كلية العلوم و التقنيات فاس
ⵜⴰⵖⵙⵍⵓⵜ ⵏ ⵜⵎⴰⵡⴰⵙⵉⵏ ⴷ ⵜⵙⵙⵜⵉⵏ
Faculté des Sciences et Techniques de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵡⴰⵍⵓⵜ ⵙⵙⴷ ⵎⵓⵃⴰⵎⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

N° d'ordre 34/2019

THESE DE DOCTORAT

Présentée par

Mme : Safae El Houfi

Spécialité : Informatique

Sujet de la thèse : Contribution à la reconnaissance des Objets 3D

Thèse présentée et soutenue le mardi 09 juillet 2019 à 10h du matin au centre de conférence devant le jury composé de :

Nom Prénom	Titre	Etablissement	
Najia Es-SBAI	PES	Faculté des Sciences et Techniques de Fès	Présidente
Brahim AKSASSE	PES	Faculté des Sciences et Techniques d'Errachidia	Rapporteur
Fatima GUEROUATE	PH	Ecole Supérieure de Technologie de Salé	Rapporteur
Khalid ZENKOUAR	PH	Faculté des Sciences et Techniques de Fès	Rapporteur
Arsalane ZARGHILI	PES	Faculté des Sciences et Techniques de Fès	Examinateur
Khalid ABBAD	PH	Faculté des Sciences et Techniques de Fès	Examinateur
Majda Aicha	PH	Faculté des Sciences et Techniques de Fès	Directrice de thèse

Laboratoire d'accueil : Laboratoire Systèmes Intelligents et Application (SIA).

Etablissement : Faculté des Sciences et Techniques (FST)-Fès.

A mes êtres les plus chers, en guise de gratitude, pour leurs amour et sacrifices : mes parents, mon mari, qui n'a jamais cessé de m'encourager, mon frère, mes sœurs, mes amis, et à tous les gens qui ont été là pour moi.

Remerciements

Je dédie ce travail de thèse à mes parents, Mme Amina Mdaraa et M. Mimoun El houfi, pour leur remercier d'être toujours aussi fière de moi. Cette thèse représente beaucoup pour moi et je suis donc d'autant plus heureuse de pouvoir la dédier à eux.

Tout n'a pas toujours été facile pendant ces années de doctorat, et je tiens donc à exprimer ma profonde gratitude à ma directrice de thèse Mme. Aicha Majda, sans qui je n'aurais jamais réussi à faire ce travail de thèse. Je la remercie pour sa gentillesse, son soutien, ses conseils, sa confiance et toute l'attention qu'elle m'a porté tout au long de ce travail. Je la remercie également pour sa disponibilité, sa compréhension et sa relecture minutieuse du rapport.

Je tiens aussi à remercier les membres de mon jury, d'avoir accepté de s'intéresser à nos travaux. Un grand merci à :

- *Mme Es-sbai Najia, de m'avoir fait l'honneur d'accepter de présider mon jury de thèse.,*
- *MM. les professeurs Brahim AKSASSE, Khalid ZENKOUAR et Mme Fatima GUEROUATE d'avoir accepté de rapporter ce travail. Je tiens d'ailleurs à leur témoigner toute ma gratitude pour leur gentillesse et pour la pertinence de leurs remarques. J'aimerais remercier de plus monsieur Brahim AKSASSE et madame Fatima GUEROUATE pour leurs éventuels déplacement afin d'être présent parmi les membres du jury.*
- *MM. Arsalane ZARGHILI, Khalid ABBAD, d'avoir accepté d'être examinateurs de ma thèse et pour leurs encouragements.*

Je souhaite par ailleurs remercier laboratoire système intelligent et applications (LSIA) grâce à lequel j'ai pu bénéficier de très bonnes conditions de travail, Enfin, je remercie énormément ma famille et surtout mon mari Zakariae El atmani qui m'a encouragé et soutenue pendant ces dernières années. Enfin toute ma sympathie à mes collègues et à mes chers amis.

Résumé

De nos jours, les modèles 3D jouent un rôle important dans de nombreuses applications. En raison de la croissance rapide du nombre de modèles 3D, la compréhension, la classification et la reconnaissance de tels modèles sont devenues des domaines intéressants de la vision par ordinateur. Dans ce contexte, notre objectif est de reconnaitre un objet 3D, Soit à partir d'une base d'apprentissage d'objets 3D, où bien d'une scène 2D contenant plusieurs objets. Notre idée est de combiner des aspects d'approches existantes et d'apporter une amélioration des performances de la reconnaissance et de la segmentation sémantique.

Cette thèse sera divisée en deux grands axes: dans le premier axe, nous nous sommes penchés sur la classification et la reconnaissance des objets 3D, où nous présentons deux approches différentes. La première approche est basée sur l'indexation 2D/3D. Elle caractérise la forme de l'objet 3D à l'aide des projections 2D. Ce type d'approches nécessite de mettre en place trois traitements distincts à savoir: l'extraction des vues, la description pertinente de leur forme et le processus de classification qui doit permettre de répondre aux requêtes de l'utilisateur. Pour ce dernier traitement, nous avons utilisé l'algorithme de reconnaissance de nuages de points $P. cet algorithme nous offre deux possibilités de recherche, soit directement avec une image requête soit avec un autre objet 3D en comparant leurs ensembles de vues. La deuxième approche agit directement sur la forme du modèle 3D (indexation 3D/3D). L'idée clé est qu'à partir d'une fonction de forme mesurant les propriétés géométriques globales d'un objet, nous représentons sa signature sous forme d'une distribution. En effet, la stratégie proposée est la suivante: à partir d'un objet polygone à reconnaître, une triangulation est effectuée. Ensuite, les distances sont calculées entre deux points aléatoires de la surface de l'objet 3D. Puis, la distribution de ces distances sera représentée par un histogramme normalisé. Finalement, les valeurs de ces histogrammes alimentent un réseau de neurones multicouches.

Dans le second axe, nous nous sommes intéressés à la segmentation sémantique d'une scène 2D contenant plusieurs objets. Le défi est d'effectuer une segmentation sémantique en temps réel tout en garantissant un compromis remarquable en termes de précision et de rapidité. Pour cela, nous avons développé un nouveau module basé sur des techniques récentes telles que les convolutions séparables profondes et la convolution dilatée. Ensuite, nous avons choisi d'utiliser la technique dense connectivité, afin de rassembler les caractéristiques extraites de différentes couches et de réduire notablement le nombre de paramètres.

Des résultats quantitatifs et qualitatifs sont présentés tout au long de ce manuscrit. Sur certains aspects de la reconnaissance d'objet et de la segmentation des scènes.

Mots clés : Objet 3D, VRML, Indexation 2D/3D et 3D/3D, Reconnaissance, Classification, Descripteur de forme, $P Recognizer, Segmentation sémantique, Réseaux de neurones convolutifs.

Abstract

Nowadays, 3D models play an important role in many applications. Due to the rapid growth in the number of 3D models, understanding, classifying and recognizing such models have become interesting areas for computer vision. In this context, our goal is to recognize a 3D object, either from a 3D objects learning base, or from a 2D scene containing several objects. Our idea is to combine aspects of existing approaches and to improve the performances of semantic recognition and segmentation.

This thesis will be divided into two main axis: firstly, we focused on the classification and recognition of 3D objects. Where we present two different approaches: one is based on 2D / 3D indexing. It characterizes the shape of the 3D object using 2D projections. This type of approach requires the establishment of three distinct treatments, namely: The views extraction, the relevant description of their form and the classification process that allows to answer the user requests. For this last treatment, we used the point cloud recognition algorithm $P. This algorithm offers us two search possibilities, either directly with a query image or with another 3D object by comparing their sets of views. The second approach acts directly on the shape of the 3D model (3D / 3D indexing). The key idea is that from a shape function measuring the global geometric properties of an object, we represent its signature as a distribution. Indeed, the proposed strategy is as follows: from a polygon object to be recognized, a triangulation is performed. Then, the distances are calculated between two random points of the surface of the 3D object. Then, the distribution of these distances will be represented by a normalized histogram. Finally, the values of these histograms feed a network of multilayer neurons.

In the second axis, we were interested in the semantic segmentation of a 2D scene containing several objects. The challenge is to perform a real time semantic segmentation while maintaining a good accuracy. For that, we have developed a new module based on recent techniques such as deep separable convolutions and dilated convolution. Then, we have chosen to use the dense connectivity technique, in order to gather the characteristics extracted from different layers and significantly reduce the number of parameters.

Quantitative and qualitative results are presented throughout this manuscript on some aspects of object recognition and segmentation of scenes.

Keywords: 3D Object, VRML, 2D/3D and 3D/3D Indexing, Recognition, Classification, Shape Descriptor, $P Recognizer, Semantic Segmentation, Convolutional Neural Networks.

Publications

Revues internationales:

- M. Jazouli, S.Elhoufi and A.Majda « ***Stereotypical Motor Movement Recognition Using Microsoft Kinect with Artificial Neural Network*** » World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:10, No:7, P 1270-1274, 2016

- Safae Elhoufi, Aicha Majda, Khalid Abbad « ***D2 Shape distribution and artificial neural networks for 3D objects recognition*** » International Journal of Engineering & Technology, 7 (2.13) (2018) 103-108

- Safae Elhoufi, Maha Jazouli, Aicha Majda, Arsalane Zarghili « ***A 3-D classification based on $P Recogniser*** » International Journal of Computational Vision and Robotics, Vol. 8, No. 6, 2018
- Safae Elhoufi, Aicha Majda, « ***Efficient use of recent progresses for Real-Time Semantic Segmentation*** » EURASIP Journal on Image and Video Processing (En cours de soumission)

Communications:

- Safae Elhoufi, Aicha Majda « ***Reconnaissance et Classification des images 3D sous format VRML*** », In First International Workshop on Wireless Technologies and Distributed Systems, Avril 09-10, 2014, Fes, Maroc.

- Safae Elhoufi, Aicha Majda and Khalid Abbad « ***3D objects recognition using D2 shape distribution*** », 12 th ACS/IEEE International Conference

on Computer Systems and Applications AICCSA 2015, November 17-20, 2015. Marrakech, Morocco.

- Maha Jazouli, Safae Elhoufi, Aicha Majda, Arsalane Zarghili, Rachid Aalouane « ***Stereotypical Motor Movement Recognition Using Microsoft Kinect with Artificial Neural Network*** » 18th International Conference on Machine Vision, Image Processing and Pattern Analysis, Stockholm, Sweden July, 11-12, 2016.

- Safae Elhoufi, Maha Jazouli, Aicha Majda, Arsalane Zarghili, Rachid Aalouane « ***Automatic Recognition of Facial Expressions using Microsoft Kinect with artificial neural network*** », In the The International Conference on Engineering & MIS (ICEMIS 2016), 22-24 September 2016, Agadir, Morocco.

- S.Elhoufi, A.Majda « ***Segmentation pour la classification supervisée des objets 3D en se basant sur l'approche 2D/3D*** » Deuxième édition du Colloque National Signal, ImaGe, Multimédia et Applications SIGMA2018, 17-18 Avril 2018, Rabat, Maroc.

Table des matières

Liste des figures

Liste des tableaux

Liste d'abréviations

La littérature relative aux thèmes traités dans cette thèse est rarement traduite. Pour la plupart des acronymes employés, nous avons utilisé la dénomination la plus courante, qui est de fait en langue anglaise.

2D: Bi-dimensionnel

3D: Tri-dimensionnel

CAO: Conception assistée par ordinateur

ASCII: American Standard Code for Information Interchange

PLY: Polygon File Format

STL: STereo-Lithography

MTL: Material Template Library

OFF: Object File Format

VRML: Virtual Reality Modeling Language

MMT: Machines à mesurer tridimensionnelles (MMT)

CSG: Représentation par construction de solides

LFD: LightField Descriptor

ACP: Analyse en composantes principales

EGI: Extended Gaussian Images

DH3DO: descripteur de Hough 3D optimal

KNN: K plus proches voisins – k-nearest neighbors

SVM: Séparateur à Vaste Marge – Support Vector Machine

RBF: Radial basis function

RNA: Réseau de neurone artificiel– Artificial Neural Network

PMC: Perceptron multicouche

SGD: Stochastic gradient descent

$P/ PCR: Point-Cloud Recognizer

FFT: Fast Fourier Transform

TFD: Transformée de Fourier Discrète – Discrete Fourier Transform

EMD: Earth Mover's Distance

KLD: Kullback-Leibler divergence

IH: Intersection d'histogrammes

CNN: Réseau de neurones convolutif

DCNN: Réseaux de neurones profonds convolutifs

ReLU: Rectified Linear Units

FCN: Réseau entièrement convolutionnel

FPS: Inference speed

BN: Batch normalisation

IoU: Intersection Over Union

FLOPs : Opérations en virgule flottante par seconde

Introduction générale

Le monde change et la technologie se développe rapidement pour améliorer la qualité de vie humaine. De nos jours, personne ne peut imaginer la vie quotidienne sans utiliser des réalisations technologiques telles que l'Internet, les téléphones mobiles, les smartphones, etc. La croissance de la technologie est si rapide que chaque jour nous découvrons de nouvelles informations multimédia, dont nous distinguons Les modèles tridimensionnels, qui ont récemment ouvert une nouvelle porte aux utilisateurs pour profiter du monde incroyable de la 3D. Cette croissance est en partie due au développement de scanners 3D, d'apparition des logiciels de modélisation, des progrès des capteurs et des capacités de stockage et même de manipulation d'information.

Actuellement, La reconnaissance de model tridimensionnel reste l'un des domaines ouverts de recherche et d'expérimentation. Au cours de ces dernières années, les objets 3D jouent un rôle très important dans de nombreux domaines. Ils sont motivés par un large éventail d'applications à savoir : la conception assistée par ordinateur (CAO), l'imagerie médicale, la réalité virtuelle, les films et jeux vidéo, la reconnaissance faciale, le contrôle de qualité, etc.

L'émergence de nouvelles technologies pour sauvegarder et manipuler l'information multimédia dans les applications mentionnées ci-dessus, a entraîné une forte augmentation de nombre de modèles 3D, et par suite augmenté le besoin en outils automatiques pour la recherche, la récupération et la compréhension de ces modèles. Ces outils devraient être en mesure d'aider les utilisateurs, qui proviennent d'universités ou des entreprises, à gérer les modèles disponibles. La gestion des modèles comprend une variété de tâches pour différentes applications: Un ingénieur ou un concepteur peut vouloir décomposer un modèle mécanique en ses composants afin de les réutiliser pour créer de nouveaux modèles. Un chercheur en médecine devrait être en mesure de détecter via les images CT-Scan (Computerized Tomography) disponibles la présence de cellules anormales. Un producteur peut faire un usage intensif des modèles 3D pour améliorer la réalisation. D'autres activités peuvent être exigées par les animateurs, les chimistes, etc.

- ✓ **Motivation et objectifs**

Suite à la croissance des modèles 3D sur Internet ou d'autres ensembles de données spécifiques au domaine, la question de "comment générer des modèles 3D?" a évolué en "comment les trouvons-nous?". Cela signifie que la tendance dans la recherche est de proposer de nouveaux systèmes efficaces pour, indexer, rechercher, segmenter et récupérer un modèle souhaité dans des bases de modèles 3D, ou bien même dans une scène 2D ou 3D.

Un remède efficace à ce besoin consiste à concevoir un système automatique pour effectuer l'appariement et la récupération basés sur le contenu. Le système devrait être en mesure d'interagir avec les utilisateurs en obtenant un modèle en tant que requête, et nous permettra par la suite de trouver des modèles similaires à partir de l'ensemble de données disponibles. Ceci s'effectue en se basant sur les descripteurs de forme, avec lesquels les modèles 3D seront représentés. Un système de récupération de modèle 3D avec descripteur de forme idéal devrait être capable de récupérer des modèles similaires dans une période de temps raisonnable. De plus, le descripteur de forme devrait offrir une qualité de discrimination élevée ainsi qu'une robustesse à la déformation, au bruit et à d'autres changements de surface.

Il est convenable de mentionner que, dorénavant, les expressions "modèle 3D", "objet 3D", "modèle tridimensionnel" et "forme 3D" sont utilisées de manière interchangeable puisqu'elles désignent le même terme.

- ✓ **Résumé des contributions**

Dans cette thèse, nous apportons trois contributions différentes au domaine de la reconnaissance des modèles 3D et de la segmentation des scènes. Les deux premières contributions incluent l'introduction de nouvelles méthodes d'indexation et de classification d'objets 3D basées sur des approches dites 2D/3D et 3D/3D, par lesquelles des modèles 3D similaires à une requête sont mis en correspondance et récupérés. La dernière contribution est liée à une technique de segmentation sémantique en temps réel d'une scène 2D contenant plusieurs objets. C'est dans cette optique que nous listons maintenant nos contributions principales :

- **L'approche 2D/3D :** Pour la première approche, l'extraction des informations caractéristiques d'un modèle ne se fait pas directement sur le modèle 3D, mais sur une représentation en deux dimensions de celui-ci. De ce fait, nous utilisons des images 2D extraites de différents points de vue. Le descripteur FFT est appliqué aux vues pour attribuer le rendu spectral. Cette description nous a permis de définir pour chaque vue un vecteur caractéristique basé sur l'extraction carrée des spectres de Fourier. En ce qui

concerne la classification, nous nous sommes basés sur l'utilisation de l'algorithme de reconnaissance de nuages de points $P. Ce type d'approche permet deux possibilités de comparaison soit directement avec une image requête ou avec un autre objet 3D en comparant leurs ensembles de vues.

- **L'approche 3D/3D :** La seconde approche agit directement sur la forme du modèle 3D. La méthode utilisée permet de caractériser globalement la surface de l'objet. Elle est basée sur des fonctions de forme qui mesurent les propriétés géométriques globales du modèle 3D. D'abord, nous choisissons la fonction de forme dont la distribution fournit une bonne signature pour l'objet polygonal 3D. Cette distribution est invariante aux transformations géométriques et permet une description discriminante et robuste de la forme 3D. Ensuite, nous évaluons différentes mesures de similarité, afin de conclure celle qui nous offre une bonne discrimination en fonction du temps de calcul. Finalement, nous présentons des techniques d'apprentissage automatique telles que RNA, KNN, SVM, pour la classification et la reconnaissance d'une requête inconnue par rapport aux données d'apprentissage.
- **Segmentation sémantique en temps réel :** L'objectif ici n'était pas uniquement de pouvoir identifier des objets, mais aussi de les segmenter en temps réel. Afin d'atteindre cet objectif nous avons développé un nouveau réseau de neurones convolutif, qui intègre des techniques récentes et efficientes tels que « depthwise separable convolution », « shuffled convolution », « grouped convoltuion » et utilise la convolution dilatée pour élargir le champ réceptif. Afin de rassembler les caractéristiques extraites de différentes couches et de réduire notablement le nombre de paramètres, nous avons choisi d'utiliser la technique dense connectivité. Notre réseau fonctionne sur des images de haute résolution 512x1024 dont il atteint une précision avec une durée d'inférence satisfaisantes sur l'ensemble de données Cityscapes.

✓ **Description des chapitres**

Cette thèse s'organise comme suit :

– Introduction (0).

– Chapitre 1 : Terminologie et notions de base sur les objets 3D.

– Chapitre 2 : Les méthodes d'indexation.

– Chapitre 3 : La classification des objets 3D en se basant sur l'approche 2D/3D

– Chapitre 4 : La classification des objets 3D en se basant sur l'approche 3D/3D

– Chapitre 5 : Segmentation sémantique par les réseaux convolutifs.
– Conclusions et perspectives (6).

Dans le premier chapitre, nous présentons tout d'abord les notions de bases liées aux modèles 3D, à savoir : la définition du modèle 3D, la modélisation, l'acquisition et la représentation. Ensuite, nous présentons les principaux aspects de l'indexation et de la recherche d'objets 3D.

Dans le second chapitre, nous proposons un état de l'art détaillé traitant les méthodes d'indexation, nous les regroupons dans plusieurs catégories, dites '2D/3D', '2.5/3D' et '3D/3D' et nous retirons pour chacune d'entre elles les principaux avantages et inconvénients.

Dans le troisième chapitre, nous présentons notre contribution liée à la reconnaissance et la classification des objets 3D en se basant sur l'indexation 2D/3D. Tout d'abord, nous présentons les principes de base de la méthode d'indexation 2D/3D, ainsi que les moyens techniques permettant de transformer un objet 3D en vues 2D. Ensuite, nous définissons les différentes méthodes de classification. Puis, nous présentons la solution proposée pour la mise en œuvre de notre système. Finalement, nous exposons les résultats expérimentaux visant à évaluer l'efficacité et la robustesse de notre proposition.

Dans Le quatrième chapitre, nous présentons notre contribution liée à la reconnaissance et la classification des objets 3D en se basant sur l'indexation 3D/3D. Tout d'abord, nous citons quelques notions de base sur l'indexation 3D/3D. Ensuite, nous définissons diverses formules de calcul de distances pour estimer la similarité des vecteurs caractéristiques. Puis, nous détaillons la description du problème et nous proposons des solutions pour mettre en œuvre notre approche. Finalement, nous présentons et discutons les résultats intéressants obtenus par cette approche.

Dans le cinquième chapitre, nous présentons notre contribution liée à la segmentation sémantique en temps réel en se basant sur les réseaux convolutifs profonds. Dans un premier temps, nous introduisons les bases théoriques en apprentissage profond sur lesquelles s'appuie notre architecture. Ensuite, nous rappelons brièvement les motivations et le fonctionnement des réseaux convolutifs profonds. Puis, nous étudions plus en détail les techniques des réseaux convolutifs appliquées à la segmentation sémantique spécialement en temps réel. Finalement, nous présentons et détaillons notre architecture accompagnée des principaux résultats obtenus.

Dans la conclusion, nous résumons les réalisations effectuées et nous rappelons les principaux résultats obtenus. Enfin, nous proposons de nouvelles pistes et nous évoquons les perspectives qui pourraient orienter nos futures recherches.

Chapitre 1 : Terminologie et notions de base sur les objets 3D

Chapitre 1 : Terminologie et notions de base sur les objets 3D

1.1 Introduction

Au cours des dernières décennies, les recherches scientifiques et technologiques dans les domaines d'imagerie, de la télécommunication et de l'infographie ont contribué à l'émergence de nouveaux médias, en particulier des données numériques tridimensionnelles (3D). De nos jours, le traitement, l'indexation et la recherche d'objets 3D font partie des fonctionnalités abordables sur Internet. Une grande communauté scientifique travaille dur sur les problèmes ouverts et les nouveaux défis, dont la représentation et la manipulation des objets 3D, l'accès rapide à une énorme base de données 3D ou la reconnaissance et la classification de ces derniers.

Les progrès récents dans l'acquisition de scanners et les technologies de rendu graphique stimulent la création d'archives de modèles 3D pour plusieurs domaines d'application. Ce progrès a contribué aussi à l'évolution des activités 3D que ce soit pour les professionnels (modélisation 3D et outils de création et de manipulation) ou bien pour les utilisateurs finals (matériel graphique accéléré 3D, Web3D, nouvelle génération de téléphones cellulaires afin de visualiser interactivement les modèles 3D).

Dans le présent chapitre, Nous nous concentrons sur la description de certains sujets liés aux modèles 3D, à savoir : la définition du modèle 3D, la modélisation, l'acquisition et la représentation. Nous présentons par la suite les principaux aspects de l'indexation et de la recherche d'objets 3D.

1.2 Modèle 3D

Un modèle 3D est une représentation abstraite d'un objet en trois dimensions, combinant à la fois la longueur, la largeur et la profondeur. Il représente un objet 3D utilisant une collection de points dans l'espace 3D, reliés par diverses entités géométriques telles que des triangles, des lignes, des surfaces courbes, etc.

c'est à partir des années 1990 que les modèles 3D se popularisèrent et se développèrent de façon importante. Ils peuvent être considérés comme la quatrième génération d'informations multimédia, qui a émergé après le son numérique dans les années 1970, les images numériques dans les années 1980 et les vidéos numériques dans les années 1990. Ils sont devenus populaires en parallèle au développement de l'acquisition de données 3D, de la modélisation graphique 3D et des technologies matérielles 3D. De nos jours, les modèles 3D sont largement adoptés dans diverses applications telles que l'industrie médicale, les films, la technologie des jeux, les

environnements interactifs, l'architecture, etc. Il existe plusieurs formats de fichiers 3D, parmi lesquels nous citons:

- **PLY**

Ply a été développé au milieu des années 90. Il est connu sous le format de fichier Polygone File Format ou le Stanford Triangle Format. Le format a été principalement conçu par le laboratoire graphique Stanford pour stocker des données tridimensionnelles provenant de scanners 3D. Il existe deux versions de ce format de fichier, un en ASCII, l'autre en binaire.

- **3DS**

3DS existe depuis 1990, est l'un des formats de fichiers utilisés par le logiciel de modélisation, d'animation et de rendu 3D Autodesk 3ds Max. C'était le format de fichier natif de l'ancien DOS Autodesk 3D Studio (versions 1 à 4), qui était populaire jusqu'à ce que son successeur (3D Studio MAX 1.0) le remplace en avril 1996. Ce format est devenu un standard d'industriel pour le transfert de modèles entre programmes 3D ou pour stocker des modèles de catalogues de ressources 3D (avec OBJ, plus fréquemment utilisé comme format de fichier d'archivage). Alors que le format 3DS a pour but de fournir un format d'import/export, ne conservant que les données essentielles de géométrie, texture et éclairage.

- **STL**

Stl est un format de fichier natif du logiciel de stéréolithographie. Il est inventé par 3D Systems en 1987. Ce format de fichier est pris en charge par de nombreux autres logiciels, il est largement utilisé pour le prototypage rapide, l'impression 3D et la fabrication assistée par ordinateur. Néanmoins, Il ne décrit que la géométrie de surface d'un objet en 3 dimensions. Ce format ne comporte cependant pas d'informations concernant la texture, la couleur ou les autres paramètres habituels d'un modèle de conception assistée par ordinateur.

- **OBJ**

Le format OBJ est un format de fichier contenant la description d'une géométrie 3D. Il a été défini par la société Wavefront Technologies. Il est basé sur un format ascii avec une syntaxe simple. Il se divise en deux fichiers : un fichier .OBJ qui donne toutes les informations sur les sommets et les faces, et un fichier .mtl (Material Template Library) qui contient les données sur les matériaux. On peut décomposer un fichier .OBJ de cette manière :

- ✓ Indication du fichier .MTL
- ✓ Définition des sommets
- ✓ Attribution des faces.

- **OFF**

Le format OFF a été développé par Digital Equipment Corporation's Workstation Systems Engineering en 1986 pour l'échange et l'archivage d'objets 3D. C'est un format ASCII, il est indépendant des langages, des périphériques et des systèmes d'exploitation. Ce format décrit uniquement des objets ou des scènes 3D statiques. Il a l'avantage d'être d'une grande simplicité. Il a été largement utilisé et on peut dire qu'il est maintenant plus connu que répandu. Il est en entrée ou en sortie de la plupart des convertisseurs de formats CAO.

- **VRML**

En mai 1995, les spécifications définitives de VRML 1.0 marquèrent la naissance officielle du VRML. Le Virtual Reality Modeling Language est un langage de description d'univers virtuels en trois dimensions destiné à être exploité sur le Web. Le but premier de ce langage est de permettre la représentation d'univers interactifs 3D virtuels. Comme il est destiné au Web, VRML a plus d'analogies avec HTML, comme la possibilité d'ajouter des ancres : En cliquant sur un objet, nous appelons alors un document qui peut être une page Web. En fait, proprement dit c'est un langage de présentation et non de programmation. Le fichier VRML ne contient généralement pas une suite d'instructions mais plutôt les informations permettant au visionneur d'afficher les éléments formes, senseurs, lumières, etc.

Pour modéliser une scène, on fabrique des objets, on les positionne dans la scène, nous ajoutons éventuellement des sources de lumière, etc. Pour modéliser des objets en 3D, on les décrit au moyen de formes de base ou par des méthodes plus complexes, on leur applique des matériaux, des textures. Les fichiers VRML ont habituellement pour extension « **.wrl** » ou bien « **.vrml** », néanmoins il existe bien d'autres extensions à savoir « **.wrz** » et « **.vrw** ». Les fichiers .wrl, qui peuvent être stockés localement sur un ordinateur ou téléchargés depuis un serveur web. Ilssont visualisés à l'aide d'un visionneur, qui est soit un plugin ajouté au navigateur web ou encore un logiciel autonome indépendant du navigateur web, qui est installé sur l'ordinateur de l'utilisateur.

- **3D-XML**

Le nouveau format développé en 2004 est rendu publique le 15 juin 2005 par Dassault Systèmes accompagné d'un visionneur gratuit permet de lire les fichiers et de connaître la structure arborescente d'un assemblage. Le plug-in fourni par l'éditeur permet également d'intégrer des modèles 3D-XML au sein de documents Office et de choisir le point de vue dynamique sans quitter le document édité. Il s'agit d'un format riche et souple mais encore peu utilisé par les industriels.

La possession d'objets 3D passe par deux étapes nécessaires : la modélisation et la représentation. Pour cela, nous nous concentrerons dans la partie suivante sur ces deux phases. En raison qu'elles conditionnent la qualité du modèle 3D et que ses propriétés peuvent influencer la pertinence de l'indexation et de la recherche.

1.3 Modélisation et acquisition des modèles 3D

La modélisation 3D est le processus qui consiste à créer, dans un logiciel de modélisation 3D un modèle numérique d'objet en trois dimensions. Il est largement utilisé dans une multitude de domaines.

Les industries du jeu vidéo et du cinéma sont probablement les exemples les plus courants et les plus connus d'utilisation de la modélisation 3D. La majorité des jeux publiés aujourd'hui sont en 3D, avec des personnages et des environnements 3D. Ces modèles sont devenus plus réalistes, complexes et détaillés au fil du temps. L'industrie cinématographique pousse encore plus loin l'utilisation de la modélisation 3D, en utilisant la modélisation pour ajouter des effets spéciaux ou créer des environnements [Jonpolygon2016].

Nous mentionnons aussi l'exemple de l'impression 3D, c'est une industrie émergente et en croissance rapide avec la modélisation 3D. Les utilisations de l'impression 3D sont extrêmement diverses. Les industries aéronautiques et spatiales utilisent l'impression 3D pour accélérer et améliorer le processus de construction de véhicules spatiaux et de pièces d'aéronefs. La bio-impression est un domaine de la médecine où l'impression 3D est utilisée pour créer des organes et des tissus de remplacement [3DPrinting.com 2016]. De plus, l'impression 3D est également utilisée en privé pour imprimer des objets décoratifs ou des pièces de rechange.

La modélisation 3D peut être effectuée en utilisant l'une des techniques suivantes:

1.3.1 Modélisation manuelle (infographie)

Il existe un large éventail de logiciels permettant aux concepteurs de construire leurs modèles 3D préférés. Cette classe de logiciels permet aux infographistes de modifier les modèles en ajoutant, soustrayant les parties souhaitées des modèles. Ces logiciels permettent aussi de définir des attributs propres à l'objet tels que : la couleur, la texture, le type de surface (pierre, bois, métal…) ainsi que des points de vue et des lumières pour rendre l'animation encore plus réaliste. Un logiciel de CAO typique, par exemple, permet aux ingénieurs de générer un plan 3D d'un bâtiment dans un court laps de temps. Bien qu'il existe d'autres logiciels de modélisation exemplaire tels que Maya, 3D Max, Cheetah3D, Anim8or, Blender etc.

1.3.2 Acquisition 3D

Dans cette classe de techniques de modélisation 3D, l'objet du monde réel est numérisé pour être sauvegardé et traité par les ordinateurs. Les scanners laser 3D et les machines à mesurer tridimensionnelles (MMT) sont deux exemples de dispositifs de modélisation. Dans le premier cas, un objet du monde réel est analysé, puis ses données brutes (typiquement un nuage de points x, y, z) sont utilisées pour générer une représentation précise de maillage polygonal ou autre. D'une manière plus claire, Les scanners 3D transforment la surface d'un objet réel en un nuage de points cohérent permettant une visualisation dans un univers en trois dimensions. Tandis que dans le deuxième cas, les machines à mesurer tridimensionnelles sont conçues pour déplacer une sonde de mesure afin de déterminer les coordonnées des points sur l'objet. La figure 1.1 montre un scanner laser 3D et un dispositif MMT utilisé pour la construction de modèles 3D à partir d'objets réels. Les techniques d'acquisition de modèles 3D sont principalement utilisées dans les applications du patrimoine culturel, de l'animation et de l'ingénierie inverse.

Figure1. 1: Exemples de périphériques d'acquisition de modèles 3D. À gauche: acquisition par le laser du modèle David de Michelangel, à droite: un dispositif MMT pour générer un chat.

1.3.3 Reconstruction 3D par images

Les modèles 3D peuvent être construits à partir d'un ensemble d'images 2D, lorsque les contraintes de temps et/ou de budget ne permettent pas aux producteurs de générer manuellement un modèle 3D entièrement réalisé. Dans cette situation, un modèle 3D est dérivé algorithmiquement d'un ensemble d'images 2D statiques qui sont capturées à partir d'angles de vue différents. Ensuite, un algorithme approprié est appliqué pour combiner les images pour construire un modèle 3D. La figure 1.2 montre deux échantillons de modèles de visages 3D construits à partir d'un ensemble d'images 2D.

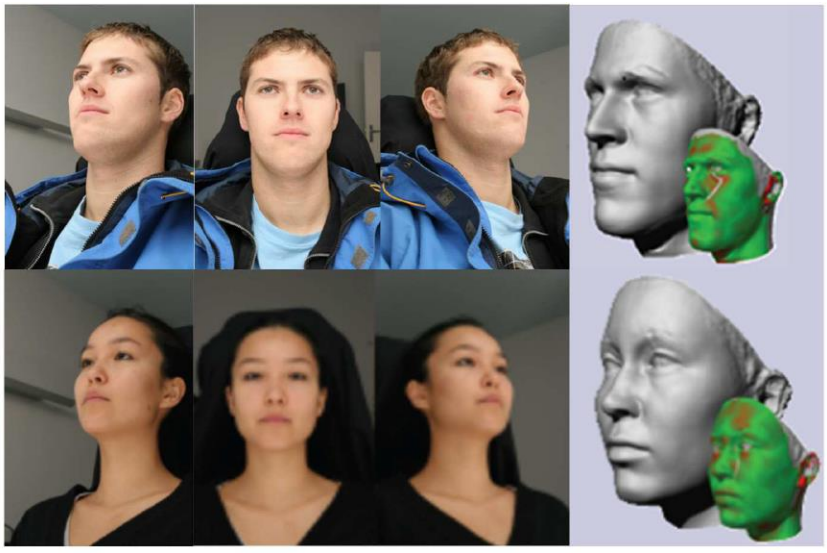


Figure1. 2: Modèles de visage 3D reconstruit à partir d'un ensemble d'images 2D [Amberg2007].

Dans la littérature, on trouve trois familles de méthodes permettant la reconstruction de modèles 3D. La première appelée approche par silhouette, elle est décrite par Sullivan et Ponce [Sullivan1998] détermine l'enveloppe visuelle initiale du modèle 3D à partir d'un ensemble de silhouettes. Cette méthode est robuste, mais reste limitée à des formes simples. La seconde variété d'approches, introduite par Horn [Horn1975], se base sur l'analyse de l'ombrage d'un objet donnée par les propriétés de réflexion diffuse des surfaces. Toutefois cette catégorie d'approche est difficile à mettre en place car extrêmement dépendante des conditions d'éclairage du modèle. Finalement, la troisième approche est traitée par Seitz et Dyer [Seitz1997]. Elle utilise l'information couleur contenue dans les photographies de l'objet. La combinaison de ces méthodes permet d'améliorer la robustesse comme le montrent les travaux de Hernandez [Hernandez2004] sur la fusion d'informations de silhouettes et de textures [Napoléon2010].

1.4 Représentation 3D

L'accession des modèles 3D devient de plus en plus importante, pour cela la demande de visualisation de ces données dans différentes applications a généré plusieurs intérêts de recherche. Bien que la majorité des objets réels soient des volumes, les représentations utilisées pour les modéliser varient en fonction des applications. Dans la littérature, on distingue cinq types de description d'un objet 3D : les nuages de points, les représentations polygonales, les représentations paramétriques de la surface, représentation volumique et la représentation par construction de solides "CSG".

Comme le montre la Figure 1.3, les modèles 3D peuvent être représentés en fonction de leurs informations de surface ou de volume. La surface d'un modèle peut être sous forme d'un nuage de points, d'un maillage polygonal ou d'une surface paramétrique, tandis que les caractéristiques volumétriques d'un objet sont représentées par des voxels.

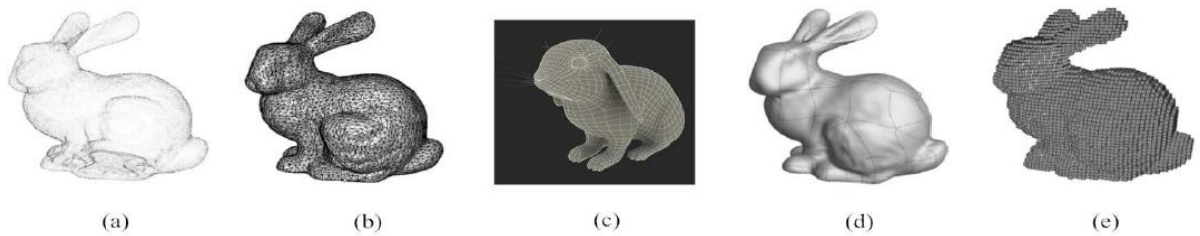


Figure1. 3: Différentes représentations d'un exemple de modèle Bunny. (a) Nuages de points, (b), (c) maillages polygonaux, (d) ensemble de surface paramétrique et (e) voxel. Source [Dugelay2008].

1.4.1 Représentation par Nuage de points

Les nuages de points sont simplement composés d'un ensemble de points et sont souvent destinés à représenter la surface externe d'un objet. Tandis que, les relations de connectivité entre les points ne sont pas considérées dans ce format de représentation (Voir figure 1.3 (a)). Un modèle représenté via un nuage de points peut être directement fournit par les scanners 3D. Ce nuage est souvent transformé, pour une meilleure visualisation, en un maillage polygonal avec une reconstruction surfacique. Selon le type du capteur, le nuage de points peut être organisé (structuré) ou désorganisé (non structuré) [Shaiek2013].

La représentation par un nuage de points est directe et simple, par contre elle requiert un très grand nombre de points pour reproduire un modèle complexe et n'est pas convenable à des manipulations de modèles articulés. Les principales limites de cette représentation sont: (i) elle dépend de l'échantillonnage des points et (ii) elle ne génère pas la surface de l'objet. Pour cette raison, les nuages de points sont souvent maillés [Mathieu2015].

1.4.2 Représentation polygonale

Ou maillage polygonal: Il est parmi les formats les plus souvent adoptés pour la représentation d'objets en utilisant une collection de sommets, d'arêtes et de facettes [Sheshina2016]. Cela se fait en discrétisant la surface de l'objet en une multitude de facettes jointes, composées chacune de sommets reliés par des arrêtes. La position des sommets de ces facettes dans l'espace 3D, souvent exprimée dans un repère cartésien, définit la géométrie du maillage, tandis que les relations d'adjacence entre sommets, arêtes et facettes définissent sa topologie (information de connexité) [Olfa2005]. Ils utilisent la connectivité (citons l'exemple de la relation d'adjacence entre les sommets), la géométrie (Exemple de l'emplacement du sommet) et les données de propriété (Exemple des vecteurs normaux) pour représenter les objets. Cette technique offre une apparence réaliste au modèle, mais il est important de déterminer le nombre de polygones à utiliser pour un modèle 3D spécifique. Plus le nombre de polygones utilisés est élevé, plus le modèle 3D est lisse et réaliste [Mathieu2015]. Cette représentation bénéficie d'une grande liberté d'édition et d'une très grande popularité. Elle leur permet de prendre place dans les jeux vidéo et dans la plupart des logiciels 3D. Néanmoins, elle ne permet pas de représenter précisément certaines surfaces. Les figures 1.3 (b) et 1.3 (c) montrent un modèle de lapin dans deux types de facette différents.

1.4.3 Représentation paramétrique de la surface

Ce format représente l'objet comme un ensemble de surfaces dans l'espace euclidien R3, ces surfaces sont données directement par des équations paramétriques calculables. Cette classe de représentation est l'approche la plus efficace. Contrairement aux maillages polygonaux, qui ne font qu'approcher la surface du modèle, cette représentation définit mathématiquement le modèle exact (Figure 1.3 (d)).

1.4.4 Représentation volumique

La représentation volumique est une autre représentation des modèles dont l'objet est considéré comme une densité volumétrique définie sur une grille régulière 3D. Ce format vise à encoder les informations volumétriques des objets, elle est largement utilisée en imagerie médicale. Cette représentation se base sur la notion de voxel. Les voxels décomposent l'espace en éléments cubiques de tailles différentes de manière à décrire la forme de l'objet en réduisant le nombre d'éléments nécessaires [Olfa2005]. Ils sont l'équivalent, pour l'objet 3D, des pixels pour l'image 2D. Par conséquent, comme dans le cas des pixels, les voxels eux-mêmes n'incluent pas leurs positions. En revanche, les coordonnées d'un voxel dans un modèle sont spécifiées en fonction des positions relatives des autres voxels (Figure 1.3 (e)). Le problème avec cette représentation est qu'elle est très coûteuse, en terme de stockage et temps de traitement [Faugeras1998].

