ESIREM
ÉCOLE SUPÉRIEURE D'INGÉNIEURS
NUMÉRIQUE ET MATÉRIAUX

École associée Polytech

UNIVERSITÉ DE BOURGOGNE

Histogrammes d'orientation 3D

Rapport détaillé du projet

Enzo BOISSENIN, Thomas GIROUD, Firmin LAUNAY
ESIREM – Semestre 2

Juin 2022

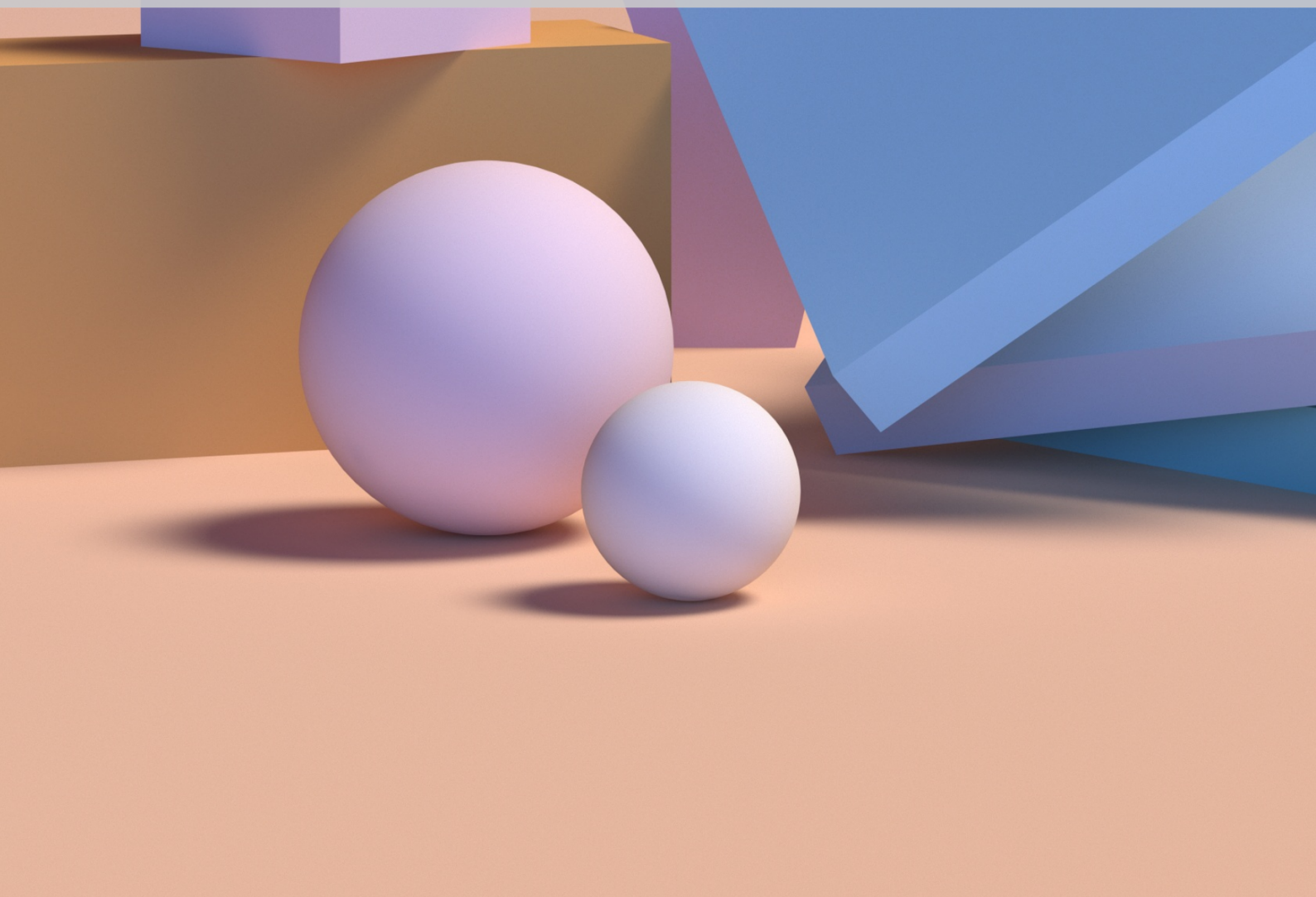


Table des matières

1

Introduction

Un histogramme est un outil fréquemment utilisé pour résumer et présenter lisiblement des données récoltées pendant une étude. Les résultats y sont généralement représentés sous forme de bâtons, et montrent rapidement au lecteur quels sont les résultats dominants de l'étude, ainsi que leur distribution globale.

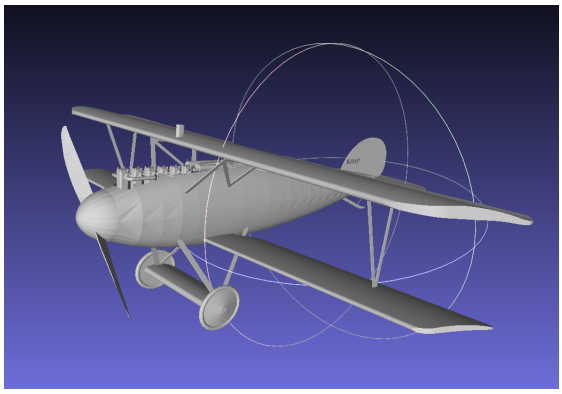
Dans notre cas, on parle d'*histogramme d'orientation 3D*, car les histogrammes que nous générons ont pour but de décrire la forme d'un solide – en trois dimensions donc – à partir de vecteurs qui définissent leur forme. C'est au départ le principe d'un *histogramme de gradient orienté* : des histogrammes d'orientation de gradients – définis par les différences de luminosité notamment – sont calculés sur toutes les zones de l'image, en suivant une grille. On obtient alors un gradient comme celui de la figure 1.1, où les contours du piéton apparaissent distinctement. Cela est par la suite utile pour reconnaître programmatiquement des objets (ou des piétons !) en calculant simplement leur gradient.

