "Stacked Local Convolutional AutoEncoders, SLCAE". SLCAE fonctionne sur plusieurs images de profondeur multi-vues de l'objet 3D. Dans SLCAE, les couches FC ont été remplacées par des couches qui ont été connectées localement grâce à l'utilisation de l'opération de convolution. Plusieurs AE ont été empilées, la sortie de la dernière AE étant utilisée comme représentation finale de l'objet 3D. Expériences sur différents ensembles de données : SHREC'09, NTU et PSB ont prouvé les capacités de ce modèle. Dans le prolongement des travaux précédents, Leng et al. ont proposé un "3D Convolutional Neural Network, 3D-CNN" pour traiter simultanément différentes vues 2D de l'objet 3D [Leng et al. 2015]. Les différentes vues sont triées dans un ordre spécifique pour garantir que toutes les vues des objets suivent la même convention lors de l'entraînement. Le 3DCNN proposé est composé de quatre couches Conv, trois couches Pooling et deux couches FC. Le réseau proposé a été testé pour les tâches de recherche sur les mêmes ensembles de données que ceux utilisés pour les tests [Leng et al. 2015]. Cependant, les résultats ont montré que le modèle ultérieur était plus performant sur les trois ensembles de données, ce qui implique que le modèle précédent a pu apprendre des caractéristiques plus discriminantes pour représenter divers objets 3D.

Un nouveau "Multi-View Convolutional Neural Network, MVCNN" a été proposé par Su et al. dans [Su et al. 2015] pour les tâches de recherche et de reconnaissance/classification d'objets 3D. Contrairement au modèle de Leng dans [Leng et al. 2015], MVCNN a traité plusieurs vues pour les objets 3D sans ordre spécifique en utilisant une couche de Pooling de vues. La figure II-22 montre l'architecture complète du modèle. Deux configurations différentes pour capturer les vues multiples des objets 3D ont été testées. La première a rendu 12 vues de l'objet en plaçant 12 caméras virtuelles

équidistantes autour de l'objet, tandis que l'autre configuration comprenait 80 vues virtuelles. MVCNN a été pré-entraîné à l'aide de l'ensemble de données ImageNet1K et mis au point sur ModelNet40 [Wu et al. 2015]. Le réseau proposé comporte deux parties, la première où les vues de l'objet sont traitées séparément et la seconde où l'opération de mise en commun maximale est effectuée sur toutes les vues traitées dans la couche de mise en commun des vues (pooling layer), ce qui donne une représentation compacte unique pour l'ensemble de la forme 3D. Dans la couche de pooling des vues, la vue avec l'activation maximale est la seule considérée, tout en ignorant toutes les autres vues avec des activations non maximales. Cela signifie que seules quelques vues contribuent à la représentation finale de la forme, ce qui entraîne une perte de l'information visuelle. Pour surmonter ce problème, des expériences ont montré que le MVCNN avec la couche de pooling de vues maximales (max-view pooling layer) surpassait ShapeNet [Wu et al. 2015] dans les tâches de classification et de recherche par une marge remarquable. Dans [Johns et al. 2016], Johns et al. ont exploité la représentation de données multi-vues à l'aide de CNN en représentant des objets 3D sous des trajectoires de caméra non contraintes avec un ensemble de paires d'images 2D. La méthode proposée classe chaque paire séparément et pondère ensuite la contribution de chaque paire pour obtenir le résultat final. L'architecture VGG-M a été adoptée dans ce cadre et se compose de cinq couches Conv et de trois couches FC. Les vues des objets 3D sont représentées soit par des images en profondeur, soit par des images en niveaux de gris, soit par les deux. Ce modèle est plus performant que le MVCNN proposé par Su et al [Su et al. 2015] et les architectures ShapeNet basées sur les voxels [Wu et al. 2015].

L'efficacité des modèles DL multi-vues a poussé les chercheurs à étudier des méthodes plus basées sur les GPU pour apprendre les caractéristiques des données 3D multi-vues. C'est ce qui a poussé Bai et al [Bai et al. 2016] à proposer un moteur de recherche CNN en temps réel basé sur les GPU pour les vues 2D multiples d'objets 3D. Le modèle proposé, appelé GIFT, utilise deux fichiers inversés : le premier sert à accélérer le processus de mise en correspondance multi-vues et le second à classer les résultats initiaux. La requête traitée est complétée en une seconde. Le modèle GIFT a été testé sur un ensemble de différents ensembles de données et il a produit une meilleure performance par rapport aux méthodes de l'état de l'art.

Les efforts pour apprendre les représentations de données 3D multi-vues ont continué à évoluer et en [Zanuttigh et al. 2017], Zanuttigh et Minto ont proposé un CNN multibranche pour la classification des objets 3D. Ce modèle s'appuie sur des cartes de profondeur rendues à partir de différents points de vue pour l'objet 3D. Chaque branche de CNN se compose de cinq couches Conv pour traiter une carte de profondeur produisant un vecteur de classification. Les vecteurs de classification résultants sont l'entrée d'un classificateur linéaire pour identifier la catégorie/classe de l'objet 3D. Le modèle proposé a produit des résultats comparables à l'état de l'art. Sur la base des ensembles dominants, Wang et al. dans [Wang et al. 2017] ont proposé des couches récurrentes de clustering et de pooling de vues. Le concept clé de ce modèle est de mettre en commun des vues similaires et de les regrouper de manière récurrente pour construire un vecteur d'entités mis en commun. Ensuite, les vecteurs d'entités groupées construits sont alimentés comme entrées dans la même couche de manière récurrente dans la couche de clustering récurrente. Dans cette couche, un graphe de similarité des vues est calculé, dont les nœuds représentent les vecteurs de caractéristiques et les bords représentent les poids de similarité entre les vues. Dans le graphe construit, les similitudes et les différences entre les différentes vues sont affichées, ce qui est très efficace dans la tâche de reconnaissance des formes 3D. Le modèle proposé a obtenu des résultats très comparables aux méthodes précédentes [Su et al. 2015] [Wu et al. 2015]. A l'origine des progrès des modèles DL multi-vues, Qi et al. [Qi et al. 2016] ont fourni une étude comparative entre les techniques DL multi-vues et les techniques DL volumétriques pour la tâche de reconnaissance d'objets. Dans le cadre de cette étude, les auteurs ont proposé une approche de Sphererendering pour filtrer les objets 3D multirésolution à plusieurs échelles. Avec l'augmentation des données, les auteurs ont réussi

à améliorer les résultats des MVCNN sur ModelNet40. Récemment, Kanezaki et al [Kanezaki et al. 2016] ont atteint des résultats de pointe à la fois sur ModelNet10 et ModelNet40 dans le problème de classification en utilisant RotationNet. RotationNet forme un ensemble d'images multi-vues pour l'objet 3D mais ne nécessite pas toutes les vues en même temps. Il permet plutôt une saisie séquentielle et met à jour la probabilité de la catégorie de l'objet en conséquence.

La représentation multi-vues s'est avérée légèrement plus performante que la représentation volumétrique avec une puissance de calcul moindre. Cependant, cette représentation pose certains problèmes. Le nombre suffisant de vues et la façon dont elles ont été acquises est un facteur critique pour la représentation de la forme 3D. De plus, la représentation multi-vues ne préserve pas les propriétés géométriques intrinsèques de la forme 3D. C'est ce qui a poussé à définir une nouvelle notion de convolution opérant sur les formes 3D pour capturer leurs propriétés intrinsèques.

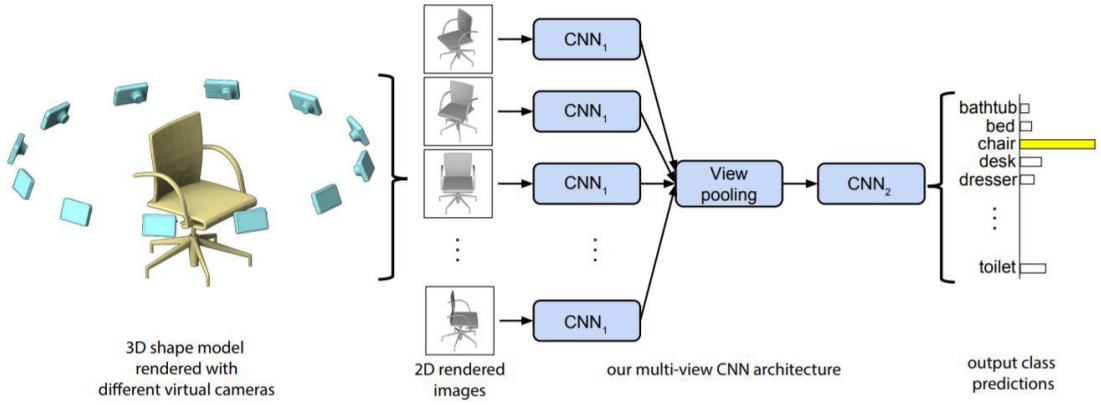


Figure II-22: Architecture MVCNN [Su et al. 2015] appliquée à la multi-vue d'objets 3D sans ordre spécifique.

II.4.8.6 Architectures d'apprentissage approfondi basées sur les nuages de points

Les nuages de points fournissent une représentation expressive, homogène et compacte de la géométrie de surface 3D sans les irrégularités combinatoires et les complexités des maillages. C'est pourquoi les nuages de points sont faciles à apprendre. Cependant, le traitement des nuages de points est délicat en raison de leur double nature. Les nuages de points peuvent être considérés comme des données à structure euclidienne locale lorsqu'on considère un point par rapport à son voisinage (un sous-ensemble de points) de telle sorte que l'interaction entre les points forme un espace euclidien avec une métrique de distance qui est invariable aux transformations comme la translation, la rotation. Cependant, compte tenu de la structure globale du nuage de points, il s'agit d'un ensemble de points non ordonnés sans ordre spécifique qui impose la nature irrégulière non euclidienne à la structure globale des données.

Certains travaux récents ont considéré les nuages de points comme une collection d'ensembles de tailles différentes. Vinyals et al. dans [Vinyals et al. 2015] utilisent un réseau de lecture-processus-écriture pour le traitement des ensembles de points afin de montrer la capacité du réseau à apprendre à trier les nombres. Il s'agissait d'une application directe de DL sur un ensemble non ordonné pour l'application de traitement du langage naturel (Natural Language Processing, NLP). Inspirés par ce travail, Ravanbakhsh et al [Ravanbakhsh et al. 2016] ont proposé ce qu'ils ont appelé la couche d'équivalence de permutation dans un cadre supervisé et semi-supervisé. Cette couche est obtenue par le partage des paramètres pour apprendre l'invariance de permutation ainsi que les transformations rigides à travers les données. Ce réseau a effectué la classification 3D et l'ajout de chiffres MNIST. Bien que ce réseau soit relativement simple, il n'a pas bien

fonctionné pour la tâche de classification 3D sur l'ensemble de données ModelNet. Une version plus approfondie de ce modèle a été étendue dans [Zaheer et al. 2017] où le modèle DeepSet proposé a donné de meilleurs résultats que les méthodes de pointe de classification 3D sur les données de ModelNet. Ceci est dû à la propriété d'invariance de permutation que la couche d'équivalence de permutation apporte aux modèles précédents. [Qi et al. 2016] a également utilisé une couche similaire avec une différence majeure puisque la couche d'équivalence de permutation est normalisée au maximum.

PointNet [Qi et al. 2017a] est le pionnier de l'utilisation directe du nuage de points comme entrée où chacun de ses points est représenté à l'aide des coordonnées (x, y, z). Lors de l'étape de prétraitement, la transformation des caractéristiques et les données d'entrée sont intégrées dans l'architecture de PointNet. PointNet est composé de trois modules principaux : un module "Spatial Transformer Network, STN", un module RNN et une fonction symétrique simple qui regroupe toutes les informations de chaque point du nuage de points. Le STN canonise les données avant de les transmettre au RNN, c'est-à-dire qu'il traite toutes les données sous une forme canonique, et apprend les points clés du nuage de points qui correspondent approximativement au squelette de l'objet 3D. Ensuite vient le module RNN qui apprend le nuage de points comme un signal séquentiel de points et, tout en entraînant ce modèle avec une séquence permutée de façon aléatoire, ce RNN devient invariant à la séquence de l'ordre d'entrée du point du nuage de points. Enfin, le réseau agrège toutes les caractéristiques de points résultantes en utilisant l'opération max-pooling qui est également invariante à la permutation. PointNet a prouvé qu'il est robuste contre les perturbations partielles des données et des entrées. Il a été testé sur des tâches de classification et de segmentation où il s'est avéré produire des résultats comparables à l'état de l'art.

Malgré les résultats compétitifs obtenus par PointNet, il n'est pas en mesure de tirer pleinement parti de la structure locale du point pour saisir les modèles détaillés à grain fin en raison de l'agrégation de toutes les caractéristiques du point. Pour résoudre ce problème, PointNet++ [Qi et al. 2017b] s'appuie sur PointNet en l'appliquant de manière récursive à une partition imbriquée des ensembles de points d'entrée. Malgré la capture d'un plus grand nombre de caractéristiques, l'architecture résultante est compliquée, ce qui augmente la taille des caractéristiques les plus élevées et le temps de calcul.

Au lieu d'opérer directement sur la structure des nuages de points, Kd-Networks de Klokov et al [Klokov et al. 2017] propose d'imposer une structure en kd-tree du nuage de points d'entrée à utiliser pour apprendre les poids partagés entre les points de l'arbre. Kd-tree est un réseau de feed-forward qui possède des paramètres apprentissables associés aux poids des nœuds de l'arbre. Ce modèle a été testé pour la classification des formes, la recherche de formes et la segmentation des parties de formes, produisant des résultats compétitifs. Suivant le même concept de ne pas travailler directement sur la structure du nuage de points, Roveri et al [Roveri et al. 2018] ont proposé d'extraire un ensemble de cartes de profondeur en 2D à partir de différentes vues du nuage de points et de les traiter en utilisant des réseaux résiduels (ResNet50) [He et al. 2015]. Le système proposé comprend 3 modules. Le premier module est chargé d'apprendre k vues directionnelles du nuage de points d'entrée pour générer les cartes de profondeur en conséquence dans le deuxième module. Le troisième et dernier module traite les k cartes de profondeur générées pour la classification des objets. L'innovation de ce framework est principalement axée sur la transformation automatique de nuages de points non ordonnés en cartes de profondeur 2D informatives sans qu'il soit nécessaire d'adapter le module réseau pour tenir compte de l'invariance de permutation et des différentes transformations des données d'entrée.

Récemment, peu d'articles ont fait état de leurs travaux sur l'apprentissage non supervisé sur les nuages de points. Dans FoldingNet [Yang et al. 2018], Yang et al. ont proposé d'utiliser l'AE pour modéliser différents objets 3D représentés comme des nuages de points par un nouveau décodeur basé sur le pliage qui déforme une grille canonique 2D dans la surface sous-jacente du nuage de points 3D. FoldingNet est en mesure d'apprendre comment générer des coupes sur la grille 2D pour créer des surfaces 3D et généraliser à

certaines variations intra-classe de la même classe d'objets 3D. Un SVM a été utilisé en plus de ce FoldingNet pour la classification 3D, où il s'est avéré performant avec la représentation discriminante apprise pour différents objets 3D. FoldingNet a obtenu une grande précision de classification sur ModelNet40. Un autre modèle non supervisé a été proposé par Li et al. appelé SO-Net [Li et al. 2018]. SO-Net est un réseau à permutation invariante qui peut tolérer des entrées de nuages de points non ordonnées. SO-Net construit cette distribution spatiale des points du nuage de points en utilisant des "cartes auto-organisées" (Self-Organizing Maps, SOM). Ensuite, une extraction de caractéristiques hiérarchiques sur les points du nuage de points et les nœuds SOM est utilisée, ce qui donne un vecteur de caractéristiques singulières qui représente l'ensemble du nuage de points. L'agrégation de caractéristiques locales se fait selon un champ de réception réglable où le chevauchement est contrôlé pour obtenir des caractéristiques plus efficaces. Le SO-Net a été testé sur des tâches de classification et de segmentation, produisant des résultats prometteurs et très comparables aux techniques de l'état de l'art.

Comme on l'a remarqué dans toutes les méthodes proposées précédemment, le principal problème du traitement des nuages de points est la structure non ordonnée de cette représentation où les chercheurs essaient de rendre le processus d'apprentissage invariant à l'ordre du nuage de points. La plupart de ces méthodes ont eu recours à des techniques de clustering afin d'opter pour des points similaires et de les traiter ensemble.

II.4.8.7 Architectures d'apprentissage approfondi basées sur les graphiques et les maillages

La représentation idéale pour les graphiques et les maillages est celle qui peut capturer toute la structure intrinsèque de l'objet et qui peut également être apprise avec les méthodes de descente de gradient. Cela est dû à leur stabilité et à leur utilisation fréquente dans les CNN. Cependant, l'apprentissage de ces représentations irrégulières est une tâche difficile en raison des propriétés structurelles de ces représentations.

Motivés par le succès des CNN dans un large éventail de tâches de vision par ordinateur, les récents efforts de recherche ont été orientés vers la généralisation des CNN à de telles structures irrégulières. L'analyse des propriétés de ces données montre que les maillages peuvent être convertis en graphiques. Ainsi, les modèles proposés pour les graphes peuvent être utilisés sur des données à structure maillée, mais pas l'inverse. La plupart des travaux existants portent explicitement sur les données structurées en graphes et quelques travaux ont été adaptés à des représentations en mailles. Nous présentons ici une vue d'ensemble des travaux récents sur chaque représentation en fournissant une classification générale des méthodes existantes basée sur l'approche utilisée.

Graphiques : L'étude des propriétés structurelles des graphes et des maillages suggère que les méthodes d'apprentissage proposées pour les graphes sont également applicables aux maillages. Les méthodes existantes pour les réseaux neuronaux convolutifs de graphes (Graph Convolutional Neural Networks, GCNN) peuvent être classées en deux grandes catégories : les méthodes de filtrage spectral et les méthodes de filtrage spatial. Nous examinons ici le concept sous-jacent de chaque méthode et donnons un aperçu du travail effectué dans chaque direction. La distinction entre les deux directions réside dans la manière dont le filtrage est employé et dont les informations traitées localement sont combinées.

Méthodes de filtrage spectral

La notion de convolution spectrale sur des données structurées par des graphes a été introduite par Bruna et al. dans [Bruna et al. 2013] où les auteurs ont proposé la CNN spectrale (SCNN) fonctionnant sur des graphes. La base des méthodes de filtrage spectral est d'utiliser la décomposition propre spectrale du graphe Laplacien pour définir un opérateur de type convolution. Cela redéfinit l'opération de convolution dans le domaine spectral où les deux pierres angulaires principales sont analogues : les patchs du signal dans

le domaine euclidien correspondent aux fonctions définies sur les nœuds du graphe, par exemple les caractéristiques, mises en correspondance avec le domaine spectral en se projetant sur les vecteurs propres du graphe Laplacien. L'opération de filtrage elle-même se produit dans le domaine euclidien et correspond à la mise à l'échelle des signaux dans la base propre. Cette définition implique que la convolution est un opérateur linéaire qui commute avec l'opérateur Laplacien [Bronstein et al. 2017]. Malgré l'aspect innovant du modèle de Bruna, il présente de sérieuses limitations du fait qu'il dépend de la base et qu'il est coûteux en termes de calcul. Être dépendant de la base signifie que si les coefficients du filtre spectral ont été appris par rapport à une base spécifique, l'application des coefficients appris sur un autre domaine avec une autre base produira des résultats très différents comme illustré dans [Bronstein et al. 2017]. L'autre limite du coût de calcul provient du fait que le filtrage spectral est une opération non locale qui implique des données sur l'ensemble du graphique, sans compter que le graphique Laplacian est coûteux à calculer. Cela constitue une charge de calcul pour la généralisation à d'autres bases et le traitement de graphiques à grande échelle.

Les travaux de [Kovnatsky et al. 2013] ont abordé le problème de la dépendance de la base en construisant une base orthogonale compatible entre différents domaines par le biais d'une diagonalisation conjointe. Cependant, cela nécessitait une connaissance préalable de la correspondance entre les domaines. Pour certaines applications comme les réseaux sociaux, cette hypothèse est valable car la correspondance peut être facilement calculée entre deux occurrences temporelles dans lesquelles de nouveaux bords et sommets ont été ajoutés. Cependant, l'application de cette hypothèse aux maillages est plutôt peu raisonnable car trouver la correspondance entre deux maillages est une tâche difficile en soi. Par conséquent, il est irréaliste de supposer que l'on connaît la correspondance entre les domaines dans un tel cas [Bronstein et al. 2017]. Étant donné la nature non locale du filtrage spectral et la nécessité d'impliquer toutes les données du graphique dans le traitement, des travaux récents ont proposé l'idée d'approximation pour produire des filtres spectraux locaux [Defferrard et al. 2016] [Kipf et al. 2016]. Ces méthodes proposent de représenter les filtres via une expansion polynomiale au lieu d'opérer directement sur le domaine spectral. Defferrard et al. dans [Defferrard et al. 2016] ont réalisé un filtrage spectral local sur des graphes en utilisant des polynômes de Tchebychev afin d'approcher les filtres spectraux des graphes. Les caractéristiques résultant de l'opération de convolution sont ensuite grossies en utilisant l'opération de pooling des graphes. Kipf et Welling [Kipf et al. 2016] ont simplifié l'approximation polynomiale proposée dans [Defferrard et al. 2016] et ont utilisé une approximation linéaire du premier ordre des filtres spectraux des graphes pour produire des filtres spectraux locaux qui sont ensuite utilisés dans un GCNN à deux couches. Chacune de ces deux couches utilise les filtres spectraux locaux et agrège les informations provenant du voisinage immédiat des sommets.

Encouragés par le succès des modèles locaux de filtrage spectral, Wang et al. [Wang et al. 2018] ont proposé de tirer parti de la puissance des GCNN spectraux dans le cadre de pointNet++ [Qi et al. 2017b] pour traiter les nuages de points non ordonnés. Ce modèle fusionne l'innovation du framework pointNet++ avec le filtrage spectral local tout en remédiant indépendamment à deux défauts de ces modèles. Ainsi, au lieu de traiter chaque point indépendamment dans les nuages de points comme proposé dans pointNet++, ce modèle utilise le filtrage spectral comme technique d'apprentissage pour modifier les informations structurelles du voisinage de chaque point. De plus, plutôt que d'utiliser la méthode du "gagnant gourmand" dans l'opération de max-pooling du graphique, cette méthode adopte une stratégie récursive de pooling et de clustering. Contrairement aux précédentes méthodes de filtrage spectral, cette méthode ne nécessite aucun pré-calcul et peut être entraînée de bout en bout, ce qui permet de construire le graphe de manière dynamique et de calculer le graphe Laplacian et la hiérarchie de pooling à la volée, contrairement à [Bruna et al. 2013] [Defferrard et al. 2016] [Kipf et a. 2016]. Cette méthode a permis d'obtenir de meilleurs résultats de reconnaissance que les techniques existantes sur divers ensembles de données.

Méthodes de filtrage spatial

Le concept de filtrage spatial des graphes a débuté en [Scarselli et al. 2009] lorsque les GNN ont été proposés pour la première fois comme une tentative de généraliser les modèles DL aux graphes. Les GNN sont des constructions simples qui tentent de généraliser la notion de filtrage spatial sur les graphes via les poids du graphe. Les GNN sont composés de plusieurs couches où chaque couche est une combinaison linéaire d'opérateurs passe-haut et passe-bas du graphe. Cette formulation suggère que l'apprentissage des caractéristiques du graphe dépend du voisinage de chaque sommet. Comme pour les CNN euclidiens, une fonction non linéaire est appliquée à tous les nœuds du graphe où le choix de cette fonction varie en fonction de la tâche. La variation de la nature de la fonction non-linéaire du sommet conduit à des architectures riches [Battaglia et al. 2016] [Chang et al. 2016] [Duvenaud et al. 2015] [Li et al. 2015]. De plus, comme pour les CNN, l'opération de pooling peut être utilisée sur les données structurées par le graphe en les rendant plus grossières. Les couches de pooling des graphes peuvent être réalisées en entrelaçant les couches d'apprentissage des graphes. En comparaison avec le filtrage spectral des graphes, les méthodes de filtrage spatial ont deux points clés qui les distinguent des méthodes spectrales. Les méthodes spatiales agrègent les vecteurs de caractéristiques des nœuds de voisinage directement en fonction de la topologie du graphe en tenant compte de la structure spatiale du graphe d'entrée. Les caractéristiques agrégées sont ensuite résumées via une opération supplémentaire. Le framework GNN présenté dans [Gori et al. 2005] [Scarselli et al. 2009], propose d'intégrer chaque sommet du graphe dans un espace euclidien avec un RNN. Au lieu d'utiliser les connexions récursives dans le RNN, les auteurs ont utilisé une simple fonction de diffusion pour leur fonction de transition, en propageant la représentation des nœuds de façon répétée jusqu'à ce qu'elle soit stable et fixe. Les représentations de nœuds qui en résultent sont considérées comme les caractéristiques des problèmes de classification et de régression. La propagation répétée des caractéristiques des nœuds dans ce framework constitue une charge de calcul qui est allégée dans les travaux proposés par Li et al [Li et al. 2015]. Ils ont proposé une variante du modèle précédent qui utilise les unités récurrentes à portes pour effectuer les mises à jour d'état afin d'apprendre la représentation graphique optimale. Bruna et al. dans [Bruna et al. 2013] ont imposé le champ de réception spatial local sur les GNN pour produire leur formulation spatiale locale des GNN. L'idée principale derrière le champ de réception local est de diminuer le nombre de paramètres appris en regroupant des caractéristiques similaires basées sur une mesure de similarité [Coates et al. 2011] [Gregor et al. 2010]. Dans [Bruna et al. 2013], les auteurs ont utilisé ce concept pour calculer un clustering multi-échelle du graphique qui sera ensuite transmis à la couche de pooling. Ce modèle impose la localisation des caractéristiques traitées et réduit la quantité de paramètres traités. Cependant, il n'effectue aucun partage de poids similaire à celui des CNN 2D. Niepert et al. Dans [Niepert et al. 2014] effectue une convolution spatiale du graphe de manière simple en convertissant le graphe localement en séquences et en alimentant ces séquences dans un CNN 1D. Cette méthode est simple mais nécessite une définition explicite de l'ordre des nœuds des graphes lors d'une étape de prétraitement. Dans [Venkatakrishnan et al. 2018], les auteurs ont fourni une étude détaillée prouvant que les méthodes spectrales et les méthodes spatiales sont mathématiquement équivalentes en termes de capacités de représentation. La différence réside dans la façon dont la convolution et l'agrégation des caractéristiques apprises sont effectuées. Selon la tâche, l'architecture du GCNN (spectrale ou spatiale) est formée où les couches de convolution peuvent être entrelacées avec des couches de grossissement et de pooling pour résumer la sortie des filtres de convolution pour une représentation compacte du graphique. Ceci est crucial dans les applications de classification où la sortie n'est qu'une classe déduite des caractéristiques apprises [Bruna et al. 2013]. D'autres applications nécessitent une décision par nœud, comme la détection de communauté. Une pratique courante dans ces cas est d'avoir plusieurs couches de convolution qui calculent les représentations du graphe au niveau du nœud [Bruna et al. 2017] [Khalil et al. 2017] [Nowak et al. 2017]. Toutes ces GCNN sont

des méthodes différenciables de bout en bout qui peuvent être formées aux techniques d'apprentissage supervisé, semi-supervisé ou de renforcement.

Des maillages

Dans le domaine euclidien, l'opération de convolution est effectuée en passant un gabarit à chaque point du domaine spatial et en enregistrant la corrélation entre les gabarits à l'aide de la fonction définie à ce point. Ceci est possible grâce à la propriété de déplacement et d'invariance du domaine euclidien. Cependant, malheureusement, cela n'est pas directement applicable sur les maillages car la propriété de décalage et d'invariance est absente. C'est ce qui a poussé à définir des patchs locaux qui représentent la surface 3D d'une manière qui permette d'effectuer la convolution. Cependant, en raison du manque de paramétrage global sur les données non euclidiennes, ces patchs sont définis dans un système local de coordonnées locales, ce qui signifie que ces patchs sont également dépendants de la position. Récemment, différents frameworks de CNN non euclidiens ont été proposés. Le schéma principal utilisé par ces frameworks est très similaire, à l'exception de la façon dont les patchs sont définis pour la plupart. Les patchs locaux sont définis soit en les façonnant à la main, soit en fonction de la connectivité des sommets, tout en utilisant directement les caractéristiques du voisinage à un saut comme patch [Fey et al. 2017]. La convolution utilisée dans de tels frameworks est très similaire à la convolution 2D classique où il s'agit essentiellement d'une multiplication par élément entre le filtre de convolution et le patch et de la somme des résultats. En effet, les patchs extraits par de tels frameworks réduisent la représentation en 2D où la convolution classique peut être utilisée.

La CNN géodésique [Masci et al. 2015] a été introduite comme une généralisation des CNN classiques aux maillages triangulaires. L'idée principale de cette approche est de construire des patchs locaux en coordonnées polaires locales. Les valeurs des fonctions autour de chaque sommet du maillage sont mappées en coordonnées polaires locales à l'aide de l'opérateur de patch. Cela permet de définir les zones où la convolution géodésique est utilisée. La convolution géodésique suit l'idée de la multiplication par un modèle. Cependant, les filtres de convolution dans ce framework sont soumis à quelques rotations arbitraires en raison de l'ambiguïté des coordonnées angulaires [Masci et al. 2015]. Cette méthode a ouvert la porte à de nouvelles innovations sur l'extension du paradigme CNN aux maillages triangulaires. Cependant, ce framework souffre de multiples inconvénients. Premièrement, il ne peut être appliqué que sur des maillages triangulaires où il est sensible à la triangulation du maillage et il peut échouer si le maillage est extrêmement irrégulier. Deuxièmement, le rayon de la pièce géodésique construite doit être petit par rapport au rayon d'injectivité de la forme réelle pour garantir que la pièce obtenue est topologiquement un disque. Troisièmement, les rotations employées sur les filtres de convolution rendent le framework coûteux en termes de calcul, ce qui limite l'utilisation d'un tel framework. La CNN anisotrope (ACNN) [Boscaini et al. 2016] a été proposée pour surmonter certaines des limites de la CNN géodésique. Par rapport à la CNN géodésique, le framework ACNN n'est pas limité aux maillages triangulaires et peut également être appliqué aux graphiques. De plus, la construction des patchs locaux est plus simple et est indépendante du rayon d'injectivité du maillage. ACNN utilise le concept de filtrage spectral où l'information spatiale est également incorporée par une fonction de pondération pour extraire une fonction locale définie sur les maillages. Les filtres spectraux appris sont appliqués aux valeurs propres de l'opérateur anisotrope Laplacian Beltrami (Laplacian Beltrami Operator, LBO) et les noyaux de chaleur anisotropiques agissent comme des fonctions de pondération spatiale pour les filtres de convolution. Cette méthode a montré de très bonnes performances pour les tâches de correspondance locale. Plutôt que d'utiliser une construction à noyaux fixes comme dans les modèles précédents, Monti et al [Monti et al. 2017] ont proposé MoNet comme construction générale des correctifs. Les auteurs ont proposé de définir un système local de coordonnées de pseudo-coordonnées autour de chaque sommet avec des fonctions de pondération. Sur ces coordonnées, un ensemble de noyaux paramétriques sont appliqués sur ces pseudo-coordonnées pour définir les fonctions de pondération à chaque sommet.

C'est pourquoi les méthodes précédentes [Boscaini et al. 2016] [Masci et al. 2015] peuvent être considérées comme des exemples spécifiques de MoNet. Certains travaux récents ont été proposés pour éliminer la nécessité de définir explicitement les correctifs locaux sur les graphes ou les maillages tels que SplineCNN [Fey et al. 2017]. SplineCNN est un framework convolutif qui peut être utilisé sur des graphes dirigés de n'importe quelle dimension. Il peut donc également être appliqué sur des maillages. Au lieu de définir les correctifs locaux par une méthode graphique comme les méthodes précédentes, SplineCNN utilise les caractéristiques de l'anneau de voisinage à 1 saut du graphique comme correctif où le filtre convolutif peut fonctionner. Le filtre convolutif lui-même est un filtre spatial continu basé sur des fonctions de base B-Spline qui ont un support local. Ce framework produit des résultats de pointe sur la tâche de correspondance tout en étant très efficace sur le plan du calcul. Ceci est dû au support local de la base B-Spline qui rend le temps de calcul indépendant de la taille du noyau.

II.5 Calcul de similarité entre descripteurs d'objet 3D

Après le processus d'extraction des caractéristiques, des mesures de similarité appropriées doivent être conçues pour mesurer la similarité du contenu. L'objectif idéal de la mesure de similarité comporte deux aspects : (1) rendre les vecteurs de caractéristiques de modèles 3D similaires aussi proches que possible dans l'espace des caractéristiques et (2) maintenir les plus grandes distances possibles pour les modèles 3D dissemblables. Par conséquent, la tâche de la correspondance de similarité consiste à calculer les distances ou dissimilitudes appropriées dans l'espace multidimensionnel entre la requête de l'utilisateur et tous les modèles 3D de la base de données et à les classer également dans l'ordre décroissant des similarités. Un nombre variable de modèles est ensuite récupéré en listant les éléments les mieux classés.

À l'heure actuelle, les méthodes de recherche de similitudes disponibles dans la recherche de modèles 3D basée sur le contenu peuvent être classées en quatre catégories : (1) mesures de distance ; (2) correspondance de graphiques ; (3) apprentissage machine ; (4) mesures sémantiques. Vous trouverez ci-dessous des descriptions détaillées de ces quatre types de méthodes de comparaison des similarités.

II.5.1 Mesure de la distance

Actuellement, les mesures de distance sont peut-être les méthodes de comparaison de similarités les plus populaires et les plus utilisées, dont la plupart ont déjà été utilisées pour la recherche de médias 2D basée sur le contenu.

II.5.1.1 Distances Minkowski

Une métrique de distance est une mesure de dissimilitude avec certaines propriétés particulières, pour lesquelles il existe un vaste ensemble de recherches. Pour la recherche de modèles d'objets 3D basés sur le contenu, les mesures de distance utilisées avec succès comprennent les distances de Manhattan [Ohbuchi et al. 2002], les distances euclidiennes [Novotni et al. 2003] et les distances de Hausdorff [Suzuki et al. 2000]. Les mesures Manhattan et Euclidienne sont toutes deux basées sur les distances L_p $(p = 1, 2)$, c'est-à-dire les distances de Minkowski. La distance L_p entre deux points x, y dans l'espace à N dimensions R^N est définie comme :

$$L_p = \left(\sum_{i=1}^{N} (x_i - y_i)^p \right)^{1/p} \qquad \text{(II-34)}$$

Toutes les distances sont métriques lorsque $p \geq 1$. La distance L_p elle-même peut également être directement utilisée comme mesure de similarité. Par exemple, Osada et al [Osada et al. 2001] l'ont utilisée pour mettre en œuvre une correspondance de similarité sur

la fonction de densité de probabilité des caractéristiques de distribution des formes. En particulier, pour attribuer différents impacts à différentes caractéristiques ou pour permettre une rétroaction sur la pertinence, la distance euclidienne est souvent modifiée en distance euclidienne pondérée avec la matrice de pondération [Osada et al. 2001] [Elad et al. 2001] [Kazhdan et al. 2002].

II.5.1.2 Distances de Hausdorff

La distance de Hausdorff, une autre métrique fréquemment utilisée, est définie comme suit pour comparer deux ensembles de points de tailles différentes :

$$h(A,B) = \min_{a\in A} \max_{b\in B} d(A,B) \tag{II-35}$$

Où $d(A,B)$ est une métrique de distance, par exemple la distance euclidienne. Cependant, elle est très sensible au bruit, car même une seule valeur aberrante peut modifier la distance de Hausdorff [Veltkamp et al. 2000].

II.5.1.3 Distances d'appariement élastiques

De nombreuses autres mesures de distance ont également été étudiées pour la tâche de recherche de modèles 3D. Ohbuchi et al [Ohbuchi et al. 2002][Ohbuchi et al. 2003] ont introduit une distance de correspondance élastique afin de compenser l'effet "plus grand que souhaité" causé par les mesures de distance "rigides", par exemple la distance euclidienne, et les résultats ont été prometteurs. L'appariement élastique a été largement utilisé dans la reconnaissance vocale. Ohbuchi et al. ont effectué un appariement élastique le long de l'axe des distances, en utilisant la technique de programmation dynamique pour sa mise en œuvre afin de calculer la distance $D_E(X,Y)$. Elle étire et rétrécit localement l'axe des distances de l'histogramme afin de trouver des correspondances de distance minimales. Si la correspondance est trop élastique, une paire de formes ayant des histogrammes très différents pourrait avoir une faible valeur de distance. Ohbuchi et al. ont mis en œuvre et comparé expérimentalement les performances des fonctions de pénalité linéaire et quadratique, cette dernière étant représentée dans Eq. (III-38). Ohbuchi et al. ont utilisé la fonction de pénalité quadratique la plus performante pour leurs expériences :

$$D_E(X,Y) = g(X_n, Y_n) \tag{II-36}$$

$$g(X_n,\, Y_n) = min \begin{bmatrix} g(X_n,\, Y_{n-1}) + \Delta g(X_n,\, Y_n) \\ g(X_{n-1},\, Y_{n-1}) + 2\Delta g(X_n,\, Y_n) \\ g(X_{n-1},\, Y_n) + \Delta g(X_n,\, Y_n) \end{bmatrix} \tag{II-37}$$

$$\Delta g(X_i,\, Y_i) = |i-j| \sqrt{\sum_{k=1}^{I_a} \left(x_{i,k} - y_{i,k}\right)^2} \tag{II-38}$$

Où $X = (x_{i,k})$ et $Y = (y_{i,k})$ sont les vecteurs de caractéristiques (histogrammes 2D ayant des éléments $I_d \times I_a$) pour le modèle A et B, respectivement.

II.5.1.4 Distances améliorées d'Earthmover

Tangelder et al [Tangelder et al. 2003] ont utilisé une version améliorée de la distance d'Earthmover (Earthmover's Distance, EMD) [Rubner et al. 2000] comme mesure de la distance. Intuitivement, étant donné deux distributions, l'une peut être considérée comme une masse de terre correctement répartie dans l'espace, l'autre comme un ensemble de trous dans ce même espace. Ensuite, l'EMD mesure la moindre quantité de travail nécessaire pour remplir les trous avec de la terre. Ici, une unité de travail correspond au transport d'une unité de terre par une unité de distance au sol. Le calcul de l'EMD est basé

sur une solution au problème de transport bien connu, le problème Monge-Kantorovich. En d'autres termes, la correspondance des signatures peut être naturellement considérée comme un problème de transport en définissant une signature comme celle du fournisseur et l'autre comme celle du consommateur, et en fixant le coût d'une paire fournisseur-consommateur à une distance égale à la distance au sol entre un élément de la première signature et un élément de la deuxième signature. Intuitivement, la solution est alors la quantité minimale de "travail" nécessaire pour transformer une signature en l'autre. Ainsi, l'EMD étend naturellement la notion de distance entre des éléments individuels à celle de distance entre des ensembles ou des distributions d'éléments. Les avantages de l'EMD par rapport aux définitions précédentes des distances de distribution devraient maintenant être évidents. Tout d'abord, l'EMD s'applique aux signatures, qui englobent les histogrammes. La plus grande compacité des signatures est en soi un avantage, et il est important de disposer d'une mesure de distance qui puisse traiter ces structures de taille variable. Deuxièmement, le coût du déplacement de la "terre" reflète correctement la notion de proximité, sans les problèmes de quantification de la plupart des mesures actuelles. Même pour les histogrammes, en fait, les articles des bacs voisins contribuent maintenant à des coûts similaires, selon le cas. Troisièmement, l'EMD permet des correspondances partielles de manière très naturelle. C'est important, par exemple, pour traiter les occlusions et l'encombrement dans les applications de recherche d'images et lorsque la correspondance ne concerne que des parties d'une image. Quatrièmement, si la distance au sol est une métrique et que les poids totaux de deux signatures sont égaux, l'EMD est une véritable métrique, ce qui permet de doter les espaces d'image d'une structure métrique. Bien sûr, il est important que l'EMD puisse être calculé efficacement, surtout s'il est utilisé pour des systèmes de recherche d'images où une réponse rapide est nécessaire. En outre, la vitesse de recherche peut être augmentée si les limites inférieures de l'EMD peuvent être calculées à faible coût. Ces limites peuvent réduire considérablement le nombre d'EMD qui doivent réellement être calculées en pré-filtrant la base de données et en ignorant les images qui sont trop éloignées de la requête. Heureusement, des algorithmes efficaces pour le problème du transport sont disponibles. Par exemple, nous pouvons utiliser la méthode de transport simplex, un algorithme simplex simplifié qui exploite la structure spéciale du problème de transport. Une bonne solution de base initiale réalisable peut réduire considérablement le nombre d'itérations nécessaires. Nous pouvons calculer la solution de base initiale réalisable par la méthode de Russell [Min et al. 2004].

II.5.2 **Algorithmes de correspondance des graphes**

Lorsque deux objets 3D à comparer sont représentés par des structures de type graphe, des algorithmes spécifiques de correspondance de graphe doivent être conçus pour la correspondance de similarité entre eux. Cependant, la comparaison de deux graphes est généralement considérée comme le plus grand problème de sous-graphes isomorphes, ce qui est presque impossible à résoudre au sens général du terme. Par conséquent, les mesures de similarité de formes 3D actuellement disponibles pour la correspondance de graphes sont toutes adaptées aux caractéristiques topologiques 3D données.

Pour comparer deux modèles 3D à partir de leurs graphiques relationnels attribués (Attributed Relational Graphs, ARG) basés sur le squelette, nous devons résoudre un problème de correspondance des graphiques. Bardinet et al [Bardinet et al. 2000] ont comparé deux graphes en trouvant leur matrice d'association optimale P de sorte qu'une fonction objective E impliquant tous les types de nœuds, liens et attributs du graphe soit minimisée. Certaines contraintes heuristiques sont également exploitées dans la fonction objectif pour garantir l'exactitude de la correspondance des graphes. Ils ont proposé un algorithme d'appariement de graphes à correction d'erreurs et à étiquetage cohérent adapté au traitement des ARG et ont adopté une méthode d'optimisation non linéaire appelée "graduation". Étant donné deux ARG G et H, avec des nœuds I et J respectivement, on suppose qu'il y a des types de liens R et des types d'attributs S. Le problème est de trouver la matrice d'association P telle que la fonction objective suivante soit minimisée :

$$E_{ARG} = -\frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{I}\sum_{l=1}^{J} P_{ij}P_{kl}\sum_{r=1}^{R} C_{ijkl}^{(r)} + \alpha\sum_{i=1}^{I}\sum_{j=1}^{J} P_{ij}\sum_{s=1}^{S} C_{ij}^{(s)} \quad \text{(II-39)}$$

Soumis à :

$$\begin{cases} \forall i, & \sum_{j=1}^{J} P_{ij} \leq 1; \\ \forall j & \sum_{i=1}^{I} P_{ij} \leq 1; \\ \forall i,j & P_{ij} \in \{0,1\}, \end{cases} \quad \text{(II-40)}$$

Où $C_{ijkl}^{(r)}$ est la matrice de compatibilité pour un lien de type r, dont les composantes sont définies comme $C_{ijkl}^{(r)} = cl^{(r)}(G_{ij}^{(r)}, H_{kl}^{(r)})$ (0 si $G_{ij}^{(r)}$ ou $H_{kl}^{(r)}$ est NULL); $\{C_{ij}^{(s)}\}$ est la matrice de similarité pour un attribut de type s, dont les composantes sont définies comme $C_{ij}^{(s)} = cn^{(s)}(G_i^{(s)}, H_j^{(s)})$; $\{G_{ij}^{(r)}\}$ et $\{H_{kl}^{(r)}\}$ sont les matrices de similarité pour le lien r ; $cl^{(r)}(\cdot,\cdot)$ est une mesure de compatibilité entre un lien r en G et un lien r en H ; $\{G_i^{(s)}\}$ et $\{H_j^{(s)}\}$ sont les vecteurs correspondant à l'attribut s des nœuds de G et H ; $cn^{(s)}(\cdot,\cdot)$ est une mesure de similarité entre un nœud de G et un nœud de H, par rapport au même attribut s. P est une matrice d'association $I \times J$ qui, à la fin du processus de minimisation, fournit les correspondances entre un ensemble de primitives et l'autre : $P_{ij} = 1$ si le nœud i en G correspond au nœud j en H, 0 sinon. Notez que l'approche ne converge pas toujours vers une matrice de permutation exacte, il faut donc définir une heuristique de nettoyage. Bardinet et al. fixent dans chaque colonne de la matrice d'association P l'élément maximum à 1 et les autres à 0. Dans ce cas précis, P fournit les correspondances entre les parties du squelette des deux objets à comparer. Les contraintes ci-dessus adoptées dans la fonction objectif garantissent que deux nœuds de graphes, ou deux parties du squelette d'un objet, ne seront appariés que s'ils sont similaires et s'ils partagent le même type de relations avec leurs primitives voisines dans leurs graphes respectifs.

Dans [Siddiqi et al. 1998] [Shokoufandeh et al. 1999] [Shokoufandeh et al. 2002], un algorithme de correspondance de graphes pour les graphes de choc 2D a été proposé. Le graphe de choc est une représentation de forme émergente pour la reconnaissance d'objets, où une silhouette 2D est décomposée en un ensemble de parties qualitatives, capturées dans un graphe acyclique dirigé. Une "signature" structurelle est définie pour chaque nœud du graphe, qui caractérise la structure sous-jacente du sous-graphe du nœud, dont les composantes sont basées sur les valeurs propres de la matrice de contiguïté du sous-graphe. Tous les bords du graphe sont écartés et le problème est transformé pour trouver la cardinalité maximale et la correspondance de poids minimale dans les graphes bidirectionnels. Toutefois, on ne peut garantir que cette approche se conforme aux structures hiérarchiques de deux graphes. Pour résoudre ce problème, une recherche récursive en profondeur doit être combinée afin d'exploiter la correspondance aux niveaux supérieurs pour limiter la correspondance aux niveaux inférieurs [Sundar et al. 2003]. L'algorithme d'appariement des graphes produit généralement un certain nombre de paramètres qui peuvent être utilisés pour déterminer la "qualité" des résultats de l'appariement par similarité, tels que le nombre de nœuds appariés et des informations sur les nœuds qui sont appariés à d'autres nœuds. En outre, une stratégie de comparaison de graphes grossiers à fins peut également être facilement adoptée.

De plus, Hilaga et al. [Hilaga et al. 2001] ont associé chaque nœud de graphe à plusieurs attributs et ont défini la similarité entre deux nœuds comme étant la similarité

entre leurs attributs. Ensuite, la similarité pour un ensemble donné de paires de nœuds a été calculée comme une mesure de similarité globale.

II.5.3 Méthodes d'apprentissage machine

L'idée principale de la mise en correspondance des similarités basée sur l'apprentissage machine est d'entraîner un classificateur d'apprentissage spécifique pour calculer et classer les degrés de similarité sur un ensemble d'échantillons d'entraînement présélectionnés avec une échelle spécifique en utilisant des méthodes d'apprentissage machine telles que les réseaux neuronaux artificiels (Artificial Neural Networks, ANN) et les machines à vecteurs de support (Support Vector Machines, SVM), etc. Cela est particulièrement approprié dans les cas où aucune mesure de distance appropriée ne peut mesurer efficacement la similarité, par exemple entre deux vecteurs de caractéristiques à haute dimension. Dans ces cas, certaines mesures de similarité appropriées peuvent être approximées en apprenant les corrélations cachées et les mappages à partir d'un certain nombre d'échantillons d'apprentissage connus, ce qui permet une grande flexibilité dans le processus de la recherche.

II.5.3.1 SVM

Les SVM [Vapnik et al. 1999] sont un ensemble de méthodes d'apprentissage supervisées connexes utilisées pour la classification et la régression. En considérant les données d'entrée comme deux ensembles de vecteurs dans l'espace n-dimensionnel, un SVM génère un hyperplan de séparation qui maximise la marge entre les deux ensembles de données. Pour calculer cette marge, deux hyperplans parallèles sont construits, un de chaque côté de l'hyperplan de séparation, qui sont "poussés contre" les deux ensembles de données. Intuitivement, une bonne séparation peut être obtenue par l'hyperplan ayant la plus grande distance par rapport aux points de données voisins des deux classes puisque, en général, plus la marge est grande, plus l'erreur de généralisation du classificateur obtenu est faible. L'idée de base de l'approche SVM peut être décrite comme suit.

Compte tenu de certaines données relatives à l'apprentissage, un ensemble de points ayant la forme suivante :

$$D = \{(x_i, c_i) | x_i \in R^p, c_i \in \{-1, 1\}\}_{i=1}^{n} \quad \text{(II-41)}$$

Où c_i est soit 1 soit -1, indiquant l'une des deux classes auxquelles le point x_i appartient. Chaque xi est un vecteur réel en p-dimension. Le but est de trouver l'hyperplan à marge maximale qui divise les points avec $c_i = 1$ de ceux avec $c_i = -1$. En fait, tout hyperplan peut être écrit comme l'ensemble des points x satisfaisant

$$w \cdot x - b = 0 \quad \text{(II-42)}$$

Où · désigne le produit scalaire entre deux vecteurs. Le vecteur w est un vecteur normal qui est perpendiculaire à l'hyperplan. Le paramètre $b/||w||$ est le décalage de l'hyperplan par rapport à l'origine le long du vecteur normal w. Le but est de choisir w et b pour maximiser la marge, c'est-à-dire la distance entre les deux hyperplans parallèles les plus éloignés possibles tout en séparant les données en deux classes. Ces hyperplans peuvent être décrits par les équations

$$w \cdot x - b = 1 \quad \text{(II-43)}$$

Et

$$w \cdot x - b = -1 \quad \text{(II-44)}$$

Notez que si les données d'entraînement sont linéairement séparables, nous pouvons sélectionner les deux hyperplans de la marge de telle sorte qu'il n'y ait pas de points entre eux, puis essayer de maximiser leur distance. Selon la géométrie, nous pouvons constater que la distance entre ces deux hyperplans est égale à $2/||w||$, ainsi le but est transformé pour minimiser $||w||$. Comme nous devons également empêcher les points de données de tomber dans la marge, il est possible d'ajouter la contrainte suivante : pour chaque i, soit $w \cdot x_i - b \geq 1$ pour x_i dans la première classe ou $w \cdot x_i - b \geq -1$ pour x_i dans la deuxième classe. On a alors

$$c_i(w \cdot x_i - b) \geq 1 \; pour \; \forall 1 \leq i \leq n \qquad \text{(II-45)}$$

Sur la base des descriptions ci-dessus, nous obtenons le problème d'optimisation suivant :

Minimiser (en w, b) : $||w||$, Sous réserve de (pour $\forall 1 \leq i \leq n$) :

$$\text{Sous réserve de (pour } \forall 1 \leq i \leq n \text{)} : c_i(w \cdot x_i - b) \geq 1 \qquad \text{(II-46)}$$

Le problème d'optimisation ci-dessus est très difficile à résoudre car il dépend de $||w||$, la norme de w, qui implique une racine carrée. Heureusement, il est possible de modifier l'équation en remplaçant $||w||$ par $\frac{1}{2}||w||^2$ sans changer la solution optimale, puisque le minimum de l'équation originale et l'équation modifiée ont le même w et b. Il s'agit d'un problème d'optimisation par programmation quadratique (PQ). Plus clairement :

Minimiser (en w, b) : $\frac{1}{2}||w||^2$,

$$\text{Sous réserve de (pour } \forall 1 \leq i \leq n \text{)} : c_i(w \cdot x_i - b) \geq 1 \qquad \text{(II-47)}$$

Notez que le facteur de 0,5 est utilisé pour des raisons de commodité mathématique. Ce problème peut maintenant être résolu par des techniques et des programmes de programmation quadratique standard.

Ibato et al. [Ibato et al. 2002] ont présenté une méthode de recherche de similitude de forme qui combine un objet 3D indépendant de la pose et de la taille du modèle avec le classificateur d'apprentissage basé sur le SVM. En repérant les modèles similaires et dissemblables dans la liste des résultats de recherche précédents, le système apprend les modèles souhaités par l'utilisateur en utilisant l'approche SVM. Ibato et al. ont réalisé de nombreuses expériences en combinant les caractéristiques de forme D2 invariantes par transformation [Osada et al. 2000] avec le SVM, en fournissant le vecteur de caractéristique à un SVM pour calculer la dissimilitude. Les résultats expérimentaux montrent que, malgré sa simplicité, le système fonctionne bien pour récupérer les formes qu'un utilisateur juge "similaires" aux exemples donnés.