1.4.5 Représentation par construction de solides "CSG"

La représentation par construction de solides [Watt2000] est une représentation largement utilisée pour les modèles architecturaux et les applications de CAO, également pour les modèles 3D reconstruits. Ces types d'objets sont généralement composés de plusieurs éléments de forme régulière (figure 1.4). Le CSG est né du constat que de nombreux objets créés par l'homme peuvent être représentés par des combinaisons d'objets solides simples ou de primitives géométriques, (exemple : cylindre, sphère, cône, tore, etc.) à l'aide d'opérateurs géométriques booléens (exemple : union, intersection, soustraction). C'est une représentation de haut niveau qui fonctionne à la fois comme une représentation de forme et un enregistrement de la façon dont il a été construit. Contrairement au maillage polygonal qui représente la forme en utilisant des surfaces. Le CSG utilise une représentation volumétrique; représentant la forme par des volumes élémentaires ou des primitives.

La représentation par construction de solides offre un espace de stockage réduit, avec une très bonne qualité de rendu. En revanche, elle est coûteuse en temps de calcul et possède un nombre limité de primitives de bases rendant difficile la représentation d'objets complexes.

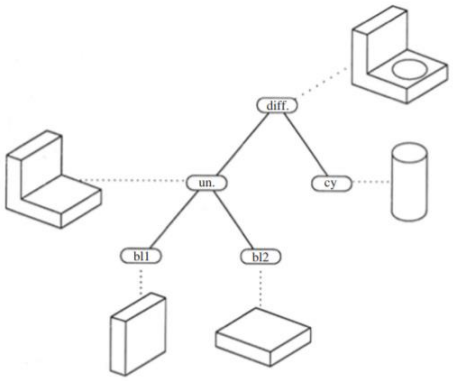


Figure1. 4: Représentation par construction du solide d'un objet 3D.

Pour conclure cette section, la représentation polygonale reste la plus utilisée car elle s'appuie sur le développement conjoint de techniques d'accélération matérielles et d'interfaces de programmation puissantes offrant, de ce fait, une accessibilité supérieure aux autres méthodes. C'est ainsi que nous avons opté pour ce type de représentation dans nos recherches.

1.5 L'indexation et la recherche appliquées aux objets 3D

1.5.1 Qu'est-ce que l'indexation d'objets 3D ?

L'indexation est un procédé qui permet de codifier le contenu d'un document dans le but d'accélérer l'interrogation de grandes bases de données (Figure 1.5). Le document peut être de différents types citons par exemple: un texte, une photo, un son ou, dans notre cas, un objet 3D [Napoléon2010]. Le processus d'indexation d'objet 3D doit nécessairement répondre à un enchaînement précis d'opérations. Celui-ci consiste d'analyser les différentes propriétés du modèle 3D afin d'en extraire les principales caractéristiques. Cette description est une représentation synthétique, mais pertinente du contenu, dont, il est possible de comparer rapidement deux objets en analysant leur description.

Les objectifs d'une indexation efficace sont les suivants:

- Éliminer les index non utilisés.
- Minimiser la redondance (utilisation de la mémoire et surcharge pour les écritures).
- Minimiser les analyses séquentielles.

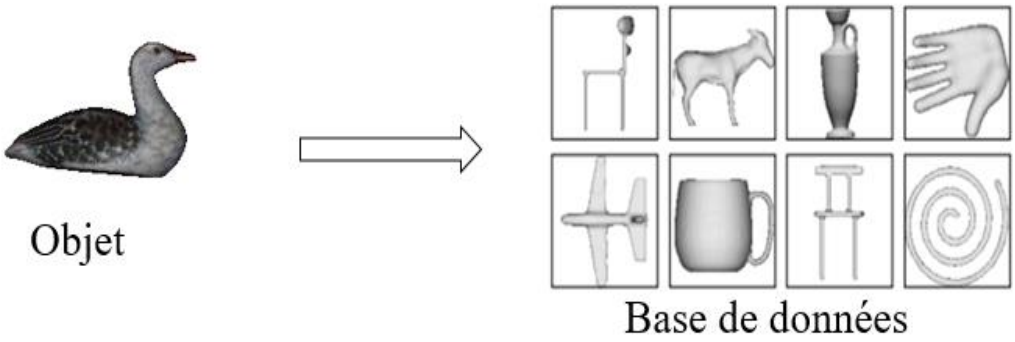


Figure1. 5: La recherche dans une base de données d'objets 3D.

En termes généraux, un index est une réorganisation d'un ensemble de données de son ordre naturel à un ordre dans lequel certaines tâches de recherche deviennent plus faciles. Dans la reconnaissance d'objet tridimensionnel, les données sont des formes, ou des apparences, et le réarrangement est d'une base de données de modèles d'objets étiquetés à un ordre par représentation numérique de la forme.

Dans un moteur de recherche, le rôle de l'index est de représenter les données d'origine de manière très succincte. Intuitivement, cela signifie que l'index devrait être invariant à certaines transformations géométriques de l'objet (translation, rotation, mise à l'échelle), mais aussi aux modes de représentation des objets 3D ou aux imperfections qui ont pu apparaitre lors de sa création. Le but de l'indexation est de récupérer efficacement, à partir d'une base de données volumineuse, des formes similaires qui pourraient représenter la requête [Douass2017]. Ces modèles candidats seront ensuite comparés directement à la requête, c'est-à-dire, vérifiés, afin de déterminer quel modèle candidat représente le mieux la requête. Plusieurs caractéristiques d'objet peuvent composer l'index. Cependant, avant même de chercher à caractériser L'objet 3D, il faut se poser la question du type d'informations que l'on souhaite extraire.

1.5.2 Méthodes d'indexation 3D

L'analyse des différentes techniques d'indexation d'objets 3D montre qu'il existe de nombreux types de signatures permettant de décrire les informations intrinsèques à chaque modèle. Dans ce contexte, trois approches peuvent être discutées:

- L'approche 3D / 3D agit directement sur la forme du modèle 3D. Elle utilise les facettes de l'objet 3D, appelé également maillage. Cette approche permet l'analyse de la forme du modèle tridimensionnel indépendamment de sa position dans l'espace ou du point de vue d'un observateur.

- L'approche 2D/3D se base sur la projection de l'objet 3D sur un plan donné. C'est-à-dire, que même lorsque l'on compare deux modèles 3D, la comparaison se fait via des projections en 2D de ces objets. Leur particularité réside dans le fait qu'un modèle 3D n'est pas directement caractérisé dans l'espace 3D, mais à travers un ensemble de vues de projection 2D, qui fournissent l'apparence 2D du modèle sous différents angles de vue.
- L'approche 2.5D/3D utilise des projections 2D de l'objet 3D associées à des informations 3D pour décrire la forme de l'objet. Généralement, il s'agit d'informations de profondeurs ou de courbures, encapsulées dans une image ou dans une série de coupes de l'objet 3D.

1.5.3 Le choix des requêtes

En effet, l'utilisateur ne possède pas forcement une copie de l'objet recherché. De ce fait, il est important de lui garantir une interrogation variée. Nous citons ci-dessous différentes possibilités de requêtes :

- Dessin au trait : C'est le moyen le plus simple pour rechercher un objet 3D. Grâce à ce type de requête, l'utilisateur à la possibilité de transcrire l'image mentale de sa recherche dans une représentation adaptée au système.
- Images 2D : Il est possible de détourer rapidement l'objet que l'on souhaite rechercher sur une photo et ainsi d'interroger la base de données de modèles 3D, éventuellement à partir de plusieurs photos.
- Objet 3D : Bien qu'étant le moyen d'interrogation le moins adapté à l'utilisateur, ce type de requête offre une grande robustesse grâce à l'information qu'elle véhicule.

1.5.4 Recherche par le contenu d'objets 3D

Contrairement aux documents, les modèles 3D ne sont pas facilement récupérés. Tenter de trouver un modèle 3D à l'aide d'une annotation textuelle et d'un moteur de recherche textuel conventionnel ne fonctionnerait pas dans de nombreux cas. Les annotations ajoutées par les êtres humains dépendent de la langue, de la culture, de l'âge, du sexe et d'autres facteurs. Ils peuvent être trop limités ou ambigus. En revanche, les méthodes de récupération de formes 3D basées sur le contenu, utilisant les propriétés de forme des modèles 3D pour rechercher des modèles similaires, fonctionnent mieux que les méthodes basées sur le texte [Thibault2011].

Dans ce contexte, la recherche de modèles 3D basée sur le contenu est devenue un sujet de recherche important. Plusieurs chercheurs ont étudié la possibilité d'effectuer une récupération efficace de modèles 3D à partir de grandes bases de données, en se basant sur les propriétés globale ou locale de forme.

1.5.4.1 Similarité entre objets 3D

Le principe général d'une recherche par le contenu d'objets repose sur l'hypothèse que la mesure de similarité entre deux objets 3D peut se ramener au calcul de la distance entre deux descriptions de ces objets. Un processus comparant deux objets comporte généralement trois étapes principales (Figure 1.6) :

- **Prétraitement de l'objet 3D:** Il est fréquent qu'une étape de prétraitement soit appliquée avant l'extraction de la signature. Dont le but est de transformer l'objet 3D afin d'obtenir une description pertinente. On distingue deux types de prétraitements :
    - Les problèmes de définition de la surface : l'objet 3D peut être incorrectement défini topologiquement et géométriquement. Par exemple, le maillage associé peut présenter des problèmes des facettes dégénérées, de bruit et de déformation, ... Il existe plusieurs façons pour résoudre ce type de problème, citons à titre d'exemple le débruitage, le filtrage et le rééchantillonnage.
    - Les invariances aux transformations géométriques : Il s'agit de rendre la description de l'objet invariante aux transformations simples de l'espace, à savoir la translation, la rotation et le changement d'échelle. Une phase de normalisation est nécessaire. Afin de centrer, de normaliser en taille et d'orienter l'objet 3D d'une manière précise.
- **Extraction de la signature:** L'extraction des informations caractéristiques d'un modèle 3D décrive ce dernier sous forme d'un vecteur, d'un graphe ou d'une séquence. Ceci est obtenue en se basant sur un ou plusieurs descripteurs de forme. Dans un processus général de recherche d'objets 3D, la signature de l'objet requête est la clé de recherche avec laquelle les éléments de la base vont pouvoir être comparés.
- **Mesure de similarité entre deux objets 3D :** Elle consiste à comparer les deux signatures extraites en utilisant une distance. Le but de cette mesure est d'évaluer la similarité entre l'objet requête et les objets de la base.

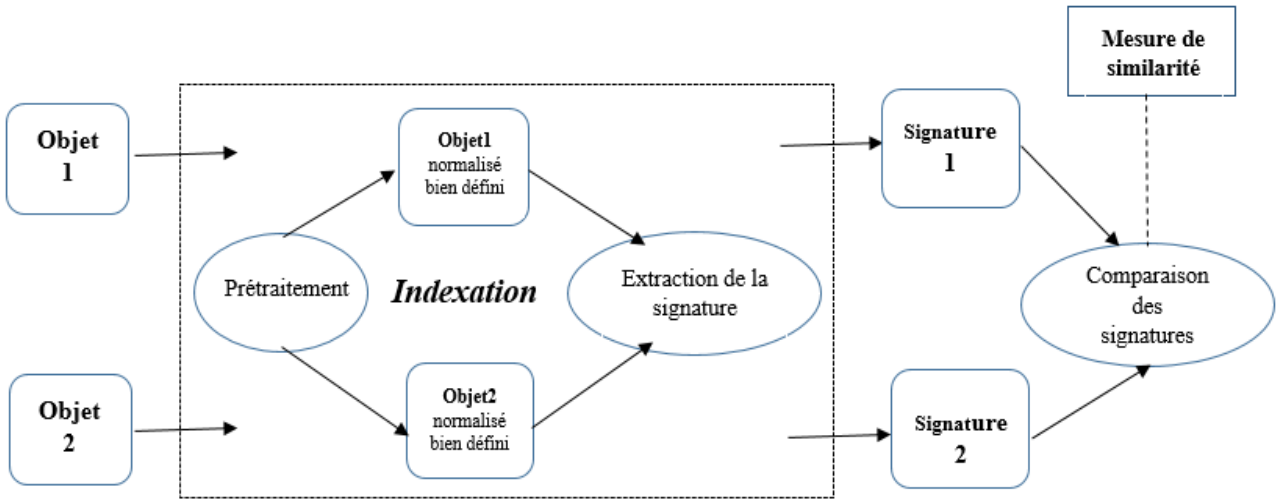


Figure1. 6: Processus comparant deux objets 3D.

1.5.4.2 Architecture d'un système de recherche d'objets 3D

Afin de permettre une interrogation efficace d'une base de données. Un système général de recherche d'objets 3D par requête doit nécessairement répondre à un enchaînement précis d'opérations. Celui-ci est constitué de deux phases indépendantes :

- Une phase hors ligne (Offline) : Cette phase se compose de toutes les étapes nécessaires pour effectuer l'indexation de la base de données. Il s'agit des étapes de normalisation, de description de la forme, de codage dans la signature synthétique et du stockage. Ceci permettra à l'utilisateur de retrouver rapidement le modèle 3D associé à une signature donnée.
- Une phase en ligne (Online) : Durant cette seconde phase, l'utilisateur interroge la base à l'aide de l'objet requête. Le temps de réponse du système doit être réduit au maximum. Dans un premier temps, les similarités entre l'objet requête et tous les modèles de la base de données sont mesurées à l'aide d'une distance associée au descripteur. Ces mesures font appel à la signature de l'objet requête (indexation online) et à la base des signatures stockées (indexation offline). Ensuite, les résultats seront récupérés en fonction du degré de similitude.

1.6 Conclusion

Nous avons présenté dans ce chapitre le contexte de nos travaux sur la reconnaissance des objets 3D. Il s'est étendu sur la définition, modélisation, acquisition et représentation du modèle 3D, ainsi que le fonctionnement d'un processus complet d'indexation et de recherche.

La recherche par le contenu d'objets 3D se base sur les propriétés de forme globale ou locale de l'objet. Elle s'avère la solution la plus prometteuse pour consulter et parcourir des bases d'objets. C'est finalement grâce aux différentes approches, dites "2D/3D", "2.5D/3D" et "3D/3D", que nous allons pouvoir offrir de nombreux moyens de requêtes, mais aussi une extraction efficace et effective des informations de formes contenues dans un objet 3D.

La reconnaissance et la classification des objets 3D demandent la mise en place des traitements d'indexation. Pour cela, nous décrirons dans le chapitre suivant un état d'art exhaustif et pertinent des différentes approches d'indexation.

Chapitre 2 : Méthodes d’indexation

2.1 Introduction

La disponibilité croissante des modèles 3D nécessite des algorithmes évolutifs et efficaces pour les structurer et les analyser. La recherche et la récupération des modèles 3D, en fonction des requêtes des utilisateurs spécifiquement avec des résultats pertinents, restent une préoccupation des chercheurs, la figure 2.1 montre un exemple d'une application de recherche dans une base de données d'objets 3D [Thomas2003]. L'utilisateur spécifie un aéronef en tant que modèle de requête (à gauche). Le système compare ensuite la requête à chaque modèle de la base de données, renvoyant les pointeurs vers les modèles les plus similaires (à droite).

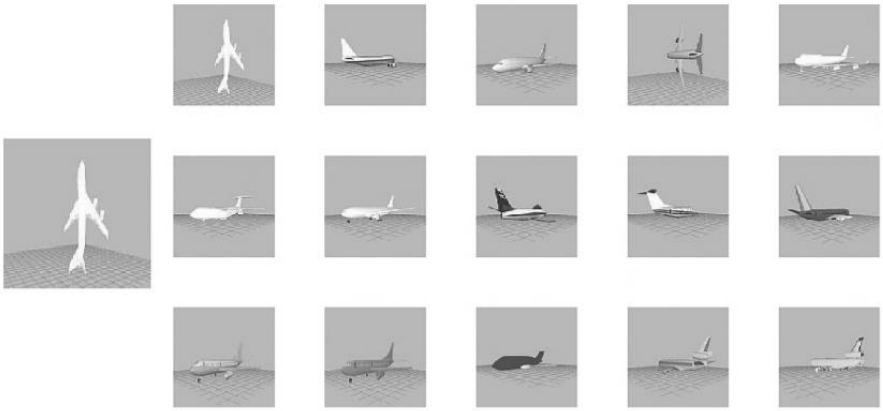


Figure 2. 1: Exemple de recherche de similarité sur une base de données de modèles 3D.

Ce chapitre ayant pour but d'établir un état d'art sur l'ensemble des méthodes permettant l'indexation des objets 3D à l'aide des descripteurs de forme. Le rôle des descripteurs de forme est de représenter les données d'origine de manière très courte. Intuitivement, cela signifie que l'index devrait être invariant à certaines transformations géométriques de l'objet (translation, rotation, mise à l'échelle), et devrait avoir une certaine robustesse au bruit.

Diverses méthodes de récupération ont été proposées pour permettre la recherche efficace des objets d'une forme 3D souhaitée, soit directement avec les propriétés géométriques, soit via une projection du modèle 3D sur plusieurs plans. Trois familles d'approches ont été déduites: les approches 2D/3D, les approches 2.5D/3D et les approches 3D/3D. Nous présentons par la suite une liste non exhaustive de ces approches. Et nous invitons le lecteur à se référer aux récents livre [Dugelay2008] et état de l'art [Tangelder2008], ainsi qu'aux études comparatives entre systèmes de recherche par le contenu, reportées dans [Tangelder2008, Li2014, Liu2013, Liu2012].

Nous présentons dans les sections 2.2, 2.3 et 2.4 un état de l'art sur les méthodes d'indexations 2D/3D, 2.5D/3D et 3D/3D. Nous retirons pour chaque méthode les principaux avantages et inconvénients. Finalement, nous proposons les méthodes que nous avons choisies d'implémenter pour nos tests.

2.2 Les méthodes d'indexation 2D/3D

Plusieurs recherches comme celles de Riesenhuber et Poggio ont conclu que dans un système de vision et de reconnaissance humain, l'objet 3D est représenté par un ensemble de vues [Riesenhuber2000] pour cela de nombreux algorithmes utilisent cette approche qui se base sur les informations provenant d'images 2D de l'objet 3D. Pour extraire l'image 2D, il suffit de faire une projection de l'objet sur un plan en deux dimensions. La majeure partie de ces approches utilisent non pas une, mais plusieurs projections afin de décrire la forme de l'objet dans différents angles de vue. L'idée principale de l'indexation 2D/3D se base sur le fait que si différents points de vue de deux objets 3D sont similaires, alors ces deux objets 3D sont également similaires. D'une autre manière Deux modèles 3D sont considérés semblables si les vues 2D qui les caractérisent le sont aussi.

Dans cette section, nous nous intéressons aux approches multi-vues qui sont généralement qualifiées d'indexation "2D/3D". En effet, elles caractérisent la forme d'un modèle à partir des informations provenant des projections de l'objet sur un ensemble de plans en deux dimensions. Contrairement aux autres approches, celle-ci ne garde aucune information tridimensionnelle. Les descripteurs sont le plus souvent basés sur la géométrie des silhouettes obtenues après la projection et s'appuient sur des approches multi-vues pour accroitre la pertinence des résultats.

Le problème de cette approche réside dans le choix de: la normalisation de la pose, la zone de projection, la taille de l'image, la nature des descripteurs de forme 2D et la taille de la signature [Elkhal2014]. Les auteurs dans [Ricard2005, Ansary2005, Mahmoudi2002, Nidhal2014] ont proposé des méthodes de sélection optimale des vues 2D pour représenter un modèle 3D. Le processus de sélection des vues caractéristiques repose sur un algorithme de classification adaptatif permettant de sélectionner le nombre optimal de vues calculées à partir de différents points de vue.

Nous divisons cette approche d'indexation en deux groupes : les méthodes basées sur les silhouettes et les images de profondeurs (Voir figure 2.2).

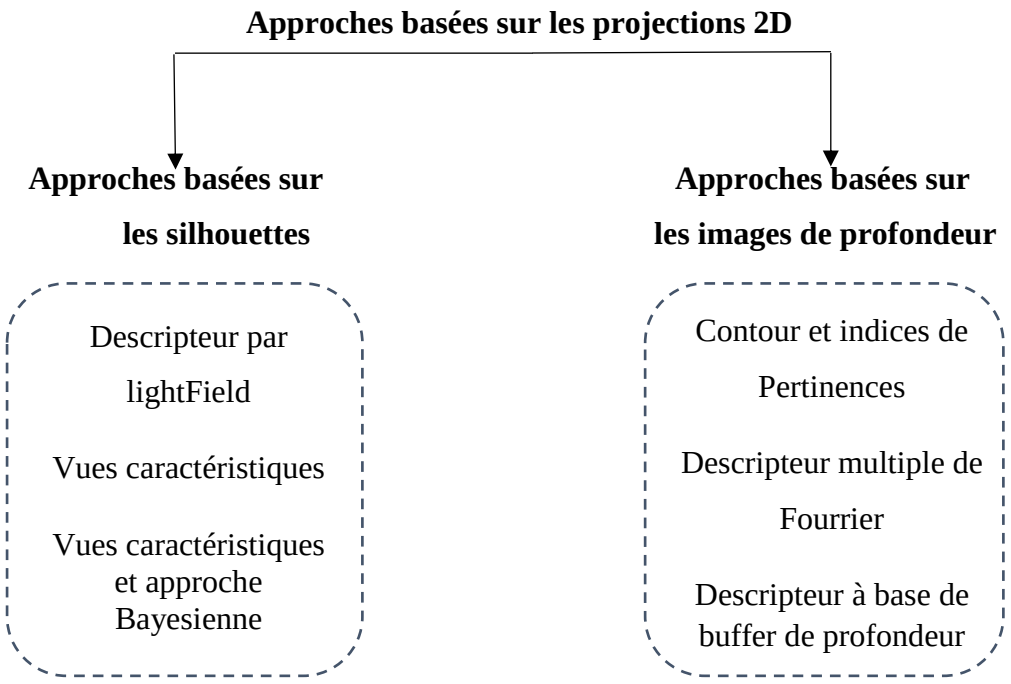


Figure 2. 2: Classification de l'approche 2D/3D en deux sous-groupes.

2.2.1 Approches basées sur les silhouettes

- **Descripteur par lightField**

Chen et al. [Chen2002] [Chen2003] proposent une méthode originale nommée "LFD", pour LightField Descriptor dont le but est d'indexer un objet 3D par des silhouettes. Afin de constituer le vecteur caractéristique, Ils proposent pour chaque silhouette 2D, obtenue par projection du modèle 3D sur un plan, de calculer ses 35 moments de Zernike [Zernike1934] ainsi que 10 cœfficients de Fourier [Zahn1972]. Pour une meilleure caractérisation de l'objet 3D, dix vues sont extraites à partir de dix angles répartis uniformément sur le dodécaèdre, comme l'illustre la figure 2.3. Il n'est pas essentiel de capturer une silhouette pour vingt sommets car les images créées sur les faces inverses du dodécaèdre sont identiques par la projection orthographique. Le point novateur de cette approche est l'utilisation de la notion de "Lightfield" pour se libérer de l'analyse en composantes principales tout en garantissant une invariance en rotation. Les "Lightfields" [Damien2010] sont un réagencement des silhouettes capturées à partir des sommets du dodécaèdre. Cette méthode permet de simuler une rotation de l'objet 3D en permutant uniquement les vues extraites.

Les résultats montrent que l'approche utilisée a une meilleure qualité de récupération et surpasse certaines autres méthodes. Sauf qu'elle est très couteuse à la fois en temps de calcul et

en place mémoire requise. L'un des points forts de cette approche est sa capacité à traiter des requêtes multimodales.

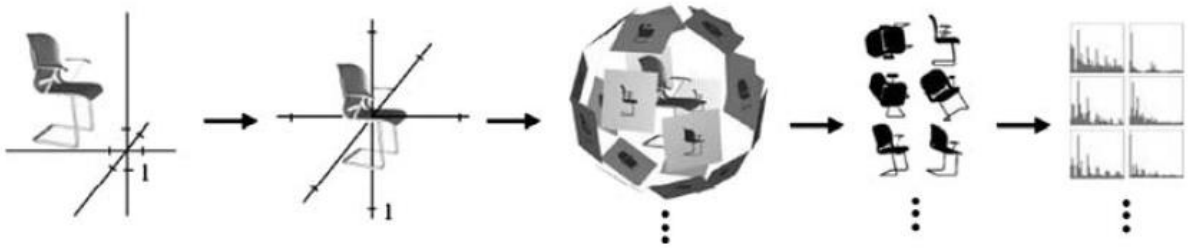


Figure 2. 3: Extraction du descripteur LightField pour l'objet chaise [Shen2003].

- **Vues caractéristiques (Silhouette)**

Développée par Mahmoudi et Daoudi [Mahmoudi2002], cette approche se base sur la caractérisation des objets 3D par un ensemble de 7 vues caractéristiques, dont trois principales, et quatre secondaires. Les directions principales sont déterminées par l'analyse du vecteur propre de la matrice de covariance associée à l'objet 3D. Elle associe pour chaque vecteur propre, une vue principale. Les vues secondaires sont déduites des vues principales, (Voir figure 2.4).

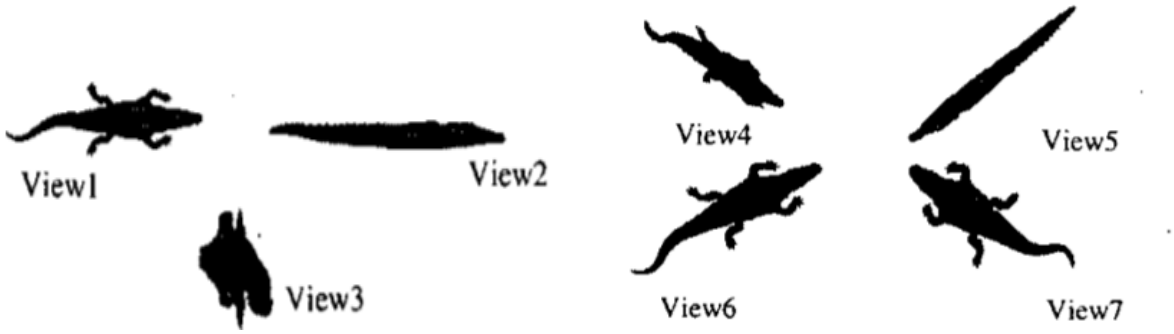


Figure 2. 4: Les vues principales et secondaires de l'objet Crocodile.

L'objet 3D est aligné par une analyse en composantes principales. Les auteurs utilisent par la suite le descripteur CSS (Curvature Scale Space) organisé autour d'une structure arborescente M-Tree. Ce descripteur caractérise le contour en exploitant les maximas de courbure, détectés à travers une analyse multi-échelles.

- **Vues caractéristiques et approche Bayesienne**

Partant du principe que toutes les vues d'un modèle 3D ne contiennent pas la même quantité d'informations, Filali-Ansary et al. [Filali-Ansary2006] [Filali-Ansary2007] propose un algorithme de classification adaptatif nommé AVC (Adaptive Vues Clustering); dont le but est de fournir un choix « optimal » de vues 2D à partir d'un modèle 3D, et une méthode bayésienne probabiliste.

Plus le modèle 3D est géométriquement complexe, plus le nombre de ses vues 2D est différent. A partir de 320 points de vue initiaux, leur algorithme sélectionne un ensemble de vues caractéristiques qui représentent le mieux le modèle 3D. Le nombre de vues caractéristiques varie de 1 à 40. Ils introduisent par la suite une nouvelle approche de recherche probabiliste qui prend en compte le fait que toutes les vues de modèles 3D n'ont pas la même importance, et aussi le fait que géométriquement les modèles simples ont plus de probabilité d'être pertinents que d'autres plus complexes.

Les résultats obtenus sur la base « Princeton Shape Benchmark » ont montré que la méthode proposée se situe en 2ème position juste après la méthode LFD [Chen2003]. En fait, ces résultats montrent la supériorité des modèles orientés vues.

2.2.2 Approches basées sur les images de profondeur

- **Descripteur à base de buffer de profondeur**

Cette approche traitée par Heczko et al. [Heczko2002] nécessite en premier une étape de normalisation en utilisant une analyse en composantes principales continue (ACP). Les auteurs incluent par la suite l'objet 3D dans un cube dont les côtés sont parallèles au repère du modèle. Puis, ils le projettent sur les six faces du cube et calculent des images de profondeurs en niveaux de gris à partir de ces silhouettes. En ce qui concerne le descripteur de forme, ils proposent la transformée de Fourrier rapide qui permet de fournir les basses fréquences de chaque vue extraite.

Comparée à de nombreux autres descripteurs 3D [Bustos2004] [Vranic2004], cette approche prouve une meilleure capacité de discrimination des objets 3D. Elle sera détaillée dans le chapitre suivant.

- **Descripteur multiple de Fourrier**

Ohbuchi et al. [Ohbuchi2003] proposent un algorithme qui permet d'extraire une signature de l'objet 3D. Cette approche nécessite une normalisation pour les transformations géométriques à savoir la translation et la mise en échelle. Les auteurs calculent ensuite 42 images de

profondeurs réparties uniformément sur la sphère unité, afin d'avoir une invariance par rapport à la rotation. Ils font ensuite passer ces images d'un repère cartésien (x, y) à un repère polaire (r, θ). Les basses fréquences de la transformée de Fourrier de chaque image résultante représente le vecteur caractéristique de la vue, et l'ensemble des vecteurs des 42 vues constitue la signature d'objet 3D. La similarité entre deux objets 3D, est mesurée en minimisant la distance entre toutes les 42^2 combinaisons possibles des vecteurs des deux ensembles.

Cette méthode demande un temps de calcul très élevé. Les expérimentations faites par les auteurs montrent que leur méthode a les meilleurs résultats de recherche comparée aux méthodes fondées sur les distributions de forme.

- **Contour et indices de pertinences (Silhouette et profondeur)**

Pour but d'associer aux silhouettes ou aux images de profondeurs des indices de pertinence, Chaouch et al. [Chaouch2006] [Chaouch2009] proposent une approche dont la démarche est basée sur les contours des silhouettes. En effet cela permet de favoriser des images contenant plus d'informations lors de la mesure de similarité. Elle propose de caractériser la forme d'un objet 3D en étudiant la position des points qui compose le contour. Afin de normaliser la longueur du contour, l'auteur effectue un échantillonnage angulaire en sélectionnant pour chaque secteur les N points les plus éloignés. Le descripteur est alors composé des cœfficients basses fréquences de la transformée de Fourier unidimensionnelle pour les N points, en coordonnées polaires, des trois vues correspondant aux directions des axes principaux.

La simplicité de cette approche la rend peu robuste et offre donc des résultats peu intéressants. Elle peut cependant servir, grâce à sa rapidité, à une phase d'élagage des objets peu pertinents.

Comme nous venons de le citer, plusieurs chercheurs ont essayé de réduire l'appariement de modèles 3D à un processus d'appariement d'images 2D. Malgré le fait que ces techniques rejettent des informations précieuses des modèles 3D, elles ont un bon pouvoir discriminant.

2.3 Les méthodes d'indexation 2.5D/3D

Dans cette section, nous nous intéressons aux méthodes qui projettent l'objet en trois-dimensions sur un ensemble de plans en deux dimensions, en gardant une information 3D pour décrire la forme de l'objet. D'où leur appellation "2.5D/3D". Dans la plupart des cas, il s'agit d'informations de profondeurs ou de courbures, encapsulées dans une image ou dans une série

de coupes de l'objet 3D. Il est en général difficile de construire une telle description sans le modèle 3D, ce qui oblige à avoir comme requête un objet de ce type.

Nous devisons cet état de l'art non exhaustif en trois groupes (Voir figure 2.5): Méthodes par cartes de courbures, Méthodes par coupes et Méthodes par images de profondeurs.

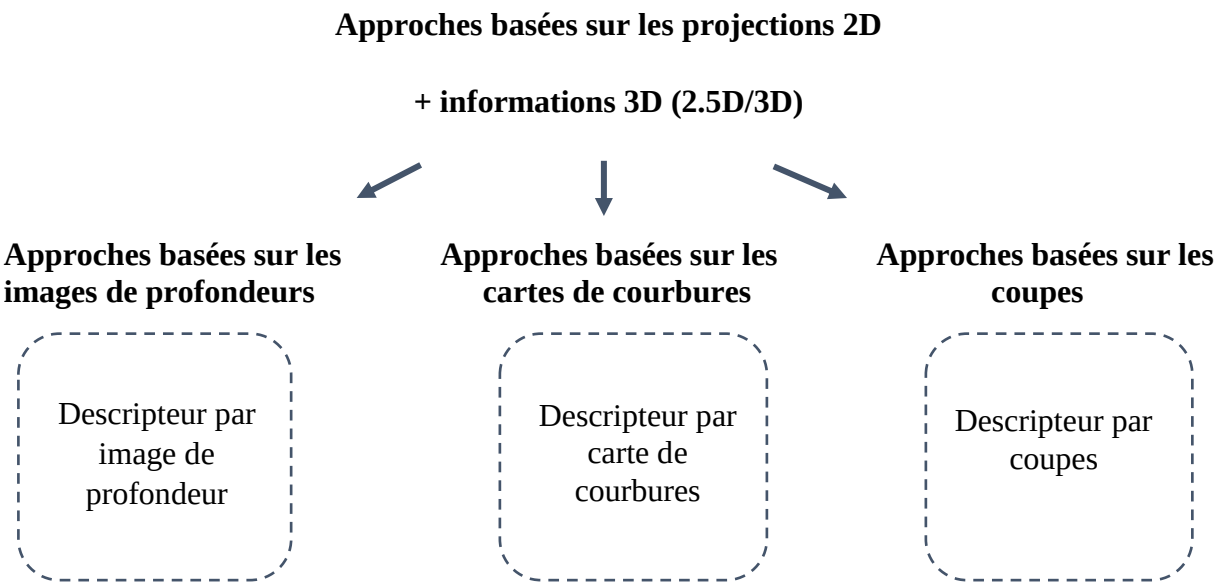


Figure 2. 5: Classification de l'approche 2.5D/3D en trois sous-groupes.

2.3.1 Approches basées sur les images de profondeurs

- **Descripteur par image de profondeur**

Vranic [Vranic2004] propose un descripteur nommé Depth Buffer Descriptor, basé sur les images de profondeurs. Son approche se base sur la description de la surface du modèle, où celui-ci est projeté sur les six faces du cube englobant où la profondeur des images obtenues correspond à la profondeur de l'objet. Le descripteur correspond finalement aux cœfficients basses fréquences de la transformée de Fourier rapide 2D de chacune des six images de profondeurs.

Les auteurs ont évalué leur approche sur la base du Princeton Shape Benchmark [Princeton2004]. Cette méthode montre de très bons résultats. Son point faible peut venir de sa nature "2.5D/2D" qui nécessite souvent des requêtes en 3D.

- **Images de profondeurs améliorées**

Comparée à l'approche originale, Passalis et al. [Passalis2007] améliorent cette approche à la fois sur le plan de l'alignement que sur celui de la description réelle de l'objet 3D. Les auteurs capturent toujours un ensemble de six images de profondeurs extraites des faces d'un cube.

Les améliorations proposées visent deux aspects:
– La correction de la pose du modèle en analysant les informations de symétrie, qui apparaissent en soustrayant les images de profondeurs de deux faces opposées.
– Le calcul du descripteur sur la somme et la différence des images, associées aux faces opposées, afin d'obtenir une invariance en translation dans la direction des axes de projections. Cette nouvelle méthode qui est une amélioration de la méthode originale accroît sensiblement la pertinence des résultats.

2.3.2 Approches basées sur les cartes de courbures

- **Descripteur par carte de courbures**

Cette approche développée par Assfalg et al. [Assfalg2003] présente une solution pour la récupération basée sur le contenu des modèles 3D qui s'appuie sur une description de chaque modèle en utilisant la notion d'une carte de courbure. Après un prétraitement initial du modèle cherchant à lisser l'objet et à réduire la complexité de son maillage, les auteurs stockent dans une carte 2D la courbure des objets déformés sur ellipsoïde. Tout en conservant l'information de courbure du modèle original. La correspondance est effectuée en comparant la carte de la requête avec les cartes de modèles de base de données. Les résultats expérimentaux montrent que cette technique, initialement développée pour décrire des images, peut être appliquée avec succès aux cartes de courbure. Afin de garantir une invariance aux transformations géométriques, cette méthode nécessite une normalisation de l'objet.

2.2.3 Approches basées sur les coupes

- **Descripteur par coupes**

Présentée par Jiantao et al. [Jiantao2004], cette démarche propose de décrire un objet 3D par une série de coupes tout au long de ses trois axes principaux. Pour garantir une invariance aux transformations géométriques, les auteurs sélectionnent parmi l'analyse en composantes principales sur les sommets et sur les normales, celle qui minimise la boite englobante.

Le modèle 3D est alors représenté par N coupes elles-mêmes décrites par la distribution de la distance entre deux points pris au hasard (Voir figure 2.6). Le point fort de cette approche est son indépendance au maillage de l'objet 3D.

Figure 2. 6: Deux objets 3D et leurs coupes, Source [Jiantao2004].

2.4 Les méthodes d'indexation 3D/3D

Cette section dresse un état de l'art sur l'indexation d'objets tridimensionnels, précisément l'indexation 3D/3D. Contrairement aux approches précédentes, les descripteurs de forme basés sur 3D tentent d'extraire la distribution des entités 3D pour caractériser les informations relatives à un modèle 3D. Elle Capture la forme directement à partir du modèle tridimensionnel. Plusieurs grandes voies se distinguent dans les différentes approches proposées, et nous avons fait le choix de les séparer en quatre groupes: Les approches globales qui cherchent à décrire l'objet dans son ensemble (section 2.4.1), approches locales qui basent leur description sur des informations locales de sa forme (section 2.4.2), approches par transformées qui visent à déterminer des représentations de forme globales, définies en terme de transformation intégrale (section 2.4.3) et Finalement, approches structurelles qui décrivent les objets 3D en s'appuyant sur des informations de haut niveau sur la structure de la scène et décrivent la forme intrinsèque de l'objet 3D (section 2.4.4) (voir figure 2.2). Nous présentons par la suite les principales méthodes qui existent et leurs avantages et faiblesses respectifs.

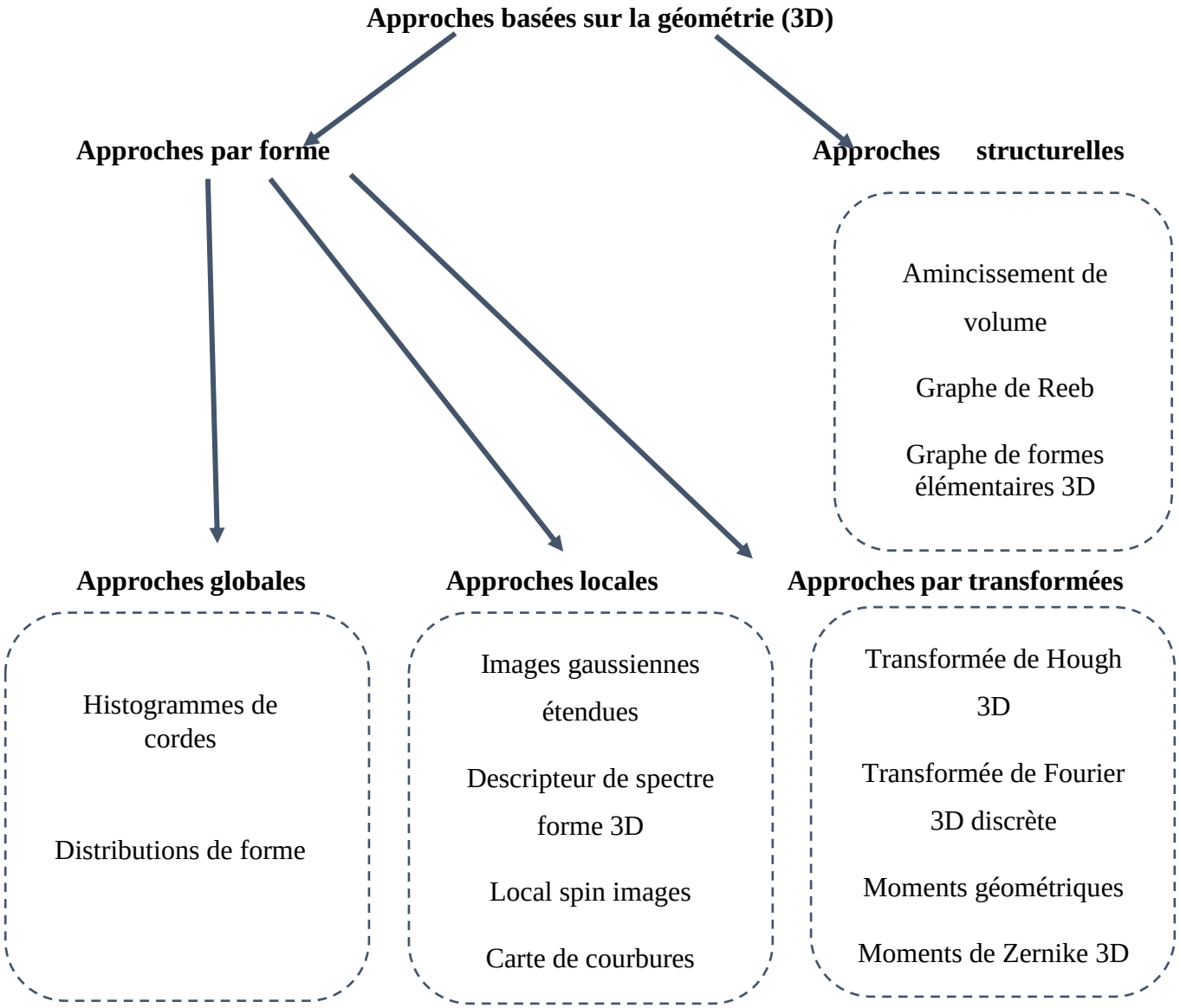


***Figure 2. 7**:* Classification de l'approche 3D/3D en cinq sous-groupes

2.4.1 Approches globales

Les descripteurs globaux sont un moyen de traiter la nature globale de l'objet. Cela signifie que, plutôt que les détails de l'objet, plus d'importance est donnée à son aspect général.