FIGURE 1.1 – Gradient d'un piéton

Source: *Milwaukee_(WIS)_N_5th_St_"_Tree,_Rain,_Wind_"_Pedestrian_1.jpg: vincent desjardinsderivative work: Sylenius, CC BY 2.0, via Wikimedia Commons*

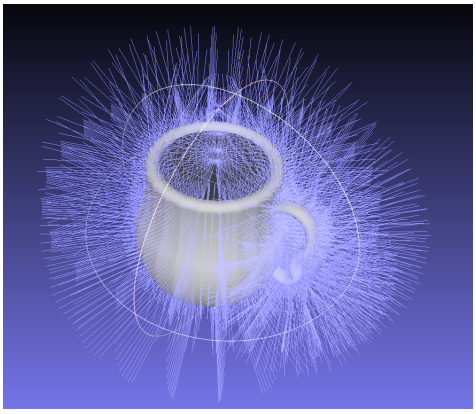
Notre but est de faire la même chose, mais avec des objets en trois dimensions, et non plus de simples images. C'est alors plus délicat, car l'introduction d'une troisième dimension dans l'orientation du vecteur – et donc d'une deuxième composante angulaire – rend plus difficiles les calculs et comparaisons sans traitement préalable. Nous allons donc rapporter nos modèles à des formes caractéristiques et régulières, en l'occurrence le tétraèdre, le cube, l'octaèdre, le dodécaèdre et l'icosaèdre. On doit alors comparer le vecteur normal de chaque face du modèle avec le vecteur normal de chaque face de la forme utilisée, à l'aide de produits vectoriels et de produits scalaires. Se rapporter à un nombre de faces et d'orientations limitées permet de discrétiser l'orientation de chaque face. On perd bien sûr en précision, puisqu'on ne se réfère plus qu'à une plage d'orientations possibles, mais il n'est pas envisageable d'effectuer des calculs exacts sur les milliers de faces de chaque échantillon.

Les objets que nous allons traiter sont sous la forme de nuages de points, issus de la base de données « ModelNet40 » de l'Université de Princeton. Nous disposons de quarante classes d'objets divers, composés d'échantillons allant d'un bol à un avion, que nous pouvons visualiser avec MeshLab, un logiciel de traitement de maillages 3D, duquel sont issues les visualisations de la figure 1.2. On cherche de notre côté à déterminer les normales de chaque face de chacun de ces modèles, comme celles représentées sur la figure 1.2b.



(a) Visualisation du nuage de points « airplane_0001 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Visualisation du nuage de points « cup_0001 », avec affichage des normales à chaque plan

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*

FIGURE 1.2 – Visualisations de différents modèles de la base de données avec MeshLab

Nous expliquerons tout d'abord en détail la manière dont nous avons calculé les histogrammes, en se rapportant aux polyèdres réguliers et en analysant chaque nuage de points, puis analyserons les résultats que nous pouvons obtenir à l'aide de différentes méthodes.

2

Calcul des histogrammes

Tout d'abord, afin de générer des données exploitables à partir des échantillons, il faut calculer leur histogramme en les rapportant à des polyèdres réguliers.

2.1 Présentation des polyèdres réguliers

Il existe cinq polyèdres réguliers convexes : le tétraèdre, l'hexaèdre (plus communément appelé le cube), l'octaèdre, le dodécaèdre et l'icosaèdre. Dans un premier temps, nous avons donc calculé les vecteurs normaux de chaque face de chacun d'entre eux.

2.1.1 Tétraèdre

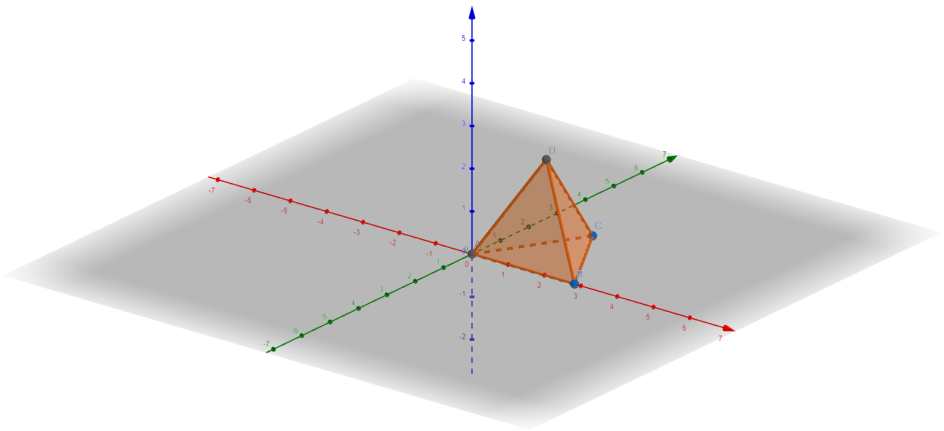


FIGURE 2.1 – Représentation d'un tétraèdre en trois dimensions

Source: *Figure réalisée par nos soins sur GeoGebra*

L'information qui nous intéresse sur les vecteurs normaux que nous allons obtenir est leur orientation. Leur norme nous importe peu. On peut alors sans aucun problème fixer des valeurs pour chaque point de chaque polyèdre, et cela simplifie grandement les calculs.

Dans le cas du tétraèdre, les coordonnées des points sont :

$$\text{A}\,(0;0;0) \qquad \text{B}\,(1;0;0) \qquad \text{C}\left(\frac{1}{2};\frac{\sqrt{3}}{2};0\right) \qquad \text{D}\left(\frac{1}{2};\frac{\sqrt{3}}{4};\frac{\sqrt{3}}{2}\right)$$

Pour chacune des faces du cube (qui correspondent donc mathématiquement à des plans, caractérisés par trois points), on peut obtenir deux vecteurs directeurs, desquels on déduit un vecteur normal au plan à l'aide d'un produit vectoriel :

— Plan ABC : $\vec{BA}\,(-1;0;0)$ $\vec{BC}\left(-\frac{1}{2};-\frac{\sqrt{3}}{2};0\right)$ $\Longrightarrow$ $\vec{N_1} = \vec{BA} \wedge \vec{BC} = \left(0;0;\frac{\sqrt{3}}{2}\right)$

— Plan ADB : $\vec{DA}\left(\frac{1}{2};\frac{\sqrt{3}}{2};\frac{\sqrt{3}}{2}\right)$ $\vec{DB}\left(-\frac{1}{2};-\frac{\sqrt{3}}{2};0\right)$ $\Longrightarrow$ $\vec{N_2} = \vec{DA} \wedge \vec{DB} = \left(0;\frac{\sqrt{3}}{2};-\sqrt{3}\right)$

— Plan ADC : $\vec{DA}\left(\frac{1}{2};\frac{\sqrt{3}}{2};\frac{\sqrt{3}}{2}\right)$ $\vec{DC}\left(0;0;-\frac{\sqrt{3}}{2}\right)$ $\Longrightarrow$ $\vec{N_2} = \vec{DA} \wedge \vec{DC} = \left(\frac{3}{4};\frac{\sqrt{3}}{4};0\right)$

— Plan BDC : $\vec{DB}\left(-\frac{1}{2};\frac{\sqrt{3}}{2};\frac{\sqrt{3}}{2}\right)$ $\vec{DC}\left(0;0;-\frac{\sqrt{3}}{2}\right)$ $\Longrightarrow$ $\vec{N_2} = \vec{DB} \wedge \vec{DC} = \left(\frac{3}{4};-\frac{\sqrt{3}}{4};0\right)$