II.5.3.2 SOM

SOM (Self-organizing Map) ou SOFM (Self Organizing Feature Map) est un type de réseau neuronal artificiel qui est entraîné en exploitant des méthodes d'apprentissage non supervisées afin de produire une représentation discrète (généralement en 2D) de l'espace d'entrée des échantillons d'entraînement, appelée map. Les SOM sont différents des autres réseaux neuronaux artificiels en ce sens qu'ils adoptent une fonction de voisinage pour préserver les propriétés topologiques de l'espace d'entrée. Cela rend les SOM utiles pour visualiser des vues en basse dimension pour des données en haute dimension, un peu comme une mise à l'échelle multidimensionnelle. Le professeur finlandais Teuvo Kohonen a d'abord décrit le modèle comme un réseau neuronal artificiel, parfois appelé "Kohonen map". Comme la plupart des réseaux neuronaux artificiels, les SOM fonctionnent selon deux modes : l'entraînement et le mappage. Le processus d'entraînement construit la carte

sur la base d'exemples d'entrée, ce qui est un processus compétitif également appelé quantification vectorielle, tandis que le processus de mappage classifie automatiquement un nouveau vecteur d'entrée. Un SOM est constitué de composants appelés nœuds ou neurones. À chaque nœud est associé un vecteur de poids de la même dimension que les vecteurs de données d'entrée, et c'est un point dans l'espace de la carte. La disposition commune des nœuds est un espacement régulier dans une grille hexagonale ou rectangulaire. Le SOM décrit un mappage d'un espace d'entrée de dimension supérieure à un espace mappé de dimension inférieure. La procédure pour placer un vecteur de l'espace de données sur l'espace de la carte consiste à trouver le nœud dont le vecteur de poids est le plus proche du vecteur pris dans l'espace de données et à attribuer les coordonnées cartographiques de ce nœud à notre vecteur. Bien qu'il soit typique de considérer ce type de structure de réseau comme étant lié aux réseaux de feedforward où les nœuds sont visualisés comme étant attachés, ce type d'architecture est fondamentalement différent en termes d'arrangement et de motivation. Les extensions utiles comprennent l'utilisation de grilles toroïdales où les bords opposés sont connectés et utilisent un grand nombre de nœuds. Il a été démontré que si les SOM comportant un petit nombre de nœuds se comportent d'une manière similaire à la méthode de la moyenne K, les SOM plus importants réorganisent les données d'une manière qui est fondamentalement topologique. Il est également courant d'utiliser la matrice U. La valeur de la matrice U d'un nœud particulier est la distance moyenne entre le nœud et ses voisins les plus proches. Dans une grille rectangulaire, par exemple, on peut considérer les 4 ou 8 nœuds les plus proches. Les grands SOMs affichent des propriétés qui sont émergentes. Par conséquent, les grandes cartes sont préférables aux petites. Si le SOM est constitué de milliers de nœuds, il est possible d'effectuer des opérations de regroupement sur la carte elle-même.

L'objectif de l'apprentissage basé sur le SOM est de faire en sorte que les différentes parties du réseau réagissent de manière similaire à certains modèles d'entrée. Ceci est en partie motivé par la façon dont les informations visuelles, auditives ou autres informations sensorielles sont traitées dans des parties distinctes du cortex cérébral dans le cerveau humain. Les poids des neurones sont initialisés soit sous forme de petites valeurs aléatoires, soit échantillonnés de manière égale dans le sous-espace couvert par les deux plus grands composants principaux des vecteurs propres. Évidemment, avec cette dernière alternative, l'apprentissage est beaucoup plus rapide puisque les poids initiaux donnent déjà une bonne approximation des poids SOM. Le réseau doit être alimenté par un grand nombre d'exemples de vecteurs qui représentent, le plus fidèlement possible, les types de vecteurs attendus au cours du processus de mappage. L'entraînement utilise des méthodes d'apprentissage compétitives. Lorsqu'un exemple d'entraînement est fourni au réseau, sa distance euclidienne par rapport à tous les vecteurs de poids est calculée. Le neurone dont le vecteur de poids est le plus similaire à l'entrée est appelé l'unité de meilleure correspondance (Best Matching Unit, BMU). Les poids de la BMU et des neurones proches de l'entrée dans le réseau SOM sont ensuite ajustés en fonction du vecteur d'entrée. L'ampleur de la modification diminue avec le temps et la distance par rapport à la BMU. Dans sa forme la plus simple, la magnitude est de 1 pour tous les neurones suffisamment proches de la BMU et de 0 pour les autres. Une fonction gaussienne est également un choix courant. Quelle que soit la forme fonctionnelle, la fonction de voisinage diminue avec le temps. Au début, lorsque le voisinage est large, l'opération d'auto-organisation se déroule à l'échelle globale. Lorsque le voisinage s'est réduit à quelques neurones seulement, les poids convergent vers des estimations locales. Ce processus est répété pour chaque vecteur d'entrée pendant un grand nombre de cycles. Le réseau enroule les nœuds de sortie associés avec des groupes ou des modèles dans l'ensemble des données d'entrée. Si ces modèles peuvent être nommés, les noms peuvent être attachés aux nœuds associés dans le réseau formé. Au cours du processus de mappage, il y aura un seul neurone gagnant, c'est-à-dire le neurone dont le vecteur de poids est le plus proche du vecteur d'entrée. Cela peut être simplement déterminé en calculant la distance euclidienne entre le vecteur d'entrée et le vecteur de poids. Il convient de noter que tout type d'objet pouvant être représenté numériquement, auquel est associée

une mesure de distance appropriée et dans lequel les opérations nécessaires à l'entraînement sont possibles, peut être utilisé pour construire une SOM.

Pedro et al [Pedro et al. et al 2002] ont décrit un système d'interrogation de bases de données de modèles 3D basé sur la représentation d'images de spin comme signature de forme pour des objets représentés par des mailles triangulaires. La représentation de l'image de spin facilite l'alignement de l'objet interrogé par rapport aux modèles correspondants. La principale contribution de ce travail est l'introduction d'un schéma d'indexation à trois niveaux avec des réseaux de neurones artificiels. Le schéma d'indexation améliore considérablement l'efficacité de la comparaison entre les images de spin de la requête et celles stockées dans la base de données. Leurs résultats sont adaptés à la recherche basée sur le contenu dans les bases de données d'objets 3D. Leur méthode permet à la fois la compression et l'indexation de l'ensemble original d'images de spin. Fondamentalement, un SOM est construit à partir de la pile d'images de spin d'un objet donné. C'est une façon de "résumer" l'ensemble de la pile en un ensemble d'images de spin représentatives. Ensuite, l'algorithme de clustering du noyau K-means est utilisé afin de regrouper les vues représentatives dans la carte SOM en un ensemble réduit de clusters. Au moment de l'interrogation, les images de spin en entrée seront d'abord comparées aux centres des grappes résultant de la méthode K-means et ensuite à la carte SOM si une réponse plus fine est demandée.

II.5.3.3 Apprentissage KNN

En reconnaissance de formes, l'algorithme du plus proche voisin (k-Nearest Neighbor, KNN) est une méthode de classification des objets basée sur des exemples d'entraînement les plus proches dans l'espace des caractéristiques. Le KNN est une sorte d'apprentissage par instance ou d'apprentissage paresseux, où la fonction n'est qu'approximée localement et où tous les calculs sont reportés jusqu'à la classification. KNN peut également être utilisé pour la régression. KNN est l'un des algorithmes d'apprentissage machine les plus simples. Un objet est classé par un vote majoritaire de ses voisins, l'objet étant assigné à la classe la plus courante parmi ses k plus proches voisins. k est un entier positif, généralement petit. Si $k = 1$, alors l'objet est simplement assigné à la classe de son plus proche voisin. Dans les problèmes de classification binaire (c'est-à-dire à deux classes), il est utile de choisir k comme nombre impair pour éviter les votes à égalité. La même méthode peut être utilisée pour la régression en assignant simplement la valeur de propriété de l'objet pour être la moyenne des valeurs de ses k plus proches voisins. Il est utile de pondérer les contributions des voisins, de sorte que les voisins les plus proches contribuent davantage à la moyenne que les plus éloignés. Les voisins sont tirés d'un ensemble d'objets pour lesquels la classification correcte (ou, dans le cas d'une régression, la valeur de la propriété) est connue. Ceci peut être considéré comme l'ensemble d'apprentissage de l'algorithme, bien qu'aucune étape d'apprentissage explicite ne soit nécessaire. Afin d'identifier les voisins, les objets sont représentés par des vecteurs de position dans l'espace multidimensionnel des caractéristiques. En général, la distance euclidienne est adoptée, bien que d'autres mesures de distance, comme la distance de Manhattan, puissent en principe être utilisées au lieu de celle-ci. L'algorithme du plus proche voisin k est sensible à la structure locale des données.

Les exemples de formation sont des vecteurs dans l'espace multidimensionnel des caractéristiques. L'espace est divisé en régions par les emplacements et les étiquettes des exemples d'entraînement. Un point dans l'espace est attribué à la classe c s'il s'agit de l'étiquette de classe la plus fréquente parmi les k échantillons d'entraînement les plus proches. Habituellement, la distance euclidienne est adoptée comme mesure de la distance, mais cela ne fonctionne que pour les valeurs numériques. Dans d'autres cas, par exemple pour la classification des textes, une autre métrique, comme la métrique de chevauchement (ou la distance de Hamming) peut être adoptée. La phase d'apprentissage de l'algorithme consiste uniquement à stocker les vecteurs de caractéristiques et les étiquettes de classe des échantillons d'apprentissage. Au stade de la classification proprement dite, l'échantillon de

test (dont la classe est inconnue) est représenté sous forme de vecteur dans l'espace des caractéristiques. Les distances entre le nouveau vecteur et tous les vecteurs stockés sont calculées et k échantillons les plus proches sont sélectionnés. Il existe de nombreuses façons de classer le nouveau vecteur dans une classe particulière, et l'une des techniques les plus fréquemment utilisées consiste à prédire le nouveau vecteur dans la classe la plus courante parmi les k voisins les plus proches. Le principal inconvénient de cette technique est que les classes avec les exemples les plus fréquents ont tendance à dominer la prédiction du nouveau vecteur, car elles ont tendance à apparaître dans les k plus proches voisins lorsque les voisins sont calculés en raison de leur grand nombre. Une façon d'atténuer ce problème est de considérer la distance de chacun des k plus proches voisins avec le nouveau vecteur à classer et de prédire la classe du nouveau vecteur en fonction de ces distances.

Le meilleur choix de k dépend des données. En général, des valeurs plus élevées de k réduisent l'effet du bruit sur la classification, mais rendent les limites entre les classes moins distinctes. Un k approprié peut être sélectionné par diverses techniques heuristiques, par exemple la validation croisée. Le cas particulier où l'on prévoit que la classe sera celle de l'échantillon d'entraînement le plus proche (c'est-à-dire lorsque k=1) est appelé algorithme du plus proche voisin. La précision de l'algorithme KNN sera fortement dégradée s'il existe des caractéristiques bruyantes ou non pertinentes, ou si les échelles des caractéristiques ne sont pas cohérentes avec leur importance. De nombreux efforts de recherche ont été consacrés à la sélection ou à la mise à l'échelle des caractéristiques afin d'améliorer la classification. Une approche particulièrement populaire consiste à utiliser des algorithmes évolutifs pour optimiser la mise à l'échelle des caractéristiques. Une autre approche populaire consiste à mettre à l'échelle les caractéristiques par l'information mutuelle des données d'apprentissage avec les classes d'entraînement.

Ip et al [Ip et al. 2003] ont proposé une fonction de similarité pondérée pour la classification des modèles CAO, basée sur une représentation sous-jacente des caractéristiques de distribution des formes et un algorithme d'apprentissage KNN. Étant donné un ensemble de modèles solides CAO et de classes correspondantes, la méthode d'apprentissage KNN a été utilisée pour extraire les modèles associés afin de construire automatiquement un classificateur de modèle et d'identifier des classifications nouvelles ou cachées en utilisant la fonction de distribution des formes, en apprenant à partir des exemples d'apprentissage stockés et correctement catégorisés. En outre, les approches probabilistes, telles que le théorème de Bayes, sont également un moyen pratique de comparaison des similarités, dans lequel des probabilités spécifiques de caractéristiques sont calculées et le modèle 3D ayant la plus grande probabilité sera identifié comme le résultat le plus proche de la correspondance [Ansary et al. 2004].

II.5.4 **Mesures sémantiques**

Comme les résultats de la recherche de modèles 3D obtenus par les caractéristiques de bas niveau se sont avérés moins discriminants que prévu, cela soulève une autre question importante, à savoir la mesure sémantique subjective dans la comparaison des similarités. En outre, les utilisateurs jugent également si un modèle 3D récupéré est "pertinent" ou "non pertinent" par rapport à la requête, en fonction de leur perception subjective, liée au contenu sémantique. Par conséquent, il est très important de développer des méthodes de comparaison de similarité sémantique qui prennent en compte la perception humaine dans les systèmes de recherche de modèles 3D basés sur le contenu.

De nombreuses approches qui ont été proposées dans la recherche de médias en 2D pour réduire le "fossé sémantique" tentent d'effectuer une mesure de similarité basée sur une sémantique de haut niveau. Une méthode consiste à apprendre les liens entre un modèle 3D et un ensemble de descripteurs sémantiques, ou les significations sémantiques à partir des caractéristiques du modèle 3D extraites automatiquement. Cette approche est généralement basée sur l'apprentissage machine et la classification statistique, qui regroupe les modèles 3D en catégories sémantiquement significatives à l'aide de caractéristiques de bas niveau,

afin que des méthodes de recherche sémantiquement adaptées puissent être appliquées à différentes catégories. En voici quelques exemples. Suzuki et al [Suzuki et al. 2000] ont construit un mécanisme de mise à l'échelle multidimensionnelle de sorte que les descripteurs sémantiques des mots clés utilisés dans la requête et les caractéristiques des formes calculées à partir des formes 3D étaient fortement corrélés, sur la base d'un ensemble de données d'apprentissage. Le mécanisme d'échelle multidimensionnelle peut analyser des matrices de données similaires ou différentes en représentant les lignes et les colonnes comme un point dans l'espace euclidien, puis mesurer leurs similarités en utilisant les distances euclidiennes. Ils ont ensuite créé un espace de préférences utilisateur spécial selon ce principe, dans lequel une fonction provenant de l'espace du modèle 3D a été construite pour intégrer des mots-clés sémantiques et des formes 3D en tant que représentation de la perception subjective humaine. Zhang et al [Zhang et al. 2001] ont introduit le concept d'"annotation cachée" pour construire un arbre sémantique de l'ensemble de la base de données des modèles 3D. Ils ont utilisé une méthode d'apprentissage actif pour calculer une liste de probabilités pour chaque modèle 3D, qui indiquait la probabilité que le modèle ait un certain attribut sémantique. La liste des probabilités a ensuite été utilisée pour calculer la distance sémantique entre deux modèles, ou entre la requête de l'utilisateur et un modèle de la base de données. La dissimilitude globale entre deux modèles a finalement été déterminée en combinant la somme pondérée de la distance sémantique avec la distance des caractéristiques de bas niveau. Dans [Ibato et al. 2002], une nouvelle mesure sémantique qui pourrait simuler la perception visuelle humaine a également été présentée. Elle a été réalisée en utilisant un classificateur d'apprentissage SVM bien entraîné, construit en effectuant un apprentissage SVM sur les modèles similaires et dissemblables étiquetés dans les résultats de l'étape d'interrogation en cours. Une méthode de regroupement et d'extraction sémantique basée sur le SVM a également été mise en œuvre avec succès dans le système prototype de recherche de formes 3D conçu par l'université de Purdue [Hou et al. 2005].

Une autre méthode efficace consiste à effectuer un feedback sur la pertinence de l'utilisateur après chaque itération de recherche dans la base de données [Elad et al. 2001] [Rui et al. 1998] [Ishikawa et al. 1998].Cela permet de réduire l'écart entre la similarité des caractéristiques de bas niveau et la similarité sémantique de haut niveau [Elad et al. 2001], ce qui permet de mieux saisir ce que l'utilisateur a à l'esprit. Dans une certaine mesure, elle est également considérée comme une méthode de mesure sémantique et a été largement utilisée pour la recherche de médias en 2D [Ishikawa et al. 1998] [Rui et al. 1999]. Dans le cas de la recherche 3D, Leifman et al [Leifman et al. 2004] ont proposé une méthode de feedback de la pertinence combinant le raffinement de la requête et l'extraction de caractéristiques supervisée à chaque étape, qui a tenté de trouver une transformation linéaire optimale qui repondère les composants des caractéristiques de bas niveau afin d'obtenir la séparation maximale de l'ensemble des résultats originaux. Ils ont découvert que cette projection par maximisation d'une fonction de coût est définie comme le critère discriminatoire linéaire de Fisher. Atmosukarto et al [Atmosukarto et al. 2005] ont également présenté un processus de feedback de la pertinence basé sur une mesure de similarité subjective en combinant différentes distances mesurées pour différentes représentations de caractéristiques. Ce processus a été mis en œuvre en calculant le rang entier $r_k(O_i|O_j)$ de l'objet 3D O_i par rapport à l'objet 3D O_j, en se basant sur une méthode d'estimation de la probabilité dans l'espace des caractéristiques des ensembles de résultats "pertinents" et "non pertinents".

II.6 Critères d'évaluation des méthodes de recherche et de classification d'objets 3D

Dans cette section, nous examinons en détail les mesures qui sont couramment utilisées pour évaluer la performance de la recherche et de la classification d'objets 3D. Nous abordons tout d'abord les mesures d'évaluation pour la recherche d'objets 3D qui sont la courbe de rappel-précision, le Nearest Neighbor (NN), le First-Tier (FT), le Second-Tier (ST), le E-measure (E), le décompte de gain cumulé (Discounted Cumulative Gain, DCG),

l'image de distance et Tier image. Ensuite, nous présentons les mesures d'évaluation pour la tâche de classification d'objets 3D, notamment, la précision de la classification, la précision moyenne (mean Average Precision, mAP) et la matrice de confusion.

Graphique de rappel-précision

Un graphique de rappel-précision démontre le comportement de la précision et du rappel dans une liste classée d'objets 3D récupérés. Supposons que la catégorie à laquelle appartient l'objet d'interrogation ait C membres, y compris l'objet en question lui-même, et nous récupérons les K meilleures correspondances. Le rappel est le ratio des objets de la catégorie de la requête qui sont récupérés parmi les K premiers résultats, tandis que la précision est le ratio des K premiers résultats qui appartiennent à la catégorie de la requête. Les résultats de la recherche parfaits doivent donner la plus grande précision (c'est-à-dire 100 %) pour tous les rappels qui peuvent être illustrés par une ligne horizontale en haut du graphique (c'est-à-dire précision = 1:0). Par conséquent, un graphique de rappel-précision décalé vers le haut et vers la droite indique une performance supérieure.

Nearest Neighbor (NN)

La métrique de NN est le pourcentage des correspondances les plus proches qui appartiennent à la même catégorie d'interrogation, c'est-à-dire que pour chaque objet 3D de l'ensemble de données, le deuxième meilleur résultat (évidemment, le meilleur résultat est une correspondance avec l'interrogation elle-même) est vérifié s'il s'agit d'un membre de la même catégorie que celle à laquelle appartient l'objet interrogé. Le score idéal est certainement de 100% et le score le plus élevé indique les meilleurs résultats.

First-Tier (FT) et Second-Tier (ST)

La métrique du FT est le pourcentage d'objets 3D appartenant à la catégorie de la requête qui sont retrouvés dans les premiers résultats $C-1$, où la catégorie de la requête a C membres. Le rappel pour la métrique du ST est deux fois plus important que pour la métrique du FT, c'est-à-dire le pourcentage d'objets 3D appartenant à la catégorie de la requête qui sont retrouvés dans les $2(C-1)$ premiers résultats. Il est évident que le score idéal pour les deux mesures est de 100 % et que les valeurs les plus élevées représentent de meilleurs résultats, tandis que le score le plus élevé est plus susceptible d'apparaître pour la mesure du ST car les membres de la catégorie de la requête ont plus de chances d'être retrouvés parmi les meilleurs résultats.

E-measure (EM)

Cette mesure est obtenue lorsque la précision et le rappel sont calculés pour les 32 premières correspondances de la liste de classement (c'est-à-dire K = 32). La mesure E est définie comme suit :

$$E = \frac{2}{\frac{1}{P}+\frac{1}{R}} \qquad \text{(II-48)}$$

Où P et R sont respectivement la précision et le rappel. La valeur maximale de cette mesure est de 1:0 (ou équivalent à 100% en termes de pourcentages) et les scores les plus élevés indiquent les meilleurs résultats.

Le décompte de gain cumulé (Discounted Cumulative Gain (DCG))

Cette mesure pèse davantage les résultats pertinents en tête de liste que les résultats pertinents en fin de liste. L'intuition est que les résultats des premières pages sont plus intéressants pour un utilisateur d'un moteur de recherche que ceux des pages suivantes. Cette mesure a des scores allant de 0% à 100% et le score le plus élevé indique la meilleure performance de recherche.

La contribution du $k^{ième}$ objet retourné, notée G_k, est égale à 0 si cet objet n'appartient pas à la classe de la requête, et est égale à $\frac{1}{\log_2(k)}$ dans le cas contraire. Les

équations suivantes donnent les formules du DCG normalisées d'un modèle M_i^j, d'une classe C_i et de la base de données B:

$$DCG\left(M_i^j\right) = \frac{1 + \sum_{k=2}^{N} G_k}{1 + \sum_{k=1}^{NC_i - 1} \frac{1}{\log_2(k+1)}} \quad \text{(II-49)}$$

$$DCG(C_i) = \frac{1}{NC_i} \sum_{i=1}^{NC_i} DCG(M_i^j) \quad \text{(II-50)}$$

$$DCG(B) = \frac{1}{N_c} \sum_{i=1}^{NC_i} DCG(C_i) \quad \text{(II-51)}$$

Image de distance

Une image de la matrice de distance où la luminosité de chaque pixel (i, j) est proportionnelle à la magnitude de la distance entre les objets i et j [Osada et al. 2001]. Les objets sont regroupés par classe le long de chaque axe, et des lignes sont ajoutées à des classes séparées, ce qui permet d'évaluer qualitativement les modèles dans les résultats de correspondance - c'est-à-dire que le résultat optimal est un ensemble de blocs de pixels les plus sombres, de la taille de la classe, le long de la diagonale, indiquant que chaque objet correspond mieux aux objets de sa classe qu'à ceux des autres classes. Sinon, les raisons des mauvais résultats de correspondance sont souvent visibles dans l'image - par exemple, des blocs de pixels sombres hors diagonale indiquent que deux classes se correspondent bien.

Tier image

Une image visualisant les correspondances du voisin le plus proche, le First Tier et Second Tier [Osada et al. 2001]. Plus précisément, pour chaque ligne représentant une requête avec l'objet j dans une classe avec $|C|$ membres, le pixel (i, j) est : (a) blanc si l'objet i est l'objet j ou son plus proche voisin, (b) jaune si l'objet i est parmi les $|C| - 1$ premières correspondances (First Tier), et orange si l'objet i est parmi les $2 * (|C| - 1)$ premières correspondances (Second Tier). Comme pour l'image de distance, les objets sont regroupés par classe le long de chaque axe, et les lignes sont ajoutées à des classes séparées. Cette image est souvent plus utile que l'image de distance car les meilleures correspondances sont clairement indiquées pour chaque objet, quelle que soit l'importance de leurs valeurs de distance. Le résultat optimal est un ensemble de blocs de pixels blanc/jaune de la taille d'une classe le long de la diagonale, indiquant que chaque objet correspond mieux aux objets de sa classe qu'à ceux des autres classes. Sinon, un plus grand nombre de pixels colorés dans les blocs de la taille de la classe le long de la diagonale représente un meilleur résultat.

Précision de la classification

Une autre mesure intuitivement intéressante est la précision de la classification, qui est une statistique sommaire qui peut être facilement calculée à partir de la matrice de confusion comme étant le nombre total d'instances correctement classées (c'est-à-dire les éléments diagonaux de la matrice de confusion) divisé par le nombre total d'instances d'essai. Alternativement, la précision d'un modèle de classification sur un ensemble de tests peut être définie comme suit :

$$Précision = \frac{Nombre\ de\ classifications\ correctes}{Nombre\ total\ d'exemples\ de\ tests} \quad \text{(II-52)}$$

Précision moyenne (mean Average Precision (mAP))

La métrique mAP est définie comme suit :

$$mAP = \sum_{K} P(K)R(K) \quad \text{(II-53)}$$

Où la précision et le rappel sont calculés pour toutes les valeurs de K. Intuitivement, mAP est considéré comme la zone située sous le graphique précision-rappel. Un algorithme de recherche parfait a mAP = 100% et une valeur plus élevée indique de meilleurs résultats.

La matrice de confusion

La performance d'un classificateur est généralement évaluée via la matrice de confusion, qui affiche le nombre de prédictions correctes et incorrectes faites par le classificateur par rapport aux classifications réelles dans l'ensemble de test. La matrice de confusion montre comment les prédictions sont faites par le modèle. Les lignes correspondent à la classe réelle (vraie) des données (c'est-à-dire les étiquettes dans les données), tandis que les colonnes correspondent à la classe prédite (c'est-à-dire les prédictions faites par le modèle). Lorsqu'une instance est classée, c'est la même chose que de faire une prédiction que l'instance est correctement classée. Les éléments de la matrice de confusion pour le problème de classification binaire (à deux classes) sont les suivants :

- TP (true positives) est le nombre de cas positifs correctement classés ;
- FP (false positives) est le nombre de cas négatifs incorrectement classés comme positifs ;
- FN (false negatives) est le nombre de cas positifs incorrectement classés comme négatifs ;
- TN (true negatives) est le nombre de cas négatifs correctement classés ;

La valeur de chaque élément dans la matrice de confusion est le nombre de prédictions faites avec la classe correspondant à la colonne pour les instances (exemples) ayant la valeur correcte telle que représentée par la ligne. Ainsi, les éléments diagonaux indiquent le nombre de classifications correctes faites pour chaque classe, et les éléments hors diagonale indiquent les erreurs commises.

II.7 Conclusion

Dans ce chapitre, nous avons discuté plusieurs sujets pour la recherche et la classification d'objets 3D basée sur le contenu, notamment le prétraitement, l'extraction de caractéristiques, la correspondance de similarité et les critères d'évaluation des méthodes de recherche et de classification d'objets 3D. Puisque l'extraction de caractéristiques de forme est une étape importante dans le processus de recherche et de classification d'objet 3D, nous avons essayé de fournir une classification et un résumé rationnels et compréhensibles de la littérature de recherche existante. Actuellement, la plupart des travaux sur l'extraction de caractéristiques de forme mettent l'accent sur les propriétés géométriques et topologiques de surface des caractéristiques de forme 3D, en se basant sur les surfaces, les voxels, les ensembles de sommets et les formes structurelles des modèles. En général, les caractéristiques géométriques représentent la forme spécifique et la position spatiale des surfaces, des arêtes et des sommets, tandis que les caractéristiques topologiques maintiennent la relation de liaison entre les surfaces, les arêtes et les sommets.

La caractéristique commune des méthodes basées sur l'analyse géométrique globale est qu'elles sont presque toutes dérivées directement de l'unité élémentaire d'un modèle 3D, c'est-à-dire le sommet, le polygone ou le voxel, et un modèle 3D est visualisé et traité comme un ensemble de sommets, un ensemble de mailles de polygones ou un ensemble de voxels. Leurs avantages résident dans leur facilité de dérivation directe à partir de structures de données 3D, ainsi que dans leur pouvoir de représentation relativement bon. Cependant, les processus de calcul sont généralement trop longs et trop sensibles pour les petites caractéristiques. De plus, les exigences de stockage sont trop élevées en raison de la

difficulté à mettre en place un mécanisme d'indexation concis et efficace pour ces objets dans les grandes bases de données de modèles 3D.

Les méthodes basées sur la représentation sphérique produisent des caractéristiques de forme invariantes, ce qui évite le long processus de normalisation des coordonnées canoniques dans l'extraction des caractéristiques. Cependant, elles présentent également quelques inconvénients. Tout d'abord, on suppose généralement qu'un modèle 3D aura une topologie valide (pour les maillages), ou un volume explicite (pour les modèles volumétriques), ce qui ne peut être garanti dans la pratique. Deuxièmement, le processus de mappage des fonctions sphériques est compliqué et prend beaucoup de temps.

Comme il est possible d'extraire beaucoup plus de caractéristiques pour une forme 2D, les méthodes de mappage de fonctions rendent le processus d'extraction plus flexible. Elles peuvent également réduire considérablement la complexité du calcul des caractéristiques et rendre le descripteur de caractéristique plus compact. Toutefois, cela entraîne inévitablement une perte importante d'informations 3D, car le processus de mappage des fonctions est limité par différentes contraintes. En outre, pour la représentation des vues planes en 2D, la détermination du nombre nécessaire de vues de projection en 2D pose un autre problème dans la pratique.

De nombreux descripteurs de caractéristiques de forme statistiques sont simples à calculer et utiles pour conserver des propriétés invariantes. Dans de nombreux cas, ils sont également résistants au bruit, ou aux petites fissures et trous qui existent dans un modèle d'objet 3D. Malheureusement, comme inconvénient inhérent à la représentation d'un histogramme, ils ne permettent qu'une discrimination limitée entre les objets : Ils ne préservent ni ne construisent d'informations spatiales. Ainsi, ils ne sont souvent pas assez discriminants pour faire de petites différences entre des formes 3D dissemblables et ne distinguent généralement pas les différentes formes ayant le même histogramme.

Les caractéristiques topologiques et de forme squelettique sont intéressantes pour la recherche en 3D car elles permettent de capturer les structures de forme significatives d'un objet 3D. En même temps, elles sont relativement élevées et proches de la perception intuitive de l'homme, ce qui les rend utiles pour définir une représentation plus naturelle des requêtes 3D. Ils peuvent également effectuer des tâches de correspondance partielle en contenant des propriétés structurelles à la fois locales et globales. Cependant, les modèles 3D ne sont pas toujours suffisamment bien définis pour être facilement et naturellement décomposés en un ensemble canonique de caractéristiques ou de formes de base. En outre, le processus de décomposition est généralement coûteux en termes de calcul. En outre, les processus de décomposition des modèles sont assez sensibles au bruit pour les petites perturbations du modèle. Ainsi, un effort supplémentaire est, à son tour, nécessaire pour les gérer. Enfin, par rapport aux algorithmes d'indexation et d'appariement des similarités relativement simples basés sur des vecteurs de caractéristiques numériques, les algorithmes d'indexation et d'appariement des représentations sous forme de graphes sont relativement plus complexes et plus longs, en raison des processus de recherche de graphes nécessaires. Et, comme il n'existe actuellement aucune solution universelle de correspondance de graphes à usage général, différents algorithmes de correspondance de graphes doivent être conçus pour s'adapter à différentes représentations de graphes.

Enfin, les performances des méthodes d'apprentissage approfondi sur différentes représentations de données 3D ont été examinées. La principale force de ces approches réside dans leur capacité à apprendre progressivement les caractéristiques hiérarchiques discriminantes des données d'entrée. Ces dernières peuvent avoir différentes représentations dont la structure et les propriétés géométriques varient d'une représentation à l'autre. En se basant sur la catégorisation des différentes représentations d'objets 3D, l'importance de choisir une représentation de données 3D appropriée qui dépend de la simplicité, de la facilité d'utilisation et de l'efficacité a été mise en évidence. On peut donc conclure qu'un apprentissage approfondi associé à une représentation de données 3D appropriée constitue

une approche efficace pour extraire les caractéristiques d'un modèle 3D, et permet d'obtenir des résultats impressionnants dans de nombreuses tâches telles que la classification, la recherche, la segmentation, la détection et la localisation, etc.

Après l'état de l'art fait sur les approches de description de la forme d'objets 3D, il nous a semblé intéressant de proposer des approches de classification et de recherche basées sur une représentation qui permet, à la fois, de profiter de la richesse des données 3D, faire face aux problèmes liés à l'imperfection des objets 3D, et surtout offre la possibilité de la correspondance partielle. C'est pour ces différentes raisons que nous introduisons, dans la suite de ce manuscrit, des approches de recherche et de classification basées sur les images de coupe 2D permettant de représenter de façon pertinente la forme d'objets 3D ainsi qu'une métrique de mesure de similarité adaptée à la correspondance partielle de ces derniers.

Chapitre II : Recherche d'objets 3D basée sur les images de coupe 2D et des algorithmes d'exploration de données

III.1 Introduction

Dans ce chapitre, nous introduisons nos trois approches d'indexation et de recherche d'objets 3D ainsi que les concepts associés.

La deuxième section passe en revue la littérature de recherche dans le domaine de l'Extraction de Connaissances à partir des Données (ECD) et de l'exploration de données, en particulier les algorithmes de partitionnement des données (Clustering) et l'apprentissage des règles d'association, une forme d'apprentissage et de classification non supervisée. L'objectif de cette recherche est de tirer parti de la puissance des algorithmes de partitionnement et d'exploration des données pour sélectionner les images de coupe 2D représentatives de l'objet 3D.

Dans la troisième section, nous présentons notre première approche K_RS (K Representative Slices), qui commence par extraire, pour chaque objet 3D, un ensemble d'images de coupe 2D correspondant à ses trois axes principaux, puis l'algorithme de Clustering K-means est utilisée pour sélectionner les images de coupe 2D représentatives, ce qui transforment la comparaison entre les objets 3D en un calcul de similarité entre leurs images de coupe 2D représentatives. Cette approche donne des résultats satisfaisants si le nombre de clusters est correctement choisi. Dans le cas contraire, l'étape de Clustering génère une sur-partition ou une sous-partition ce qui affecte la performance de l'approche. Afin de remédier à ce problème, nous présentons dans la quatrième section notre deuxième approche ASC (Adaptive Slices Clustering) qui utilise un indice de validité de cluster pour adapter le nombre d'images de coupe 2D à la complexité de chaque objet 3D. Dans la section 5, nous exposons notre troisième méthode qui commence par extraire un nombre important des images de coupe 2D, et par la suite, utilise l'algorithme Apriori pour réduire cet ensemble initial des images de coupe afin d'en garder que les plus importantes.

III.2 Extraction de Connaissances à partir des Données (ECD) et exploration de données

L'Extraction de Connaissances à partir des Données (ECD) est le processus non trivial d'identification de modèles de données valables, nouveaux, potentiellement utiles et finalement compréhensibles [Fayyad et al. 1996]. Le but du processus de l'ECD est de recueillir des informations à partir de grands ensembles de données. La figure III-1 donne un aperçu des étapes de la collecte d'informations et de connaissances à partir des sources de données [Fayyad et al. 1996]. Le processus de l'ECD comprend plusieurs étapes : sélection, prétraitement, transformation, exploration des données et interprétation/évaluation. L'extraction de règles d'association est une des applications d'exploration de données permettant d'extraire des modèles dans les données.

L'exploration de données constitue l'une des étapes du processus de l'ECD. L'objectif de l'étape d'exploration des données est d'identifier des modèles qui peuvent ensuite être interprétés et permettre de prendre des décisions plus éclairées. Les auteurs déclarent que l'exploration de données est une étape du processus de l'ECD qui consiste à appliquer des algorithmes d'analyse et de découverte de données qui, dans des limites d'efficacité informatique acceptables, produisent une énumération particulière de modèles sur les données.

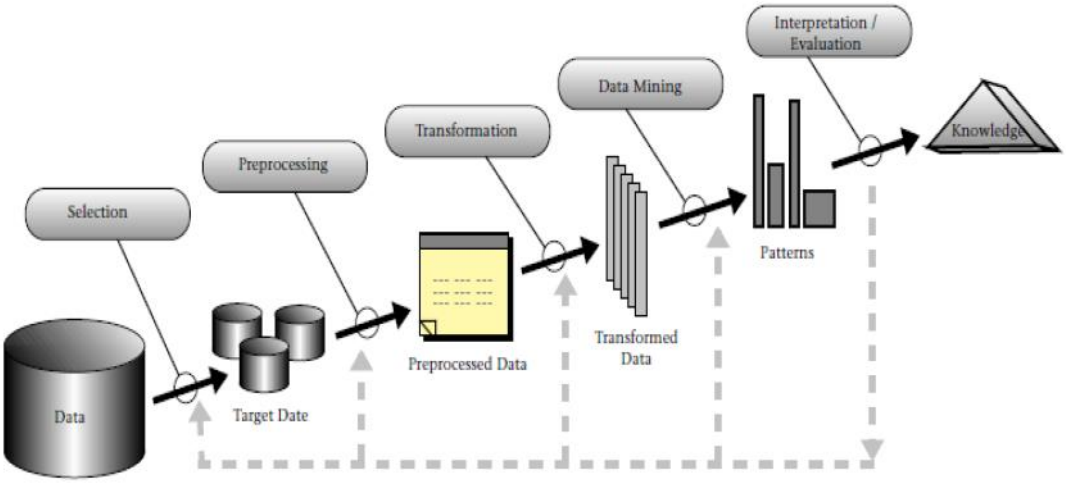


Figure III-1 : Aperçu du processus ECD [Fayad et al. 1996]

Dans l'étape d'exploration des données du processus de découverte des connaissances, il existe trois formes principales d'algorithmes d'apprentissage, à savoir les algorithmes d'apprentissage supervisé, non supervisé et par renforcement.

Dans l'apprentissage supervisé, les données fournies au modèle ont des étiquettes de classe connues qui sont les résultats corrects correspondants. Les algorithmes d'apprentissage supervisé créent une fonction qui modélise les données en utilisant des instances de données d'entrainement et la fonction est ensuite appliquée pour prédire le résultat de données inédites. La fonction créée sur les données d'entrainement est utilisée pour classer les nouvelles instances de données inédites. Parmi les exemples réels d'apprentissage supervisé, citons la reconnaissance d'image [He et al. 2016], la détection du spam [Androutsopoulos et al. 2000], les modèles de prévision par défaut dans les services financiers [Atiya, 2001], la prévision du cancer dans les services de santé [Shipp et al. 2002] et la reconnaissance vocale [Hinton et al. 2012]. Les algorithmes d'exploration de données utilisés pour effectuer des tâches d'apprentissage supervisées comprennent la régression logistique [McCullagh 1984], les arbres de décision [Quinlan 1986], les forêts aléatoires [Breiman 2001], les machines à vecteurs de support [Cortes et al. 1995] et les réseaux de neurones.

Une autre forme de fouille de données est l'apprentissage par renforcement [Barto et al. 1997]. L'apprentissage par renforcement consiste à apprendre ce qu'il faut faire et à faire correspondre les situations aux actions nécessaires. Dans l'apprentissage par renforcement, l'apprenant ne se fait pas dire ce qu'il doit faire, mais doit plutôt apprendre quelle action lui rapporte le maximum de récompense. Un exemple d'apprentissage par renforcement consiste à apprendre à un agent à jouer à des jeux informatiques tels que Super Mario ou Pac Man. Un exemple d'algorithme d'apprentissage par renforcement est le Q-learning [Watkins et al. 1992]. Dans le Q-learning, l'objectif est d'atteindre l'état avec la plus grande récompense, de sorte que si l'apprenant arrive à l'objectif, il y restera indéfiniment. Dans l'apprentissage par renforcement, ce type d'objectif est appelé un objectif absorbant.

L'apprentissage non supervisé est appliqué lorsque les instances de données ne sont pas étiquetées. En appliquant ces algorithmes non supervisés, les chercheurs espèrent découvrir des classes d'éléments inconnues, mais utiles [Jain et al. 1999]. Parmi les

algorithmes d'apprentissage non supervisé les plus étudiés figurent le Clustering, l'apprentissage de règles d'association et la détection d'anomalies.

Des recherches considérables ont été menées sur les techniques d'apprentissage supervisé pour prédire les classes, y compris des modèles tels que les arbres de décision et les approches de réseaux de neurones. Des études [Liu et al. 1997] [Li et al. 2001] proposent l'utilisation de règles d'association non supervisées à des fins de classification en utilisant un ensemble de règles d'association de haute qualité pour faire les prédictions de classe.

Le reste de cette section se focalise sur les algorithmes d'apprentissage non supervisé utilisés dans nos approches d'indexation et de recherche d'objets 3D, à savoir : le Clustering et l'apprentissage de règles d'association.

III.2.1 Les algorithmes de Clustering

Le Clustering consiste à affecter un ensemble d'objets à des groupes, les clusters. Les objets affectés à un même cluster sont similaires en termes de distance ou de métrique de dissimilarité.

La mesure de distance utilisée est la distance euclidienne. Cette tâche de clustering peut être mise en œuvre de différentes manières, ce qui permet d'obtenir un large ensemble d'algorithmes de clustering distincts. Les études de [Berkhin et al. 2002] et de [Xu et al. 2005] examinent un grand nombre de ces algorithmes, mais dans cette section, nous nous concentrons sur les familles d'algorithmes les plus couramment utilisées.

III.2.1.1 Le Clustering basé sur les centroïdes

Dans la famille des algorithmes de clustering basés sur les centroïdes, les clusters résultants sont représentés par un vecteur central, qui peut faire partie de l'ensemble de données (un médoïde) ou non (un centroïde). Les autres objets de l'ensemble de données sont affectés au centre de cluster le plus proche.

L'algorithme de K-means [Hartigan et al. 1979] est l'exemple classique d'algorithme de clustering basé sur les centroïdes. Il consiste à diviser l'ensemble des points de données en k clusters C_j . Chaque cluster est représenté par la valeur moyenne (ou moyenne pondérée) c_j, le centroïde. La somme des écarts entre un point et son centroïde est utilisée comme fonction objective dans le schéma d'optimisation itératif. La distance habituelle à minimiser est la distance euclidienne.

L'algorithme exige de l'utilisateur qu'il fournisse le nombre de clusters k souhaité, et le schéma d'optimisation est composé de trois étapes simples, comme le montre la figure III-2 :

1. Calculer les centroïdes. Dans la première itération, ils peuvent être calculés de manière aléatoire ou par plusieurs autres méthodes. Dans les itérations suivantes, les centroïdes sont la valeur moyenne des points attribués à chaque cluster.

2. Calculer les distances de chaque point par rapport à tous les centroïdes.

3. Regroupez les points au centroïde le plus proche, pour minimiser les écarts.

Lorsque chaque itération est terminée, l'algorithme vérifie s'il y a eu un changement dans les points attribués à chaque cluster. Si les clusters n'ont pas changé, l'algorithme se termine.

La figure III-3 illustre ce processus. La première étape est l'élection des centroïdes initiaux, les points de forme ronde. Les étapes 1 et 2 constituent le noyau itératif de l'algorithme, où les points sont attribués au centroïde le plus proche, puis les centroïdes sont mis à jour. L'algorithme converge lorsque les objets assignés à chaque cluster ne changent pas au cours de deux itérations consécutives, dernière étape.

La sélection de k est l'un des principaux inconvénients des algorithmes de K-moyennes/médoïdes, car il n'est pas toujours possible de deviner le nombre de clusters différents présents dans l'ensemble de données. L'algorithme X-means [Pelleg et al. 2000] s'attaque à ce problème en appliquant les critères d'information bayésiens (Bayesian Information Criterion, BIC) qui évaluent la qualité des clusters en fonction de leur représentation des distributions gaussiennes. Fondamentalement, X-means exécute itérativement K-means en augmentant la valeur de k, en vérifiant à chaque itération si les clusters résultants augmentent le score BIC.

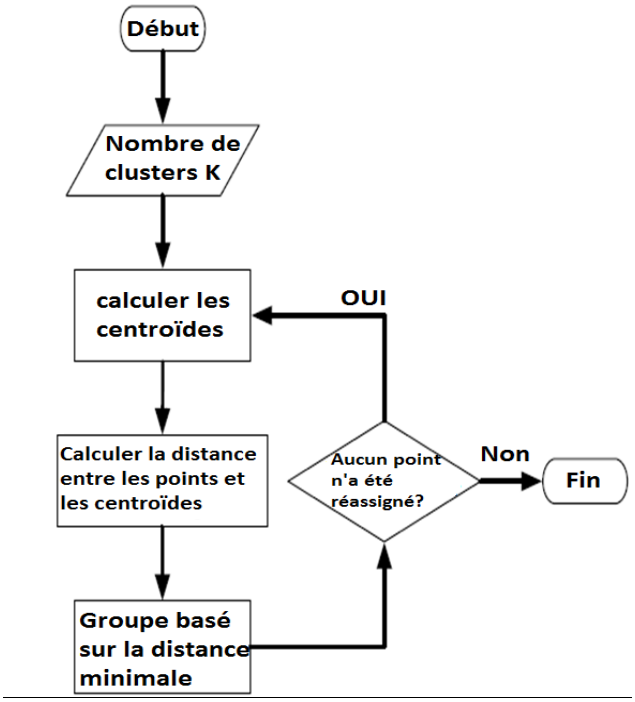


Figure III-2 : Schéma d'optimisation de l'algorithme de K-means

L'algorithme équivalent, mais utilisant des éléments réels dans l'ensemble de données, les médoïdes, est appelé K-medoids, la partition autour des médoïdes (Partition Around Medoids (PAM)) [Kaufman et al. 1990] étant l'implémentation de référence.

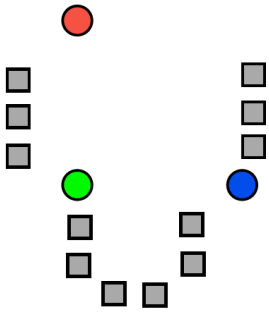
Les algorithmes basés sur les centroïdes sont, de loin, les algorithmes de clustering les plus utilisés. Le principal avantage de ces algorithmes est la facilité d'implémentation et la rapidité d'exécution.

Malheureusement, ces algorithmes manquent de certains aspects importants : ils ne sont pas robustes contre les valeurs aberrantes et ils supposent une structure hyper-sphérique de la distribution des données.

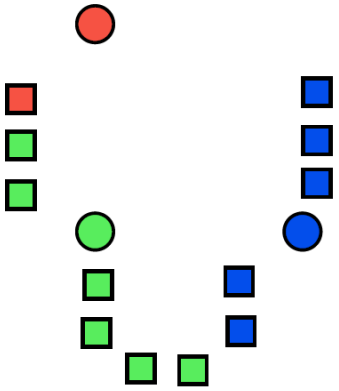
III.2.1.2 Le clustering basé sur la connectivité

L'objectif du clustering basé sur la connectivité [Joe et al. 1963], également appelé regroupement hiérarchique, est la construction d'une hiérarchie d'individus dans un ensemble de données. Cette hiérarchie est représentée par un dendrogramme, un arbre dont les feuilles sont les individus et la racine représente l'ensemble des données. Les nœuds intermédiaires représentent des groupes de deux ou plusieurs individus dont la hauteur par rapport aux feuilles exprime la valeur d'une métrique de liaison nécessaire pour conformer un tel groupe. Cette métrique de liaison est basée sur une fonction de dissimilarité et peut être calculée de différentes manières : la liaison simple utilise la valeur minimale de la

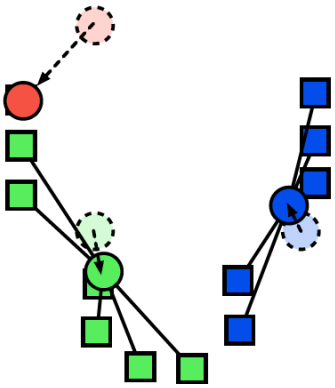
métrique de dissimilarité entre deux points/groupes ; ou la liaison moyenne, la moyenne de la distance ; la liaison complète utilise la valeur maximale de la métrique de dissimilarité.



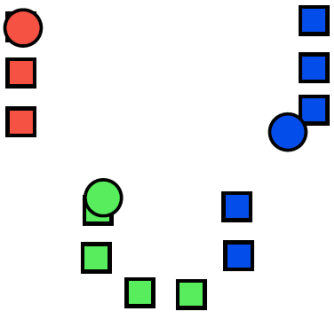
Etape 1 : génération des premiers centriodes



Etape 2 : attribuer les points aux centriodes les plus proches



Etape 3 : recalculer les centriodes



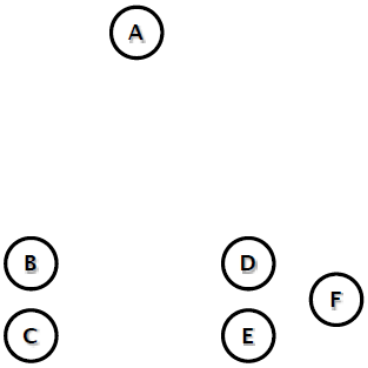
Fin : état de convergence

Figure III-3: Exemple graphique de l'algorithme de K-means utilisant k = 3. Après avoir généré les centroïdes initiaux à l'étape 1, les points de l'ensemble de données sont attribués au plus proche à l'étape 2 et les centroïdes sont recalculés à l'étape 3. Les étapes 2 et 3 sont exécutées jusqu'à ce que l'algorithme converge.

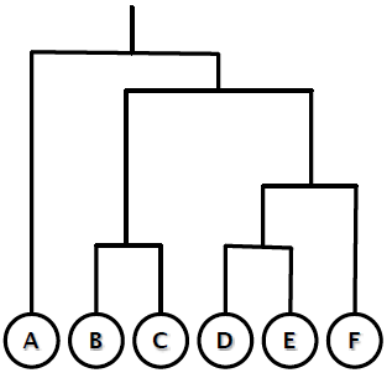
La figure III-4 contient un exemple d'un petit ensemble de données, la figure III-4-a, et le dendrogramme résultant obtenu avec un seul lien de la distance euclidienne, la figure III-4-b. Les points B et C, et D et E, ont la même distance, de sorte qu'ils se confondent à la même hauteur dans le dendrogramme. Ensuite, le groupe de D et E fusionne avec F à une hauteur légèrement supérieure. Le reste du dendrogramme exprime une fusion du groupe formé par B et C et du groupe formé par D, E, F. Enfin, l'ensemble des données est fusionné au niveau supérieur.

La stratégie de construction du dendrogramme donne lieu à deux types de regroupement hiérarchique : agrégatif, une approche ascendante, qui fusionne les individus/groupes de la feuille à la racine ; ou divisif, une approche descendante, qui sépare les individus/groupes de la racine aux feuilles.

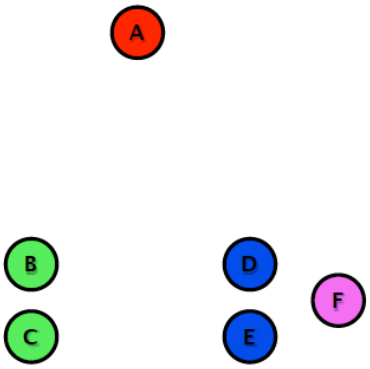
Pour obtenir une partition de données dans un regroupement hiérarchique, il est nécessaire d'effectuer une coupe horizontale à une certaine hauteur dans le dendrogramme. La figure III-4 donne un exemple de deux partitions différentes : le graphique III-4-c montre une partition obtenue en coupant le dendrogramme à la hauteur indiquée en figure III-4-d; de même, le graphique III-4-e correspond aux clusters obtenus en coupant aux hauteurs indiquées en figure III-4-f. On peut voir qu'une coupe à proximité des feuilles produit un grand nombre de clusters plus compacts. En revanche, une coupe proche de la racine du dendrogramme produit moins de clusters avec plus de variabilité.



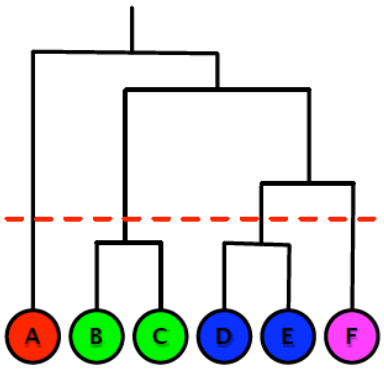
(a) Données d'entrée



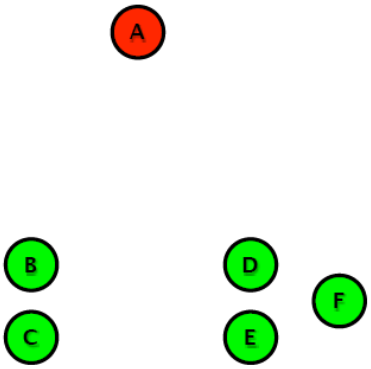
(b) dendogramme resultant



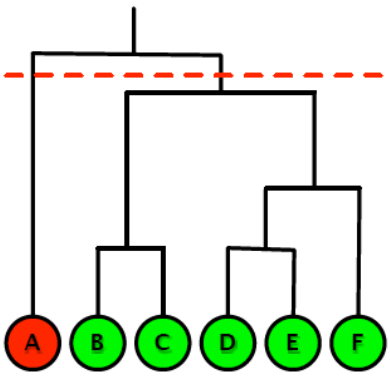
(c) Clusters résultants



(d) Coupe de dendogramme



(e) Clusters resultants



(f) Coupe de dendogramme

Figure III-4 : Exemple graphique de construction de dendrogrammes utilisés dans un regroupement hiérarchique et les partitions possibles obtenues à partir de coupes de hauteurs différentes.

Contrairement au clustering basé sur les centroïdes, où les algorithmes génèrent des clusters gaussiens autour d'un centroïde/médoïde, le clustering basé sur la connectivité ne suppose pas le modèle sous-jacent des données. En d'autres termes, la hiérarchie de construction des individus selon la métrique de dissimilarité est orthogonale à la façon dont les individus sont distribués. D'autre part, décider quel niveau du dendrogramme exprime la division la plus valable des données pour effectuer la coupe est une tâche difficile. Ce problème s'aggrave lorsqu'il s'agit de grands ensembles de données en raison de la difficulté à représenter et à analyser le dendrogramme

III.2.1.3 Le clustering basé sur la densité

Les algorithmes de clustering basés sur la densité sont des algorithmes de partition, comme K-means, mais ils partagent certaines caractéristiques avec le clustering basé sur la connectivité. Le but du clustering basé sur la densité est de regrouper des points qui sont liés par une propriété de connectivité particulière et dont la densité est suffisamment grande pour être considérée comme un véritable cluster. Ces algorithmes sont largement utilisés dans le domaine de la reconnaissance d'images.