- **Histogramme de cordes**

Cette approche proposée par Paquet et Rioux [Paquet1997] [Paquet2000] repose sur des statistiques construites à partir des cordes reliant les centres de gravité des triangles du maillage et le centre de gravité de l'objet 3D. En vue d'acquérir une mesure efficace, Les auteurs utilisent deux descriptions distinctes. La première est construite sur l'angle de la normale de chacune des faces, alors que la seconde est basée sur la direction des cordes entre le centre de masse de

l'objet et ses faces. Avant tout calcul, une phase de prétraitement est requise pour garantir un comportement invariant géométriquement. Pour cela, ils ont utilisé l'ACP afin de normaliser les données.

Les auteurs construisent ensuite trois histogrammes :

- Un histogramme des longueurs des cordes.
- Un histogramme des angles entre les cordes et le premier axe principal.
- Un histogramme des angles entre les cordes et le deuxième axe principal.

Grace à sa rapidité et son pouvoir discriminant pour les formes simples. Cette méthode est intéressante est pertinente uniquement lorsque les objets sont uniformément maillés tandis qu'elle n'est pas robuste vis-à-vis de perturbations sur la connectivité du maillage comme l'absence de face, face de tailles différentes …

- **Distributions de forme**

Afin de permettre de mesurer les propriétés géométriques d'un objet 3D polygonal, Osada et al. [Osada2001] présentent une démarche qui propose des méthodes probabilistes. Pour indexer un objet 3D, La gamme des fonctions possibles est très large. Les auteurs évaluent cinq distributions de forme basées sur des ensembles aléatoires de point.

1. A3 : Mesure l'angle entre trois points de la surface.

2. D1 : Mesure la distance entre un point fixe et un point du modèle (utilisent le centre de gravité autant que point fixe).

3. D2 : Mesure la distance entre deux points aléatoires de la surface.

4. D3 : Mesure la racine carrée de l'aire d'un triangle formé par trois points aléatoires.

5. D4 la racine cubique du volume du tétraèdre formé par quatre points aléatoires.

Ces fonctions de forme ont été choisies pour leur simplicité de calcul et leur invariance. Après analyse des cinq méthodes, les auteurs déduisent que la distribution de la distance entre les paires de points aléatoires (D2) donne les meilleurs résultats par rapport aux autres fonctions. La fonction D2 représente sous forme d'un histogramme normalisé les probabilités d'occurrence d'une distance entre deux points choisis aléatoirement sur les faces triangulaires du modèles 3D, Les faces triangulaires étant elles-mêmes prises au hasard (Voir figure 2.8).

Ce descripteur est rapide à calculer, facile à implémenter, robuste aux transformations (Rotations, translations, bruit, miroir, suppression, et insertion). Pour obtenir une robustesse à

la mise à l'échelle, les auteurs choisissent de normaliser les distributions de formes. Les résultats obtenus avec cette approche permettent de distinguer la forme globale de l'objet sans avoir un pouvoir discriminant suffisant pour capturer les petites variations du maillage. La méthode parait la plus adaptée aux recherches d'objets similaires dans des bases de données d'objets de formes très différentes.

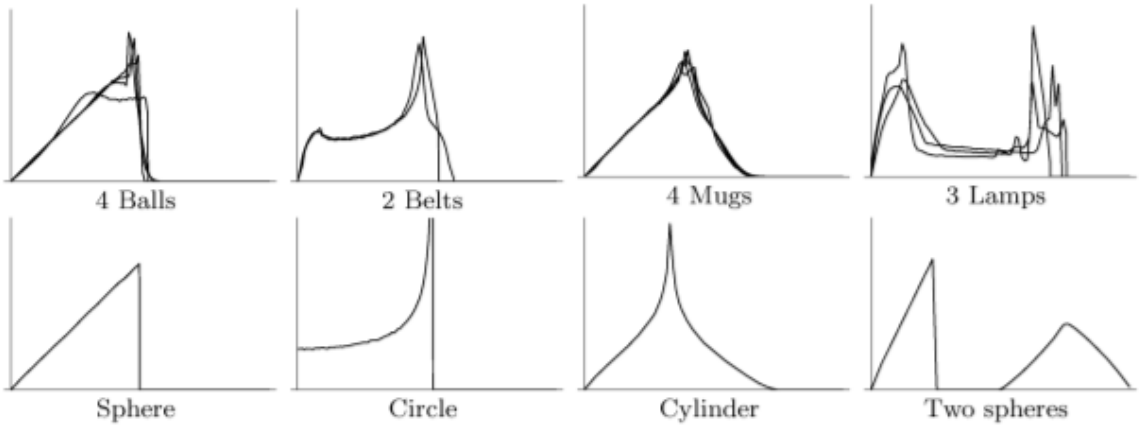


Figure 2. 8: Les distributions de forme D2 indiquent la forme globale d'un objet.

2.4.2 Approches locales

Les approches locales visent à utiliser des mesures d'entités locales sur la surface représentant la forme 3D. Ces caractéristiques peuvent être comprises comme des informations du modèle 3D quand elles sont observées et analysées de tout près.

- **Images gaussiennes étendues**

Notée souvent « EGI » pour Extended Gaussian Images, cette approche est développée par Horn [Horn1984]. Elle cherche à synthétiser l'information de chaque facette du maillage 3D triangulé en se basant sur une fonction sphérique, dont le but est de caractériser l'information d'orientation des points d'une surface 3D. En effet, pour calculer le descripteur, les auteurs construisent un histogramme défini sur un ensemble discret d'orientations couvrant la sphère unité (appelé sphère de Gauss). Chaque composante de l'histogramme est donnée par l'aire des facettes ayant des orientations appartenant à l'intervalle angulaire associé.

Dans l'intention d'améliorer cette approche, Kang et Ikeuchi [Kang1993] et par la suite Wang et al. [Wang2007] ajoutent une information de distance dans leur descripteur. Cette nouvelle donnée permet d'améliorer la pertinence des résultats.

L'un des principaux inconvénients de EGI et ses nouvelles versions est sa forte dépendance à l'orientation des facettes, elle reste sensible à l'orientation définie par la topologie des objets 3D. En générale elle n'est pas invariante aux transformations géométriques. Pour cela une étape de normalisation est nécessaire. Il est préférable de les utiliser en plus d'autres méthodes plutôt que seules.

- **Descripteur de spectre de forme 3D**

Cette approche proposée par Zaharia et Prêteux [Zaharia2002], est adoptée comme descripteur de forme pour le format standard MPEG-7. Il s'agit d'un descripteur caractérisant les courbures locales de la surface des objets 3D. Ce descripteur est fondé sur la notion d'index de forme de Koen [Koen1992] basé sur les courbures.

L'index de forme est défini comme la valeur de la coordonnée angulaire de la représentation polaire du vecteur des courbures principales. Soient p un point d'une surface régulière et k_p^1 et k_p^2 avec $k_p^1 \geq k_p^2$ les courbures principales au point p. L'indice de forme I_ps'exprime par:

$$I_p = \frac{1}{2} - \frac{1}{\pi}\arctan\frac{k_p^1 + k_p^2}{k_p^1 - k_p^2} \qquad (2.1)$$

L'indice de forme prend ses valeurs dans l'intervalle [0,1], il n'est pas défini pour les surfaces planes, mais il est invariant aux transformations euclidiennes et aux homothéties. Il permet donc de caractériser différentes formes élémentaires. La figure 2.9 illustre certaines surfaces bien connues et leurs valeurs sur la définition de l'intervalle d'indice de forme.

Le spectre de forme du maillage 3D est l'histogramme des indices de forme calculés sur l'ensemble du maillage 3D.

Ce descripteur est très compact. Mais malheureusement, il souffre d'une limitation qui réside dans sa grande sensibilité à la topologie. De plus, cette méthode requiert des maillages réguliers avec des normales bien orientées.

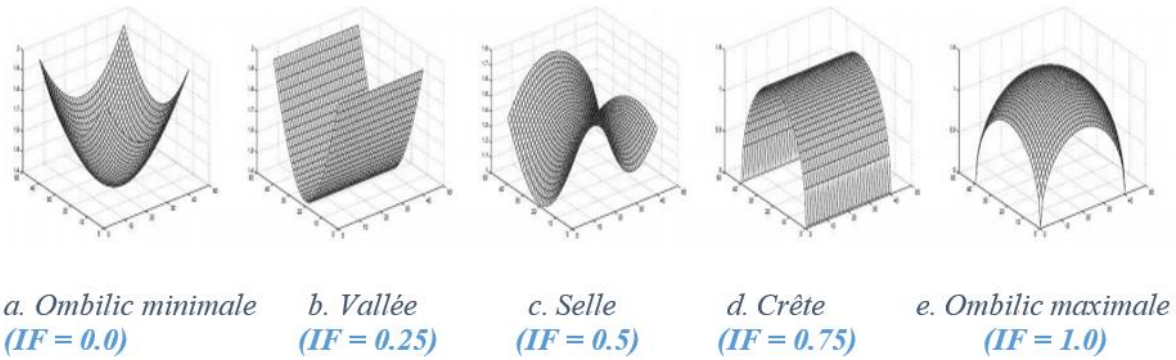


Figure 2. 9: Exemples d'indices de forme(IF) calculés sur cinq formes élémentaires.

- **Local spin Images**

Johnson et Hebert [Johnson1999] ont développé et utilisés cette approche pour l'adaptation des surfaces et la reconnaissance des objets dans une scène 3D. L'élément clé de la production d'une image spin est l'utilisation des points orientés qui sont des points de la surface 3D avec une direction associée. La surface à laquelle correspondent ces points est représentée comme une maille polygonale de sommet. Un point orienté d'un sommet d'une surface de maillage est définit par une position 3D (notée p) et un vecteur normal (n), avec (n, p) nous permet de définir une base 2D, qui correspond à un système de coordonnées local.

Lors de l'étude comparative. [As'ari2014] As'ari et all. ont déduit que local spin image peut fournir des propriétés locales de différentes images de profondeur avec un degré élevé de similitude tandis que les autres souffrent de la dégradation des performances en raison des variations de forme globale. L'inconvénient de cette approche peut apparaître lorsque nous avons différentes instances par classe.

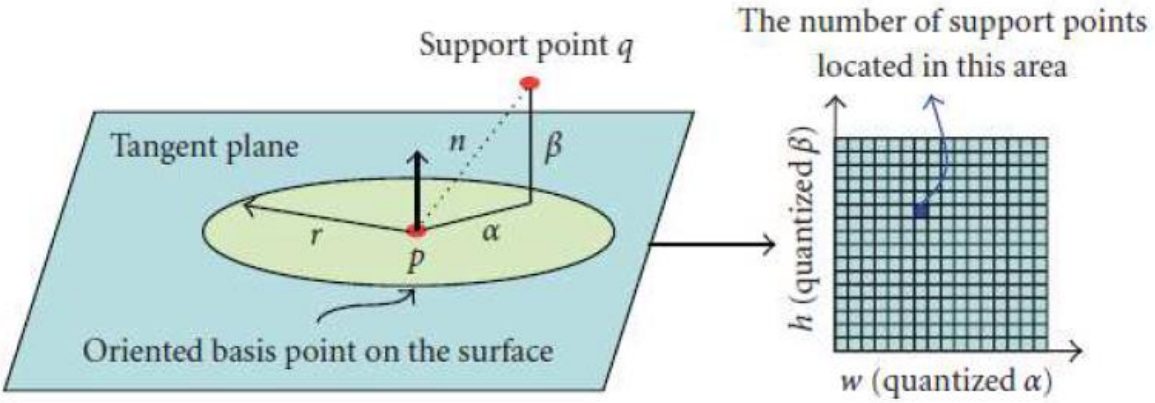


Figure 2. 10: La démonstration de Spin Image (la figure est prise de [Xiaolan2010]).

- **Carte de courbures**

Présentée par Assfalg et all. [Assfalg2003], cette approche aborde le problème de la récupération des modèles 3D basés sur le contenu. En se basant sur l'idée que la forme d'un objet 3D peut être décrite par une carte de courbures de sa surface. Après un traitement préalable du modèle. Pendant lequel les propriétés différentielles de la surface d'objet 3D sont calculées. La surface du modèle est ensuite déformée pour adopter la surface d'une sphère, tout en conservant les informations lors de la déformation. La projection du maillage sur la sphère avec la donnée de courbure associée aux sommets forment la carte de courbure de l'objet 3D. La similarité entre deux objets est calculée en comparant leurs cartes de courbures à l'aide des descripteurs qui tiennent en compte la surface des régions et de leur arrangement spatial.

Cette approche a été mise en œuvre dans un système prototype qui soutient l'extraction par le contenu d'objets 3D à travers une interface Web. Les résultats obtenus par cette approche sont meilleurs qu'une méthode qui utilise les histogrammes de courbures calculés directement sur les objets 3D.

2.4.3 Approches basées sur la transformée de la forme

Les méthodes par transformées visent à déterminer des représentations de forme globales, définies en terme de transformation intégrale. De nombreuses techniques existent dans la littérature. L'aspect multi-résolution des vecteurs caractéristiques et le gain en détails lorsque nous prenons en compte un plus grand nombre de coefficients sont les principaux avantages de ce type d'approche. Nous nous sommes principalement intéressés dans cette partie, aux méthodes basées sur la transformée de Hough 3D, transformée de Fourier et aux approches basées sur les moments.

- **Transformée de Hough 3D**

Zaharia et Prêteux [zaharia2001] présentent dans leur approche un nouveau descripteur de forme, Afin de représenter et récupérer les similitudes des modèles 3D polygonaux uniformément maillés. Elle est fondée sur un principe d'accumulation des points sur des plans de R^3. Chaque plan est représenté en coordonnées sphériques par le triplet (r, θ, φ) où $r \geq 0$ est sa distance à l'origine, $\theta \epsilon [0,2\pi[$ l'angle d'azimut et $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right[$ l'angle d'élévation. En échantillonnant uniformément l'espace des paramètres sphériques, un ensemble de $N_r \times N_\theta \times N_\varphi$ composantes est créé pour Définir l'histogramme de Hough3D (voir figure 2.11). Ce

descripteur est intrinsèquement invariant aux problèmes de représentation de la connectivité, mais pas à la transformation géométrique.

Zaharia et Preteux [zaharia2002] proposent une nouvelle représentation, fondée sur un descripteur de Hough 3D optimal (DH3DO) qui permet de s'affranchir des problèmes d'alignement spatial par un échantillonnage de la sphère unité invariant aux repères issus d'une analyse en composantes principales de l'objet 3D. Cela permet de définir des mesures de similarité intrinsèquement symétriques, qui conduisent à une réduction importante du temps de calcul des réponses aux requêtes.

En comparant les deux descripteurs le deuxième s'est révélé plus performant sur la base de MPEG-7 et plus stable en terme de topologie.

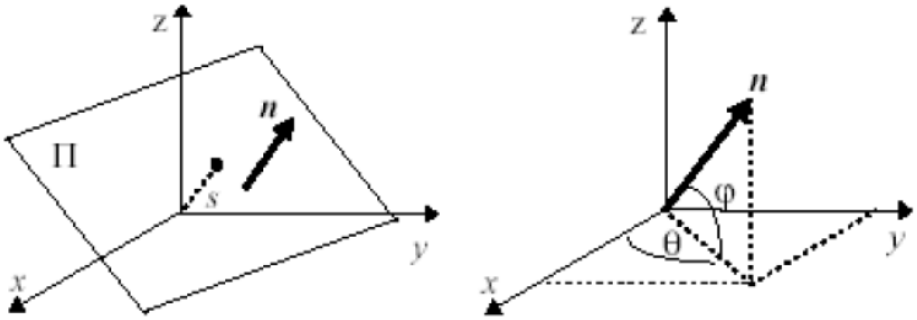


Figure 2. 11: Passage aux coordonnées sphériques pour la transformation de Hough.

- **Transformée de Fourier 3D discrète**

Cette approche présentée par Saupe [Saup2001] [Dutagaci2005] apporte un nouveau descripteur de forme 3D dans le cadre de l'indexation par le contenu. Ce descripteur est invariant aux translations, changements d'échelles, rotations, réflexions et peu sensible aux changements de résolution. Les auteurs utilisent une analyse en composante principale continue comme prétraitement de l'extraction du descripteur sur l'objet voxelisé, dont le but est d'obtenir l'invariance aux transformations de l'espace. Les coefficients de Fourier sont calculés selon la formule:

$$f'_{pqr} = \frac{1}{\sqrt{N^3}} \sum_{a=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{b=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{c=-\frac{N}{2}}^{\frac{N}{2}-1} f_{pqr} e^{-j2\pi(pa+qb+rc)/N} \qquad (2.2)$$

Où $-N/2 \leq a, b, c, p, q, r \leq N/2$ et f_{pqr} est l'ensemble des voxels caractérisant l'objet3D. Les normes des coefficients $|f'_{pqr}|$ vérifiant $1 \leq |p| + |q| + |r| \leq k \leq N/2$ constituent le

descripteur de Fourier. En effet, le terme $|p| + |q| + |r|$ permet de limiter une partie des coefficients à cause de la symétrie hermitienne de la transformée de Fourier et la borne k permet de retenir seulement les basses fréquences pour leur pouvoir discriminant.

Le problème majeur est que l'espace d'application doit être continu, ce qui implique que les maillages doivent être réguliers, et donc les objets complexes ne peuvent pas être bien décrits. Ajoutant le fait que les descripteurs de Fourier ne sont pas très discriminants lorsque le nombre de coefficients est trop faible, et de plus les coefficients des hautes fréquences sont très sensibles aux bruits ou aux petites variations de la connectivité du modèle 3D de l'objet.

- **Moments géométriques**

Introduite pour la première fois par Paquet et Rioux [Paquet1999], cette approche se focalise sur les objets 3D maillés. Les auteurs calculent les moments géométriques directement sur les points du maillage de l'objet. Cette méthode fut améliorée par Saupe et Vranic [Saupe2001]. La phase de prétraitement est nécessaire. Donc l'objet 3D est centré, mis à l'échelle, et aligné en utilisant une analyse en composantes principales continue. Cette approche permet le calcul des moments sur la surface discrétisée de l'objet en utilisant la formule :

$$f(x,y,z) = \iiint_{\Omega} x^p y^q z^r f(w,y,z) dx dy dz \;\; p,q,r = 0,1,2,3,4 \ldots \qquad (2.3)$$

Une étude comparative en terme de performance, réalisée par Vranic [Vranic2004] sur différentes bases d'objet 3D, montre que l'utilisation des moments géométriques n'est pas trop conseillée, elle place cette méthode la dernière.

- **Moments de Zernike 3D**

Définie par Canterakis [Canterakis1999] et utilisée par Novotni et Klein [Novotni2004], cette approche permet le calcul de façon discrète sur un objet discrétisé et normalisé, en combinant linéairement les moments géométriques. Les auteurs procèdent à l'extraction du descripteur sur l'objet voxelisé en quatre étapes distinctes :

2 Centrage et mise à l'échelle du modèle après calcul du centre de gravité.

3 Calcul des moments géométriques $m_{p,r,s}$ d'ordre : q, r, s > 0 et q + r + s ≤ N.

4 Calcul des moments de Zernike 3D $m_{p,r,s}$ avec :

$$m_{p,r,s} = \frac{3}{4\pi} \sum_{\mathrm{q+r+s \le N}} X_{nlm}^{qrs} M_{qrs} \qquad (2.4)$$

où X_{nlm}^{qrs} représente la combinaison linéaire des moments géométriques.

5 Calcul des normes $F_{nl}=\|m_{p,r,s}\|$ comme descripteur de l'objet.

La méthode proposée est invariante aux translations, aux rotations, aux symétries et aux changements d'échelle. Afin d'évaluer leur méthode les auteurs ont effectué des expérimentations sur la base du Princeton Shape Benchmark [Princeton2004]; ils ont prouvé que leur méthode donne des résultats équivalents ou meilleurs que les approches par harmoniques sphériques.

2.4.4 Approches structurelles

Contrairement à d'autres approches, les représentations structurelles visent à décrire la notion de forme de manière plus complète et intuitive. Un premier type d'approche repose sur une segmentation initiale de l'objet en sous-parties satisfaisant certains critères d'homogénéité par rapport à un attribut de forme préétabli et représenté par des structures spécifiques telles que des arbres ou des graphiques. Si l'étape de segmentation vise à identifier les différentes structures élémentaires qui constituent l'objet considéré, la seconde phase permet de représenter les relations d'adjacence et éventuellement les positions relatives de ces différentes structures. Nous allons dans cette section citer des exemples parmi les méthodes de ce type d'approche.

- **Amincissement de volume**

Cette approche proposée par Sundar et al. [Sundar2003] cherche à décrire la forme d'un modèle 3D par son squelette. Pour cela, les auteurs voxelisent ce dernier et amincissent cette représentation par une transformée de distance [Gagvani1999]. Les attributs d'amincissement permettent de réduire le volume en voxels par couches successives. Afin de créer le squelette final de l'objet, Ils transforment les points obtenus en un graphe acyclique en appliquant l'algorithme de l'arbre couvrant minimum. En vue de comparer les squelettes de deux modèles 3D, et dans l'intention d'éviter un calcul trop coûteux, Les auteurs se reposent sur un algorithme de recherche de la mise en correspondance de coût minimal dans un graphe biparti (voir figure.2.12).

Figure 2. 12 : Exemples de mise en correspondance. Seulement les nœuds mis en correspondance sont montrés

Cette approche permet d'extraire des informations intrinsèques de la forme générale des objets 3D, et fournit des résultats robustes spécifiquement dans la comparaison d'objets dit "articulés". Il est aussi intéressant de noter qu'avec cette méthode il est possible de faire des recherches partielles.

- **Graphe de Reeb**

Reeb [Reeb1946] définie une approche qui permet d'obtenir une représentation de type squelette en conservant la structure topologique des objets (voir figure 2.13). Sa création repose sur la théorie de Morse [Shinagawa1991] qui, à partir d'une fonction continue μ définie sur une surface fermée, caractérise la topologie de la surface en ses points critiques. Le graphe s'obtient à partir d'une segmentation de la surface de l'objet obtenue selon les valeurs de la fonction μ, Les nœuds du graphe Reeb représentent les composantes connexes d'une même partition, alors que les arrêtes permettent de relier les nœuds adjacents selon une fonction μ. Il existe différentes fonctions continues μ [Biasotti2008] qui peuvent être utilisées dans la construction du graphe de Reeb. En effet l'aspect du graphe résultant est entièrement lié au choix de la fonction μ, pour cela il est important de choisir soigneusement cette fonction.

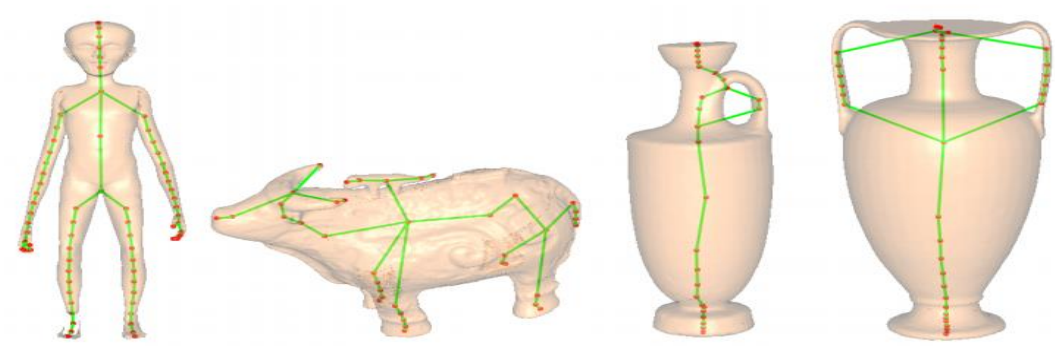


Figure 2. 13: Modèles 3D et leur graphe de Reeb**.**

En vue d'améliorer cette approche, plusieurs améliorations ont été apportées par Tung et Schmitt [Tung2005] et Biasotti et al. [Biasotti2003] qui proposent d'enrichir les nœuds du graphe par des attributs supplémentaires. Par exemple des attributs topologiques ou géométriques. Grâce à ceci, la mise en correspondance du graphe est clairement améliorée et permet d'obtenir des résultats très pertinents pour les bases de données d'objets dits "articulés". (Voir figure 2.14).

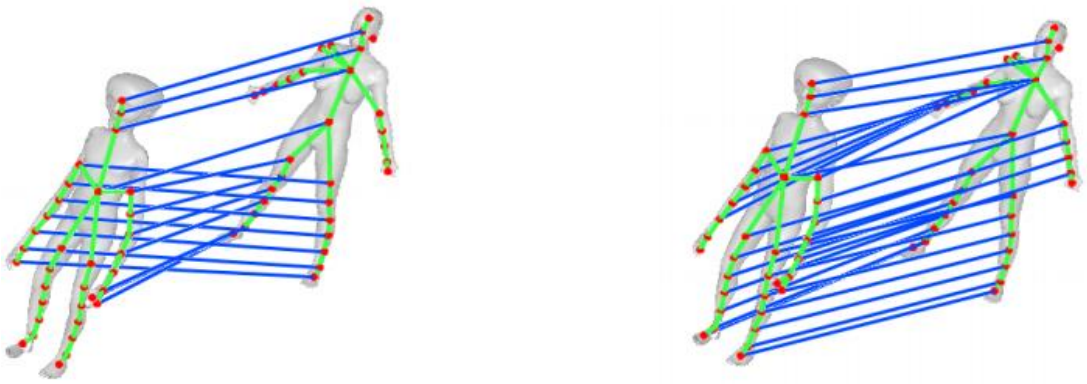


Figure 2. 14: Comparaisons des résultats sans l'information géométrique (à gauche), les jambes peuvent être appariées aux bras car ils sont topologiquement équivalents et en ajoutant l'information géométrique (à droite), tous les membres sont bien appariés.

- **Graphe des formes élémentaires 3D**

Cette approche proposée par Medioni et François [Medioni2002] utilise une modélisation par sous-parties Volumiques des objets 3D à base de formes géométriques élémentaires appelées Geons. Une fois l'objet 3D est décomposé en geons, ces éléments sont hiérarchiquement organisés au sein d'un arbre de description, intégrant également des relations d'adjacence. Cette structure hiérarchique permet de définir et de calculer efficacement une mesure de similarité fondée sur un coût de transition issu d'une méthode d'appariement par graphes. Irani et Ware [Irani2003] transforment le graphe de geons 3D en un diagramme UML et comparent par la suite les diagrammes UML entre eux.

2.5 Conclusion

Dans un système de reconnaissance d'objets, l'extraction des caractéristiques utilisées pour la reconnaissance jouent un rôle crucial. Les caractéristiques extraites doivent être suffisantes pour représenter et pouvoir identifier l'objet. Le présent chapitre donne un aperçu des descripteurs développés pour extraire l'information de forme des objets 3D.

Trois approches sont distinguées dans le monde d’indexation et de reconnaissance d’objet 3D. Dans cette thèse, nous nous sommes focalisés sur les deux approches : L’indexation 2D/3D et l’indexation 3D/3D.

Dans le chapitre qui suit, nous nous sommes concentrés sur une méthode d’indexation 2D/3D. Ce choix est dû à sa flexibilité, elle permet de faire des requêtes à partir de diverses sources, à savoir : un modèle 3D, un ensemble de photos. La méthode choisit caractérise un objet 3D par un ensemble de vues 2D et les classe en comparant leurs ensembles de vues. Pour chaque objet 3D, un certain nombre de vues est créé et indexé par un descripteur de forme 2D. Cela nous garantit les possibilités de comparer soit directement avec une image requête soit avec un autre objet 3D en comparant leurs ensembles de vues.

Chapitre 3 : La classification des objets 3D en se basant sur l’approche 2D/3D

3.1 Introduction

La recherche d'objets 3D, dans les grandes bases de données, nécessite de mettre en place des nouveaux outils permettant l'analyse et l'accès rapide à ces objets. De ce fait, l'indexation de contenus offre une solution efficace en sélectionnant les données caractérisant le modèle 3D et en les représentant dans une représentation synthétique, populairement nommée signature ou descripteur. Lors de la création de ce descripteur, qui permet de stocker les caractéristiques d'un modèle 3D, une attention particulière doit être apportée à la conception d'une description discriminante et qui garantit la recherche rapide et efficiente.

Ce chapitre est consacré aux études que nous avons menées en classification et en reconnaissance des objets 3D en se basant sur l'indexation 2D/3D. Ce choix émane de notre conviction de l'importance de ce domaine de recherche. Le processus sur lequel nous nous sommes basés lors de nos travaux de recherche a pour objectif l'interrogation d'une base de données d'objets 3D. Pour cela, nous proposons une méthode de classification de modèles 3D basée sur des vues 2D. Cette méthode vise à fournir une sélection de vues 2D à partir d'un modèle 3D en utilisant des projections orthogonales et une méthode de transformation de Fourier rapide (FFT) pour la description des formes 3D issues de ces vues. Cette description nous permettra de définir les vecteurs caractéristiques basés sur une extraction carrée des spectres de Fourier. En ce qui concerne la classification, nous nous sommes basés sur l'utilisation de l'algorithme de reconnaissance de nuages de points $P ($P Point-Cloud Recognizer (PCR)) [Vatavu2012]. $P évite la complexité du stockage en représentant les objets sous forme de nuages de points et en ignorant le comportement variable des données en termes d'ordre et de direction.

Le but de l'approche que nous avons développé est de pouvoir retrouver, à partir d'une ou plusieurs vues 2D d'un objet, l'objet correspondant dans une bibliothèque d'objets 3D. En effet, l'utilisateur ne possède pas forcement une copie de l'objet recherché. Pour cela, nous lui garantissons une interrogation variée. Notre programme prend en argument une ou plusieurs images ou même directement un objet 3D. Il compare ces données à la base de données des objets 3D. C'est-à-dire, que même lorsque nous comparons deux modèles 3D, la comparaison se fait via des projections en 2D de ces objets. Cela permet avec la même méthode d'obtenir un moyen de soumettre, soit des images 2D soit un modèle 3D. Mais surtout, ce passage par la 2D permet une comparaison rapide des objets.

Ce chapitre est divisé en plusieurs sections. Dans la section 2, nous présentons les principes de base de notre méthode d'indexation 2D/3D, ainsi que les moyens techniques permettant de transformer un objet 3D en vues 2D. Ensuite dans la section 3, nous définissons les différentes méthodes de classification. Puis dans la section 4, nous présentons la solution proposée pour la mise en œuvre de notre système. Finalement, Les résultats expérimentaux visant à évaluer l'efficacité et la robustesse de notre système seront au cœur de la section 5. Enfin nous tirons notre conclusion.

3.2 Indexation 2D/3D

3.2.1 Projection 3D-2D

La projection du modèle 3D en images 2D représente une phase clé du processus d'indexation 2D/3D. Le modèle de maillage M est projeté et rendu en 2D à partir de différents angles de vue de N_P (positions de la caméra virtuelle dans l'espace 3D) (Figure 3.1), ce qui donne un ensemble de projections de N_P, noté P_i (M), avec i = 1… N_P

La projection peut être une image binaire (la silhouette de l'objet) ou une image en niveaux de gris représentant la carte de profondeur (Figure 3.1 b et c). Cependant, seule la représentation de la silhouette permet de faire la correspondance entre les contenus 2D et 3D, car aucune information de profondeur n'est disponible dans les images 2D [Raluca2013].

Une direction de vue $\{n_i\}$ est associée à chaque angle de vue; elle représente la direction de la droite qui relie la position de la caméra à l'origine du système cartésien (après la normalisation, elle coïncide avec le centre du modèle 3D).

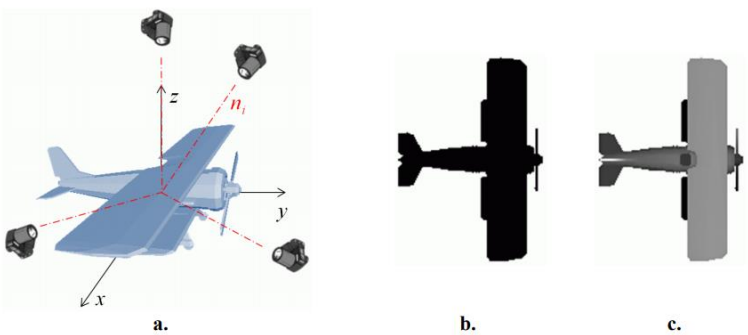


Figure 3.1: Projection 3D-2D.
a. directions d'observation; b. exemple d'image de silhouette; c. exemple d'image en profondeur

Lors de la projection d'un modèle 3D sur un plan bidimensionnel (indexation 2D/3D), les principaux aspects à prendre en considération sont: le nombre de vues et la spécification des angles de vue.

3.2.2 Le choix du nombre des vues

Le choix du nombre de vues (N_p) est un aspect qui affectera l'efficacité et la rapidité de ces méthodes. Ce nombre est directement lié aux performances du système. Il faut le réduire au minimum pour avoir une méthode rapide, tout en conservant la capacité descriptive des vues.

Deux catégories de problèmes peuvent apparaître et rendre difficile l'analyse des caractéristiques :

- La sous-indexation : En effet, l'extraction d'un nombre trop réduit de silhouettes ne permettant pas une durabilité des propriétés indispensables à la bonne analyse du contenu de celui-ci. Ce qui rendra difficile, voire impossible, l'indexation, et conséquemment la recherche, de l'objet 3D.
- La sur-indexation : C'est le cas inverse, par peur de perdre les informations nécessaires pour une description adéquate d'un objet 3D, nous capturons un nombre de silhouettes trop élevé. Dans ce cas, l'information est présente, mais difficile à extraire car elle est noyée dans la multitude d'informations non pertinentes.

Un grand nombre de vues assure une bonne description du modèle 3D, cela convient le mieux à l'indexation et la récupération. Cependant, les aspects informatiques associés doivent être pris en considération. Le temps requis pour la projection et l'extraction des descripteurs, ainsi que les besoins en mémoire/stockage, sont proportionnels au nombre de vues. Par conséquent, un équilibre doit être assuré entre le niveau de détail de la représentation 2D/3D et les coûts de calcul impliqués.

Quelle que soit la stratégie de sélection du choix des vues adoptée, les descripteurs de forme 2D utilisés dans la description des silhouettes, influencent fortement le pouvoir discriminant de la représentation. Les aspects liés aux descripteurs sont abordés dans la section suivante.

3.2.3 Descripteur de forme 2D

Les descripteurs sont des représentations mathématiques des principales caractéristiques du contenu multimédia qui permettent une comparaison objective et quantitative entre divers

objets. Pour les besoins de correspondance 2D-3D, la forme est reconnue comme étant l'une des principales caractéristiques qui décrivent le contenu d'un objet.

Nous rappelons tout d'abord les différents critères cités dans la littérature, afin qu'un descripteur de forme soit satisfaisant [Zaharia2004, Tangelder2004]:

- Portée: le descripteur devrait pouvoir caractériser tout type de forme.
- Unicité: une forme donnée est décrite par un seul descripteur et un descripteur donné correspond à une seule forme.
- Efficacité: pour une base de données volumineuse, le système doit pouvoir décrire rapidement les modèles et effectuer une extraction rapide. Par conséquent, l'extraction des fonctionnalités est rapide et complexe. Certains descripteurs permettent le rejet précoce de modèles non similaires basés sur un sous-ensemble de fonctionnalités. Cette capacité est utile pour accélérer le processus d'appariement.
- Robustesse: le descripteur doit être presque insensible au bruit et aux petites fonctionnalités supplémentaires.
- Sensibilité: un descripteur doit présenter la capacité de décrire et de prendre en compte même les détails les plus fins de la forme.
- Pouvoir de discrimination: le descripteur devrait pouvoir saisir les propriétés qui distinguent le mieux la forme.
- Prise en charge multi-résolution: le descripteur ne devrait pas dépendre de la résolution de la forme.
- Possibilité d'effectuer une correspondance partielle: une telle fonctionnalité est utile dans le cas de formes incomplètes, par exemple lorsqu'une partie de l'objet est invisible.
- Invariance géométrique et topologique: la description d'un objet donné ne doit pas dépendre de l'échelle, de l'orientation ou de la position qu'il a dans l'image.
- Accord avec la perception humaine: il est important que les similitudes données par un descripteur correspondent à la perception humaine.
- Possibilité de faire correspondre des objets articulés: un descripteur doit extraire des caractéristiques similaires pour différentes instances d'un objet articulé.
- La taille de stockage du descripteur: une propriété importante, en particulier dans le cas de grandes bases de données où un grand nombre de descripteurs doivent être stockés.

Ces critères sont pris en compte de différentes manières par les différentes approches. Les descripteurs de forme 2D peuvent être scindés en deux familles principales:

- Descripteurs de forme basés sur les régions: dans ce cas, les informations d'entrée qui sera décrite représentent la région de support de la forme 2D (Figure 3.2b).
- Descripteurs de forme basés sur les contours: cette catégorie ne peut pas détecter la structure interne de la forme, seules les informations de contour externes sont conservées (Figure 3.2c). Par conséquent, elle est limitée à un certains types d'applications [Tabbone2006]. Elle n'est pas adaptée aux formes complexes ou creuses (par exemple, des trous, des composants connectés multiples ...).

a. **b.** **c.**

Figure 3.2: Différentes représentations d'une forme 2D.
a. L'objet 3D; b. représentation par région; c. représentation par contour.

Nous avons déjà présenté et discuté un état d'art sur les différents descripteurs de formes 2D dans le chapitre 2 plus précisément la section 2.2. Le choix du descripteur constitue un problème majeur dans les systèmes d'analyse d'images, car le descripteur conditionne fortement le résultat final de la recherche et de la classification.

3.3 Classification

La classification est une technique d'exploration de données, généralement basée sur l'apprentissage automatique (ML pour machine Learning). Elle est considérée comme un processus permettant d'obtenir les informations à partir des exemples disponibles. Le type d'apprentissage automatique le plus utilisé est l'apprentissage dit "supervisé" qui permet à la machine d'apprendre ses paramètres en utilisant une base de données annotée. Par exemple, dans le cadre de la classification d'images, un modèle entraîné grâce à l'apprentissage supervisé prédit le type d'objet (sa classe) présent dans l'image. Cette section ne cherche pas à présenter une étude exhaustive des méthodes de classification, mais elle serait restreinte aux méthodes utilisées dans les travaux de cette thèse. De ce fait, nous discutons trois algorithmes d'apprentissage automatique. Plus proche voisin (K-PPV ou KNN pour *K-Nearest-Neighbor*),

Machine à vecteurs de support (SVM pour *Support Vector Machines*) et Réseau de neurones artificiel (RNA ou ANN pour Artificial Neural Network).

3.3.1 Plus proche voisin

Le plus proche voisin est considéré comme l'algorithme le plus simple des méthodes d'apprentissage automatique à base d'instances. Il est souvent utilisé dans les applications de reconnaissance de formes et d'exploration de données. Cet algorithme ne nécessite pas d'apprentissage mais simplement le stockage des données d'apprentissage. Lors de l'utilisation du k-NN, un objet est classé selon le vote majoritaire de ses voisins [Li2008]. En général, l'algorithme k-NN est traité comme une méthode de classification basée sur l'exemple d'entrainement le plus proche du vecteur caractéristique [Teli2014]. Son principe est très simple, nous lui fournissons un ensemble de données d'apprentissage **D**, une fonction de distance **d** et un entier **k.** Pour tout nouveau point de test **X**, pour lequel il doit prendre une décision, l'algorithme recherche dans **D** les **k** points les plus proches de **X** au sens de la distance **d**. Finalement, il attribue **X** à la classe qui est la plus fréquente parmi ces **k** voisins. La valeur de k est déterminée en fonction de la taille des données utilisées. Si k = 1, l'objet est simplement affecté à la classe de son plus proche voisin. Le choix de grandes valeurs de k permet de lisser le bruit qui peut exister dans l'échantillon d'étude. Cependant, il rend les limites entre les classes moins différentes.

En général, les étapes de l'algorithme sont données comme suit :

- Étape 1: déterminer k
- Étape 2: calculer les distances entre la nouvelle entrée et toutes les données d'apprentissage
- Étape 3: trier la distance et déterminer k les plus proches voisins en fonction de la distance minimale k-ième
- Étape 4: rassembler les catégories de ces voisins
- Étape 5: déterminer la catégorie en fonction du vote majoritaire.

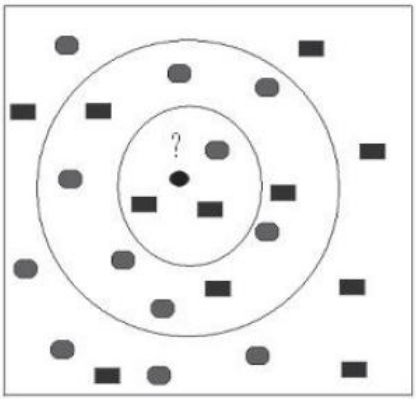


Figure 3.3: Exemple de classification du k-plus proche voisin.

La figure 3.3 illustre la fonctionnalité de la classification du k plus proche voisin. Le modèle test (ellipse) indiqué par un point d'interrogation doit être classé soit dans la classe du modèle indiquée par les rectangles ou dans l'autre classe du modèle indiquée par des rectangles arrondis. Si la valeur de k est 3, le modèle test est classé dans la classe rectangle car il y a 2 rectangles et un rectangle arrondi à l'intérieur du petit cercle. Si la valeur de k est 12, le modèle test est classé dans la classe rectangle arrondi car à l'intérieur du grand cercle se trouve 7 rectangles arrondis et 5 rectangles.