2.1.2 Hexaèdre

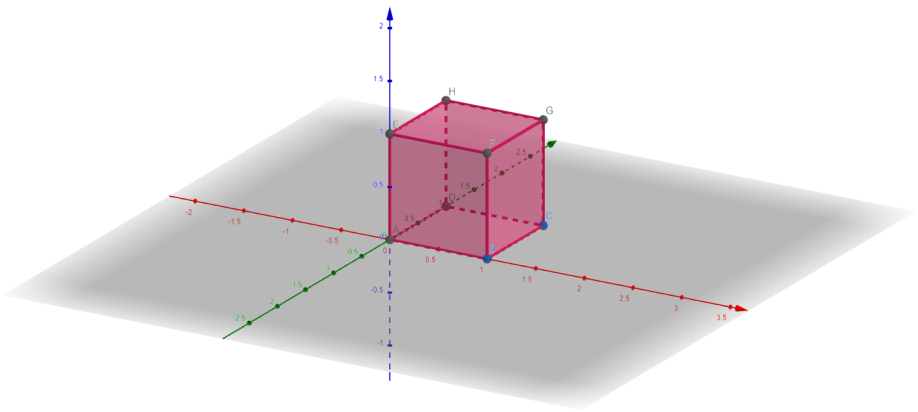


FIGURE 2.2 – Représentation d'un cube en trois dimensions

Source: *Figure réalisée par nos soins sur GeoGebra*

Coordonnées des points :

$$\text{A}\,(0;0;0) \qquad \text{B}\,(1;0;0) \qquad \text{C}\,(1;1;0) \qquad \text{D}\,(1;0;0)$$

$$\text{E}\,(0;0;1) \qquad \text{F}\,(1;0;1) \qquad \text{G}\,(1;1;1) \qquad \text{H}\,(0;1;1)$$

— Plan ABEF : $\vec{AB}\,(1;0;0)$ $\vec{AE}\,(0;0;1)$ $\Longrightarrow$ $\vec{N_1} = \vec{AB} \wedge \vec{AE} = (0;-1;0)$

— Plan ABCD : $\vec{AB}\,(1;0;0)$ $\vec{AD}\,(0;1;0)$ $\Longrightarrow$ $\vec{N_2} = \vec{AD} \wedge \vec{AB} = (0;0;-1)$

— Plan FGCB : $\vec{BF}\,(0;0;1)$ $\vec{BC}\,(1;0;0)$ $\Longrightarrow$ $\vec{N_3} = \vec{BF} \wedge \vec{BC} = (1;0;0)$

— Plan GHDC : $\vec{CG}\,(0;0;1)$ $\vec{CD}\,(-1;0;0)$ $\Longrightarrow$ $\vec{N_4} = \vec{CD} \wedge \vec{CG} = (0;1;0)$

— Plan HDAE : $\vec{DH}\,(0;0;-1)$ $\vec{DA}\,(0;-1;0)$ $\Longrightarrow$ $\vec{N_5} = \vec{DH} \wedge \vec{DA} = (-1;0;0)$

— Plan HGFE : $\vec{HG}\,(1;0;0)$ $\vec{NE}\,(0;-1;0)$ $\Longrightarrow$ $\vec{N_6} = \vec{HG} \wedge \vec{NE} = (0;0;1)$

2.1.3 Octaèdre

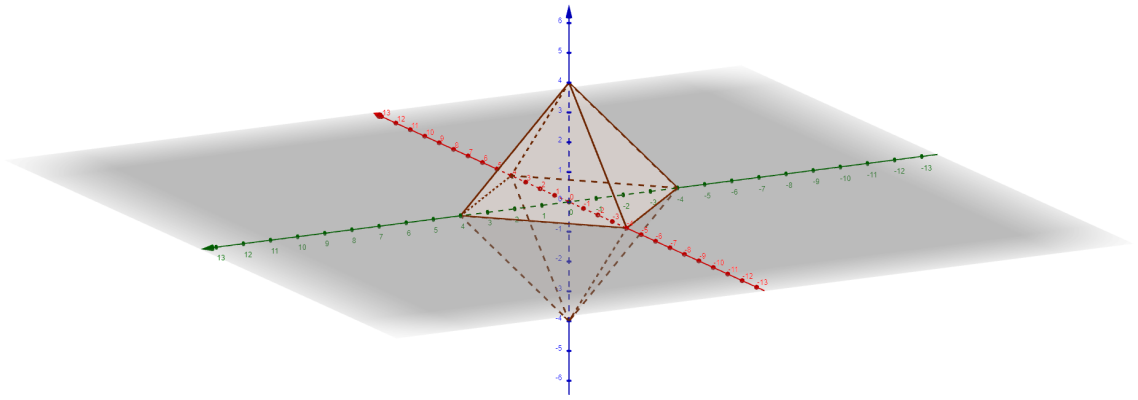


FIGURE 2.3 – Représentation d'un octaèdre en trois dimensions

Source: *Figure réalisée par « ogilson » sur GeoGebra*

- Plan AEC : $\vec{N_1}\left(\frac{1}{\sqrt{2}};0;\frac{1}{2}\right)$
- Plan AED : $\vec{N_2}\left(0;\frac{1}{\sqrt{2}};\frac{1}{2}\right)$
- Plan DEB : $\vec{N_3}\left(-\frac{1}{\sqrt{2}};0;\frac{1}{2}\right)$
- Plan BEC : $\vec{N_4}\left(0;-\frac{1}{\sqrt{2}};\frac{1}{2}\right)$
- Plan CAF : $\vec{N_5}\left(\frac{1}{\sqrt{2}};0;-\frac{1}{2}\right)$
- Plan ADF : $\vec{N_6}\left(0;\frac{1}{\sqrt{2}};-\frac{1}{2}\right)$
- Plan AED : $\vec{N_7}\left(-\frac{1}{\sqrt{2}};0;-\frac{1}{2}\right)$
- Plan DFC : $\vec{N_8}\left(0;-\frac{1}{\sqrt{2}};-\frac{1}{2}\right)$