DBSCAN [Ester et al. 1996] est un exemple classique d'algorithme basé sur la densité. Les entrées de DBSCAN sont deux paramètres, le rayon Epsilon (Eps) et le nombre minimum de points (MinPoints). Cet algorithme est basé sur deux définitions de base :

1. Un point p est directement accessible en densité à partir d'un point q si sa distance est inférieure ou égale à Eps, et le voisinage Eps de q (voisins à une distance inférieure ou égale à Eps) est supérieur ou égal à MinPoints. Cette relation n'est pas symétrique.

2. Deux points p et q sont atteignables par densité s'il existe une séquence $p_1, \dots, p_n$, où $p = p_1$ et $q = p_n$ où chaque p_{i+1} est directement atteignable par densité à partir de p_i.

3. Deux points p et q sont reliés par la densité s'il existe un point o tel que p et q sont tous deux des densités atteignables à partir de o.

Les clusters obtenus sont les sous-ensembles C_i des données où tous les points sont reliés entre eux par densité. Les points qui ne font pas partie d'un cluster sont considérés comme du bruit. Notez que les définitions conduisent à deux types de points à l'intérieur d'un cluster : les points centraux, points intérieurs d'un cluster qui servent à remplir la connectivité de densité à travers tous les points du cluster, et les points de frontière, dans les bords du cluster, d'où il n'y a pas de points directement accessibles par la densité. Ces définitions sont clarifiées dans la figure III-5 : les points rouges sont les points centraux du cluster formés par les points rouges et jaunes ; les points jaunes sont les points de frontière ; les flèches représentent la relation directe d'accessibilité de la densité et les circonférences colorées sont la portée du voisinage Eps ; le point bleu est un point de bruit qui n'appartient pas au cluster.

En tant que regroupement hiérarchique, les algorithmes de clustering basés sur la densité ne tiennent pas compte de la structure ou du modèle des données. En outre, ils sont robustes contre les valeurs aberrantes et le bruit. Un inconvénient important est le manque d'interprétabilité des clusters résultants.

III.2.2 Apprentissage des règles d'association

L'apprentissage des règles d'association, une forme de modélisation des dépendances, examine l'ensemble des données pour déterminer les relations entre les variables ou les éléments. L'application classique des algorithmes de règles d'association s'inscrit dans le contexte des transactions d'achat dans les magasins de détail et des articles faisant partie de ces transactions [Agrawal et al. 1993]. L'analyse des règles d'association dans les bases de données des magasins de détail est plus connue sous le nom d'analyse du panier de la ménagère. Le concept général de l'apprentissage des règles d'association

consiste à identifier des règles telles qu'un client qui achète un produit A achète également un produit B avec un niveau de confiance identifiable. Un autre domaine dans lequel les règles d'association ont été utilisées est la recherche médicale pour identifier les patients à haut risque [Obenshain 2004] et l'identification précoce des infections [Brossette et al. 1998].

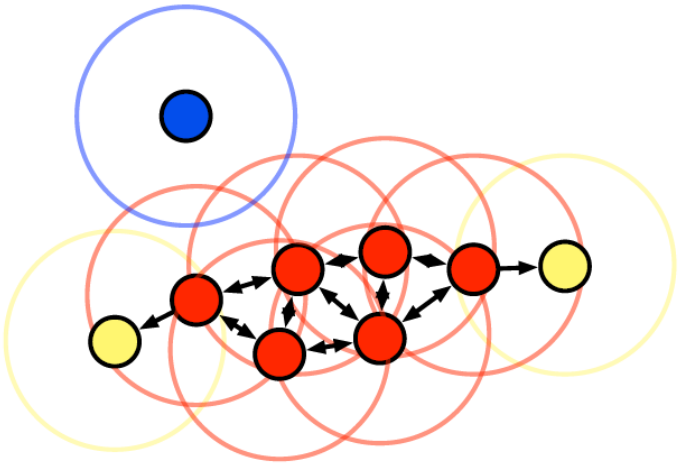


Figure III-5 : Exemple graphique d'un cluster résultant en utilisant le clustering DBSCAN basé sur la densité. Les points jaunes sont les points de bordure, les points rouges sont les points centraux. Le point bleu est un point de bruit.

Lors de l'application d'algorithmes de règles d'association, l'objectif est l'exhaustivité, l'algorithme est nécessaire pour trouver toutes les règles intéressantes dans l'ensemble de données. La difficulté de l'extraction des règles d'association est la taille et la complexité du problème. Le nombre de règles possibles dans l'ensemble de données augmente de manière exponentielle avec le nombre d'éléments. Les algorithmes développés pour l'extraction de règles d'association tentent de réduire ce niveau de complexité et de fournir des résultats rapides à partir des modèles développés.

L'idée d'appliquer l'extraction par règles d'association à l'analyse du panier de consommation a été introduite par [Agrawal et al. 1993]. Officiellement, le problème de l'extraction par règles d'association est défini comme suit : Soit $I = \{i_1, i_2, \dots, i_n\}$ un ensemble de n littéraux distincts appelés items. Soit $D = \{t_1, t_2, \dots, t_m\}$ un ensemble de transactions dans la base de données. Chaque transaction T est unique et contient un certain nombre d'éléments de I.

Une règle d'association est une implication conditionnelle parmi des ensembles d'éléments (itemsets), $X => Y$ où X et Y sont des éléments (items). Afin d'identifier les règles intéressantes dans l'ensemble de données, il existe deux mesures clés pour mesurer les résultats de l'extraction des règles d'association, le support et la confiance de la règle.

Le support $supp(X)$ est défini comme la proportion de transactions dans l'ensemble de données qui contiennent l'itemset X et reflète sa signification statistique. En termes simples, le nombre de transactions qui contiennent X dans la transaction est divisé par le nombre total de transactions.

Le degré de confiance des règles identifiées est mesuré comme le pourcentage de transactions contenant Y qui contiennent également X, divisé par le nombre de transactions qui contiennent X dans l'ensemble de données. Cela permet d'identifier la fréquence à laquelle la combinaison identifiée se produit ensemble. La confiance est la mesure permettant de contrôler la force individuelle des règles d'association identifiées.

Pour identifier les règles d'association dans un ensemble de données à l'aide d'un algorithme de règles d'association, il y a généralement deux étapes.

1. La première étape consiste à identifier toutes les combinaisons d'itemsets qui répondent aux seuils de support minimum (minsupp) définis par l'utilisateur. Ces itemsets sont dits importants ou fréquents et ceux qui n'atteignent pas le niveau de support sont dits petits ou peu fréquents.

2. La deuxième étape consiste à mesurer la confiance de chaque règle et à la comparer au niveau de confiance minimum choisi (minconf).

Une fois que les règles qui respectent le seuil minimum de support sont identifiées, la deuxième étape est assez simple [Agrawal et al. 1993]. Les algorithmes d'apprentissage des règles d'association se concentrent principalement sur le premier sous-problème ci-dessus et tentent de réduire la tâche coûteuse en termes de calcul qui consiste à identifier toutes les règles qui dépassent le niveau de support défini par l'utilisateur.

À mesure que le nombre d'items augmente, le nombre d'itemsets à évaluer augmente de manière exponentielle. Par exemple, si $|I| = m$, le nombre d'itemsets distincts possibles est 2^m, qui forme un réseau de sous-ensembles au-dessus de I. En général, seul un très petit nombre des itemsets de ce sous-ensemble exponentiellement grand satisfera aux niveaux de support minimums fixés. La figure III-6 fournit une illustration des itemsets qui doivent être évalués avec 4 items.

Le principal problème de l'extraction de règles d'association est d'identifier les itemsets qui répondent au niveau de support minimum défini par l'utilisateur. Lorsque ces algorithmes sont utilisés dans la pratique avec un grand nombre d'items, l'évaluation de chaque itemset n'est pas possible car la taille de l'espace de recherche est trop importante.

Pour réduire la taille de l'espace de recherche, les algorithmes s'appuient sur la propriété de fermeture vers le bas [Agrawal et al. 1994] qui empêche l'algorithme de compter des itemsets qui ne seront pas fréquents à la fin. L'utilisation de cette propriété réduit considérablement le nombre d'itemsets à évaluer.

Il existe quatre principaux types d'algorithmes de règles d'association, chacun d'entre eux utilisant une stratégie différente pour identifier les itemsets qui répondent au niveau de support minimum défini. Les différences entre les modèles sont les suivantes : premièrement, l'algorithme emploie-t-il la recherche en largeur d'abord (breath-first search) ou la recherche en profondeur d'abord (depth-first search) et, deuxièmement, l'algorithme utilise-t-il la génération de candidats ou l'intersection d'ensembles pour déterminer les valeurs de support des candidats? Dans les intersections d'ensembles, les algorithmes utilisent une tidlist. Une TID est un identificateur de transaction unique pour toutes les transactions dans les bases de données. Pour chaque item de I, la tidlist correspondante est une liste de tous les identificateurs de transaction pour les transactions qui contiennent l'item. L'utilisation de Tidlists est appliquée dans les algorithmes Partition et EClat.

III.2.2.1 Algorithme AIS

Dans [Agrawal et al. 1993], les auteurs ont introduit pour la première fois l'idée de fouiller de grands ensembles de données pour en extraire des règles d'association. Les auteurs ont présenté l'algorithme AIS, qui génère de nouveaux itemsets en étendant les grands itemsets trouvés dans la précédente passe de base de données avec d'autres items dans les transactions, une étape connue sous le nom de génération de candidats. Il en résulte le comptage d'un grand nombre d'itemsets qui, en fin de compte, ne respecteront pas les niveaux de support minimums fixés. Les auteurs de [Houtsma et al. 1995] ont ensuite présenté un algorithme appelé SETM, qui a introduit l'idée d'essayer de résoudre le problème des règles d'association en utilisant une base de données relationnelle.

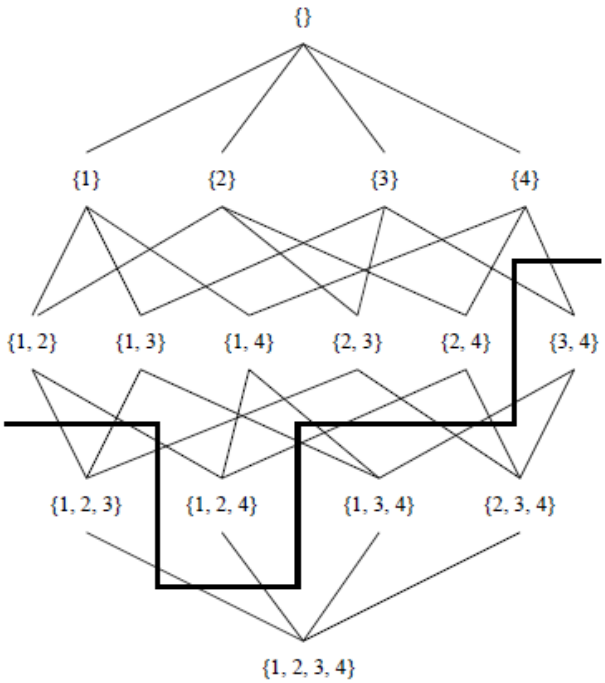


Figure III-6: Treillis pour I = {1,2,3,4} [Hipp et al. 2000]

III.2.2.2 L'algorithme Apriori

Les auteurs de [Agrawal et al. 1994], ont présenté les algorithmes Apriori et AprioriTID. L'algorithme Apriori utilise une recherche en largeur d'abord et s'appuie sur les algorithmes précédents par l'application de la propriété de fermeture vers le bas pour réduire le nombre d'itemsets à compter et donc fonctionner plus efficacement.

Par exemple, s'il s'avère que l'itemset {1,2,3} est petit, alors aucun des itemsets qui sont des extensions de {1,2,3} tels que {1,2,3,4} ou {1,2,3,5,7} ne doit être testé pour un support minimum. En pratique, l'algorithme Apriori élague des ensembles particuliers lors des passages dans la base de données et ne compte aucun itemset lors du passage suivant lorsqu'un sous-ensemble de l'itemset n'a pas atteint le niveau de support requis lors d'un passage précédent. L'une des critiques de l'algorithme Apriori est que l'algorithme nécessite plusieurs passages dans la base de données, ce qui peut être coûteux en termes de calcul.

L'AprioriTid [Agrawal et al. 1994] vise à remédier à la nature coûteuse d'Apriori en termes de calcul. L'AprioriTid améliore les performances d'Apriori en évitant de faire à chaque étape un parcours de toute la base de transactions. Donc, il permet de diminuer la taille du contexte afin de le stocker en mémoire. L'algorithme cherche à garder le contexte en mémoire afin de limiter les accès répétitifs. Cependant, pour les premiers passages, l'encodage des transactions peut être plus important que la base de données proprement dite. Pour surmonter ce problème, les auteurs proposent un hybride d'Apriori et d'AprioriTid appelé Apriori Hybrid, qui utilise l'Apriori pour les passages initiaux et l'AprioriTid pour les passages ultérieurs.

Les auteurs ont comparé les performances des deux nouveaux algorithmes avec celles des algorithmes précédents AIS et SETM. Les résultats ont montré que l'écart de performance, en faveur des deux nouveaux algorithmes, augmentait à mesure que la taille du problème augmentait, allant d'un facteur de trois pour les petits problèmes à plus d'un ordre de grandeur pour les grands problèmes.

L'algorithme CBA pour effectuer la classification en utilisant des règles d'association étend l'algorithme Apriori pour faire des prédictions de classe.

III.2.2.3 L'Algorithme Partition

Les auteurs de [Savasere et al. 1995] présentent une méthode alternative d'extraction de règles d'association connue sous le nom d'algorithme Partition. L'objectif de l'algorithme Partition est de réduire le nombre de passages nécessaires dans la base de données pour identifier les itemsets fréquents. La réduction du nombre de passages que l'algorithme doit effectuer sur la base de données réduit le temps d'exécution et diminue l'impact sur le système matériel sous-jacent.

L'algorithme Partition ne nécessite que deux passages sur la base de données. Lors du premier passage, l'algorithme divise la base de données en un certain nombre de petites partitions qui ne se chevauchent pas, puis identifie tous les itemsets fréquents dans chacun des petits ensembles. Le modèle garantit que les tailles des partitions sont choisies de manière à ce qu'il n'y ait pas de difficultés par rapport à la mémoire principale. Lors de la deuxième passe, ces itemsets fréquents sont réunis, leur nombre réel et leur support sont calculés et ceux qui atteignent le niveau de support cible sont identifiés. La deuxième étape permet de s'assurer que les itemsets, qui se révèlent être fréquents dans chaque partition, c'est-à-dire supportés localement, sont également supportés globalement sur l'ensemble de la base de données.

Comme l'algorithme Apriori, l'algorithme Partition utilise la propriété de fermeture vers le bas et élague les itemsets qui s'avèrent peu importants pour ne pas être pris en compte pour le support de comptage.

Pour tester le modèle par rapport aux algorithmes précédents, les auteurs ont utilisé les mêmes données synthétiques que dans [Agrawal et al. 1994]. Les tests ont montré que le modèle Partition a surpassé le modèle Apriori par un facteur allant jusqu'à sept tout en réduisant les niveaux d'utilisation du processeur et des opérations E/S.

III.2.2.4 Algorithme FP-Growth

Les auteurs de [Han et al. 2000] ont développé une nouvelle approche, Frequent Pattern (FP) Growth, pour identifier les règles d'association, s'éloignant d'une approche de type Apriori. Les algorithmes Apriori utilisent une approche de génération et de test qui consiste à générer des itemsets puis à tester s'ils sont fréquents. L'identification des itemsets fréquents est l'élément le plus coûteux des algorithmes de type Apriori. Les auteurs notent que l'application de la propriété de fermeture vers le bas [Agrawal et al. 1994] permet d'obtenir un bon gain de performance par rapport aux algorithmes précédents, mais reste très coûteuse en termes de performance dans les situations où il existe un grand nombre d'itemsets fréquents ou lorsque les seuils de support minimums sont faibles.

Le modèle FP-Growth propose une autre approche pour identifier les itemsets fréquents qui ne repose pas sur la génération de candidats. Le modèle FP-Growth fonctionne en deux étapes :

1. Le modèle convertit les transactions dans la base de données en une structure de données plus compacte, un arbre des éléments fréquents (Frequent Pattern Tree, "FP-Tree") qui est construit en utilisant deux passages de la base de données.

2. Dans la seconde étape, le modèle utilise alors le FP-Tree construit plutôt que la base de données pour trouver des itemsets fréquents.

Le FP-Tree est construit en deux passes sur la base de données ; dans la première passe, le support de chaque item dans la base de données est calculé et les items peu fréquents sont élagués. Les items fréquents sont triés dans un ordre fixe pour assurer l'efficacité. Ensuite, lors de la seconde passe, le FP-Tree est construit et toutes les transactions sont mises en correspondance avec un chemin sur l'arbre et le comptage est

terminé. Comme certaines transactions peuvent avoir des items en commun, leurs chemins peuvent se chevaucher, ce qui est pris en compte lors de la construction de FP-Tree et réduit donc la taille de la structure des données. La figure III-7 [Tan 2006] ci-dessous décrit le processus de création d'un FP-Tree pour 10 transactions (TIDs). Lorsque les transactions suivantes suivent des chemins similaires, les nœuds de l'arbre augmenteront le nombre par 1.

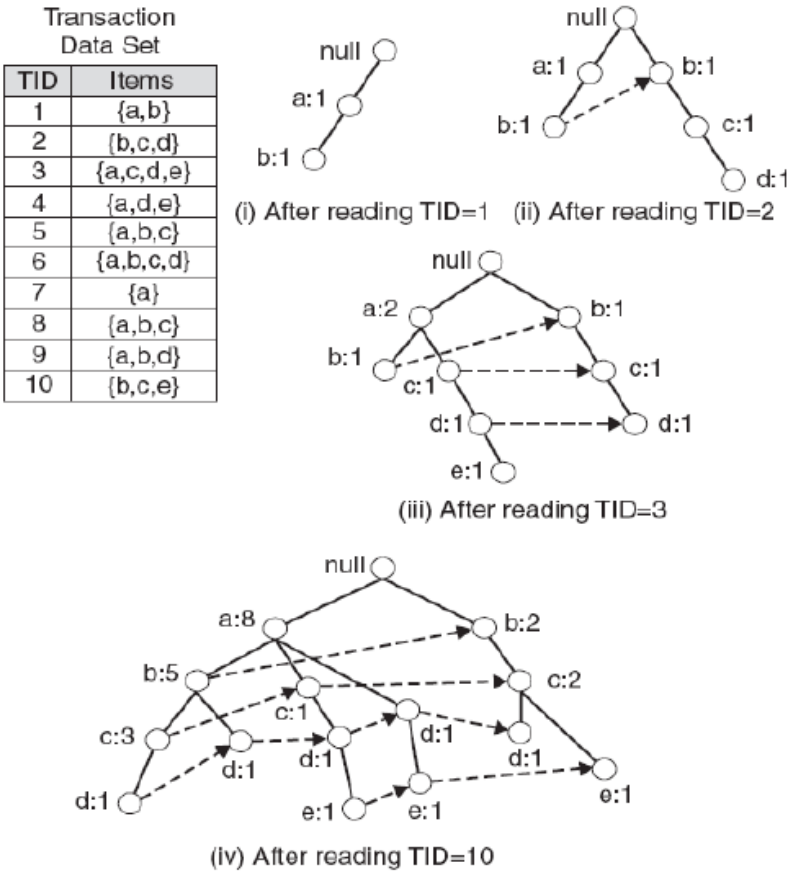


Figure III-7: Arbre FP pour 10 transactions [Tan 2006]

Une fois que l'arbre est créé pour identifier les modèles fréquents, la technique de recherche employée dans l'exploration est une méthode basée sur le partitionnement, la division et la conquête plutôt qu'une génération ascendante de type Apriori de combinaisons d'itemsets fréquents.

Afin d'identifier les modèles fréquents, les auteurs créent des chemins de préfixes sous forme de sous-arbres pour chaque item. Chaque sous-arbre est ensuite traité récursivement pour extraire les itemsets fréquents. Sur la base du chemin du préfixe, le modèle crée des FP-Trees conditionnels pour chaque itemet.

Les auteurs testent le modèle de FP-Growth par rapport aux modèles de génération des candidats et constatent que le modèle est environ un ordre de grandeur plus rapide qu'Apriori.

L'algorithme CMAR pour la classification à l'aide de règles d'association est une extension de l'algorithme FP-Growth pour l'apprentissage des règles d'association.

III.2.2.5 L'Algorithme ECLat (Equivalent Class Transformation)

Les auteurs de [Zaki et al. 1997] présentent quatre nouveaux algorithmes qui ne nécessitent qu'un seul passage sur la base de données. Les algorithmes présentés par les

auteurs diffèrent des algorithmes de type Apriori en ce sens qu'ils traversent l'arbre des préfixes en profondeur d'abord, par rapport à la recherche en largeur d'abord dans l'algorithme Apriori. L'algorithme le plus important présenté, EClat, s'appuie sur des listes d'attente comme décrit ci-dessus. Chaque transaction a un identifiant ou un tid, une tid-list est une liste de toutes les transactions qui contiennent un item particulier. Le modèle ECLat détermine le support de tout k-itemset en croisant les tid-lists de deux de ses (K-1) sous-ensembles.

Les auteurs proposent qu'un format vertical pour le stockage des données transactionnelles est plus applicable à l'extraction des règles d'association qu'un format horizontal. Selon cette méthode, le modèle n'a besoin de faire qu'un seul passage de la base de données. Apriori et FP-Growth utilisent tous deux un format de données horizontal qui commence par l'identifiant de la transaction et les itemsets au sein de la transaction. Le format vertical commence par l'itemset et énumère toutes les transactions qui contiennent cet itemset. Les auteurs déclarent qu'un format vertical semble plus approprié pour l'extraction d'associations puisque le support d'un k-itemset candidat peut être calculé par de simples intersections de listes de contrôle. La figure III-8 montre un exemple d'intersection d'un Tid-list.

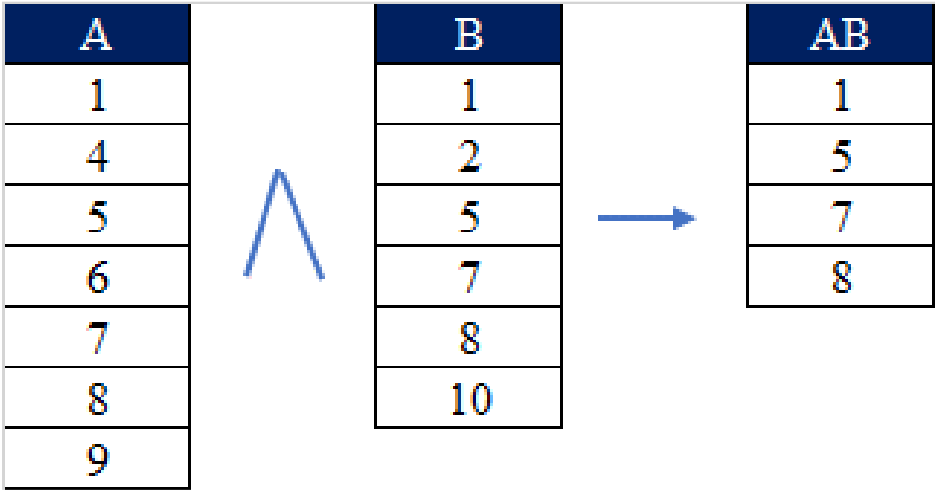


Figure III-8: Exemple d'intersection de Tid-list

Les auteurs comparent les nouveaux algorithmes présentés avec l'algorithme Apriori et l'algorithme Partition (avec 10 partitions). Les auteurs déclarent qu'EClat surpasse Apriori par un facteur de 10 et Partition par un facteur de 5. Les auteurs indiquent également que les nouveaux modèles s'adaptent à l'augmentation de la taille des transactions. L'avantage de cet algorithme est que la recherche en profondeur d'abord peut donner des résultats beaucoup plus rapides, cependant, les Tid-lists intermédiaires peuvent devenir trop grandes pour la mémoire.

III.2.2.6 Conclusion sur les algorithmes d'apprentissage des règles d'association

Zheng et al. [Zheng et al. 2001] ont réalisé la première évaluation et comparaison des algorithmes d'apprentissage des règles d'association sur des ensembles de données du monde réel. Dans cette expérience, les auteurs ont évalué cinq des algorithmes de règles d'association les plus récents, dont Apriori et FP-Growth. Les auteurs ont évalué les performances sur trois ensembles de données du monde réel et un ensemble de données artificielles en utilisant une série de valeurs de support minimales pour tester les performances et l'évolutivité. Pour l'ensemble de données artificielles, chaque algorithme a surpassé Apriori par une marge significative pour des valeurs de support minimales inférieures à 0,10%. Le FP-Growth a été plus rapide d'un ordre de grandeur qu'Apriori

lorsque le support minimum était fixé à 0,02%. Cette preuve est cohérente avec les résultats d'autres expériences précédentes [Han et al. 2000] [Zaki 2000]. L'amélioration des performances de FP-Growth par rapport à Apriori augmente à mesure que le support minimum diminue, ce qui indique que FP-Growth se développe plus rapidement qu'Apriori. Pour tous les ensembles de données du monde réel, FP-Growth est plus rapide qu'Apriori, mais les différences ne sont pas aussi importantes que pour l'ensemble de données artificielles. Le raisonnement proposé par [Zheng et al. 2001] est que l'ensemble de données artificielles présente des caractéristiques différentes de celles des ensembles de données du monde réel.

Dans [Hipp et al. 2000], les auteurs comparent un certain nombre d'algorithmes de règles d'association en termes d'efficacité en effectuant plusieurs expériences d'exécution sur des données synthétiques. Les auteurs comparent Apriori, DIC une variation d'Apriori [Brin et al. 1997], Partition et Eclat. Les auteurs déclarent que les résultats des expériences indiquent que le comportement d'exécution des différents algorithmes est plus similaire que prévu. Ce n'est que dans certains cas plus extrêmes que les auteurs ont constaté une variation des performances. Dans la figure III-9, une des expériences sur un ensemble de données plus complexe montre qu'Eclat et Partition sont plus performants qu'Apriori, en particulier avec des niveaux de support minimums faibles.

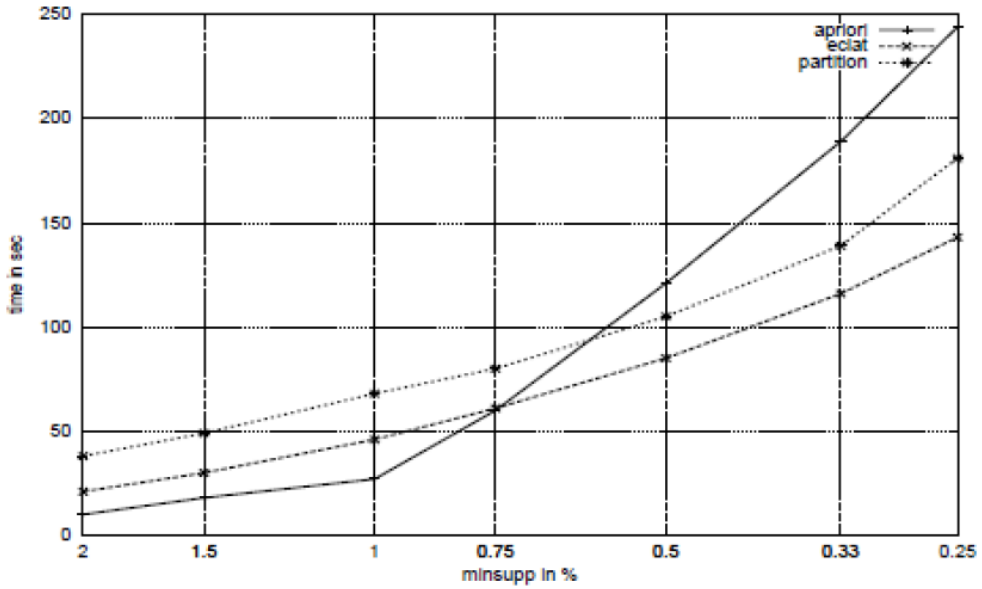


Figure III-9: Comparaison des algorithmes de règles d'association à travers différents niveaux de support [Hipp et al. 2000]

Heaton [Heaton 2016], a comparé les performances d'Apriori, d'ECLat et de FP-Growth sur différents ensembles de données créés artificiellement. Deux caractéristiques des ensembles de données ont été évaluées, la taille maximale des transactions et la densité d'item fréquent, et les algorithmes ont été testés dans diverses conditions. Les résultats montrent que Eclat et FP-Growth gèrent tous deux les augmentations de la taille maximale des transactions et de la densité des itemsets fréquents bien mieux que l'algorithme Apriori, tandis que FP-Growth a légèrement surpassé Eclat. La figure III-10 ci-dessous présente les résultats des tests pour différentes densités d'itemsets fréquents. Elle montre que les trois algorithmes ont des performances similaires jusqu'à environ 70%, point auquel les performances de l'algorithme Apriori se détériorent considérablement.

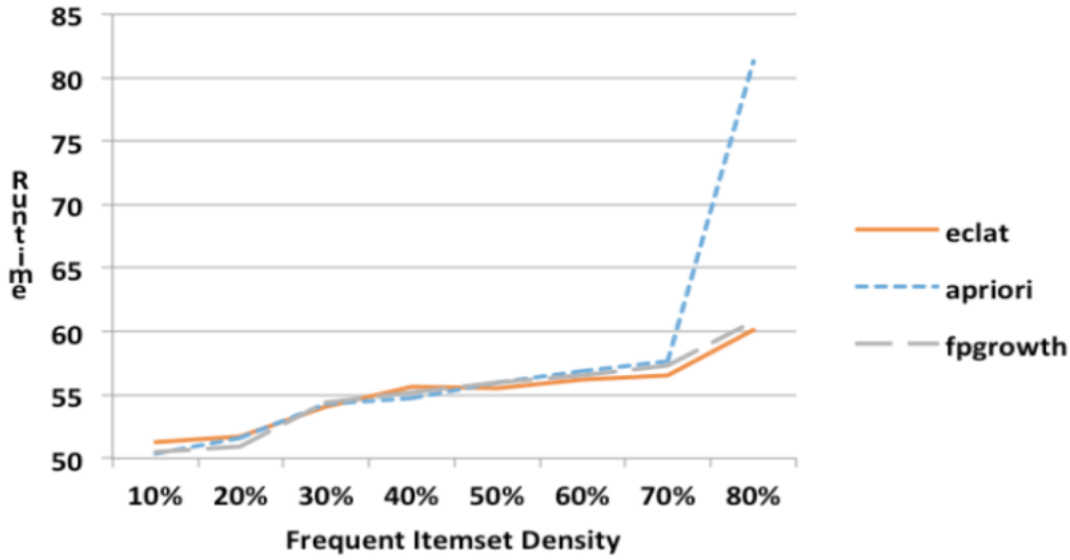


Figure III-10: Comparaison des algorithmes de règles d'association pour des densités d'itemset variables [Heaton, 2016]

III.3 Première approche proposée pour l'indexation et la recherche d'objet 3D

III.3.1 Approche proposée

Dans cette section, nous allons présenter notre première approche K-RS (K Representative Slices) [Taybi et al. 2016] pour l'indexation et la recherche d'objets 3D en se basant sur un ensemble d'images de coupe 2D, qui peut être classé comme une approche 2,5D/3D, pourrait être divisée en cinq étapes :

- Normalisation des objets 3D ;
- Création de l'ensemble initial des images de coupe 2D ;
- Calcul de la signature numérique pour chaque image 2D ;
- Extraction des images de coupe 2D représentatives ;
- Calcul de la similarité.

❖ **Normalisation des objets 3D**

Les objets 3D sont obtenus par divers outils d'acquisition ou de numérisation, par conséquent, ils ont souvent des échelles, des orientations et des positions arbitraires dans l'espace 3D (Figure III-11-a). En outre, la plupart des approches d'indexation et de recherche d'objets 3D ne satisfont pas l'invariance géométrique. Il est donc important de normaliser les objets 3D dans un repère canonique pour assurer une représentation unique.

En fait, la normalisation est une étape de prétraitement courante non seulement pour les systèmes d'indexation et de recherche d'objets 3D, mais aussi pour la reconnaissance, la visualisation et la correspondance d'objets 3D. Elle tente de normaliser les objets 3D en une pose canonique, où la représentation de l'objet est indépendante de sa position, de son échelle et de son orientation. Ainsi, tous les objets 3D peuvent être comparés sous la même pose.

Dans notre approche, la normalisation de la translation est assurée par le calcul du centre de masse de l'objet 3D et le translate, par la suite, pour qu'il coïncide avec l'origine. Pour atteindre la normalisation de l'échelle, la taille de l'objet 3D est modifiée pour que la

distance moyenne de sa surface par rapport à son centroïde soit égale à 1. La normalisation de la rotation est obtenue en utilisant l'analyse en composantes principales (ACP).

L'alignement est généralement le passage le plus difficile du processus de normalisation, et il est encore à l'étude. Dans notre travail, nous avons mis en œuvre l'analyse en composantes principales (ACP) et l'analyse en composantes principales continue (ACPC), et nous avons déduit que la version dite "Continue" de l'algorithme est plus stable que la version "Classique" pour un temps de calcul un peu plus long, confirmant les résultats de [Vranic et al. 2001a] [Vranic et al. 2001b]. Cependant, nous avons remarqué que l'application de l'ACP pour certains objets 3D donne des résultats erronés, y compris les inversions entre leurs trois axes principaux. Les auteurs [Zaharia et al. 2002] ont également montré que l'application de cette méthode dans le cas discret pouvait présenter des faiblesses comme l'inversion des axes (Figure III-12). En effet, si l'objet 3D a des tailles similaires en deux ou trois dimensions, l'ordre des axes peut être instable.

❖ **Création de la série initiale des images de coupe 2D**

Parmi les différents outils de représentation des objets 3D, les maillages triangulaires constituent un moyen efficace. Les données de géométrie et de connectivité sont utilisées à la fois pour représenter un maillage triangulaire 3D. En fait, notre approche consiste à créer un ensemble initial d'images de coupe 2D obtenues par l'intersection d'un ensemble de plans avec le maillage triangulaire 3D. Afin d'obtenir l'image de coupe 2D dans une direction déterminée, nous déplaçons le rayon dans le plan correspondant et nous calculons à chaque fois la distance D entre l'origine O du rayon et l'intersection avec le maillage triangulaire 3D.

Considérons I le point obtenu par l'intersection d'un maillage triangulaire ABC et le rayon orienté par le vecteur $\vec{v}$. La relation suivante détermine le point d'intersection :

$$OI = D.\vec{v} \tag{III-1}$$

Le point d'intersection I, dans la surface délimitée par la facette ABC, a vérifié cette équation :

$$\overrightarrow{OA}.\vec{n} = \overrightarrow{OI}.\vec{n} \tag{III-2}$$

Avec $\vec{n}$ est le vecteur normal du triangle ABC, il est défini par l'équation suivante :

$$\vec{n} = \frac{\overrightarrow{AB} \wedge \overrightarrow{AC}}{\| \overrightarrow{AB} \wedge \overrightarrow{AC} \|} \tag{III-3}$$

Pour s'assurer que le point I n'est pas vide, il suffit que ce point vérifie les conditions dans l'équation suivante :

$$\begin{cases} (\overrightarrow{IA} \wedge \overrightarrow{IB}).\vec{n} > 0 \\ (\overrightarrow{IB} \wedge \overrightarrow{IC}).\vec{n} > 0 \\ (\overrightarrow{IC} \wedge \overrightarrow{IA}).\vec{n} > 0 \end{cases} \tag{III-4}$$

Afin de mieux décrire la forme des objets 3D, et aussi pour contourner la faiblesse de l'ACP, notamment le problème de l'inversion des axes principaux, nous extrayons un ensemble d'images de coupe 2D correspondant aux trois axes principaux pour chaque objet 3D.

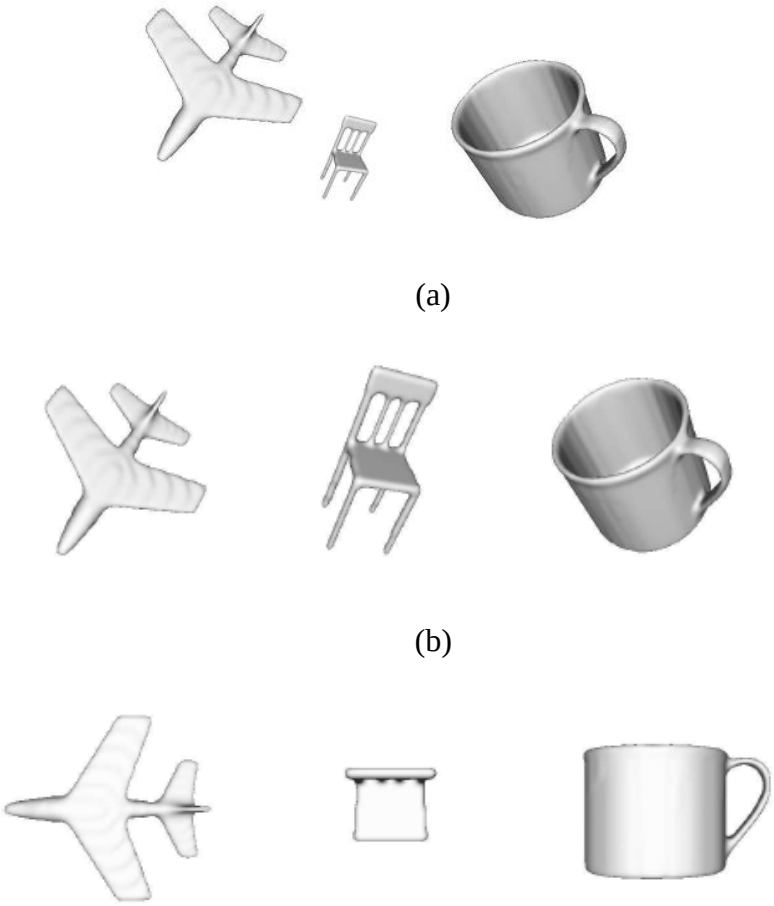


Figure III-11: Exemple de normalisation d'objets 3D, (a) objets 3D en position arbitraire, (b) normalisation de la position et de l'échelle, (c) normalisation de l'orientation

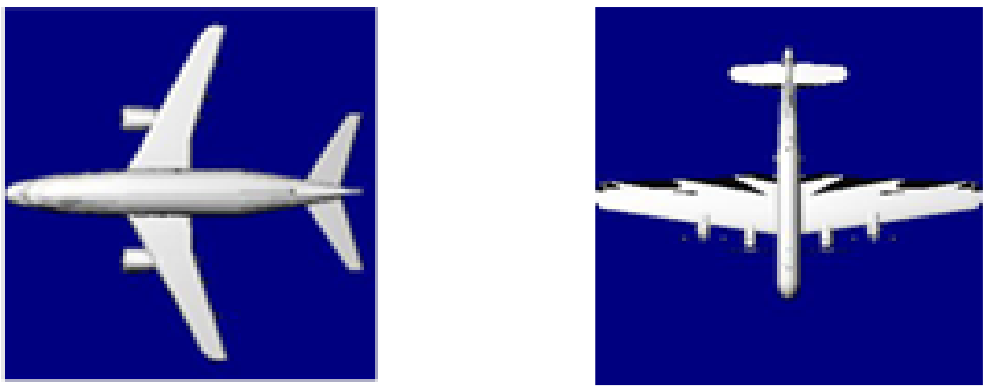


Figure III-12: Exemple de permutation des deux axes sur des objets 3D de la même classe " avion ".

En fait, pour chaque axe principal, nous prenons l'intersection de l'objet 3D avec 100 plans équidistants et orthogonaux à l'axe. A la fin de cette opération, nous obtenons l'ensemble initial des images de coupe 2D (Initial Slices IS) : $IS = \{IS_{ox}, IS_{oy}, IS_{oz}\}$ telles

que: $IS_{ox} = \{IS_1^{ox}, \dots, IS_{100}^{ox}\}$, $IS_{oy} = \{IS_1^{oy}, \dots, IS_{100}^{oy}\}$, et $IS_{oz} = \{IS_1^{oz}, \dots, IS_{100}^{oz}\}$. Avec IS_{ox}, IS_{oy} et IS_{oz} représentent respectivement l'ensemble initial des images de coupe 2D correspondant aux axes X, Y et Z de l'objet.

❖ **Calcul de la signature numérique pour chaque image de coupe 2D**

Les moments de Hu ont été largement utilisés comme descripteurs de caractéristiques de base dans l'analyse d'images, la reconnaissance d'objets, la classification d'images et la correspondance de modèles [Dragisa et al. 2014] [Zhang et al. 2014] [Huang et al. 2010]. Les moments de Hu présentent deux avantages : (1) leur calcul est algorithmiquement simple et défini de manière unique pour toute fonction d'image; (2) les moments de Hu sont invariants en ce qui concerne la translation, la mise à l'échelle, ainsi que la rotation [Hu 1962] [Dragisa et al. 2014] [Zhang et al. 2014] [Huang et al. 2010]. Il est raisonnable de conclure que les moments de Hu pourraient être convenables pour caractériser les images de coupe 2D des objets 3D.

Les moments de Hu sont décrits en utilisant les moments statistiques et les moments centraux. Les moments statistiques sous forme discrète sont définis comme :

$$m_{pq} = \sum_{i=0}^{u-1} \sum_{j=0}^{v-1} x_i^p y_j^q f(x,y) \quad \text{(III-5)}$$

Où f(x,y) est la structure analysée, f(x,y) est l'intensité du pixel et (x,y) sont les coordonnées du pixel. Comme les images de coupe 2D des objets 3D sont binaires, f(x,y) est considéré comme 1. Les moments centraux sont définis par :

$$\mu_{pq} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (x_i - \bar{x})^p \left(y_j - \bar{y}\right)^q f(x,y) \quad \text{(III-6)}$$

Où $\bar{x} = \frac{m_{10}}{m_{00}}$ et $\bar{y} = \frac{m_{01}}{m_{00}}$

Les moments normalisés sont définis par :

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{(III-7)}$$

Où $\gamma = \frac{p+q}{2} + 1$.

Les moments invariants de Hu sont définis par :

$$\phi_1 = \eta_{20} + \eta_{02} \quad \text{(III-8)}$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 - 4\eta_{11}^2 \quad \text{(III-9)}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 - (3\eta_{11}^2 - \eta_{03})^2 \quad \text{(III-10)}$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 - (\eta_{21}^2 - \eta_{03})^2 \quad \text{(III-11)}$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad \text{(III-12)}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad \text{(III-13)}$$

$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{03} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad \text{(III-14)}$$

Ces moments de Hu sont appliqués à chaque image de coupe 2D des objets 3D. Par conséquent, chaque image de coupe est représentée par les sept moments invariants de Hu sous la forme d'un vecteur.

❖ **Extraction des images de coupe 2D représentatives**

L'idée générale de notre approche est de générer un premier ensemble d'image de coupe 2D et de sélectionner celles qui représentent le mieux l'objet 3D. En fait, la tâche de trouver le sous-ensemble d'images de coupe 2D représentatives équivaut à un problème de clustering. De nombreuses études couvrent ce domaine, notamment les travaux de [El far et al. 2011] qui proposent une étude comparative de quatre méthodes d'exploration de données appliquées pour résoudre le problème de la sélection de vues représentatives d'un objet 3D.

Comme l'utilisation de toutes les images de coupe 2D extraites peut augmenter le temps de recherche, après avoir extrait l'ensemble initial des images de coupe 2D, nous exécutons l'algorithme de clustering k-means, pour obtenir les images de coupe 2D les plus représentatives.

En fait, l'algorithme de clustering K-means [Hartigan et al. 1979] est une méthode de clustering de données qui est l'un des algorithmes les plus courants, utilisant une technique de raffinement itérative. Il tente de diviser N observations en K clusters dans lesquels chaque observation se rattache au cluster dont la moyenne est la plus proche (voir la section III.2.1.1 pour plus de détail sur l'algorithme K-means).

Soit $\{x_1, x_2, \dots, x_N\}$ un ensemble d'observations vectorielles de dimension D, cet algorithme de clustering s'efforce de diviser les données d'observation en K clusters (K<N), de sorte que le critère des moindres carrés (WCSS, pour Within-Cluster Sum of Squares) soit minimisée comme suit :

$$WCSS = arg\min_{S} \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \quad \text{(III-15)}$$

Où μ_i est la moyenne des points dans S_i.

Dans l'algorithme standard de clustering K-means, les centres initiaux de clusters sont choisis au hasard, nous pouvons donc trouver qu'un centre de cluster appartient à deux clusters ou plus. Afin de surmonter ce problème, une nouvelle approche est adoptée pour définir les centres initiaux de clusters. Le nombre d'images de coupe 2D initiales est divisé par le nombre de clusters K, ce qui donne un intervalle. Par la suite, les centres initiaux sont choisis en fonction de cet intervalle, de sorte que les centres des clusters s'éloignent avec une valeur fixe.

Comme mentionné précédemment, nous avons extrait, pour chaque objet 3D, 300 images de coupe 2D correspondant au trois axes principaux (100 images de coupe 2D par axe). Ensuite, nous appliquons l'algorithme modifié de K-means pour diviser l'ensemble initial des images de coupe 2D en K clusters. A la fin de cette opération, nous choisissons

les images de coupe 2D représentatives, en prenant celle la plus proche du centroïde pour chaque cluster afin d'en former un sous-ensemble représentatives, noté (Representative Slices RS): RS $= \{RS_{ox}, RS_{oy}, RS_{oz}\}$ comme: $RS_{ox} = \{RS_1^{ox}, \dots, RS_k^{ox}\}$; $RS_{oy} = \{RS_1^{oy}, \dots, RS_k^{oy}\}$ et $RS_{oz} = \{RS_1^{oz}, \dots, RS_k^{oz}\}$.

La figure III-14 présente un exemple visuel d'un objet 3D et de ses images de coupe 2D représentatives en utilisant notre approche.

❖ **Calcul de la similarité**

Sachant que nous avons choisi une approche basée sur des images de coupe 2D pour décrire les objets 3D, il est nécessaire de mettre en place un processus pour comparer l'ensemble d'images de coupe 2D représentatives de chaque objet 3D dans la base de données avec celles de l'objet 3D requête. En fait, le problème de la correspondance est converti en la manière de mesurer la similarité entre les sous-ensembles d'images de coupe 2D représentatives de différents objets 3D.

Afin de répondre à ces exigences, nous avons utilisé la distance de Hausdorff [Atallah et al. 1983]. Cette distance a montré sa force dans les travaux existants [Gao et al. 2014] [Russ et al. 2005]. En fait, cette distance peut constituer un critère de dissimilitude entre deux ensembles. Considérons le point du vecteur (v1) le plus éloigné du vecteur (v2), d'une part, et le point du vecteur (v2) le plus éloigné du vecteur (v1), d'autre part. Ensuite, la distance de Hausdorff est le maximum des deux. Qui est formulé comme suit :

$$d_h(v_1, v_2) = \max\{\sup_{p \in v_1} \inf_{q \in v_2} (d_{pq}), \sup_{q \in v_2} \inf_{p \in v_1} (d_{pq})\} \quad \text{(III-16)}$$

Où d_{pq} représente la distance euclidienne entre p et q.

Étant donné qu'un objet 3D de la base de données O est représenté par $O = \{\{S_1^{O_{ox}}, \dots, S_l^{O_{ox}}\}, \{S_1^{O_{oy}}, \dots, S_m^{O_{oy}}\}, \{S_1^{O_{oz}}, \dots, S_n^{O_{oz}}\}\}$ et un objet 3D requête Q représenté par $Q = \{RS_{ox}^Q, RS_{oy}^Q, RS_{oz}^Q\}$. La distance donc entre O et Q devient :

$$d_h(O, Q) = \max\{\sup_{1 \le i \le N} \inf_{1 \le j \le N} (d_{O_i Q_j}), \sup_{1 \le j \le N} \inf_{1 \le i \le N} (d_{O_i Q_j})\} \quad \text{(III-17)}$$

Où N représente le nombre d'image de coupe 2D représentatives d'un objet 3D.

III.3.2 Résultats expérimentaux

Dans cette section, nous présentons la base de données de tests, les résultats expérimentaux et les critères d'évaluation que nous avons utilisés pour valider notre méthode. L'approche proposée a été mise en œuvre en utilisant le langage C++ sur la plate-forme Windows.

Afin de mesurer la performance de la méthode proposée, nous avons utilisé 146 objets 3D collectés dans la base de données Princeton Shape Benchmark (PSB) au format OFF. Le PSB a vu le jour en 2004 et il est maintenant considéré comme une référence de forme standard largement utilisée dans la communauté de recherche d'objets 3D. Il contient un ensemble de 1814 objets 3D classés, collectés à partir de 293 domaines Web différents. Les objets 3D utilisés sont classés manuellement en 12 classes en fonction de leur similitude visuelle.

Au cours de nos expériences, et pour indiquer la puissance et la performance de notre méthode, nous avons comparé les résultats obtenus avec ceux du descripteur de Zernike 3D.

La figure III-13 montre des images qui représentent chaque classe de notre base de données de tests.

Afin de valider notre méthode, nous avons effectué deux tests en comparant les résultats avec ceux obtenus par le descripteur de Zernike 3D. Les tests réalisés sont :

- Sélectionner quelques objets 3D requêtes dans la base de données de test et montrer les six premiers objets 3D récupérés en utilisant notre approche et comparer les résultats avec ceux fournis par le descripteur de Zernike 3D;

- Comparaison des performances de recherche de notre approche par rapport au descripteur de Zernike 3D en utilisant les courbes de rappel/précision.

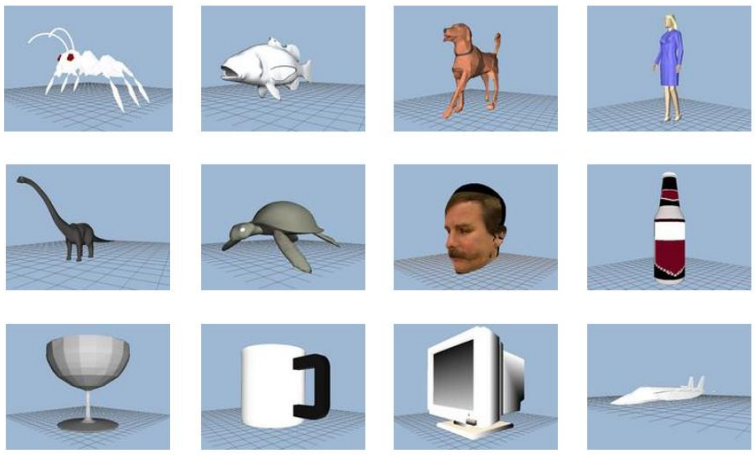
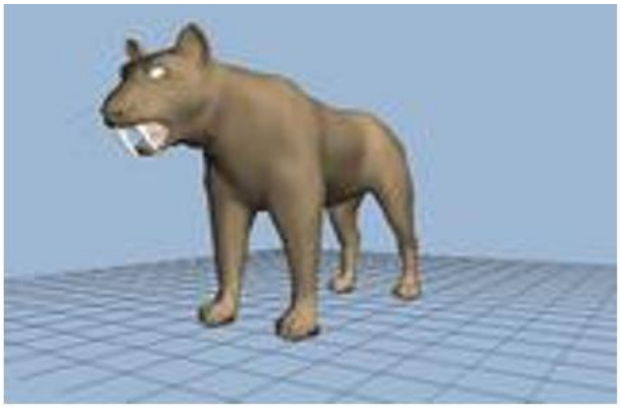
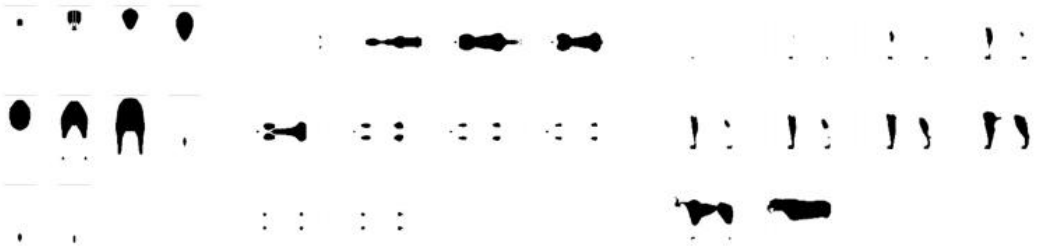


Figure III-13: Exemples d'objets 3D de différentes classes.



(a)



(b) (c) (d)

Figure III-14 : Exemple d'un objet 3D (a) et de ses images de coupe représentatives correspondant à ses trois axes principaux en utilisant notre approche.

La figure III-15 et la figure III-16 montrent respectivement certains objets 3D utilisés comme requête ainsi que les six tops objets 3D récupérés dans la base de données en utilisant l'approche proposée et le descripteur de Zernike 3D. Comme nous pouvons déduire à partir des résultats obtenus par notre méthode, presque tous les objets 3D récupérés appartiennent à la même classe de l'objet 3D requête, ce qui n'est pas le cas pour les résultats donnés par le descripteur de Zernike 3D.