Afin de mettre en œuvre l'algorithme k-plus proche voisin en tant que classifieur permettant de reconnaître les objets, la métrique de distance joue un rôle important. Différentes métriques de distance sont disponibles, telles que la distance euclidienne, la distance de Manhattan, la distance de Chebychey, la distance de Hamming, etc. Parmi ces différentes distances disponibles, la distance euclidienne est plus facile à calculer et prometteuse. C'est un calcul simple et rapide.

La formule de la distance euclidienne entre un point $X(x_1, x_2, etc.)$ et un point Y$(y, y_2, etc.)$ est impliquée dans le calcul de la racine carrée de la somme des carrés des différences entre les valeurs correspondantes, elle est donnée dans l'équation (3.1).

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (3.1)$$

L'équation (3.1) est applicable pour des dimensions supérieures et même peut être appliqué à deux dimensions et à trois dimensions.

Pour les espaces euclidiens à deux dimensions, la distance entre les deux points X (x1, y1) et Y (x2, y2) est calculée en se basant sur l'équation (3.2).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (3.2)$$

Pour les espaces euclidiens à trois dimensions, la distance entre les deux points X (x_1, y_1, z_1) et Y (x_2, y_2, z_2) est calculée en se basant sur l'équation (3.3).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \qquad (3.3)$$

L'algorithme KNN reste une méthode facile à comprendre et souvent performante. De plus, son apprentissage est assez simple et rapide. Il est adapté aux domaines où chaque classe est représentée par plusieurs prototypes et où les frontières sont irrégulières. Cependant, son temps de prédiction est très long car il faut revoir tous les exemples à chaque fois, ce qu'implique un coût de classification élevé. On note aussi qu'il est sensible aux attributs non pertinents et corrélés.

3.3.2 Machine à vecteurs de support

SVM ou séparateurs à vaste marge est l'une des techniques d'apprentissage automatique supervisé, introduite pour la première fois par Cortes et Vapnik [Cortes1995]. Ils sont destinées à résoudre des problèmes de discrimination (c'est-à-dire décider à quelle classe appartient un échantillon) ou de régression (c'est-à-dire prédire la valeur numérique d'une variable). Les SVM sont une généralisation des classifieurs linéaires [Chen2010], très utilisées dans la reconnaissance de forme et la reconnaissance d'objet. L'objectif du SVM est de former un hyperplan en tant que surface de décision de manière à maximiser la marge de séparation entre les exemples positifs et négatifs en utilisant une approche d'optimisation. Généralement les fonctions linéaires sont utilisées comme hyperplans séparateurs dans un espace intermédiaire. Lors de l'utilisation des fonctions noyau, le produit scalaire peut être implicitement calculé dans l'espace des fonctions du noyau.

La fonction noyau est une astuce mathématique qui permet au SVM d'effectuer une classification en deux dimensions d'un ensemble de données unidimensionnelles. En général, une fonction noyau projette des données d'un espace de petite dimension vers un espace de dimension supérieure [William2006]. Il existe différentes fonctions noyau, nous pouvons citer les exemples suivants : linéaire, polynomial et radiale (RBF pour Radial basis function).

Le noyau linéaire: Si les données sont linéairement séparables, nous n'avons pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision. Il prend la forme de l'équation (3.4) :

$$K(x_i, x_j) = (x_i^T, x_j) \quad (3.4)$$

Le noyau polynomial : Il élève le produit scalaire à une puissance naturelle p. Il prend la forme de l'équation (3.5)

$$K(x_i, x_j) = (x_i^T, x_j)^p \quad (3.5)$$

Où p est le degré du polynôme

Le noyau radial : Un exemple des noyaux RBF est le noyau Gaussien. Il est donné dans l'équation (3.6).

$$K(x, y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (3.6)$$

Où, σ est un réel positif qui représente la largeur de bande du noyau.

Il existe deux manières d'adapter la classification SVM binaire en classification multi-classe [Muralidharan2012].

- **L'approche un contre tous** :

Selon la formulation de Vapnik [Vapni1998], elle consiste à déterminer pour chaque classe K un hyperplan $H_{k(w_k,b_k)}$ la séparant de toutes les autres classes. Cette classe k est considérée comme étant la classe positive (+1) et les autres classes comme étant la classe négative (−1), ce qui résulte, pour un problème à K classes, en K SVM binaires. Un hyperplan H_k est défini pour chaque classe k par la fonction de décision suivante :

$$H_k(x) = signe(\langle w_k, x\rangle + b_k) \quad (3.7)$$

$$= \begin{cases} +1 & si\ f_k(x) > 0; \\ 0 & sinon \end{cases}$$

La valeur retournée de l'hyperplan permet de savoir si x appartient à la classe k ou non. Dans le cas où il n'appartient pas à k $(H_k(x) = 0)$, nous n'avons aucune information sur l'appartenance de x aux autres classes. Pour le savoir, nous présentons x à tous les hyperplans, ce qui donne la fonction de décision de l'équation (3.8) suivante :

$$k^* = \underbrace{Arg}_{(1\leq k\leq K)} Max(H_k(x)) \quad (3.8)$$

- **L'approche un-contre-un**

Cette méthode revient à Knerr et ses co-auteurs [Knerr1990]. Elle consiste à utiliser un classifieur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, cette approche discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises. Pour chaque paire de classes (k, s), la méthode un-contre-un définit une fonction de décision binaire $h_{ks}: \Re \rightarrow \{-1, +1\}$. L'affectation d'un nouvel exemple se fait par liste de vote. Nous testons un exemple par le calcul de sa fonction de décision pour chaque hyperplan. Pour chaque test, nous votons pour la classe à laquelle appartient l'exemple (classe gagnante). Afin de réaliser, Nous définissons la fonction de décision binaire $h_{ks}(x)$ de l'équation 3.9.

$$H_{ks}(x) = signe\big(f_{ks}(x)\big) \qquad (3.9)$$

$$= \begin{cases} +1 & si\ f_{ks}(x) > 0; \\ 0 & sinon \end{cases}$$

Sur la base des $K(K-1)/2$ fonctions de décision binaires, nous définissons K autres fonctions de décision (équation 3.10) :

$$H_k(x) = \sum_{s=1}^{m} H_{ks}(x) \qquad (3.10)$$

Un nouvel exemple est affecté à la classe la plus votée. La règle de classification d'un nouvel exemple x est donnée par l'équation 3.11 :

$$k^* = \underbrace{Arg}_{(1 \le k \le K)} (MaxH_k(x)) \qquad (3.11)$$

L'avantage de l'utilisation de la méthode SVM par rapport aux autres méthodes est qu'elle peut être utilisée même si les exemples contiennent des attributs symboliques [Djeffal2012]. La rapidité d'entrainement de la méthode SVM et l'existence des méthodes d'accélération favorisent aussi son utilisation. Néanmoins, SVM reste sensible aux paramètres (Difficulté à identifier les bonnes valeurs des paramètres), et elle demande une capacité en mémoire et en temps de calcul.

3.3.3 Réseau neurones artificiels

Un réseau de neurone artificiel est un ensemble de nœuds interconnectés. Il est initialement introduit par [McCulloch1943]. Chaque nœud est appelé neurone, qui est une modélisation mathématique du neurone biologique. Le réseau neuronal artificiel doit traiter l'entrée (information) et produire la sortie ciblée à l'aide d'une fonction d'activation. Il consiste en une fonction mathématique appliqué à un signal et renvoyant une valeur d'activation. En considérant un signal d'entrée $x = [x_1, \dots, x_n]^T$ le neurone artificiel renvoie la valeur d'activation y :

$$y = f(b + \sum_i w_i x_i) \qquad (3.12)$$

Dans cette formulation, les w_i sont communément appelés les poids et b est appelé le biais. La fonction f est nommée fonction d'activation ou peu fréquemment fonction de transfert. La Figure 3.4 illustre le fonctionnement de celui-ci.

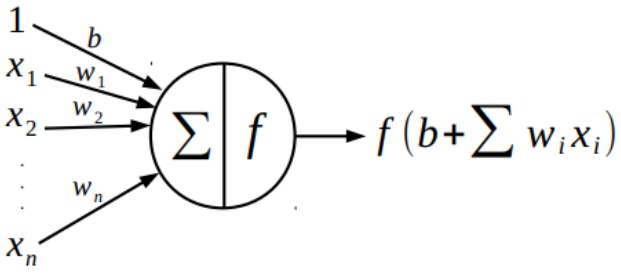


Figure 3.4: Fonctionnement d'un réseau artificiel.

Avec le potentiel qu'offre l'apprentissage humain, il n'est pas étonnant de constater que, dès les débuts de l'informatique, les chercheurs se sont demandés si les machines étaient capables d'apprendre. Pour cette raison, en 1958, [Rosenblatt1958] prend l'initiative d'inventer un algorithme d'apprentissage automatique nommé perceptron. Le perceptron est un modèle simple et adapté pour des problèmes linéairement séparables. Il apprend une fonction linéaire discriminante à partir d'une séquence d'exemples décrits par leurs attributs. Il a la possibilité d'induire un concept à partir d'un échantillon d'exemples, sans savoir à l'avance la distribution de probabilité sur tous les exemples. Pour résoudre des problématiques plus réels et complexes comme par exemple le problème XOR, le perceptron simple ne suffit plus [Minsky1969]. Les chercheurs se sont vite rendu compte qu'en combinant plusieurs neurones le pouvoir de calcul était augmenté. De ce fait, un réseau de neurones plus complexe a été introduit en 1985 : le perceptron multicouche (PMC).

- **Le perceptron multicouche**

Un perceptron multicouche (PMC) est constitué de trois types de couches :

- Une couche d'entrée qui correspond aux données d'entrée $x = [x_1, \dots, x_n]^T$. Cette couche ne contient pas de neurones.
- Une couche de sortie constituée de K neurones, correspondants aux K classes, et produisant les sorties du réseau $y = [y_1, \dots, y_k]^T$, c'est-à-dire les valeurs de sortie associées aux données d'entrée x.
- Des couches cachées constituées chacune de plusieurs neurones. Ces couches permettent la transformation non-linéaire du signal d'entrée vers le signal de sortie. Leur nom vient du fait que les valeurs de sortie de leurs neurones ne sont pas accessibles de l'extérieur.

Dans le réseau du PMC, tous les neurones d'une couche sont connectés aux neurones de la couche précédente. Lorsque le vecteur des caractéristiques d'un objet est présenté à l'entrée du réseau, il est communiqué à tous les neurones de la première couche. Les sorties des neurones de cette couche sont alors communiquées aux neurones de la couche suivante, et ainsi de suite. La figure 3.5 illustre un perceptron multicouche constitué de trois couches cachées. Chaque cercle représente un neurone formel. Les neurones dans le même rectangle font partie de la même couche cachée.

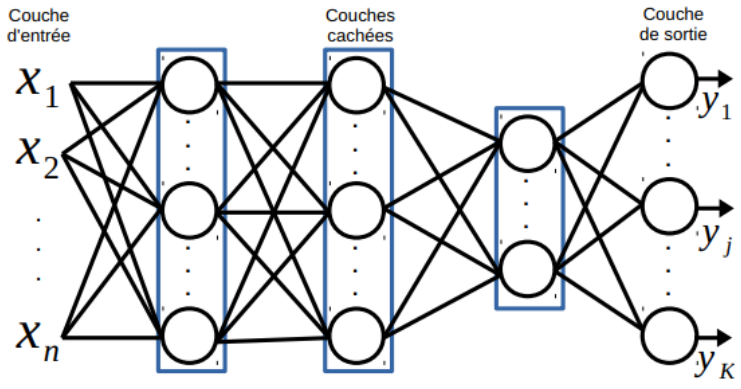


Figure 3.5: Exemple de perceptron multicouche constitué de trois couches cachées.

- **Apprentissage du perceptron multicouche**

Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie désirée). Dans un premier temps, les paramètres du réseau (c'est-à-dire les poids et les biais de chaque

neurone formel qui le constitue) sont inconnus et initialisés avec des valeurs aléatoires. Par conséquent, l'apprentissage consiste à adapter les poids des neurones de manière à ce que le réseau soit capable de réaliser une fonction donnée. Cela s'effectue grâce à l'algorithme de rétropropagation du gradient introduit la même année par [Rumelhart1986].

Pour l'apprentissage, il faut considérer un sous-ensemble de la base d'apprentissage composé de N exemples $\{(x^t, r^t)\}_{t=1}^{N}$avec $x^t = [x_1^t, \dots, x_n^t]^T$ une donnée d'entrée et $r^t = [r_1^t, \dots, r_k^t]^T$ la sortie désirée lui correspondant. Ainsi, (x^t, r^t) est l'exemple t. Nous formalisons ici l'apprentissage du PMC pour un problème multi-classes (K classes). En d'autres termes, $r_j^t = 1$ si x^t appartient à la classe j, et $r_j^t = 0$ sinon.

Nous considérons ici, un PMC à K neurones de sortie. Les paramètres d'un neurone formel j seront notés respectivement $w_{j,i}$ et b_j pour ses poids et son biais. Le poids $w_{j,i}$ correspond au poids de connexion du neurone j au neurone i de la couche précédente.
Il existe plusieurs possibilités de fonction d'activation. Les trois les plus utilisées sont les fonctions « seuil », « linéaire » et « sigmoïde ». Lors de nos travaux, La fonction d'activation f utilisée pour chaque neurone formel du PMC est la fonction différentiable sigmoïde soit :

$$f(x) = \frac{1}{1+e^{-x}} \qquad (3.13)$$

Nous noterons la réponse du neurone formel j, $y_j^t = f\left(\sum_{i=1}^{R} w_{j,i} y_j^t + b_j\right)$ avec R le nombre de neurones de la couche précédente. L'erreur quadratique observée pour la donnée x^t sur les K neurones de la couche de sortie s'écrit :

$$E^t = \frac{1}{2}\sum_{j=1}^{k}\left(e_j^t\right)^2 \qquad \text{Avec } e_j^t = r_j^t - y_j^t \qquad (3.14)$$

- **Adaptation des poids**

La mise à jour des poids du réseau se fait par la méthode de la descente de gradient stochastique (Stochastic gradient descent SGD). Cette méthode est une procédure itérative durant laquelle la méthode d'apprentissage ajuste ses paramètres. A chaque mise à jour, des exemples de la base d'apprentissage sont présentés au réseau qui se charge de faire une prédiction et une estimation de l'erreur associée. Par la suite, le gradient moyen des erreurs est calculé pour indiquer la direction à choisir pour diminuer le risque associé. En d'autres termes, le poids $w_{j,i}$ du neurone j est mis à jour en lui ajoutant le terme $-\alpha\Delta w_{j,i}$. La mise à jour du biais b_j du neurone j est effectuée en lui ajoutant le terme $-\alpha\Delta b_j$. α est appelé taux d'apprentissage. Il permet de pondérer la mise à jour des paramètres du réseau. Les deux termes $\Delta w_{j,i}$ et Δb_j sont définis comme suit :

$$\Delta w_{j,i} = \frac{\partial E}{\partial w_{j,i}} = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial E^T}{\partial w_{j,i}} \qquad \Delta b_j = \frac{\partial E}{\partial b_j} = \frac{1}{N}\sum_{t=1}^{N}\frac{\partial E^T}{\partial b_j} \tag{3.15}$$

Pour une itération τ, permettant de passer dans le réseau un ensemble de données, la mise à jour se fait comme suit :

$$w_{j,i}(\tau) = w_{j,i}(\tau - 1) - \alpha\, \Delta w_{j,i}(\tau) + \beta \Delta w_{j,i}(\tau - 1) \tag{3.16}$$

Avec β, une valeur (entre 0 et 1) permettant de donner une inertie à la descente de gradient en prenant en compte les corrections appliquées à l'itération précédente τ – 1.

Nous détaillons par la suite la rétropropagation du gradient selon le type de couche considéré (couche de sortie ou couche cachée).

- **Rétropropagation pour la couche de sortie**

En reprenant l'équation 3.15 et grâce à la décomposition en chaîne des dérivées partielles, nous pouvons écrire :

$$\frac{\partial E^T}{\partial w_{j,i}} = \frac{\partial E^t}{\partial e_j^t}\frac{\partial e_j^t}{\partial y_j^t}\frac{\partial y_j^t}{\partial a_j^t}\frac{\partial a_j^t}{\partial w_{j,i}} \qquad \frac{\partial E^T}{\partial b_j} = \frac{\partial E^t}{\partial e_j^t}\frac{\partial e_j^t}{\partial y_j^t}\frac{\partial y_j^t}{\partial a_j^t}\frac{\partial a_j^t}{\partial b_j} \tag{3.17}$$

Ces dérivées partielles peuvent individuellement s'exprimer comme suit :

$$\frac{\partial E^T}{\partial e_j^t} = \frac{\partial}{\partial e_j^t}\frac{1}{2}\sum_{j=1}^{k}\left(e_j^t\right)^2 = e_j^t \tag{3.18}$$

$$\frac{\partial e_j^t}{\partial y_j^t} = \frac{\partial}{\partial y_j^t}\left(r_j^t - y_j^t\right) = -1 \tag{3.19}$$

$$\frac{\partial y_j^t}{\partial a_j^t} = \frac{\partial}{\partial a_j^t}\frac{1}{1+a^{-a_j^t}} = \frac{a^{-a_j^t}}{\left(1+a^{-a_j^t}\right)^2} = y_j^t\left(1 - y_j^t\right) \tag{3.20}$$

$$\frac{\partial a_j^t}{\partial w_{j,i}} = \frac{\partial}{\partial w_{j,i}}\sum_{i=1}^{R} w_{j,i}y_j^t + b_j = y_j^t \tag{3.21}$$

$$\frac{\partial a_j^t}{\partial b_j} = \frac{\partial}{\partial b_j}\sum_{i=1}^{R} w_{j,i}y_j^t + b_j = 1 \tag{3.22}$$

Ce qui permet d'obtenir :

$$-\alpha\, \Delta w_{j,i} = \frac{\alpha}{N}\sum_{t=1}^{N}\delta_j^t y_i^t \qquad -\alpha\, \Delta b_j = \frac{\alpha}{N}\sum_{t=1}^{N}\delta_j^t \tag{3.23}$$

En posant:

$$\delta_j^t = e_j^t y_j^t\left(1 - y_j^t\right) \tag{3.24}$$

- **Rétropropagation pour les couches cachées**

Dans le cas de la couche cachée précédent la couche de sortie, l'erreur e_j^t du neurone caché j est inconnue. Pour ce type de couche, la dérivée partielle de l'erreur quadratique de l'équation 3.15 s'écrit comme suit :

$$\frac{\partial E^T}{\partial w_{j,i}} = \frac{\partial E^t}{\partial y_j^t}\frac{\partial y_j^t}{\partial a_j^t}\frac{\partial a_j^t}{\partial w_{j,i}} \qquad \frac{\partial E^T}{\partial b_j} = \frac{\partial E^t}{\partial y_j^t}\frac{\partial y_j^t}{\partial a_j^t}\frac{\partial a_j^t}{\partial b_j} \tag{3.25}$$

La dérivée partielle $\frac{\partial E^t}{\partial y_j^t}$ peut s'exprimer ainsi :

$$\frac{\partial E^t}{\partial y_j^t} = \frac{\partial}{\partial y_j^t}\frac{1}{2}\sum_k (e_k^t)^2 = \sum_k e_k^t \frac{\partial e_k^t}{\partial y_j^t} = \sum_k e_k^t \frac{\partial e_k^t}{\partial a_k^t}\frac{\partial a_k^t}{\partial y_j^t}$$

$$= \sum_k e_k^t \frac{\partial (r_k^t - y_k^t)}{\partial a_k^t}\frac{\partial \left(\sum_l w_{k,l} y_l^t + b_k\right)}{\partial y_j^t} \tag{3.26}$$

$$= \sum_k e_k^t \left(-y_k^t(1-y_k^t)\right) w_{k,j}$$

Nous obtenons donc :

$$\text{-}\alpha\, \Delta w_{j,i} = \frac{\alpha}{N}\sum_{t=1}^N \delta_j^t y_i^t \qquad \text{-}\alpha\, \Delta b_j = \frac{\alpha}{N}\sum_{t=1}^N \delta_j^t \tag{3.27}$$

Avec :

$$\delta_j^t = y_j^t\left(1-y_j^t\right)\sum_k \delta_k\, w_{k,j} \tag{3.28}$$

- **Algorithme de la rétropropagation**

Etape 1: Initialiser les poids et les seuils internes des neurones à des petites valeurs aléatoires.

Etape 2: Calculer le vecteur d'entrée et de sortie désirée, correspondant.

Etape 3: Calculer la sortie du réseau en utilisant l'expression $y_j^t = f\left(\sum_{i=1}^R w_{j,i} y_j^t + b_j\right)$ (3.29)

Etape 4: Calculer l'erreur de sortie en utilisant l'expression (3.24).

Etape 5: Calculer l'erreur dans les couches en utilisant l'expression (3.28).

Etape 6: Calculer le gradient de l'erreur par rapport aux poids en utilisant l'expression (3.15).

Etape 7: Ajuster les poids selon l'expression (3.16).

Etape 8: Si la condition sur l'erreur ou sur le nombre d'itérations est atteinte, aller à **l'étape 9**, sinon aller à **l'étape 2**.

Etape 9: Fin.

Les poids sont ajustés au fur et à mesure, jusqu'à ce que l'erreur de sortie se stabilise à une valeur acceptable.

L'avantage majeur des réseaux de neurones artificiels est le fait qu'ils sont robustes aux données bruitées et Permettent de modéliser de grandes variétés de comportements. Par conséquent, Le RNA représente une boîte noire, et il est difficile voire impossible d'analyser et comprendre son fonctionnement en face d'un problème donné, ce qui laisse le choix de la structure (type, nombre de nœuds, organisation, connexions, etc) empirique.

Après avoir cité les notions de base de chaque étape de notre processus. La section suivante introduit la première contribution de notre thèse portant sur la reconnaissance et la classification des objets 3D basé sur deux types de requêtes.

3.4 Contribution à la classification des objets 3D en se basant sur $P et l'indexation 2D/3D

Dans ce travail, nous proposons une nouvelle méthode de classification des objets tridimensionnels en se basant sur le classifieur $P et l'indexation 2D/3D. Comme nous l'avons déjà mentionné, notre travail consiste à caractériser les objets 3D par un ensemble de vues 2D. Pour chaque objet 3D, un certain nombre de vues sont créées et indexées par un descripteur de forme 2D. Le but est de pouvoir retrouver, à partir d'une ou plusieurs vues 2D, l'objet 3D correspondant. Notre travail vise deux possibilités : une comparaison avec une image requête; ou avec l'objet 3D en comparant leurs ensembles de vues.

La procédure sur laquelle nous nous sommes basés, pour réaliser ce système, peut être résumée dans le diagramme présenté à la figure 3.6. L'objet 3D est un maillage polygonal, sous l'un des formats de fichier 3D (VRML, OFF, OBJ). Dans un premier temps, une estimation de la pose est effectuée. Elle comprend la translation, la mise à l'échelle et la rotation de l'objet. Après l'étape de pré-traitement, un ensemble de 6 vues bidimensionnelles est extrait à partir des facettes d'un cube englobant. Par la suite des images binaires (noir / blanc) sont générées. Pour chaque image 2D, nous appliquons le descripteur de forme nommé transformée de Fourier rapide (sigle anglais : FFT ou Fast Fourier Transform). Cette description nous a permis de définir pour chaque vue un vecteur caractéristique basé sur l'extraction carrée des spectres de Fourier. Ces vecteurs caractéristiques alimentent par la suite le classifieur $P.

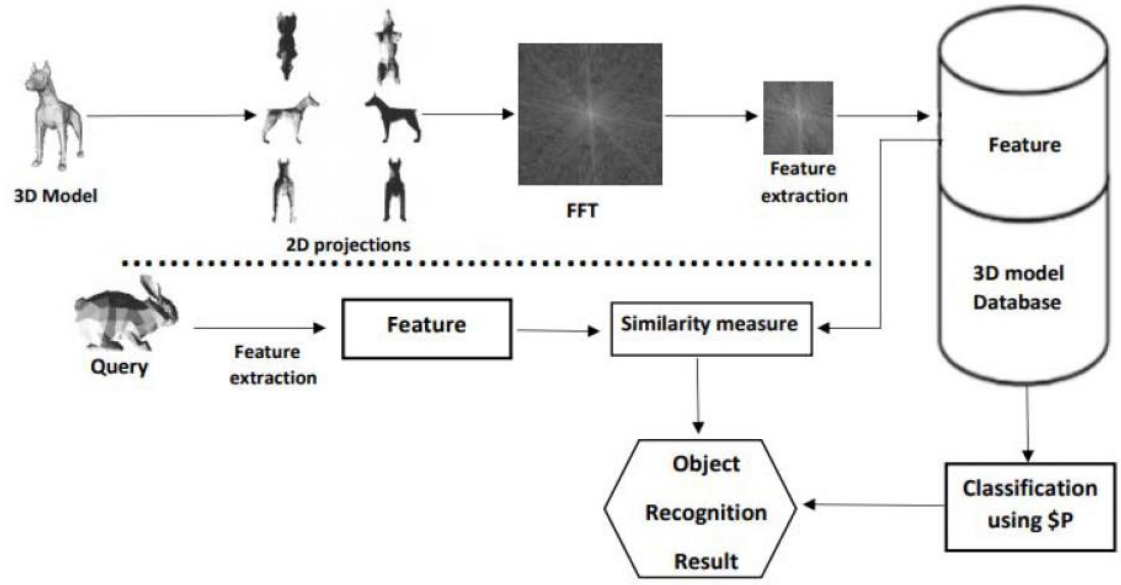


Figure 3.6: Architecture globale du système de reconnaissance et de classification des images 3D.

Comme nous l'avons déjà indiqué au deuxième chapitre de cette thèse, le comportement des méthodes d'indexation 2D/3D est fortement influencé par les différents choix impliqués dans la chaîne de son traitement. Dans ce cadre, l'adoption des stratégies de sélection des angles de vue appropriés et de descripteur 2D discriminant est déterminante. Voyons d'abord la stratégie de sélection des angles de vue retenue dans notre travail.

3.4.1 Extraction des vues

Dans notre approche, nous avons choisi de travailler avec le système de projections orthogonales. Dans ce type de représentation, l'observateur se place perpendiculairement à l'une des faces de l'objet, appelée vue de face. À partir de cette vue, vue principale considérée, il est possible de définir cinq autres vues ou projections orthogonales (analogie avec les six faces d'un dé ou d'un cube). Les projections obtenues s'appellent les vues de droite, de gauche, de haut, de bas et d'arrière (voir figure 3.7).

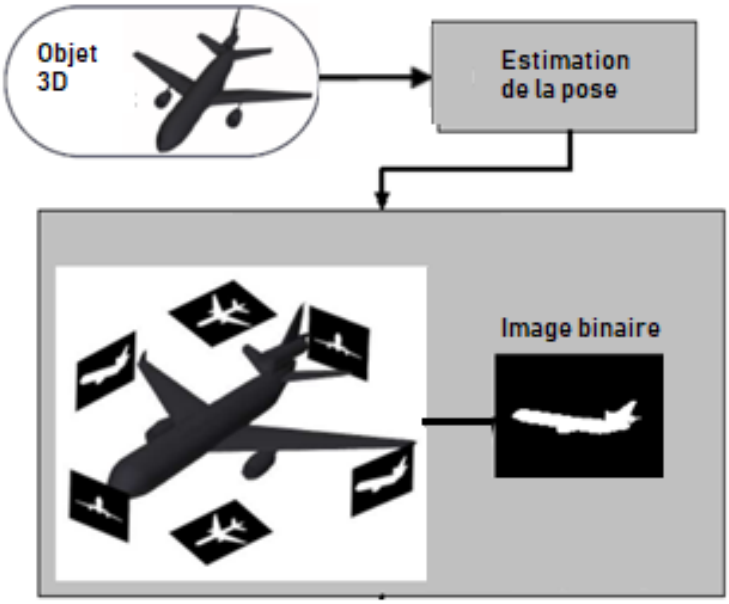


Figure 3.7: Projection orthogonale 2D.

3.4.2 Transformée de fourrier rapide

La Transformée de Fourier Rapide (notée par la suite FFT) est simplement une (transformée de Fourier discrète (notée par la suite TFD) calculée selon un algorithme permettant de réduire le nombre d'opérations et, en particulier, le nombre de multiplications à effectuer [Smach2008] [Journaux2010]. Cependant, il faut noter, que la réduction du nombre d'opérations arithmétiques à effectuer, n'est pas synonyme de réduction du temps d'exécution. Tout dépend de l'architecture du processeur qui exécute le traitement.

Pour calculer une TFD, on doit calculer N valeurs $X(k)$:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}} \qquad (3.30)$$

Et ceci pour $k \in [0, N-1]$.

Si on effectue le calcul directement sans algorithme efficace, on doit effectuer:

$$\begin{cases} N^2 \text{ multiplications complexes} \\ N(N-1) \text{ additions complexes} \end{cases}$$

Il existe différents algorithmes de FFT. Le plus connu est celui de Cooley-Tukey.

3.4.3 Classifieur $P

L'algorithme de reconnaissance de nuage de points $P ($P Point-Cloud Recognizer (PCR)) est le dernier de la famille des dollars ($1 pour uni-stroke, $N pour les multi-stroke). $P est une approche de reconnaissance de mouvements 2D conçue pour le prototypage rapide d'interfaces utilisateur basées sur les gestes [Vatavu et al., 2012]. L'avantage majeur de cet algorithme est le fait qu'il est simple à implémenter, il peut être encodé en une centaine de lignes tout en donnant à la fois un taux de reconnaissance élevé et un faible cout de stockage.

Le processus de reconnaissance du $P comporte deux étapes principales (figure 3.8): La première étape consiste à rendre les nuages de points invariant aux transformations géométriques, A savoir: Ré-échantillonnage; afin d'avoir un nombre de points constant indépendamment de la fréquence d'échantillonnage du périphérique, translation permettant de le rendre indépendant à sa position initiale et la mise à l'échelle permettant de rendre l'objet indépendant à la taille. La seconde étape repose sur la comparaison en se basant sur la distance euclidienne.

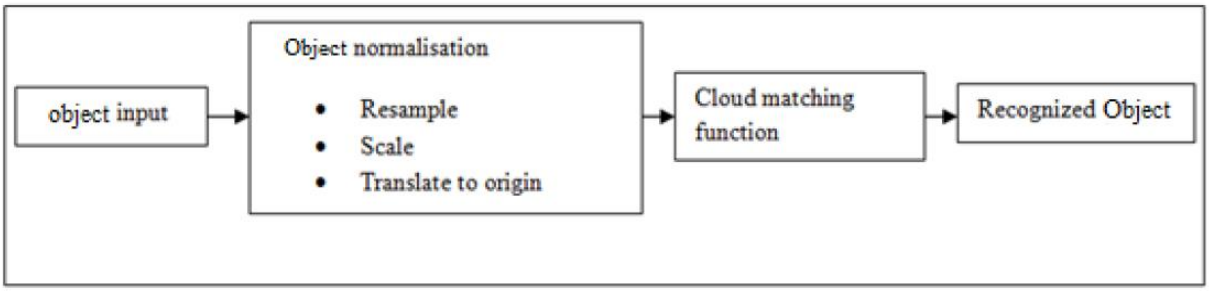


Figure 3.8: Algorithme de reconnaissance de nuage de points $P.

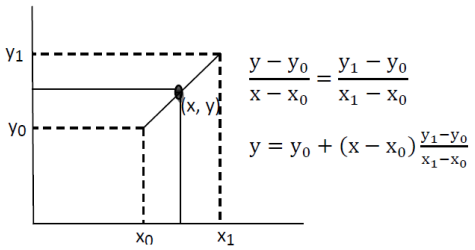
Un objet est défini par un ensemble de points. Ces points sont ensuite comparés à un autre ensemble de points précédemment stockés (modèles) en utilisant une mesure de la distance euclidienne. Contrairement aux autres algorithmes d'apprentissage automatique, citons comme exemple les réseaux de neurones artificiels. La phase d'apprentissage pour l'algorithme $P est effectuée une seule fois, c'est-à-dire qu'il ne nécessite qu'un seul passage pour créer un modèle.

- ***Ré-échantillonnage***

Pour comparer l'objet donné en requête avec des modèles préexistants, nous devons d'abord ré-échantillonner l'ensemble de ces points créés. Les M points d'un objet sont ré-échantillonnés en N points espacés à un intervalle régulier. Premièrement, nous avons calculé la longueur totale des points générés par un objet. Deuxièmement, nous avons divisé ce total par N-1 où N ne

devrait pas être trop petit, ce qui causerait une perte significative de précision ou serait trop important et causerait trop de comparaisons en consommant du temps (équation (3.33)). Nous regardons les nuages de point original, puis nous déterminons les points en utilisant l'interpolation linéaire (équation (3.34)).

$$l = length \div (N - 1) \tag{3.31}$$



$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \qquad y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} \tag{3.32}$$

- ***Mise à l'échelle et translation***

L'objet est mis à l'échelle en fonction du rectangle de référence (carré de référence ou cadre de sélection). Ceci est calculé en prenant le minimum et le maximum de points x, y et z (minx, maxx, miny, maxy, minz, maxz) (equation 3.35). Après la mise à l'échelle, l'objet est translaté en un point de référence. Pour simplifier le calcul, nous choisissons (0, 0, 0) comme centroïde pour translater l'objet.

$$Scale = Max(x_{max} - x_{min}, y_{max} - y_{min}, z_{max} - z_{min}) \tag{3.33}$$

- ***Comparaison à l'aide de la distance euclidienne***

La liste des points obtenus a ensuite été comparée à chacun des modèles existants. Le modèle correspond à un objet de référence auquel nous associons une liste de points et un nom. La liste de points obtenue est alors comparée à chacun des modèles existants en appliquant une série d'ajustements angulaires pour trouver l'alignement optimal. Chaque comparaison, basée sur une distance euclidienne, est ensuite utilisée pour calculer un score reflétant le degré de similitude entre le modèle et l'objet, [formule (3.36)]. Le modèle qui obtient le score le plus élevé est considéré comme l'objet reconnu.

$$\sum_{i=1}^{n} \|C_i - T_j\| = \sum_{i=1}^{n} \sqrt{(C_{i.x} - T_{j.x})^2 + (C_{i.y} - T_{j.y})^2 + (C_{i.z} - T_{j.z})^2} \tag{3.34}$$

3.5 Expérimentations

3.5.1 Bases de données

Afin d'expérimenter et de tester l'efficacité de notre approche, nous avons utilisé une base d'objets 3D composée de 76 modèles sous format VRML 2.0. Nos objets ont été extraits de la base de données standard de référence de Princeton (Princeton Shape Benchmark, 2004) avec l'extension «.off». Les modèles similaires géométriquement sont regroupés manuellement dans la même catégorie. Nous avons obtenu 12 classes (Voir figure 3.9): Avions, poisson, requin, bouteille, humain, squelette, chien, lapin, chevale, cochon, chaise et guitare.

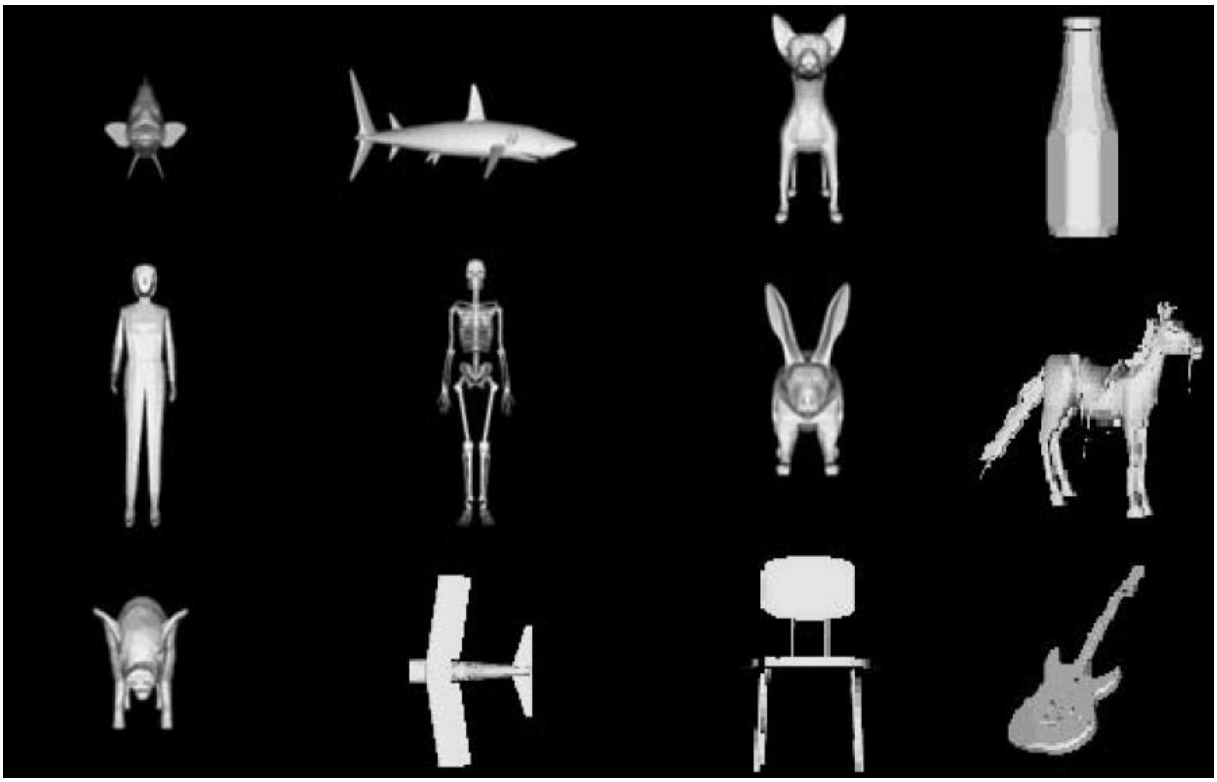


Figure 3.9: Exemples de modèles 3D.

3.5.2 Résultats

- *Extraction des vues 2D*

Jusqu'à présent, notre application se contentait d'afficher la scène 3D dans une unique fenêtre, que nous pouvons l'appeler vue de face. Or, lors de l'utilisation des logiciels de modélisation 3D par exemple 3D Studio MAX, nous avons remarqué qu'en plus de la vue de face, d'autres vues comme la vue de dessus ou de côté permet de mieux visualiser les objets 3D. Java 3D offre la possibilité de visualiser une scène 3D sous plusieurs angles, c'est ce que nous allons

voir dans cette section. La vue de l'objet par défaut est celle de face. Désormais, dans le cas qui nous préoccupe, nous aimerions avoir une classe qui nous permet la création des vues multiples. Dans notre cas, nous allons créer une vue de face, de derrière, de gauche, de droite, de dessous et de dessus (Voir figure 3.10).

- Création et placement des 6 vues :
  - Vue de face perpendiculaire à Z placée à Z = 3,
  - Vue d'arrière perpendiculaire à Z placée à Z = -3,
  - Vue de droite perpendiculaire à X placée à X = 3,
  - Vue de gauche perpendiculaire à X placée à X = 3,
  - Vue de haut perpendiculaire à Y placée à Y = 3,
  - Vue de bas perpendiculaire à Y placée à Y = 3.

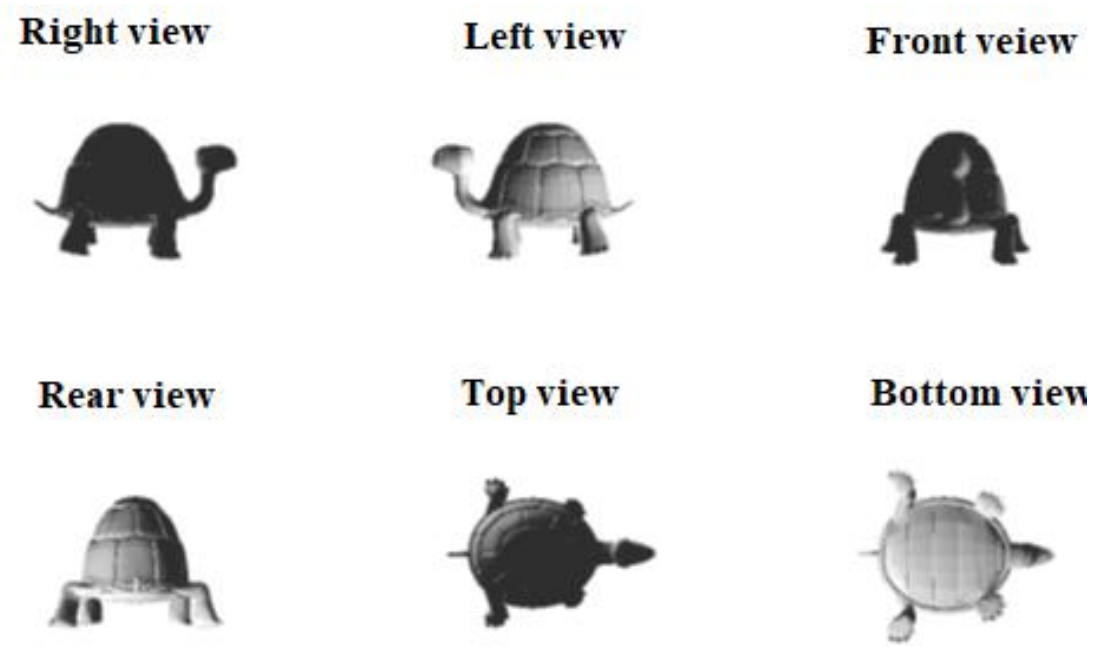


Figure 3. 10: Les vues 2D extraites de l'objet 3D sélectionné.