2.1.4 Dodécaèdre

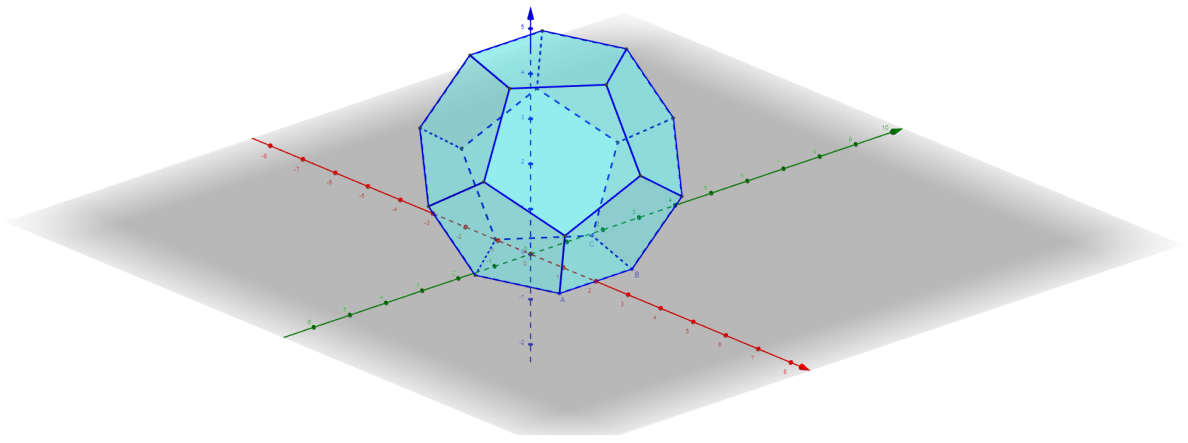


FIGURE 2.4 – Représentation d'un dodécaèdre en trois dimensions

Source: *Figure réalisée par « Debart Patrice » sur GeoGebra*

1	$\vec{N_1}\,(0;-1.23607;0.763932)$	7	$\vec{N_7}\,(-1.23607;0.763932;0)$
2	$\vec{N_2}\left(5.96046\times10^{-8};-1.23607;0.763932\right)$	8	$\vec{N_8}\,(-1.23607;-0.763932;0)$
3	$\vec{N_3}\,(0;-1.23607;-0.763932)$	9	$\vec{N_9}\left(-0.763932;5.96046\times10^{-8};-1.23607\right)$
4	$\vec{N_4}\,(0;-1.23607;-0.763932)$	10	$\vec{N_{10}}\left(1.23607;-0.763932;5.96046\times10^{-8}\right)$
5	$\vec{N_5}\,(-0.763932;0;1.23607)$	11	$\vec{N_{11}}\,(1.23607;0.763932;0)$
6	$\vec{N_6}\left(0.763932;-5.96046\times10^{-8};1.23607\right)$	12	$\vec{N_{12}}\,(0.763932;0;-1.23607)$

2.1.5 Icosaèdre

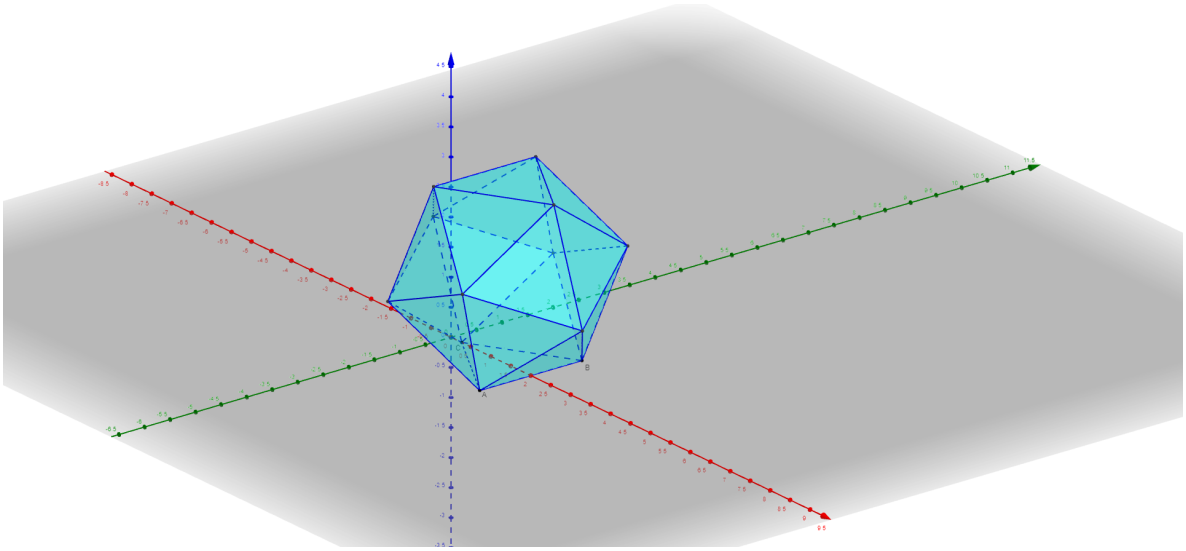


FIGURE 2.5 – Représentation d'un icosaèdre en trois dimensions

Source: *Figure réalisée par « Debart Patrice » sur GeoGebra*

1	$\vec{N_1}\,(0;1.23607;3.23607)$	11	$\vec{N_{11}}\,(3.23607;0;-1.23607)$
2	$\vec{N_2}\,(0;-1.23607;3.23607)$	12	$\vec{N_{12}}\,(2;-2;-2)$
3	$\vec{N_3}\,(3.23607;0;1.23607)$	13	$\vec{N_{13}}\,(2;2;-2)$
4	$\vec{N_4}\,(2;-2;2)$	14	$\vec{N_{14}}\,(1.23607;3.23607;0)$
5	$\vec{N_5}\,(2;2;2)$	15	$\vec{N_{15}}\,(-1.23607;3.23607;0)$
6	$\vec{N_6}\,(-2;2;2)$	16	$\vec{N_{16}}\,(-2;2;-2)$
7	$\vec{N_7}\,(-3.23607;0;1.23607)$	17	$\vec{N_{17}}\,(-3.23607;0;1.23607)$
8	$\vec{N_8}\,(-2;-2;2)$	18	$\vec{N_{18}}\,(-2;-2;-2)$
9	$\vec{N_9}\,(-1.23607;-3.23607;0)$	19	$\vec{N_{19}}\,(0;-1.23607;-3.23607)$
10	$\vec{N_{10}}\,(1.23607;-3.23607;0)$	20	$\vec{N_{20}}\,(0;1.23607;-3.23607)$

2.2 Analyse des fichiers de la base de données

Les calculs que nous effectuons peuvent être théoriquement effectués à la main pour des modèles extrêmement simples, mais la base de données à analyser ne comportant pas moins de 9843 échantillons plus complexes les uns que les autres, cela se serait vite avéré complexe et fastidieux !

Nous avons donc automatisé ce processus. Nous avons réalisé plusieurs programmes, aux buts distincts.

Remarque Les algorithmes réalisés sont disponibles sur un dépôt Git public, à cette adresse : github.com/Firmin-ESIREM/projet-histogrammes-3d

La première partie consiste donc à concevoir un programme qui analyse le nuage de points qu'on lui passe en argument.

En effet, les échantillons de la base de données sont au format OFF. Notre programme d'analyse doit alors comprendre la structure de ces fichiers, à l'instar de MeshLab, qui permet de les visualiser comme sur la figure 2.6.