Figure III-15 : Les tops six objets 3D récupérés en utilisant notre approche.

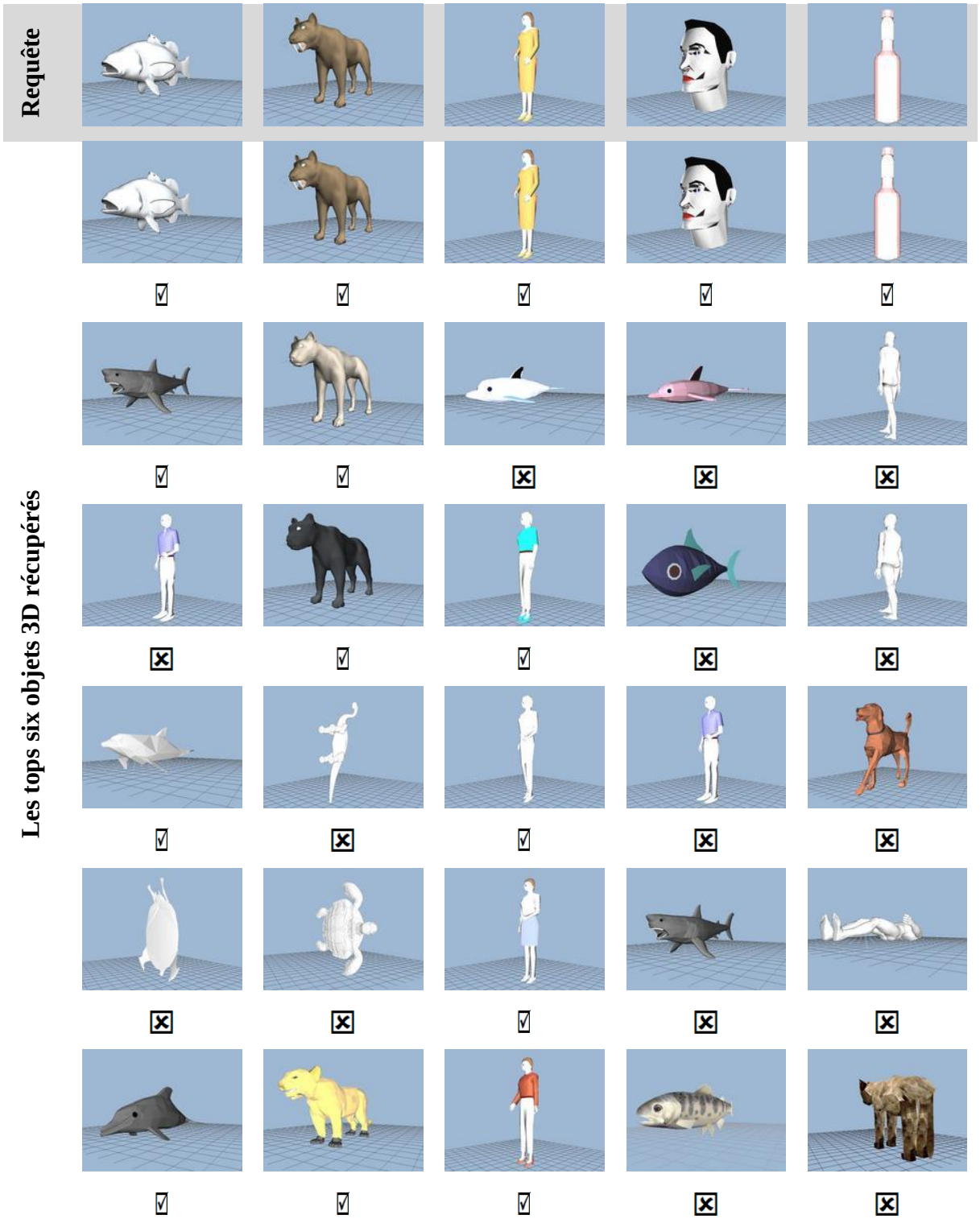


Figure III-16 : Les tops six premiers objets 3D récupérés en utilisant le descripteur de Zernike 3D.

Pour mesurer les performances de notre approche, une étude en termes de courbes "rappel/précision" a été également réalisée. La courbe "rappel/précision" est connue dans le domaine de la reconnaissance et de la recherche par le contenu (voir la section II.6 pour plus de détail sur ce critère d'évaluation).

Pour choisir le nombre optimal d'images de coupe représentatives, nous avons effectué différentes expériences et les résultats trouvés en termes de courbes de précision/rappel pour trois valeurs (10, 20 et 30) de K sont présentés dans la figure III-17.

Nous pouvons observer que les courbes de précision/rappel de l'approche proposée pour différentes valeurs du nombre d'images de coupe représentatives sont presque similaires. Par conséquent, nous choisissons la valeur minimale de K (K = 10) pour diminuer le temps de recherche.

La figure III-18 présente une comparaison entre notre méthode et le descripteur de Zernike 3D en termes de rappel/précision. Comme le montrent les résultats exposés, notre approche a donné de bons résultats par rapport au descripteur de Zernike 3D, qui est l'un des descripteurs les plus utilisés dans la littérature.

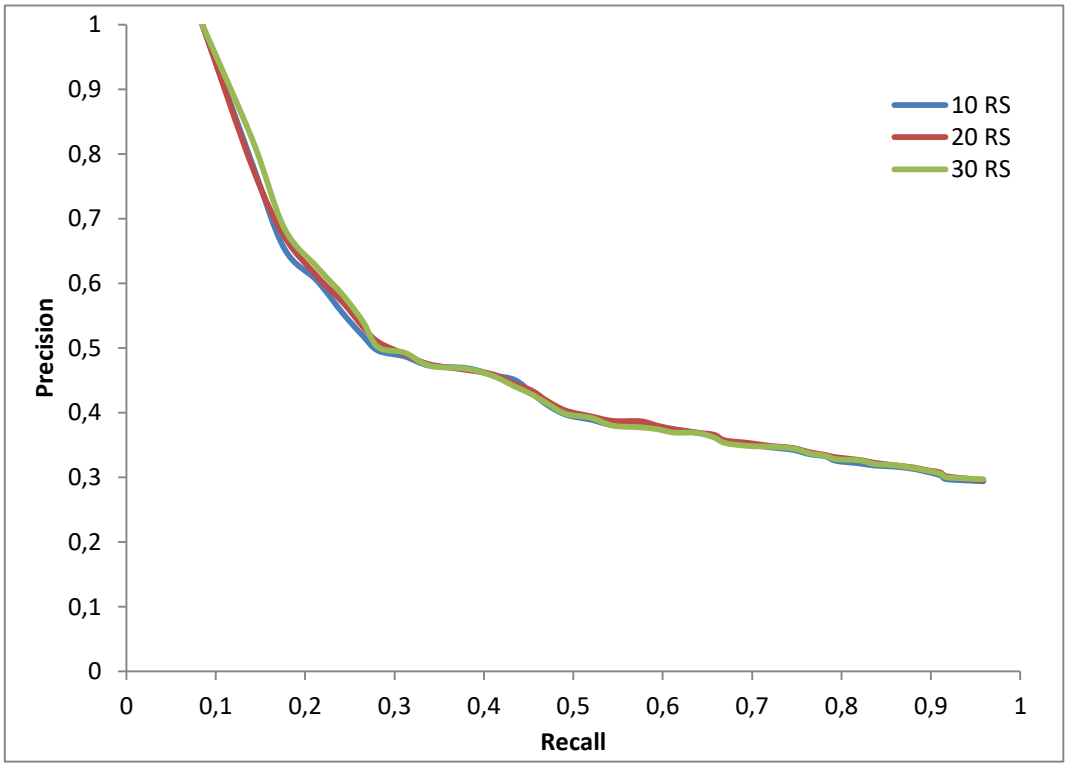


Figure III-17 : Les courbes de rappel de précision de l'approche proposée pour différentes valeurs de K (nombre d'images de coupe représentatives).

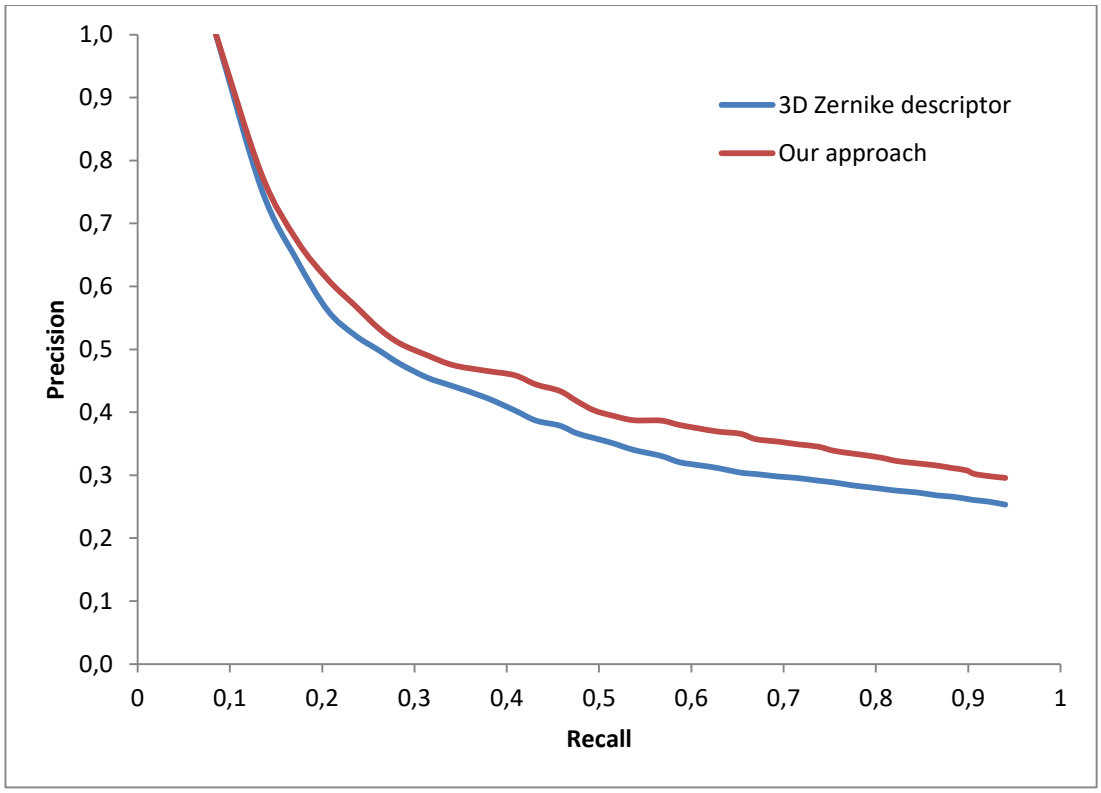


Figure III-18 : les courbes de rappel/précision de l'approche proposée et le descripteur de Zernike 3D.

III.3.3 Conclusions

Dans la présente section, nous avons présenté notre première approche pour l'indexation et la recherche d'objets 3D, basée sur la génération de nombreuses images de coupe 2D correspondant aux trois axes principaux de l'objet 3D, puis nous avons choisi les images de coupe les plus représentatives en utilisant l'algorithme modifié de K-means. Nous avons également mené quelques tests expérimentaux pour valider la méthode proposée. Les résultats obtenus montrent les performances de notre approche, qui surpassent le descripteur de Zernike 3D.

III.4 Deuxième approche proposée pour l'indexation et la recherche d'objet 3D

III.4.1 Approche proposée

Dans cette section, nous allons décrire notre deuxième approche pour l'indexation et la recherche d'objets 3D, appelée Adaptive Slices Clustering (ASC) [Taybi et al. 2019]. Contrairement à l'approche présentée dans la section précédente (K_RS) [Taybi et al. 2016], qui caractérise tous les objets 3D par le même nombre d'images de coupe 2D, la nouvelle méthode proposée améliore notre K_RS descripteur en adaptant le nombre d'images de coupe représentative à la complexité de chaque objet 3D.

En fait, la méthode K_RS donne des résultats satisfaisants si le nombre d'images de coupe 2D représentative est correctement choisi. Cependant, les objets 3D ont des complexités différentes au niveau de leurs structures, ce qui rend impossible de trouver un nombre convenable à tous les objets 3D. Par conséquent, on se retrouve dans une sur-représentation ou sous-représentation des objets 3D, ce qui diminue les performances de la méthode.

Nous avons décrit dans la section précédente toutes les étapes de notre première approche k_RS, à savoir : (1) Normalisation des objets 3D ; (2) Création de l'ensemble initial d'images de coupe 2D ; (3) Calcul de la signature numérique pour chaque image de coupe 2D ; (4) Extraction des images de coupes représentatives; et (5) Calcul de la similarité.

Dans cette section, nous allons présenter, seulement, les points d'amélioration, notamment : la méthode utilisée pour sélectionner les images de coupe représentatives ; et la métrique proposée pour mesurer la similarité entre les ensembles d'images de coupe représentatives des objets 3D.

❖ **Extraction des images de coupe représentatives**

Le nombre d'images de coupe représentatives d'un objet 3D doit être suffisamment grand pour donner une représentation complète de l'objet 3D, mais pas trop grand pour ne pas obtenir une surreprésentation préjudiciable aux performances de l'approche. Pour répondre à ces contraintes, nous avons adopté une démarche pour sélection les images de coupe représentatives en adaptant son nombre en fonction de la complexité de l'objet 3D.

En effet, en fonction de la complexité de l'objet 3D à caractériser, le nombre d'images de coupe peut varier fortement. Par exemple, pour représenter les images de coupe correspondant à l'axe Ox de l'objet 3D dans la figure III-19-a, un petit nombre d'image de coupe suffit pour le caractériser (Figure III-19-b), puisque l'ensemble des images de coupe extraites sont presque identiques. Par contre, pour représenter les images de coupe correspondant aux axes principaux d'un objet 3D complexe comme l'objet de la figure III-20-a, il faut un grand nombre d'images de coupe (figure III-20.b-d). L'idée générale de notre approche est de générer, dans un premier temps, un ensemble d'images de coupe 2D et de sélectionner celles qui représentent au mieux l'objet 3D. En fait, la tâche consiste à trouver le sous-ensemble d'images de coupe représentatives équivaut à un problème de clustering.

Au début de nos travaux, nous nous sommes intéressés à l'algorithme de classification K-means. Cette méthode donne des résultats satisfaisants si le nombre de clusters est correctement choisi. Dans le cas contraire, le processus de clustering produit une sous-partition ou une sur-partition. Dans notre cas également, le nombre de clusters est lié à la complexité de chaque objet 3D. Par conséquent, la détermination automatique du nombre optimal de clusters (K*) est d'une grande importance. À cette fin, nous utilisons l'indice de validité des clusters proposé par Do-Jong et al. [Do-Jong et al. 2001].

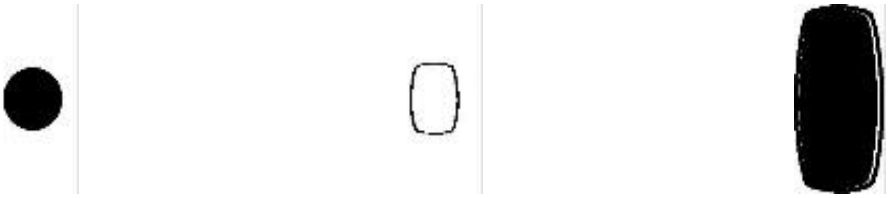
En comparaison avec d'autres études utilisant d'autres indices, cet indice prend en compte les caractéristiques autour du nombre optimal de clusters pendant le processus de partitionnement. En fait, les structures de clusters peuvent avoir l'un des trois états suivants : état de sous-partition (K < K*), état de partition optimale (K = K*) ou état de sur-partition (K > K*). Il est possible de trouver le nombre optimal de clusters en utilisant deux mesures : la distance moyenne intra-cluster (mean intra-cluster distance (MICD)) et la distance minimale inter-clusters (minimum inter-cluster distance (ICMD)).

La MICD du i^{ème} cluster $MICD_i$ est défini par :

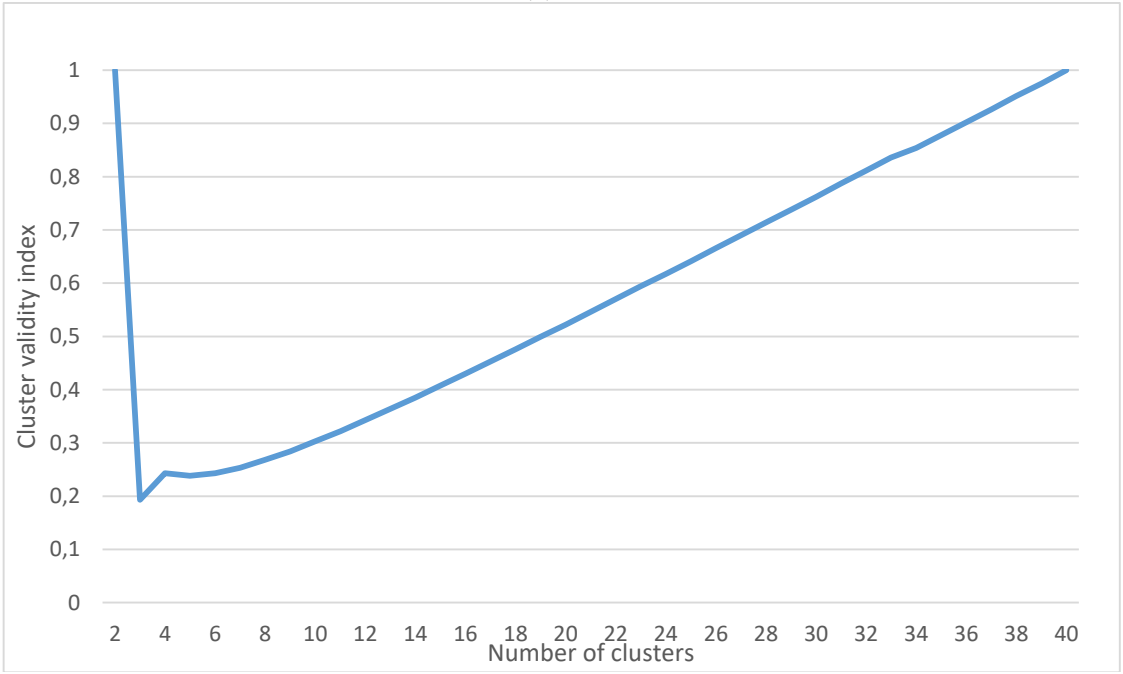
$$MICD_i = \frac{1}{n_i} \sum_{x \in \chi_i} \| V_i - x \| \qquad \text{(III-18)}$$

Où χ_i, V_i et n_i représentent respectivement l'ensemble de données du i^{ème} cluster, le centroïde du i^{ème} cluster et le nombre de données dans le i^{ème} cluster.

(a)



(b)



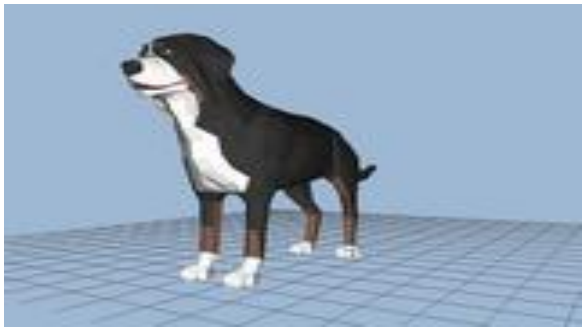
(c)

Figure III-19: Exemple d'un simple objet 3D (a) avec ses images de coupe représentatives (b) et la fonction de mesure de l'indice de validité du cluster correspondant à son premier axe principal (c).

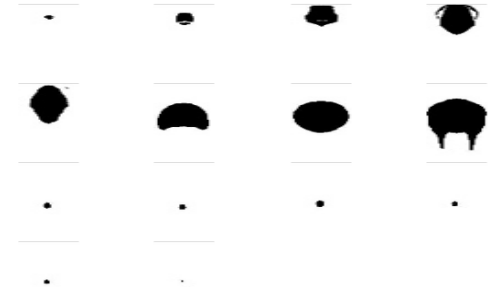
$$ICMD_{min} = \min_{i \neq j} \sum_{x \in \chi_i} \| V_i - V_j \| \qquad \text{(III-19)}$$

Où V_i et V_j représentent respectivement le centroïde du i[ème] et du j[ième] cluster.

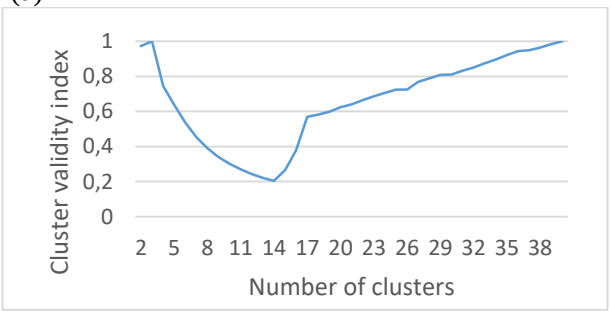
Soit $X = [x_1, x_2, \dots, x_n]^T$ un ensemble de données fini, et soit $V = [v_1, v_2, \dots, v_K]^T$ un centroïde K, chaque v_i caractérisant un des K clusters. La mesure de sous-partition $v_{under}(K, V, X)$ et la mesure de sur-partition $v_{over}(K, V)$, respectivement définies par les équations (III-20) et (III-21), ont des échelles différentes selon la structure et le nombre de données. Une normalisation de ces fonctions est donc nécessaire.



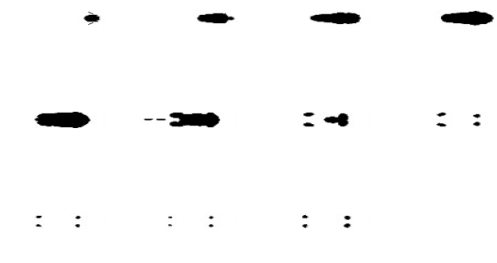
(a)



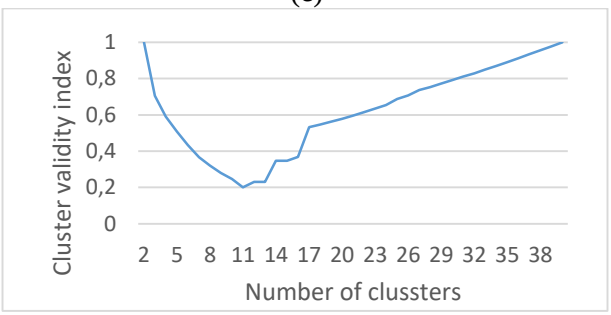
(b)



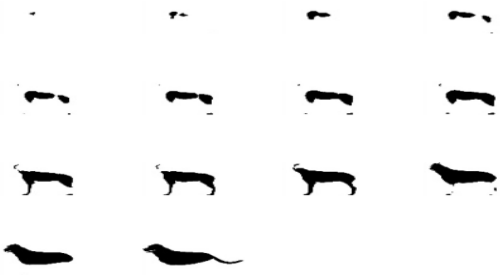
(e)



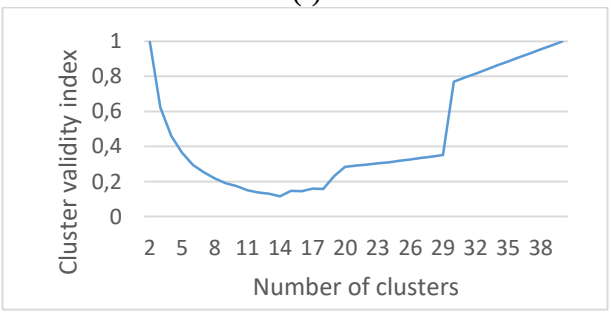
(c)



(f)



(d)



(g)

Figure III-20: Exemple d'un objet 3D complexe (a) avec ses images de coupe 2D représentatives correspondant à ses trois axes principaux en utilisant notre approche (b-d) et la fonction de mesure de l'indice de validité du cluster correspondant à ses axes principaux (e-g).

$$v_{under}(K,V,X) = \frac{1}{K}\sum_{i=1}^{K} MICD_i \qquad \text{(III-20)}$$

$$v_{over}(K,V) = \frac{K}{ICMD_{min}} \qquad \text{(III-21)}$$

Pour 2≤K≤Kmax.

Définissons les vecteurs de mesure de partition comme :

$$V_{under} = [v_{under}(2, V, X), \dots, v_{under}(K_{max}, V, X)] \quad \text{(III-22)}$$

$$V_{over} = [v_{over}(2, V), \dots, v_{over}(K_{max}, V)] \quad \text{(III-23)}$$

Pour chaque vecteur, les valeurs maximales et minimales sont calculées comme suit :

$$V_{under\_max} = \max_{K}(v_{under}(K, V, X)) \quad \text{(III-24)}$$

$$V_{over\_max} = \max_{K}(v_{over}(K, V)) \quad \text{(III-25)}$$

$$V_{under\_min} = \min_{K}(v_{under}(K, V, X)) \quad \text{(III-26)}$$

$$V_{over\_min} = \min_{K}(v_{over}(K, V)) \quad \text{(III-27)}$$

Pour $K = 1,2, \dots, K_{max}$

La normalisation de chaque fonction devient :

$$V_{underN}(K, V, X) = \frac{v_{under}(K, V, X) - v_{under\_min}}{v_{under\_max} - v_{under\_min}} \quad \text{(III-28)}$$

$$V_{overN}(K, V) = \frac{v_{over}(K, V) - v_{over\_min}}{v_{over\_max} - v_{over\_min}} \quad \text{(III-29)}$$

Par conséquent, v_{under} et v_{over} se situent toujours entre 0 et 1. La normalisation des vecteurs de mesure de partition est donc définie comme suit :

$$v_{underN} = [v_{underN}(2, V, X), \dots, v_{underN}(K_{max}, V, X)]^T \quad \text{(III-30)}$$

$$v_{overN} = [v_{overN}(2, V), \dots, v_{overN}(K_{max}, V)]^T \quad \text{(III-31)}$$

L'indice de validité, noté par V_{SV}, est formulé en additionnant v_{underN} et v_{overN}, s'écrit donc comme :

$$V_{SV}(K, V, X) = v_{underN}(K, V, X) + v_{overN}(K, V) \quad \text{(III-32)}$$

Le nombre optimal de groupe est obtenu pour la plus petite valeur de $V_{SV}(K, V, X)$ pour K variant de 2 à K_{max}. Dans notre méthode, l'application de cet indice de validité de cluster, prenant l'intervalle [2, 40], nous permet de déterminer entièrement le nombre optimal de cluster en fonction de la complexité de l'objet 3D. La figure III-19-c et la figure III-20-e-g montrent les fonctions de validité de cluster liées au nombre K. En fait, nous pouvons facilement remarquer une vallée au nombre optimal de clusters K*.

Par la suite, nous représentons chaque cluster par une seule image de coupe 2D, qui correspond à celle qui est la plus proche du centroïde du cluster, et nous l'appelons image de

coupe représentative pour former un sous-ensemble d'images de coupe représentatives (Representative Slices (RS)) :

$RS = \{RS_{ox}, RS_{oy}, RS_{oz}\}$ telles que: $RS_{ox} = \{RS_1^{ox}, \dots, RS_l^{ox}\}$, $RS_{oy} = \{RS_1^{oy}, \dots, RS_m^{oy}\}$ et $RS_{oz} = \{RS_1^{oz}, \dots, RS_n^{oz}\}$.

Où l, m et n représentent respectivement le nombre d'images de coupe représentatives correspondant aux axes X, Y et Z de l'objet 3D.

❖ **Calcul de la dissimilitude**

Dans notre approche, le problème de la correspondance des formes d'objets 3D est transformé en comment mesurer la dissimilitude entre les ensembles d'images de coupe représentatives, représentées par leurs moments de Hu, correspondant à différents objets 3D. Ainsi, une fois que les objets 3D sont représentés sous forme d'images de coupe représentatives, leur similarité peut être calculée en utilisant des fonctions de distance, telles que la distance euclidienne, la distance de Minkowsky, la distance de Manhattan, etc. Cependant, le nombre d'images représentatives dépend de la complexité de chaque objet 3D, ce qui donne des ensembles non ordonnés de tailles différentes.

Au début de notre travail, nous nous sommes intéressés à la distance de Hausdorff [Atallah et al. 1983] pour mesurer la dissemblance du contenu. Cette mesure minimise la comparaison de deux ensembles à une comparaison d'un seul élément de chacun. En fait, la distance de Hausdorff ne se soucie pas que tous les éléments d'un ensemble soient appariés, elle exprime seulement le degré de dissimilitude entre la paire d'éléments la plus dissemblable, et cette méthode de comparaison peut conduire à des résultats incorrects lorsque certains éléments perturbés existent dans un ensemble. Par exemple, supposons qu'il existe deux ensembles A et B composés respectivement de 50 éléments, dont 49 éléments sont très similaires, mais qu'une paire d'éléments est différente. Alors la mesure de la distance de Hausdorff donnera un résultat basé sur la dissimilitude entre la paire différente, sans tenir compte des 49 éléments similaires. Pour cela, nous définissons notre métrique qui prend en considération tous les éléments de l'ensemble.

Puisque chaque objet 3D O de la base de données est représenté par un ensemble d'images de coupe représentatives (Representative Slices (RS^O)), $O = \{RS_{ox}^O, RS_{oy}^O, RS_{oz}^O\}$ et l'objet 3D requête Q représenté par un ensemble d'images de coupe représentatives (RS^Q), $Q = \{RS_{ox}^Q, RS_{oy}^Q, RS_{oz}^Q\}$. La distance entre Q et O est donc définie comme suit :

$$D(O,Q) = \max\left(\frac{\sum_{1\le i\le N} inf_{1\le j\le M}\left(d_{O_iQ_j}\right)}{N}, \quad \frac{\sum_{1\le j\le M} inf_{1\le i\le N}\left(d_{O_iQ_j}\right)}{M}\right) \qquad \text{(III-33)}$$

Où $d_{O_iQ_j}$ représente la distance euclidienne entre les moments de Hu de i^{ème} image de coupe représentative de l'objet 3D O et les moments de Hu de j^{ème} image de coupe représentative de la requête Q, N et M indiquent respectivement le nombre global d'images de coupe représentatives correspondent à l'objet O et à la requête Q.

III.4.2 **Résultats expérimentaux**

Afin d'évaluer notre système, nous avons créé deux bases de données différentes. Tous les objets 3D utilisés dans les deux bases de données ont été tirés de la base de données Princeton Shape Benchmark (PSB) ; cette base de données est disponible gratuitement en ligne et largement utilisée dans de nombreux ouvrages. Elle est apparue en 2004 et contient 1814 objets 3D collectés sur Internet.

Pour réaliser la première base de données (DB1), nous avons choisi dans la base de données PSB 225 objets répartis selon leurs formes en 14 classes. Afin de démontrer la robustesse de notre approche concernant les objets 3D incomplets, nous avons créé une seconde base de données (DB2) composée de 225 objets normaux et 112 objets 3D incomplets. Les objets 3D incomplets ont été créés à l'aide du logiciel meshLab en retirant d'une façon aléatoire des parties de l'objet 3D.

Afin d'évaluer équitablement notre approche et de comparer le fonctionnement des méthodes de recherche d'objets 3D, nous utilisons des visualisations qualitatives (Courbes de rappel-précision, et les six premiers résultats de recherche) et des statistiques quantitatives (Nearest neighbor (NN), First-tier (FT) and Second-tier (ST), E-Measure (EM), Discounted Cumulative Gain (DCG), Normalized Discounted Cumulative Gain (N-DCG)) comme critère d'évaluation sur les deux bases de données. Ces métriques sont décrites avec plus de détails dans la section II-6.

Dans ce qui suit, nous allons décrire les tests réalisés, les résultats obtenus et une étude comparative de notre approche (ASC), Spherical Harmonics Descriptor (SHD) [Kazhdan et al. 2003], K-Representative Slices (K_RS) [Taybi et al. 2016], Shape Distribution (D2) [Osada et al. 2002] et (D2a) [Monteverdel et al. 2007], Extended Gaussian Image (EGI) [Horn 1984], et Shape Histogram (SHELL et SECTOR) [Ankerst et al. 1999a] sur les deux bases de données.

- **Première base de données (DB1)**

Dans cette section, nous présentons et comparons les résultats expérimentaux sur la première base de données (DB1). Tableau III-1 résume les performances de la recherche et la figure III-21 présente les courbes de rappel et de précision correspondantes de notre approche et des sept descripteurs mentionnés ci-dessus. La figure III-22 et la figure III-23 montrent respectivement les six premiers résultats de notre approche (ASC) et de Spherical Harmonics Descriptor (SHD) [Kazhdan et al. 2003].

Comme nous pouvons le voir, dans le tableau III-1 et la figure III-21, l'approche proposée (ASC), SHD et K_RS ont montré des performances supérieures par rapport aux autres descripteurs. Nous remarquons également que notre ASC obtient les meilleurs scores sur toutes les statistiques quantitatives utilisées (NN, FT, ST, EM, DCG et N-DCG), ce qui signifie que notre approche permet de retrouver les bons résultats en tête de liste plutôt que les bons résultats plus loin dans le classement.

Tableau III-1 : Performances de recherche pour la première base de données.

3D object descriptor	NN (%)	FT (%)	ST (%)	EM (%)	DCG (%)	N-DCG (%)
ASC	**88.00**	**61.85**	**75.60**	**53.59**	**85.12**	**19.35**
SHD	86.67	58.93	75.53	53.49	84.43	18.39
K_RS	84.00	57.12	71.47	50.33	81.72	14.58
D2a	57.33	36.27	55.55	37.55	67.12	-5.89
Sector	49.33	34.99	53.76	35.35	65.68	-7.90
Shell	50.67	34.46	53.17	35.48	65.32	-8.41
D2	48.44	33.61	53.61	36.33	63.61	-10.81
EGI	40.44	24.86	42.83	27.43	57.55	-19.31

La figure III-22 et la figure III-23 montrent les six premiers résultats de recherche en utilisant respectivement notre approche (ASC) et le descripteur SHD [Kazhdan et al. 2003]. Les premières colonnes de chaque figure montrent six requêtes de différentes classes, et chaque ligne montre les six premiers résultats de recherche en utilisant notre méthode (ASC) et le descripteur SHD. Comme le montrent les figures, notre approche donne de bons résultats pour presque toutes les requêtes. Le descripteur SHD donne aussi des résultats satisfaisants.

❖ **Deuxième base de données (DB2)**

Dans une deuxième série d'expériences, nous essayons de tester le comportement des descripteurs dans une base de données contenant des objets 3D complets et incomplets. Nous comparons principalement notre approche avec des descripteurs basés sur la description globale de l'objet 3D (SHD, D2, D2a, EGI, SHELL et SECTOR) ; seules nos approches (ASC et K_RS) utilisent la similarité partielle pour faire correspondre les objets 3D.

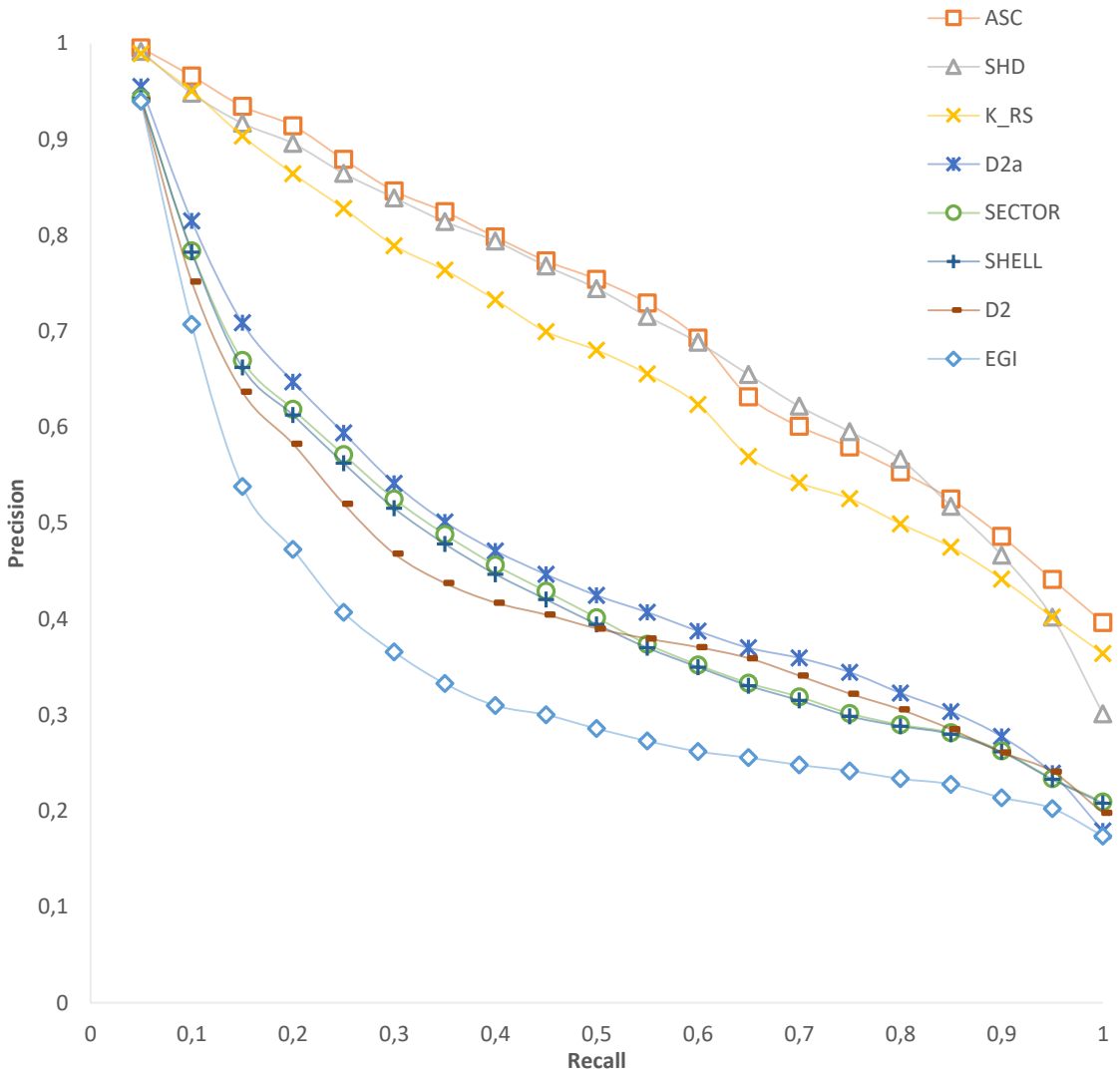


Figure III-21: courbes de rappel-précision de ASC, SHD, K_RS, D2, D2a, EGI, SHELL et SECTOR en utilisant la première base de données.

Requête Les six premiers objets 3D récupérés

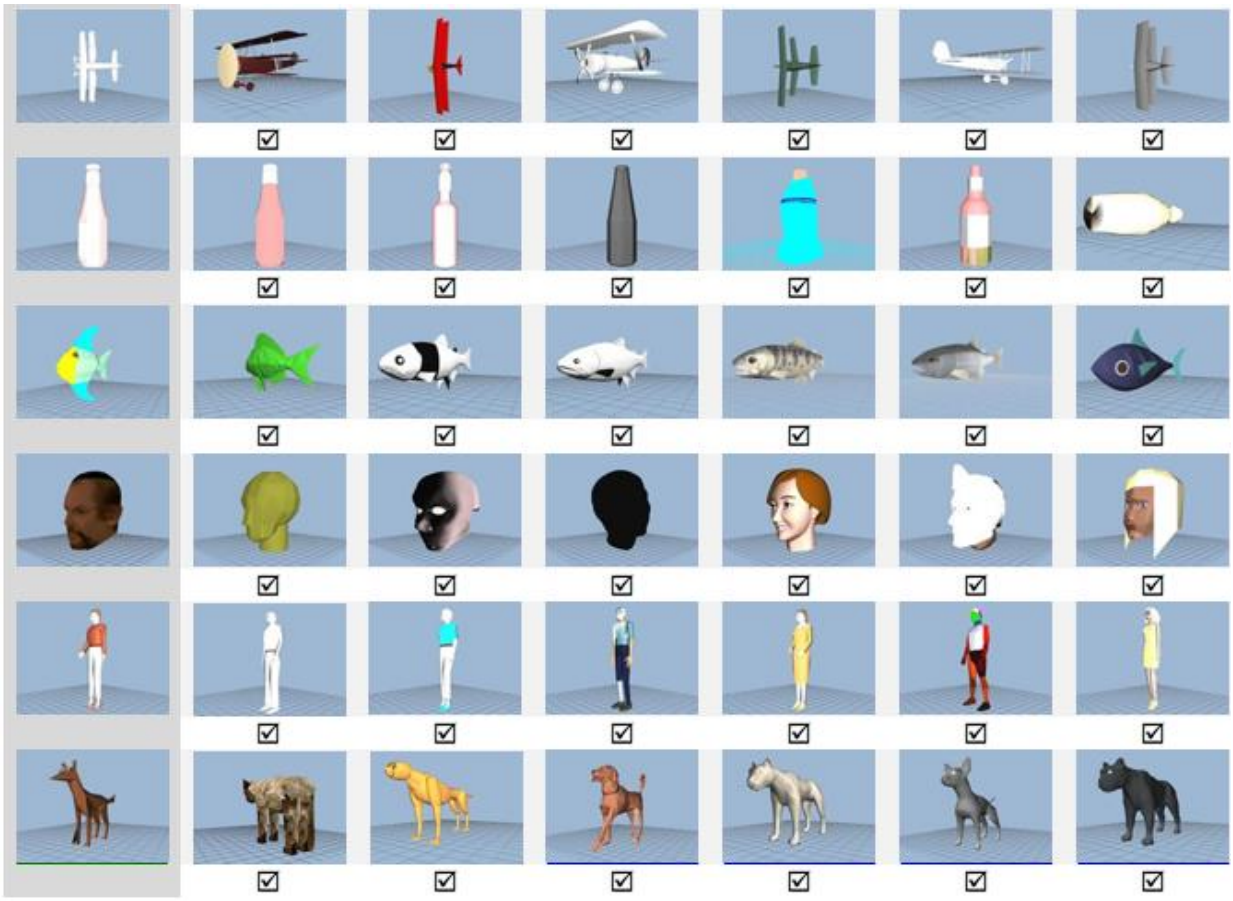


Figure III-22 : Les six premiers objets 3D récupérés dans la première base de données en utilisant notre ASC.

Tableau III-2: Performances de recherche pour la deuxième base de données.

3D object descriptor	NN (%)	FT (%)	ST (%)	E-M (%)	DCG (%)	N-DCG (%)
ASC	**94.36**	**62.23**	**75.69**	**57.20**	**87.67**	**19.28**
K_RS	91.39	57.41	71.51	53.17	84.75	15.31
SHD	91.10	55.30	72.08	51.95	84.44	14.89
D2a	62.31	34.52	53.72	33.89	69.60	-5.31
Sector	62.59	32.98	52.83	32.29	68.48	-6.83
Shell	64.09	32.46	52.50	31.92	67.98	-7.51
D2	43.92	31.67	52.26	31.93	64.74	-11.92
EGI	48.37	23.84	40.66	23.29	60.34	-17.90

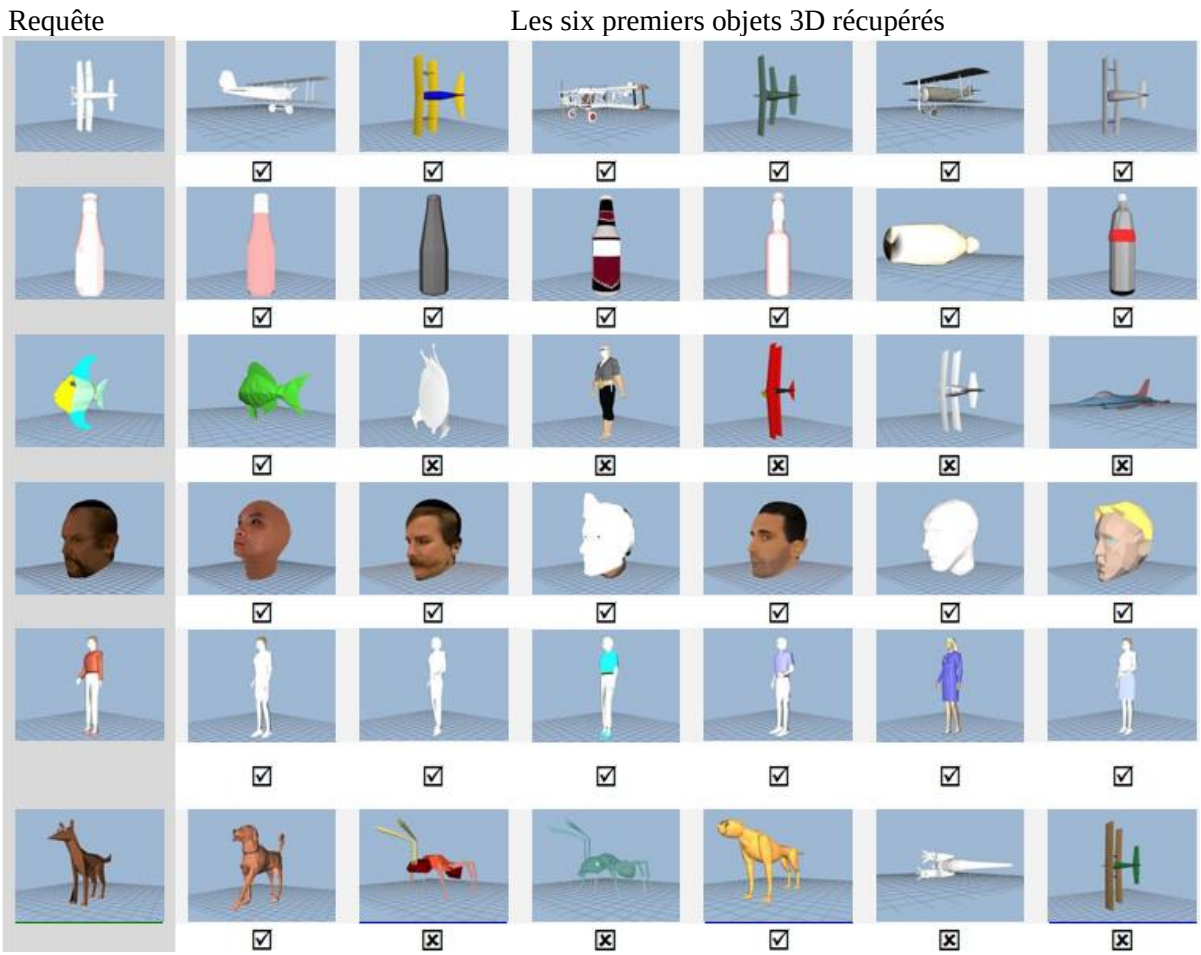


Figure III-23 : Les six premiers objets 3D récupérés dans la première base de données en utilisant le descripteur SHD.

Le tableau III-2 et la figure III-24 montrent respectivement les statistiques de recherche et les courbes de rappel-précision pour chaque descripteur. Comme nous pouvons déduire à partir du tableau III-2, notre approche (ASC) surpasse nettement les autres descripteurs de comparaison dans toutes les statistiques quantitatives utilisées (NN, FT, ST, EM, DCG et N-DCG). Surtout, dans la figure III-24, la courbe entière de rappel-précision de notre méthode diminue beaucoup plus lentement que les autres descripteurs chaque fois que le rappel augmente, ce qui est appréciable car cela prouve que l'approche est plus stable. De plus, notre approche présente un gain de performance élevé (jusqu'à 23 %) lorsque le rappel est si proche de 1. D'autre part, nous constatons que les performances des SHD, D2, D2a, Sector, Shell et EGI, par rapport à nos approches (ASC et K_RS), sont sensiblement réduites dans la seconde base de données, c'est à cause de la difficulté et de l'incapacité de ces descripteurs à récupérer correctement les objets 3D incomplets.

La figure III-25 et la figure III-26 représentent respectivement quelques objets 3D récupérés à l'aide de notre ASC et de SHD. Nous pouvons facilement déduire, à partir des résultats obtenus, que notre ASC fonctionne exceptionnellement bien dans la deuxième base de données (DB2), ce qui n'est pas le cas des résultats donnés par SHD.

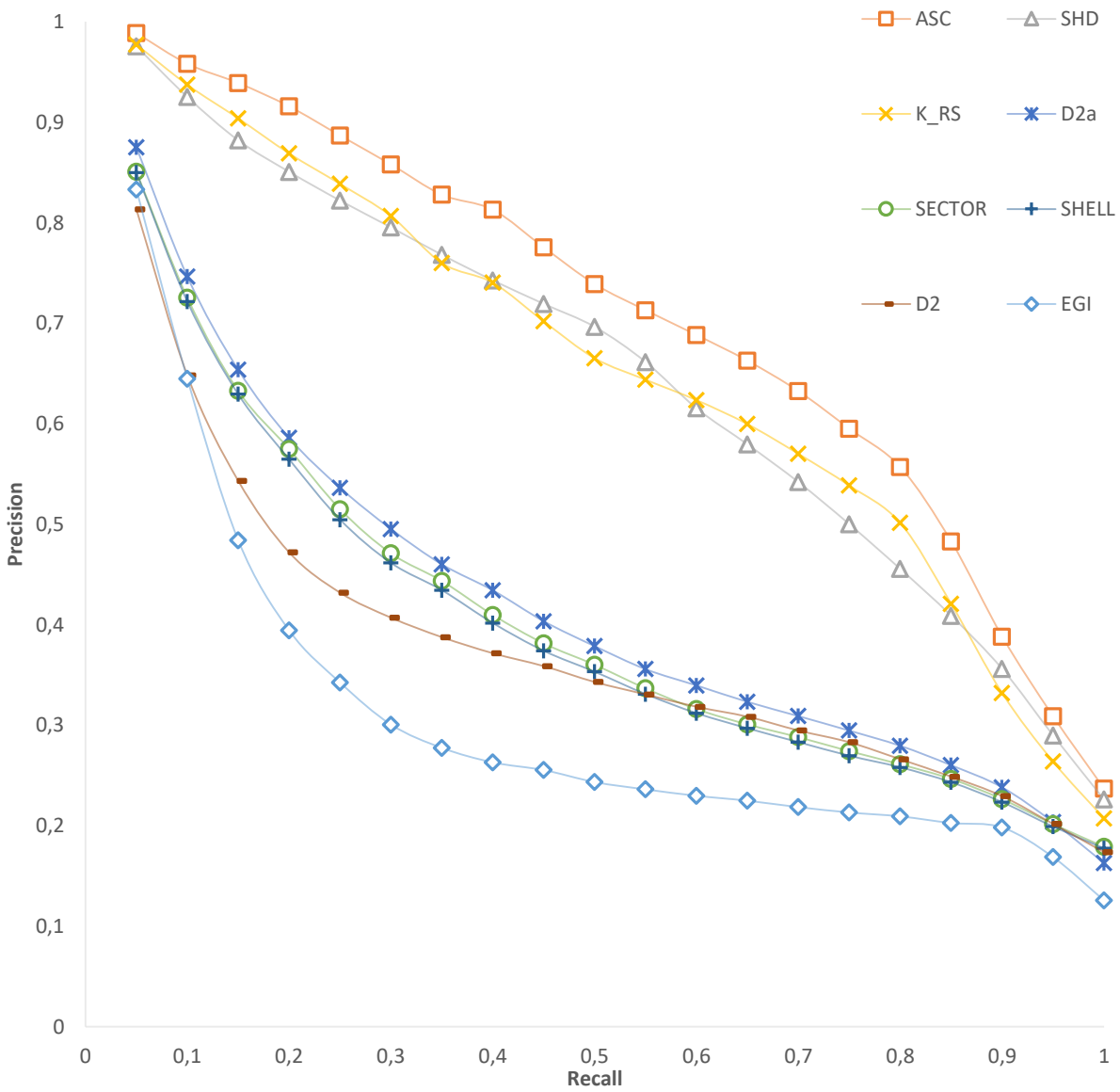


Figure III-24: courbes de précision/rappel de ASC, SHD, K_RS, D2, D2a, EGI, SHELL et SECTOR en utilisant la deuxième base de données.

III.4.3 **Conclusion**

Dans cette section, nous avons présenté notre deuxième approche pour l'indexation et la recherche d'objets 3D basée sur la similarité partielle. La première étape consiste à normaliser les objets 3D pour nous assurer que les objets similaires seront décomposés de la même manière. Ensuite, nous créons de nombreuses images de coupe 2D correspondant aux trois axes principaux de l'objet 3D. Puis, nous sélectionnons les images de coupe les plus représentatives en utilisant un indice de validité de cluster pour définir automatiquement le nombre optimal d'images de coupe représentatives. Enfin, nous utilisons la métrique que nous avons proposée pour calculer la similarité entre les ensembles d'images de coupe représentatives.

À partir des résultats expérimentaux, en particulier dans la deuxième base de données (DB2), nous pouvons déduire que notre méthode donne les meilleurs résultats en termes de performances de recherche (NN (94,36%), FT (62,23%), ST (75,69%), E-M (57,20%), DCG (87,67%), et N-DCG (19,28%)), surpassant certaines méthodes bien connues dans la littérature, qui caractérisent l'objet 3D de manière globale.