Les images multi-vues sont restituées à partir des points de vue de la caméra. Notre programme nous offre par la suite des images 2D binaire. Ces dernières ne sont que des silhouettes. Les valeurs des pixels égaux à 1 si le pixel se trouve dans la vue 2D du modèle et 0 sinon.

- *Le rendu spectral*

Afin d'avoir une représentation synthétique de l'image et de garantir une création de vues rapide, nous avons opté pour la mise en place d'un processus de rendu spectral basé sur le Fourier rapide des vues extraites de l'objet 3D (Voir figure 3.11).

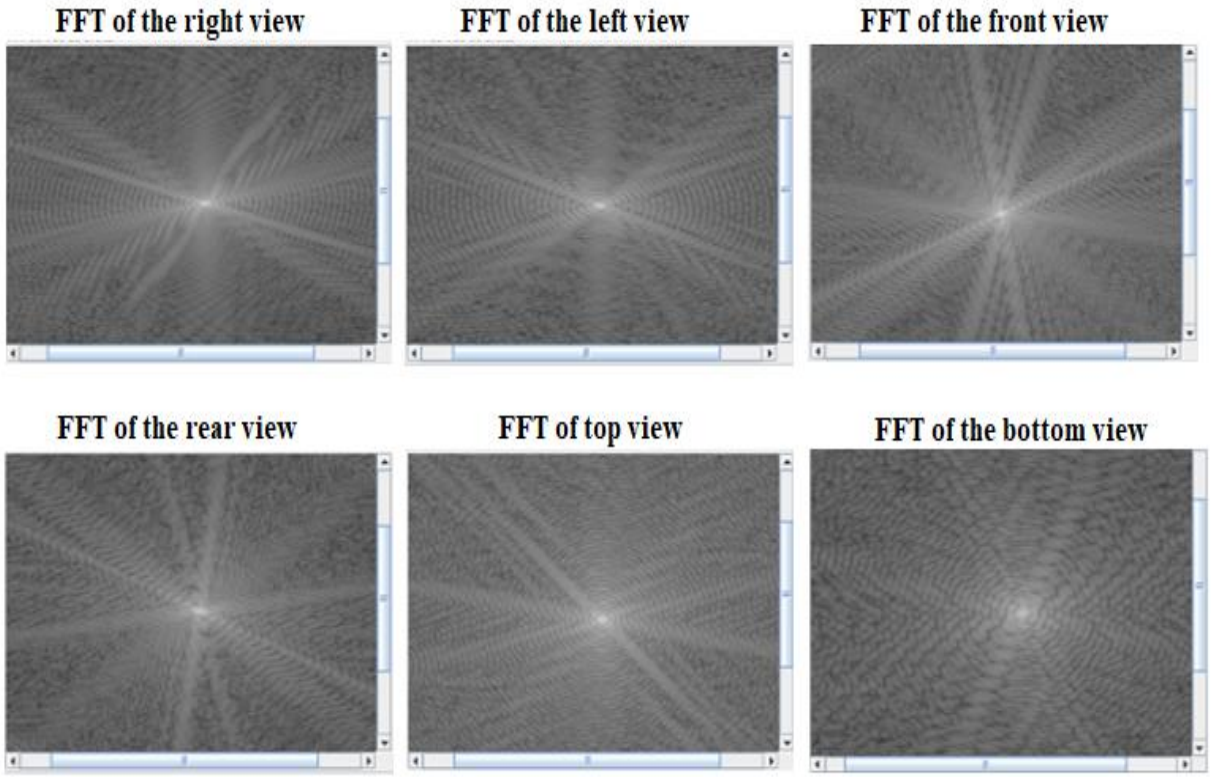


Figure 3. 112: La FFT appliquée sur les 6 vues extraites de l'objet 3D.

- *Extraction du vecteur caractéristique*

Une fois le descripteur de Fourier est appliqué sur les images extraites de l'objet 3D nous procédons par la suite à une extraction des composantes du vecteur caractéristique en prenant en compte des parties contenant des propriétés intrinsèques.

Nous avons choisi 16*16 comme taille du vecteur caractéristique qui décrit l'information condensée de l'image analysée (Voir figure 3.12). C'est ce vecteur qui sera utilisé dans l'étape suivante: La classification

L'utilisation des vecteurs caractéristiques de ces vues permet une mesure précise pour l'entrée des classifieurs ($P, RNA, SVM, KNN). Notre algorithme prend en entrée une ou plusieurs images ou même directement un objet 3D. Il compare ces données à la base de données d'objets 3D: même lors de la comparaison de deux modèles 3D, la comparaison est effectuée via des projections 2D de ces objets. La requête doit elle-même être indexée, elle doit subir toutes les transformations effectuées précédemment sur la base de données.

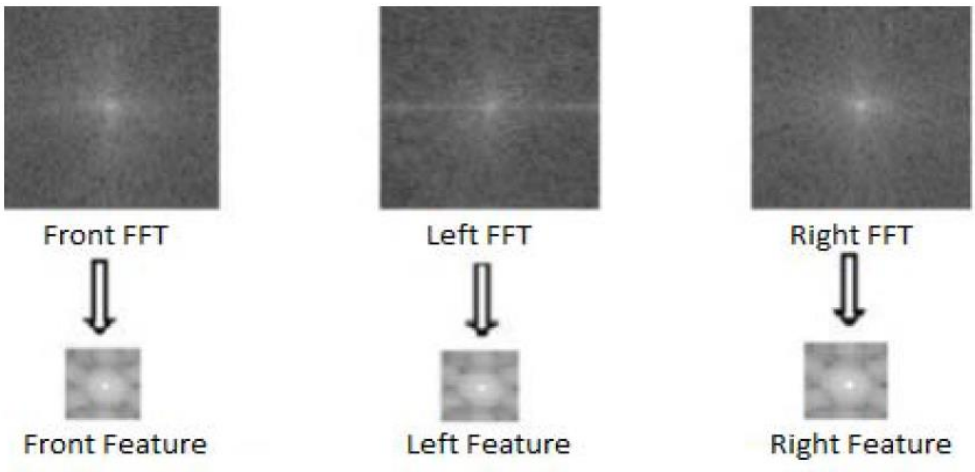


Figure 3. 32: Les images FFT et leurs vecteurs de caractéristiques associés.

- *Classification en utilisant l'algorithme $P*

Pour reconnaître un nouvel objet, l'algorithme $P prend en entrée une liste de points correspondants à cet objet. Le vecteur d'objet passe par des phases de pré-traitement (voir Section 3.4.3). Ensuite, il effectue la correspondance des nuages de points de l'objet candidat (nouvel objet) avec les nuages de points de chaque modèle de l'ensemble d'apprentissage en se basant sur une distance euclidienne. Cette distance est utilisée pour décrire la plus grande similarité entre les objets en calculant la distance minimale entre eux. Les valeurs des tableaux 3.1 et 3.2 correspondent aux résultats de classification de nos objets 3D en utilisant l'algorithme $P.

Tableau 3.1 classification avec une requête image.

Classes	Bouteille	Cochon	Cheval	Chien	Requin	Poisson	Lapin	Humain	Squelette	Guitare	Avion	Chaise
Bouteille	87%	-	-	-	-	-	-	-	-	-	-	-
Cochon	-	92%	-	-	-	-	-	-	-	-	-	-
Cheval	-	-	89%	-	-	-	-	-	-	-	-	-
Chien	-	-	-	92%	-	-	-	-	-	-	-	-
Requin	-	-	-	-	86%	-	-	-	-	-	-	-
Poisson	-	-	-	-	-	89%	-	-	-	-	-	-
Lapin	-	-	-	-	-	-	94%	-	-	-	-	-
Humain	-	-	-	-	-	-	-	96%	-	-	-	-
Squelette	-	-	-	-	-	-	-	-	94%	-	-	-
Guitare	-	-	-	-	-	-	-	-	-	96%	-	-
Avion	-	-	-	-	-	-	-	-	-	-	89%	-
Chaise	-	-	-	-	-	-	-	-	-	-	-	94%

Nous comparons dans le tableau 3.3 les résultats de l'algorithme $P à d'autres méthodes; telles que les réseaux de neurones multicouches, les méthodes Machine à Vecteurs Support et K-Plus Proches Voisins. Nous avons divisé notre base de données 3D en deux sous bases; la base d'apprentissage est constituée de 70% et la base de test est constituée de 30%.

Plusieurs architectures RNA ont été testées. Nous avons constaté qu'un réseau de 46 neurones dans la couche cachée donne des meilleurs résultats. Pour déterminer le noyau correct de SVM. Nous avons examiné les performances du classificateur SVM avec différents noyaux linéaires et non linéaires (polynôme et RBF). Et par la suite nous avons conclu que les SVM linéaires étaient les plus performants en termes de taux de reconnaissance. De même, l'algorithme KNN a été testé avec plusieurs nombres de voisins dans l'intervalle [1–5].

Tableau 3.2 Classification avec une requête objet en comparant leurs ensembles de vues.

Classes	Bouteille	Cochon	Cheval	Chien	Requin	Poisson	Lapin	Humain	Squelette	Guitare	Avion	Chaise
Bouteille	94%	-	-	-	-	-	-	-	-	-	-	-
Cochon	-	94%	-	-	-	-	-	-	-	-	-	-
Cheval	-	-	91%	-	-	-	-	-	-	-	-	-
Chien	-	-	-	96%	-	-	-	-	-	-	-	-
Requin	-	-	-	-	94%	-	-	-	-	-	-	-
Poisson	-	-	-	-	-	94%	-	-	-	-	-	-
Lapin	-	-	-	-	-	-	91%	-	-	-	-	-
Humain	-	-	-	-	-	-	-	96%	-	-	-	-
Squelette	-	-	-	-	-	-	-	-	96%	-	-	-
Guitare	-	-	-	-	-	-	-	-	-	91%	-	-
Avion	-	-	-	-	-	-	-	-	-	-	91%	-
Chaise	-	-	-	-	-	-	-	-	-	-	-	94%

Tableau 3.3 Les taux de reconnaissance pour $P, RNA, SVM et KNN.

Classifieurs / Type de requête	$P	RNA	SVM	KNN
Requête image	**91.5%**	**86.1%**	**79.3%**	**80.6%**
Requête objet	**93.5%**	**88.9%**	**81.9%**	**85.2%**

Comparé à d'autres algorithmes, $P offre un niveau de satisfaction élevé par rapport aux autres classifieurs (RNA, SVM, KNN). Nos résultats ont prouvé que l'algorithme de reconnaissance $P est plus efficace, rapide, simple et précis [elhoufi2018]. La plupart des objets sont parfaitement classés avec un très bon taux de reconnaissance.

3.6 Conclusion

Dans ce chapitre nous avons proposé une méthode pour la classification, en vue de l'indexation, des objets 3D. La méthode est basée sur l'algorithme $P. Notre choix de traitement de ces données était motivé par la quantité phénoménale d'objets disponibles aujourd'hui, qui ne cesse de croître.

Le but de ce travail a été d'extraire, dans les objets 3D, une certaine information pertinente de la forme. Ces informations permettent une classification efficace des objets 3D. Pour atteindre cet objectif, nous avons commencé par l'extraction des six vues pour chaque objet soulevé de la base. Ensuite, nous avons extrait des descripteurs caractéristiques afin d'étiqueter nos objets 3D. Enfin, nous nous sommes intéressés à la reconnaissance de ces objets en se basant sur l'algorithme $P. Nous avons aussi développé une interface graphique qui permet à l'utilisateur de proposer facilement une requête sous forme de vecteur caractéristique et d'afficher les résultats de reconnaissance. Notre système a été testé sur un ensemble d'objets de différentes formes et comparé à d'autres méthodes de classification. Les résultats de simulation sont encourageants et démontrent l'intérêt et la performance du système proposé. Les taux de reconnaissance obtenus sont supérieurs à 91.5%.
Le principal inconvénient des stratégies de vues représentatives réside dans les coûts de calcul, qui comprennent la réalisation et la description d'un grand nombre de vues, ainsi que le nombre élevé de comparaisons par paires entre les paires de vues nécessaires. Pour cela dans le chapitre suivant, nous allons se concentrer sur l'indexation d'objet 3D/3D. Le traitement sera alors effectué directement sur la forme du modèle 3D sans avoir besoin des projections 2D.

Chapitre 4 : La Classification des objets 3D en se basant sur l'approche 3D/3D

4.1 Introduction

Le processus de reconnaissance comporte trois étapes principales [Bagci2012]: La première étape consiste à calculer des descripteurs, comme nous l'avons déjà vu dans le chapitre 2. Ils peuvent être globales [Ion2008], locales [Assfalg2006, Lavoue2012], structurels [Tung2005], basés sur les transformées [Vranic2001] ou fondés sur les vues [Beis1999]. La seconde est la mesure de similarité qui peut être basée sur la distance [Veltkamp2001], la probabilité [Super2004] ou sur les graphes [Baeza-Yates2000]. Enfin, le processus de la prise de décision peut ainsi classer les objets 3D en fonction de la mesure de similarité utilisée. Ce chapitre traite le sujet de la reconnaissance des formes libre 3D. Nous nous concentrons sur la problématique liée à la reconnaissance et à la classification d'objets 3D en se basant sur l'indexation 3D/3D.

Dans la première partie de ce chapitre, nous décrivons et analysons une méthode de calcul des signatures de forme 3D pour des objets arbitraires décrits par des modèles polygonaux 3D. L'idée clé est de représenter la signature d'un objet sous forme d'une distribution de forme, en se basant sur une fonction de forme mesurant les propriétés géométriques globales de l'objet. Dans la deuxième partie, nous évaluons différentes mesures de similarité, pour conclure celle qui nous offre une bonne discrimination en fonction du temps de calcul. Finalement, nous présentons des techniques d'apprentissage automatique (« Machine Learning ») en vue de reconnaitre ou classer une requête inconnue par rapport aux données d'apprentissage.

Le premier défi de cette approche est de choisir la fonction de forme discriminante. Le second consiste à construire la distribution de forme la plus adéquate et le dernier consiste à comparer et classer ces distributions. Pour atteindre notre objectif, nous utilisons la distribution de formes (D2) en tant que descripteur de formes et les réseaux de neurones artificiels pour la classification. L'objectif derrière l'utilisation du RNA dans ce travail est de générer une meilleure fonction de sortie que les méthodes d'approximation classiques [Munoz-Rodriıguez2003]. Nous avons constaté que note système est non seulement simple et rapide, mais offre également une classification avec un taux de reconnaissance satisfaisant.

Ce chapitre est divisé en plusieurs sections. Dans la section 2, nous citons quelques notions de base sur l'indexation 3D/3D, ensuite nous mentionnons les méthodes qui ciblent les objets 3D de forme libre. Puis dans la section 3, nous définissons diverses formules de calcul de distances pour estimer la similarité des vecteurs caractéristiques. Dans la section 4, nous détaillons la description du problème et nous proposons des solutions pour mettre en œuvre notre approche.

La section 5 présente les résultats des expériences visant à évaluer l'efficacité et la robustesse de notre système.

4.2 Indexation 3D/3D

L'indexation 3D est considérée comme le pilier du processus de recherche et de reconnaissance d'objets 3D. Le principe consiste à caractériser la forme des objets de manière compacte et pertinente pour en déduire une signature. La similarité entre les objets 3D est alors mesurée en comparant leurs signatures. Dans ce chapitre, nous présentons un descripteur extrait directement de la géométrie de l'objet.

Généralement, les caractéristiques sont classées en deux types [Chaieb2014]. Les caractéristiques locales et les caractéristiques globales. Dans la première catégorie, nous trouvons les caractéristiques extraites d'une certaine partie de l'objet. Tandis que pour la deuxième, les caractéristiques sont calculées sur tout l'objet.

Des chercheurs en psychologie ont montré l'effet des caractéristiques de forme d'un objet (par exemple, les contours, la symétrie, parallélisme, etc.) sur la perception de la vision humaine [Levi2007]. Ces idées ont été adoptées dans le domaine de la vision par ordinateur en analysant les caractéristiques de forme dans le but d'améliorer la précision de la reconnaissance des objets. Concernant les objets de forme libre, nous mentionnons le travail de Asari et al. [Asari2014] qui ont développé une étude comparative entre quatre descripteurs de formes 3D à savoir: distribution de formes [Osada2002], image de spin local [Zaharia2001], image de spin globale [Huang2010] et histogramme de formes [Ankerst1999]. Ils ont conclu que dans l'évaluation inter-classe, l'image de spin locale (en utilisant une mesure de similarité locale) et la distribution de forme montrent d'excellentes performances par rapport aux autres descripteurs de forme 3D. L'image de spin locale a surperformé la distribution de forme, grâce aux propriétés de forme locales qui sont légèrement similaires dans plusieurs images de profondeur de la même instance. Tandis que la distribution de forme offre d'excellentes performances par rapport à l'image de spin locale, lorsque plusieurs instances par classe sont utilisées dans l'évaluation intra-classe. Cela est dû au fait que l'image de spin locale souffre de l'invariance des propriétés locales dans différentes instances par classe, alors que les propriétés de forme globales extraites par la distribution de forme sont légèrement conservées. Puisque nous travaillons avec différents objets, nous avons utilisé le descripteur global nommé distribution de forme. Le but

de notre travail est de développer une méthode rapide, simple et robuste pour la reconnaissance de modèles polygonaux 3D. Dans la section suivante, nous décrirons en détail la fonction de forme choisie et les mesures de similarité ainsi que le classifieur utilisé dans notre travail.

4.3 Les mesures de similarité

L'utilisation de la mesure de similarité reste presque toujours une étape essentielle, dont l'objectif est de pouvoir comparer les ressemblances et les différences entre deux vecteurs. La mesure de similarité utilise la représentation mathématique des caractéristiques de forme (c'est-à-dire le descripteur), afin d'associer une appréciation quantitative à la similarité entre les formes [Domenach2017]. Selon le descripteur de forme considéré, l'une des méthodes de mesure de similarité suivantes peut être utilisée:

- ✓ Méthodes basées sur la distance (métrique), appropriées pour la représentation vectorielle des entités et supposées calculer des métriques telles que la distance euclidienne.
- ✓ Méthodes de mise en correspondance de graphes, spécifiquement adaptées aux représentations basées sur des graphes.

4.3.1 Méthode basée sur les mesures de distance

La distance calcule la dissemblance entre deux vecteurs. Une valeur minimale indique que les deux vecteurs sont très similaires alors que des valeurs plus élevées correspondent à des vecteurs différents. Afin de garantir une bonne estimation de la similarité, une mesure de distance doit satisfaire plusieurs propriétés, citées ci-dessous.

4.3.1.1 Propriétés des mesures de distances

Soit X, Y, Z trois vecteurs dans un espace à n dimensions et $d(X,Y)$ une fonction définie comme

$$\mathrm{d} : R^n * R^n \rightarrow R; \qquad X,Y,Z \in R^n$$

La fonction d est une distance si elle vérifie les propriétés suivantes:

- **Identité** : $d(X,Y) = 0$; La distance entre deux vecteurs identiques doit être égale à zéro.

- **Positivité** : $d(X,Y) \geq 0$; la distance entre deux vecteurs différents doit toujours avoir une valeur positive.
- **Symétrie** : $d(X,Y) = d(Y,X)$; La distance entre X et Y doit être égale à la distance entre Y et X.
- **Inégalité triangulaire** : $d(X,Z) \leq d(X,Y) + d(Y,Z)$; la distance de X à Z est inférieur à la somme des distances de X à Y et de Y à Z.
- **Invariance au transformation** : $d(g(X),Y) = d(X,Y)$; où g est une transformation dans un groupe donné. Cela signifie que la distance entre deux formes est indépendante de leur position, de leur taille ou de leur orientation.

Il existe un grand nombre de mesures de distance qui peuvent être utilisées. Nous citons dans la section suivante quelques-unes.

4.3.1.2 Mesures de distances

Soit $X = (x_1, x_2, \dots x_n)$ et $Y = (y_1, y_2, \dots y_n)$ deux point dans l'espace R^n. Plusieurs métriques sont définies afin de mesurer la distance entre X et Y:

La distance la plus connue est la **distance Euclidienne**, qui définit l'espace cartésien.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (4.1)$$

(Pour des vecteurs de dimension n). La distance Euclidienne n'est pas discrète car elle n'est pas à valeurs entières. Cette distance, n'est qu'un cas particulier pour p = 2 de la **distance de Minkowsky** :

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \qquad (4.2)$$

Pour p = 1, on obtient la **distance de Manhattan** (aussi appelée métrique absolue) :

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \qquad (4.3)$$

et pour $p = \infty$, la **distance de Chebychev** (aussi appelée métrique maximum) :

$$d(x,y) = \max_{i=1}^{n} |x_i - y_i| \quad (4.4)$$

Les techniques géométriques, telles que la distance euclidienne, peuvent être appliquées sur des histogrammes. Il existe cependant des mesures spécifiques aux histogrammes.

L'intersection d'histogrammes:

$$d(x,y) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} (y_i)} \quad (4.5)$$

Il ne s'agit pas d'une distance, puisqu'elle ne respecte pas la propriété de symétrie. Pour y remédier, on peut utiliser l'expression suivante

$$d(x,y) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\min(\sum_{i=1}^{n} x_i, \sum_{i=1}^{n} y_i)} \quad (4.6)$$

Earth Mover's Distance (EMD)

Earth Mover's Distance est basée sur la minimisation du coût nécessaire pour transformer une distribution en une autre distribution. Elle peut être appliquée pour calculer la similarité entre deux distributions ou entre deux ensembles de distributions. C'est l'une des seules distances qui permet de travailler sur des histogrammes qui n'ont pas forcément le même nombre de bins.

$$EMD(x,y) = min \frac{\sum_{ij} f_{ij} d_{ij}}{\sum_{ij} f_{ij}} \quad (4.7)$$

Avec

$$\sum_{ij} f_{ij} \leq x_i \, , \sum_{ij} f_{ij} \leq y_j , \sum_{ij} f_{ij} \equiv min\{ \sum_i x_i , \sum_j y_j \}, f_{ij} \leq 0$$

Kullback-Leibler divergence (KLD)

Issue de la théorie de l'information, **la divergence de Kullback-Leibler** [Kullback 1968] permet de mesurer la dissimilarité entre deux distributions. Elle exprime l'entropie relative de la distribution x par rapport à la distribution y:

$$div(x,y) = \sum_{i=1}^{n} x_i log_2 \frac{x_i}{y_i} \quad (4.8)$$

Cette mesure n'est pas une distance, car elle n'est pas une métrique. Elle n'est pas symétrique et ne satisfait pas l'inégalité du triangle. $div(x,y) \neq div(y,x)$ Si l'on souhaite une distance, on peut utiliser **la distance de Kullback-Leibler (KL)**.

$$d(x,y) = div(x,y) + div(y,x) \quad (4.9)$$

La distance de Kullback-Leibler [Roldan2012] admet deux principaux avantages. D'abord, elle s'exprime uniquement en fonction des paramètres d'échelle et de forme ce qui ne nécessite pas la normalisation des vecteurs de caractéristiques. Elle est simple à écrire et son calcul n'est pas couteux. Ensuite, elle est exactement équivalente au critère de maximum de vraisemblance. Grâce à ces propriétés, la KL est une bonne mesure de similarité.

Cependant, **la distance de Jeffrey** lui est préférée pour son respect de la symétrie et de l'inégalité triangulaire

$$d(x,y) = \sum_{i=1}^{n} \left(x_i log \frac{2x_i}{x_i+y_i} + y_i log \frac{2y_i}{x_i+y_i} \right) \quad (4.10)$$

Pour limiter la sensibilité au bruit de ces distances ou similitudes, on peut remplacer les histogrammes par les histogrammes cumulés.

Mesures statistiques:

Le **test statistique du X^2** (chi-square) permet de décider si deux vecteurs x et y sont engendrés par la même distribution. La version symétrique du test est :

$$d(x,y) = \sum_{i=1}^{n} \frac{(x_i-y_i)^2}{x_i+y_i} \quad (4.11)$$

C'est une des mesures de similarité parmi les plus rapides, elle donne de bons résultats sur les grands ensembles de données.

4.3.2 Méthode basée sur les graphes

Si la forme des objets 2D est représentée par un graphe, une procédure de correspondance de graphe est nécessaire pour comparer les deux formes.

Le but d'une méthode de correspondance de graphe est de déterminer la meilleure correspondance entre les deux représentations de graphe. Le niveau de ressemblance entre les graphes est donné par une fonction qui mesure la similarité entre des couples de sommets et d'arêtes correspondants. L'approche de correspondance de graphe peut également être considérée comme un algorithme de minimisation d'énergie.

Il existe deux classes de méthodes de correspondance. Lorsque les deux graphes ont la même taille (c.-à-d. Le même nombre de sommets), un mappage isomorphe peut être trouvé entre eux.

Cette famille s'appelle l'appariement exact. Contrairement, lorsque les deux graphes ont des tailles différentes, la correspondance un à un devient impossible. Ce cas est appelé appariement inexacte.

La nature combinatoire des approches de correspondance des graphes conduit à une complexité de calcul élevée, qui est le plus souvent NP-complète [Garey1979] [Conte2004].

4.5 Contribution à la classification des objets 3D en se basant sur la distribution de forme 2D

4.5.1 Distribution de forme

L'approche consiste à représenter et reconnaître des objets de forme libre sous la forme d'une distribution de probabilité échantillonnée à partir d'une fonction de forme mesurant les propriétés géométriques du modèle 3D. Nous appelons cette généralisation des histogrammes géométriques une distribution de forme. Par exemple, une telle distribution de forme nommée D2, représente la distribution des distances euclidiennes entre des paires de points choisis au hasard sur la surface d'un modèle 3D. Les échantillons de cette distribution peuvent être calculés rapidement et facilement. Une fois que nous avons calculé les distributions de formes pour deux objets, la similarité entre les objets peut être évaluée à l'aide d'une métrique mesurant la distance entre les distributions (par exemple, la norme LN), éventuellement avec une étape de normalisation pour l'adaptation des échelles.

L'idée principale est de représenter la signature de forme de l'objet sous forme de distribution de probabilité. Cette représentation produit des propriétés géométriques globales informatives pour la reconnaissance et la classification. Son principe est d'obtenir une fonction paramétrée de notre modèle 3D et de la comparer facilement avec d'autres, comme le montre la figure 4.1.

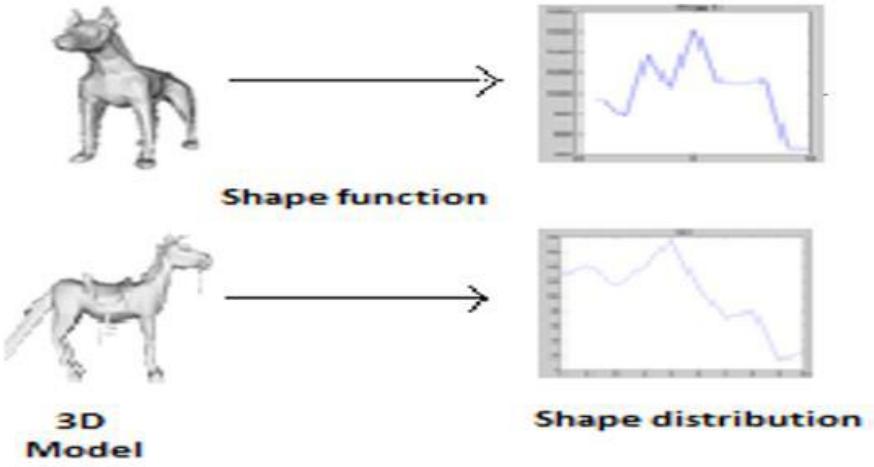


Figure 4. 1: Les distributions de formes facilitent l'appariement des formes.

Malgré sa simplicité, nous nous attendons à ce que l'approche soit capable de discriminer efficacement des objets polygonaux de différentes formes (les résultats des expériences testant cette hypothèse sont présentés à la section 5). En outre, elle possède plusieurs propriétés:

- L'invariance: les distributions de formes ont toutes les propriétés d'invariance de transformation. Par exemple, la fonction de forme D2 produit une invariance sous des mouvements rigides et une image miroir. Dans ce cas, l'invariance d'échelle peut être ajoutée en normalisant les distributions de formes avant de les comparer. Les autres fonctions de forme qui mesurent les angles ou les rapports entre les longueurs sont invariantes à toutes les transformations.
- Robustesse: l'échantillonnage aléatoire garantit que les distributions de formes sont insensibles aux petites perturbations. Intuitivement, chaque point d'un modèle 3D contribue d'une manière égale à la distribution de forme, l'ampleur des modifications de la distribution de forme est liée à l'ampleur des modifications apportées au modèle 3D. Par exemple, si un petit pourcentage d'un modèle 3D est perturbé (En ajoutant du bruit aléatoire : une petite bosse sur une surface), une distribution d'échantillons aléatoires du modèle doit alors: aussi changer par un petit pourcentage. Cette propriété fournit une insensibilité au bruit, aux fissures et à la poussière dans les modèles 3D en entrée. Nous supposons que les distributions pour la plupart des fonctions de forme globales basées sur des distances et / ou des angles varient également de manière continue et monotone pour les changements de forme locaux.

- Métrique: la mesure de dissimilitude produite par l'approche adopte les propriétés de la norme que nous utilisons pour comparer les distributions de formes. En particulier, si la norme est une métrique, notre mesure de similarité l'est aussi. Cette propriété est valable pour la plupart des normes courantes, y compris la norme LN, la distance de déplacement de Earth Mover, etc.
- Efficacité: la construction des distributions de formes pour une base de données de modèles 3D est généralement rapide et efficace. Par exemple, la complexité de prendre S échantillons de la fonction de forme D2 à partir d'un modèle 3D avec N triangles est S*log (N).
- Généralités: les distributions de formes sont indépendantes à la représentation, de la topologie ou du domaine d'application des modèles 3D. En conséquence, la méthode de similarité de forme peut s'appliquer aussi bien aux bases de données contenant des modèles 3D stockés sous forme de polygone, mailles, solide, voxels ou toute autre représentation géométrique, à condition qu'une fonction de forme appropriée puisse être calculée à partir de chaque représentation. De plus une base de données (telle que le World Wide Web) peut contenir des modèles 3D dans une variété de représentations et de formats de fichiers différents. Enfin, les distributions de formes peuvent être utilisées dans de nombreux domaines d'application pour comparer des formes naturelles (par exemple des animaux) et/ou des objets fabriqués par l'homme (par exemple des pièces d'usines).

La stratégie choisie est la suivante: à partir d'un objet polygone à reconnaître, une triangulation est réalisée. Puis, des distances sont calculées entre deux points aléatoires de la surface triangulés de l'objet 3D. La fréquence de ces distances sera par la suite représentée par un histogramme normalisé. Ensuite, nous allons comparer ces distributions en utilisant les mesures de similarité les plus utilisées. Finalement, Les valeurs de ces histogrammes alimentent un réseau de neurones multicouches entraîné par l'algorithme de rétropropagation. La figure 4.2 présente le diagramme global de cette approche.

Les étapes à suivre dans cette démarche sont les suivants: 1) sélectionner les fonctions de forme de discrimination, 2) construire les fonctions de forme pour chaque objet 3D, comparer et classer toutes les distributions en utilisant d'une part les mesures de similarité puis d'autre part les réseaux de neurones artificiels.

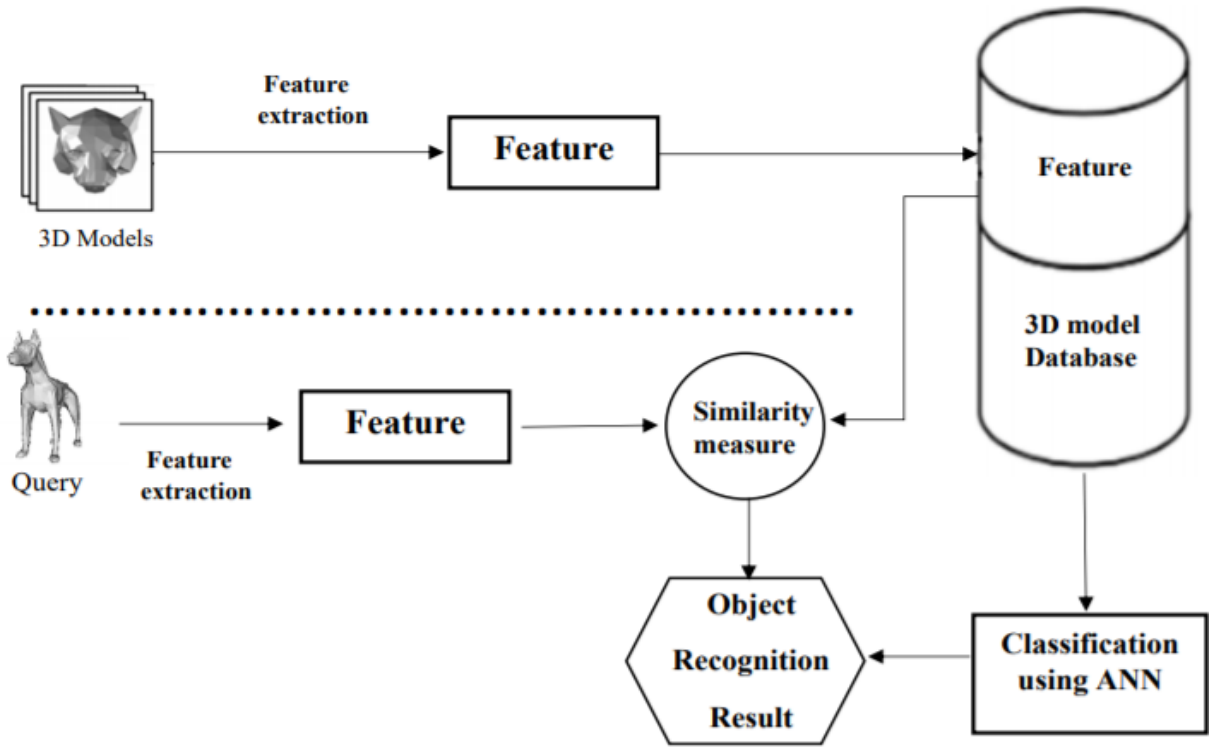


Figure 4. 2: Le diagramme global de notre approche.

4.5.1.1 Sélection de la fonction de forme

Le problème majeur consiste à sélectionner une fonction dont la distribution fournit une bonne signature pour la forme d'un modèle polygonal 3D. Normalement, la distribution devrait être invariante aux transformations géométriques, insensible au bruit, aux fissures, et à la suppression ou l'insertion des petits polygones. Cependant, pour un souci de clarté, nous nous concentrons sur un petit ensemble de fonctions de forme basées sur des mesures géométriques (par exemple, des angles, des distances, des surfaces et des volumes). Plus précisément, nous avons étudié les fonctions de forme suivantes (voir figure 4.3):

- A3: mesure l'angle entre trois points aléatoires de la surface d'un modèle 3D.
- D1: mesure la distance entre un point fixe et un point aléatoire de la surface. Nous utilisons le centre de gravité du modèle comme point fixe.
- D2: mesure la distance entre deux points aléatoires de la surface d'un modèle 3D.
- D3: Mesure la racine carrée de la surface du triangle entre trois points aléatoires de la surface d'un modèle 3D.
- D4: mesure la racine cubique du volume du tétraèdre entre quatre points aléatoires de la surface d'un modèle 3D.

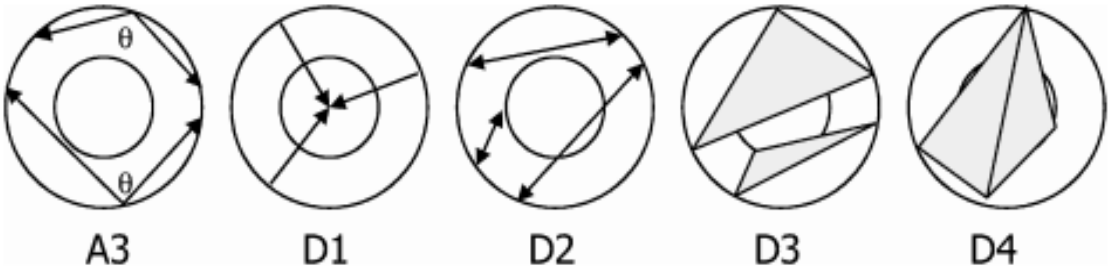


Figure 4. 3: Cinq fonctions de forme simples basées sur les angles (A3), les longueurs (D1 et D2), les surfaces (D3) et les volumes (D4).

Ces fonctions de forme ont été choisies principalement pour leur simplicité et leurs invariances. En particulier, Elles sont faciles à calculer et produisent des distributions invariantes aux mouvements rigides. Elles sont invariantes au pavage du modèle polygonal 3D, car les points sont sélectionnés de manière aléatoire à partir de la surface. Elles sont insensibles aux petites perturbations dues au bruit, aux fissures et au désagrément d'insertion de polygones, car l'échantillonnage est pondéré en fonction de la surface. Notons que la fonction de forme A3 est invariante à l'échelle, tandis que les autres doivent être normalisées afin de pouvoir les comparer.

Une fois que les cinq fonctions sont analysées, les auteurs [Osada2002] concluent que les distributions de distances entre deux points aléatoires donnent les meilleures caractéristiques. La distribution de forme D2 a plusieurs avantages, elle est à la fois invariante en rotation et en translation et peu coûteuse en calcul. Elle décrit la forme générale d'un objet, ce qui signifie qu'elle n'est pas facilement affectée par des déformations mineures de la forme. Par exemple, lorsqu'un modèle 3D de voiture est comparé à une version d'elle-même comportant 5% moins de polygones, les formes globales (les valeurs D2) des deux modèles restent très proches.

4.5.1.2 Construction des distributions de formes

Après avoir sélectionné une fonction de forme, le problème consiste à savoir comment calculer et stocker une distribution. Le calcul de la distribution D2 est facile si nous générons l'objet en un ensemble de points. L'objet doit être maillé avec le maximum de triangle. Le maillage est défini par la triangulation de l'objet 3D et donné sous forme d'une liste de points suivie d'une liste de facettes ou triangles. Par conséquent, l'un des problèmes qui doit nous préoccuper est la densité d'échantillonnage. Plus nous sélectionnons d'échantillons, plus nous pouvons reconstruire avec précision la distribution de la forme. Par ailleurs, le temps nécessaire pour

échantillonner une distribution de forme est linéairement proportionnel au nombre d'échantillons. Il existe donc un compromis précision/temps dans le choix de N. De même, un plus grand nombre de sommets donne des distributions avec une résolution plus élevée, tout en augmentant le coût de stockage et de comparaison de la signature de forme. Dans nos expériences, nous avons choisi de privilégier la robustesse en prélevant un grand nombre d'échantillons pour chaque groupe d'histogrammes. De manière empirique, nous avons constaté qu'en utilisant N = 1024^2 échantillons, B = 1024 cases et V = 64 sommets, nous obtenons des distributions de formes avec une variance suffisamment faible et une résolution assez élevée pour l'utilité de nos expériences.

Le deuxième problème concerne la génération d'échantillons. Comme nos fonctions de forme sont décrites de manière aléatoire sur la surface d'un modèle 3D polygonale, nous avons implémenté l'algorithme de triangulation de Delaunay [Ming2015] pour chaque triangle sélectionné avec les sommets (A; B; C), nous construisons un point sur sa surface en générant deux nombres aléatoires, r_1 et r_2, entre 0 et 1, et en évaluant l'équation suivante:

$$P = (1 - \sqrt{r_1})A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C \qquad (4.1)$$

Intuitivement, $\sqrt{r_1}$ définit le pourcentage du sommet A sur le bord opposé, tandis que r2 représente le pourcentage du long de ce bord (voir figure 4.4).

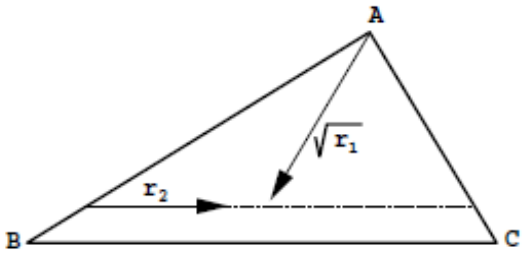


Figure 4. 4: Échantillonnage du point aléatoire dans le triangle.

La figure ci-dessous (voir la figure4.5) résume les étapes de construction de la distribution de forme D2. Cette dernière représente sous forme d'un histogramme normalisé la probabilité d'apparition d'une distance entre deux points aléatoires des faces triangulaires.

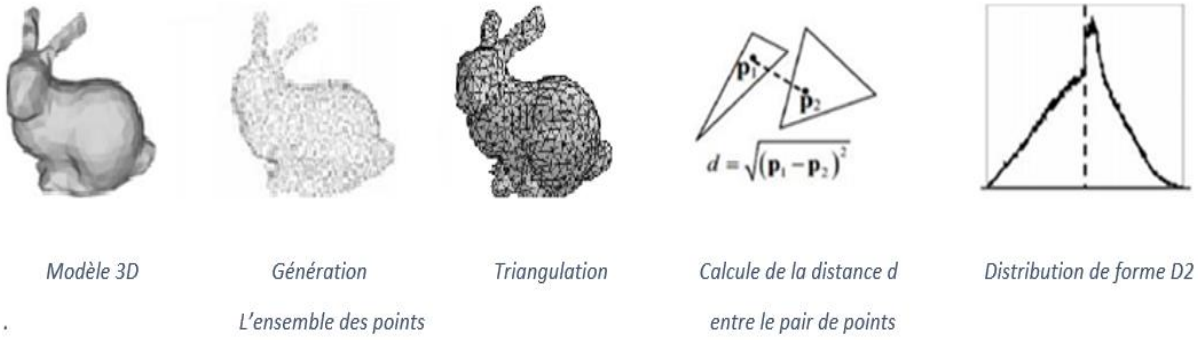


Figure 4. 5: Calcul de la fonction de forme D2 à partir du modèle polygonale 3D.