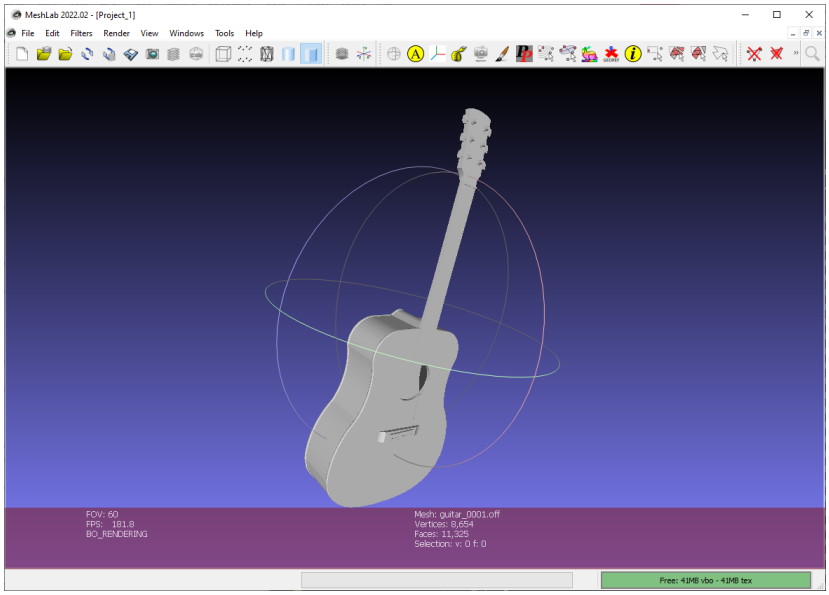


FIGURE 2.6 – Objet « guitar_0001.off » affiché par MeshLab

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*

2.2.1 Structure d'un fichier OFF

Un fichier OFF a une structure bien définie, rappelée en figure 2.7.

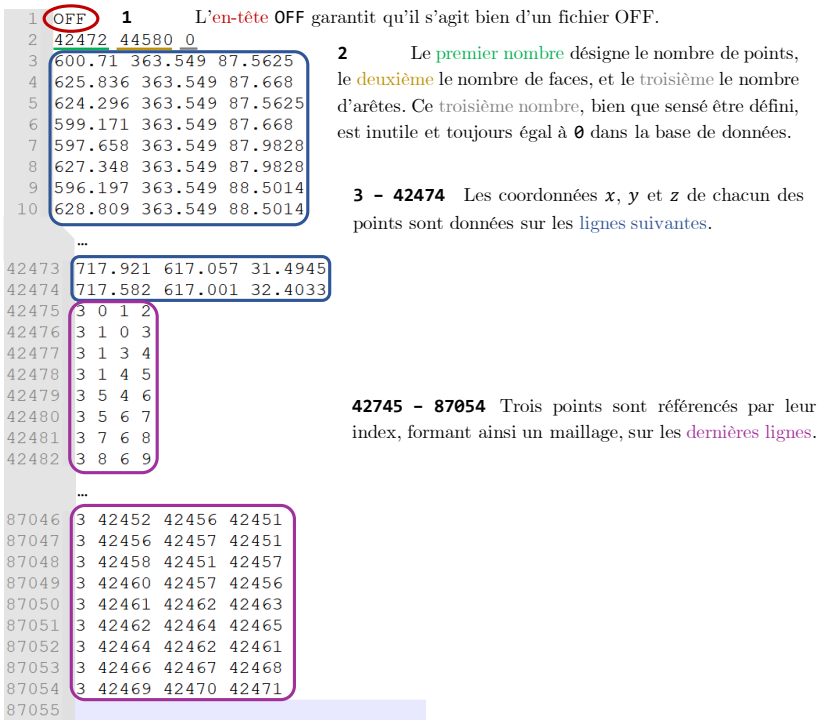


FIGURE 2.7 – Structure d'un fichier OFF

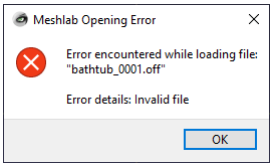
Source: *Documentation et base de données « ModelNet40 » de l'Université de Princeton / Infographie réalisée par nos soins*

Certains modèles ne sont pas formatés correctement et ne sont donc pas reconnus en l'état par notre programme. En effet, leur en-tête `OFF` est situé sur la même ligne que leurs propriétés. C'est par exemple le cas de tous les modèles de baignoires, comme celui de la figure 2.8a. Nous avons donc assoupli les conditions de notre programme pour qu'il puisse tout de même les traiter. Par ailleurs, ces fichiers lèvent aussi une exception quand ils sont ouverts dans *MeshLab*, comme montré en figure 2.8b.

```
1 OFF3514 3546 0
2 18.514300 -29.876500 -23.038200
3 18.514300 29.873400 -23.038200
4 18.501200 -30.031500 -23.288200
5 18.501200 29.975000 -23.288200
```

(a) En-tête du fichier « bathtub_0001.off »

Source: *Base de données « ModelNet40 » de l'Université de Princeton*



(b) Exception lancée par MeshLab à l'ouverture du fichier « bathtub_0001.off »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*

FIGURE 2.8 – Mauvais formatage déclenchant une exception

2.2.2 Implémentation

Nous avons choisi d'utiliser le langage C++ pour effectuer ceci, car il est rapide et performant. Cela nous permet d'optimiser le temps nécessaire à analyser tous les modèles de la base de données, certains étant assez volumineux. Par exemple, le modèle `curtain_0130.off`, qui représente une structure scénique avec des sortes de rideaux, pèse près de 170 MB et ne possède pas moins de 2 501 688 points et 4 548 836 faces !

Afin de simplifier le processus, nous avons choisi d'effectuer de la programmation orientée objets. Notre programme s'articule autour de quatre classes :

1. `Model` est la classe qui représente un objet en trois dimensions. Les modèles de la base de données, comme les formes auxquelles nous les comparons ont cette classe. Chaque modèle possède des points, des plans (faces), un barycentre et un nom.
2. `Point` représente un point, avec des coordonnées x, y et z.
3. `Plane` représente un plan (c'est-à-dire une face), composé de trois points desquels sont déduits deux vecteurs directeurs et un vecteur normal.
4. Enfin, `MathVect` représente un vecteur au sens mathématique, composé de trois coordonnées de déplacement x, y et z. Mais surtout, cette classe implémente les nombreuses opérations entre vecteurs (addition, soustraction, produit scalaire et produit vectoriel) à l'aide de définitions d'opérateurs.

2.3 Comparaison entre les nuages de points et les polyèdres

Une fois le nuage de points à analyser et les différents polyèdres instanciés, on peut les comparer et générer nos histogrammes.

La première étape est de calculer un vecteur normal pour chaque face de l'échantillon. En l'occurence, c'est le vecteur normal orienté vers l'extérieur que l'on souhaite obtenir. Pour ce faire, il faut donc tout d'abord calculer le barycentre de l'objet, en calculant la moyenne des coordonnées x, y et z de tous les points. Ensuite, on peut calculer un vecteur normal à chaque face en utilisant le produit vectoriel. Par exemple, pour calculer le vecteur normal d'un plan ABC :

$$\vec{N_{\mathrm{ABC}}} = \vec{\mathrm{AB}} \wedge \vec{\mathrm{AC}}$$

On prend ensuite un point du plan, que l'on translate de ce vecteur normal. Si la distance entre ce point et le barycentre se réduit, cela signifie qu'il faut prendre le vecteur opposé, sinon on garde le vecteur actuel.