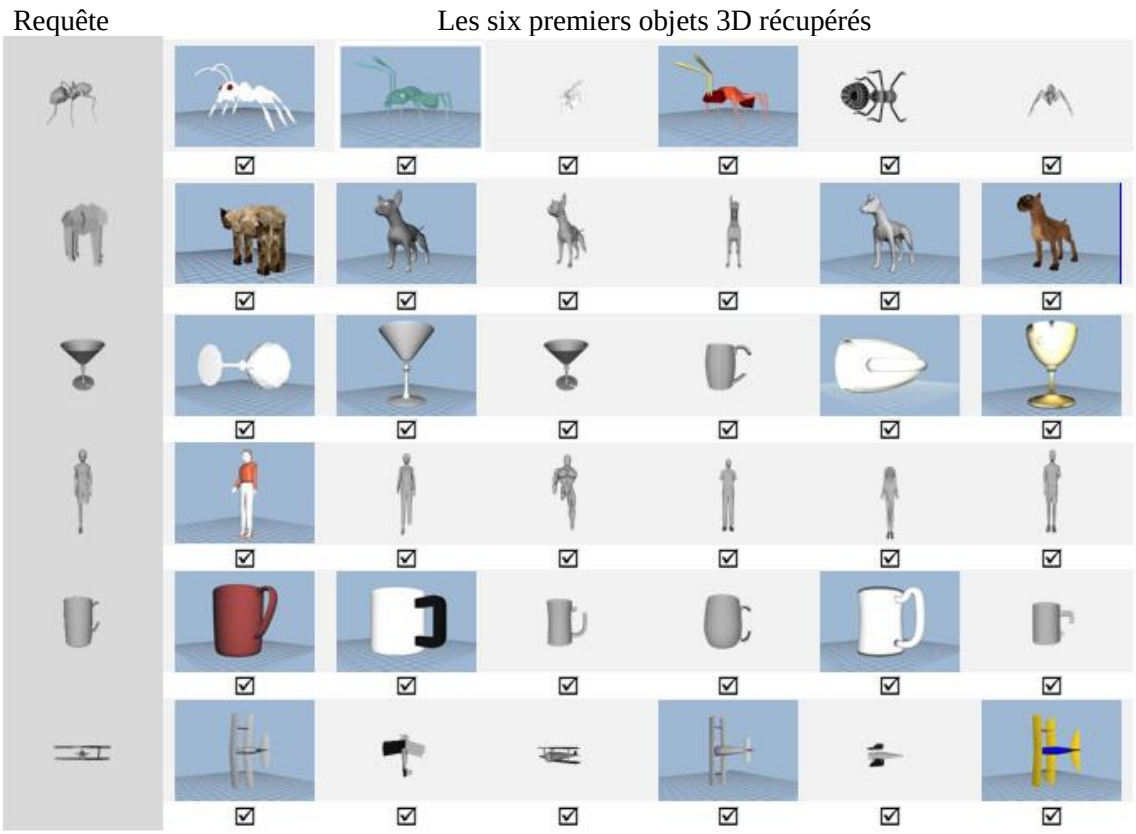


Figure III-25 : Les six premiers objets 3D récupérés dans la deuxième base de données en utilisant notre approche ASC.

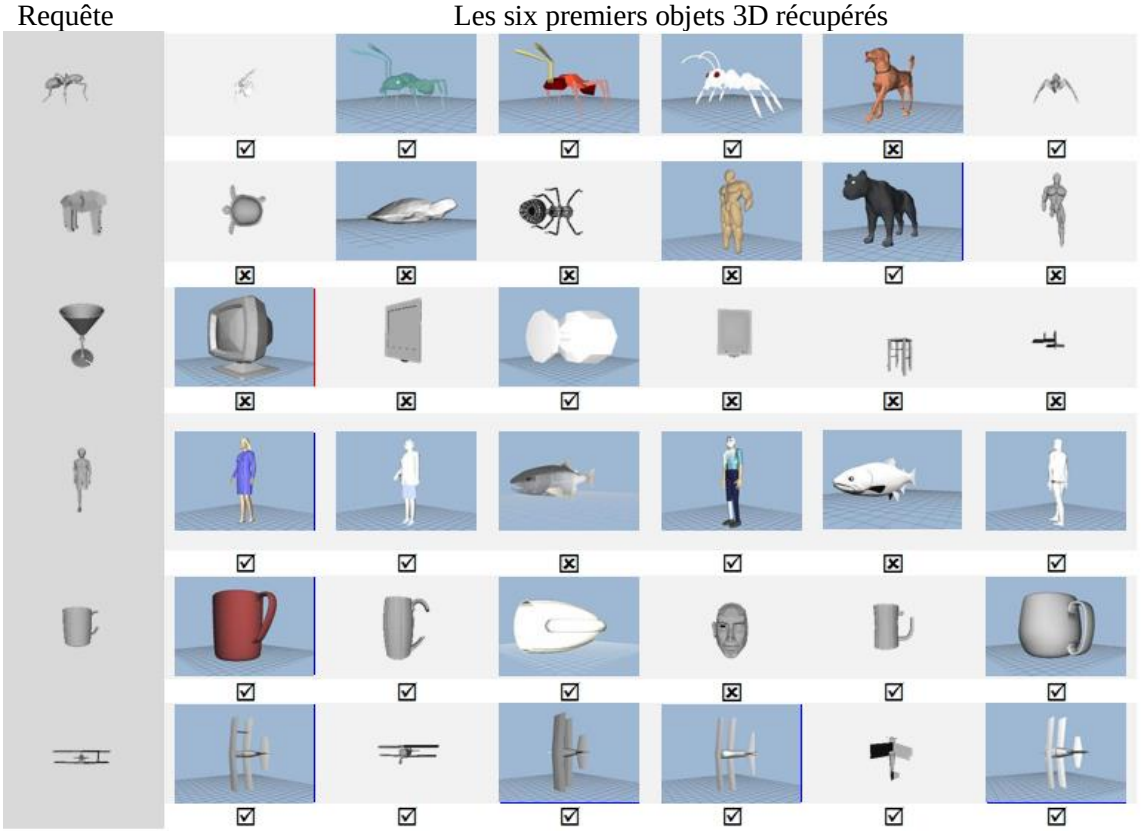


Figure III-26 : Les six premiers objets 3D récupérés dans la deuxième base de données en utilisant le descripteur SHD.

III.5 Troisième approche proposée pour l'indexation et la recherche d'objet 3D

III.5.1 Approche proposée

Dans cette section, nous présentons notre troisième méthode d'indexation et de recherche d'objets 3D qui combine les images de coupe 2D et l'algorithme Apriori [Taybi et al 2020]. L'idée principale de l'approche est de représenter les objets 3D par un ensemble d'images de coupe 2D transformant le problème de correspondance de forme entre les objets 3D en mesurant la similarité entre leurs images de coupe. La figure III-27 montre l'architecture de l'approche proposée. Tout d'abord, nous commençons par la normalisation des objets 3D pour assurer l'invariance au niveau de l'échelle, de la translation et de la rotation. Puis, pour chaque objet 3D, nous extrayons un ensemble initial d'images de coupe 2D correspondant à des axes déterminés. Ensuite, nous décrivons chaque image de coupe 2D par un vecteur de moments de Zernike. Après, nous représentons les images de coupe de chaque objet 3D dans une base de données transactionnelle. Par la suite, nous appliquons l'algorithme Apriori pour diminuer l'ensemble initial d'images de coupe en sélectionnant les plus représentatives. Enfin, nous utilisons notre propre métrique pour mesurer la similarité entre les images de coupe représentatives des objets 3D.

❖ **Normalisation des objets 3D**

En général, les objets 3D sont fournis dans des positions, des orientations et des échelles aléatoires dans l'espace 3D. Dans de nombreux processus d'extraction de caractéristiques d'objet 3D, il est nécessaire de normaliser l'orientation, la position et la taille de l'objet 3D avant l'extraction de caractéristiques pour garantir une représentation distinctive. En effet, l'étape de normalisation vise à garantir que les objets 3D similaires ayant des positions, des orientations et des échelles différentes peuvent être correctement représentés par presque la même signature.

Par conséquent, pour garantir l'invariance de notre descripteur aux transformations affines, qui correspond à placer l'objet 3D en coordonnées canoniques, nous translatons le centre de masse de l'objet 3D pour qu'il coïncide avec l'origine. Pour la normalisation de l'échelle, la taille de l'objet 3D est modifiée afin que la distance moyenne de sa surface par rapport à son centroïde soit égale à 1. L'analyse en composantes principales (ACP) est utilisée pour atteindre la normalisation de la rotation.

❖ **Création de l'ensemble initial d'images de coupe**

Notre approche consiste à créer un ensemble d'images de coupe 2D obtenues par l'intersection d'un ensemble de plans avec le maillage triangulaire 3D. En fait, les maillages triangulaires constituent un moyen efficace de représenter des objets 3D. De manière caractéristique, les données de géométrie et de connectivité sont utilisées à la fois pour représenter un maillage triangulaire 3D. Afin de créer l'ensemble initial d'images de coupe 2D, nous prenons l'intersection du maillage triangulaire 3D avec des plans équidistants et orthogonaux aux axes déterminés. La figure III-28 montre un exemple d'un objet 3D, à une position donnée, avec ses images de coupe correspondant à son axe OY en utilisant notre approche. La méthode d'extraction des images de coupe est décrite en détails dans la section III-3.

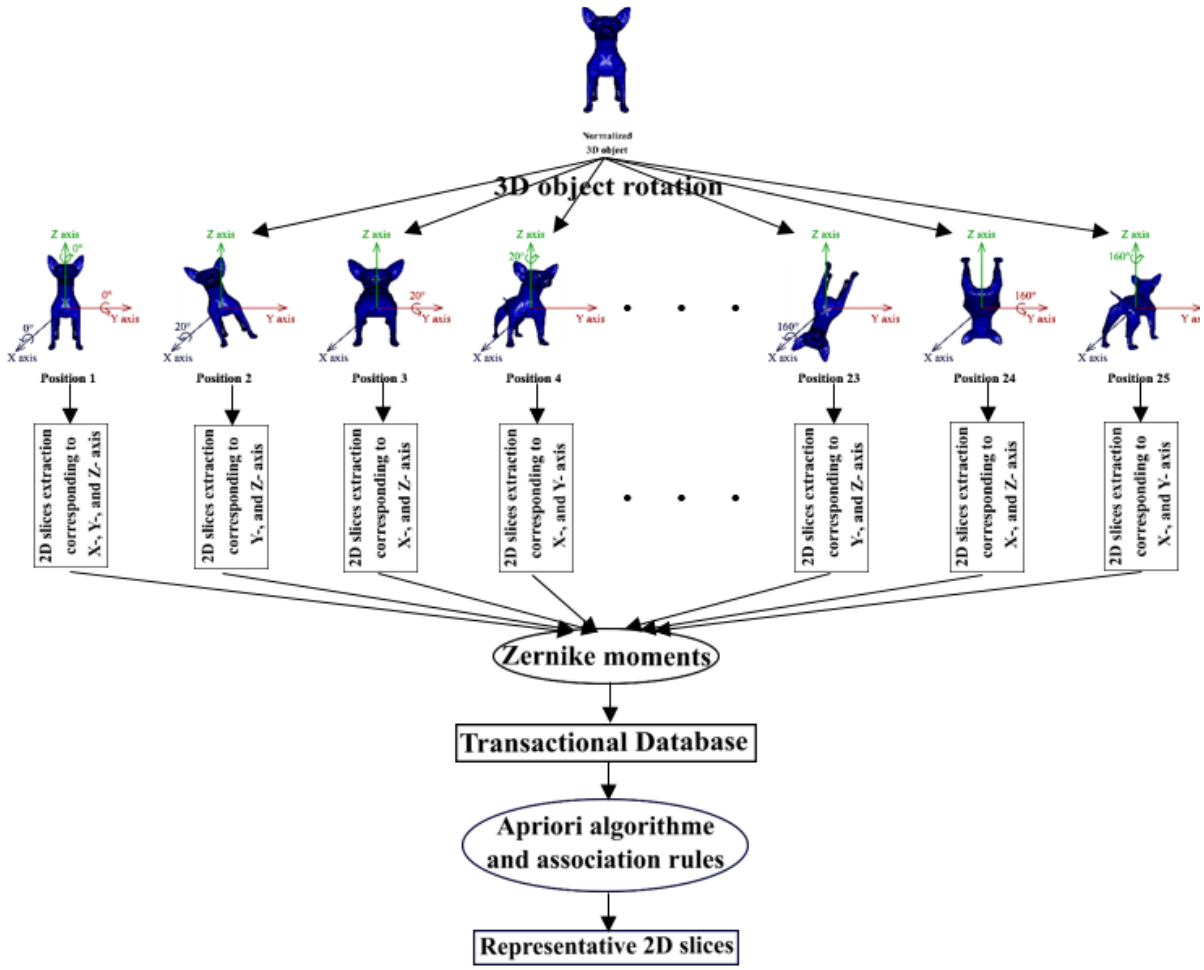


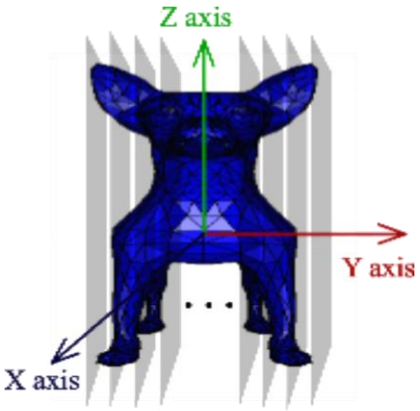
Figure III-27: L'architecture de l'approche proposée.

Au départ, nous prenons, pour chaque axe cartésien (axes Ox, Oy et Oz), l'intersection de l'objet 3D normalisé avec 50 plans équidistants et orthogonaux à l'axe. Ensuite, nous tournons l'objet 3D sur les trois axes cartésiens (axe par axe) par 20° jusqu'à ce que nous atteignions les 160°. À chaque rotation de l'objet 3D, nous captons, pour seulement deux axes cartésiens, l'intersection de l'objet 3D tourné avec 50 plans équidistants et orthogonaux à l'axe (comme le montre la figure III-27) ; c'est-à-dire que si nous avons tourné l'objet 3D sur l'axe Ox, nous extrayons les images de coupe correspondant aux axes Oy et Oz.

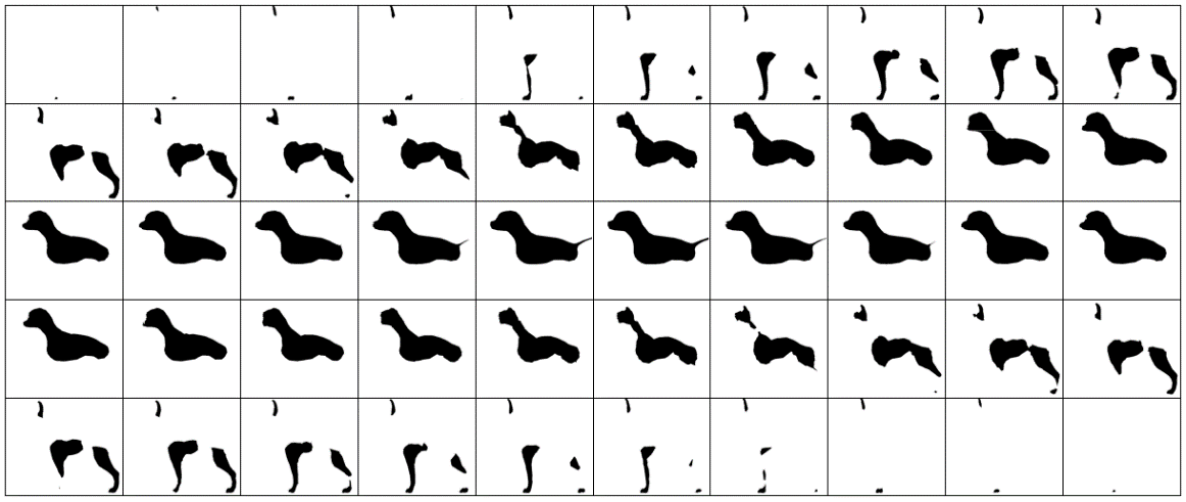
En fait, lorsque nous tournons l'objet 3D sur un axe, les images de coupe 2D correspondantes à cet axe restent les mêmes que les images de coupe correspondant au même axe dans la première position ; il n'y a qu'un changement de rotation. Comme nous utiliserons le descripteur de Zernike, qui est un descripteur invariant par rapport à la rotation, pour caractériser les images de coupe 2D, il est judicieux d'éliminer les images de coupe correspondant à l'axe de rotation de l'objet 3D.

- **Calcul de la signature numérique pour chaque image de coupe**

Parmi les descripteurs d'images existant dans la littérature, les moments de Zernike sont considérés comme les plus appropriés pour représenter les images 2D, en raison de ses caractéristiques distinctives telles que l'invariance en rotation, la taille réduite des caractéristiques, la capacité de représenter finement les images, etc. Les moments de Zernike ont été utilisés avec succès dans diverses applications d'analyse d'images et de reconnaissance d'objets [Tan et al. 2014] [Dai et al. 2014] [Singh et al. 2011] [Gao et al. 2011]. Les auteurs de [Zhang et al. 2004] ont observé que les moments de Zernike sont très



(a)



(b)

Figure III-28: Exemple d'un objet 3D (a) avec ses images de coupe 2D correspondant à son axe OY.

.

utiles pour capturer les principales caractéristiques des images. Cela est dû au fait que ces moments sont orthogonaux par nature, ce qui garantit que les valeurs des moments à différents ordres représentent des caractéristiques indépendantes et uniques d'une image.

En fait, les moments de Zernike sont définis comme les projections de f(x, y) sur une classe de polynômes, appelés polynômes de Zernike. La fonction de base de Zernike avec l'ordre n et la répétition m est définie sur un cercle unitaire en coordonnées polaires comme suit :

$$V_{nm}(\rho,\theta) = R_{nm}(\rho)e^{jm\theta} \tag{III-34}$$

Où $R_{nm}(\rho)$ sont des polynômes radiaux à valeur réelle. Sa définition se trouve dans [Teague 1980]. Ici, n est un nombre entier non négatif et m est un nombre entier remplissant les conditions : $n-|m|$ est pair et $|m| \leq n$. Ce qui suit indique la propriété orthogonale de $V_{nm}(\rho,\theta)$:

$$\iint\limits_{\substack{0\leq\rho\leq1\\0\leq\theta\leq2\pi}} V_{nm}^{*}(\rho,\theta)V_{n'm'}(\rho,\theta)\rho d\rho d\theta = \frac{\pi}{n+1}\delta_{nn'}\delta_{mm'} \tag{III-35}$$

Où le symbole $*$ désigne le conjugué complexe et $\delta_{nn'}$ satisfait

$$\delta_{nn'} = \begin{cases} 1, & n = n' \\ 0, & sinon. \end{cases} \quad \text{(III-36)}$$

Les moments de Zernike d'ordre n avec répétition m pour une fonction d'image continue f(x, y) sur un disque unitaire est :

$$A_{nm} = \frac{n+1}{\pi} \int \int_{unit\ disk} V_{nm}^{*}(x,y) f(x,y) dxdy \quad \text{(III-37)}$$

Pour les images numériques, les intégrales peuvent être remplacées par des sommations :

$$A_{nm} = \frac{n+1}{\pi} \sum_{x} \sum_{y} f(x,y) V_{nm}^{*}(x,y), \quad x^2 + y^2 \leq 1 \quad \text{(III-38)}$$

Supposons que θ_0 soit l'angle de rotation ; A_{nm}et A_{nm}^{R}représentent respectivement les moments de Zernike originaux et la version tournée. Nous avons

$$|A_{nm}^{R}| = |A_{nm} \exp(-jm\theta_0)| = A_{nm}; \quad \phi_{nm}^{R} = \phi_{nm} - m\theta_0 \quad \text{(III-39)}$$

Où $|A_{nm}|$ et ϕ_{nm} représentent respectivement le module et l'argument. Dans (III-39), le module reste le même, alors que l'argument change avec la rotation de l'image. Par conséquent, de nombreuses applications [Ballesteros et al. 2005], [Gope et al. 2004], [Maaoui et al. 2005] utilisent uniquement les modules des moments de Zernike comme caractéristiques invariants en rotation de l'image. Aussi dans notre approche, nous n'avons utilisé que les modules des moments de Zernike pour caractériser les images de coupe de chaque objet 3D.

Sans parler du fait que les moments de Zernike d'ordre élevé ont non seulement une grande complexité de calcul [Hwang et al. 2006][Wee1 et al. 2007][Fu et al. 2007], mais représentent également une grande sensibilité au bruit [Theh et al. 1988], et peuvent diminuer les performances du système s'ils ne sont pas sélectionnés avec précision. En fait, le nombre de coefficients requis n'a pas besoin d'être important, puisque les caractéristiques de la forme peuvent normalement être saisies par quelques coefficients de basse fréquence seulement. De plus, puisque $A_{n,-m} = A_{n,m}^{*}$, seul $A_{n,m}(m \geq 0)$est nécessaire pour saisir les caractéristiques de la forme. C'est pourquoi un groupe de moments de Zernike de bas ordre a été extrait à partir des images de coupe 2D. Ce groupe de moments de Zernike est présenté sous forme de tableau dans le tableau III-3. Le groupe comprend 36 moments de bas ordre qui satisfont les conditions suivantes :

$$\{A_{n,m}\} \forall \begin{cases} 0 \leq n \leq 10 \\ |m| \leq n \\ n - |m| = 2k \\ k \in N \end{cases} \quad \text{(III-40)}$$

Tableau III-3 : Les moments de Zernike utilisés

n	m	Nombre de moments
0	0	36
1	1	
2	0, 2	
3	1, 3	
4	0, 2, 4	
5	1, 3, 5	
6	0, 2, 4, 6	
7	1, 3, 5, 7	
8	0, 2, 4, 6, 8	
9	1, 3, 5, 7, 9	
10	0, 2, 4, 6, 8, 10	

❖ **Sélection les images de coupe représentatives**

Maintenant que nous avons caractérisé chaque image de coupe 2D par un ensemble de moments de Zernike, nous utilisons l'algorithme Apriori pour sélectionner les plus représentatives. Pour ce faire, nous représentons les images de coupe 2D de chaque objet 3D dans une base de données transactionnelle. Ainsi, chaque ligne (transaction) de la base de données transactionnelle correspond aux images de coupe 2D d'un axe d'extraction.

Afin d'étiqueter l'ensemble initial d'images de coupe dans la base de données transactionnelle, nous utilisons l'indice de validité de clusters proposé par [Do-Jong et al. 2001] pour définir automatiquement le nombre optimal de clusters en fonction de la complexité de l'objet 3D. Nous invitons le lecteur à voir notre deuxième approche proposée dans la section précédente (section III-4), qui fournit plus de détails sur cet indice de validité de clusters.

Le nombre optimal de groupe est obtenu pour la plus petite valeur de l'indice pour K variant de 2 à K_{max}. Dans notre méthode, l'application de cet indice de validité de cluster, prenant l'intervalle [2, 500], nous permet de déterminer entièrement le nombre optimal de cluster en fonction de la complexité de l'objet 3D.

Maintenant que nous avons déterminé le nombre optimal de clusters en fonction de la complexité de l'objet 3D, nous attribuons une étiquette identique aux images de coupe qui se trouvent dans le même cluster. Ensuite, dans chaque ligne de la base de données transactionnelle, nous réduisons au minimum le nombre d'items (qui correspondent aux images de coupe) en éliminant les redondances.

Pour extraire les images de coupe représentatives, nous utilisons la base de données transactionnelle associée à chaque objet 3D, et nous appliquons l'algorithme Apriori pour extraire les Itemsets fréquents et, par la suite, les règles d'association qui seront utilisées pour déterminer les images de coupe associées. Après quelques expériences menées pour

déterminer le minsup et la minconf appropriés, nous concluons qu'il est approprié de choisir 25 % et 90 % comme seuils, respectivement, de minsup et de minconf. Pour plus de détails sur l'algorithme Apriori nous invitons le lecteur à voir la section III-2-2.

❖ **Calcul de la similarité**

L'objectif de la mesure de similarité est de maintenir les distances les plus petites possibles pour des objets similaires, et de rendre les objets dissemblables aussi loin que possible dans l'espace des caractéristiques. Par conséquent, les mesures de similarité appropriées doivent être conçues pour calculer avec précision la similarité du contenu.

Dans notre approche, nous avons représenté chaque objet 3D par un ensemble d'images de coupe caractéristiques, transformant ainsi la problématique de la correspondance entre les objets 3D en comment calculer la similarité entre leurs images de coupe représentatives. Ainsi, la correspondance individuelle s'est transformée en une correspondance multiple. Parmi les nombreuses mesures de distance existantes, la distance de Hausdorff a démontré son efficacité et sa puissance dans les travaux de recherche actuels [Zhao et al. 2015] [Gao et al. 2013].

Considérons deux ensembles $A = \{a_1, \dots, a_l\}$ et $B = \{b_1, \dots, b_m\}$, la distance de Hausdorff H(A,B) calcule le niveau de discordance entre A et B en calculant la distance du point de A qui est le plus éloigné de tout point de B et vice versa. En fait, la distance de Hausdorff accorde une attention particulière à la dissimilitude des deux ensembles, mais cela peut conduire à des résultats inappropriés lorsque certains composants perturbés existent dans un ensemble. Par exemple, supposons que tous les éléments de A et B présentent une forte similitude, sauf une paire qui est différente. La distance de Hausdorff ignorera tous les éléments similaires en ne prenant en considération que la dissimilitude entre les paires les plus différentes.

Afin de surmonter la faiblesse de la distance de Hausdorff, nous avons défini notre métrique qui est basée sur la distance de Hausdorff, en prenant en considération la dissimilitude entre toutes les paires dans les deux ensembles. Par conséquent, la dissimilitude entre les N images de coupe représentatives de l'objet O et celles de la requête Q est définie comme suit :

$$D(O,Q) = \max\left(\frac{\sum_{1\le i\le N} inf_{1\le j\le M}\left(d_{O_iQ_j}\right)}{N}, \quad \frac{\sum_{1\le j\le M} inf_{1\le i\le N}\left(d_{O_iQ_j}\right)}{M}\right) \quad \text{(III-41)}$$

Où $d_{O_iQ_j}$ représente la distance euclidienne entre les moments de Zernike de i^{ème} image de coupe représentative de l'objet 3D O et les moments de Zernike de j^{ème} image de coupe représentative de la requête Q.

III.5.2 Résultats expérimentaux

Dans cette étude, les objets 3D de la base de données Princeton Shape Benchmark (PSB) sont utilisés pour évaluer notre approche ; cette base de données est disponible gratuitement en ligne et largement utilisée dans de nombreux ouvrages. Elle contient 1814 objets 3D collectés depuis Internet, et classés manuellement selon la fonction et la forme des objets 3D. Elle comprend un ensemble de classifications hiérarchiques, des ensembles d'apprentissage et de test séparés, des annotations pour chaque modèle, et une suite d'outils logiciels pour la génération, l'analyse et la visualisation des résultats de la correspondance des formes [Shilane et al. 2004].

Afin d'étudier les performances de notre approche, nous l'avons comparée à 12 approches d'indexation et de recherche d'objets 3D utilisées dans le PSB: à savoir : D2 Shape Distribution (D2) [Osada et al. 2001] ; Extended Gaussian Image (EGI) [Horn 1984] ; Complex Extended Gaussian Image (CEGI) [Kang et al. 1991] ; Shape Histogram

(SHELLS, SECTORS, SECSHEL) [Ankerst et al. 1999a], Spherical Extent Function (EXT) [Saupe et al. 2001] ; Radialized Spherical Extent Function (REXT) [Vranic 2003] ; Gaussian Euclidean Distance Transform (GEDT) [Kazhdan et al. 2003] ; Spherical Harmonic Descriptor (SHD) [Kazhdan et al. 2003], and Light Field Descriptor (LFD) [Chen et al. 2003]).

Pour examiner objectivement notre approche, nous avons utilisé les outils d'évaluation du PSB en ce qui concerne la classification de base. En fait, les outils d'évaluation proposés par le PSB génèrent des visualisations (Precision-recall plot, Tier image, and the top five retrieval results), et des statistiques (Nearest neighbor (NN), First-tier (FT) and Second-tier (ST), E-Measure (EM), Discounted Cumulative Gain (DCG), and Normalized Discounted Cumulative Gain (N-DCG)) pour faciliter la comparaison des approches d'indexation et de recherche d'objets 3D. Nous invitons le lecteur à voir la section II-6, qui fournit plus de détails sur les critères d'évaluation.

Le tableau III-4 résume les statistiques de recherche pour chaque méthode. LFD a légèrement surpassé notre approche dans ST (48,7% contre 48,6%), et E-Measure (28,0% contre 27,7%). Cependant, notre approche donne les meilleurs scores dans les mesures de similarité de résultats les plus proches (NN (74,0%), FT (39,6%), DCG (66,8%), et N-DCG (23,5%)), ce qui signifie que notre méthode est la meilleure en ce qui concerne le placement des résultats correctes en tête de liste d'objets 3D récupérés qu'à la fin.

La figure III-29 montre les courbes de rappel-précision pour chaque descripteur. Comme nous pouvons l'observer, les courbes de rappel-précision montrent que notre approche est plus performante que les autres méthodes, confirmant les statistiques de recherche présentées dans le tableau III-4. En outre, lorsque le rappel augmente, la courbe entière de notre approche diminue lentement relativement aux autres descripteurs, ce qui signifie que notre méthode est plus stable.

La figure III-30 présente une image visualisant les correspondances "Nearest Neighbor" (blanc), "First Tier" (jaune) et "Second Tier" (orange) en utilisant notre approche sur la base de données PSB. Une approche de recherche solide devrait avoir un groupe de pixels blanc-jaune dans les blocs de la taille de la classe le long de la diagonale. Comme nous pouvons le voir sur la figure III-30, notre méthode comporte des pixels plus brillants dans les blocs diagonaux de la classe, ce qui montre que les objets 3D de la même classe présentent une grande similarité.

La figure III-31 montre une partie de la recherche d'objets 3D sur l'ensemble de test du PSB en utilisant notre approche. La première colonne de la figure montre les objets 3D requêtes et les autres colonnes présentent les dix objets 3D récupérés par ordre de classement. Comme le montrent les résultats obtenus par notre méthode, pratiquement tous les modèles 3D extraits appartiennent à la classe de l'objet recherché.

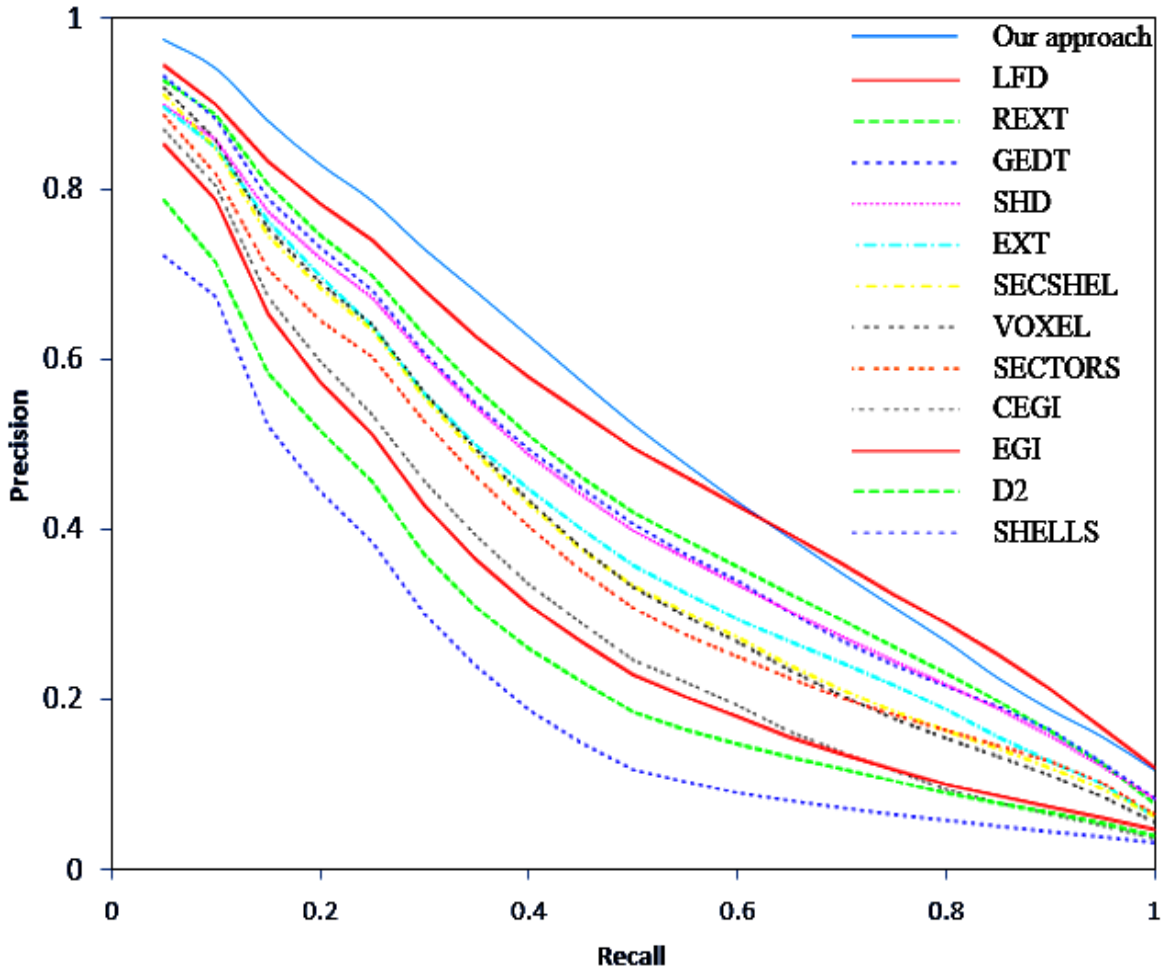


Figure III-29: Courbe Rappel-Précision en utilisant notre descripteur ainsi que 12 autres descripteurs de la littérature.

Tableau III-4 : Comparaisons entre les performances de notre descripteur ainsi que 12 autres descripteurs de la littérature.

Shape descriptors	NN	FT	ST	E-Measure	DCG	N-DCG
Our approach	**74.0%**	**39.6%**	**48.6%**	**27.7%**	**66.8%**	**23.5%**
LFD	65.7%	38.0%	48.7%	28.0%	64.3%	18.9%
REXT	60.2%	32.7%	43.2%	25.4%	60.1%	11.1%
SHD	55.6%	30.9%	41.1%	24.1%	58.4%	8.0%
GEDT	60.3%	31.3%	40.7%	23.7%	58.4%	8.0%
EXT	54.9%	28.6%	37.9%	21.9%	56.2%	3.9%
SECSHEL	54.6%	26.7%	35.0%	20.9%	54.5%	0.8%
VOXEL	54.0%	26.7%	35.3%	20.7%	54.3%	0.4%
SECTORS	50.4%	24.9%	33.4%	19.8%	52.9%	-2.2%
CEGI	42.0%	21.1%	28.7%	17.0%	47.9%	-11.4%
EGI	37.7%	19.7%	27.7%	16.5%	47.2%	-12.7%
D2	31.1%	15.8%	23.5%	13.9%	43.4%	-19.7%
SHELLS	22.7%	11.1%	17.3%	10.2%	38.6%	-28.6%

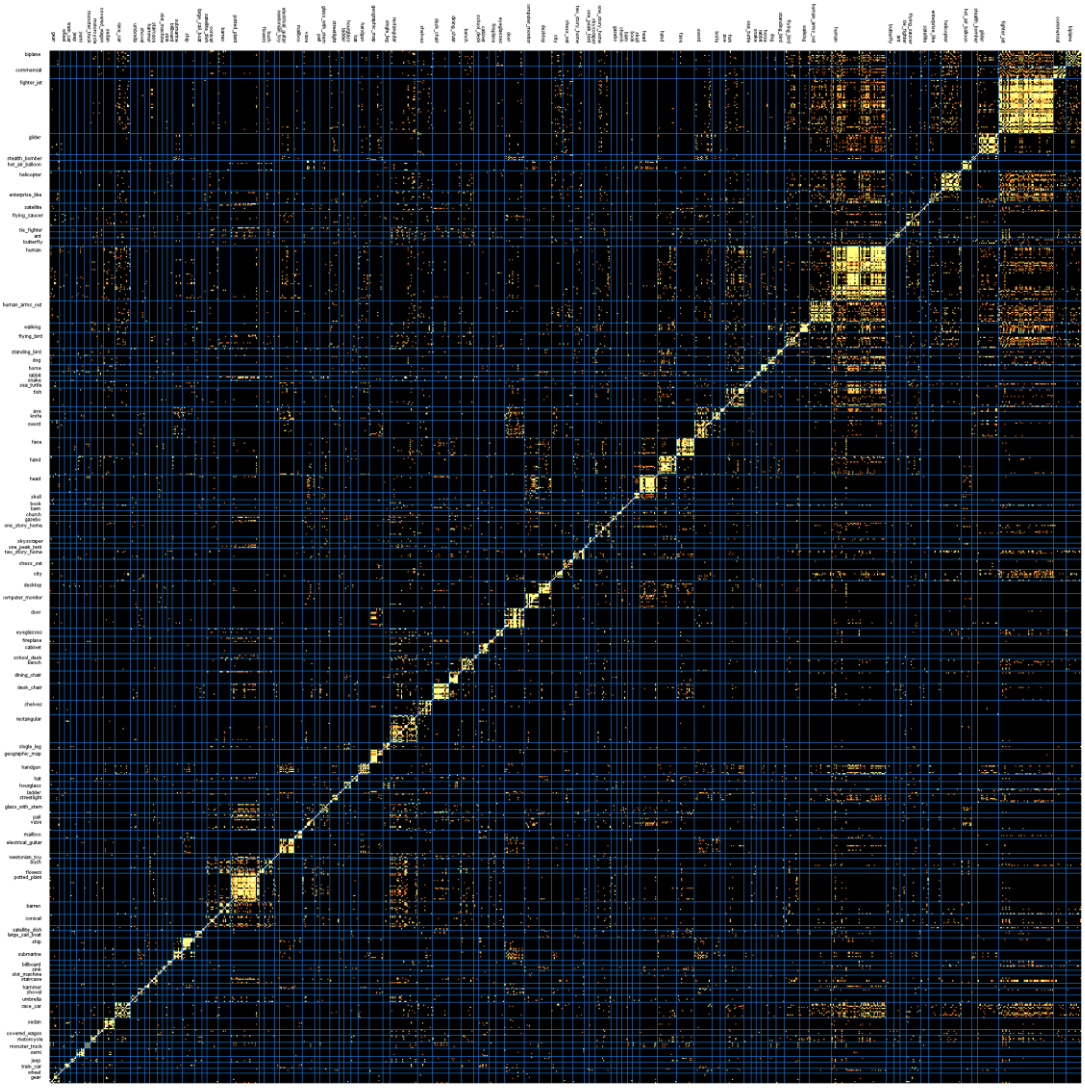


Figure III-30: Image de niveau visualisant "Nearest Neighbor" (blanc), "First Tier" (jaune) et "Second Tier" (orange) calculée en faisant correspondre chaque objet 3D (lignes) avec chaque autre objet 3D (colonnes) dans la base de données PSB en utilisant notre approche.

Afin d'étudier la stabilité de notre méthode par rapport aux objets 3D incomplets, des expériences supplémentaires ont été réalisées. En fait, nous avons créé un ensemble d'objets 3D incomplets à l'aide du logiciel meshLab en retirant arbitrairement des parties de l'objet 3D, et nous les avons utilisés comme requêtes. La figure III-32 montre quelques exemples de résultats obtenus à l'aide de notre méthode. La première colonne présente sept objets 3D incomplets utilisés comme des requêtes, et chaque ligne présente les dix premiers résultats de recherche obtenus en utilisant notre méthode. Les résultats obtenus nous permettent de déduire que notre approche fonctionne bien et parvient à faire correspondre correctement les objets 3D incomplets.

Requête Les dix premiers objets 3D retrouvés

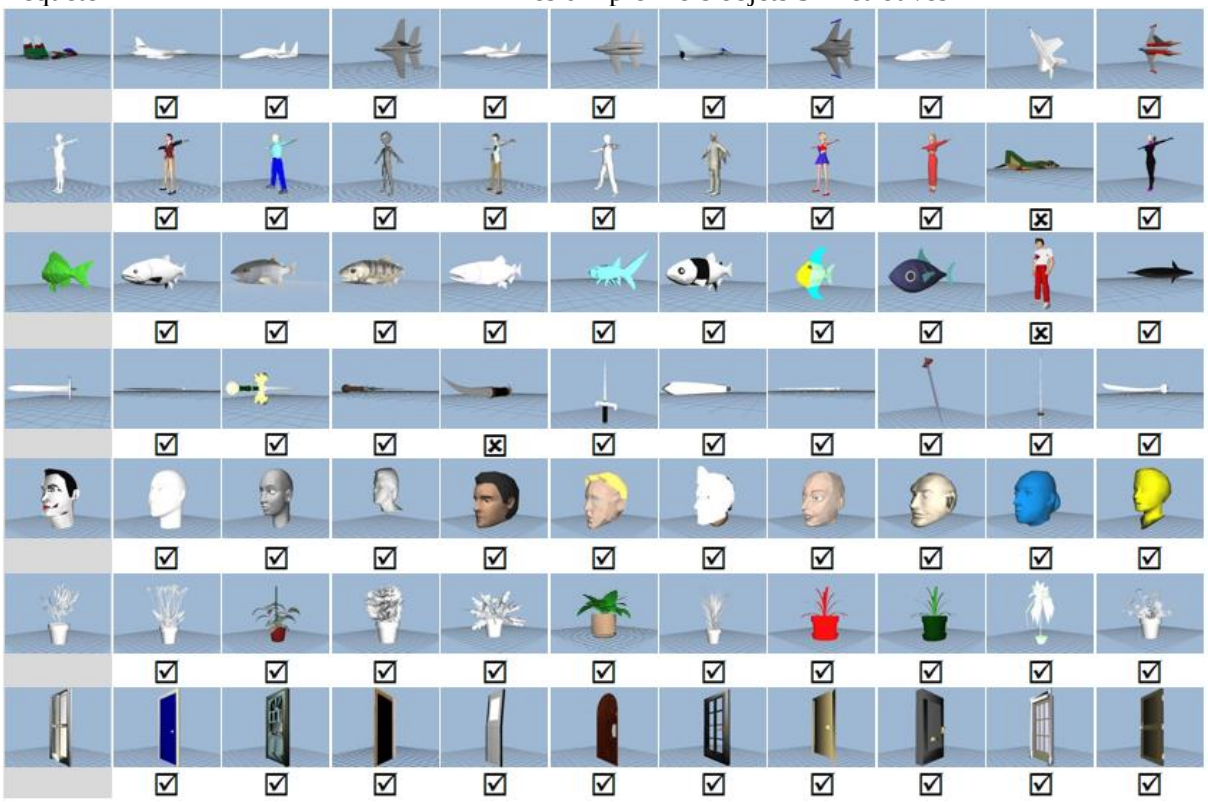


Figure III-31 : Les 10 premiers objets 3D retrouvés en utilisant l'approche proposée avec des requêtes normales.

Requête Les dix premiers objets 3D retrouvés

Figure III-32 : Les 10 premiers objets 3D retrouvés en utilisant l'approche proposée avec des requêtes incomplètes.

III.5.3 **Conclusion**

Dans cette section, nous avons présenté une nouvelle approche d'indexation et de recherche basée sur la similarité partielle d'objets 3D combinant des images de coupe 2D et l'algorithme Apriori. L'idée principale de notre travail était de tirer profit des images de coupe et des algorithmes de Data mining pour bien décrire la forme des objets 3D, qu'ils soient complets ou incomplets. En fait, nous avons utilisé l'algorithme Apriori pour choisir parmi un ensemble initial d’images de coupe 2D extraites à partir de l'objet 3D, les plus représentatives, et les utiliser ensuite pour représenter l'objet 3D, transformant le problème de correspondance des formes entre les objets 3D en une mesure de similarité entre leurs images de coupe représentatives.

Des expériences approfondies ont montré que notre approche donne des résultats efficaces en termes de performances de recherche, en surpassant certaines méthodes bien connues dans la littérature.

Chapitre III : Classification et recherche d'objets 3D basée sur les images de coupe 2D et Les réseaux de neurones convolutifs

IV.1 Introduction

Dans ce chapitre, nous présentons notre quatrième contribution et les concepts les plus pertinents pour celle-ci.

Nous commençons par une brève présentation de l'apprentissage automatique, et plus particulièrement l'apprentissage supervisé. Après nous fournissons une vue d'ensemble sur les réseaux de neurones, de leurs origines jusqu'aux récentes percées dans le domaine, qui ont largement amélioré la qualité des résultats de nombreuses tâches de vision par ordinateur. Nous présentons ensuite les réseaux de neurones convolutifs (CNN).

Enfin, nous introduisons notre quatrième approche nommée « 2DSlicesNet », qui tire profit de la puissance des réseaux de neurones convolutifs afin d'extraire les caractéristiques des images de coupe 2D de chaque objet 3D. En effet, la méthode proposée utilise les images de coupe 2D d'objets 3D pour entraîner un réseau de neurones convolutif 3D (CNN-3D) à la classification et la recherche de ces derniers.

IV.2 Cadre et notations d'apprentissage automatique

Dans cette section, nous donnons quelques notions de bases de l'apprentissage automatique, où nous nous concentrons sur le cas de l'apprentissage supervisé.

IV.2.1 Apprentissage automatique

L'apprentissage automatique est le domaine de l'informatique qui étudie les systèmes qui apprennent à partir d'exemples sans être explicitement programmés. Ces algorithmes construisent des modèles à partir d'exemples et les utilisent pour faire des prédictions, plutôt que de suivre des règles conçues à la main.

Ces modèles peuvent être paramétriques ou non paramétriques. Un exemple simple d'approche non-paramétrique est l'algorithme des k-plus proches voisins. Dans un tel modèle, les décisions sont prises en considérant les propriétés des k plus proches voisins d'un exemple. Pour définir un voisinage, on doit définir une mesure de distance entre les exemples. Pour les images, une approche possible consiste à considérer les valeurs RVB de tous les pixels de l'image comme la représentation des caractéristiques dans un espace vectoriel, et à utiliser la distance euclidienne pour calculer la relation de voisinage entre les images. Un exemple de modèle paramétrique est la régression linéaire. Dans ce modèle, la relation entre les exemples d'entrée, qui appartiennent à un espace vectoriel, et les cibles à valeur scalaire souhaitée sont approchées par une fonction linéaire.

L'apprentissage automatique est généralement divisé en trois sous-champs :

- Apprentissage supervisé : à partir d'exemples d'entrée avec leurs étiquettes, l'objectif de l'apprentissage supervisé est d'apprendre une fonction qui associe l'entrée aux étiquettes, de sorte que des prédictions puissent être faites sur des données non vues ;
- Apprentissage non supervisé : dans le cadre de l'apprentissage non supervisé, seuls les exemples d'entrée sont donnés, et l'algorithme tente de découvrir la structure ou les modèles dans les données. Le clustering est un exemple d'algorithme d'apprentissage non supervisé ;

- Apprentissage par renforcement : l'algorithme (ou l'agent) interagit avec un environnement dynamique visant à effectuer une tâche spécifique. Il reçoit un feedback pour chaque décision qu'il prend. L'agent adapte ensuite sa stratégie afin de maximiser une fonction objectif qui mesure la qualité de la tâche.

Dans la section suivante, nous développons l'apprentissage supervisé, qui est le cadre utilisé dans notre quatrième contribution.

IV.2.2 **L'apprentissage supervisé**

Dans le cadre supervisé, nous supposons que nous avons un ensemble de données D avec N_s exemples. L'ensemble de données se compose d'observations de données $x^i \in \boldsymbol{X}$, qui représentent l'entrée qui est fournie au système, et les cibles $c^i \in \boldsymbol{C}$, qui correspondent à la sortie souhaitée du modèle. Plus formellement, nous définissons l'ensemble de données comme suit:

$$D = \{(x^i, c^i), i = 1, \dots, N_s\} \quad \text{(IV-1)}$$

Dans ce qui suit, nous nous limiterons au cas paramétrique. Soit $f^w : X \to Y$ la fonction de décision, paramétrée par w. L'espace de sortie Y peut être différent de l'espace cible, dans ce cas une fonction prédéfinie $g : Y \to C$ est utilisée pour obtenir la prédiction finale du système. Nous expliquerons plus en détail le rôle de g plus loin dans cette section, lors de l'introduction du cas de classification.

Soit $y^i = f^w(x^i)$ la sortie de la fonction de décision f^w sur l'exemple x^i. Pour mesurer l'écart entre la sortie y^i et la cible réelle c^i, on définit une fonction de perte $\ell(y^i, c^i : Y \times C \to \mathbb{R}^+)$: qui évalue la qualité de l'estimation. On cherche à rapprocher autant que possible les prédictions $g(y^i)$ des cibles c^i. Afin de quantifier l'écart entre les prédictions et les cibles pour un ensemble de données d'entraînement donné D et une fonction de décision f^w, nous définissons le risque empirique comme la moyenne des pertes sur l'ensemble d'apprentissage:

$$R_{emp}(f^w) = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell(f^w(x^i), c^i) \quad \text{(IV-2)}$$

Des valeurs élevées pour le risque empirique signifient que f^w ne se rapproche pas bien des données d'apprentissage, tandis qu'un risque de zéro indique que le modèle décrit parfaitement la relation entre les données d'entrée et celles de sortie.

Afin de modéliser correctement les dépendances entre les données et les cibles, nous recherchons les paramètres w tels que le risque empirique sur les données d'apprentissage D soit minimisé. Nous appelons la fonction que nous voulons minimiser la fonction objectif.

Minimiser la fonction objectif ne garantit pas que f^w fonctionnera bien dans les données non-vues. Par exemple, lorsque f^w a une capacité suffisante, il est possible que le modèle mémorise exactement les données d'apprentissage, qui sont souvent bruyantes, et peut-être mal performant avec des nouvelles données, car il commence à modéliser le bruit sous-jacent.

Dans une telle situation, le modèle est surajusté sur les données d'entraînement. Ce comportement n'est pas souhaitable, car il indique que le modèle est incapable de bien généraliser à de nouvelles données. Un moyen efficace pour lutter contre le sur-ajustement consiste à appliquer une régularisation $\mathcal{R}(f^w)$ sur la fonction objectif :

$$L(f^w) = \frac{1}{N_s}\sum_{i=1}^{N_s} \ell(f^w(x^i), c^i) + \lambda \mathcal{R}(f^w) \quad \text{(IV-3)}$$

Avec λ un facteur d'échelle qui définit un compromis sur l'importance de la régularisation sur l'objectif à minimiser. La régularisation contrôlera la complexité du modèle, par exemple en appliquant la norme des paramètres w pour être petit, pour laquelle un exemple courant est la décroissance du poids, donnée par $\mathcal{R}(f^w) = \|w\|_2^2$. Cela aide à empêcher le modèle de purement mémoriser les données d'apprentissage.

Une autre façon de réduire le risque de sur-ajustement consiste à appliquer le modèle pour pouvoir également effectuer une autre tâche. Pour un budget fixe de paramètres, faire en sorte que le modèle apprenne des représentations partagées pour différentes tâches peut conduire à des représentations plus informatives pour chaque tâche, car cela réduit l'effet des particularités de la distribution des données et peut aider à surmonter les limitations sur la quantité de données d'apprentissage pour chaque tâche.

L'apprentissage supervisé est subdivisé en deux branches: la classification et la régression. Cependant, dans la section suivante seulement la classification sera développée parce qu'elle est pertinente pour notre approche.

IV.2.2.1 La classification

Dans le cadre de la classification, on suppose que chaque observation de données appartient à un nombre discret de classes, et le but du modèle est de pouvoir prédire à quelle classe appartient l'observation. Un exemple typique d'un problème de classification est d'affecter un e-mail à l'une des deux classes: spam ou non-spam. Soit N_c le nombre de classes que chaque observation peut prendre, et N_s le nombre de données d'apprentissage dans la base de données d'apprentissage D. On définit $T = \{1, \dots, N_c\}$ comme l'espace d'étiquettes possibles. Soit l'espace des entrées possibles $X \subset \mathbb{R}^{N_D}$ un sous-ensemble de l'espace euclidien à N_D dimensions, avec N_D la dimensionnalité de l'espace d'entrée, et soit $Y \subset \mathbb{R}^{N_C}$ l'espace de sortie de f^w. Comme précédemment, on considère $y^i = f^w(x^i)$ comme la sortie de la fonction de décision pour l'entrée x^i. Dans notre notation, nous utilisons des indices pour définir les éléments individuels d'un vecteur. En d'autres termes, nous définissons l'indice v_i comme la i$^{\text{ème}}$ coordonnée d'un vecteur v. Une fonction de perte couramment utilisée pour l'entrainement de modèles de classification est la perte d'entropie croisée, qui est définie comme suit:

$$\ell^{EC}(y^i, t^i) = -\log(softmax(y^i)_{t^i}) \quad \text{(IV-4)}$$

Avec la fonction softmax définie comme :

$$softmax(x)_i = \frac{\exp(x_i)}{\sum_{c=1}^{N_c}\exp(x_c)} \quad \text{(IV-5)}$$

La sortie de la fonction softmax peut être considérée comme convertissant le vecteur d'entrée x de sorte qu'il peut être interprété comme une distribution de probabilité, car toutes les entrées sont positives (en raison de l'exponentielle) et totalisent 1.