4.5.1.3 Reconnaissance des distributions

A partir du moment où nous avons construit des distributions de formes pour chaque objet 3D. Pour les deux sections suivantes, nous allons dans un premier temps les comparer pour produire une mesure de similarité. Ensuite nous allons classer ces distributions à l'aide des réseaux de neurones artificiels.

4.5.2 Comparaison des distributions de forme

Nous nous concentrons sur quelques mesures de similarité dans le cadre de la recherche des distributions similaires. Nous donnons ci-dessous les mesures utilisées lors de nos expériences.

- Distance euclidienne (L2) [Liu2006].
- Distance manhattan (L1) [Liu2006].
- Distance Chebyshev (L∞) [Liu2006].
- Intersection d'histogrammes (IH) [Muselet2004].
- The Earth Mover's Distance (EMD) [Ling2006].
- Distance de Kullback-Leibler (DKL) [Najjar2009].
- Statistique X2 (X2) [Liu2006].

4.5.3 Classification des distributions de forme

La possessions d'une large base d'objets 3D, rend la recherche des modèles les plus proches de l'objet requête de plus en plus difficile, ce qui nécessite une structuration des index pour minimiser l'espace de recherche. Afin de résoudre ce problème, la recherche d'objets 3D similaires est ramenée à un problème de classification. Particulièrement, en calculant la classe

d'objets 3D la plus proche de l'objet requête selon des critères spécifiques. Le résultat de recherche sera donc un groupe ou une classe de modèles tridimensionnels ayant des caractéristiques similaires à celles de l'objet requête. Les performances de la classification et de la reconnaissance ont été évaluées à l'aide de trois classifieurs:

- Réseaux de neurones artificiels (RNA);
- Plus proche voisin (KNN);
- Machine à vecteurs de support (SVM).

Ces trois classifieurs ont été annoncés en détails dans le chapitre 3. Pour le classifieur SVM, nous avons traité le problème comme une classification multi-classe.

4.5 Expérimentations

Nous avons développé une approche qui est basée sur la reconnaissance des objets à partir de leurs distributions de forme. Les modèles tridimensionnels les plus courants se trouvent sous format d'un maillage (VRML). Pour tester l'efficacité de notre approche, nous avons effectué une série d'expériences sur une base de données contenant différents modèles 3D. Nos objets ont été extraits de la base de données standard de référence de Princeton [Princeton2004]. La figure 4.6 illustre des exemples d'objets 3D de chacune des 12 classes.

Notre système de reconnaissance d'objets 3D peut être résumé dans la figure 4.2. Tout d'abord, pour chaque objet 3D, une étape de normalisation est effectuée. Ensuite, l'objet 3D est décrit par sa distribution de forme D2, qui sera stockée sous forme d'histogramme normalisé. Les valeurs de ces histogrammes alimentent un réseau de neurones multicouches entraîné par l'algorithme de rétropropagation. La figure 4.7 montre la diversité au sein des classes contenant des images de l'objet 3D du requin, chien, femme, lapin et squelette. Notez que les objets "femmes" sont visuellement similaires, alors que les chiens sont différents. Une autre classe, des lapins, est encore plus diversifiée: ils ont tous la même fonction, mais une forme différente.

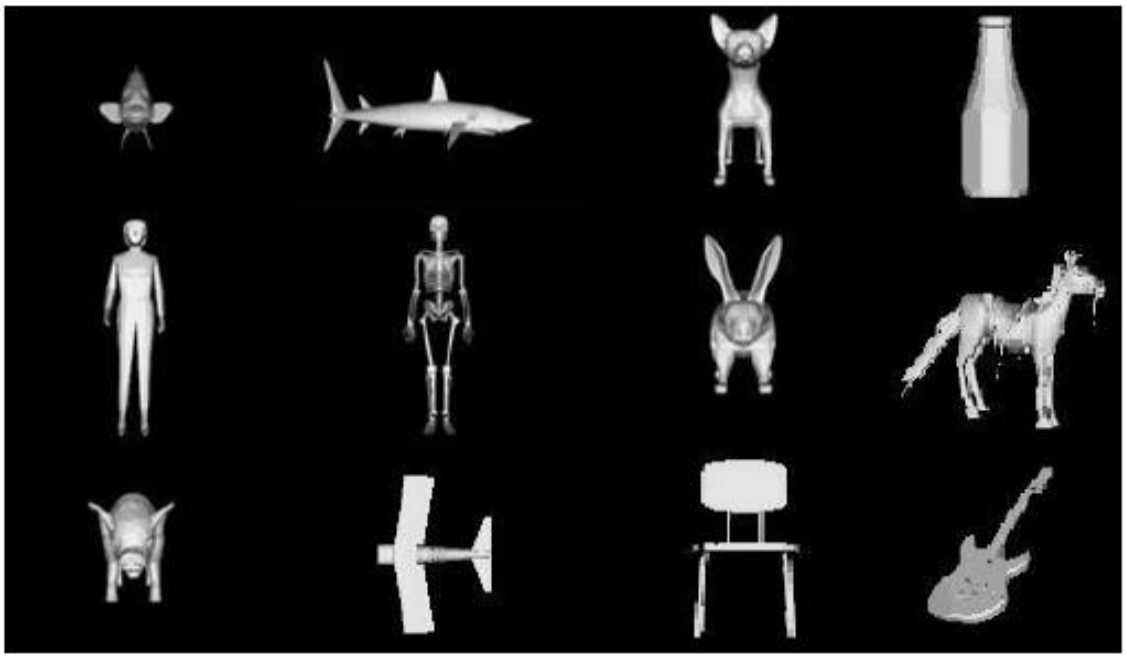


Figure 4. 6: Exemples de modèles 3D.

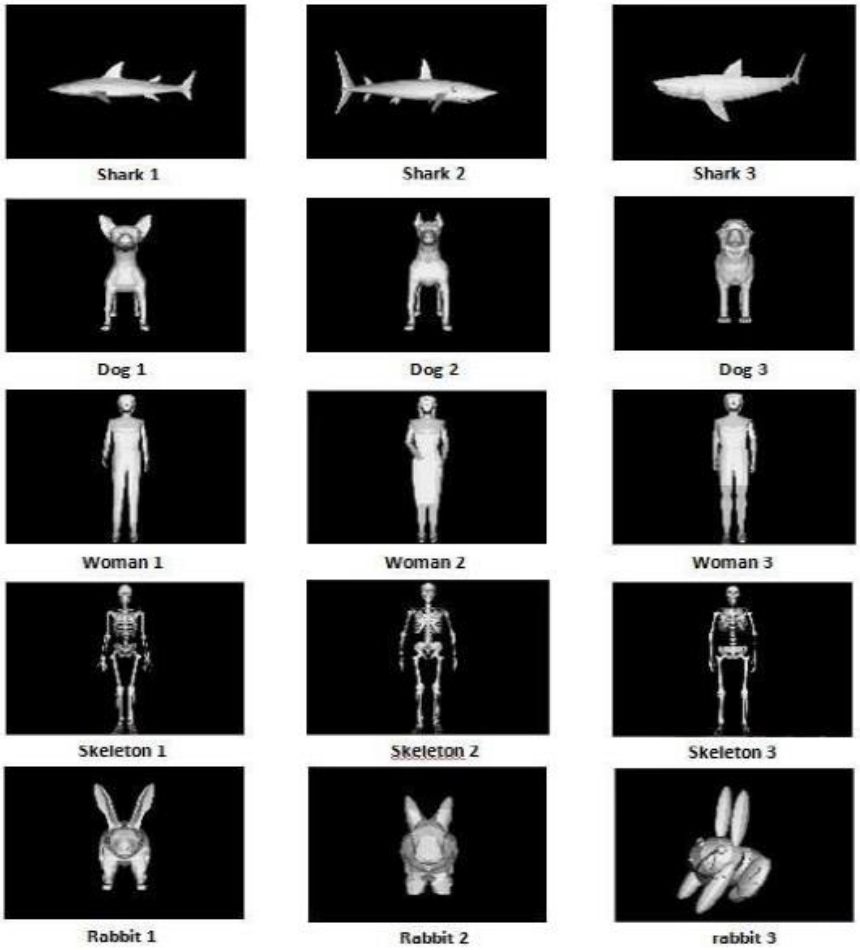


Figure 4. 7: Exemple de différentes classes de notre base de données.

Pour étudier la capacité de discrimination de la méthode, nous réalisons une expérience en calculant les distributions de formes D2 pour tous les modèles de notre base de données. La figure 4.8 montre quelques exemples. En examinant ces distributions qualitativement, nous trouvons que les distributions de formes pour la plupart des objets d'une même classe sont fortement corrélées, cela revient du fait que plusieurs courbes de la même classe apparaissent avec la même forme.

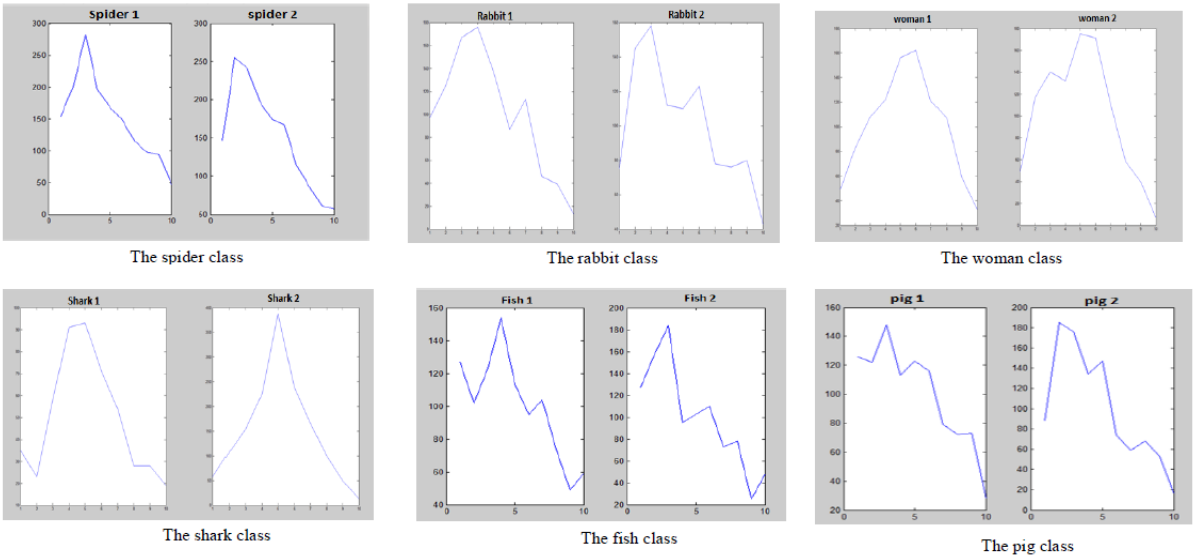


Figure 4. 8: Distributions de formes D2 pour certains objets. Chaque Plot représente la probabilité de la distribution d'une distance.

Dans le but d'effectuer une comparaison entre les histogrammes qui représentent nos objets 3D, nous employons plusieurs distances, à savoir: distance euclidienne, distance de Manhattan, distance de Chebyshev, Intersection d'histogramme, The Earth Mover's Distance, distance de Kullback-Leibler, statistique X2. La comparaison est effectuée entre l'objet "dog1" et l'objet "dog2" de la classe chien, "woman1" et "woman2" de la classe femme, "fish1" et "fish2" de la classe poisson, "skeleton1", "skeleton2" de la classe squelette et "pig1", "pig2" de la classe cochon. (Voir tableau 4.1).

Tableau 4. 1 Comparaison des distances.

Obj / D	Dog 1	Dog2	Woman1	Woman2	Fish1	Fish2	Skeleton1	Pig1	Pig2	Skeleton2	T.C
DKL	0	**617,58**	801,19	930,97	797,93	719,01	812,15	630.42	752,61	1028,9	**0.000332** s
EMD	0	4,2780e +04	1,6808e +05	1,8361e +05	1,4459e +05	**1,9864e +04**	1,6800e +05	8,4359e +04	1,2761e +05	1,8294e +05	0.000051 s
X^2	0	**0,14**	0,33	0,38	0,27	0,19	0,33	0,19	0.23	0.38	**0.000174** s
IH	0	0,26	0,10	**0,09**	0,12	0,33	0,09	0,23	0,15	0,09	0.000048 s
L_1	0	**319,70**	375,92	396,90	355,39	371,84	373,47	339,30	344,00	399,49	**0.000032** s
L_2	0	**12,46**	14,51	15,18	13,96	14,41	14,47	13.21	13,51	15,34	**0.000042** s
L_∞	0	1,14	1,12	1,12	1,11	1,19	**1,07**	1,14	1,15	1,15	0.000042 s

De la lecture du tableau 4.1, nous pouvons constater que les distances Kullback-Leibler, Statistique X2, Manhattan et Euclidienne offrent des meilleurs résultats en termes d'efficacité. Afin de mieux distinguer la distance la plus robuste, nous effectuons une normalisation des valeurs du tableau précèdent (Voir tableau 4.2):

Tableau 4. 2 Après la normalisation des distances.

Obj / D	Dog1	Dog2	Woman1	Woman2	Fish1	Fish2	Skeleton 1	Pig1	Pig2	Skeleton2	T.C
DKL	0	**0**	0.4464	0.7619	0.4385	0.2466	0.4730	**0.0312**	0.3283	1	**0.000332** s
X^2	0	**0**	0.7917	1.0000	0.5417	**0.2083**	0.7917	0.2083	0.3750	1.0000	**0.000174** s
L_1	0	**0**	0.7046	0.9675	0.4473	0.6535	0.6739	**0.2456**	0.3045	1.0000	**0.000032** s
L_2	0	**0**	0.7118	0.9444	0.5208	0.6771	0.6979	**0.2604**	0.3646	1.0000	**0.000042** s

Le tableau 4.2 montre que la distance euclidienne offre des meilleurs résultats en termes de temps de calcul et permet une forte discrimination entre les modèles 3D.

La recherche d'objets 3D similaires est ramenée à un problème de classification. Pour cela nous nous sommes basés sur un réseau de neurone artificiel à 3 couches, entraîné par l'algorithme de rétropropagation. La couche d'entrée est alimentée avec les distributions de forme. Une fois l'apprentissage est terminé, le réseau peut traiter de nouveaux objets. 12 classes ont été définies. Le tableau suivant (tableau 4.3) indique la sortie désiré pour chacune des douze classes:

Tableau 4. 3 Les sorties désirées pour chacune des 12 classes.

Classes	dénomination	Sorties désirés
1	Plan	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
2	Bouteille	(0, 1, 0, 0, 0, 0, 0, 0,0, 0, 0, 0)
3	Cochon	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0,0, 0)
4	Chaise	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
5	Cheval	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,0)
6	Chien	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,0)
7	Femme	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,0)
8	Guitare	(0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,0)
9	Lapin	(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,0)
10	Poison	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,0)
11	Requin	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,0)
12	Squelette	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1)

Dans la partie suivante, nous présentons les résultats et les performances du processus d'apprentissage et de test des réseaux de neurones désigné pour la classification et la reconnaissance des objets 3D. Notre base de données contient 120 objets. Nous avons divisé l'ensemble des données en deux sous bases, 70% pour la phase d'apprentissage et 30% restant de la base pour le test et l'évaluation de la qualité. Différentes architectures du réseau ont été testé. "La figure 4.9" montre l'architecture du réseau utilisé. Notons que si le nombre de neurones dans les couches cachées est trop élevé, le réseau effectuera un apprentissage par cœur des donnés (sur apprentissage). De ce fait, il aura du mal à généraliser des nouvelles données. Tandis que s'il est trop petit (sous apprentissage), il ne possédera pas assez de variables internes pour procéder à la reconnaissance. Le choix du nombre de neurones est un compromis entre ces deux aspects. Il est obtenu de manière empirique. Enfin, nous avons choisi des combinaisons de paramètres offrant le meilleur taux de reconnaissance. Donc, nous avons constaté, qu'avec deux couches cachés et 32 neurones dans chacune des couches, ce réseau nous a permis d'avoir le résultat le plus performant.

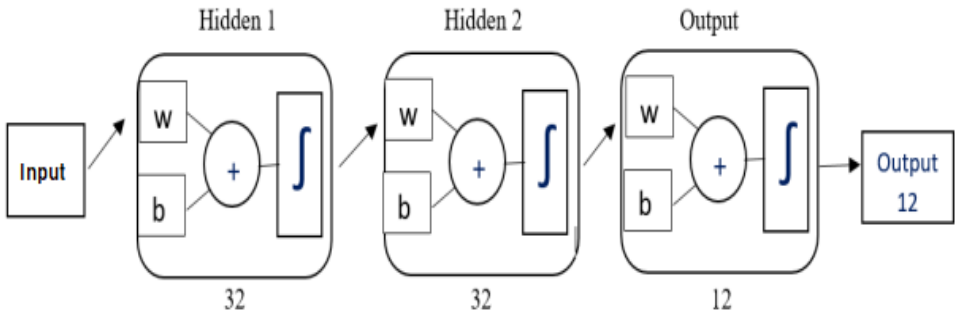


Figure 4. 9: L'Architecture du réseau de neurone utilisée.

A. Phase d'apprentissage

Pour démarrer le processus d'entraînement, les poids initiaux sont choisis de manière aléatoire. Ensuite, la phase apprentissage s'entame. Les entrées et les sorties désirées sont données. Le réseau traite les entrées et compare ses résultats aux sorties désirées. Les erreurs sont renvoyées pour que le système ajuste les poids qui contrôlent le réseau. Ce processus continuera jusqu'à ce que les poids soient ajustés et l'erreur à la sortie se stabilise à une valeur acceptable. L'idée consiste à faire apprendre au réseau de neurones plusieurs objets de chaque classe. A la fin de la phase d'apprentissage, notre réseau devrait être capable de reconnaître des objets non appris, des objets de la base de test. Ci-dessous, nous donnons la matrice de confusion qui correspondant à la phase d'apprentissage. Nous avons pu obtenir un taux de reconnaissance de 95,2% avec un taux d'erreur de 4,8% (voir tableau 4.4).

B. Phase de test

Il est recommandé d'utiliser une base de test indépendante de la phase d'apprentissage. Le pourcentage d'une bonne classification lors de la phase d'apprentissage donne une première indication. Cependant, la performance de la base de test est plus appropriée. Les 30% d'exemples inconnus de la base de test sont présentés au réseau au cours de la phase de test de façon à mesurer le degré de généralisation. Le tableau 4.5 montre la matrice de confusion de la base de test. Nous avons pu obtenir un taux de reconnaissance de 91,7% avec un taux d'erreur de 8,3%.

Finalement, nous comparons les résultats du classifieur RNA à d'autres méthodes de classification, telles que les méthodes Machine à Vecteurs Support (SVM) et les k-plus proche voisin (KNN).

Pour déterminer le noyau correct du SVM, nous avons examiné et comparé les performances du SVM avec différents types de noyaux; linéaires et non linéaires (polynôme et RBF). Cela pour les deux bases de données: d'apprentissage et de test (Voir tableau 4.6).

Nous avons conclu que les SVM linéaires étaient les plus performants en termes de taux de reconnaissance. De même, l'algorithme KNN a été testé avec plusieurs nombres de voisins dans l'intervalle [1–5]. Comme le montre le tableau 4.7, le RNA obtient le meilleur résultat par rapport aux autres clasifieurs (SVM et KNN) [Elhoufi'2018]. Cela prouve l'efficacité de cette technique pour la reconnaissance et la classification des objets 3D.

Tableau 4. 4 La matrice de confusion de la phase d'apprentissage.

Class	1	2	3	4	5	6	7	8	9	10	11	12	Rate Error
1	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**6** **7.1%**	0 0.0%	0 0.0%	**1** **1.2%**	0 0.0%	85.7% 14.3%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**6** **7.1%**	**1** **1.2%**	0 0.0%	0 0.0%	85.7% 14.3%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**7** **8.3%**	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**1** **1.2%**	**6** **7.1%**	0 0.0%	85.7% 14.3%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**1** **1.2%**	0 0.0%	0 0.0%	**6** **7.1%**	85.7% 14.3%
Rate Error	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	85.7% 14.3%	77.8% 22.2%	85.7% 14.3%	100% 0.0%	**95.2%** **4.8%**

Tableau 4. 5 La matrice de confusion de la phase test

Class	1	2	3	4	5	6	7	8	9	10	11	12	Rate Error
1	**3** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 2.8%	75.0% 25.0%
3	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	**1** **2.8%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	75.0% 25.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	**1** **2.8%**	0 0.0%	0 0.0%	0 0.0%	75.0% 25.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**2** **5.6%**	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**2** **5.6%**	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	0 0.0%	100% 0.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**3** **8.3%**	0 0.0%	100% 0.0%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	**2** **5.6%**	100% 0.0%
Rate Error	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	66.7% 33.3%	66.7% 33.3%	100% 0.0%	100% 0.0%	66.7% 33.3%	**91.7%** **8.3%**

Tableau 4. 6 Taux de reconnaissance obtenu en utilisant les SVM avec différents types de noyaux.

Datasets	Linear	Polynomial	RBF
Training set	86.1%	80.6%	80.6%
Testing set	76.6%	70.8%	75.3%

Tableau 4. 7 Les taux de reconnaissance.

RNN	SVM	KNN
91.7%	76.6%	77.3%

4.6 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur la problématique liée à la reconnaissance des objets 3D en se basant sur l'indexation 3D/3D. Plus précisément, nous nous sommes intéressés à la classification des objets en fonction de leurs caractéristiques globales, par l'utilisation du descripteur distribution de formes D2 et le réseau perceptron multicouche. L'idée était d'extraire, dans des objets 3D, des informations pertinentes sur leurs formes. Ces informations nous ont permis une classification efficace de ces objets 3D. Le défi de ce travail est de trouver une signature de forme qui peut être construite et classée rapidement, puis à apprendre au réseau de neurones artificiels plusieurs exemples de chaque classe possible, pour qu'à la fin de l'apprentissage, notre réseau peut reconnaître des objets non appris. La solution proposée a pu distinguer la forme générale de l'objet et semble plus adaptée à la recherche d'objets similaires dans des bases de données de formes multiples. Finalement, pour évaluer l'efficacité de notre méthode, nous avons comparé le classifieur à d'autres méthodes de classification. Les résultats de nos expériences sont encourageants et montrent la performance de l'approche proposée, avec un taux de reconnaissance supérieure à 91,7%.

Chapitre 5 : Segmentation sémantique par les réseaux de neurones convolutifs

5.1 Introduction

L'art de reconnaître des modèles et de tirer des conclusions sur des bases de connaissances antérieures sont des compétences qui manquent aux machines. La compréhension de scènes est une tâche visuelle reposant à l'heure actuelle sur une segmentation sémantique des images. Ce type de segmentation effectue une prédiction d'étiquette au niveau des pixels. Elle décrit le processus qui associe chaque pixel d'une image à une étiquette de classe (telle qu'une personne, une route, un ciel, une voiture, …). L'objectif est d'attribuer une étiquette de catégorie à chaque pixel d'une image ou, en d'autres termes, à partir d'une image d'entrée, nous voulons connaître tous les objets visibles, leurs positionnements et à quelle catégorie ils appartiennent. Il existe de nombreuses applications qui nécessitent une segmentation sémantique, citons à titre d'exemples la segmentation routière pour la conduite autonome et la segmentation des cellules cancéreuses pour le diagnostic médical. La figure 5.1 illustre un exemple d'image et son résultat de segmentation sémantique créé par la méthode de [Frôhlich2013] Où le résultat doit être proche de l'image de la vérité terrain.

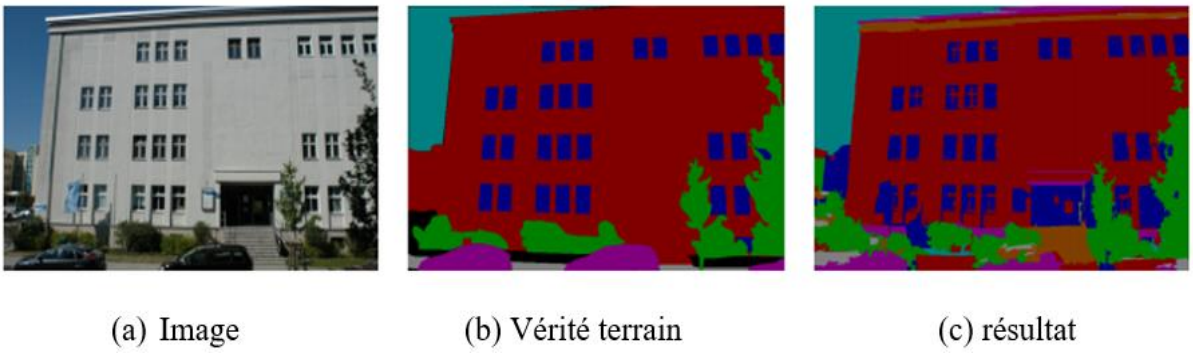


Figure 5. 1: Exemple d'algorithme de segmentation d'image sémantique. Images prises de [Frôhlich2013].

Ces dernières années, les réseaux de neurones profonds convolutifs (DCNN) ont montré de bonnes performances dans plusieurs tâches de reconnaissance d'images. Particulièrement, dans le domaine de la segmentation sémantique [Long2015, Chen2017, Zhao2018]. Contrairement aux autres techniques d'apprentissage supervisé, les CNNs déterminent les caractéristiques de chaque image et c'est là que leur force réside. La réalisation de ces réseaux repose principalement sur la conception complexe de leurs modèles, qui ont une largeur et une profondeur considérables ainsi qu'une longue durée d'inférence. Cependant, les intérêts récents dans de nombreuses applications du monde réel, telles que la conduite autonome, la réalité augmentée, les interactions robotiques et la surveillance intelligente, ont généré une forte demande des systèmes de compréhension de scène capables de fonctionner en temps réel. Il est

donc primordial de développer des réseaux de convolution efficaces pour la segmentation sémantique en temps réel.

Dans ce chapitre, nous proposons un nouveau module basé sur les progrès récents permettant une réduction importante des paramètres (nombre de poids et le nombre de multiplications nécessaire) avec une faible dégradation des performances. Notre module repose sur l'utilisation des techniques efficientes telles que « depthwise separable convolution », « shuffled convolution » et « grouped convoltuion ». Ces techniques permettent de réduire considérablement le nombre de paramètres avec une dégradation minime des performances. Nous avons également utilisé la technique dense connectivité (DenseNet) présentée par [Gao2017] et nous l'avons adapté par la suite à la segmentation sémantique en temps réel. Bien que DenseNet ait été initialement créé pour les problèmes de classification, nos expériences montrent que sa capacité à rassembler les caractéristiques extraites de différentes couches et à agréger les informations multi-échelles est naturellement bénéfique pour les tâches de segmentation. Cette structure peut également réduire considérablement le nombre de paramètres. La convolution dilatée est également utilisée. L'idée est d'élargir le champ de réception de notre réseau grâce à ce type de convolution, afin de conserver la résolution de la carte de caractéristiques et d'éviter de perdre des informations spatiales. Finalement, nous validons notre architecture à travers un ensemble d'expériences appliquées sur la base de données Cityscape.

Ce chapitre est organisé comme suit. Premièrement, nous allons présenter les notions en relation avec les réseaux de neurones profonds convolutifs. A savoir les différents blocs de construction et les types de convolution. Ensuite, nous allons introduire l'application des réseaux convolutifs dans la segmentation sémantique en temps réel. Finalement, nous allons présenter la conception de l'architecture du réseau proposé et les résultats expérimentaux visant à évaluer la robustesse et la rapidité de ce réseau.

5.2 Réseaux de neurones convolutionnels

En apprentissage automatique, un réseau de neurones convolutionnel (ou réseau de neurones convolutif ou réseau de neurones à convolution, ou CNN pour Convolutional Neural Networks ou ConvNet) est un type de réseau de neurones artificiels multi-couches. Ils sont souvent utilisés pour la reconnaissance de forme [Girshick2014], et comme son nom l'indique, son concept s'inspire de la structure neuronale du cortex visuel animal. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors

du pavage du champ visuel. L'architecture d'un CNN est composée de deux phases principales (voir figure 5.2).

La première phase fait la particularité de ce type de réseaux de neurones, puisqu'elle fonctionne comme un extracteur de caractéristiques des images. L'image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions qui seront par la suite normalisées et/ou redimensionnées.

Ce processus peut être répété plusieurs fois : on filtre les cartes de convolution obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles cartes de convolution à normaliser et redimensionner, et qu'on peut filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières cartes de convolution sont mises à plat et concaténées en un vecteur de caractéristiques. Ce vecteur définit la sortie de la première phase, et l'entrée de la seconde.

La seconde phase en un réseau de neurones multicouches pour la classification. Les valeurs du vecteur en entrée sont transformées pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur contient autant d'éléments qu'il y a de classes. Chaque élément est donc compris entre 0 et 1. Ces probabilités sont calculées par la dernière couche de cette phase, qui peut utiliser, comme fonction d'activation, la fonction logistique (classification binaire) ou la fonction softmax (classification multi-classe). Comme pour les réseaux de neurones ordinaires, les poids de connexion des couches sont déterminés par rétropropagation du gradient.

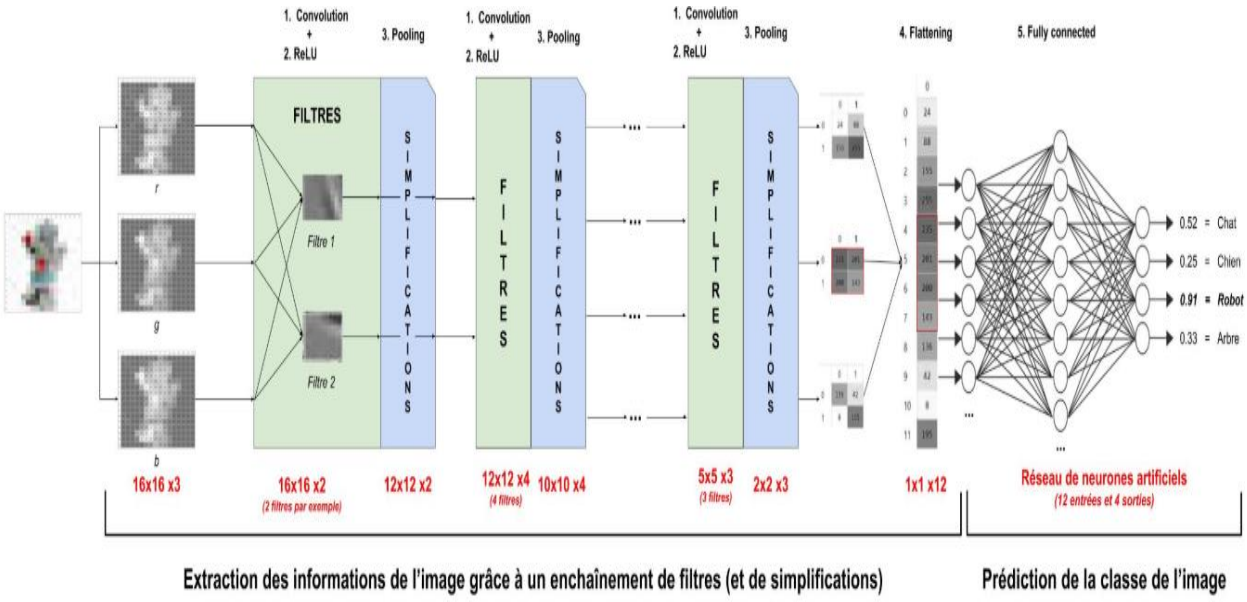


Figure 5. 2: Les deux phases d'un CNN: l'extraction de l'information et l'analyse de cette information.

Les réseaux de neurones convolutifs visent à limiter le nombre d'entrées tout en conservant la forte corrélation des images. Ils se basent sur les notions suivantes [Karpathy2015]:

1. ***Champ réceptif :*** Contrairement aux perceptrons multicouches qui ne restreignent pas le nombre de neurones d'entrée et comprennent généralement un grand nombre de paramètres. CNNs exploitent le fait que les pixels voisins contiennent des informations supplémentaires les uns des autres que ceux plus éloignés et, par conséquent, ils limitent les neurones d'entrée à de plus petites régions. Cette zone restreinte est appelée champ réceptif.
2. ***Partage des poids:*** Dans les réseaux de neurones convolutifs, les paramètres de filtre d'un champ réceptif donné sont identiques pour tous les autres champs réceptifs de l'image. Cette propriété permet de réduire considérablement la complexité en diminuant le nombre de paramètres à apprendre, et d'avoir ainsi des architectures multi-couches qui opèrent sur des entrées de grande dimension tout en étant de taille réaliste.
3. Des opérations de ***sous-échantillonnage*** qui permettent de réduire la sensibilité aux translations, ainsi que de réduire le coût du traitement.

5.2.1 Différents modules d'un réseau de neurones convolutif

- Couche de convolution (CONV)

La couche de convolution calcule la combinaison linéaire des neurones d'entrée et des poids de modèle. Son objectif est de repérer la présence d'un ensemble de caractéristiques dans les images reçues en entrée. A cet égard, nous réalisons un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la caractéristique sur l'image, et de calculer le produit de convolution entre la caractéristique et chaque segment de l'image balayée.

Nous obtenons pour chaque paire (image, filtre) une carte de convolution, qui nous indique où se localise la caractéristique dans l'image. Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (Figure 5.3): la 'taille du filtre', le 'pas' et la 'marge'.

- ✓ Taille du filtre (Kernel) est le nombre de noyaux de convolution (ou nombre de neurones associés à un même champ réceptif).
- ✓ Le pas (stride) contrôle le chevauchement des champs réceptifs. Plus le pas est petit, plus les champs réceptifs se chevauchent et plus le volume de sortie sera grand.

- ✓ La marge (à 0) (zero padding) : Parfois, il est obligé de mettre des zéros à la frontière du volume d'entrée. Cette marge permet de contrôler la dimension spatiale du volume de sortie.

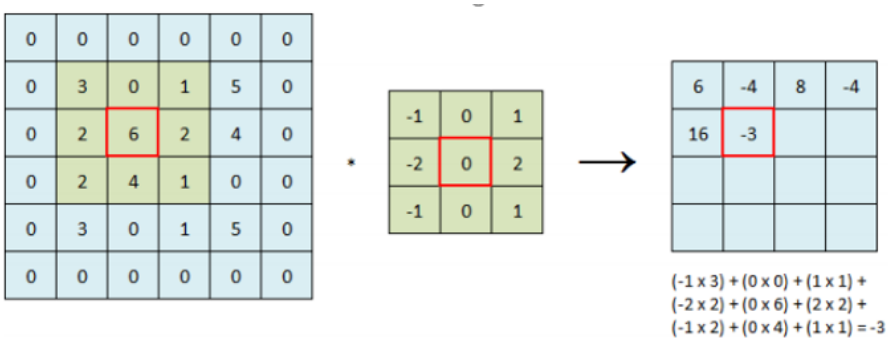


Figure 5. 3: Convolution par filtre sobel (détecteur de bord) avec une taille de noyau de 3*3, une marge de 1 et un pas de 1.

- Couche de pooling (POOL)

Cette couche effectue un sous-échantillonnage sur chaque carte de convolution (appelée en anglais feature map). Son objectif principal est de réduire leur dimensionnalité spatiale tout en maintenant des informations importantes. A titre d'impact, le nombre de paramètres est réduit, ainsi que le risque de sur apprentissage. Il existe différents types de pooling (moyenne, somme, etc.), mais le pooling maximal ‘max pooling’ est le plus utilisé. Supposons que nous glissions une fenêtre vide 2x2 sur une carte de convolution avec un pas de 2. Pour chaque pas, nous représentons cette sous-région par le plus grand élément de la fenêtre. Le concept du max pooling est illustré dans la figure 5.4 [Adit2016].

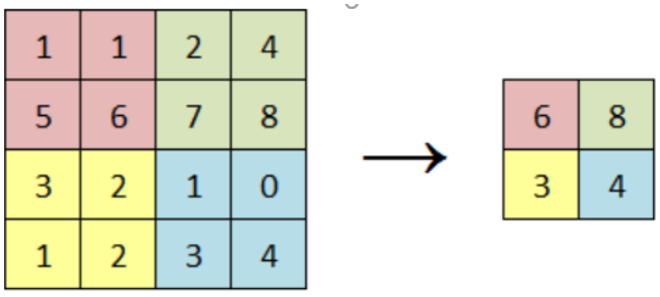


Figure 5. 4: Pooling maximale de taille 2*2 et avec un pas de 2.

- Couche de correction/Fonction d’activation

Il existe différentes fonctions d’activation permettant la non-linéarité dans les différentes couches des CNNs. Parmi les plus utilisées:

- La fonction sigmoïde utilisée notamment dans le perceptron multicouche (voir chapitre 3, section 3.2.3).
- La fonction ReLU [Richard2000] (*Rectified Linear Units*) (Figure 5.5) définie par :

$$RELU(x) = max(0, x) \qquad (5.1)$$

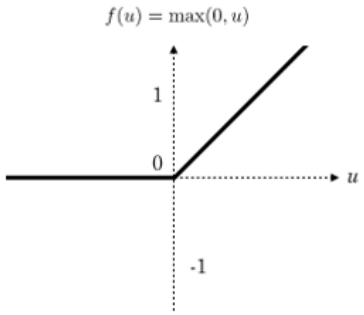


Figure 5.5: La fonction RELU.

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Cette fonction est la plus utilisée dans les CNNs profonds [Agarap2019]. Son principal avantage réside dans le fait qu'elle renvoie un gradient constant pour une grande entrée permettant ainsi d'apprendre plus rapidement et elle permet de réduire les problèmes de disparition de gradient. Il existe d'autres fonctions d'activation de la même famille que les ReLU comme la LReLU [Maas2013], la PReLU [He2015] et la eLU [Clevert2016].

- Le dropout

La couche de dropout a été introduite par [Srivastava2014], elle est utilisée pendant la phase d'apprentissage, dans le but d'éviter le sur-apprentissage (over fitting). Elle s'agit de masquer une partie du réseau aléatoire durant les différentes itérations de l'apprentissage. Pour cela, chaque neurone du réseau a une probabilité p d'être abandonné (dropped) à chaque passage. La figure 5.6 montre à gauche le réseau initial, et à droite un exemple avec 7 neurones abandonnés, notés X. En d'autres termes, le dropout permet au réseau d'apprendre des sous-réseaux contenant moins de paramètres. Une fois l'apprentissage terminé, tous les neurones sont réactivés.

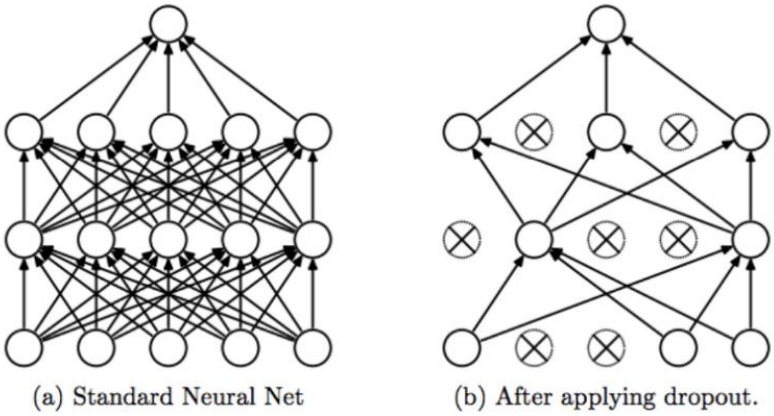


Figure 5.6: Illustration du dropout. Source [Srivastava2014].

- La batch normalisation

Cette technique a été introduite par [Ioffe2015], dont le but est d'accélérer l'apprentissage des CNNs. L'idée est de normaliser les entrées de chaque couche dans l'intention que les distributions de celles-ci soient de moyenne nulle et de variance unitaire. Pendant l'apprentissage, les couches de batch normalisation apprennent des paramètres (un facteur d'échelle et un biais) permettant d'ajuster cette normalisation. Autrement dit, lors de l'apprentissage, si le réseau considère que la distribution normalisée n'est pas adaptée pour une couche donnée, il ajuste les paramètres.

- Le flattening (ou mise à plat)

C'est la dernière couche de la phase « extraction des caractéristiques», le flattening consiste simplement à mettre bout à bout toutes les images (matrices) que nous avons pour en déduire un (long) vecteur (Figure 5.7). Les pixels sont récupérés ligne par ligne et ajoutés au vecteur final.

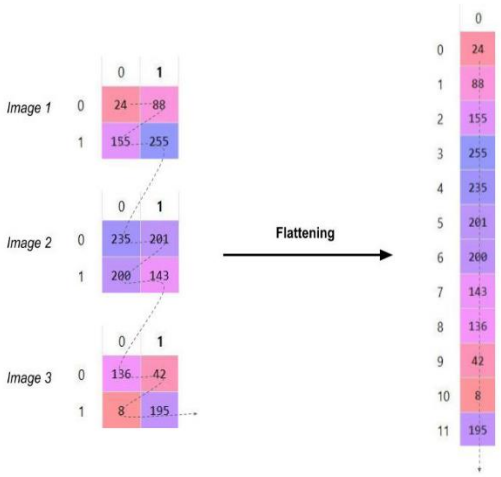


Figure 5.7: Mise à plat des images finales.

- Couche entièrement connectée (FC)

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Tous les neurones d'entrée sont connectés par des synapses à tous les neurones de sortie. Cela correspond en fait au perceptron de Rosenblatt [Rosenblatt1957]. La non-linéarité est appliquée sous la forme d'une fonction d'activation sur le vecteur de sortie. Une couche entièrement connectée peut se conceptualiser comme une simple multiplication matricielle, transformant un vecteur de dimensions 1×N en vecteur de dimensions 1×M par le biais d'une matrice de poids N × M.