Après avoir calculé toutes nos normales, il est temps de comparer les normales obtenues avec celles des polyèdres. Pour chacun d'entre eux, pour chacune des faces, on effectue la somme des produits scalaires positifs entre les vecteurs normaux, afin d'obtenir un « index de similarité » :

$$I_{\mathrm{poly}_1} = \sum_{k=1}^{\text{nb de faces}} \max(\vec{N_k} \cdot \vec{N_{\mathrm{poly}_1}}, 0)$$

Pour rendre la comparaison entre des objets de différentes tailles possible, il faut normaliser les valeurs obtenues en rapportant les valeurs des faces à 1 :

$$I_{\mathrm{poly}_1,\ \mathrm{normalisé}} = \frac{I_{\mathrm{poly}_1}}{\sum_k I_{\mathrm{poly}_k}}$$

2.4 Génération des histogrammes

Le programme génère alors, selon la méthode énoncée précédemment, un histogramme au format CSV comme celui de la figure 2.9, pour chacun des échantillons rapporté à chacun des polyèdres.

```
face_no;scal_sum;normalized_sum
Face 1;60660.3;0.091424
Face 2;77052;0.116129
Face 3;78038.9;0.117616
Face 4;61291.8;0.0923757
Face 5;198876;0.299736
Face 6;187587;0.28272

```

FIGURE 2.9 – Histogramme du modèle « airplane_0001.off » comparé à un cube, au format CSV

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Fichier généré par nos soins*

Nos histogrammes au format CSV sont très pratiques à analyser informatiquement, mais assez peu lisibles par un humain en l'état. Afin de pouvoir aisément générer des histogrammes graphiques à partir de leurs données, nous avons réalisé un script en R qui génère automatiquement un histogramme PDF du même type que celui de la figure 2.10 à partir du fichier CSV.

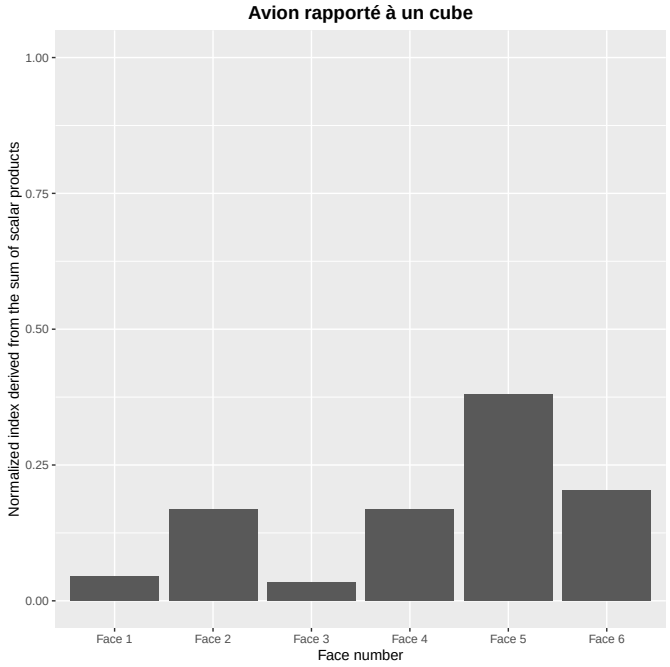


FIGURE 2.10 – Exemple d'histogramme généré par R, pour le modèle « airplane_0003.off »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

3

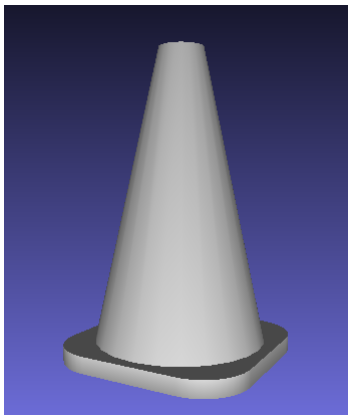
Analyse des résultats obtenus

3.1 Analyse des histogrammes obtenus

Nous pouvons analyser à l'aide d'outils tels que la matrice de corrélation les histogrammes que nous obtenons, afin de déterminer si on obtient bien un histogramme « caractéristique » de la classe de l'objet.

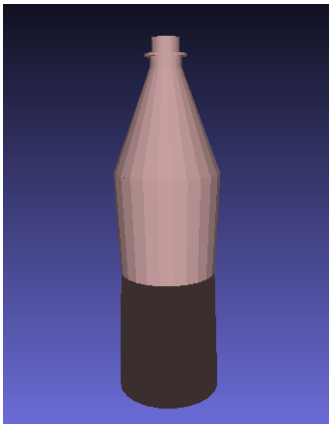
3.1.1 Échantillons rapportés à un tétraèdre

Nous allons prendre dans un premier temps un cône ainsi qu'une bouteille, visibles en figure 3.1, qui ont une forme générale similaire à celle d'un tétraèdre : le but est de montrer que l'on peut comparer deux objets qui se ressemblent visuellement pour voir si leurs histogrammes sont également similaires.



(a) Visualisation du nuage de points « cone_0001 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Visualisation du nuage de points « bottle_0001 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*

FIGURE 3.1 – Cône et bouteille visualisés sur MeshLab

À première vue, en regardant les fichiers CSV, en figure 3.2, les histogrammes du cône et de la bouteille semblent assez proches.

face_no	scal_sum	normalized_sum
Face 1	366.982	0.299894
Face 2	343.207	0.280465
Face 3	343.178	0.280442
Face 4	170.339	0.139199

(a) Histogramme CSV du nuage de points « cone_0001 » rapporté à un tétraèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Fichier généré par nos soins*

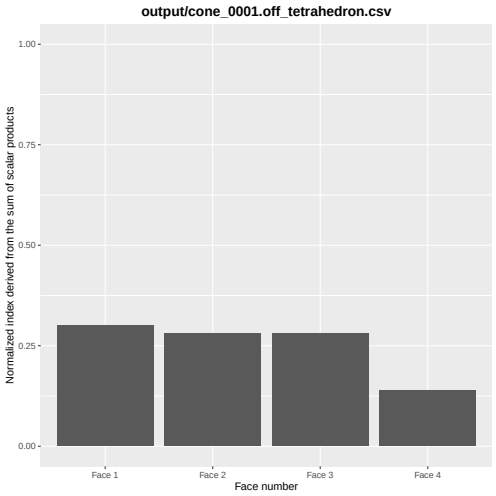
face_no	scal_sum	normalized_sum
Face 1	66.3834	0.312234
Face 2	63.1587	0.297067
Face 3	63.1587	0.297067
Face 4	19.9067	0.0936315

(b) Histogramme CSV du nuage de points « bottle_0001 » rapporté à un tétraèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Fichier généré par nos soins*