Le modèle de classification attribue un score $f^w(x^i)_c$, avec $c = 1, \dots, N_c$, à chacune des classes de T pour chaque observation x^i. Pour obtenir la classe prédite à partir des scores donnés par f^w, nous définissons la fonction de conversion $g : Y \to T$ comme $g : x \to \arg\max_c x_c$. Ainsi, la classe prédite t^i est celle avec le score le plus élevé, et peut être obtenue via:

$$t^i = \arg\max_{c\in\{1,\dots,N_c} f^w(x^i)_c \quad \text{(IV-6)}$$

IV.3 Réseaux de neurones artificiels

IV.3.1 Origines des réseaux de neurones artificiels

Les réseaux de neurones artificiels sont une famille de modèles paramétriques qui ont une structure hiérarchique spécifique. La structure est une combinaison de fonctions linéaires suivie de non-linéarités, ce qui permet au modèle d'apprendre des fonctions non linéaires complexes de manière compacte. Dans cette section, nous donnons un bref aperçu des modèles mathématiques à l'origine des réseaux de neurones artificiels.

Le perceptron

Les origines des réseaux neuronaux artificiels remontent à la fin des années 1950, avec le développement du perceptron par Rosenblatt [Rosenblatt 1958]. Le terme réseau de neurones trouve son origine dans les tentatives de découvrir des représentations mathématiques du traitement de l'information dans les systèmes biologiques [Bishop 2006]. Dans ce qui suit, nous donnerons une brève définition du modèle perceptron.

Considérons une observation $x^i \in \mathbb{R}^{N_D}$. Pour faciliter la notation, nous ajoutons un 1 à x^i pour prendre en compte le biais, ce qui en fait un vecteur $\mathbb{R}^{N_D+1}$. Le perceptron mappe l'entrée à une sortie binaire $f^w(x) \in \{0,1\}$ en considérant :

$$f^w(x) = \begin{cases} 1 & si\ w\cdot x > 0 \\ 0 & sinon \end{cases}, \quad \text{(IV-7)}$$

Où $w \in \mathbb{R}^{N_D+1}$ est un vecteur de poids en valeur réelle. Notez que cela est équivalant à une fonction linéaire $x \to w\cdot x$ suivie d'une fonction d'activation non linéaire $\psi(x)$, et peut s'écrire de manière équivalente :

$$f^w(x^i) = \psi(w\cdot x^i) \quad \text{(IV-8)}$$

Où $\psi(x)$ ici est la fonction de pas de Heaviside, définie par $\psi(x)$ si $x > 0$ et 0 sinon. Une illustration du perceptron est présentée dans la figure IV-1.

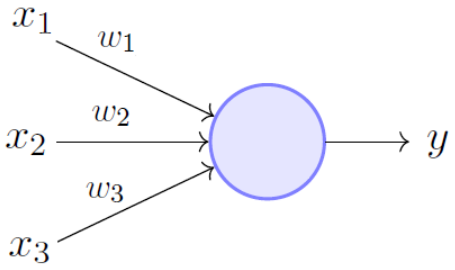


Figure IV-1: Illustration du perceptron, pour une entrée x de 3 dimensions et un nœud de sortie unique.

Une question cruciale est de savoir comment sélectionner le paramètre w pour que le perceptron défini par $f^w(x)$ puisse effectuer une tâche spécifique. Nous considérons le cadre supervisé où nous avons une base de données $D = \{(x^i, t^i)_{i\in\{1,\dots,N_s\}}\}$, avec, pour tout i, $x^i \in \mathbb{R}^{N_D+1}$ et $t^i \in \{0,1\}$.

Dans l'algorithme original du perceptron, les paramètres sont mis à jour de manière itérative en réévaluant les prédictions à chaque mise à jour des paramètres et en modifiant les paramètres qui produisent des prédictions incorrectes.

Plus formellement, soit $y^i = \psi(w \cdot x^i) \in \{0,1\}$ la sortie du modèle de perceptron. Afin de trouver l'ensemble de paramètres w qui explique le mieux la base de données D, on effectue des mises à jour stochastiques des paramètres pour chaque paire d'apprentissage (x^i, t^i) dans D comme suit:

$$w \leftarrow w + (t^i - y^i)x^i \quad \text{(IV-9)}$$

Et les mises à jour de (V-9) sont effectuées soit pour un nombre prédéterminé d'itérations, soit lorsque l'erreur d'itération $\frac{1}{N_s}\sum_{i=1}^{N_s} |t^i - y^i|$ est inférieur à un seuil prédéfini.

Fonctions d'activation différentiables et la règle Delta

Le perceptron contient une fonction d'activation discontinue (et donc non différenciable) $\psi(x)$. Si nous remplaçons la fonction d'activation par une fonction différentiable, nous pouvons dériver une règle d'apprentissage plus générique, appelée la règle delta. En utilisant la même notation que dans la section précédente, la règle delta, qui met à jour stochastiquement les poids pour chaque exemple d'apprentissage, peut être énoncée comme suit:

$$\mathrm{w} \leftarrow \mathrm{w} + \eta(\mathrm{t}^i - \mathrm{y}^i)\psi'(\mathrm{w} \cdot \mathrm{x}^i)\mathrm{x}^i \quad \text{(IV-10)}$$

Où $\psi'(x)$ est la dérivée de $\psi(x)$ par rapport à x, et η est le taux d'apprentissage, une valeur réelle qui contrôle la vitesse à laquelle les mises à jour des poids sont effectuées. Le taux d'apprentissage est un hyper-paramètre très important de l'apprentissage; des valeurs trop grandes rendent l'apprentissage instable car les paramètres oscillent autour de la solution souhaitée ou peuvent même diverger, tandis que des valeurs trop petites conduisent à un entraînement lent et sont plus susceptibles de rester coincés dans un minima local.

La règle delta peut être dérivée en minimisant la perte dans la sortie du réseau de neurone pour chaque exemple dans la base de données d'apprentissage via une descente de gradient stochastique, en utilisant une perte de distance au carré. La descente de gradient utilise le gradient de la fonction de perte par rapport aux poids du modèle w pour effectuer les mises à jour des poids dans une direction qui diminuera la perte.

Pour les fonctions d'activation linéaires $\psi(x) = x$, la règle delta peut être simplifiée comme suit:

$$W \leftarrow W + \eta(t^i - y^i)x^i \quad \text{(IV-11)}$$

Qui est très similaire à la règle de mise à jour du perceptron dans (IV-9), même si leurs dérivations sont différentes car la fonction Heaviside n'est pas différenciable.

IV.3.2 Réseaux de neurones multicouches

Malgré le succès initial du perceptron dans l'identification des chiffres dans les petites images, son pouvoir de représentation est très limité. En effet, il ne peut apprendre que des prédictions linéairement séparables dans l'espace d'entrée, ce qui est rarement le cas. Plusieurs extensions ont été proposées afin de surmonter ces limitations. En particulier, le fait d'avoir des réseaux contenant des représentations internes (également appelées couches cachées) non linéaires par rapport aux données d'entrée permet une puissance d'expression plus importante. Malheureusement, la règle delta expliquée ci-dessus ne s'applique pas dans de telles situations, car elle a été spécialement conçue pour le cas où il n'y a pas de couches cachées, donc d'autres techniques d'apprentissage sont nécessaires. Un des premiers exemples est le Neocognitron [Fukushima 1980], qui empilait plusieurs couches de fonctions linéaires suivies de non-linéarités, et utilisait une approche d'apprentissage non supervisée basée sur l'auto-similarité entre les éléments d'entrée et les poids du modèle pour effectuer l'apprentissage. Bien qu'une telle approche d'apprentissage permette d'apprendre

des réseaux avec des couches cachées, il n'y a aucune contrainte explicite qui garantit que les couches cachées apprennent une mapping appropriée. Comme nous le verrons plus loin dans cette section, il est possible d'étendre la règle delta pour qu'elle fonctionne pour de tels réseaux de neurones multicouches [Rumelhart 1985] [LeCun 1985]. Avant cela, nous introduisons d'abord une sous-catégorie des réseaux multicouches appelée réseau de neurones à propagation avant (en anglais feedforward neural network), qui est une architecture couramment utilisée dans plusieurs tâches.

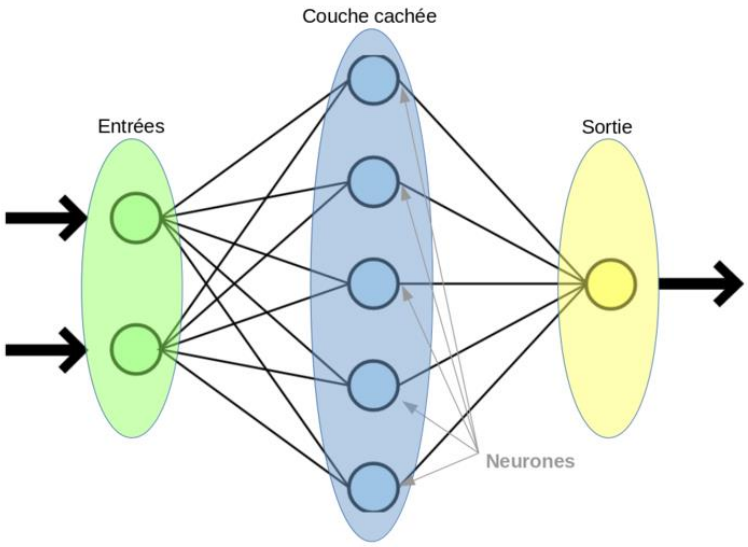


Figure IV-2 : Illustration d'un réseau de neurones à propagation avec une seule couche cachée.

Réseaux de neurones à propagation avant

Dans un réseau de neurones à propagation avant, la sortie de chaque couche est transmise en entrée à la couche suivante. Chaque couche se compose d'un certain nombre d'unités (ou neurones) qui calcule la combinaison linéaire pondérée de la couche d'entrée, suivie d'une fonction non-linéaire. La figure IV-2 illustre un réseau de neurones à propagation avant avec une couche cachée. Soit N le nombre de couches cachées. Notant par o_n la sortie de la couche n avec les poids w_n, avec comme avant le biais ajouté à des fins de notation, la procédure de rétroaction (feed-forward) peut être écrite comme suit:

$$o_n = \psi_n(w_n o_{n-1}) \qquad \text{(IV-12)}$$

Où $\psi_n(\cdot)$ est une fonction de non-linéarité sous-différentiable. Les choix courants pour la non-linéarité incluent les linéarités rectifiées comme ReLU (Rectified Linear Unit), définies par $\psi(x) = \max(0, x)$, la fonction sigmoïde $\psi(x) = (1 + e^{-x})^{-1}$ ou la tangente hyperbolique $\psi(x) = \tanh(x)$.

Rétropropagation

L'ensemble des paramètres $w = \{w_i\}_{i=1,\dots,N}$ est optimisé pour minimiser la fonction objectif $L(f^w)$ sur l'ensemble d'apprentissage D. Comme nous l'avons mentionné précédemment, la règle delta n'est pas adaptée pour les réseaux multicouches car sa formulation ne considère que le cas sans couches cachées. Pour obtenir une procédure d'optimisation pour le cas multicouche, commençons par une formulation similaire à celle utilisée pour dériver la règle delta. Nous considérons la perte $\ell(y^i, t^i)$ calculée pour chaque élément de l'ensemble d'apprentissage D, que nous voulons minimiser. Quant à la règle delta, nous utilisons la descente de gradient pour effectuer l'optimisation, qui écrit:

$$w \leftarrow w - \eta \frac{\partial \ell(f^w(x^i), t^i)}{\partial w} \quad \text{(IV-13)}$$

Où η est le taux d'apprentissage, qui contrôle la taille des étapes de mise à jour. Nous notons qu'il s'agit d'une généralisation de la règle du delta, dans le cas où nous considérons une distance au carré (squared distance) comme fonction de perte et où le gradient est calculé sur l'ensemble des exemples d'apprentissage, et pas seulement élément par élément. Afin d'effectuer une descente de gradient pour trouver les paramètres w qui minimisent la perte, nous avons besoin d'un moyen pour calculer les dérivées de la fonction de perte par rapport aux paramètres de manière efficace.

La réponse à cette question, qui est une généralisation de la règle du delta, a été donnée dans [Rumelhart et al. 1985] [LeCun 1985], et est traditionnellement appelée rétropropagation. La rétropropagation consiste à calculer, de manière récursive, les gradients d'un module en fonction des gradients des modules qui le suivent. La dérivation de la rétropropagation est obtenue en appliquant récursivement la règle de dérivation en chaîne sur la fonction de perte que nous cherchons à minimiser. Cela permet de construire des fonctions arbitrairement complexes en combinant un certain nombre de blocs plus petits, pour lesquels la dérivée est connue, et le gradient de l'ensemble de la fonction compliquée peut être facilement calculé.

L'ajout de couches cachées à un réseau augmente potentiellement la capacité du réseau à modéliser des fonctions complexes. Au début des années 1990, Hornik [Hornik 1991] a montré qu'un réseau à propagation avant avec une seule couche cachée contenant un nombre fini de neurones était capable d'approximer n'importe quelle fonction continue définie sur des sous-ensembles compacts de $\mathbb{R}^n$. Mais comme discuté dans [Bengio 2009], un résultat important en faveur des réseaux plus profonds est que les fonctions qui peuvent être représentées de manière compacte par un réseau de profondeur k peuvent nécessiter un nombre exponentiel de paramètres par rapport à la taille d'entrée à représenter par un réseau de profondeur $k - 1$.

La fonction objectif optimisée lors de la rétropropagation du réseau multicouche n'est pas convexe par rapport aux poids, du fait des multiples couches de non-linéarités qui sont présentes. La descente en gradient ne peut trouver que des minima locaux, et pour de telles fonctions non convexes, il est naturel de s'interroger sur la qualité des minima locaux trouvés. Depuis le milieu des années 80, il existait déjà des preuves que les minima locaux différents dans les réseaux multicouches fonctionnaient de manière similaire pour un certain nombre de tâches [Rumelhart et al. 1985]. Récemment, [Kawaguchi 2016] a fourni une preuve mathématique du fait que tous les minima locaux dans les réseaux de neurones profonds sont en fait un minimum global, compte tenu de certaines hypothèses raisonnables.

IV.3.3 **Réseaux de neurones convolutifs**

Les réseaux de neurones convolutifs (CNN) [LeCun et al. 1989] sont une sous-catégorie de réseaux de neurones avec des modèles de connectivité contraints dans le mapping linéaire wx. Un principe qui s'est révélé très efficace dans les images naturelles consiste à émettre l'hypothèse que la représentation des caractéristiques d'une image devrait être approximativement covariante en translation. En d'autres termes, pour une image x avec une carte de caractéristiques $f(x)$, si une translation τ est appliquée à x, alors la carte de caractéristique de l'image translatée doit correspondre approximativement à $f(x)$ translatée par τ. Cette covariance peut être imposée en contraignant le mapping linéaire wx à être une convolution. Faire en sorte que le mapping linéaire soit une convolution apporte le bénéfice supplémentaire que des images plus grandes peuvent être utilisées sans une augmentation considérable de la quantité de paramètres du modèle. Chaque unité devient responsable de la détection d'un motif particulier dans l'image, par exemple un bord orienté dans une image. Avec des convolutions, la sortie d'une couche est translatée de la même quantité que la

translation de l'entrée. Afin de rendre la sortie du réseau invariante par de petites translations et déformations, une opération de Max-Pooling a été introduite. Le Max-Pooling est une forme de sous-échantillonnage non linéaire, qui utilise l'opération maximale dans un voisinage local pour agréger la représentation des caractéristiques.

CNN et la compétition ILSVRC

Même si la plupart des fondations de CNNs ont été établies depuis [LeCun et al. 1989], il n'y avait qu'une poignée de tâches dans lesquelles les CNNs ont excellé en comparaison avec la plupart des approches traditionnelles basées sur des caractéristiques conçues à la main. Ce n'est qu'après les travaux fondateurs de [Krizhevsky et al. 2012] sur la compétition ImageNet de Reconnaissance Visuelle à Grande Échelle (ILSVRC) [Russakovsky et al. 2015] que les CNNs ont commencé à attirer l'attention générale. Cette percée est principalement due aux raisons suivantes :

GPU : les convolutions, les non-linéarités ponctuelles (pointwise non-linearities) et les multiplications matricielles, qui composent tous les éléments de base des réseaux neuronaux convolutifs traditionnels, sont naturellement susceptibles de parallélisation. Avec l'avènement du langage de programmation CUDA, dont la syntaxe ressemble à la syntaxe de C++, la mise en œuvre de programmes qui se parallélisent sur des centaines ou des milliers de cœurs est devenue beaucoup plus accessible à la communauté d'apprentissage automatique. Cette parallélisation apporte une accélération cruciale à la fois au moment de l'apprentissage et de test, permettant d'entrainer de plus gros modèles dans un délai raisonnable.

ReLU : L'un des plus gros problèmes avec les CNNs profonds avant 2012 était qu'ils étaient très difficiles à entrainer. Les réseaux profonds ont souffert du problème de la disparition du gradient (Vanishing Gradient), où les gradients dans les couches initiales sont devenus de plus en plus petits. Une façon courante pour résoudre ce problème était d'initialiser les poids du réseau via un pré-entraînement non-supervisé couche après couche. Cette solution n'était pas optimale pour plusieurs raisons : le pré-entrainement couche après couche était longue, car chaque couche devait être entrainée séparément avant l'entrainement supervisée, et l'apprentissage non supervisé des filtres de convolution était difficile à optimiser. Les unités linéaires rectifiées (Rectified Linear Units (ReLU)) [Glorot et al. 2011] ont aidé à résoudre ces difficultés. Contrairement aux fonctions de non-linéarités comme la sigmoïde ou la tangente hyperbolique, ReLU ne souffre pas de gradients disparaissant lorsque les activations deviennent plus importantes. Cela accélère considérablement le temps d'apprentissage, permettant aux réseaux plus profonds d'être entrainés dans un délai raisonnable, comme le montre la figure IV-3.

Dropout : Dropout [Srivastava et al. 2014] est une technique de régularisation qui s'est avérée cruciale pour lutter contre le sur-ajustement. Au cours de la phase d'apprentissage, il définit au hasard des sorties d'une couche spécifique à 0, ce qui signifie que ces éléments mis à zéro ne participeront pas à la rétropropagation. Cela empêche les co-adaptations complexes entre les caractéristiques, car chaque neurone ne pourra pas s'appuyer sur la sortie d'un autre neurone. Au moment du test, tous les neurones restent actifs, mais la sortie est multipliée par le pourcentage des neurones désactivés pendant la phase d'entrainement pour compenser que plus de neurones sont actifs en phase de test.

Grand modèle : L'architecture proposée par [Krizhevsky et al. 2012] contient 60 millions de paramètres et 8 couches de non-linéarités. La figure IV-4 illustre cet architecture. L'utilisation d'un réseau d'une telle taille était sans précédent, mais s'est avérée nécessaire pour modéliser une tâche aussi complexe que celle requise pour classer entre les 1000 classes d'ImageNet. En effet, [Krizhevsky et al. 2012] mentionne qu'en supprimant une des couches intermédiaires, il en résulte une perte d'environ 2% pour les performances du réseau, indiquant que la profondeur était fondamentale pour leurs bons résultats.

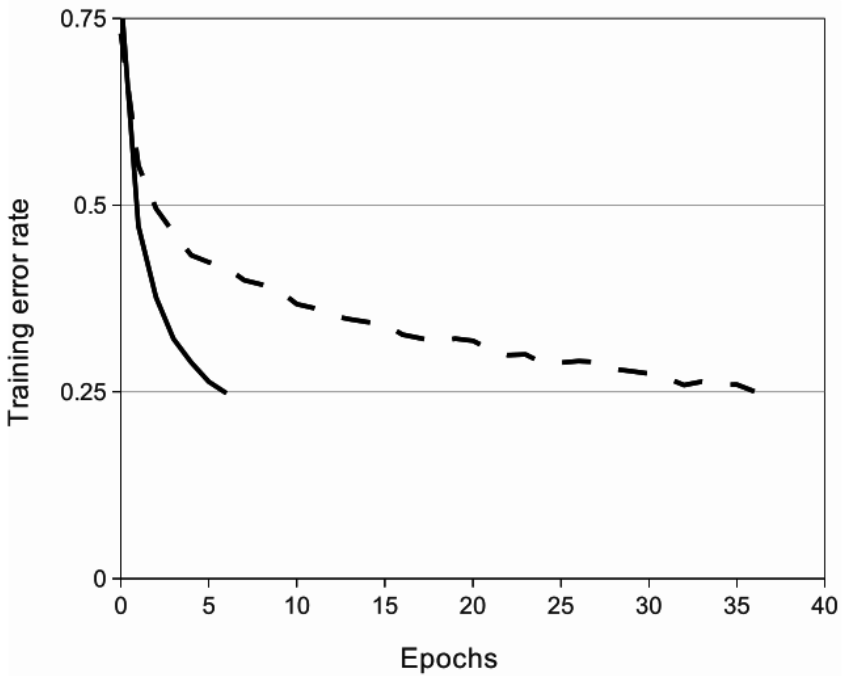


Figure IV-3 : Un réseau de neurones convolutionnels à quatre couches avec ReLU (ligne continue) atteint un taux d'erreur d'apprentissage de 25% sur CIFAR-10 six fois plus rapide qu'un réseau équivalent avec des neurones tanh (ligne pointillée). Figure de [Krizhevsky et al. et al. 2012]

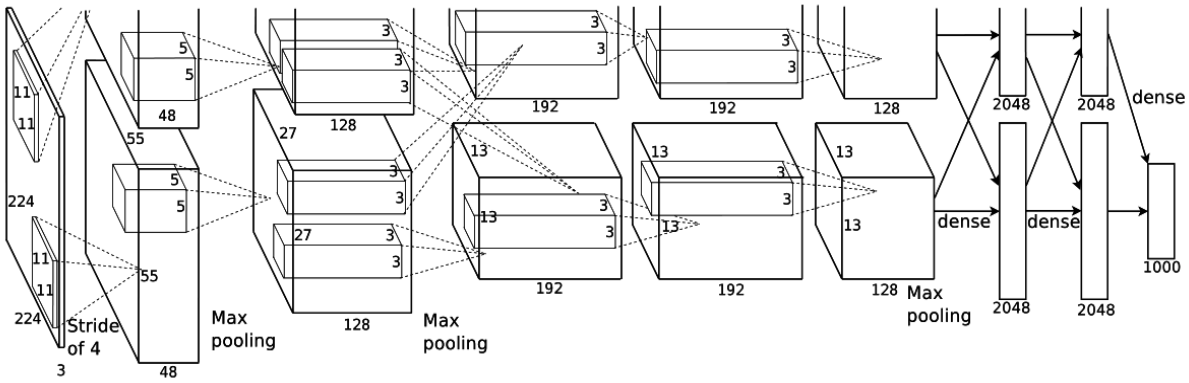


Figure IV-4: Une illustration de l'architecture de réseau présentée dans [Krizhevsky et al. 2012] également appelée AlexNet dans la littérature. Il se compose de 5 couches de convolution, suivies de 3 couches entièrement connectées. Figure prise de [Krizhevsky et a. 2012]

Grands base de données + augmentation de la base de données : l'entrainement d'un modèle aussi énorme de manière supervisée n'est possible que s'il existe suffisamment de données d'apprentissage pour permettre au modèle de se généraliser et de ne pas être surajusté. Pour les très grands réseaux, même les 1,2 million d'images d'ImageNet peuvent ne pas être suffisantes, donc plusieurs techniques pour augmenter artificiellement la taille de la base de données ont été utilisées, telles que les échelles et les recadrages aléatoires, le retournement horizontal des images et les petites déformations des couleurs.

La plupart de ces contributions ont déjà été présentées individuellement avant [Krizhevsky et a. 2012], mais il a fallu des résultats révolutionnaires dans une compétition

difficile comme ILSVRC pour attirer l'attention de la communauté de la vision par ordinateur sur les CNN. Depuis ce jour, plusieurs améliorations majeures dans les réseaux de neurones convolutifs ont été apportées, améliorant les performances de nombreuses tâches [Eigen et al. 2014] [Gidaris et al. 2015] [Yu et al. 2016] [Xie et al. 2015].

Visualiser les représentations internes

Qu'est-ce qui fait que les CNNs profonds fonctionnent si bien ? Étant donné les résultats impressionnants obtenus par les CNNs profonds pour les tâches de vision, il est naturel de se demander ce qui est appris en interne par le réseau. En inspectant les cartes de caractéristiques internes du réseau, [Zeiler et al. 2014] ont montré que les couches antérieures du réseau sont responsables de la détection des bords et des couleurs orientés, tandis que les couches ultérieures apprennent des motifs plus complexes, tels que des grilles, des cercles ou même des visages.

Entrainant des réseaux plus vastes et plus profonds

Simonyan et al. [Simonyan et al. 2014] a proposé une architecture CNN profonde qui remplace les grands filtres de convolution présents dans l'architecture originale de [Krizhevsky et al. 2012] par une série de filtres 3 × 3, avec des non-linéarités ReLU entre les deux. Par exemple, en remplaçant un filtre 5 × 5 par deux filtres 3 × 3, le champ récepteur effectif reste le même, ce qui signifie que la même région de l'image est couverte par les convolutions. Cette factorisation augmente le nombre de non-linéarités présentes dans le réseau et diminue en outre la quantité de paramètres. Cela s'est révélé très bénéfique et améliore considérablement la puissance de représentation du modèle, conduisant à une amélioration importante en termes de précision de la classification.

Peu de temps après, Ioffe et Szegedy ont proposé la normalisation par lots [Ioffe et al. 2015.], une technique simple qui supprime le décalage de la covariable des caractéristiques en normalisant les cartes de caractéristiques sur chaque mini-lot. Cela présente également l'avantage positif que les sorties de chaque couche sont dans la même plage. La normalisation par lots permet un entrainement plus rapide des réseaux et élimine le problème de gradient disparaissant.

Compte tenu de ces améliorations, il était naturel de se demander si nous n'avions besoin que de machines plus puissantes et de modèles plus grands pour obtenir de meilleurs résultats. Dans [He et al. 2016], He et al. ont montré qu'une simple augmentation de la profondeur dans les CNNs à action directe n'améliore pas nécessairement la précision de la classification, mais en apprenant les fonctions résiduelles $h(x) = x + f(x)$, il est possible d'entraîner des réseaux beaucoup plus profonds avec une précision croissante.

Bien que l'augmentation de la profondeur se soit avérée très bénéfique pour l'apprentissage de fonctions plus complexes, elle augmente également dans une large mesure la quantité de mémoire requise par le CNN. Dans la section suivante, nous présenterons notre méthode de classification et de recherche d'objet 3D, qui est basée sur un réseau de neurones convolutifs 3D peu profond en extrayant les caractéristiques les plus distinctives des images de coupe 2D de chaque objet 3D.

IV.4 Quatrième approche proposée pour la classification et la recherche d'objets 3D

IV.4.1 Approche proposée

En se basant sur la convolution 3D, une diversité de structures CNN peut être développée. Nous présentons donc une architecture CNN 3D basée sur des images de coupe 2D que nous avons extrait pour la classification et la recherche d'objets 3D. Dans notre approche, présentée dans la figure IV-5, nous commençons par une étape de normalisation, afin de garantir que les modèles similaires auront la même orientation, la même position et la même échelle. Ensuite, les images de coupe 2D correspondant au premier axe principal de

chaque modèle 3D sont extraites, redimensionnées, empilées et ensuite utilisées comme données d'entrée pour notre CNN 3D.

Afin de comparer les objets 3D, nous devons obtenir leur descripteur. La dernière couche entièrement connectée de notre CNN 3D est utilisée pour obtenir les caractéristiques des objets 3D. Enfin, la distance euclidienne est utilisée pour calculer la similarité entre les descripteurs des modèles 3D. Les détails seront présentés dans les sous-sections suivantes.

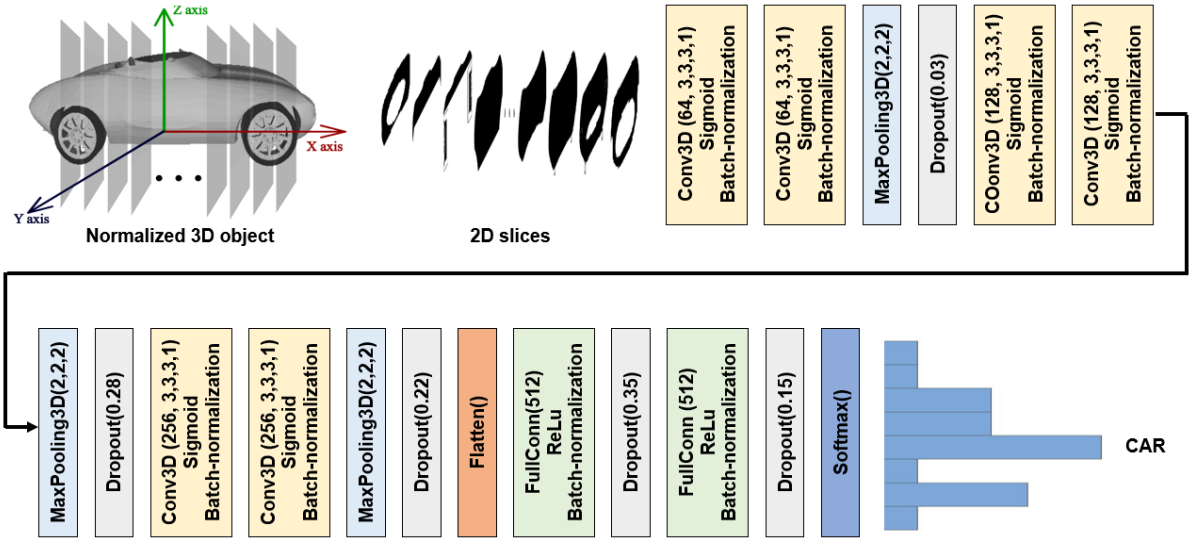


Figure IV-5: L'aperçu de 2DSlicesNet. Les images de coupes 2D correspondant au premier axe principal de l'objet 3D normalisé sont extraites, redimensionnées, empilées puis utilisées comme donnée d'entrée pour notre CNN 3D.

❖ **Prétraitement des objets 3D**

Les modèles 3D obtenus par divers systèmes de numérisation et de modélisation ont des cadres de coordonnées différents, car les formes des modèles 3D sont toujours formées dans un système de coordonnées spécifique. Pour plusieurs applications, telles que la recherche, la classification, la visualisation et la modélisation basées sur le contenu, la normalisation des cadres de coordonnées doit généralement être effectuée dans un premier temps. Par exemple, de nombreuses approches de classification et de recherche d'objets 3D nécessitent une étape de normalisation, a priori, pour un objet requête donné et les objets 3D de la base de données, de sorte que tous les objets 3D normalisés sont alignés dans un cadre de coordonnées habituel avant d'être traités.

En fait, l'étape de normalisation comprend la normalisation de l'échelle, de la translation et de la rotation. Dans notre approche, l'étape de normalisation consiste uniquement à mettre à l'échelle et à translater chaque modèle 3D, étant donné que les objets 3D de ModelNet10 et ModelNet40 sont alignés manuellement par l'équipe de Princeton. Pour effectuer la normalisation de l'échelle, la distance moyenne de la surface d'un objet 3D par rapport à son centroïde est égale à 1. La normalisation de la translation est réalisée en calculant le centre de masse de l'objet 3D et en le faisant coïncider avec l'origine.

❖ **Extraction des images de coupe 2D**

Pour obtenir les images de coupe des objets 3D, nous prenons l'intersection d'objet 3D avec 64 plans équidistants et orthogonaux à son premier axe principal. Nous invitons le lecteur à voir notre première approche proposée dans la section III.3.1, qui fournit plus de détails sur la méthode utilisée afin d'extraire les images de coupe 2D.

Toutes les images de coupe 2D extraites sont d'abord redimensionnées en 64 × 64 et ensuite empilées dans une matrice. À la fin de ce processus, chaque objet 3D est représenté par un ensemble de 64 images de coupe 2D, que nous utiliserons comme données d'entrée de notre CNN 3D. Un exemple d'objet 3D est illustré sur la figure IV-6, avec ses images de coupe 2D correspondant à son premier axe principal.

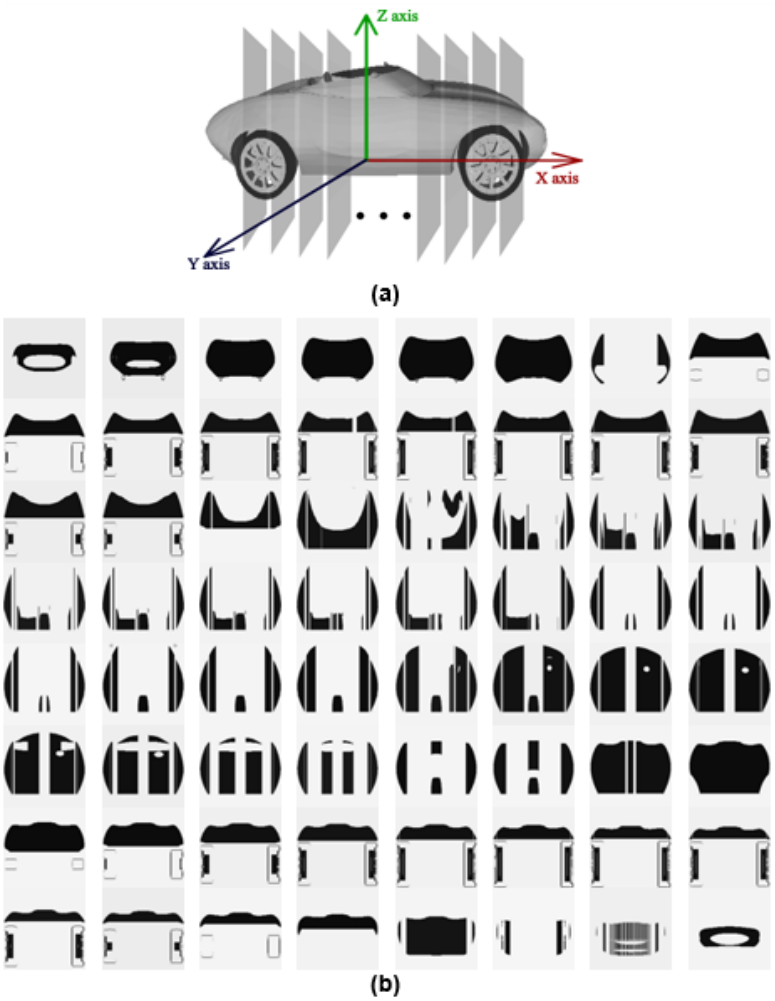


Figure IV-6: Exemple d'un objet 3D (a) avec ses images de coupe 2D (b) correspondant à son premier axe principal en utilisant 2DSlicesNet.

❖ **Augmentation des données**

L'augmentation des données comprend un ensemble de pratiques qui permettent d'augmenter de manière significative la variété des données existantes pour les approches d'apprentissage, sans pour autant collecter de nouvelles données. En fait, les techniques d'augmentation des données nous permettent d'augmenter la taille et la qualité des ensembles de données d'apprentissage et d'offrir une certaine variété, car par la suite, de meilleures approches d'apprentissage approfondi peuvent être construites en les utilisant, ce qui augmente la performance de généralisation et réduit le sur-ajustement.

La rotation des données est l'une des techniques d'augmentation des données les plus utilisées dans les architectures CNN. Ainsi, nous avons utilisé cette technique afin de mieux entraîner notre réseau. En fait, les objets 3D dans la base de données sont orientés vers le haut. Cependant, ils ne sont pas orientés régulièrement le long de l'axe ; ils pourraient être tournés de manière aléatoire le long de la direction verticale. Comme notre approche repose sur des images de coupe 2D d'objets 3D, les données d'entrée peuvent

bénéficier d'une augmentation de la rotation sur l'axe vertical pour l'apprentissage et, éventuellement, pour les tests. Pendant l'entrainement, nous faisons pivoter les images de coupe 2D de chaque objet 3D de 11,25 degrés et nous considérons chaque version tournée comme une instance d'apprentissage différente. Au moment des tests, les images de coupe 2D des objets 3D et toutes leurs versions pivotées sont introduites dans notre architecture CNN 3D en un seul lot, et la moyenne des cartes de caractéristiques de la dernière couche entièrement connectée est calculée.

❖ **Architecture de 2DSlicesNet**

L'architecture de 2DSlicesNet est basée sur un schéma ordinaire, à savoir une couche d'entrée, six couches de convolution, trois couches de regroupement, deux couches entièrement connectées et une couche de sortie softmax.

Pour toutes les couches convolutionnelles, nous utilisons un filtre convolutionnel de taille 3×3×3 suivi d'une fonction de non-linéarité sigmoïde, les cartes de caractéristiques correspondantes sont respectivement 64, 64, 128, 128, 256 et 256. Pour accélérer l'entrainement du réseau, la normalisation par lots [Ioffe et al. 2015] est utilisée après chaque couche convolutive.

Après chaque deux couches convolutives, nous intégrons une couche de regroupement maximale (Max-Pooling) de 2×2×2 pour rendre les cartes de caractéristiques insensibles aux petites translations et pour réduire leurs résolution. Afin d'améliorer la capacité de généralisation et d'éviter les surajustements, nous ajoutons des couches de Dropout après chaque couche de regroupement. Le choix d'utiliser de petits champs de réception (3×3×3), en utilisant le pas 1, au lieu d'utiliser des champs de réception relativement larges dans les premières couches convolutionnelles, est soutenu par l'idée qu'une pile de deux couches convolutionnelles 3×3×3, sans couche de regroupement entre elles, a un champ de réception réel de 5×5×5. En fait, lorsque nous utilisons, par exemple, un empilement de deux couches convolutionnelles 3×3×3, plutôt qu'une seule couche convolutionnelle 5×5×5, nous réduisons d'abord le nombre de paramètres, puis nous intégrons deux couches de rectification non linéaire plutôt qu'une seule, ce qui augmente la discrimination de la fonction de décision.

La sortie de notre architecture comprend deux couches entièrement connectées, chacune ayant 512 unités de sortie. Les deux couches entièrement connectées sont suivies par la fonction ReLu et la couche Dropout. Enfin, une couche Softmax est utilisée à la fin de notre système pour permettre l'optimisation des paramètres en minimisant les erreurs de classification des modèles 3D. La classe ayant la plus grande probabilité est traitée comme la classe prédite pour le modèle 3D. Les détails de notre architecture sont présentés sur la figure IV-5.

❖ **Extraction de caractéristiques pour la tâche de recherche**

Le descripteur utilisé pour la tâche de recherche est la sortie de la dernière couche entièrement connectée après la fonction d'activation ReLU de notre 2DSlicesNet, qui est un vecteur à 512 dimensions. Chaque descripteur est comparé aux autres descripteurs à l'aide de la distance euclidienne. Pour deux objets 3D *Q* et *O*, leur descripteur sont extraits de 2DSlicesNet V_q et V_o, respectivement. La formule de mesure de la distance euclidienne est déterminée comme suit :

$$d(Q,O) = \|V_q - V_o\|_2 \qquad \text{(IV-14)}$$

IV.4.2 **Résultats expérimentaux**

Dans cette section, nous présentons d'abord les bases de données utilisées pour évaluer notre méthode. Puis, un bref aperçu sur les détails de la mise en œuvre est fourni. Ensuite, nous présentons les expériences menées concernant la classification et la recherche

d'objets 3D. Nous discutons également les résultats de ces expériences et les comparons avec des approches bien connues. Il convient de mentionner que les scores des approches comparées sont ceux déclarés par les auteurs dans les articles respectifs.

❖ **Bases de données**

Dans nos expériences, nous avons testé notre approche sur les deux versions de l'ensemble de données ModelNet de Princeton [Wu et al. 2015], ModelNet10 et ModelNet40, qui sont deux sous-ensembles couramment utilisés dans ModelNet. Le ModelNet10 contient 4 899 objets 3D provenant de 10 classes, tandis que le ModelNet40 comprend 12 311 objets 3D issus de 40 classes. Tous les objets 3D de ModelNet10 et ModelNet40 sont nettoyés et alignés manuellement par l'équipe de Princeton. Dans nos tests, les bases de données d'apprentissage et de test sont divisées selon le même paramétrage, décrit dans [Wu et al. 2015].

❖ **Détails de la mise en œuvre**

Notre approche « 2DSlicesNet » a été évalué sur un CPU Intel Xeon E5-2670, avec 64 Go de RAM et un GPU NVIDIA QUADRO M6000 avec 12 Go de RAM. L'approche d'extraction des images de coupe 2D a été mise en œuvre en C++, tandis que l'architecture du réseau a été développée en Python 3.7.7 en utilisant Keras 2.3.1 et Tensorflow 2.1.0 via le jeu d'instructions CUDA sur le GPU.

Pour l'optimisation des hyperparamètres, nous avons utilisé Hyperas, qui choisit les paramètres les plus performants parmi les options proposées. En ce qui concerne le paramètre de l'algorithme, Hyperas a opté pour la descente de gradient stochastique avec 0,9 en inertie (Momentum) [Sutskever et al. 2013], au lieu d'Adam et RME. Pour le taux d'apprentissage, elle a choisi 0,001 à la place de 0,01 et 0,1. Pour la fonction d'activation, Hyperas a sélectionné la fonction Sigmoïde pour les six couches convolutives. Cependant, elle a opté pour la fonction ReLu pour les deux couches entièrement connectées. Nous avons également utilisé Hyperas pour sélectionner les meilleures valeurs pour les couches de Dropout dans l'intervalle [0, 0.9]. Enfin, la taille du lot a été fixée à 32. Il est à noter que nous avons entraîné notre réseau pour 200 époques.

❖ **Classification des objets 3D**

Nous avons d'abord évalué notre méthode « 2DSlicesNet » en tâche de classification sur le sous-ensemble de test de ModelNet10 et ModelNet40. Nous avons comparé notre méthode avec des approches récentes, y compris des méthodes qui n'utilisent pas l'apprentissage automatique ; les descripteurs proposés par [Kazhdan et al. 2003] (Spherical Harmonics (SPH)) et [Chen et al. 2003] (Light Field Descriptor (LFD)), et ceux qui utilisent l'apprentissage automatique, à savoir RadialNet [Ng et al. 2020], LonchaNet [Gomez-Donoso et al. 2017], Primitive-GAN [Khan et al. 2019], VSL [Liu et al. 2018], BinVoxNetPlus [Ma et al. 2018], DeepSets [Zaheer et al. 2017], 3D-DescriptorNet [Xie et al. 2018], l'approche proposée par Soltani et al. [Arsalan Soltani, et al. 2017], l'approche presentée par Zanuttigh et Minto [Zanuttigh et al. 2017], ECC [Simonovsky et al. 2017], FPNN [Li et al. 2016], PointNet [Qi et al. 2017a], PointNet [Garcia et al. 2016], LightNet [Zhi et al. 2018], l'approche proposée par Xu et Todorovic [Xu et al. 2016], Geometry Image [Sinha et al. 2016], 3D-GAN [Wu et al. 2016], Pairwise [Johns et al. 2016], MVCNN [Su et al. 2015], GIFT [Bai et al. 2016], VoxNet [Maturana et al. 2015], DeepPano [Shi et al. 2015] et 3DShapeNets [Wu et al. 2015]. Le tableau IV-1 récapitule la précision de la classification des approches mentionnées ci-dessus.

Notre approche surpasse toutes les autres méthodes comparées dans la base de données ModelNet40. Toutefois, dans la base de données ModelNet10, elle est uniquement surpassée par LonchaNet [Gomez-Donoso et al. 2017] par une faible marge (94,05 % contre 94,37 %). En effet, LonchaNet est basé uniquement sur trois images de coupe 2D correspondant aux axes principaux de l'objet 3D, ce qui peut l'amener à représenter certains objets 3D de la même catégorie par des images de coupe 2D totalement dissemblables, ou à

l'inverse, des images de coupe 2D similaires pour des objets 3D de classes différentes. Les auteurs de LonchaNet n'ont pas testé leur méthode sur une grande base de données comportant plus de catégories, car nous pensons que, si on augmente le nombre d'objets et de classes, on risque de tomber dans le problème mentionné ci-dessus.

Tableau IV-1: Résultats de la précision de classification atteints par notre approche (2DSlicesNet) et quelques approches de la littérature sur ModelNet10 et ModelNet40. Le signe "-" signifie qu'aucune information n'est présentée pour la méthode correspondante dans le papier concerné.

Approches	**ModelNet10**	**ModelNet40**
2DSlicesNet (Notre approche)	94.05%	**91.05%**
LonchaNet [Gomez-Donoso et al. 2017]	**94.37%**	-
RadialNet [Ng et al. 2020]	89.53%	86.42%
Primitive-GAN[Khan et al. 2019]	92.20%	86.40%
VSL [Liu et al. 2018]	91.00%	84.50%
BinVoxNetPlus [Ma et al. 2018]	92.32%	85.47%
DeepSets [Zaheer et al. 2017]	-	90.30%
3D-DescriptorNet [Xie et al. 2018]	92.40%	-
Soltani et al. [Arsalan Soltani, et al. 2017]	-	82.10%
Zanuttigh et Minto [Zanuttigh et al. 2017]	91.50%	87.80%
ECC [Simonovsky et al. 2017]	90.00%	83.20%
FPNN [Li et al. 2016]	-	88.40%
PointNet [Qi et al. 2017a]	-	89.20%
PointNet [Garcia et al. 2016]	77.60%	-
LightNet [Zhi et al. 2018]	93.94%	88.93%
Xu et Todorovic [Xu et al. 2016]	88.00%	81.26%
Geometry Image [Sinha et al. 2016]	88.40%	83.90%
3D-GAN [Wu et al. 2016]	91.00%	83.30%
Pairwise [Johns et al. 2016]	92.80%	90.70%
MVCNN [Su et al. 2015]	-	90.10%
GIFT [Bai et al. 2016]	92.35%	83.10%
VoxNet [Maturana et al. 2015]	92.00%	83.00%
DeepPano [Shi et al. 2015]	85.45%	77.63%
3DShapeNets [Wu et al. 2015]	83.50%	77.00%
SPH [Kazhdan et al. 2003]	79.79%	68.23%
LFD [Chen et al. 2003]	79.87%	75.47%

La figure IV-7 montre la matrice de confusion pour la partie test de ModelNet10. Comme nous pouvons l'observer, l'approche classifie précisément les objets 3D, à l'exception de quelques-uns, qui semblent similaires d'un point de vue visuel. Comme nous pouvons le voir dans la figure IV-7, la majorité des erreurs de classification proviennent des paires de classes similaires telles que "Dresser" contre "Night Stand" et "Desk" contre "Table".

La figure IV-8 présente les courbes de rappel-précision de la classification pour les différentes classes de la partie test du ModelNet10. Les courbes montrent la puissance et la stabilité de notre approche en termes de classification des objets 3D. En fait, toutes les classes atteignent une précision moyenne de 1,00 sauf " Desk " (0,96), " Table " (0,97), " Dresser " (0,98) et " Night Stand " (0,97), ce qui confirme les résultats présentés dans la figure IV-7.

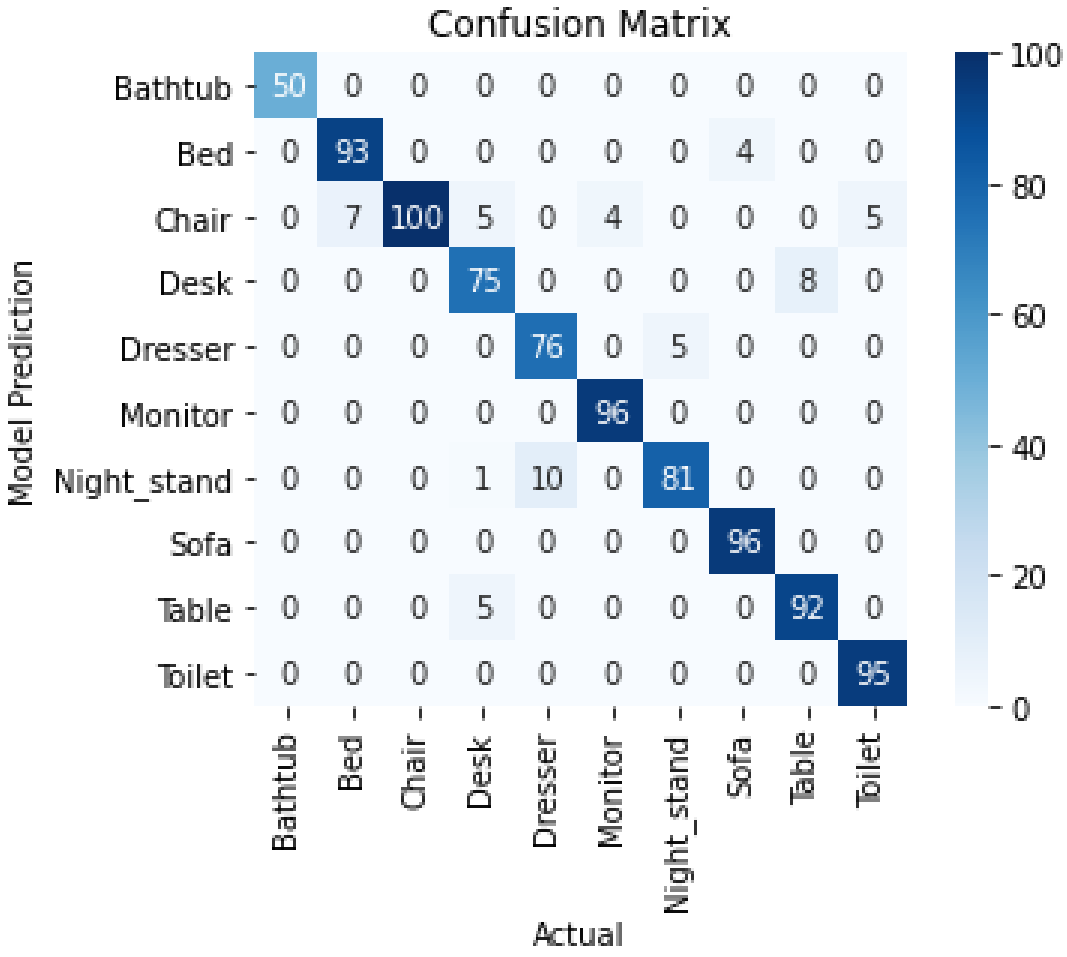


Figure IV-7: Matrice de confusion des résultats de classification obtenus par notre approche « 2DSlicesNet » en utilisant la base de données ModelNet10.

Recherche d'objets 3D

Afin d'examiner la performance de notre approche en ce qui concerne la recherche d'objets 3D, nous avons procédé à une évaluation supplémentaire. L'efficacité de notre système 2DSlicesNet pour la recherche d'objets 3D a été mesurée sur les bases de données ModelNet10 et ModelNet40, en utilisant les courbes de précision/rappel et le score moyen de précision (mAP), par rapport à certaines approches bien connues. Plus précisément, Spherical Harmonics (SPH) [Kazhdan et al. 2003], Light Field (LFD) [Chen et al. 2003], Geometry Image [Sinha et al. 2016], PANORAMA [Papadakis et al. 2010] et 3DShapeNets [Wu et al. 2015], les résultats des approches mentionnées ci-dessus sont ceux énoncés par les chercheurs dans leurs propres papiers.

Le tableau IV-2 présente les résultats des expériences relatives à la recherche d'objets 3D en termes de mAP où notre approche « 2DSlicesNet » surpasse les autres méthodes dans les deux bases de données. Comme nous pouvons le constater, l'approche proposée se place en première position, suivie par « Geometry Image » [Sinha et al. 2016].