5.2.2 Types de convolutions

- Convolution standard

L'opération de convolution est une opération de multiplication de matrice par élément. Où l'une des matrices est l'image et l'autre le filtre ou le noyau qui transforme l'image en carte de convolution (Figure 5.8).

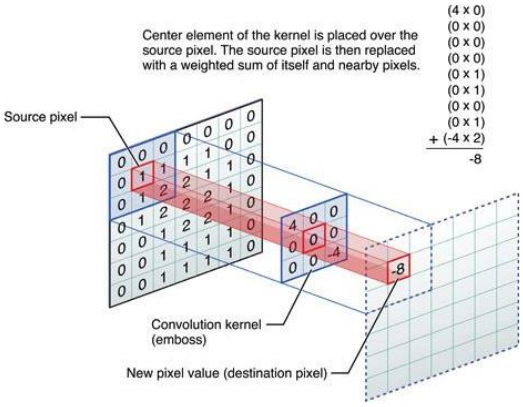


Figure 5. 8: Une simple convolution.

- Les convolutions séparables

Il existe deux principaux types de convolutions séparables: les convolutions séparables en espace (en anglais Spatial Separable Convolutions) [Mamaleté2012] et les convolutions séparables en profondeur (en anglais depthwise separable convolutions) [Chollet2017].

➢ **Les convolutions séparables en espace**

Cette convolution illustre l'idée de séparer une convolution en deux. Sa nomination vient du fait qu'elle traite principalement les dimensions spatiales d'une image et du noyau: la largeur et la hauteur. (La troisième dimension, « profondeur », correspond au nombre de canaux de

chaque image). Elle divise simplement un noyau en deux noyaux plus petits. Le cas le plus courant serait de diviser un noyau 3*3 en un noyau 3*1 et 1*3 (Figure 5.9), comme exemple:

$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

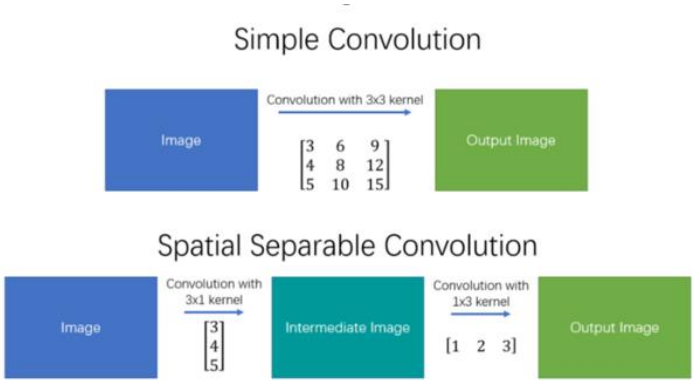


Figure 5.9: La convolution séparable simple et spatiale.

Au lieu de faire une convolution avec 9 multiplications, nous faisons deux convolutions avec 3 multiplications chacune (6 au total). Donc, nous bénéficions de moins de multiplications. Par conséquent, la complexité informatique diminue et le réseau peut fonctionner plus rapidement. Le principal problème de cette convolution est que pas tous les noyaux peuvent être «séparés» en deux noyaux plus petits.

- **Les convolutions séparables en profondeur**

Contrairement aux convolutions séparables en espace, les convolutions séparables en profondeur fonctionnent avec des noyaux qui ne peuvent pas être « factorisés » en deux noyaux plus petits. Par conséquent, il est plus couramment utilisé. Sa nomination est dû au fait qu'il concerne non seulement les dimensions spatiales, mais aussi la dimension de profondeur - le nombre de canaux -. Une image d'entrée peut avoir 3 canaux: chacune correspond aux couleurs RVB (rouge, bleu, vert). Cette convolution est divisée en deux types : depthwise convolution et pointwise convolution.

- **Depthwise convolution/ Convolution profonde**

Contrairement au CNN standard, où la convolution est effectuée pour tous les M canaux de l'image. Depthwise convolution est appliquée à un seul canal à la fois. Donc, les filtres seront de taille Dk*Dk*1. Étant donné qu'il y a M canaux dans les données d'entrée, alors M filtres sont nécessaires. La sortie sera de taille Dp*Dp*M, comme montré sur la (Figure 5.10).

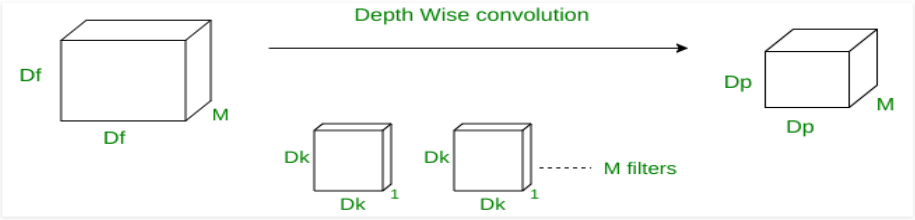


Figure 5.10: Depth-wise convolution.

- **Pointwise convolution/ Convolution ponctuelle**

En mode point par point, une opération de convolution 1*1 est appliquée sur les M canaux. La taille du filtre pour cette opération sera donc de 1*1*M. Disons que nous utilisons N filtres de ce type, la taille de sortie devient Dp*Dp*N (Figure 5.11). Elle est utilisée pour réduire le nombre de canaux d'entrée [He2016].

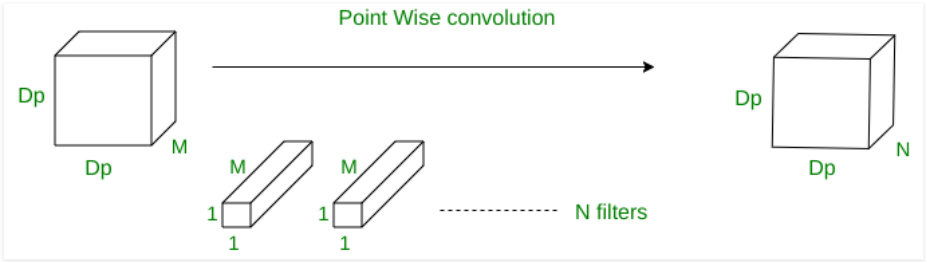


Figure 5. 11: point-wise convolution.

- Convolution dilatée/Dilated convolution

Les convolutions régulières ont des filtres exclusivement contigus, ce qui signifie qu'il n'y a aucun espace entre leurs cellules. Yu et Koltun ont introduit une extension aux convolutions régulières appelée convolution dilatée [Yu2016]. Son principe est d'insérer des zéros entre deux valeurs consécutives de filtre [Chen2015], [Zhao2017] et peut être définie comme suit:

$$O(i,j) = \sum_{i=-M}^{M} \sum_{i=-N}^{N} W(i,j) I(x - i * r, y - j * r) \qquad (5.2)$$

Où I est une image d'entrée, O est une image de sortie, W est un filtre de convolution et r est un taux de dilatation. Ce type de convolution est capable d'élargir efficacement le champ réceptif sans augmenter le nombre de paramètres. Par exemple, la taille effective d'un filtre de convolution n × n avec un taux de dilatation r est égal à [r (n-1) +1] × [r (n-1) +1]. (5.3)

Un filtre 3*3 avec un taux de dilatation de 2 aura le même champ de vision qu'un filtre 5*5, tout en utilisant seulement 9 paramètres. Cela offre un champ de vision plus large au même coût de calcul (Figure 5.12). Les convolutions dilatées sont particulièrement populaires dans le domaine de la segmentation en temps réel.

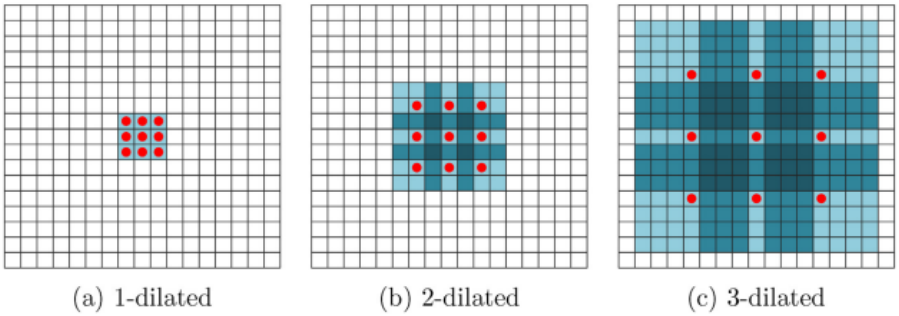


Figure 5. 12: Convolutions dilatées avec différents taux de dilatation [Yu2016]: (a) taux = 1 et champs récepteurs=3 × 3, (b) taux=2 et champs récepteurs=7 × 7 et c) taux= 3 et champs récepteurs=15 × 15.

- Convolution transposée/ Transposed convolution

Les convolutions transposées - également appelées déconvolutions. Elles fonctionnent en permutant les passes en avant et en arrière d'une convolution. Il est à noter que le filtre définit une convolution, mais qu'il s'agisse d'une convolution directe ou transposée est déterminé par la manière dont les passes en avant et en arrière sont calculées. Lorsque nous appliquons une convolution, nous réduisons les dimensions spatiales de l'image. Les déconvolutions sont effectuées après la couche de convolution afin de conserver la taille de l'image de sortie comme auparavant [Fu2017].

- Convolution groupée/Grouped convolution

Elles ont été initialement mentionnées dans AlexNet [Krizhevsky2012], puis réutilisées dans ResNeXt [Saining2017]. La principale motivation de telles convolutions est de réduire la complexité de calcul en divisant les caractéristiques sur les groupes.

- Convolution Shuffled

[Zhang2017, Zhao2017] introduisent le shuffling de canaux afin de mélanger de manière aléatoire la sortie de la convolution de groupe.

- Convolution Asymétrique/Asymetric convolution

C'est une forme factorisée de la convolution standard à deux dimensions, nxn en deux noyaux unidimensionnels nx1 et 1xn [Szegedy2015, Szegedy[1]2016], dont le but est de réduire considérablement le coût de calcul.

Par rapport à la convolution standard avec une taille de noyau de 3, le coût de calcul peut être réduit de 33%, tandis que la perte de précision est considérablement faible.

5.3 Les réseaux convolutifs pour la segmentation sémantique

En raison de leur énorme succès dans la classification des images, les chercheurs ont été motivés à les appliquer à des problèmes de classification structurés tels que la segmentation sémantique. Au cours de ces dernières années, il y a eu différentes approches, nous citons entre autres; le réseau entièrement convolutionnel (FCN) [Long2015] et l'architecture encodeur / décodeur appelé SegNet [Badrinarayanan2015]

5.3.1 Réseaux entièrement convolutionnels (FCN)

Les réseaux entièrement convolutionnel (FCN) utilisent l'efficacité naturelle des réseaux neuronaux convolutifs conçus pour la classification des images. Un CNN existant peut-être transformé en un FCN [Long2015]. Cela se fait en changeant simplement ses couches entièrement connectées en couches de convolution. La conversion permet au réseau de classer chaque région locale dans une image et de produire ainsi une carte de classification grossière. Pour inverser le sous-échantillonnage causé par les couches de regroupement, la carte est étendue à la taille de l'image d'origine. Cela se fait par une seule étape de sur-échantillonnage utilisant une interpolation bilinéaire (Figure 5.13).

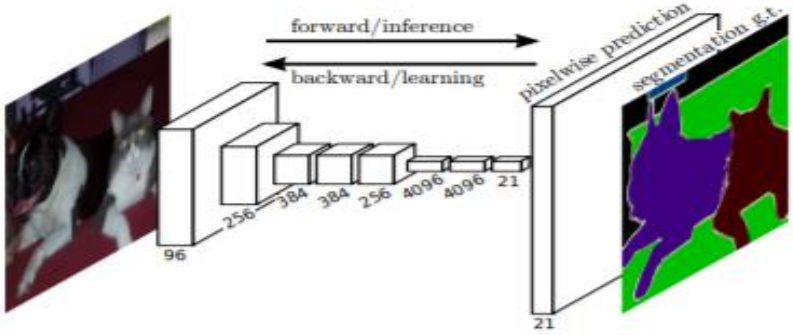


Figure 5. 13: Les réseaux entièrement convolutifs peuvent efficacement apprendre à faire des prédictions par pixel, source [long2015].

5.3.2 L'architecture encodeur / décodeur

SegNet [Badrinarayanan2015] est un réseau complet entièrement convolutif doté d'une architecture codeur-décodeur [Ronneberger2015] [Paszke2016] [Romera2018]. Le modèle comprend un codeur, un décodeur correspondant et un classifieur pixel par pixel. Le codeur est presque identique à l'architecture du réseau de neurones convolutif VGG16 mentionné dans [Simonyan2015]. Le décodeur est fondamentalement une version inversée du codeur. Au lieu

de regrouper les couches, le décodeur utilise des couches d'échantillonnage ascendant (Figure 5.14).

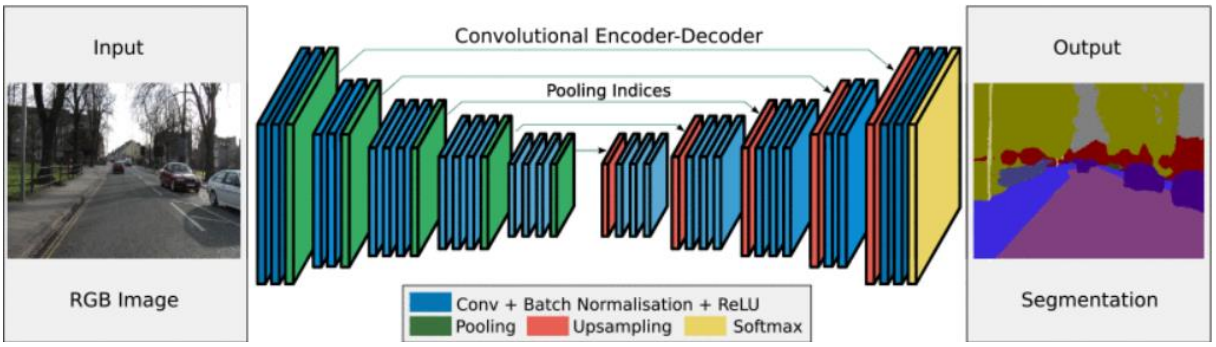


Figure 5. 14: Une illustration de l'architecture SegNet. Il n'y a pas de couches entièrement connectées et par conséquent, il s'agit uniquement de convolution. Un décodeur échantillonne ses entrées. Les cartes de caractéristiques de sortie finales du décodeur sont envoyées à un classifieur soft-max pour une classification par pixel, source [Badrinarayanan2015].

5.3.3 Autres architectures

Différents réseaux de segmentation basés sur CNN ont été proposés, tels que les réseaux de neurones récurrents multidimensionnels [Graves2007], les hypercolonnes [Hariharan2015], les représentations basées sue les régions [Dai2015] [Caesar2016] et les réseaux en cascades [lin2017]. Plusieurs techniques de support, associées à ces réseaux, ont été utilisées pour obtenir une grande précision, notamment l'assemblage des caractéristiques [Chen2018], l'apprentissage en multi-étapes [Long2015], les données d'apprentissage supplémentaires provenant d'autres ensemble de données [Zhao2017] [Chen2018], Post-traitement basé sur CRF [Chen2018] et ré-échantillonnage des caractéristiques basé sur une pyramide [Zhao2017].

5.4 Techniques utilisées dans la segmentation sémantique en temps réel

La plupart des méthodes décrites ci-dessus souffrent du nombre de paramètres trop élevé [lin2017], ou du grand nombre d'opérations en virgule flottante [Chen2017] ou des deux [long2015, Yang2017, Paszke2016]. Ces problèmes constituent un inconvénient majeur dans des applications nécessitant un traitement en temps réel.

Pour les atténuer, il existe plusieurs approches, par exemple Li et al. [Li2017] proposent une approche en cascade où, dans chaque niveau progressif de la cascade, seule une certaine partie des pixels est traitée en définissant explicitement un seuil strict sur les sorties de classifieur intermédiaire. Ils démontrent des performances de 14,3 FPS avec une résolution d'entrée de 512×512 et une moyenne de 78,2% MIoU sur la base de données PASCAL VOC [Everingham2010]. Nous notons également que plusieurs autres réseaux de segmentation en temps réel ont suivi l'architecture encodeur-décodeur, mais n'ont pas été en mesure d'acquérir des performances décentes. Concrètement, SegNet [badrinarayanan2015] a atteint 57.0% MIoU pour 40 FPS et avec des entrées de taille 360×480 sur la base CityScapes, tandis qu'ENet [Paszke2016] a pu effectuer une inférence de 20 FPS sur des entrées de taille 1920 × 1080 et avec une moyenne de 58,3% MIoU.

Ces dernières années, la nécessité de mettre en place des réseaux de neurones profonds en temps réel et dans diverses applications a augmenté. Différentes techniques, telles que la factorisation des convolutions, la compression de réseau, les réseaux à faible débit, et la convolution dilatée ont été proposées pour accélérer les réseaux de neurones convolutifs. Dans cette section, nous décrivons brièvement ces approches.

- **La factorisation des convolutions:** Cette technique décompose l'opération de convolution en plusieurs étapes pour réduire la complexité de calcul (Voir l'exemple cité dans **Les convolutions séparables en espace**, section 5.2.2**).** Cette factorisation a montré avec succès son potentiel de réduction de la complexité de calcul des réseaux CNN profonds (à titre d'exemple, Inception [Szegedy2015, Szegedy[1]2016, Szegedy[2]2016], réseau factorisé [Jin2O14], ResNext [Xie2017], Xception [Chollet2017] et MobileNets [Howard2017]).
- **La compression des réseaux:** La compression est une autre approche permettant de créer des réseaux efficaces. Ces méthodes utilisent des techniques telles que le hachage [Chen2015], la quantification vectorielle [Wu2016] et le rétrécissement [Zhao2017, Jaderberg2014] dans le but de réduire la taille du réseau pré-entrainé.
- **Réseaux à faible débit:** Une autre approche dans le même cadre des réseaux efficaces consiste à utiliser des réseaux à faible débit. Ces réseaux quantifient les poids pour réduire la taille et la complexité du réseau (nous citons d'exemple, [Rastegari2016, Courbariaux2016, Hubara2016]).
- **Convolution dilatée :** Comme nous avons déjà vu dans la section 5.2.3, la convolution dilatée est une forme spéciale des convolutions standards dans

lesquelles le Champ réceptif est augmenté en insérant des zéros entre chaque pixel du filtre de convolution. Le taux de dilatation spécifie le nombre de zéros entre les pixels. Cependant, cette technique nous permet de réduire le coût de calcul tout en augmentant la taille effective du filtre.

5.5 Contribution à la segmentation sémantique en temps réel

Comme nous l'avons mentionné précédemment, la réalisation des réseaux de neurones convolutionnels repose principalement sur la conception complexe de leurs modèles, qui ont une largeur et une profondeur considérables, qui nécessitent un grand nombre de paramètres et une longue durée d'inférence. Toutefois, les intérêts récents dans de nombreuses applications du monde réel ont généré une forte demande pour des systèmes de segmentation sémantique en temps réel. A cet égard, Il est donc nécessaire de développer une architecture d'un réseau convolutionnel qui nous garantit un compromis entre la vitesse et la précision.

La figure 5.15 illustre le défi que représente la conception des réseaux de neurones convolutifs en prenant en compte à la fois l'efficacité et la rapidité. Par exemple, la plupart des méthodes les plus performantes, telle que PSPNet [Zhao2017] se concentre sur l'amélioration de la précision au détriment de l'augmentation importante des coûts de calcul. En revanche, certaines autres approches, telles qu'ENet [Paszke2016] et EDANet [Lo2018], insistent sur la vitesse, mais leur précision diminue de façon remarquable. L'objectif est de créer un modèle suffisamment rapide pour gérer un débit en temps réel (au moins 30 images par seconde) et aussi précis que possible. Notre architecture ne figure pas uniquement parmi les rares systèmes dont la vitesse d'inférence dépasse 30 FPS, mais aussi garantit une précision satisfaisante.

5.5.1 Architecture de notre réseau

Dans cette section, nous développons une architecture de segmentation sémantique en temps réel permettant une utilisation efficace des paramètres par rapport aux architectures existantes, L'architecture de notre réseau convolutif est illustrée dans la figure 5.16. Elle se compose de trois blocs de sous-échantillonnage (downsampling), de deux blocs de notre module (Voir figure 5.18) et d'un léger décodeur. Les deux premiers blocs de sous-échantillonnage se succèdent au début de l'architecture et utilisent la même structure que le premier bloc de l'architecture ENet [Paszke2016] (Voir figure 5.17). Tandis que, le troisième bloc de sous-échantillonnage n'est qu'une convolution standard 3x3 avec un pas de 2. Les autres blocs de

notre réseau sont composés respectivement de cinq et neuf de nos modules connectés de manière dense.

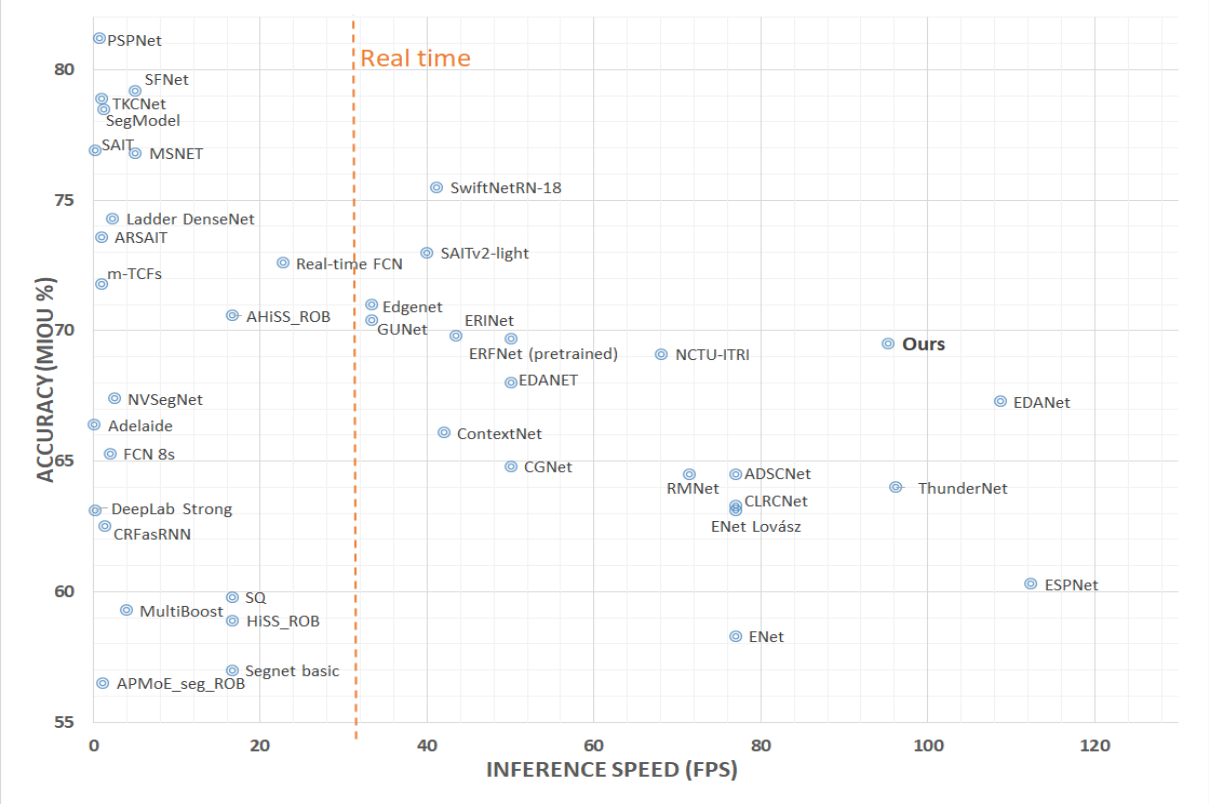


Figure 5. 15: La durée d'inférence et la précision (MIoU) sur la base de données Cityscapes [Cordts2016].Parmi les réseaux inclus nous trouvons SegNet [Badrinarayanan2015], FCN [Long2015], ENet [Paszke2016], ContextNet [Poudel2018], ERFNet [Romera2017], PSPNet [Zhao2017], ICNet [Zhao2018], EDANet [Lo2018] et notre architecture.

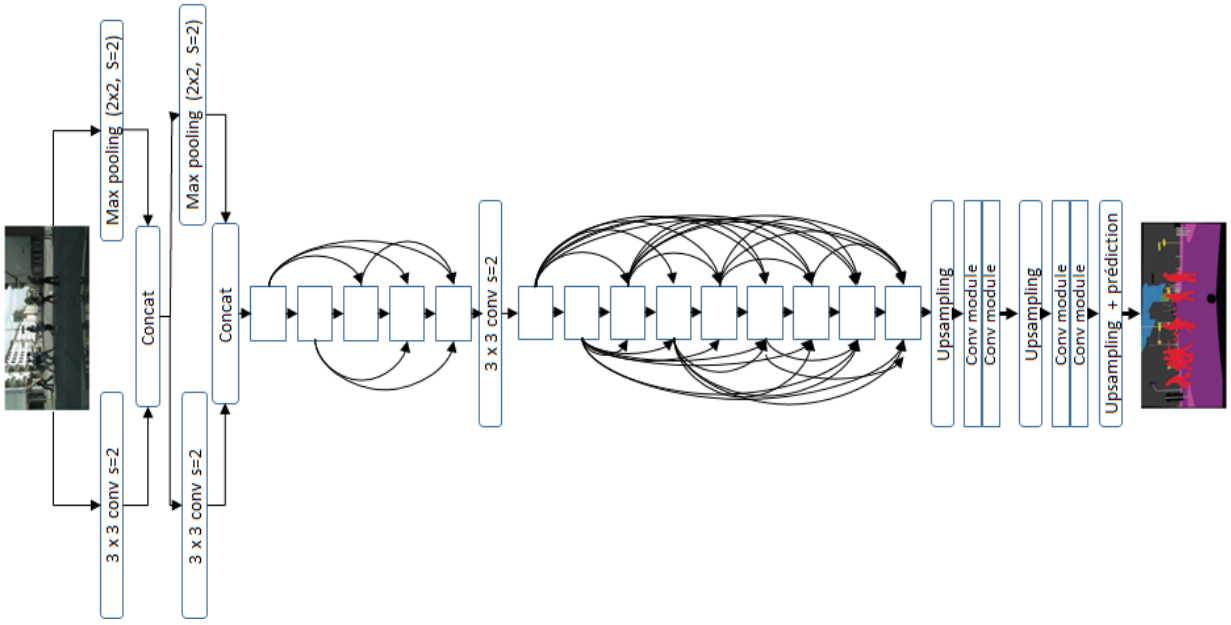


Figure 5. 16: L'architecture globale de notre réseau convolutif.

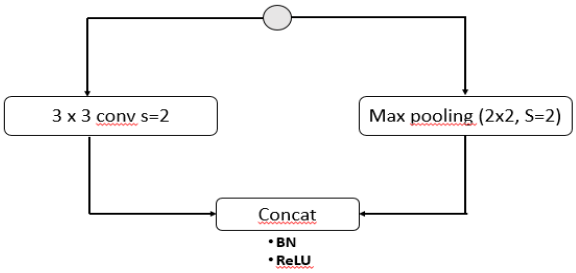


Figure 5. 17: Le bloc de l'architecture ENet.

Le sous-échantillonnage permet au réseau de neurones convolutif d'avoir un champ réceptif plus large, contribuant ainsi à une plus grande collecte d'informations contextuelles. Mais la réduction de la résolution d'image engendrée entraine la perte des détails spatiaux. Alors que la majorité des architectures élaborent des techniques de sous échantillonnage pour réduire la résolution d'image avec un rapport de 1/32 [Badri-narayanan2015], ils consolident le champ réceptif en dépit de la perte d'information spatiale. Pour résoudre ce problème, nous utilisons une structure équilibrée qui ne réalise que trois opérations de sous échantillonnage afin d'aboutir à une réduction de résolution d'un rapport de 1/8. Et pour atteindre efficacement l'élargissement du champ réceptif, nous avons utilisé la convolution dilatée (Voir tableau 5.1).

5.5.2 Module proposé

Le module proposé est le cœur de l'ensemble du notre réseau convolutif. Il se base sur l'agrégation des plus efficaces dernières techniques pour garantir un traitement en temps réel (Figure 5.18). Tout d'abord, les cartes de convolution (feature maps) de grande dimension sont projetées par une convolution ponctuelle 1x1 dans un espace de faible dimension avec une profondeur k. Par la suite une couche de convolution séparable profonde « depthwise separable convolution » 3x3 ré-échantillonne les cartes et les envoie à deux autres convolutions semblables aux deux précédentes mais en ordre inverse. Une connexion parallèle (skip-connection) est ajoutée pour concaténer l'entrée et la sortie de chaque bloc et améliorer ainsi le flux d'informations. Une fois la concaténation est effectuée nous procédons à un repositionnement aléatoire des différents canaux. Notre module préserve un champ de réception important et réduit considérablement le nombre de paramètres et la mémoire requise. Nous citons ci-dessous toutes ces composantes. Pour plus de détails voir table 5.1.

- **Convolution ponctuelle** (Point-wise convolution) Voir section 5.5.2;
- **Convolution profonde** (depth-wise convolution) Voir section 5.5.2;

- **Convolution dilatée (**dilated Convolution**)** Voir section 5.5.3**;**
- **Shuffling de canaux (**Shuffled convolution**)** Voir section 5.5.6**.**

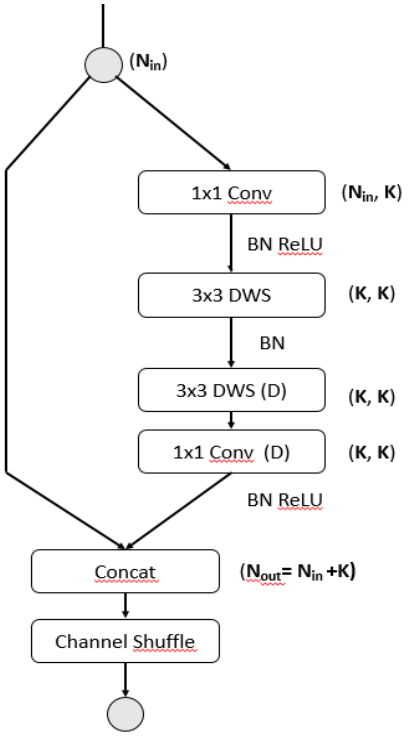


Figure 5. 18: La structure du module proposé, "(D)": est une convolution dilatée. "BN": est la batch normalisation.

- **Dense connectivité**

Pour améliorer d'avantage le temps de calcul nous avons utilisé la technique de la dense connectivité. Cette technique a été proposée par DenseNet [Gao2017], dans laquelle chaque couche prend en entrée toutes les cartes de caractéristiques précédentes. Nous adoptons cette stratégie dans notre réseau, En effet, chaque module est responsable de générer une profondeur limitée (growth rate k) Puis, il procède à une concaténation entre les caractéristiques appris et son entrée. De ce fait chaque sortie est constituée des toutes les cartes de convolution précédentes. Pour former le résultat final :

Soit y_m est la sortie du module m, H_m est sa fonction composite est :

$$y_m = [H_m(y_{m-1}), y_{m-1}] \qquad (5.4)$$

Cette technique peut considérablement augmenter l'efficacité du traitement, car chaque module n'est responsable que de l'acquisition de quelques nouvelles caractéristiques. De plus, il est bien connu que les couches les plus profondes ont des champs récepteurs plus grands [Simonyan2015]. Par exemple, un empilement de deux couches de convolution 3 × 3 a un

champ réceptif effectif qu'une seule couche de convolution 5 × 5. Ainsi, dense connectivité concaténant les caractéristiques apprises de chaque module possédant un champ de réception différent, elle permet à notre réseau de rassembler naturellement des informations multi-échelles. Et par la suite d'obtenir de bons résultats de segmentation sémantique à un faible coût de calcul.

Tableau 5. 1: Détail des couches de l'architecture proposée.

	Layer	**Name**	**Output channels**	**Output resolution**
	1	Downsampler block	15	256×512
	2	Downsampler block	60	128×256
	3	Conv module	100	128×256
	4	Conv module	140	128×256
	5	Conv module	180	128×256
	6	Conv module (dilated 2)	220	128×256
	7	Conv module (dilated 2)	260	128×256
	8	Downsampler block	130	64×128
	9	Conv module (dilated 2)	170	64×128
	10	Conv module (dilated 2)	210	64×128
	11	Conv module (dilated 4)	250	64×128
	12	Conv module (dilated 8)	290	64×128
	13	Conv module (dilated 16)	330	64×128
	14	Conv module (dilated 2)	370	64×128
	15	Conv module (dilated 4)	410	64×128
	16	Conv module (dilated 8)	450	64×128
	17	Conv module (dilated 16)	490	64×128
	18	Transposed convolution	60	128×256
	19-20	2 x Conv module	60	128×256
	21	Transposed convolution	16	256x512
	22-23	2 x Conv module	16	256x512
	24	Transposed convolution	Nombre de classes	512×1024

5.5.3 Expérimentation

Nous évaluons notre méthode sur l'ensemble de données de Cityscapes [Cordts2016]. Dans cette section, nous décrivons d'abord la base de données et l'étape d'entrainement de notre réseau. Ensuite, nous menons une série d'expériences pour examiner le réseau proposé. Enfin, nous effectuons quelques comparaisons avec d'autres architectures récentes. Nous avons utilisé

la bibliothèque Tensorflow avec un GPU de nvidia GTX 1060 6GB DDR5 ayant les caractéristiques suivantes : 1280 cores, une fréquence de 1708 Mhz et une vitesse mémoire de 8Gbps.

5.5.3.1 Base de données et métriques d'évaluation

- **Cityscapes [Cordts2016]**

La base de données Cityscapes est un ensemble de données de scènes de rues urbaines contenant 19 classes d'objets. Elle est composée de 5000 images avec une résolution élevée de 1024 × 2048, qui sont divisées en trois ensembles: 2975 images pour l'entrainement, 500 images pour la validation et 1525 images pour les tests. Nos expériences sont menées sur des images avec une résolution de 512x1024 sous-échantillonnées.

- **Métriques d'évaluation**

Nous avons utilisé Mean Intersection over Union (IoU) comme métrique de précision. Et pour comparer l'efficience, nous avons utilisé deux indicateurs FLOPs et le temps d'inférence.

  - L'intersection sur l'union est souvent utilisée pour comparer des méthodes de segmentation sémantique à plusieurs classes [Everingham2014]. La mesure IoU est définie comme suit :

$$IoU = \frac{TP}{TP + FP + FN}$$

    TP, FP et FN sont les nombres de pixels vrais positifs, faux positifs et faux négatifs, respectivement, d'une classe sur l'ensemble de données d'évaluation. Contrairement à la mesure de précision globale, qui calcule simplement les pixels correctement classés, cette métrique pénalise les pixels faux positifs et faux négatifs.
  - FLOPs est l'opération en virgule flottante par seconde. Le calcul de nombres en virgule flottante est souvent requis dans les applications scientifiques ou de traitement en temps réel. FLOPs Mesure la vitesse d'un processeur ou d'une des unités de calcul arithmétique d'un processeur. Un préfixe indique toujours une quantité : mégaflops (10^6 flops), gigaflops (10^9 flops), téraflops (10^{12} flops) ... Les puissances des processeurs actuels se mesurent en général en gigaflops.

5.5.3.2 Entrainement de notre réseau

Pour l'entrainement, nous avons utilisé la descente du gradient stochastique. Afin d'ajuster les paramètres via la rétro-propagation du gradient. Nous avons utilisé un taux d'apprentissage (learning rate) dynamique initialisé à 0,0005, et qui subit une réduction chaque 50 époques (epochs). Après différentes configurations, Chaque modèle a été entrainé pour au moins 300 époques. Nous avons utilisé l'augmentation des données avec la rotation et la translation aléatoires jusqu'à trois pixels.

5.5.3.3 Etude comparative

La structure de convolution séparable en profondeur « depthwise seprable convolution » et le concept de la dense connectivité [Gao2017] sont les deux éléments clés du module que nous avons proposé. Afin d'étudier plus en détail les améliorations potentielles, Nous avons comparé notre module de base à trois autres variantes B, C et D (figure 5. 19). Où dans la variante B, on a remplacé les convolutions ponctuelles de notre module de base par une forme factorisée de deux groupes (grouped convolution). La variante C, est similaire à B mais elle ne procède pas un réarrangement des canaux de sorties (channel shuffling). Si ces deux variantes ont le même coût, la variante C enregistre une diminution de précision par 2% par rapport à B. Tandis que B à un cout de calcul de 10% inférieur à notre module, avec une diminution de 3% en précision. Notre modèle nécessite 7,69 GFLOPs, qui est déjà un coût léger. Pour cela nous privilégions notre structure de base pour le gain de précision apporté. La variante D, présente la structure du module ERFnet [Romera2018], qui est une référence dans le domaine de segmentation en temps réel. Afin de pouvoir effectuer une comparaison au même coût de calcul, nous utilisons ces variantes dans l'architecture de notre réseau. Nous concevons le même placement des couches que notre réseau convolutif, nous construisons trois réseaux composés respectivement des trois variantes. Le tableau 5.2 représente les résultats obtenus pour chaque variante.

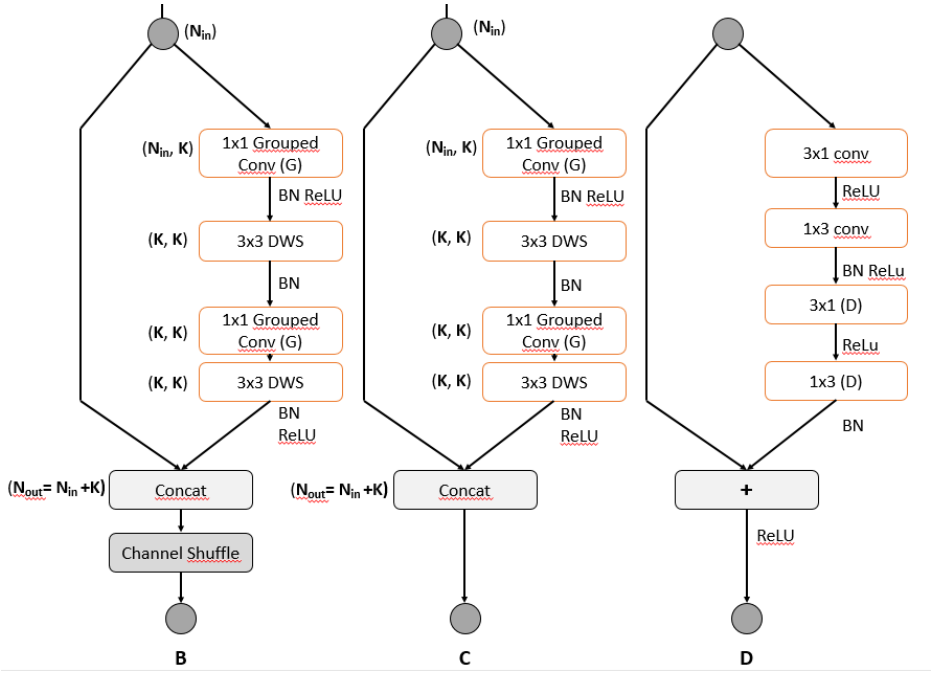


Figure 5.19: Différentes variantes B) l'utilisation de la convolution groupée. (C) variante b sans la convolution « Shuffled ». (D) le module ERFnet.

Tableau 5. 2: Résultats de l'étude d'ablation

Modules	mIoU	Params	Gflops
Notre module	69,56	0,47 M	7,69
B	67,42	0,78 M	6,88
C	65,9	0,78 M	6,88
D (ERF module)	65,11	0,81 M	11,41

Nous avons procédé aussi au remplacement du décodeur de notre architecture par l'interpolation bilinéaire, qui est une méthode largement utilisée pour l'obtention des architectures légères. Ceci à améliorer le coût de calcul par 34%, mais a dégradé la précision par 4,6%.

Afin d'évaluer la rapidité et la robustesse de notre architecture nous avons établi une étude comparative avec les modèles suivants [Chollet2017] [Zhang2017] [Sandler2018]. En termes

d'efficacité de mIoU et d'inférence sur l'ensemble de tests Cityscapes. Notre réseau atteint 69,5% mIoU, ce qui est supérieur à la plupart des méthodes existantes qui fonctionnent en temps réel (supérieur à 30 FPS), telles que ENet [Paszke2016] et ESPNet [Mehta2018], et surpasse même de nombreuses approches à plus basse vitesse, comme Dilation10 [Yu2016], FCN [Long2015]. Le réseau proposé nous garantit une grande précision de segmentation tout en conservant une efficacité maximale. Certains résultats visuels sont présentés à la f*i*gure 5.20.