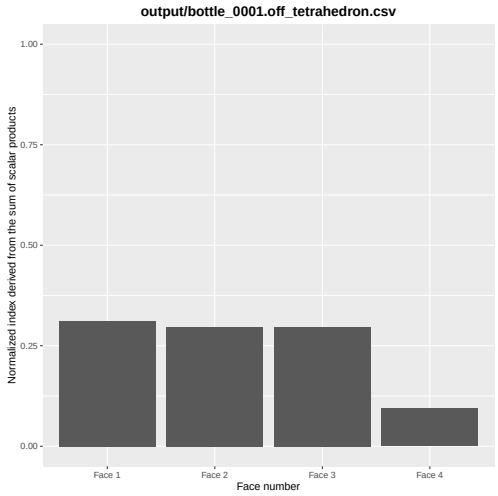
FIGURE 3.2 – Histogrammes au format CSV du cône et de la bouteille

Cela est bien corroboré par les versions graphiques des histogrammes, en figure 3.3 : les barres correspondant à chaque face du tétraèdre sont quasiment une à une égales quand on confronte les deux graphiques.



(a) Histogramme du nuage de points « cone_0001 » rapporté à un tétraèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique réalisé par nos soins*



(b) Histogramme du nuage de points « bottle_0001 » rapporté à un tétraèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique réalisé par nos soins*

FIGURE 3.3 – Histogrammes graphiques du cône et de la bouteille

Si on compare les deux modèles à l'aide de la matrice de corrélation de la figure 3.4, on voit bien que les similarités sont importantes : la moyenne des valeurs de la diagonale fait 97.72 %.

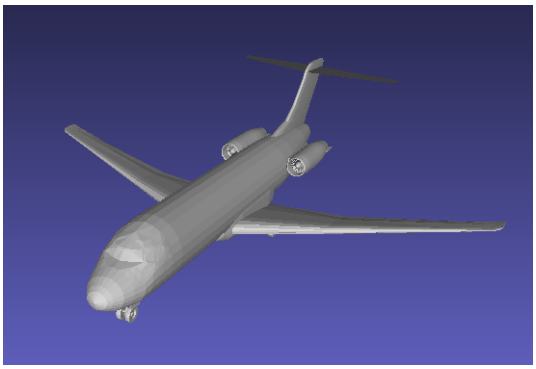
		cone_0001			
		Face 1	**Face 2**	**Face 3**	**Face 4**
bottle_0001	**Face 1**	98,77%	96,82%	96,82%	82,70%
	Face 2	99,72%	98,34%	98,34%	84,21%
	Face 3	99,72%	98,34%	98,34%	84,21%
	Face 4	79,37%	81,32%	81,32%	95,44%

FIGURE 3.4 – Matrice de corrélation entre le modèle « cone_0001 » et « bottle_0001 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Matrice générée par nos soins*

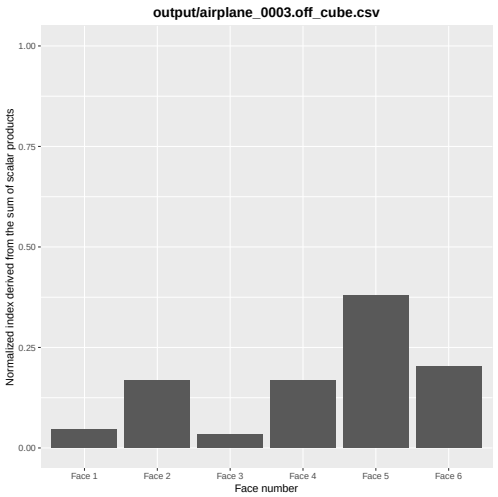
3.1.2 Échantillons rapportés à un cube

Pour montrer les capacités du cube en classification, nous avons choisi d'étudier les nuages de points `airplane_0003.off` et `airplane_0181.off`, dont la représentation et l'histogramme sont visibles en figure 3.5 et en figure 3.6. Il s'agit d'objets en trois dimensions, dont la forme est semblable à celle d'un cylindre.



(a) Visualisation du nuage de points « airplane_0003 »

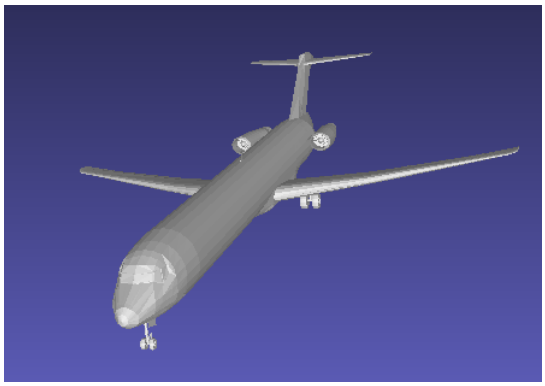
Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Histogramme du nuage de points « airplane_0003 » rapporté à un cube

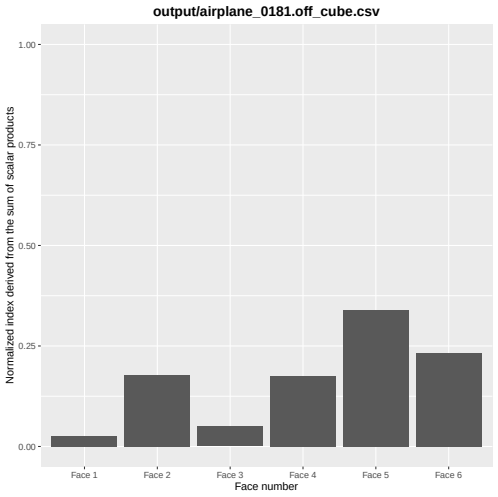
Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.5 – Nuage de points « airplane_0003 »



(a) Visualisation du nuage de points « airplane_0181 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Histogramme du nuage de points « airplane_0181 » rapporté à un cube

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.6 – Nuage de points « airplane_0181 »

Comme évoqué ci-dessus les valeurs le plus hautes se situent aux faces 5 et 6 du cube, correspondant aux bases inférieures et supérieures. De plus, les faces latérales (faces 2 et 4) ont des valeurs proches, ce qui confirme une symétrie de ces objets.