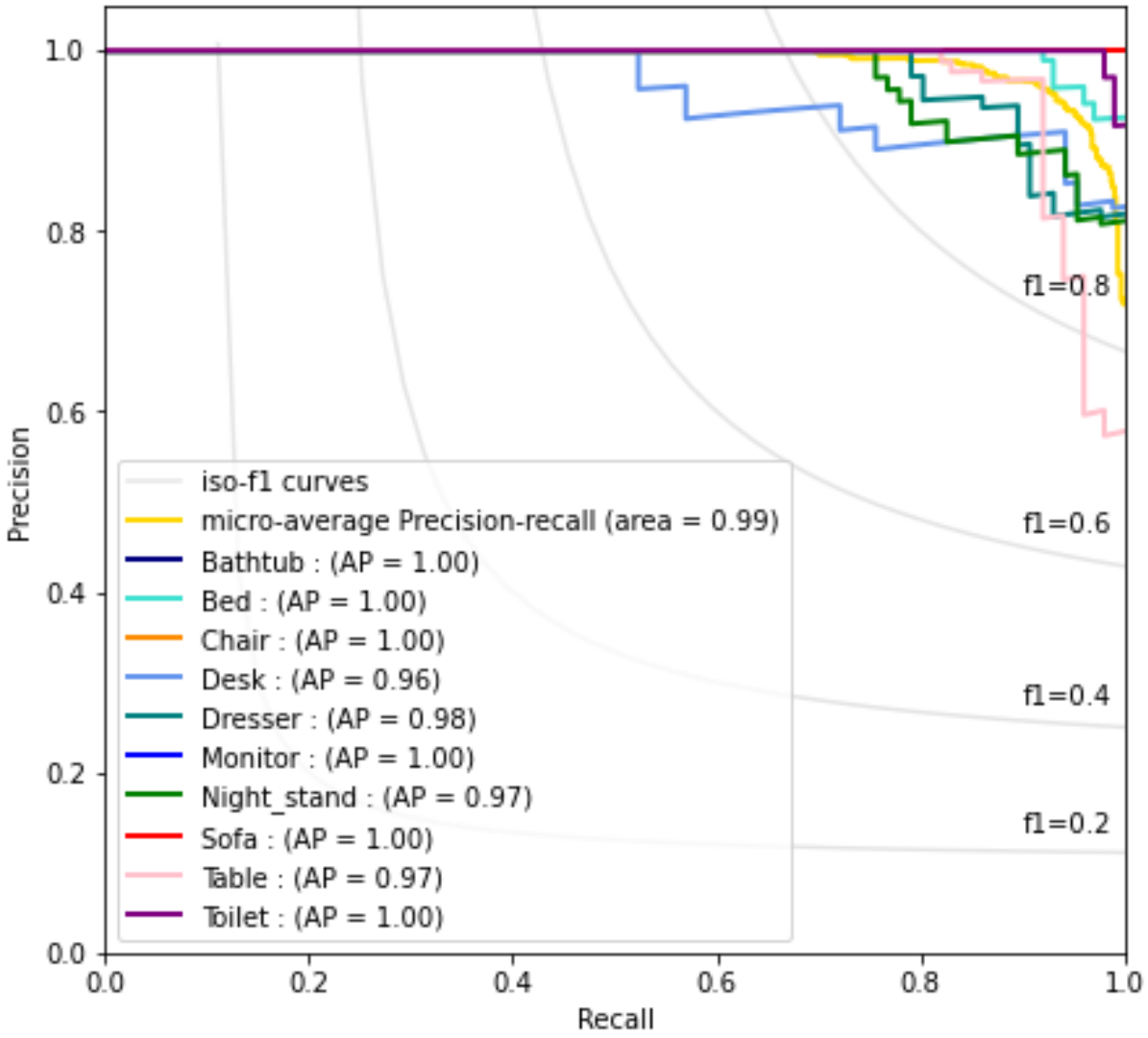


Figure IV-8 : Courbes de précision/rappel de la classification pour chaque classe de la base de données ModelNet10. 'AP' signifie la micro moyenne de précision correspondant à chaque classe de ModelNet10.

La figure IV-9 et la figure IV-10 représentent les courbes de rappel-précision de notre approche, Spherical Harmonics (SPH) [Kazhdan et al. 2003], Light Field (LFD) [Chen et al. 2003], PANORAMA [Papadakis et al. 2010] et 3DShapeNets [Wu et al. 2015] sur ModelNet10 et ModelNet40, respectivement. Comme nous pouvons l'observer, les courbes de rappel-précision montrent l'efficacité de notre méthode dans la tâche de recherche d'objets 3D en surpassant toutes les autres méthodes, et confirment les scores mAP présentés dans le tableau IV-2. En outre, par rapport aux autres approches, la courbe entière de notre méthode diminue lentement, lorsque le rappel augmente dans les deux bases de données, ce qui prouve que l'approche est plus stable. En particulier sur le ModelNet40, l'approche proposée conserve à peu près la même courbe, tandis que les courbes des autres approches régressent de manière significative, ce qui signifie que notre approche est performante bien que nous augmentions le nombre de catégories et d'objets.

Tableau IV-2: Comparaison des résultats de recherche sur les bases de données ModelNet10 et ModelNet40 mesurés en termes de score moyen de précision (mAP)

Approches	**ModelNet10**	**ModelNet40**
2DSlicesNet (Notre approche)	**76.36%**	**72.08%**
Geometry Image [Sinha et al. 2016]	74.9%	51.3%
3DShapeNets [Wu et al. 2015]	68.3%	49.2%
PANORAMA [Papadakis et al. 2010]	60.32%	46.13%
SPH [Kazhdan et al. 2003]	44.05%	33.26%
LFD [Chen et al. 2003]	49.82%	40.91%

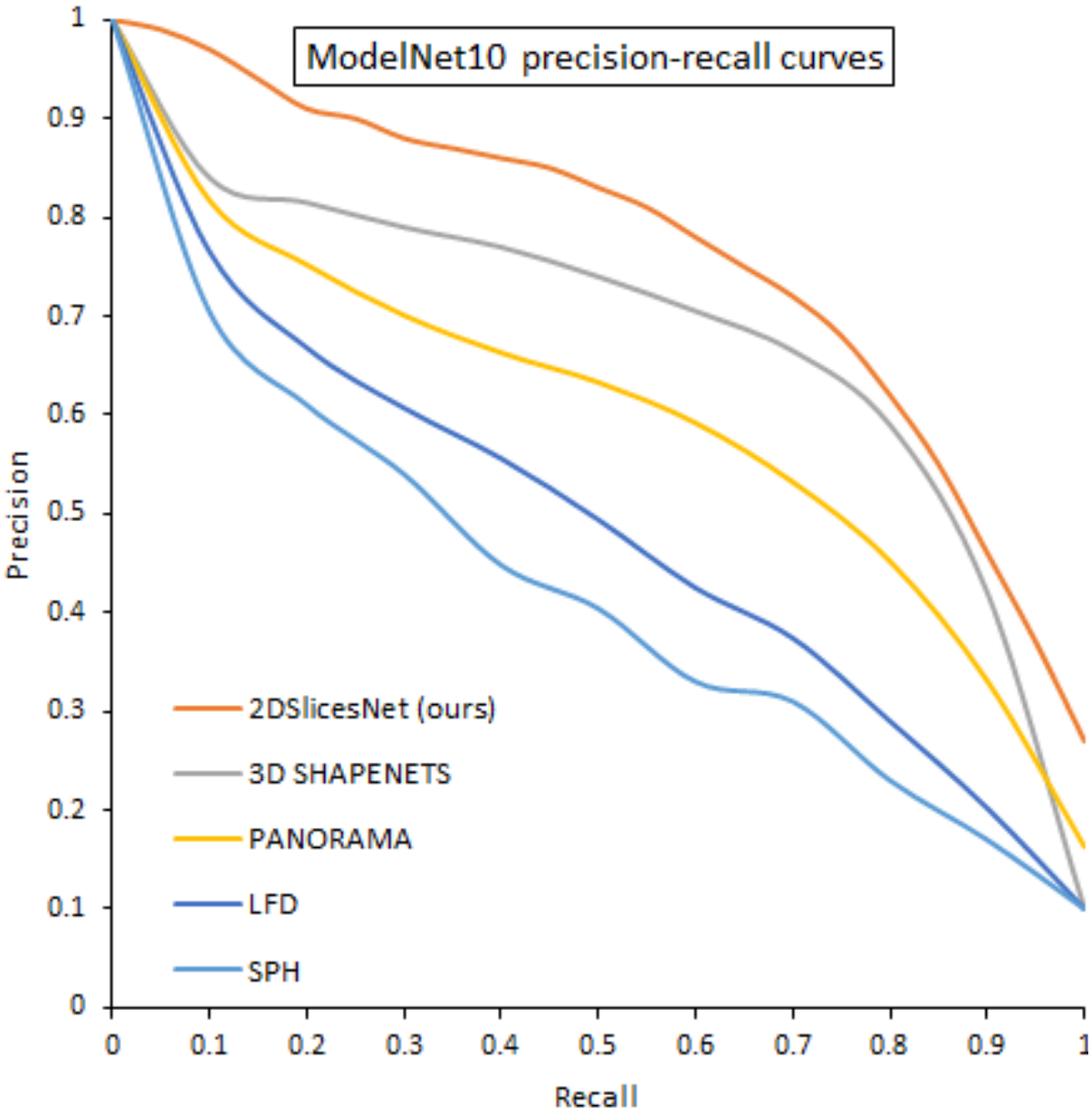


Figure IV-9: Courbes de précision/rappel pour ModelNet10. Les courbes montrent l'approche proposée « 2DSlicesNet » comparée à quatre approches de recherche d'objets 3D bien connues.

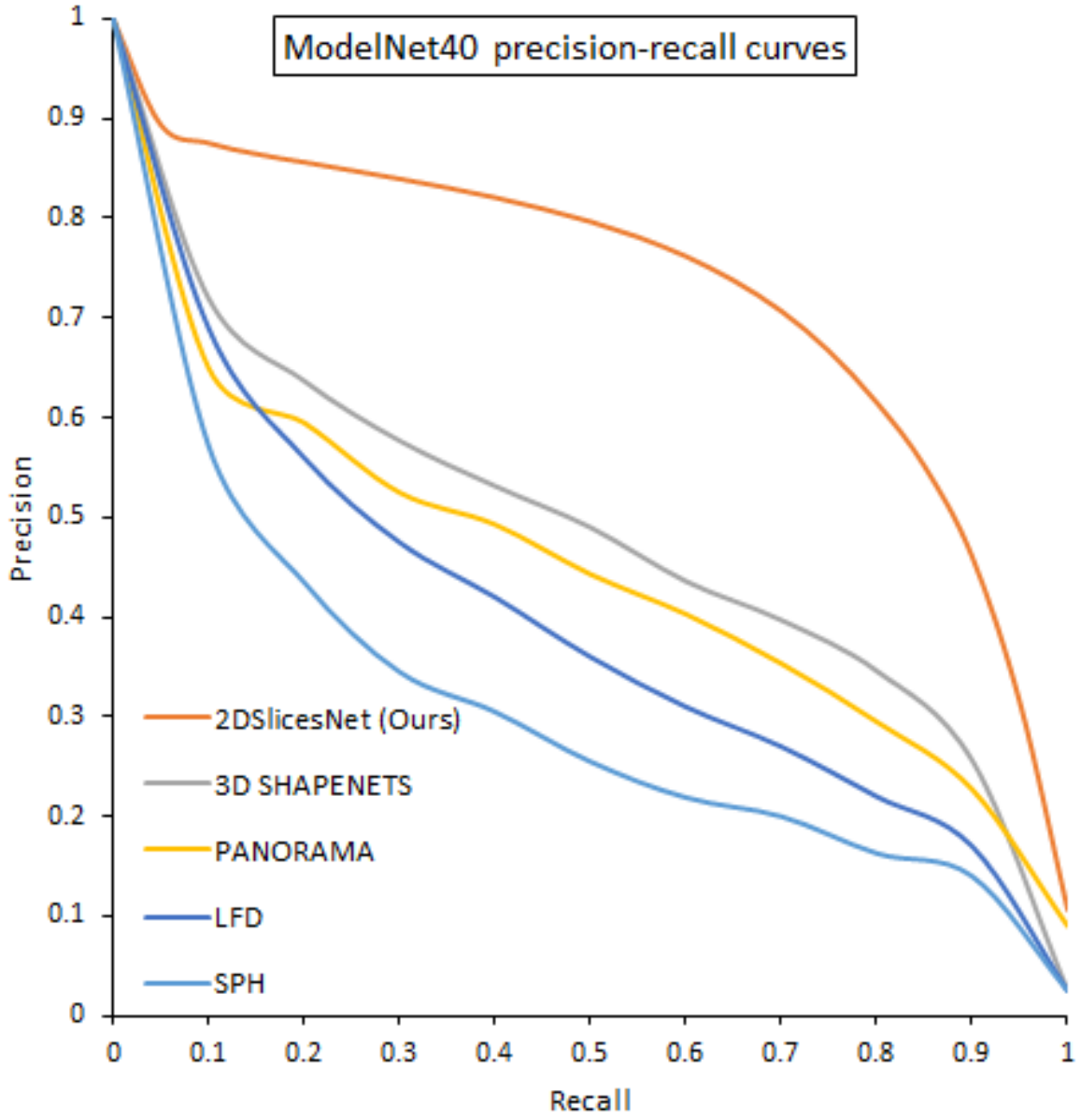


Figure IV-10: Courbes de précision/rappel pour ModelNet40. Les courbes montrent l'approche proposée « 2DSlicesNet » comparée à quatre approches de recherche d'objets 3D bien connues.

La figure IV-11 montre l'image de niveau (Tier image) pour les objets 3D de la base de données ModelNet10. Le blanc, le jaune et l'orange visualisent le plus proche voisin, le premier niveau et le deuxième niveau de correspondance, respectivement. Une méthode de recherche puissante doit comporter un ensemble de pixels blanc-jaune dans les blocs de la taille d'une classe le long de la diagonale. Il est à noter que notre approche comporte des pixels plus brillants, en particulier des pixels jaunes, dans les blocs de la diagonale des classes, ce qui montre que les objets 3D appartiennent à la même catégorie indiquent une plus grande similarité. Nous remarquons également d'autres blocs orange qui se situent en-dehors de la diagonale des classes, en particulier les classes " Desk " et " Table ". En fait, les catégories "Desk" et "Table" ont généralement une structure similaire et certains « Desk » ont des caractéristiques visuelles mineures qui les différencient de « Table », ce qui rend le système confus.

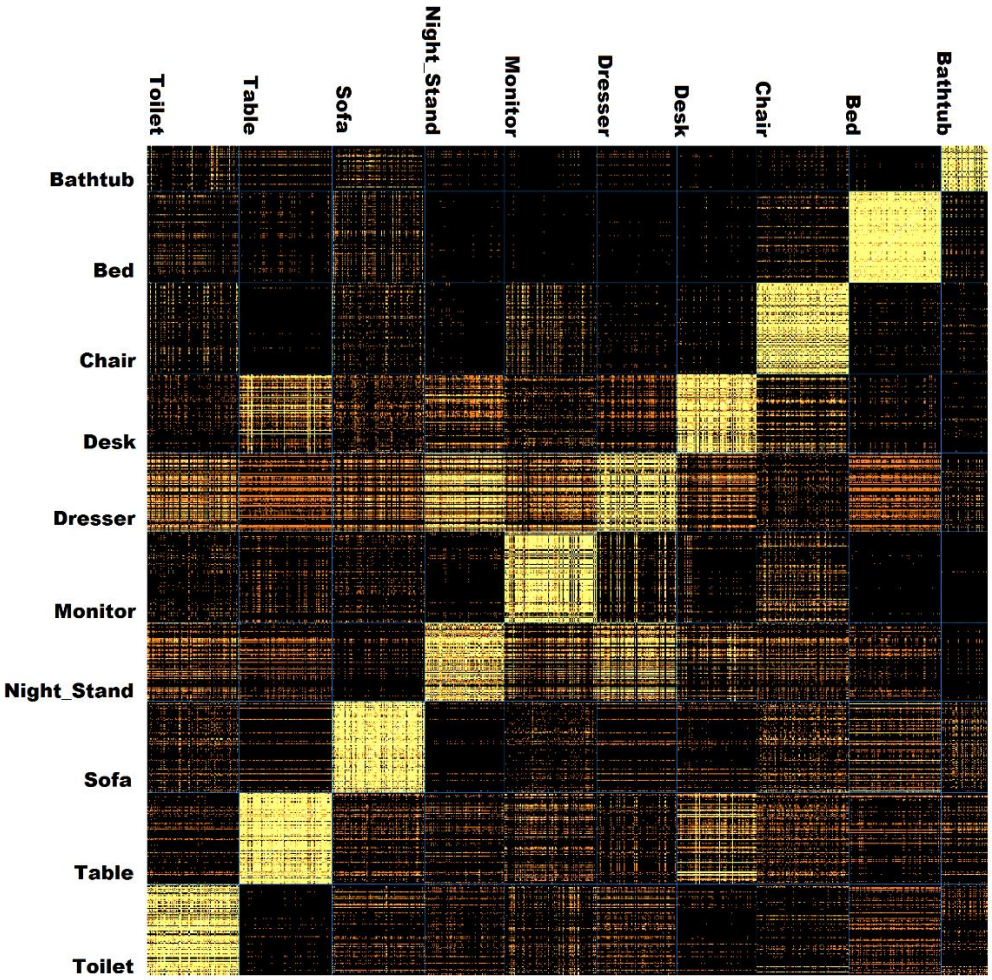


Figure IV-11: Image de niveau (Tier image) visualisant le plus proche voisin (blanc), le premier niveau (jaune), et le deuxième niveau (orange) calculée en faisant correspondre chaque objet 3D (lignes) avec chaque autre objet 3D (colonnes) dans la base de données ModelNet10 en utilisant 2DSlicesNet.

IV.4.3 **Conclusion**

Dans cette section, nous avons présenté 2DSlicesNet, une approche basée sur des images de coupe 2D et CNN3D pour la recherche et la classification d'objets 3D. Nous utilisons des images de coupe 2D d'objets 3D comme données d'entrée à un CNN3D, qui agrège les informations contenues dans les images de coupe 2D dans un descripteur robuste et discriminant. Les performances du réseau sont améliorées grâce à l'augmentation des données et à l'architecture du réseau. L'utilité et l'efficacité de 2DSlicesNet ont été démontrées sur les ensembles de données ModelNet10 et ModelNet40, en obtenant des résultats compétitifs dans une série d'expériences.

Conclusion et perspectives

V.1 Conclusion

Dans cette thèse, nous nous sommes intéressés aux problématiques associées à la classification et à la recherche de modèles 3D. Nos objectifs se sont portés sur le développement de nouvelles méthodes permettant de classer et de rechercher des objets 3D à partir de leur forme. En particulier, nous avons cherché à surmonter l'inconvénient majeur de la plupart des approches de classification et de recherche d'objets 3D proposées dans la littérature qui se manifeste dans leur faiblesse à correspondre les modèles 3D incomplets et imparfaits. Pour cela, nous avons proposé des approches de classification et de recherche d'objet 3D basées sur la similarité partielle. En fait, nous avons représenté chaque objet 3D par un ensemble d'images de coupe 2D, de sorte que la correspondance de forme entre les objets 3D soit transformée en mesure de similarité entre leurs images de coupe 2D. Quatre problèmes se sont impliqués dans ce processus : la sélection des directions de coupe, la méthode de coupe, le nombre d'images de coupe 2D et la mesure de similarité entre les images de coupe 2D.

Pour aborder ces problèmes, certaines stratégies et règles ont été proposées. Tout d'abord, nous avons commencé par une étape de normalisation pour nous assurer que les objets 3D similaires seront décomposés de la même manière. Ensuite, nous avons proposé une méthode de découpe qui peut être utilisée pour obtenir une série d'images de coupe 2D correspondant aux axes déterminés. Afin de réduire le stockage et le temps de recherche sans diminuer les performances de nos approches, nous avons utilisé des algorithmes d'exploration de données, notamment des algorithmes de Clustering et d'extraction des règles d'association, pour choisir parmi les images de coupe 2D extraites celles qui sont les plus représentatives. Enfin, une nouvelle métrique basée sur la distance de Hausdorf a été proposée pour mesurer la similarité entre les images de coupe 2D représentatives. D'autre part, nous avons profité de la puissance et de l'efficacité de l'apprentissage automatique pour extraire les caractéristiques les plus discriminantes d'images de coupe 2D. Plus précisément, nous avons utilisé les images de coupe 2D d'objets 3D afin d'entraîner un réseau de neurones convolutifs 3D (CNN-3D) à l'extraction des caractéristiques de haut niveau qui ont été utilisées, par la suite, pour concevoir des descripteurs de forme profond sur lesquels des tests de classification et de recherche ont été effectués. Dans ce qui suit, nous introduirons le résumé de nos contributions et les perspectives qui en découlent.

Notre première approche d'indexation et de recherche d'objets 3D, appelée K_RS (K Representative Slices), consiste à extraire, pour chaque objet 3D, un ensemble d'images de coupe 2D correspondant à ses trois axes principaux, puis chaque image a été représentée par les moments de Hu. Par la suite, l'algorithme de Clustering K-means a été utilisée pour sélectionner les images de coupe 2D représentatives, ce qui transforment la comparaison entre les objets 3D en un calcul de similarité entre leurs images de coupe 2D représentatives. Cette approche donne des résultats satisfaisants si le nombre de clusters est correctement choisi. Dans le cas contraire, le processus de Clustering produit un sur-partitionnement ou un sous-partitionnement ce qui affecte les performances de l'approche. Toutefois, les objets 3D ont des complexités distinctes au niveau de leurs structures, ce qui rend difficile voire impossible de trouver un nombre qui convient à tous les modèles 3D. Par conséquent, on se retrouve dans une sur-représentation ou sous-représentation des objets, ce qui diminue les performances de l'approche.

Afin de remédier à ce problème, nous avons proposé une deuxième approche, nommée ASC (Adaptive Slices Clustering), qui utilise un indice de validité de cluster pour adapter le nombre d'images de coupe 2D à la complexité de chaque objet 3D. Nous avons

présenté également une nouvelle métrique basée sur la distance de Hausdorff. En fait, la version originale de la distance de Hausdorff minimise la comparaison de deux ensembles à une comparaison d'un seul élément de chacun, ce qui peut conduire à des résultats incorrects lorsque certains éléments perturbés existent dans un ensemble. Pour cela, nous avons adopté une méthode efficace basée sur cette distance pour calculer la similarité des ensembles d'images de coupe représentatives, permettant ainsi de prendre en considération tous les éléments de l'ensemble. Les résultats expérimentaux ont montré que l'ASC donne, en particulier dans le cas d'objets 3D incomplets, des bons résultats en termes de performances de recherche d'objet 3D surpassant plusieurs méthodes bien connues dans la littérature, qui se basent sur une description globale des modèles 3D.

Notre troisième approche consiste à extraire un nombre important des images de coupe 2D selon des axes déterminés. Ensuite, les moments de Zernike ont été utilisés afin de décrire les images extraites. Par la suite, nous avons représenté les images de coupe de chaque modèle 3D dans une base de données transactionnelle. Après, nous avons utilisé l'algorithme « Apriori » afin de réduire l'ensemble initial des images de coupe pour en garder que les plus importantes. Enfin, la similarité entre les images de coupe représentatives des objets 3D a été mesurée en se basant sur notre propre métrique. L'approche proposée permet d'obtenir des performances nettement meilleures que les méthodes utilisées dans l'évaluation comparative sur la base de données Princeton Shape Benchmark (PSB).

Nous avons également proposé 2DSlicesNet, une approche de classification et de recherche d'objets 3D à base de réseaux de neurones convolutifs 3D pour apprendre hiérarchiquement des représentations de caractéristiques discriminantes et robustes. Le framework proposé a adopté une approche en plusieurs étapes qui représente d'abord chaque objet 3D dans la base de données comme un ensemble d'images de coupe 2D extraites tout au long du premier axe principal. Ces images de coupe 2D ont été d'abord redimensionnées, puis empilées dans une matrice qui a ensuite été introduite dans un réseau neuronal convolutionnel 3D pour extraire les représentations des caractéristiques de haut niveau. En fait, l'approche proposée extrait non seulement les informations contenues dans les images de coupe 2D mais elle préserve aussi leur ordre spatial, ce qui permet d'apprendre des caractéristiques complètes dans l'ensemble des images de coupe 2D de chaque objet. Nous avons montré par des évaluations expérimentales rigoureuses sur deux ensembles de données (ModelNet10 et ModelNet40) que 2DSlicesNet surpasse des méthodes de classification et de recherche d'objet 3D bien connues dans la littérature par une marge relativement importante.

V.2 Perspectives

Au cours de cette thèse, nous avons développé et mis au point de nouvelles approches de recherche et de classification d'objets 3D basées sur les images de coupe 2D. Toutefois, il existe encore des limites et des points d'amélioration qu'il conviendra d'aborder à l'avenir.

Les approches proposées donnent des résultats très satisfaisants en surpassant plusieurs méthodes de la littérature. Cependant, elles restent relativement coûteuses au niveau du temps, surtout la troisième approche parce qu'elle extrait un nombre important des images de coupe 2D. Pour cette raison, nous envisagerons d'intégrer des techniques de parallélisme comme les systèmes multi-agents pour rendre nos approches moins coûteuses en termes du temps.

En outre, nous pensons que les performances de 2DSlicesNet pourront être améliorées en prenant en considération non seulement les images de coupe 2D correspondant au premier axe principal mais également celles du deuxième et du troisième axe principal de l'objet 3D. En fait, nous pourrons améliorer considérablement 2DSlicesNet en combinant trois réseaux, qui prennent respectivement en entrée les images de coupe 2D correspondant aux trois axes principaux, et les entraînent de manière synchrone en utilisant

la technique de fusion de réseau. Ainsi, nous forçons le système à extraire des caractéristiques d'images de coupe 2D de plusieurs directions. Cet arrangement oblige les noyaux de convolution à être isotrope et à extraire des caractéristiques plus distinctes pour les tâches de classification et de recherche d'objet 3D.

Bien que nous soyons confrontés à une augmentation considérable de la disponibilité des données de formes 3D grâce aux récents progrès des techniques de numérisation 3D, le domaine de l'image 2D domine toujours le monde visuel. Cette omniprésence peut être caractérisée au mieux lorsqu'il s'agit de modèles profonds pré-entraînés où tous ont été entraînés en utilisant des images comme entrée et non des formes 3D. Ces modèles sont généralement entraînés en utilisant des dizaines de milliers d'images afin d'inclure toutes sortes de motifs et de textures pendant la phase d'entraînement, de sorte que ces modèles pré-entraînés puissent être utilisés comme extracteurs de caractéristiques plus tard sans avoir à consacrer du temps et des efforts à l'entraînement. Cependant, la représentation d'objets 3D par seulement des images 2D n'est pas toujours efficace parce que la projection ne saisit que les informations externes de l'objet 3D et écarte les informations internes qui peuvent être importantes pour reconnaitre certaines catégories spécifiques. De ce fait, il serait judicieux de proposer une architecture qui prenne en considération les informations internes et externes de l'objet 3D. En fait, malgré la supériorité de notre « 2DSlicesNet » pour la classification et la recherche d'objet 3D par rapport aux méthodes de la littérature, nous pensons que cette approche peut être améliorée en prenant également en entrée des images de projection 2D. De cette façon, nous pourrons profiter à la fois de la puissance des images de coupe 2D à représenter la structure interne d'objet 3D ainsi que les images de projection 2D qui donnent des informations externes sur l'objet 3D et qui offrent aussi la possibilité d'adopter une stratégie d'apprentissage par transfert en profitant des modèles pré-entrainés sur de grandes bases de données 2D.

Bibliographie

AGRAWAL, R., & SRIKANT, R. (1994, SEPTEMBER). FAST ALGORITHMS FOR MINING ASSOCIATION RULES. IN PROC. 20TH INT. CONF. VERY LARGE DATA BASES, VLDB (VOL. 1215, PP. 487-499).

AKGÜL, C. B., SANKUR, B., YEMEZ, Y., & SCHMITT, F. (2006). DENSITY-BASED 3D SHAPE DESCRIPTORS. EURASIP JOURNAL ON ADVANCES IN SIGNAL PROCESSING, 2007(1), 032503

AKGÜL, C. B., SANKUR, B., YEMEZ, Y., & SCHMITT, F. (2010). SIMILARITY LEARNING FOR 3D OBJECT RETRIEVAL USING RELEVANCE FEEDBACK AND RISK MINIMIZATION. INTERNATIONAL JOURNAL OF COMPUTER VISION, 89(2), 392-407

ALEXANDRE, L. A. (2016). 3D OBJECT RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS WITH TRANSFER LEARNING BETWEEN INPUT CHANNELS. IN INTELLIGENT AUTONOMOUS SYSTEMS 13 (PP. 889-898). SPRINGER, CHAM

ANDROUTSOPOULOS, I., KOUTSIAS, J., CHANDRINOS, K. V., PALIOURAS, G., & SPYROPOULOS, C. D. (2000). AN EVALUATION OF NAIVE BAYESIAN ANTI-SPAM FILTERING. ARXIV PREPRINT CS/0006013.

ANKERST, M., KASTENMÜLLER, G., KRIEGEL, H. P., & SEIDL, T. (1999A, JULY). 3D SHAPE HISTOGRAMS FOR SIMILARITY SEARCH AND CLASSIFICATION IN SPATIAL DATABASES. IN INTERNATIONAL SYMPOSIUM ON SPATIAL DATABASES (PP. 207-226). SPRINGER, BERLIN, HEIDELBERG

ANKERST, M., KASTENMÜLLER, G., KRIEGEL, H. P., & SEIDL, T. (1999B, AUGUST). NEAREST NEIGHBOR CLASSIFICATION IN 3D PROTEIN DATABASES. IN ISMB (VOL. 99, PP. 34-43).

ANSARY, T. F., VANDEBORRE, J. P., MAHMOUDI, S., & DAOUDI, M. (2004, SEPTEMBER). A BAYESIAN FRAMEWORK FOR 3D MODELS RETRIEVAL BASED ON CHARACTERISTIC VIEWS. IN PROCEEDINGS. 2ND INTERNATIONAL SYMPOSIUM ON 3D DATA PROCESSING, VISUALIZATION AND TRANSMISSION, 2004. 3DPVT 2004. (PP. 139-146). IEEE

ARBTER, K., SNYDER, W. E., BURKHARDT, H., & HIRZINGER, G. (1990). APPLICATION OF AFFINE-INVARIANT FOURIER DESCRIPTORS TO RECOGNITION OF 3-D OBJECTS. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 12(7), 640-647.

ARSALAN SOLTANI, A., HUANG, H., WU, J., KULKARNI, T. D., & TENENBAUM, J. B. (2017). SYNTHESIZING 3D SHAPES VIA MODELING MULTI-VIEW DEPTH MAPS AND SILHOUETTES WITH DEEP GENERATIVE NETWORKS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 1511-1519).

ASHBROOK, A., THACKER, N. A., ROCKETT, P. I., & BROWN, C. I. (1995, JULY). ROBUST RECOGNITION OF SCALED SHAPES USING PAIRWISE GEOMETRIC HISTOGRAMS. IN BMVC (VOL. 95, PP. 503-512).

ATALLAH, M. J. (1983). A LINEAR TIME ALGORITHM FOR THE HAUSDORFF DISTANCE BETWEEN CONVEX POLYGONS

ATIYA, A. F. (2001). BANKRUPTCY PREDICTION FOR CREDIT RISK USING NEURAL NETWORKS: A SURVEY AND NEW RESULTS. IEEE TRANSACTIONS ON NEURAL NETWORKS, 12(4), 929-935.

ATMOSUKARTO, I., LEOW, W. K., & HUANG, Z. (2005, JANUARY). FEATURE COMBINATION AND RELEVANCE FEEDBACK FOR 3D MODEL RETRIEVAL. IN 11TH INTERNATIONAL MULTIMEDIA MODELLING CONFERENCE (PP. 334-339). IEEE

ATTENE, M., FALCIDIENO, B., & SPAGNUOLO, M. (2006). HIERARCHICAL MESH SEGMENTATION BASED ON FITTING PRIMITIVES. THE VISUAL COMPUTER, 22(3), 181-193.

BAI, S., BAI, X., ZHOU, Z., ZHANG, Z., & JAN LATECKI, L. (2016). GIFT: A REAL-TIME AND SCALABLE 3D SHAPE SEARCH ENGINE. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 5023-5032).

BALLESTEROS, J., TRAVIESO, C. M., ALONSO, J. B., & FERRER, M. A. (2005). SLANT ESTIMATION OF HANDWRITTEN CHARACTERS BY MEANS OF ZERNIKE MOMENTS. ELECTRONICS LETTERS, 41(20), 1110-1112.

BARDINET, E., VIDAL, S. F., ARROYO, S. D., MALANDAIN, G., & DE LA BLANCA CAPILLA, N. P. (2000, FEBRUARY). STRUCTURAL OBJECT MATCHING. IN PROCEEDINGS

OF THE 2ND INTERNATIONAL SYMPOSIUM ON ADVANCED CONCEPTS FOR INTELLIGENT VISION SYSTEMS (ACIVS 2000) (VOL. 2, PP. 73-77).

BARTO, A. G., & SUTTON, R. S. (1997). REINFORCEMENT LEARNING IN ARTIFICIAL INTELLIGENCE. IN ADVANCES IN PSYCHOLOGY (VOL. 121, PP. 358-386). NORTH-HOLLAND.

BASRI, R., COSTA, L., GEIGER, D., & JACOBS, D. (1998). DETERMINING THE SIMILARITY OF DEFORMABLE SHAPES. VISION RESEARCH, 38(15-16), 2365-2385.

BATTAGLIA, P., PASCANU, R., LAI, M., & REZENDE, D. J. (2016). INTERACTION NETWORKS FOR LEARNING ABOUT OBJECTS, RELATIONS AND PHYSICS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 4502-4510).

BENGIO, Y. (2009). LEARNING DEEP ARCHITECTURES FOR AI. NOW PUBLISHERS INC.

BERKHIN, P. (2004). SURVEY OF CLUSTERING DATA MINING TECHNIQUES, 2002. ACCRUE SOFTWARE: SAN JOSE, CA.

BESL, P. J. (1994, DECEMBER). TRIANGLES AS A PRIMARY REPRESENTATION. IN INTERNATIONAL WORKSHOP ON OBJECT REPRESENTATION IN COMPUTER VISION (PP. 191-206). SPRINGER, BERLIN, HEIDELBERG

BINFORD, I. (1971). VISUAL PERCEPTION BY COMPUTER. IN IEEE CONFERENCE OF SYSTEMS AND CONTROL.

BISHOP, C. M. (2006). PATTERN RECOGNITION AND MACHINE LEARNING. SPRINGER.

BLUM, H. (1973). BIOLOGICAL SHAPE AND VISUAL SCIENCE (PART I). JOURNAL OF THEORETICAL BIOLOGY, 38(2), 205-287

BOGO, F., ROMERO, J., LOPER, M., & BLACK, M. J. (2014). FAUST: DATASET AND EVALUATION FOR 3D MESH REGISTRATION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 3794-3801).

BOSCAINI, D., MASCI, J., RODOLA, E., & BRONSTEIN, M. (2016). LEARNING SHAPE CORRESPONDENCE WITH ANISOTROPIC CONVOLUTIONAL NEURAL NETWORKS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 3189-3197).

BREIMAN, L. (2001). RANDOM FORESTS. MACHINE LEARNING, 45(1), 5-32.

BRIN, S., MOTWANI, R., ULLMAN, J. D., & TSUR, S. (1997, JUNE). DYNAMIC ITEMSET COUNTING AND IMPLICATION RULES FOR MARKET BASKET DATA. IN PROCEEDINGS OF THE 1997 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA (PP. 255-264).

BROCK, A., LIM, T., RITCHIE, J. M., & WESTON, N. (2016). GENERATIVE AND DISCRIMINATIVE VOXEL MODELING WITH CONVOLUTIONAL NEURAL NETWORKS. ARXIV PREPRINT ARXIV:1608.04236.

BROSSETTE, S. E., SPRAGUE, A. P., HARDIN, J. M., WAITES, K. B., JONES, W. T., & MOSER, S. A. (1998). ASSOCIATION RULES AND DATA MINING IN HOSPITAL INFECTION CONTROL AND PUBLIC HEALTH SURVEILLANCE. JOURNAL OF THE AMERICAN MEDICAL INFORMATICS ASSOCIATION, 5(4), 373-381.

BRONSTEIN, M. M., BRUNA, J., LECUN, Y., SZLAM, A., & VANDERGHEYNST, P. (2017). GEOMETRIC DEEP LEARNING: GOING BEYOND EUCLIDEAN DATA. IEEE SIGNAL PROCESSING MAGAZINE, 34(4), 18-42.

BRONSTEIN, M. M., & KOKKINOS, I. (2010, JUNE). SCALE-INVARIANT HEAT KERNEL SIGNATURES FOR NON-RIGID SHAPE RECOGNITION. IN 2010 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 1704-1711). IEEE.

BRONSTEIN, A. M., BRONSTEIN, M. M., & KIMMEL, R. (2008). NUMERICAL GEOMETRY OF NON-RIGID SHAPES. SPRINGER SCIENCE & BUSINESS MEDIA

BRONSTEIN, A. M., BRONSTEIN, M. M., CASTELLANI, U., FALCIDIENO, B., FUSIELLO, A., GODIL, A., ... & PATANÉ, G. (2010). SHREC 2010: ROBUST LARGE-SCALE SHAPE RETRIEVAL BENCHMARK. PROC. 3DOR, 5(4)

BRUNA, J., ZAREMBA, W., SZLAM, A., & LECUN, Y. (2013). SPECTRAL NETWORKS AND LOCALLY CONNECTED NETWORKS ON GRAPHS. ARXIV PREPRINT ARXIV:1312.6203.

BU, S., HAN, P., LIU, Z., HAN, J., & LIN, H. (2015). LOCAL DEEP FEATURE LEARNING FRAMEWORK FOR 3D SHAPE. COMPUTERS & GRAPHICS, 46, 117-129.

BU, S., LIU, Z., HAN, J., WU, J., & JI, R. (2014). LEARNING HIGH-LEVEL FEATURE BY DEEP BELIEF NETWORKS FOR 3-D MODEL RETRIEVAL AND RECOGNITION. IEEE TRANSACTIONS ON MULTIMEDIA, 16(8), 2154-2167.

CANTERAKIS, N. (1999). 3D ZERNIKE MOMENTS AND ZERNIKE AFFINE INVARIANTS FOR 3D IMAGE ANALYSIS AND RECOGNITION. IN IN 11TH SCANDINAVIAN CONF. ON IMAGE ANALYSIS

CAO, Z., HUANG, Q., & KARTHIK, R. (2017, OCTOBER). 3D OBJECT CLASSIFICATION VIA SPHERICAL PROJECTIONS. IN 2017 INTERNATIONAL CONFERENCE ON 3D VISION (3DV) (PP. 566-574). IEEE.

CHANG, M. B., ULLMAN, T., TORRALBA, A., & TENENBAUM, J. B. (2016). A COMPOSITIONAL OBJECT-BASED APPROACH TO LEARNING PHYSICAL DYNAMICS. ARXIV PREPRINT ARXIV:1612.00341

CHEN, D. Y., TIAN, X. P., SHEN, Y. T., & OUHYOUNG, M. (2003, SEPTEMBER). ON VISUAL SIMILARITY BASED 3D MODEL RETRIEVAL. IN COMPUTER GRAPHICS FORUM (VOL. 22, NO. 3, PP. 223-232). OXFORD, UK: BLACKWELL PUBLISHING, INC

CHEN, Z., LI, X., & BRUNA, J. (2017). SUPERVISED COMMUNITY DETECTION WITH LINE GRAPH NEURAL NETWORKS. ARXIV PREPRINT ARXIV:1705.08415.

CHRISTIAN, S., SERGEY, I., VINCENT, V., & ALEXANDER, A. A. (2017, FEBRUARY). INCEPTION-V4 INCEPTION-RESNET AND THE IMPACT OF RESIDUAL CONNECTIONS ON LEARNING. IN AAAI (VOL. 4).

CICIRELLO, V., & REGLI, W. C. (2001, MAY). MACHINING FEATURE-BASED COMPARISONS OF MECHANICAL PARTS. IN PROCEEDINGS INTERNATIONAL CONFERENCE ON SHAPE MODELING AND APPLICATIONS (PP. 176-185). IEEE

COATES, A., & NG, A. Y. (2011). SELECTING RECEPTIVE FIELDS IN DEEP NETWORKS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 2528-2536).

CORTES, C., & VAPNIK, V. (1995). SUPPORT-VECTOR NETWORKS. MACHINE LEARNING, 20(3), 273-297.

COUPRIE, C., FARABET, C., NAJMAN, L., & LECUN, Y. (2013). INDOOR SEMANTIC SEGMENTATION USING DEPTH INFORMATION. ARXIV PREPRINT ARXIV:1301.3572.

CYR, C. M., & KIMIA, B. B. (2001, JULY). 3D OBJECT RECOGNITION USING SHAPE SIMILIARITY-BASED ASPECT GRAPH. IN PROCEEDINGS EIGHTH IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. ICCV 2001 (VOL. 1, PP. 254-261). IEEE

DAI, A., CHANG, A. X., SAVVA, M., HALBER, M., FUNKHOUSER, T., & NIEßNER, M. (2017). SCANNET: RICHLY-ANNOTATED 3D RECONSTRUCTIONS OF INDOOR SCENES. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 5828-5839).

DAI, H., & LIAO, S. (2014). CENTRAL-SYMMETRICAL PROPERTY ANALYSIS ON CIRCULARLY ORTHOGONAL MOMENTS. JOURNAL OF THEORETICAL AND APPLIED COMPUTER SCIENCE, 8(2), 11-26.

DANIELS, J. I., HA, L. K., OCHOTTA, T., & SILVA, C. T. (2007, JUNE). ROBUST SMOOTH FEATURE EXTRACTION FROM POINT CLOUDS. IN IEEE INTERNATIONAL CONFERENCE ON SHAPE MODELING AND APPLICATIONS 2007 (SMI'07) (PP. 123-136). IEEE.

DARAS, P., & AXENOPOULOS, A. (2010). A 3D SHAPE RETRIEVAL FRAMEWORK SUPPORTING MULTIMODAL QUERIES. INTERNATIONAL JOURNAL OF COMPUTER VISION, 89(2-3), 229-247.

DE ALARCÓN, P. A., PASCUAL-MONTANO, A. D., & CARAZO, J. M. (2002, JULY). SPIN IMAGES AND NEURAL NETWORKS FOR EFFICIENT CONTENT-BASED RETRIEVAL IN 3D OBJECT DATABASES. IN INTERNATIONAL CONFERENCE ON IMAGE AND VIDEO RETRIEVAL (PP. 225-234). SPRINGER, BERLIN, HEIDELBERG.

DEFFERRARD, M., BRESSON, X., & VANDERGHEYNST, P. (2016). CONVOLUTIONAL NEURAL NETWORKS ON GRAPHS WITH FAST LOCALIZED SPECTRAL FILTERING. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 3844-3852).

DEMARSIN, K., VANDERSTRAETEN, D., VOLODINE, T., & ROOSE, D. (2006, JULY). DETECTION OF CLOSED SHARP FEATURE LINES IN POINT CLOUDS FOR REVERSE ENGINEERING APPLICATIONS. IN INTERNATIONAL CONFERENCE ON GEOMETRIC MODELING AND PROCESSING (PP. 571-577). SPRINGER, BERLIN, HEIDELBERG

DENG, J., DONG, W., SOCHER, R., LI, L. J., LI, K., & FEI-FEI, L. (2009, JUNE). IMAGENET: A LARGE-SCALE HIERARCHICAL IMAGE DATABASE. IN 2009 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 248-255). IEEE.

DUDA, R. O., & HART, P. E. (1973). PATTERN CLASSIFICATION AND SCENE ANALYSIS (VOL. 3, PP. 731-739). NEW YORK: WILEY

DUVENAUD, D. K., MACLAURIN, D., IPARRAGUIRRE, J., BOMBARELL, R., HIRZEL, T., ASPURU-GUZIK, A., & ADAMS, R. P. (2015). CONVOLUTIONAL NETWORKS ON GRAPHS FOR LEARNING MOLECULAR FINGERPRINTS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 2224-2232).

EDELSBRUNNER, H., & MÜCKE, E. P. (1994). THREE-DIMENSIONAL ALPHA SHAPES. ACM TRANSACTIONS ON GRAPHICS (TOG), 13(1), 43-72

EIGEN, D., PUHRSCH, C., & FERGUS, R. (2014). DEPTH MAP PREDICTION FROM A SINGLE IMAGE USING A MULTI-SCALE DEEP NETWORK. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 2366-2374).

EITEL, A., SPRINGENBERG, J. T., SPINELLO, L., RIEDMILLER, M., & BURGARD, W. (2015, SEPTEMBER). MULTIMODAL DEEP LEARNING FOR ROBUST RGB-D OBJECT RECOGNITION. IN 2015 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS) (PP. 681-687). IEEE

ELAD, M., TAL, A., & AR, S. (2002). CONTENT BASED RETRIEVAL OF VRML OBJECTS—AN ITERATIVE AND INTERACTIVE APPROACH. IN MULTIMEDIA 2001 (PP. 107-118). SPRINGER, VIENNA

ESTER, M., KRIEGEL, H. P., SANDER, J., & XU, X. (1996, AUGUST). A DENSITY-BASED ALGORITHM FOR DISCOVERING CLUSTERS IN LARGE SPATIAL DATABASES WITH NOISE. IN KDD (VOL. 96, NO. 34, PP. 226-231).

FARABET, C., COUPRIE, C., NAJMAN, L., & LECUN, Y. (2013). LEARNING HIERARCHICAL FEATURES FOR SCENE LABELING.

FAYYAD, U., PIATETSKY-SHAPIRO, G., & SMYTH, P. (1996). FROM DATA MINING TO KNOWLEDGE DISCOVERY IN DATABASES. AI MAGAZINE, 17(3), 37-37.

FENG, J., WANG, Y., & CHANG, S. F. (2016, MARCH). 3D SHAPE RETRIEVAL USING A SINGLE DEPTH IMAGE FROM LOW-COST SENSORS. IN 2016 IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV) (PP. 1-9). IEEE.

FEY, M., LENSSEN, J. E., WEICHERT, F., & MÜLLER, H. (2017). SPLINECNN: FAST GEOMETRIC DEEP LEARNING WITH CONTINUOUS B-SPLINE KERNELS. ARXIV PREPRINT ARXIV:1711.08920.

FU, B., LIU, J., FAN, X., & QUAN, Y. (2007, AUGUST). A HYBRID ALGORITHM OF FAST AND ACCURATE COMPUTING ZERNIKE MOMENTS. IN FOURTH INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD 2007) (VOL. 3, PP. 268-272). IEEE.

FUKUSHIMA, K. (1980). BIOLOGICAL CYBERNETICS NEOCOGNITRON: A SELF-ORGANIZING NEURAL NETWORK MODEL FOR A MECHANISM OF PATTERN RECOGNITION UNAFFECTED BY SHIFT IN POSITION. BIOL CYBERN, 36, 193-202.

FUKUSHIMA, K., MIYAKE, S., & ITO, T. (1983). NEOCOGNITRON: A NEURAL NETWORK MODEL FOR A MECHANISM OF VISUAL PATTERN RECOGNITION. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, (5), 826-834.

FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., & JACOBS, D. (2003). A SEARCH ENGINE FOR 3D MODELS. ACM TRANSACTIONS ON GRAPHICS (TOG), 22(1), 83-105.

GAO, Y., WANG, M., JI, R., WU, X., & DAI, Q. (2013). 3-D OBJECT RETRIEVAL WITH HAUSDORFF DISTANCE LEARNING. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 61(4), 2088-2098.

GAO, Y., DAI, Q., & ZHANG, N. Y. (2010). 3D MODEL COMPARISON USING SPATIAL STRUCTURE CIRCULAR DESCRIPTOR. PATTERN RECOGNITION, 43(3), 1142-1151.

GAO, X., WANG, Q., LI, X., TAO, D., & ZHANG, K. (2011). ZERNIKE-MOMENT-BASED IMAGE SUPER RESOLUTION. IEEE TRANSACTIONS ON IMAGE PROCESSING, 20(10), 2738-2747.

GARCIA-GARCIA, A., GOMEZ-DONOSO, F., GARCIA-RODRIGUEZ, J., ORTS-ESCOLANO, S., CAZORLA, M., & AZORIN-LOPEZ, J. (2016, JULY). POINTNET: A 3D CONVOLUTIONAL NEURAL NETWORK FOR REAL-TIME OBJECT CLASS RECOGNITION. IN 2016 INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN) (PP. 1578-1584). IEEE.

GIDARIS, S., & KOMODAKIS, N. (2015). OBJECT DETECTION VIA A MULTI-REGION AND SEMANTIC SEGMENTATION-AWARE CNN MODEL. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (PP. 1134-1142).

GLOROT, X., BORDES, A., & BENGIO, Y. (2011, JUNE). DEEP SPARSE RECTIFIER NEURAL NETWORKS. IN PROCEEDINGS OF THE FOURTEENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND STATISTICS (PP. 315-323).

GOMEZ-DONOSO, F., GARCIA-GARCIA, A., GARCIA-RODRIGUEZ, J., ORTS-ESCOLANO, S., & CAZORLA, M. (2017, MAY). LONCHANET: A SLICED-BASED CNN ARCHITECTURE FOR REAL-TIME 3D OBJECT RECOGNITION. IN 2017 INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN) (PP. 412-418). IEEE.

GOPE, C., KEHTARNAVAZ, N., & HILLMAN, G. (2004, SEPTEMBER). ZERNIKE MOMENT INVARIANTS BASED PHOTO-IDENTIFICATION USING FISHER DISCRIMINANT MODEL. IN THE 26TH ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY (VOL. 1, PP. 1455-1458). IEEE

GOSSWEILER, R., & LIMBER, M. (2006). SKETCHUP: AN EASY-TO-USE 3D DESIGN TOOL THAT INTEGRATES WITH GOOGLE EARTH. IN ADJUNCT PROCEEDINGS OF THE 19TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY (UIST06) (VOL. 19, P. 3).

GORI, M., MONFARDINI, G., & SCARSELLI, F. (2005, JULY). A NEW MODEL FOR LEARNING IN GRAPH DOMAINS. IN PROCEEDINGS. 2005 IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2005. (VOL. 2, PP. 729-734). IEEE.

GOTTSCHALK, S., MANOCHA, D., & LIN, M. C. (2000). COLLISION QUERIES USING ORIENTED BOUNDING BOXES (DOCTORAL DISSERTATION, UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL).

GREGOR, K., & LECUN, Y. (2010). EMERGENCE OF COMPLEX-LIKE CELLS IN A TEMPORAL PRODUCT NETWORK WITH LOCAL RECEPTIVE FIELDS. ARXIV PREPRINT ARXIV:1006.0448.

GUMHOLD, S., WANG, X., & MACLEOD, R. S. (2001, OCTOBER). FEATURE EXTRACTION FROM POINT CLOUDS. IN IMR (PP. 293-305).

HAN, Z., LIU, Z., HAN, J., VONG, C. M., BU, S., & CHEN, C. L. P. (2017). MESH CONVOLUTIONAL RESTRICTED BOLTZMANN MACHINES FOR UNSUPERVISED LEARNING OF FEATURES WITH STRUCTURE PRESERVATION ON 3-D MESHES. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 28(10), 2268-2281..

HAN, J., PEI, J., & YIN, Y. (2000). MINING FREQUENT PATTERNS WITHOUT CANDIDATE GENERATION. ACM SIGMOD RECORD, 29(2), 1-12.

HANDA, A., PĂTRĂUCEAN, V., STENT, S., & CIPOLLA, R. (2016, MAY). SCENENET: AN ANNOTATED MODEL GENERATOR FOR INDOOR SCENE UNDERSTANDING. IN 2016 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA) (PP. 5737-5743). IEEE

HARTIGAN, J. A., & WONG, M. A. (1979). ALGORITHM AS 136: A K-MEANS CLUSTERING ALGORITHM. JOURNAL OF THE ROYAL STATISTICAL SOCIETY. SERIES C (APPLIED STATISTICS), 28(1), 100-108

HE, K., ZHANG, X., REN, S., & SUN, J. (2015). DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION. ARXIV, ARXIV-1512

HE, K., ZHANG, X., REN, S., & SUN, J. (2016). DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 770-778).

HEATON, J. (2016, MARCH). COMPARING DATASET CHARACTERISTICS THAT FAVOR THE APRIORI, ECLAT OR FP-GROWTH FREQUENT ITEMSET MINING ALGORITHMS. IN SOUTHEASTCON 2016 (PP. 1-7). IEEE.

HECZKO, M., KEIM, D., SAUPE, D., & VRANIC, D. V. (2001). A METHOD FOR SIMILARITY SEARCH OF 3D OBJECTS. PROC. OF BTW 2001, 384-401

HILAGA, M., SHINAGAWA, Y., KOHMURA, T., & KUNII, T. L. (2001, AUGUST). TOPOLOGY MATCHING FOR FULLY AUTOMATIC SIMILARITY ESTIMATION OF 3D SHAPES. IN PROCEEDINGS OF THE 28TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE

HINTON, G., DENG, L., YU, D., DAHL, G. E., MOHAMED, A. R., JAITLY, N., ... & KINGSBURY, B. (2012). DEEP NEURAL NETWORKS FOR ACOUSTIC MODELING IN SPEECH RECOGNITION: THE SHARED VIEWS OF FOUR RESEARCH GROUPS. IEEE SIGNAL PROCESSING MAGAZINE, 29(6), 82-97TECHNIQUES (PP. 203-212).

HIPP, J., GÜNTZER, U., & NAKHAEIZADEH, G. (2000). ALGORITHMS FOR ASSOCIATION RULE MINING—A GENERAL SURVEY AND COMPARISON. ACM SIGKDD EXPLORATIONS NEWSLETTER, 2(1), 58-64.

HLAVATÝ, T., & SKALA, V. (2003). A SURVEY FOR METHODS FOR 3D MODEL FEATURE EXTRACTION

HORN, B. K. P. (1984). EXTENDED GAUSSIAN IMAGES. PROCEEDINGS OF THE IEEE, 72(12), 1671-1686.

HORNIK, K. (1991). APPROXIMATION CAPABILITIES OF MULTILAYER FEEDFORWARD NETWORKS. NEURAL NETWORKS, 4(2), 251-257.

HOU, S., LOU, K., & RAMANI, K. (2005). SVM-BASED SEMANTIC CLUSTERING AND RETRIEVAL OF A 3D MODEL DATABASE. COMPUTER-AIDED DESIGN AND APPLICATIONS, 2(1-4), 155-164.