Tableau 5. 3: Résultats des évaluations sur l'ensemble de test Cityscapes

Methodes	Extra data	Sub	mIoU (%)	Time (ms)	Speed (FPS)	Params
SegNet [Badrinarayanan2015]	ImN	4	57	37,8	26,4	29,5M
ENet [Paszke2016]	No	2	58,3	13	76,9	0,36M
ESPNet [Mehta2018]	No	2	60,3	8,9	112,9	0,36M
FCN-8s [Long2015]	ImN	no	65,3	0,5	2	134,5M
ContextNet [Poudel2018]	No	no	66,1	55	18,3	0,85M
Dilation10 [Yu2016]	ImN	no	67,1	4	0,25	140,8M
EDANet [Lo2018]	No	2	67,3	12,3	81,3	0,68 M
ERFNet [Romera2017]	No	2	68	24	41,7	2,1M
ICNet [Zhao2018]	No	no	69,5	21	47,9	-
PSPNeT [Zhao2017]	No	no	81,2	1288	0,78	-
Notre architecture	No	2	69,5	10,5	95,2	0,47 M

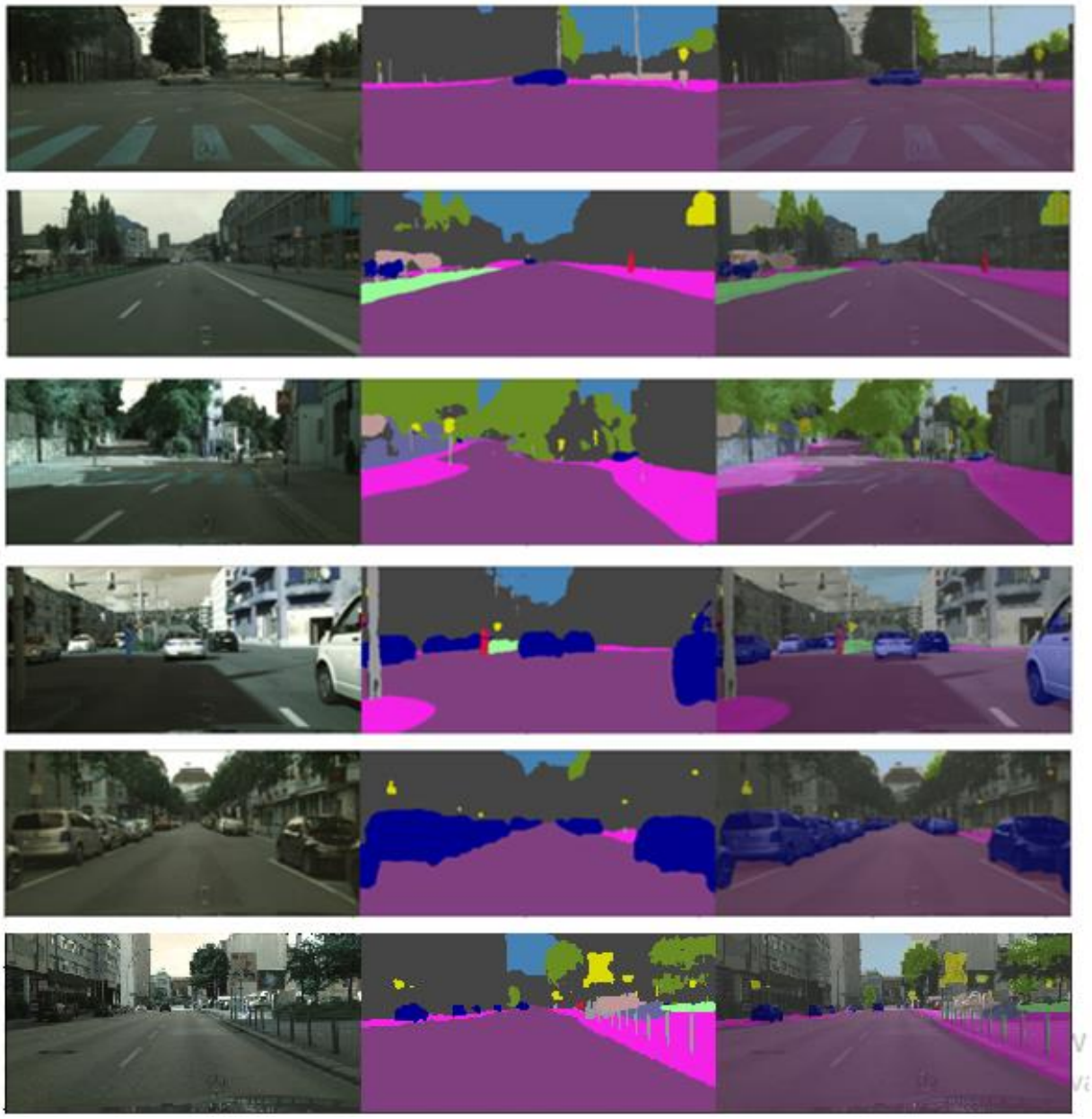


Figure 5. 20: Exemples des résultats sur la base de données Cityscapes. De gauche à droite: (a) entrée, (b) Notre résultat, (c) La fusion entre l'entrée et le résultat obtenu.

5.6 Conclusion

Comparés à d'autres algorithmes de segmentation d'images, les CNNs utilisent relativement peu de prétraitement. En d'autres termes le réseau est le seul responsable du développement de ses propres filtres. L'absence de paramétrage initial et d'intervention humaine est un atout majeur de ces types de réseaux. Parmi les avantages des réseaux convolutifs par rapport au MLP, nous trouvons l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un

même noyau de convolution. Il est important de remarquer que les architectures proposées dans la littérature ont une forte tendance à devenir de plus en plus profonde avec les années. Autrement dit, plus le réseau est profond, plus les performances sont bonnes. Néanmoins, cette profondeur implique de faire face à certaines difficultés notamment en termes de temps de calcul. C'est pour cela que dans ce chapitre, nous avons proposé une architecture du réseau de segmentation sémantique en temps réel, notre réseau est basé sur des techniques efficaces à savoir denses connectivité et les convolutions séparables en profondeur. Les résultats expérimentaux démontrent sa capacité à offrir un bon compromis entre la vitesse et la fiabilité et produire des résultats de segmentation assez précis avec un coût de calcul plus faible comparé aux autres systèmes dans l'état de l'art.

Conclusion et perspectives

Comme conclusion de ce manuscrit, nous avons pu accomplir la réalisation de nos objectifs, concernant les problématiques associées à la reconnaissance, la détection et la classification des objets 3D. Nos objectifs se sont portés sur le développement de nouvelles approches permettant d'interroger une bibliothèque de modèles 3D à partir de requête en deux, ou en trois dimensions, ainsi que d'effectuer une segmentation sémantique des scènes 2D contenant plusieurs objets 3D.

Trois domaines de traitement d'objet ont été étudiés durant cette thèse: l'indexation d'objet 2D/3D autrement dit la recherche d'objets 3D par des requêtes 2D, l'indexation d'objets 3D/3D; où, grâce à ces deux approches les modèles 3D peuvent être recherchés et récupérés de manière efficace. Et finalement, la troisième approche qu'est la segmentation sémantique d'une scène 2D en temps réel. Pour ces trois approches, nous rappelons en résumé nos contributions. Et nous finalisons ce rapport par la description de quelques perspectives.

Résumé des contributions

Le contenu des travaux de notre thèse peut être divisé en deux parties principales:

- Partie I: Reconnaissance et classification des modèles 3D (chapitres 3 et 4): dans cette partie, deux approches de classification ont été introduites en se basant sur l'indexation 2D/3D et 3D/3D et leurs performances sur des modèles de base de données d'objets 3D ont été examinées.
- Partie II: Segmentation sémantique en temps réel (chapitres 5): la deuxième partie de nos travaux était consacrée à l'introduction d'une nouvelle architecture de segmentation d'une scène 2D sémantiquement significatives et en temps réel.

Les techniques proposées pour la récupération de modèle 3D et la segmentation sémantique peuvent être détaillées comme suit:

- ***Classification des objets 3D en se basant sur $P et l'indexation 2D/3D***

La recherche et la classification d'objets 3D par des requêtes 2D a été abordée d'un point de vue novateur. Nous avons proposé une méthode de reconnaissance d'objet 3D en se basant sur l'indexation 2D/3D et l'algorithme de reconnaissance de nuages de points $P. Ce type

d'indexation caractérise la forme à partir de plusieurs projections 2D. Pour atteindre cet objectif, nous avons commencé par extraire les vues orthogonales de chaque objet 3D. Ensuite, nous avons extrait les caractéristiques en se basant sur le rendu spectral, afin d'étiqueter chacune de nos objets 3D. Enfin, nous nous sommes intéressés à la classification et la reconnaissance de ces objets à l'aide du classificateur $P. Ce procédé a été utilisé pour faire de la mise en correspondance entre une vue requête et un objet 3D et pour faire de la recherche dans une base d'objets 3D à partir d'une requête 2D. Cette approche a montré son fort potentiel. Nous avons également développé une interface graphique qui permet à l'utilisateur de proposer facilement une requête sous la forme d'une requête 2D ou même d'objet 3D et d'afficher par la suite les résultats de la reconnaissance. Les résultats de la simulation sont satisfaisants et démontrent une bonne performance de l'approche proposée, avec un taux supérieur à 91,5%. En outre, pour évaluer l'efficacité de notre méthode, nous avons comparé notre système avec d'autre classifieurs. Les résultats de la comparaison ont également mis en évidence la performance de l'algorithme $P.

- ***Classification des objets 3D en se basant sur la distribution de forme 2D***

Le principe consiste à caractériser la forme des objets 3D de manière compacte pour en déduire une signature. Nous avons présenté une nouvelle méthode de reconnaissance et de classification d'objets 3D polygonaux, en fonction de leurs géométries globales. Le but était d'extraire directement de la géométrie de l'objet 3D les informations les plus pertinentes, afin de mesurer la similarité entre les objets 3D. Pour ce faire, nous avons combiné la distribution de formes D2 et les réseaux de neurones artificiels. Le défi était de trouver une signature de forme qui peut être construite et classée rapidement, puis à apprendre au réseau de neurones artificiels plusieurs exemples de chaque classe, pour qu'à la fin de l'apprentissage, notre réseau peut reconnaître des objets non appris. La solution proposée a pu distinguer la forme générale de l'objet et semble plus adaptée à la recherche d'objets similaires dans des bases de données de formes multiples. Finalement, pour évaluer l'efficacité de notre méthode, nous avons comparé le classifieur à d'autres méthodes de classification. Les résultats de nos expériences sont encourageants et montrent la performance de l'approche proposée, avec un taux de reconnaissance supérieure à 91,7%.

- ***Segmentation sémantique en temps réel***

Nous avons conçu une architecture d'un réseau pour la segmentation sémantique en temps réel. L'objectif était de trouver un bon compromis entre le temps d'inférence et la précision. Pour

cela, nous avons proposé un réseau convolutif basé sur les techniques les plus efficaces à savoir : denses connectivité, les convolutions dilatées et les convolutions séparables en profondeur. Les résultats obtenus lors de la phase de test confirment la capacité de notre réseau. Nous avons pu aboutir à une précision élevée avec un coût de calcul plutôt faible comparé aux autres systèmes de l'état de l'art. Notre réseau atteint 69,5 MIoU sur la base Cityscapes avec une vitesse de 95,2 FPS.

Perspectives

Au cours de cette thèse, nous avons développé et mis au point de nouvelles techniques que ce soit pour l'indexation et la reconnaissance d'objets 3D ou pour la segmentation sémantique en temps réel. Étant donné que nos diverses contributions ont été guidées par des choix, souvent basés sur des contraintes, mais parfois arbitraires, des perspectives diverses et variées s'offrent à nous.

- Tout d'abord, nous prévoyons de tester nos méthodes de reconnaissance d'objet 3D sur d'autres bases de données et de prendre en compte davantage classes pour vérifier si l'approche est générique.
- Nous pensons à développer une méthode dynamique d'indexation 2D/3D, qui comparera directement l'image requête à l'objet 3D et s'adapte aux complexités de l'objet 3D et de la requête. Autrement dit, l'extraction et le choix du nombre de vues sera choisi dynamiquement, en fonction du besoin.
- D'après les travaux de [Ningning2018] il s'est avéré que deux architectures avec un Flops similaire peuvent avoir des vitesses différentes. Par conséquent, l'utilisation de cette métrique comme élément unique pour juger la complexité du calcul est insuffisante. Pour cela, nous visons d'intégrer en plus d'autres métriques qui tiennent en compte des phénomènes non prises par Flops tel que « memory access cost » (MAC) ou le degré de parallélisme.
- Nous sommes à la recherche d'une base de données contenant des scènes 2D et leurs objets 3D, afin de pouvoir extraire à partir de la scène, des informations pertinentes sur les objets 3D. Ces informations permettront par la suite une classification efficace de ces derniers. Une autre alternative aussi prometteuse est de trouver des scènes 2D issues de plusieurs capteurs, pour des informations plus complètes sur l'objet 3D.

Références

[Adit2016] Adit D., « A beginner's guide to understanding convolutional neural networks », URL:https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/. Cited on pages 7 and 8, 2016.

[Agarap2019] Agarap, A.F., «Deep Learning using Rectified Linear Units (ReLU) », arXiv:1803.08375v2 [cs.NE], 7 Feb 2019.

[Krizhevsky2012] Krizhevsky, A., Sutskever, I., Hinton, G., « Imagenet classification with deep convolutional neural networks », In NIPS, 2012.

[Amberg2007] Amberg, B., Andrew B., Andrew F., Sami R., Thomas V., « Reconstructing high quality face-surfaces using model based stereo », In Computer Vision, ICCV 2007. IEEE 11th International Conference on, pp. 1-8. 2007.

[Ankerst1999] Ankerst, M., Kastenmüller, G., Kriegel, H.P., Seidl, T., « 3D Shape Histograms for Similarity Search and Classification in Spatial Databases », In Advances in Spatial Databases, Springer, 1999.

[As'ari2014] As'ari, M.A., Sheikh, U.U., Supriyanto, E., « 3D shape descriptor for object recognition based on Kinect-like », Image and Vision Computing 32, pp: 260–269, 2014.

[Assfalg2003] Assfalg, J., Bimbo, A. D., and Pala, P., « Retrieval of 3D objects using curvature maps and weighted walkthroughs », 2003.

[Assfalg2006] Assfalg, J., Bimbo, A. D., and Pala, P., « Content-based retrieval of 3D models through curvature maps: a CBR approach exploiting media conversion », Multimedia Tools Appl, (1):29-50, 2006.

[Baeza-Yates2000] Baeza-Yates, R., Valiente, G., « An image similarity measure based on graph matching », Proceedings of the Seventh International Symposium on String Processing, Information Retrieval, 2000.

[Bagci2012] Bagci, U., Chen, X., Udupa, J.K., « Hierarchical scale-based multi-object recognition of 3D anatomical structures », IEEE Trans. Med. Imaging, (3):777-89, 2012.

Références

[Badrinarayanan2005] Badrinarayanan, V., Kendall, A., Cipolla, R., « SegNet: A deep convolutional encoder-decoder architecture for image segmentation », IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 39(12), 2481–2495, 2015.

[Beis1999] Beis, J.S., Lowe, D.G., « Indexing without invariants in 3D object recognition », IEEE Trans. Pattern, 1999.

[Biasotti2008] Biasotti, S., Giorgi, D., Spagnuolo, M. and Falcidieno, B., « Reeb graphs for shape an alysis and applications», Theoretical Computer Science, Volume 392, Issues 1–3, Pages 5-22, 28 February 2008.

[Biasotti2003] Biasotti, S., Marini, S., Mortara, M., Patanè, G., Spagnuolo, M., Falcidieno, B., « 3D shape mathing through topological structures », In: Nyström I., Sanniti di Baja G., Svensson S. (eds) Discrete Geometry for Computer Imagery. DGCI 2003. Lecture Notes in Computer Science, vol 2886. Springer, Berlin, Heidelberg, 2003.

[Bustos2004] Bustos, B., Keim, D. A., Saupe, D., Shrek, T. and Vrani, D., « An experimental comparison of feature-based 3D retrieval methods », In Second International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04), Thessaloniki, Greece, September 2004.

[Canterakis1999] Canterakis, N., « 3d zernike moments and zernike affine invariants for 3d image analysis and recognition », In 11th Scandinavian Conf. on Image Analysis, 1999.

[Caesar2016] Caesar, H., Uijlings, J., Ferrari, V., « Region-based semantic segmentation with end-to-end training », In: ECCV. 2016.

[Chaieb2014] Chaieb, F., Wieme, G., Ghorbel, F., « Indexation d'objets 3D par spectre de formes géodésiques 3D (SFG3D) », Traitement du Signal, Volume 31, n° 3-4, 2014.

[Chaouch2009] Chaouch Mohamed, « Recherche par le contenu d'objets 3D », Thèse de doctorat, Telecom ParisTech, 2009.

[Chaouch2006] Chaouch Mohamed et Verroust-Blondet Anne, « Enhanced 2d/3d approaches based on relevance index for 3d-shape retrieval », In Shape Modeling International'06, Matsushima, jun 2006.

[Chen2002] Chen Ding-Yun, Ming Ouhyoung: « A 3d model alignment and retrieval system», International Computer Symposium (ICS2002) Workshop on Multimedia Technology, oct 2002.

[Chen2003] Chen Ding-Yun, Xiao-Pei Tian, Yu-Te Shenet, Ming Ouhyoung, « On visual similarity based 3d model retrieval », In EUROGRAPHICS, Granada, Spain, sep 2003.

[chen2010] Chen, Q., Chen, X. and Wu, Y., « Optimization algorithm with kernel PCA to support vector machines for time series prediction», prediction. J. Comput., 5: 380-397, 2010.

[Chen2015] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., « Semantic image segmentation with deep convolutional nets and fully connected crfs», In ICLR, 2015.

[Chen2015] Chen, W., Wilson, J., Tyree, S., Weinberger, K., Chen, Y. « Compressing neural networks with the hashing trick », In: ICML, 2015.

[Chen2017] Chen, L.C., Papandreou, G., Schroff, F., Adam, H., « Rethinking atrous convolution for semantic image segmentation », CoRR, abs/1706.05587, 2017.

[Chen2018] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., « Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs », In TPAMI, IEEE Trans Pattern Anal Mach Intell, 40(4):834-848, 2018.

[Chollet2017] Chollet, F., « Xception: Deep learning with depthwise separable convolutions », IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[Clevert2016] Clevert, D.T., Hochreiter. S., « Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) », ICLR, 2016.

[Cortes1995] Cortes, C. et Vapnik, « V Support-Vector Networks », ML, 1995.

[Cordts2016] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., « The cityscapes dataset for semantic urban scene understanding », In in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[Courbariaux2016] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y., « Binarized neural networks: Training neural networks with weights and activations constrained to+ 1 or- 1. arXiv preprint arXiv:1602.02830 (2016).

[Dai2015] Dai, J., He, K., Sun, J., « Convolutional feature masking for joint object and stuff segmentation », In: CVPR, 2015.

[Damien2010] Damien P., Sami M, « La reconnaissance de Forme Comment améliorer les techniques de reconnaissance de forme 3D », 2010.

[Djeffal2012] Djeffal, A., « Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données », Thèse de doctorat, Université Mohamed Khider-Biskra, 2012.

[Domenach2017] Domenach, F., Portides, G., « Comparaison et Évaluation de Mesures de Similarité entre Concepts d'un Treillis », Revue des Nouvelles Technologies de l'Information, pp.303-308, 2017.

[Douass2017] Douass, S., AIT KBIR, M., « Overview on 3D models searching and indexing », BDCA'17 Proceedings of the 2nd international Conference on Big Data, Cloud and Applications Article No. 84, 2017.

Références

[Dugelay2008] Dugelay, Jean-Luc, Atilla Baskurt, and Mohamed Daoudi, eds, « 3D object processing: compression, indexing and watermarking », John Wiley & Sons, 2008.

[Dutagaci2005] Dutagaci, H. and Sankur, B., « Transform-based methods for indexing and retrieval of 3d objects», In The 5th International Conference on 3-D Digital Imaging and Modeling, Ottawa, Ontario, 2005.

[Elhoufi2018] Elhoufi, S., Jazouli, M., Majda, A. and Zarghili, A., « A 3-D classification based on $P Recogniser », International Journal of Computational Vision and Robotics, Vol. 8, No. 6, 2018.

[Elhoufi'2018] Elhoufi, S. Majda, A., Abbad, K., « D2 Shape distribution and artificial neural networks for 3D objects recognition » International Journal of Engineering & Technology, 7 (2.13), 103-108, 2018.

[Elkhal2014] Elkhal, M., Lakehal, A. and Satori, K., « A NEW METHOD FOR 3D SHAPE INDEXING AND RETRIEVAL IN LARGE DATABASE BY USING THE LEVEL CUT », Journal of Computer Science 10 (10): 1985-1993, 2014.

[Everingham2010] Everingham, M., Eslami, S.M.A., Luc V.G., Christopher K.I.W., Winn, J., Zisserman, A., « The pascal visual object classes (VOC) challenge », International Journal of Computer Vision, 88(2):303–338, 2010.

[Everingham2014] Everingham, M., Eslami, S.M.A., Luc V.G., Christopher K.I.W., Winn, J., Zisserman, A., « The pascal visual object classes challenge: A retrospective », International Journal of Computer Vision, 111(1):98–136, 2014.

[Faugeras1998] O. Faugeras and R. Keriven, « Complete dense stereovision using level set methods », In Proceedings of the European Conference on Computer Vision, pages 379–393. Springer, 1998.

[Filali-Ansary2006] Filali Ansary Tarik, «Model retrieval using 2d characteristic views», 2006.

[Filali-Ansary2007] Filali-Ansary Tarik, Daoudi Mohamed et Vandeborre Jean-Philippe, « A bayesian 3d search engine using adaptive views clustering. In IEEE Transactions On Multimedia», 2007.

[Frôhlich2013] Frôhlich, B., Rodner, E., Denzler, J., « Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach », In Computer Vision–ACCV 2012, pages 218–231. Springer, 2013.

[Fu2017] Fu, J., Liu, J., Wang, Y., Lu, H., « Stacked deconvolutional network for semantic segmentation », arXiv preprint arXiv:1708.04943, 2017.

[Gagvani1999] Gagvani, N. and Silver, D., «Parameter-controlled volume thinning», Graphical Models and Image Processing, Volume 61, Issue 3, Pages 149-164, 1999.

[Gao2017] Gao, H., Zhuang, L., Kilian, Q. W., « Densely connected convolutional networks », In CVPR, 2017.

[Girshick2014] Girshick, R., Donahue, J., Darrell, T., Malik, J., « Rich feature hierarchies for accurate object detection and semantic segmentation », In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 580–587, 2014.

[Graves2007] Graves, A., Fernandez, S., Schmidhuber, J., « Multi-dimensional recurrent neural networks », In: 17th International Conference on Artificial Neural Networks – ICANN, 2007.

[Hariharan2015] Hariharan, B., Arbelàez, P., Girshick, R., Malik, J., « Hypercolumns for object segmentation and fine-grained localization », In: CVPR, 2015.

[He2015] He, K., Zhang, X., Ren S., Sun, J. « Delving deep into rectifiers: Surpassing human-level performance on imagenet classification », ICCV, 2015.

[He2016] He, K., Zhang, X., Ren S., Sun, J. « *Deep Residual Learning for Image Recognition* », CVPR, 2016. (Cité en pages 23, 26 et 27).

[Heczko2002] Heczko, M., Keim, D. A., Saupe, D., Vrani, D. V., « Verfahren zur Ahnlichkeitssuche auf 3d-objecten: (methods for similarity search on 3d databases) »,), vol. 2. Datenbank-Spektrum, pp. 54–63, 2002.

[Hernandez2004] Carlos Hernandez, « Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion. Thèse de doctorat », Telecom ParisTech, 2004.

[Horn1984] Horn, B., « Extended gaussian images », Published in Proceedings of the IEEE, 1984.

[Horn1975] Horn, Berthold. K. P., « Obtaining shape from shading information », In Shape From Shading, pages 123–173, 1975.

[Howard2017] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., « Mobilenets: Efficient convolutional neural networks for mobile vision applications », arXiv preprint arXiv: 1704.04861, 2017.

[Huang2010] Huang, P., Hilton, A., Starck, J., « Shape similarity for 3D video sequences of people », Int. J. Comput, 89:362–381, 2010.

[Hubara2016] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y. « Quantized neural networks: Training neural networks with low precision weights and activations », 2016.

[Ion2008] Ion, A., Artner, N.M., Peyré, G., Marmol, S.B.L., Kropatsch, W.G., Cohen, L.D., « 3D shape matching by geodesic eccentricity », Proceeding of Computer Vision and Pattern Recognition Workshops, 2008.

Références

[Irani2003] Irani, P. and Ware, C., « Diagramming information structures using 3D per-ceptual primitives », ACM Transactions on Computer-Human Interaction (TOCHI), 10(1) PP:1-19, 2003.

[Jaderberg2014] Jaderberg, M., Vedaldi, A., Zisserman, A., « Speeding up convolutional neural networks with low rank expansions », BMVC, 2014.

[Jiantao2004] Jiantao, P., Guyu, L. Yi, Hongbin, X. Z., Weibin, L. and Uehara, Y., « 3d model retrieval based on 2d slice similarity measurements », 2004.

[Jin2O14] Jin, J., Dundar, A., Culurciello, E., « Flattened convolutional neural networks for feedforward acceleration », 2014.

[Johnson1999] Johnson, A.E., Hebert, M., « Using spin images for efficient object recognition in cluttered 3D scenes », Published in IEEE Trans. Pattern Anal. Mach. Intell, 1999.

[Jonpolygon2016] Jonpolygon, « Industries Utilizing 3D Models », Accessed 17 November 2016. https://jonpolygon.com/who-uses-3d-models/

[Journaux2010] Journaux, L., Destain, M.F., Cointault, F., Miteran, J., Piron A., « Plant Leaf Roughness Analysis by Texture Classification with Generalized Fourier Descriptors in different Dimensionality Reduction context », Joint International Agricultural Conference, Wageningen, Netherlands, 2010.

[Kang1993] Kang, S. B. and Ikeuchi, K., « The complex egi: A new representation for 3d pose determination », IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 15, Issue 7, July 1993.

[Karpathy2015] Karpathy, A., « CS231n: convolutional neural networks for visual recognition », http://cs231n.github.io/neural-networks-1/, accessed on 25-11-2015.

[Knerr1990] Knerr, S., Personnaz, L., Dreyfus, J., « Single-layer learning revisited: A stepwise procedure for building and training a neural network », Optimization Methods and Software, 1:23–34, 1990.

[Koen1992] Koen, D., Jan. J., Doorn, A., J. Van, « Surface shape and curvature scales », Image and Vision Computing, 10(8):557-564 ,October 1992.

[Laga2009] Laga, H., « 3D Shape Classification and Retrieval Using Heterogenous Features and Supervised Learning », ISBN 978-3-902613-56-1, pp. 450, I-Tech, Vienna, Austria, February 2009.

[Lavoue2012] Lavoue, G., « Combination of bag-of-words descriptors for robust partial shape retrieval », The Visual Computer 28, no. 9 Pp: 931-942, 2012.

[Levi2007] Levi, C.M., Conrado, R.R.Jr., Zhiyong, H., « A Shape Distribution for Comparing 3D Models », Springer-Verlag Berlin Heidelberg, LNCS 4351, Part I, pp. 54 – 63, 2007.

[Li2008] Li, L., Zhang, Y. and Zhao, Y.H., « k-Nearest Neighbors for automated classification of celestial objects », Sci. China Series G-Phys Mech. Astron, 51: 916-922. 2008.

[Li2014] Li, B., Yijuan, L., Afzal, G., Tobias, S., Benjamin Bustos, Alfredo Ferreira, Takahiko Furuya, « A comparison of methods for sketch-based 3D shape retrieval », Computer Vision and Image Understanding 119, 57-80. 168. 2014.

[Li2017] Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X., « Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade », In CVPR, 2017.

[Lin2017] Lin, G., Milan, A., Shen, C., Reid, I., « Refinenet: Multi-path refinement networks for highresolution semantic segmentation », In: CVPR, 2017.

[Ling2006] Ling, H., « An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison », IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 29, Issue 5, 2006.

[Liu2006] Liu, H., Song, D., Ruger, S., Hu, R., Uren, V., « Comparing Dissimilarity Measures for Content-Based Image Retrieval », 2006.

[Liu2012] Liu, Q., « A Survey of Recent View-based 3D Model Retrieval Methods », arXiv preprint arXiv:1208.3670, 2012.

[Liu2013] Liu, Zhen-Bao, Shu-Hui Bu, Kun Zhou, Shu-Ming Gao, Jun-Wei Han, and Jun Wu., « A survey on partial retrieval of 3D shapes », Journal of Computer Science and Technology 28, no. 5, 836-851, 2013.

[Lo2018] Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J., « Efficient dense modules of asymmetric convolution for real-time semantic segmentation », arXiv preprint arXiv: 1809.06323, 2018.

[Ioffe2015] Ioffe, S., Szegedy, C., « Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift », In Proceedings of the 32nd International Conference on Machine Learning (ICML), pages 448–456, 2015.

[Long2015] Long, J., Shelhamer, E., Darrell T., « Fully convolutional networks for semantic segmentation », In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431– 3440, 2015.

[Maas2013] Maas, A. L., Hannun, A. Y., Ng, A. Y., « Rectifier nonlinearities improve neural network acoustic models », ICML, (Cité en page 21), 2013.

[Mahmoudi2002] Mahmoudi, S., Daoudi, M., « 3d models retrieval by using characteristic views », 2002.

[Mamaleté2012] Mamalet, F., Garcia, C., « Simplifying ConvNets for Fast Learning », In International Conference on Artificial Neural Networks (ICANN), pages 58–65. Springer, 2012.

Références

[Mathieu2015] Mathieu Aubry, « Representing 3D models for alignment and recognition », 2015.

[Medioni2002] Medioni, G. and François, A., « 3D structures for generic object recognition», In 15th International Conference on Pattern Recognition, volume 1, pages 30-37, September 2000.

[Mehta2018] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi H., « ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation », In: ECCV, 2018.

[Ming2015] Ming, C., « A new Delaunay triangulation algorithm based on constrained maximum circumscribed circle », Wuhan University and Springer Verlag Berlin Heidelberg, Vol.20, No.4, PP: 313-317, 2015.

[Muralidharan2012] Muralidharan, R. and Chandrasekar, C., « 3D Object Recognition using Multiclass Support Vector Machine-K-Nearest Neighbor Supported by Local and Global Feature », J. Computer Sci., 8 (8): 1380-1388, 2012.

[Muselet2004] Muselet, D., Macaire, L., Bonnet, P., Postaire, J.G., « Object recognition by intersection between adapted color histograms », Traitement du Signal 6(6), 2004.

[Najjar2009] Najjar, A., Zagrouba, E., « Mesures non paramétriques de consistance géométrique des descripteurs locaux pour la recherche d'images par le contenu », 2009.

[Napoléon2010] Thibault Napoléon, « Indexation multi-vues et recherche d'objets 3D », Télécom ParisTech, 2010.

[Novotni2004] Novotni, M. and Klein, R., « Shape retrieval using 3d zernike descriptors », In Computer-Aided Design, 36(11), PP:1047-1062, September 2004.

[Ohbuchi2003] Ohbuchi, R., Nakazawa, M., and Takel, T., « Retrieval retrieving 3d shapes based on their appearance», In: Proceeding MIR '03 Proceeding of the 5th ACM SIGMM international workshop on multimedia information retrieval, Pages: 39-45, 2003.

[Olfa2005] Olfa Triki-Bchir, « Modélisation, reconstruction et animation de personnages virtuels 3D à partir de dessins manuels 2D », 2005.

[Osada2001] Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D., « Matching 3d models with shape distributions », In: Shape Matching International, pp. 154–166, 2001.

[Osada2002] Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D., « Shape Distributions », ACM Transactions on Graphics, Vol. 21, No. 4, October 2002.

[Paquet1997] Paquet, E. and Rioux, M., « A query by content software for three-dimensional models databases management », 1997.

[Paquet1999] Paquet, E. and Rioux, M., « The mpeg-7 standard and the content-based management of three-dimensional », June1999.

[Paquet2000] Paquet, E., Rioux, M., Murching, A., Naveen, T. and Tabatabai, A., « Description of shape information for 2-d and 3-d objects », Signal process Image Commun, 16(1), p: 103-22, 2000.

[Passalis2007] Passalis, G., Theoharis, T., Kakadiaris, I., A., « Ptk : A novel depth buffer-based shape descriptor for three-dimensional object retrieval », Visual Comput, 23:5–14, 2007.

[Paszke2016] Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., « Enet: A deep neural network architecture for real-time semantic segmentation », Computer Vision and Pattern Recognition, CoRR, abs/1606.02147, 2016.

[Poudel2018] Poudel, R.P.K., Bonde, U., Liwicki, S., Zach, C., « ContextNet: Exploring context and detail for semantic segmentation in real-time », In conference paper at British Machine Vision Conference (BMVC), 2018.

[Princeton2004] Princeton shape retrieval and analysis group: Princeton shape benchmark data-base. http://shape.cs.princeton.edu/benchmark/.

[Raluca2013] Raluca-Diana ŞAMBRA-PETRE, « Modelisation et inference 2D/3D de connaissances pour l'acces intelligent aux contenus visuels enrichis », juin,2013.

[Rastegari2016] Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A., « Xnor-net: Imagenet classification using binary convolutional neural networks », In: ECCV, 2016.

[Reeb1946] Reeb G., « Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique », Comptes-rendus de l'Academie des Sciences, 222:847-849, 1946.

[Richard2000] Richard, HR.H., Rahul, S., Misha, A.M., Rodney, J.D., Sebastian, H.,S., « Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit », Nature 405, 6789, 947, 2000.

[Riesenhuber2000] Riesenhuber Maximilian and Poggio Thomaso, « Models of object recognition », Nature neuroscience, November 2000.

[Roldan2012] Roldan, E., Juan M., Parrondo, R., « Entropy production and Kullback-Leibler divergence between stationary trajectories of discrete systems », Physical Review E 85, 031129, 2012.

[Romera2017] Romera, E., Alvarez, J. M., Bergasa, L. M., Arroyo, R., « Efficient convnet for real-time semantic segmentation », In IEEE Intelligent Vehicles Symposium, 1789-1794, 2017.

[Romera2018] Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R. « Erfnet: Efficient residual factorized convnet for real-time semantic segmentation », IEEE Transactions on Intelligent Transportation Systems, 19 (1), 263-272, 2018.

Références

[Ronneberger2015] Ronneberger, O., Fischer, P., Brox, T., « U-net: Convolutional networks for biomedical image segmentation », In: MICCAI, 2015.

[Rosenblatt1957] Rosenblatt, F., « The Perceptron: A Probabilistic Model for Information Storage and Organization In The Brain », 1957.

[Saupe2001] Saupe, D. and Vranic, D. V., « 3D Model Retrieval with Spherical Harmonics and Moments », In: Radig B., Florczyk S. (eds) Pattern Recognition. DAGM 2001. Lecture Notes in Computer Science, vol 2191. Springer, Berlin, Heidelberg, 2001.

[Seitz1997] Seitz Steven, Dyer Charles, « Photorealistic scene reconstruction by voxel coloring», In CVPR'97, Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, page 1067, 1997.

[Shaiek2013] Ayet Shaiek, « Reconnaissance d'objets 3D par points d'intérêt », 2013.

[Shinagawa1991] Shinagawa, Y., Kunii, T-L. and Kergosien, Y-L., « Surface coding based on morse theory », Computer Graphics and Applications, 1991.

[shen03] Shen Y-T, Chen D-Y, Tian X-P, Ouhyoung M., « 3D model search engine based on lightfield descriptors», In: Proc. Eurographics, 2003.

[Sheshina2016] Sheshina Evgeniya, « Designing and building a three-dimensional environment using blender 3D and unity game engine », 2016.

[Simonyan2015] Simonyan, K., Zisserman, A., « Very deep convolutional networks for large-scale image recognition », CoRR, abs/1409.1556, URL http: //arxiv.org/abs/1409.1556. Cited on pages 12 and 13, 2015.

[Smach2008] Smach, F, Lemaitre, C., Gauthier, J.P., Miteran, J., Atri, M., « Generalized Fourier Descriptors with Applications to Objects Recognition in SVM Context », Journal of Mathematical Imaging and Vision, 30 (1), pp. 43-71, Springer, 6 January 2008.

[Srivastava2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., « Dropout : a simple way to prevent neural networks from overfitting », The Journal of Machine Learning Research, 15(1) :1929–1958, 2014.

[Sullivan1998] Steve, S., Jean, P., « Automatic model construction, pose estimation and object recognition from photographs using triangular splines », IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:1091–1096, 1998.

[Sundar03] Sundar, H., Silver, D., Gagvani, N., Dickinson, S. and Silver, D., « Skeleton based shape matching and retrieval », IEEE, Shape Modeling International, 2003.

[Super2004] Super, B.J., « Learning chance probability functions for shape retrieval or classification », Proceedings of the IEEE Workshop on Learning in Computer Vision and Pattern Recognition, 2004.

[Szegedy2015] Szegedy, C., Liu, W., et al., « Going deeper with convolutions », Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–9, 2015.

[Szegedy[1]2016] Szegedy, C., Vanhoucke, V., et al., « Rethinking the Inception Architecture for Computer Vision », Computer Vision and Pattern Recognition, 2015.

[Szegedy[2]2016] Szegedy, C., Ioffe, S., Vanhoucke, V., « Inception-v4, inception-resnet and the impact of residual connections on learning », CoRR, 2016.

[Tangelder2008] Tangelder, Johan WH, and Remco C. Veltkamp, « A survey of content based 3D shape retrieval methods », Multimedia tools and applications 39, no. 3, 441-471, 2008.

[Tabbone2006] Tabbone, S.A., Wendling, L., Salmon, J-P., « A new shape descriptor defined on the Radon transform », in Computer Vision and Image Understanding, 102(1)(1):42-51, April 2006.

[Tangelder2004] Tangelder, J., Veltkamp, R., « A Survey of Content Based 3D Shape Retrieval Methods », Proceedings of the Shape Modeling International (SMI'04), pp. 145-156, Genova, Italy, 2004.

[Teli2014] Teli, S. P., and Biradar, S. « Effective Email Classification for Spam and Non-Spam», International Journal of Advanced Research in Computer Science and Software Engineering, 4(6), 273-278, 2014.

[Thibault2011] Thibault Napoléon, « Indexation multi-vues et recherche d'objets 3D », 2011.

[Thomas2003] Thomas Funkhouser and all, « A Search Engine for 3D Models », ACM Transactions on Graphics 22(1):83-105, January 2003.

[Tung2005] Tung, T. and Schmitt, F., « The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes », in International Journal of Shape Modeling, 11(1):91-120, 2005.

[Vapnik 1998] Vapnik, V.N., « Statistical Learning Theory », Edition Wiley, 1998.

[Veltkamp2001] Veltkamp, R.C., « Shape matching: similarity measure and algorithms », Proceedings Shape Modelling International, IEEE Press, 2001.

[Vranic 2001] Vranic, D.V., Saupe, D., « 3D shape descriptor based on 3D Fourier transform », EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, 2001.

[Vranic2004] Vranic, D. V., « 3d Model Retrieval ». Thèse de doctorat, Leipzig University, 2004.

Références

[Wang2007] Wang, D., Zhang, J., Wong, H.-S. and Li, Y., « 3d model retrieval based on multi-shell extended gaussian image », In: Advances in Visual Information Systems, VISUAL 2007. Lecture Notes in Computer Science, vol 4781, Springer, Berlin, Heidelberg, 2007.

[Watt2000] A. Watt, «3D computer graphics». Addison-Wesley, 2000.

[Wu2016] Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J., « Quantized convolutional neural networks for mobile devices », In: CVPR, 2016.

[Xiaolan2010] Xiaolan Li and Godil Afzal, « Investigating the bag-of-words method for 3D shape retrieval», EURASIP Journal on Advances in Signal Processing, 2010.

[Xie2017] Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., « Aggregated residual transformations for deep neural networks », in Computer Vision and Pattern Recognition (CVPR), 2017.

[Yang2017] Yang, J., Liu, Q., Zhang, K., « Stacked hourglass network for robust facial landmark localisation », In CVPR, 2017.

[Yu2016] Yu, F., Koltun, V., « Multi-scale context aggregation by dilated convolutions », CoRR, abs/1511.07122, Published as a conference paper at ICLR, 2016.

[Zaharia2001] Zaharia, T. and Prêteux, F., « Hough 3d transform-based 3d mesh retrieval », In: Proceedings SPIE Conference 4476 on Vision Geometry X, San Diego, 2001.

[Zaharia2002] Zaharia, T. and Prêteux F., « Indexation de maillages 3d par descripteurs de forme », Dans 13 ème Congrès Francophone de Reconnaissance de Forme et Intelligence Artificielle (RFIA ’02), volume 1, pages 48–57. Angers, France, 2002.

[Zaharia2002] Zaharia, T. et Preteux, F., « Shape-based retrieval of 3D mesh models », Dans IEEE International Conference on Multimedia and Expo (ICME ’02), volume 1, pages 437–440. Lausanne, Suisse. 2002.

[Zaharia2004] Zaharia, T., Preteux, F., « 3D versus 2D/3D Shape Descriptors: A Comparative study », In SPIE Conference On Image Processing: Algorithms and Systems, Vol. 2004, pp. 47-58, Toulouse, France, January 2004.

[Zahn72] Zahn, C. T., Roskies, R. Z., « Fourier Descriptors for Plane closed Curves », IEEE Transactions On Computers, Vol. 21, No. 3, pp. 269-281, 1972.

[Zhang2017] Zhang, X., Zhou, X., Lin, M., « *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices* », Available at: https://arxiv.org/pdf/1707.01083.pdf (Accessed: 5 January 2019), 2017.

[Zhao2017] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., « Pyramid scene parsing network », In: CVPR, 2017.

[Zhao2018] Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J., « Icnet for real-time semantic segmentation on high-resolution images », In ECCV, 2018.

[Zernike1934] Zernike, F. « Diffraction theory of the cut procedure and its improved form, the phase contrast method », Physica 1, 689–704, 1934.

[3DPrinting2016] 3D Printing.com. 2016. What is 3D printing? Accessed 07 November 2016. http://3dprinting.com/what-is-3d-printing/