HOUTSMA, M., & SWAMI, A. (1995, MARCH). SET-ORIENTED MINING FOR ASSOCIATION RULES IN RELATIONAL DATABASES. IN PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (PP. 25-33). IEEE.

HU, M. K. (1962). VISUAL PATTERN RECOGNITION BY MOMENT INVARIANTS. IRE TRANSACTIONS ON INFORMATION THEORY, 8(2), 179-187.

HUANG, G., SUN, Y., LIU, Z., SEDRA, D., & WEINBERGER, K. Q. (2016, OCTOBER). DEEP NETWORKS WITH STOCHASTIC DEPTH. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 646-661). SPRINGER, CHAM

HUANG, Z., & LENG, J. (2010, APRIL). ANALYSIS OF HU'S MOMENT INVARIANTS ON IMAGE SCALING AND ROTATION. IN 2010 2ND INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND TECHNOLOGY (VOL. 7, PP. V7-476). IEEE.

HWANG, S. K., & KIM, W. Y. (2006). A NOVEL APPROACH TO THE FAST COMPUTATION OF ZERNIKE MOMENTS. PATTERN RECOGNITION, 39(11), 2065-2076.

IOFFE, S., & SZEGEDY, C. (2015). BATCH NORMALIZATION: ACCELERATING DEEP NETWORK TRAINING BY REDUCING INTERNAL COVARIATE SHIFT. ARXIV PREPRINT ARXIV:1502.03167.

IP, C. Y., REGLI, W. C., SIEGER, L., & SHOKOUFANDEH, A. (2003, JUNE). AUTOMATED LEARNING OF MODEL CLASSIFICATIONS. IN PROCEEDINGS OF THE EIGHTH ACM SYMPOSIUM ON SOLID MODELING AND APPLICATIONS (PP. 322-327).

JIANTAO, P., YI, L., GUYU, X., HONGBIN, Z., WEIBIN, L., & UEHARA, Y. (2004, SEPTEMBER). 3D MODEL RETRIEVAL BASED ON 2D SLICE SIMILARITY MEASUREMENTS. IN PROCEEDINGS. 2ND INTERNATIONAL SYMPOSIUM ON 3D DATA PROCESSING, VISUALIZATION AND TRANSMISSION, 2004. 3DPVT 2004. (PP. 95-101). IEEE.

JEANNIN, S., CIEPLINSKI, L., OHM, J., & KIM, M. (2001). MPEG-7 VISUAL PART OF EXPERIMENTATION MODEL VERSION 9.0. ISO/IEC JTC1/SC29/WG11 N, 3914.

JOHNS, E., LEUTENEGGER, S., & DAVISON, A. J. (2016). PAIRWISE DECOMPOSITION OF IMAGE SEQUENCES FOR ACTIVE MULTI-VIEW RECOGNITION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 3813-3822).

JOHNSON, A. E., & HEBERT, M. (1999). USING SPIN IMAGES FOR EFFICIENT OBJECT RECOGNITION IN CLUTTERED 3D SCENES. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 21(5), 433-449

KANG, S. B., & IKEUCHI, K. (1993). THE COMPLEX EGI: A NEW REPRESENTATION FOR 3-D POSE DETERMINATION. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 15(7), 707-721.

KANG, S. B., & IKEUCHI, K. (1991, JUNE). DETERMINING 3-D OBJECT POSE USING THE COMPLEX EXTENDED GAUSSIAN IMAGE. IN CVPR (VOL. 91, PP. 580-585).

KANEZAKI, A., MATSUSHITA, Y., & NISHIDA, Y. (2016). ROTATIONNET: JOINT OBJECT CATEGORIZATION AND POSE ESTIMATION USING MULTIVIEWS FROM UNSUPERVISED VIEWPOINTS. ARXIV PREPRINT ARXIV:1603.06208.

KATZ, S., & TAL, A. (2003). HIERARCHICAL MESH DECOMPOSITION USING FUZZY CLUSTERING AND CUTS. ACM TRANSACTIONS ON GRAPHICS (TOG), 22(3), 954-961.

KATZ, S., LEIFMAN, G., & TAL, A. (2005). MESH SEGMENTATION USING FEATURE POINT AND CORE EXTRACTION. THE VISUAL COMPUTER, 21(8-10), 649-658.

KAUFMAN, L., & ROUSSEEUW, P. J. (2009). FINDING GROUPS IN DATA: AN INTRODUCTION TO CLUSTER ANALYSIS (VOL. 344). JOHN WILEY & SONS.

KAZHDAN, M., FUNKHOUSER, T., & RUSINKIEWICZ, S. (2003, JUNE). ROTATION INVARIANT SPHERICAL HARMONIC REPRESENTATION OF 3D SHAPE DESCRIPTORS. IN SYMPOSIUM ON GEOMETRY PROCESSING (VOL. 6, PP. 156-164).

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FINKELSTEIN, A., & FUNKHOUSER, T. (2002, MAY). A REFLECTIVE SYMMETRY DESCRIPTOR. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 642-656). SPRINGER, BERLIN, HEIDELBERG

KAZHDAN, M., FUNKHOUSER, T., & RUSINKIEWICZ, S. (2004). SHAPE MATCHING AND ANISOTROPY. IN ACM SIGGRAPH 2004 PAPERS (PP. 623-629).

KAWAGUCHI, K. (2016). DEEP LEARNING WITHOUT POOR LOCAL MINIMA. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 586-594).

KHALIL, E., DAI, H., ZHANG, Y., DILKINA, B., & SONG, L. (2017). LEARNING COMBINATORIAL OPTIMIZATION ALGORITHMS OVER GRAPHS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 6348-6358).

KHAN, S. H., GUO, Y., HAYAT, M., & BARNES, N. (2019). UNSUPERVISED PRIMITIVE DISCOVERY FOR IMPROVED 3D GENERATIVE MODELING. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 9739-9748).

KIM, D. J., PARK, Y. W., & PARK, D. J. (2001). A NOVEL VALIDITY INDEX FOR DETERMINATION OF THE OPTIMAL NUMBER OF CLUSTERS. IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS, 84(2), 281-285.

KIPF, T. N., & WELLING, M. (2016). SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS. ARXIV PREPRINT ARXIV: 1609.02907

KOLONIAS, L., TZOVARAS, D., MALASSIOTIS, S., & STRINTZIS, M. G. (2001, OCTOBER). FAST CONTENT-BASED SEARCH OF VRML MODELS BASED ON SHAPE DESCRIPTORS. IN PROCEEDINGS 2001 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (CAT. NO. 01CH37205) (VOL. 2, PP. 133-136). IEEE.

KOVNATSKY, A., BRONSTEIN, M. M., BRONSTEIN, A. M., GLASHOFF, K., & KIMMEL, R. (2013, MAY). COUPLED QUASI-HARMONIC BASES. IN COMPUTER GRAPHICS FORUM (VOL. 32, NO. 2PT4, PP. 439-448). OXFORD, UK: BLACKWELL PUBLISHING LTD.

KLOKOV, R., & LEMPITSKY, V. (2017). ESCAPE FROM CELLS: DEEP KD-NETWORKS FOR THE RECOGNITION OF 3D POINT CLOUD MODELS. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (PP. 863-872).

KRIZHEVSKY, A., SUTSKEVER, I., & HINTON, G. E. (2012). IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 1097-1105).

LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W., & JACKEL, L. D. (1989). BACKPROPAGATION APPLIED TO HANDWRITTEN ZIP CODE RECOGNITION. NEURAL COMPUTATION, 1(4), 541-551.

LECUN, Y. (1985). UNE PROCEDURE D'APPRENTISSAGE PONR RESEAU A SEUIL ASYMETRIQUE. PROCEEDINGS OF COGNITIVA 85, 599-604.

LEE, H., GROSSE, R., RANGANATH, R., & NG, A. Y. (2009, JUNE). CONVOLUTIONAL DEEP BELIEF NETWORKS FOR SCALABLE UNSUPERVISED LEARNING OF HIERARCHICAL REPRESENTATIONS. IN PROCEEDINGS OF THE 26TH ANNUAL INTERNATIONAL CONFERENCE ON MACHINE LEARNING (PP. 609-616).

LEIFMAN, G., MEIR, R., & TAL, A. (2004, OCTOBER). RELEVANCE FEEDBACK FOR 3D SHAPE RETRIEVAL. IN THE 5TH ISRAEL-KOREA BI-NATIONAL CONFERENCE ON GEOMETRIC MODELING AND COMPUTER GRAPHICS (PP. 15-19).

LEIFMAN, G., MEIR, R., & TAL, A. (2005). SEMANTIC-ORIENTED 3D SHAPE RETRIEVAL USING RELEVANCE FEEDBACK. THE VISUAL COMPUTER, 21(8-10), 865-875.

LENG, B., GUO, S., ZHANG, X., & XIONG, Z. (2015). 3D OBJECT RETRIEVAL WITH STACKED LOCAL CONVOLUTIONAL AUTOENCODER. SIGNAL PROCESSING, 112, 119-128.

LENG, B., ZHANG, X., YAO, M., & XIONG, Z. (2014, JANUARY). 3D OBJECT CLASSIFICATION USING DEEP BELIEF NETWORKS. IN INTERNATIONAL CONFERENCE ON MULTIMEDIA MODELING (PP. 128-139). SPRINGER, CHAM.

LENG, B., ZENG, J., YAO, M., & XIONG, Z. (2014). 3D OBJECT RETRIEVAL WITH MULTITOPIC MODEL COMBINING RELEVANCE FEEDBACK AND LDA MODEL. IEEE TRANSACTIONS ON IMAGE PROCESSING, 24(1), 94-105.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., ... & SHADE, J. (2000, JULY). THE DIGITAL MICHELANGELO PROJECT: 3D SCANNING OF LARGE STATUES. IN PROCEEDINGS OF THE 27TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES (PP. 131-144).

LEYMARIE, F. F., & KIMIA, B. B. (2001, MAY). THE SHOCK SCAFFOLD FOR REPRESENTING 3D SHAPE. IN INTERNATIONAL WORKSHOP ON VISUAL FORM (PP. 216-227). SPRINGER, BERLIN, HEIDELBERG

LI, J., CHEN, B. M., & HEE LEE, G. (2018). SO-NET: SELF-ORGANIZING NETWORK FOR POINT CLOUD ANALYSIS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 9397-9406).

LI, Y., TARLOW, D., BROCKSCHMIDT, M., & ZEMEL, R. (2015). GATED GRAPH SEQUENCE NEURAL NETWORKS. ARXIV PREPRINT ARXIV:1511.05493.

LI, Y., PIRK, S., SU, H., QI, C. R., & GUIBAS, L. J. (2016). FPNN: FIELD PROBING NEURAL NETWORKS FOR 3D DATA. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 307-315).

LI, W., HAN, J., & PEICMAR, J. (2001). ACCURATE AND EFFICIENT CLASSIFICATION BASED ON MULTIPLE CLASS-ASSOCIATION RULES PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON DATA MINING, ICDM

LIU, Z., CHEN, S., BU, S., & LI, K. (2014, JULY). HIGH-LEVEL SEMANTIC FEATURE FOR 3D SHAPE BASED ON DEEP BELIEF NETWORKS. IN 2014 IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME) (PP. 1-6). IEEE

LIU, Z. B., BU, S. H., ZHOU, K., GAO, S. M., HAN, J. W., & WU, J. (2013). A SURVEY ON PARTIAL RETRIEVAL OF 3D SHAPES. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY, 28(5), 836-851.

LIU, A., WANG, Z., NIE, W., & SU, Y. (2015). GRAPH-BASED CHARACTERISTIC VIEW SET EXTRACTION AND MATCHING FOR 3D MODEL RETRIEVAL. INFORMATION SCIENCES, 320, 429-442.

LIU, H., & SETIONO, R. (1997). FEATURE SELECTION VIA DISCRETIZATION. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 9(4), 642-645.

LIU, S., GILES, L., & ORORBIA, A. (2018, SEPTEMBER). LEARNING A HIERARCHICAL LATENT-VARIABLE MODEL OF 3D SHAPES. IN 2018 INTERNATIONAL CONFERENCE ON 3D VISION (3DV) (PP. 542-551). IEEE.

LONG, J., SHELHAMER, E., & DARRELL, T. (2015). FULLY CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 3431-3440).

MA, W. C., WU, F. C., & OUHYOUNG, M. (2003, MAY). SKELETON EXTRACTION OF 3D OBJECTS WITH RADIAL BASIS FUNCTIONS. IN 2003 SHAPE MODELING INTERNATIONAL. (PP. 207-215). IEEE

MA, C., GUO, Y., LEI, Y., & AN, W. (2018). BINARY VOLUMETRIC CONVOLUTIONAL NEURAL NETWORKS FOR 3-D OBJECT RECOGNITION. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, 68(1), 38-48

MAAOUI, C., LAURENT, H., & ROSENBERGER, C. (2005, SEPTEMBER). 2D COLOR SHAPE RECOGNITION USING ZERNIKE MOMENTS. IN IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING 2005 (VOL. 3, PP. III-976). IEEE.

MATURANA, D., & SCHERER, S. (2015, SEPTEMBER). VOXNET: A 3D CONVOLUTIONAL NEURAL NETWORK FOR REAL-TIME OBJECT RECOGNITION. IN 2015 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS) (PP. 922-928). IEEE

MASCI, J., BOSCAINI, D., BRONSTEIN, M., & VANDERGHEYNST, P. (2015). GEODESIC CONVOLUTIONAL NEURAL NETWORKS ON RIEMANNIAN MANIFOLDS. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (PP. 37-45

MCCULLAGH, P. (2018). GENERALIZED LINEAR MODELS. ROUTLEDGE

MCWHERTER, D., PEABODY, M., REGLI, W. C., & SHOKOUFANDEH, A. (2001, SEPTEMBER). TRANSFORMATION INVARIANT SHAPE SIMILARITY COMPARISON OF SOLID MODELS. IN PROC. ASME DETC.

MIN, P., HALDERMAN, J. A., KAZHDAN, M., & FUNKHOUSER, T. A. (2003, MARCH). EARLY EXPERIENCES WITH A 3D MODEL SEARCH ENGINE. IN PROCEEDINGS OF THE EIGHTH INTERNATIONAL CONFERENCE ON 3D WEB TECHNOLOGY (PP. 7-FF).

MIN, P., KAZHDAN, M., & FUNKHOUSER, T. (2004, SEPTEMBER). A COMPARISON OF TEXT AND SHAPE MATCHING FOR RETRIEVAL OF ONLINE 3D MODELS. IN INTERNATIONAL CONFERENCE ON THEORY AND PRACTICE OF DIGITAL LIBRARIES (PP. 209-220). SPRINGER, BERLIN, HEIDELBERG

MONTI, F., BOSCAINI, D., MASCI, J., RODOLA, E., SVOBODA, J., & BRONSTEIN, M. M. (2017). GEOMETRIC DEEP LEARNING ON GRAPHS AND MANIFOLDS USING MIXTURE MODEL CNNS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 5115-5124).

MONTEVERDE, L. C., RUIZ, C. R., & HUANG, Z. (2007, JANUARY). A SHAPE DISTRIBUTION FOR COMPARING 3D MODELS. IN INTERNATIONAL CONFERENCE ON MULTIMEDIA MODELING (PP. 54-63). SPRINGER, BERLIN, HEIDELBERG.

MORTARA, M., PATANE, G., SPAGNUOLO, M., FALCIDIENO, B., & ROSSIGNAC, J. (2004). BLOWING BUBBLES FOR MULTI-SCALE ANALYSIS AND DECOMPOSITION OF TRIANGLE MESHES. ALGORITHMICA, 38(1), 227-248.

MORTARA, M., PATANE, G., SPAGNUOLO, M., FALCIDIENO, B., & ROSSIGNAC, J. (2004). PLUMBER: A MULTI-SCALE DECOMPOSITION OF 3D SHAPES INTO TUBULAR PRIMITIVES AND BODIES.

MOUMOUN, L., CHAHHOU, M., GADI, T., & BENSLIMANE, R. (2011, APRIL). COMPARING BETWEEN DATA MINING ALGORITHMS:" CLOSE+, APRIORI AND CHARM" AND "KMEANS CLASSIFICATION ALGORITHM" AND APPLYING THEM ON 3D OBJECT INDEXING. IN 2011 INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS (PP. 1-6). IEEE.

NG, Y. T., HUANG, C. M., LI, Q. T., & TIAN, J. (2020). RADIALNET: A POINT CLOUD CLASSIFICATION APPROACH USING LOCAL STRUCTURE REPRESENTATION WITH RADIAL BASIS FUNCTION. SIGNAL, IMAGE AND VIDEO PROCESSING, 14(4), 747-752

NIEPERT, M., AHMED, M., & KUTZKOV, K. (2016, JUNE). LEARNING CONVOLUTIONAL NEURAL NETWORKS FOR GRAPHS. IN INTERNATIONAL CONFERENCE ON MACHINE LEARNING (PP. 2014-2023).

NOH, H., HONG, S., & HAN, B. (2015). LEARNING DECONVOLUTION NETWORK FOR SEMANTIC SEGMENTATION. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (PP. 1520-1528).

NOVOTNI, M., & KLEIN, R. (2001, MAY). A GEOMETRIC APPROACH TO 3D OBJECT COMPARISON. IN PROCEEDINGS INTERNATIONAL CONFERENCE ON SHAPE MODELING AND APPLICATIONS (PP. 167-175). IEEE.

NOVOTNI, M., & KLEIN, R. (2003, JUNE). 3D ZERNIKE DESCRIPTORS FOR CONTENT BASED SHAPE RETRIEVAL. IN PROCEEDINGS OF THE EIGHTH ACM SYMPOSIUM ON SOLID MODELING AND APPLICATIONS (PP. 216-225).

NOWAK, A., VILLAR, S., BANDEIRA, A. S., & BRUNA, J. (2017). REVISED NOTE ON LEARNING ALGORITHMS FOR QUADRATIC ASSIGNMENT WITH GRAPH NEURAL NETWORKS. ARXIV, ARXIV-1706.

OBENSHAIN, M. K. MAT, 2004, APPLICATION OF DATA MINING TECHNIQUES TO HEALTHCARE DATA. INFECTION CONTROL AND HOSPITAL EPIDEMIOLOGY, 25(8).

OHBUCHI, R., OTAGIRI, T., IBATO, M., & TAKEI, T. (2002, OCTOBER). SHAPE-SIMILARITY SEARCH OF THREE-DIMENSIONAL MODELS USING PARAMETERIZED STATISTICS. IN 10TH PACIFIC CONFERENCE ON COMPUTER GRAPHICS AND APPLICATIONS, 2002. PROCEEDINGS. (PP. 265-274). IEEE

OHBUCHI, R., & TAKEI, T. (2003, OCTOBER). SHAPE SIMILARITY COMPARISON OF 3D MODELS USING ALPHA SHAPES. IN 11TH PACIFIC CONFERENCE ONCOMPUTER GRAPHICS AND APPLICATIONS, 2003. PROCEEDINGS. (PP. 293-302). IEEE.

OHBUCHI, R., NAKAZAWA, M., & TAKEI, T. (2003, NOVEMBER). RETRIEVING 3D SHAPES BASED ON THEIR APPEARANCE. IN PROCEEDINGS OF THE 5TH ACM SIGMM INTERNATIONAL WORKSHOP ON MULTIMEDIA INFORMATION RETRIEVAL (PP. 39-45)

OHBUCHI, R., MINAMITANI, T., & TAKEI, T. (2003). SHAPE-SIMILARITY SEARCH OF 3D MODELS BY USING ENHANCED SHAPE FUNCTIONS. IN PROCEEDINGS OF THE THEORY AND PRACTICE OF COMPUTER GRAPHICS 2003 (P. 97).

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., & DOBKIN, D. (2001, MAY). MATCHING 3D MODELS WITH SHAPE DISTRIBUTIONS. IN PROCEEDINGS INTERNATIONAL CONFERENCE ON SHAPE MODELING AND APPLICATIONS (PP. 154-166). IEEE

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., & DOBKIN, D. (2002). SHAPE DISTRIBUTIONS. ACM TRANSACTIONS ON GRAPHICS (TOG), 21(4), 807-832.

PAQUET, E., RIOUX, M., MURCHING, A., NAVEEN, T., & TABATABAI, A. (2000). DESCRIPTION OF SHAPE INFORMATION FOR 2-D AND 3-D OBJECTS. SIGNAL PROCESSING: IMAGE COMMUNICATION, 16(1-2), 103-122..

PAQUET, E., & RIOUX, M. (1997, MAY). NEFERTITI: A QUERY BY CONTENT SOFTWARE FOR THREE-DIMENSIONAL MODELS DATABASES MANAGEMENT. IN PROCEEDINGS. INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN 3-D DIGITAL IMAGING AND MODELING (CAT. NO. 97TB100134) (PP. 345-352). IEEE.

PAQUET, E., & RIOUX, M. (1998, JUNE). A CONTENT-BASED SEARCH ENGINE FOR VRML DATABASES. IN PROCEEDINGS. 1998 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CAT. NO. 98CB36231) (PP. 541-546). IEEE.

PAULY, M., KEISER, R., & GROSS, M. (2003, SEPTEMBER). MULTI-SCALE FEATURE EXTRACTION ON POINT-SAMPLED SURFACES. IN COMPUTER GRAPHICS FORUM (VOL. 22, NO. 3, PP. 281-289). OXFORD, UK: BLACKWELL PUBLISHING, INC..

PAPADAKIS, P., PRATIKAKIS, I., THEOHARIS, T., & PERANTONIS, S. (2010). PANORAMA: A 3D SHAPE DESCRIPTOR BASED ON PANORAMIC VIEWS FOR UNSUPERVISED 3D OBJECT RETRIEVAL. INTERNATIONAL JOURNAL OF COMPUTER VISION, 89(2-3), 177-192.

PEARSON, K. (1901). LIII. ON LINES AND PLANES OF CLOSEST FIT TO SYSTEMS OF POINTS IN SPACE. THE LONDON, EDINBURGH, AND DUBLIN PHILOSOPHICAL MAGAZINE AND JOURNAL OF SCIENCE, 2(11), 559-572.

PELLEG, D., & MOORE, A. W. (2000, JUNE). X-MEANS: EXTENDING K-MEANS WITH EFFICIENT ESTIMATION OF THE NUMBER OF CLUSTERS. IN ICML (VOL. 1, PP. 727-734).

QI, C. R., SU, H., NIEßNER, M., DAI, A., YAN, M., & GUIBAS, L. J. (2016). VOLUMETRIC AND MULTI-VIEW CNNS FOR OBJECT CLASSIFICATION ON 3D DATA. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 5648-5656).

QI, C. R., SU, H., MO, K., & GUIBAS, L. J. (2017A). POINTNET: DEEP LEARNING ON POINT SETS FOR 3D CLASSIFICATION AND SEGMENTATION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 652-660).

QI, C. R., YI, L., SU, H., & GUIBAS, L. J. (2017B). POINTNET++: DEEP HIERARCHICAL FEATURE LEARNING ON POINT SETS IN A METRIC SPACE. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 5099-5108).

QUINLAN, J. R. (1986). INDUCTION OF DECISION TREES. MACHINE LEARNING 1. 1 (1), 81-106.

RAVANBAKHSH, S., SCHNEIDER, J., & POCZOS, B. (2016). DEEP LEARNING WITH SETS AND POINT CLOUDS. ARXIV PREPRINT ARXIV:1611.04500.

REEB, G. (1946). ON THE SINGULAR POINTS OF A COMPLETELY INTEGRABLE PFAFF FORM OR OF A NUMERICAL FUNCTION. COMPTES RANDUS ACAD. SCIENCES PARIS, 222, 847-849.

RICHARD, C. W., & HEMAMI, H. (1974). IDENTIFICATION OF THREE-DIMENSIONAL OBJECTS USING FOURIER DESCRIPTORS OF THE BOUNDARY CURVE. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, (4), 371-378.

ROSENBLATT, F. (1958). THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. PSYCHOLOGICAL REVIEW, 65(6), 386

ROSSIGNAC, J., & BORREL, P. (1993). MULTI-RESOLUTION 3D APPROXIMATIONS FOR RENDERING COMPLEX SCENES. IN MODELING IN COMPUTER GRAPHICS (PP. 455-465). SPRINGER, BERLIN, HEIDELBERG.

ROVERI, R., RAHMANN, L., OZTIRELI, C., & GROSS, M. (2018). A NETWORK ARCHITECTURE FOR POINT CLOUD CLASSIFICATION VIA AUTOMATIC DEPTH IMAGES GENERATION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 4176-4184).

RUI, Y., HUANG, T. S., ORTEGA, M., & MEHROTRA, S. (1998). RELEVANCE FEEDBACK: A POWER TOOL FOR INTERACTIVE CONTENT-BASED IMAGE RETRIEVAL. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 8(5), 644-655.

RUI, Y., HUANG, T. S., & CHANG, S. F. (1999). IMAGE RETRIEVAL: CURRENT TECHNIQUES, PROMISING DIRECTIONS, AND OPEN ISSUES. JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION, 10(1), 39-62.

RUBNER, Y., TOMASI, C., & GUIBAS, L. J. (1998, JANUARY). A METRIC FOR DISTRIBUTIONS WITH APPLICATIONS TO IMAGE DATABASES. IN SIXTH INTERNATIONAL CONFERENCE ON COMPUTER VISION (IEEE CAT. NO. 98CH36271) (PP. 59-66). IEEE.

RUBNER, Y., TOMASI, C., & GUIBAS, L. J. (2000). THE EARTH MOVER'S DISTANCE AS A METRIC FOR IMAGE RETRIEVAL. INTERNATIONAL JOURNAL OF COMPUTER VISION, 40(2), 99-121.

RUMELHART, D. E., HINTON, G. E., & WILLIAMS, R. J. (1985). LEARNING INTERNAL REPRESENTATIONS BY ERROR PROPAGATION (NO. ICS-8506). CALIFORNIA UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE.

RUSS, T. D., KOCH, M. W., & LITTLE, C. Q. (2005, SEPTEMBER). A 2D RANGE HAUSDORFF APPROACH FOR 3D FACE RECOGNITION. IN 2005 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'05)-WORKSHOPS (PP. 169-169). IEEE.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., ... & BERG, A. C. (2015). IMAGENET LARGE SCALE VISUAL RECOGNITION CHALLENGE. INTERNATIONAL JOURNAL OF COMPUTER VISION, 115(3), 211-252.

SAITO, S., LI, T., & LI, H. (2016, OCTOBER). REAL-TIME FACIAL SEGMENTATION AND PERFORMANCE CAPTURE FROM RGB INPUT. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 244-261). SPRINGER, CHAM.

SALVI, J., MATABOSCH, C., FOFI, D., & FOREST, J. (2007). A REVIEW OF RECENT RANGE IMAGE REGISTRATION METHODS WITH ACCURACY EVALUATION. IMAGE AND VISION COMPUTING, 25(5), 578-596

SAVASERE, A., OMIECINSKI, E. R., & NAVATHE, S. B. (1995). AN EFFICIENT ALGORITHM FOR MINING ASSOCIATION RULES IN LARGE DATABASES. GEORGIA INSTITUTE OF TECHNOLOGY

SAUPE, D., & VRANIĆ, D. V. (2001, SEPTEMBER). 3D MODEL RETRIEVAL WITH SPHERICAL HARMONICS AND MOMENTS. IN JOINT PATTERN RECOGNITION SYMPOSIUM (PP. 392-397). SPRINGER, BERLIN, HEIDELBERG.

SCARSELLI, F., GORI, M., TSOI, A. C., HAGENBUCHNER, M., & MONFARDINI, G. (2009). THE GRAPH NEURAL NETWORK MODEL. IEEE TRANSACTIONS ON NEURAL NETWORKS, 20(1), 61-80.

SCHWARZ, M., SCHULZ, H., & BEHNKE, S. (2015, MAY). RGB-D OBJECT RECOGNITION AND POSE ESTIMATION BASED ON PRE-TRAINED CONVOLUTIONAL NEURAL NETWORK FEATURES. IN 2015 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA) (PP. 1329-1335). IEEE.

SEDAGHAT, N., ZOLFAGHARI, M., AMIRI, E., & BROX, T. (2016). ORIENTATION-BOOSTED VOXEL NETS FOR 3D OBJECT RECOGNITION. ARXIV PREPRINT ARXIV:1604.0335

SERRA, J. (1982). IMAGE ANALYSIS AND MATHEMATICAL MORPHOL-OGY

SERMANET, P., EIGEN, D., ZHANG, X., MATHIEU, M., FERGUS, R., & LECUN, Y. (2013). OVERFEAT: INTEGRATED RECOGNITION, LOCALIZATION AND DETECTION USING CONVOLUTIONAL NETWORKS. ARXIV PREPRINT ARXIV:1312.6229

SFIKAS, K., PRATIKAKIS, I., & THEOHARIS, T. (2018). ENSEMBLE OF PANORAMA-BASED CONVOLUTIONAL NEURAL NETWORKS FOR 3D MODEL CLASSIFICATION AND RETRIEVAL. COMPUTERS & GRAPHICS, 71, 208-218.

SFIKAS, K., THEOHARIS, T., & PRATIKAKIS, I. (2017). EXPLOITING THE PANORAMA REPRESENTATION FOR CONVOLUTIONAL NEURAL NETWORK CLASSIFICATION AND RETRIEVAL. 3DOR, 6, 7.

SHARIR, M., & SCHORR, A. (1986). ON SHORTEST PATHS IN POLYHEDRAL SPACES. SIAM JOURNAL ON COMPUTING, 15(1), 193-215

SHARMA, A., GRAU, O., & FRITZ, M. (2016, OCTOBER). VCONV-DAE: DEEP VOLUMETRIC SHAPE LEARNING WITHOUT OBJECT LABELS. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 236-250). SPRINGER, CHAM

SHI, B., BAI, S., ZHOU, Z., & BAI, X. (2015). DEEPPANO: DEEP PANORAMIC REPRESENTATION FOR 3-D SHAPE RECOGNITION. IEEE SIGNAL PROCESSING LETTERS, 22(12), 2339-2343.

SHI, B., BAI, S., ZHOU, Z., & BAI, X. (2015). DEEPPANO: DEEP PANORAMIC REPRESENTATION FOR 3-D SHAPE RECOGNITION. IEEE SIGNAL PROCESSING LETTERS, 22(12), 2339-2343.

SHIH, J. L., LEE, C. H., & WANG, J. T. (2005). 3D OBJECT RETRIEVAL SYSTEM BASED ON GRID D2. ELECTRONICS LETTERS, 41(4), 179-181

SHILANE, P., MIN, P., KAZHDAN, M., & FUNKHOUSER, T. (2004, JUNE). THE PRINCETON SHAPE BENCHMARK. IN PROCEEDINGS SHAPE MODELING APPLICATIONS, 2004. (PP. 167-178). IEEE.

SHIPP, M. A., ROSS, K. N., TAMAYO, P., WENG, A. P., KUTOK, J. L., AGUIAR, R. C., ... & RAY, T. S. (2002). DIFFUSE LARGE B-CELL LYMPHOMA OUTCOME PREDICTION BY GENE-EXPRESSION PROFILING AND SUPERVISED MACHINE LEARNING. NATURE MEDICINE, 8(1), 68-74.

SHOKOUFANDEH, A., DICKINSON, S. J., SIDDIQI, K., & ZUCKER, S. W. (1999, JUNE). INDEXING USING A SPECTRAL ENCODING OF TOPOLOGICAL STRUCTURE. IN PROCEEDINGS. 1999 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CAT. NO PR00149) (VOL. 2, PP. 491-497). IEEE.

SHOKOUFANDEH, A., DICKINSON, S., JÖNSSON, C., BRETZNER, L., & LINDEBERG, T. (2002, MAY). ON THE REPRESENTATION AND MATCHING OF QUALITATIVE SHAPE AT MULTIPLE SCALES. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 759-775). SPRINGER, BERLIN, HEIDELBERG.

SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S. J., & ZUCKER, S. W. (1999). SHOCK GRAPHS AND SHAPE MATCHING. INTERNATIONAL JOURNAL OF COMPUTER VISION, 35(1), 13-32

SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S. J., & ZUCKER, S. W. (1998). SHOCK GRAPHS AND SHAPE MATCHING.

SIJBERS, J., & VAN DYCK, D. (2002, JUNE). EFFICIENT ALGORITHM FOR THE COMPUTATION OF 3D FOURIER DESCRIPTORS. IN 3D DATA PROCESSING VISUALIZATION AND TRANSMISSION, INTERNATIONAL SYMPOSIUM ON (PP. 640-640). IEEE COMPUTER SOCIETY

SIMONYAN, K., & ZISSERMAN, A. (2014). VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. ARXIV PREPRINT ARXIV:1409.1556

SIMONOVSKY, M., & KOMODAKIS, N. (2017). DYNAMIC EDGE-CONDITIONED FILTERS IN CONVOLUTIONAL NEURAL NETWORKS ON GRAPHS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 3693-3702).

SINHA, A., BAI, J., & RAMANI, K. (2016, OCTOBER). DEEP LEARNING 3D SHAPE SURFACES USING GEOMETRY IMAGES. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 223-240). SPRINGER, CHAM.

SINGH, C. (2011). IMPROVING IMAGE RETRIEVAL USING COMBINED FEATURES OF HOUGH TRANSFORM AND ZERNIKE MOMENTS. OPTICS AND LASERS IN ENGINEERING, 49(12), 1384-1396

SOCHER, R., HUVAL, B., BATH, B., MANNING, C. D., & NG, A. Y. (2012). CONVOLUTIONAL-RECURSIVE DEEP LEARNING FOR 3D OBJECT CLASSIFICATION. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 656-664

SONG, S., & XIAO, J. (2016). DEEP SLIDING SHAPES FOR AMODAL 3D OBJECT DETECTION IN RGB-D IMAGES. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 808-816).

SONG, S., YU, F., ZENG, A., CHANG, A. X., SAVVA, M., & FUNKHOUSER, T. (2017). SEMANTIC SCENE COMPLETION FROM A SINGLE DEPTH IMAGE. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 1746-1754).

SONG, J. J., & GOLSHANI, F. (2003, NOVEMBER). SHAPE-BASED 3D MODEL RETRIEVAL. IN PROCEEDINGS. 15TH IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE (PP. 636-640). IEEE.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., & SALAKHUTDINOV, R. (2014). DROPOUT: A SIMPLE WAY TO PREVENT NEURAL NETWORKS FROM OVERFITTING. THE JOURNAL OF MACHINE LEARNING RESEARCH, 15(1), 1929-1958

SU, H., MAJI, S., KALOGERAKIS, E., & LEARNED-MILLER, E. (2015). MULTI-VIEW CONVOLUTIONAL NEURAL NETWORKS FOR 3D SHAPE RECOGNITION. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (PP. 945-953).

SUBRAMANYA, R., & FALOUTSOS, C. (1998). MINDREADER: QUERYING DATABASES THROUGH MULTIPLE EXAMPLES. IN PROC. VERY LARGE DATA BASES CONF

SUN, J., OVSJANIKOV, M., & GUIBAS, L. (2009, JULY). A CONCISE AND PROVABLY INFORMATIVE MULTI-SCALE SIGNATURE BASED ON HEAT DIFFUSION. IN COMPUTER GRAPHICS FORUM (VOL. 28, NO. 5, PP. 1383-1392). OXFORD, UK: BLACKWELL PUBLISHING LTD

SUNDAR, H., SILVER, D., GAGVANI, N., & DICKINSON, S. (2003, MAY). SKELETON BASED SHAPE MATCHING AND RETRIEVAL. IN 2003 SHAPE MODELING INTERNATIONAL. (PP. 130-139). IEEE.

SUTSKEVER, I., MARTENS, J., DAHL, G., & HINTON, G. (2013, FEBRUARY). ON THE IMPORTANCE OF INITIALIZATION AND MOMENTUM IN DEEP LEARNING. IN INTERNATIONAL CONFERENCE ON MACHINE LEARNING (PP. 1139-1147).

SUZUKI, M. T., KATO, T., & OTSU, N. (2000, OCTOBER). A SIMILARITY RETRIEVAL OF 3D POLYGONAL MODELS USING ROTATION INVARIANT SHAPE DESCRIPTORS. IN SMC 2000 CONFERENCE PROCEEDINGS. 2000 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS.'CYBERNETICS EVOLVING TO SYSTEMS, HUMANS, ORGANIZATIONS, AND THEIR COMPLEX INTERACTIONS'(CAT. NO. 0 (VOL. 4, PP. 2946-2952). IEEE.

TAN, P. N., STEINBACH, M., & KUMAR, V. (2006). INTRODUCTION TO DATA MINING, PEARSON EDUCATION. INC., NEW DELHI.

TAN, C. W., & KUMAR, A. (2014). ACCURATE IRIS RECOGNITION AT A DISTANCE USING STABILIZED IRIS ENCODING AND ZERNIKE MOMENTS PHASE FEATURES. IEEE TRANSACTIONS ON IMAGE PROCESSING, 23(9), 3962-3974.

TANGELDER, J. W., & VELTKAMP, R. C. (2003). POLYHEDRAL MODEL RETRIEVAL USING WEIGHTED POINT SETS. INTERNATIONAL JOURNAL OF IMAGE AND GRAPHICS, 3(01), 209-229

TAL, A., & ZUCKERBERGER, E. (2006, FEBRUARY). MESH RETRIEVAL BY COMPONENTS. IN INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS THEORY AND APPLICATIONS (VOL. 2, PP. 142-149). SCITEPRESS.

TAYBI, I. O., ALAOUI, R., ZAKANI, F. R., ARHID, K., BOUKSIM, M., & GADI, T. (2016). A NOVEL EFFICIENT 3D OBJECT RETRIEVAL METHOD BASED ON

REPRESENTATIVE SLICES. IN 2016 5TH INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS (ICMCS) (PP. 639-644). IEEE.

TAYBI, I. O., BOUKSIM, M., ALAOUI, R., & GADI, T. (2019). A NOVEL PARTIAL 3D OBJECT RETRIEVAL METHOD USING ADAPTIVE SLICES CLUSTERING. INTERNATIONAL JOURNAL OF INTELLIGENT ENGINEERING AND SYSTEMS, 12(1).

TAYBI, I. O., GADI, T., & ALAOUI, R. (2020). A NEW PARTIAL 3D OBJECT INDEXING AND RETRIEVAL APPROACH COMBINING 2D SLICES AND APRIORI ALGORITHM. SCIENTIFIC VISUALIZATION, 12(2).

TEAGUE, M. R. (1980). IMAGE ANALYSIS VIA THE GENERAL THEORY OF MOMENTS. JOSA, 70(8), 920-930.

TEH, C. H., & CHIN, R. T. (1988). ON IMAGE ANALYSIS BY THE METHODS OF MOMENTS. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 10(4), 496-513.

TOMAS, A., & ERIC, H. (2002). REAL-TIME RENDERING. (2ND ED.). A K PETERS, LTD., 2002, PP.564-567.

TUNG, T., & SCHMITT, F. (2005). THE AUGMENTED MULTIRESOLUTION REEB GRAPH APPROACH FOR CONTENT-BASED RETRIEVAL OF 3D SHAPES. INTERNATIONAL JOURNAL OF SHAPE MODELING, 11(01), 91-120

VAPNIK, V. N. (1999). THE NATURE OF STATISTICAL LEARNING THEORY

VELTKAMP, R. C., & HAGEDOORN, M. (2000, NOVEMBER). SHAPE SIMILARITY MEASURES, PROPERTIES AND CONSTRUCTIONS. IN INTERNATIONAL CONFERENCE ON ADVANCES IN VISUAL INFORMATION SYSTEMS (PP. 467-476). SPRINGER, BERLIN, HEIDELBERG.

VENKATAKRISHNAN, S. B., ALIZADEH, M., & VISWANATH, P. (2018). GRAPH2SEQ: SCALABLE LEARNING DYNAMICS FOR GRAPHS. ARXIV PREPRINT ARXIV:1802.04948.

VINYALS, O., BENGIO, S., & KUDLUR, M. (2015). ORDER MATTERS: SEQUENCE TO SEQUENCE FOR SETS. ARXIV PREPRINT ARXIV:1511.06391.

VRANIC, D., & SAUPE, D. (2001A). 3D SHAPE DESCRIPTOR BASED ON 3D FOURIER TRANSFORM. IN EURASIP (PP. 271-274).

VRANIC, D. V., SAUPE, D., & RICHTER, J. (2001B, OCTOBER). TOOLS FOR 3D-OBJECT RETRIEVAL: KARHUNEN-LOEVE TRANSFORM AND SPHERICAL HARMONICS. IN 2001 IEEE FOURTH WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING (CAT. NO. 01TH8564) (PP. 293-298). IEEE.

VRANIC, D., & SAUPE, D. (2000). 3D MODEL RETRIEVAL. IN SPRING CONF. COMPUT. GRAPH.(SCCG 2000).

VRANIC, D. V., & SAUPE, D. (2004). 3D MODEL RETRIEVAL (DOCTORAL DISSERTATION, UNIVERSITY OF LEIPZIG).

VRANIC, D. V. (2003, SEPTEMBER). AN IMPROVEMENT OF ROTATION INVARIANT 3D-SHAPE BASED ON FUNCTIONS ON CONCENTRIC SPHERES. IN PROCEEDINGS 2003 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (CAT. NO. 03CH37429) (VOL. 3, PP. III-757). IEEE.

WANG, C., SAMARI, B., & SIDDIQI, K. (2018). LOCAL SPECTRAL GRAPH CONVOLUTION FOR POINT SET FEATURE LEARNING. ARXIV PREPRINT ARXIV:1803.05827.

WANG, C., PELILLO, M., & SIDDIQI, K. (2019). DOMINANT SET CLUSTERING AND POOLING FOR MULTI-VIEW 3D OBJECT RECOGNITION. ARXIV PREPRINT ARXIV:1906.01592

WARD JR, J. H. (1963). HIERARCHICAL GROUPING TO OPTIMIZE AN OBJECTIVE FUNCTION. JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION, 58(301), 236-244.

WATKINS, C. J., & DAYAN, P. (1992). Q-LEARNING. MACHINE LEARNING, 8(3-4), 279-292.

WEE, C. Y., PARAMESRAN, R., & TAKEDA, F. (2007, SEPTEMBER). FAST COMPUTATION OF ZERNIKE MOMENTS FOR RICE SORTING SYSTEM. IN 2007 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (VOL. 6, PP. VI-165). IEEE.

WU, Z., SONG, S., KHOSLA, A., YU, F., ZHANG, L., TANG, X., & XIAO, J. (2015). 3D SHAPENETS: A DEEP REPRESENTATION FOR VOLUMETRIC SHAPES. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 1912-1920).

WU, J., ZHANG, C., XUE, T., FREEMAN, B., & TENENBAUM, J. (2016). LEARNING A PROBABILISTIC LATENT SPACE OF OBJECT SHAPES VIA 3D GENERATIVE-ADVERSARIAL MODELING. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 82-90).

XIE, S., & TU, Z. (2015). HOLISTICALLY-NESTED EDGE DETECTION. IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (PP. 1395-1403).

XIE, J., ZHENG, Z., GAO, R., WANG, W., ZHU, S. C., & NIAN WU, Y. (2018). LEARNING DESCRIPTOR NETWORKS FOR 3D SHAPE SYNTHESIS AND ANALYSIS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 8629-8638).

XU, R., & WUNSCH, D. (2005). SURVEY OF CLUSTERING ALGORITHMS. IEEE TRANSACTIONS ON NEURAL NETWORKS, 16(3), 645-678

XU, X., & TODOROVIC, S. (2016, DECEMBER). BEAM SEARCH FOR LEARNING A DEEP CONVOLUTIONAL NEURAL NETWORK OF 3D SHAPES. IN 2016 23RD INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR) (PP. 3506-3511). IEEE.

YANG, Y., LIN, H., & ZHANG, Y. (2007). CONTENT-BASED 3-D MODEL RETRIEVAL: A SURVEY. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C (APPLICATIONS AND REVIEWS), 37(6), 1081-1098

YANG, Y., FENG, C., SHEN, Y., & TIAN, D. (2018). FOLDINGNET: POINT CLOUD AUTO-ENCODER VIA DEEP GRID DEFORMATION. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 206-215).

YIN, L., WEI, X., SUN, Y., WANG, J., & ROSATO, M. J. (2006, APRIL). A 3D FACIAL EXPRESSION DATABASE FOR FACIAL BEHAVIOR RESEARCH. IN 7TH INTERNATIONAL CONFERENCE ON AUTOMATIC FACE AND GESTURE RECOGNITION (FGR06) (PP. 211-216). IEEE.

YIN, L., SUN, Y., WORM, T., & REALE, M. A HIGH-RESOLUTION 3D DYNAMIC FACIAL EXPRESSION DATABASE 2008. IN IEEE INTERNATIONAL CONFERENCE ON AUTOMATIC FACE AND GESTURE RECOGNITION (VOL. 126).

YU, M., ATMOSUKARTO, I., LEOW, W. K., HUANG, Z., & XU, R. (2003, JUNE). 3D MODEL RETRIEVAL WITH MORPHING-BASED GEOMETRIC AND TOPOLOGICAL FEATURE MAPS. IN 2003 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2003. PROCEEDINGS. (VOL. 2, PP. II-656). IEEE

YU, F., & KOLTUN, V. (2015). MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS. ARXIV PREPRINT ARXIV:1511.07122.

XIE, Z., XU, K., SHAN, W., LIU, L., XIONG, Y., & HUANG, H. (2015, OCTOBER). PROJECTIVE FEATURE LEARNING FOR 3D SHAPES WITH MULTI-VIEW DEPTH IMAGES. IN COMPUTER GRAPHICS FORUM (VOL. 34, NO. 7, PP. 1-11).

XU, X., & TODOROVIC, S. (2016, DECEMBER). BEAM SEARCH FOR LEARNING A DEEP CONVOLUTIONAL NEURAL NETWORK OF 3D SHAPES. IN 2016 23RD INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (ICPR) (PP. 3506-3511). IEEE

ZAHARIA, T., & PRÊTEUX, F. (2002, AUGUST). SHAPE-BASED RETRIEVAL OF 3D MESH MODELS. IN PROCEEDINGS. IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (VOL. 1, PP. 437-440). IEEE.

ZAHEER, M., KOTTUR, S., RAVANBAKHSH, S., POCZOS, B., SALAKHUTDINOV, R. R., & SMOLA, A. J. (2017). DEEP SETS. IN ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS (PP. 3391-3401).

ZAKI, M. J., PARTHASARATHY, S., OGIHARA, M., & LI, W. (1997). PARALLEL ALGORITHMS FOR DISCOVERY OF ASSOCIATION RULES. DATA MINING AND KNOWLEDGE DISCOVERY, 1(4), 343-373

ZANUTTIGH, P., & MINTO, L. (2017, SEPTEMBER). DEEP LEARNING FOR 3D SHAPE CLASSIFICATION FROM MULTIPLE DEPTH MAPS. IN 2017 IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP) (PP. 3615-3619). IEEE.

ZEILER, M. D., & FERGUS, R. (2014, SEPTEMBER). VISUALIZING AND UNDERSTANDING CONVOLUTIONAL NETWORKS. IN EUROPEAN CONFERENCE ON COMPUTER VISION (PP. 818-833). SPRINGER, CHAM.

ZHANG, C., & CHEN, T. (2001, OCTOBER). EFFICIENT FEATURE EXTRACTION FOR 2D/3D OBJECTS IN MESH REPRESENTATION. IN PROCEEDINGS 2001 INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (CAT. NO. 01CH37205) (VOL. 3, PP. 935-938). IEEE.

ZHANG, C., & CHEN, T. (2001). ACTIVE LEARNING FOR INFORMATION RETRIEVAL: USING 3D MODELS AS AN EXAMPLE. CARNEGIE MELLON TECHNICAL REPORT: AMP01-04.

ZHANG, D., & LU, G. (2004). REVIEW OF SHAPE REPRESENTATION AND DESCRIPTION TECHNIQUES. PATTERN RECOGNITION, 37(1), 1-19.

ZHANG, H., & FIUME, E. (2002, MAY). SHAPE MATCHING OF 3D CONTOURS USING NORMALIZED FOURIER DESCRIPTORS. IN PROCEEDINGS SMI. SHAPE MODELING INTERNATIONAL 2002 (PP. 261-268). IEEE.

ZHANG, L., XIANG, F., PU, J., & ZHANG, Z. (2012, AUGUST). APPLICATION OF IMPROVED HU MOMENTS IN OBJECT RECOGNITION. IN 2012 IEEE INTERNATIONAL CONFERENCE ON AUTOMATION AND LOGISTICS (PP. 554-558). IEEE.

ZHANG, Y., KOSCHAN, A., & ABIDI, M. A. (2003, MAY). SUPERQUADRICS-BASED 3D OBJECT REPRESENTATION OF AUTOMOTIVE PARTS UTILIZING PART DECOMPOSITION. IN SIXTH INTERNATIONAL CONFERENCE ON QUALITY CONTROL BY ARTIFICIAL VISION (VOL. 5132, PP. 241-251). INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS.

ZHANG, X., YIN, L., COHN, J. F., CANAVAN, S., REALE, M., HOROWITZ, A., ... & GIRARD, J. M. (2014). BP4D-SPONTANEOUS: A HIGH-RESOLUTION SPONTANEOUS 3D DYNAMIC FACIAL EXPRESSION DATABASE. IMAGE AND VISION COMPUTING, 32(10), 692-706

ZHANG, Z., GIRARD, J. M., WU, Y., ZHANG, X., LIU, P., CIFTCI, U., ... & COHN, J. F. (2016). MULTIMODAL SPONTANEOUS EMOTION CORPUS FOR HUMAN BEHAVIOR ANALYSIS. IN PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (PP. 3438-3446).

ZHAO, S., YAO, H., ZHANG, Y., WANG, Y., & LIU, S. (2015). VIEW-BASED 3D OBJECT RETRIEVAL VIA MULTI-MODAL GRAPH LEARNING. SIGNAL PROCESSING, 112, 110-118.

ZHENG, Z., KOHAVI, R., & MASON, L. (2001, AUGUST). REAL WORLD PERFORMANCE OF ASSOCIATION RULE ALGORITHMS. IN PROCEEDINGS OF THE SEVENTH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (PP. 401-406).

ZHI, S., LIU, Y., LI, X., & GUO, Y. (2018). TOWARD REAL-TIME 3D OBJECT RECOGNITION: A LIGHTWEIGHT VOLUMETRIC CNN FRAMEWORK USING MULTITASK LEARNING. COMPUTERS & GRAPHICS, 71, 199-207.

ZHU, Z., WANG, X., BAI, S., YAO, C., & BAI, X. (2016). DEEP LEARNING REPRESENTATION USING AUTOENCODER FOR 3D SHAPE RETRIEVAL. NEUROCOMPUTING, 204, 41-50

ŽUNIĆ, D., & ŽUNIĆ, J. (2014). SHAPE ELLIPTICITY FROM HU MOMENT INVARIANTS. APPLIED MATHEMATICS AND COMPUTATION, 226, 406-414.

Liste des contributions

Articles de journal

ILYASS OUAZZANI TAYBI, RACHID ALAOUI, MOHCINE BOUKSIM, TAOUFIQ GADI. (2019). "A NOVEL PARTIAL 3D OBJECT RETRIEVAL METHOD USING ADAPTIVE SLICES CLUSTERING," IN INTERNATIONAL JOURNAL OF INTELLIGENT ENGINEERING AND SYSTEMS, 12(1).

ILYASS OUAZZANI TAYBI, RACHID ALAOUI, TAOUFIQ GADI. (2020), "A NEW PARTIAL 3D OBJECT INDEXING AND RETRIEVAL APPROACH COMBINING 2D SLICES AND APRIORI ALGORITHM," IN SCIENTIFIC VISUALIZATION, 12(2).

ILYASS OUAZZANI TAYBI, TAOUFIQ GADI, RACHID ALAOUI, "2DSLICESNET: A 2D SLICE-BASED CONVOLUTIONAL NEURAL NETWORK FOR 3D OBJECT RETRIEVAL AND CLASSIFICATION," IN IEEE ACCESS, VOL. 9, PP. 24041-24049, 2021, DOI: 10.1109/ACCESS.2021.3056613.

Article de conférence internationale

ILYASS OUAZZANI TAYBI, RACHID ALAOUI, FATIMA RAFII ZAKANI, KHADIJA ARHID, MOHCINE BOUKSIM, TAOUFIQ GADI. (2016). A NOVEL EFFICIENT 3D OBJECT RETRIEVAL METHOD BASED ON REPRESENTATIVE SLICES. IN 2016 5TH INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS (ICMCS) (PP. 639-644). IEEE.

Communications de conférences nationales

ILYASS OUAZZANI TAYBI, RACHID ALAOUI, TAOUFIQ GADI. (MARS 2017). RECONNAISSANCE D'OBJETS 3D BASEE SUR LES SIMILARITES LOCALES. LA 3EME EDITION DES JOURNEES SCIENTIFIQUES NATIONALES (3'EJSN) DE L'ASSOCIATION DES ETUDIANTS CHERCHEURS DE LA FACULTE DES SCIENCES D'AGADIR.

ILYASS OUAZZANI TAYBI, RACHID ALAOUI, TAOUFIQ GADI. (MAI 2016). NORMALISATION ET INDEXATION D'OBJETS 3D. JOURNEES SCIENTIFIQUES, ENSA AGADIR.