Si on compare les deux modèles à l'aide de la matrice de corrélation, de la figure 3.7, on voit bien que les deux nuages de points sont extrêmement similaires : la moyenne des valeurs de la diagonale fait 98.57 %.

		airplane_0181					
		Face 1	Face 2	Face 3	Face 4	Face 5	Face 6
airplane_0003	Face 1	99,9942	96,0849	99,9404	96,0933	93,4306	89,9902
	Face 2	94,6545	98,5638	94,5891	98,5554	98,7819	95,3415
	Face 3	98,1417	97,949	98,0763	97,9574	95,2947	91,8543
	Face 4	94,5831	98,4924	94,5177	98,484	98,8533	95,4129
	Face 5	95,6776	99,5869	95,6122	99,5785	97,7588	94,3184
	Face 6	91,0031	94,9124	90,9377	94,904	97,5667	98,9929

FIGURE 3.7 – Matrice de corrélation entre le modèle « airplane_0003 » et « airplane_0181 »

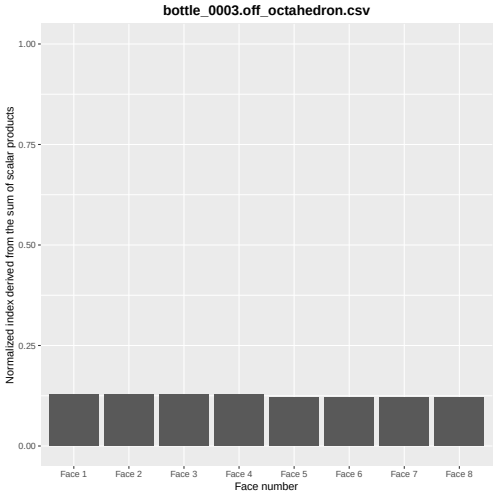
Source: *Base de données « ModelNet40 » de l'Université de Princeton / Matrice générée par nos soins*

3.1.3 Échantillons rapportés à un octaèdre

Cette fois-ci, nous nous sommes penchés sur les modèles `bottle_0003.off` et `bottle_0052.off`, dont la représentation et l'histogramme sont visibles en figure 3.8 et figure 3.9. La forme de ces bouteilles sont assimilables à des cylindres.

(a) Visualisation du nuage de points « bottle_0003 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



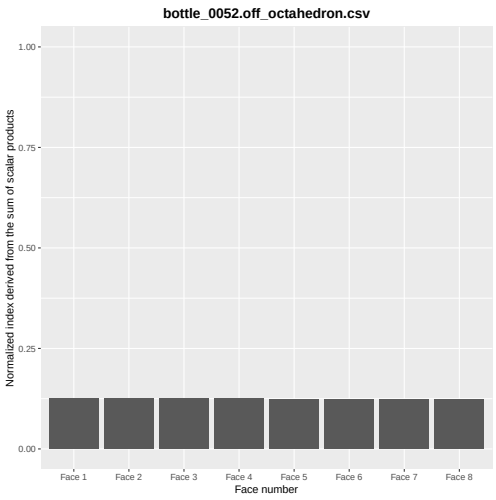
(b) Histogramme du nuage de points « bottle_0003 » rapporté à un cube

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.8 – Nuage de points « bottle_0003 »

(a) Visualisation du nuage de points « bottle_0052 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Histogramme du nuage de points « bottle_0052 » rapporté à un cube

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.9 – Nuage de points « bottle_0052 »

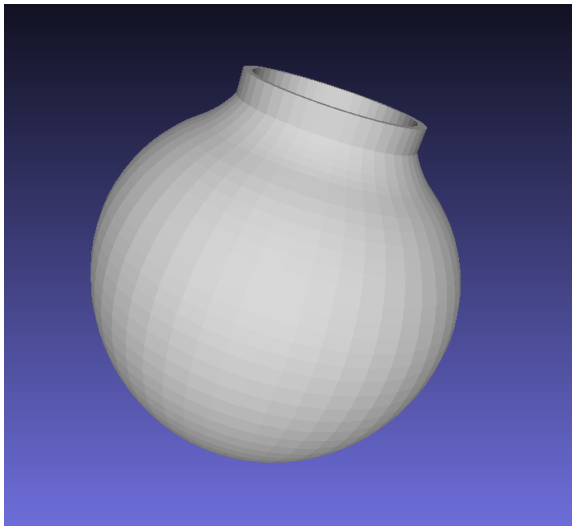
Nous observons que, pour chacun des objets, les colonnes sont presque toutes à la même valeur : on voit donc bien qu'il s'agit d'un objet très régulier et symétrique. Le polyèdre de référence, ici l'octaèdre, pourrait alors être bien adapté pour analyser ce type de forme.

Les deux histogrammes sont très similaires en proportion et semblent même identiques à l'œil nu. On peut alors penser que ça ne posera pas de problème particulier de détection, sauf si des nuages de points d'autres classes se trouvent avoir un histogramme lui aussi similaire.

3.1.4 Échantillons rapportés à un dodécaèdre

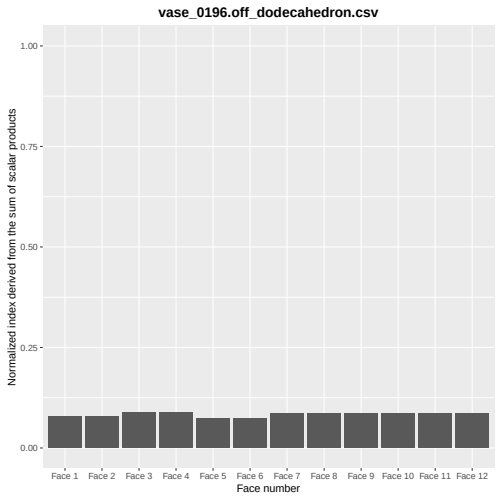
Pour le dodécaèdre, nous avons opté pour les objets `vase_0196.off` et `vase_0231.off`, dont la représentation et l'histogramme sont visibles en figure 3.10 et figure 3.11. La forme sphérique de ces objets les rend intéressants à comparer avec un polyèdre très détaillé.

Par ailleurs, ces nuages de points, bien que représentant le même type d'objets, sont assez différents en sphéricité et en détails.



(a) Visualisation du nuage de points « vase_0196 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



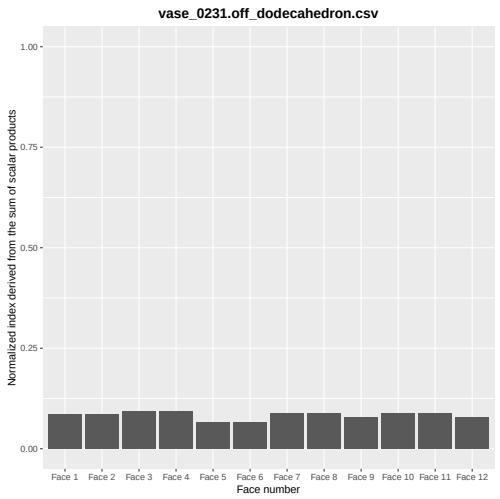
(b) Histogramme du nuage de points « vase_0196 » rapporté à un dodécaèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.10 – Nuage de points « vase_0196 »

(a) Visualisation du nuage de points « vase_0231 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Histogramme du nuage de points « vase_0231 » rapporté à un dodécaèdre

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.11 – Nuage de points « vase_0231 »

En général, les valeurs de la matrice de corrélation, en figure 3.12, oscillent entre 97 % et 99.9 %, c'est-à-dire que toutes les faces sont très similaires, ce qui est logique car un vase est un objet avec plusieurs axes de symétries. De plus, la moyenne de la diagonale est de 99.51 %.

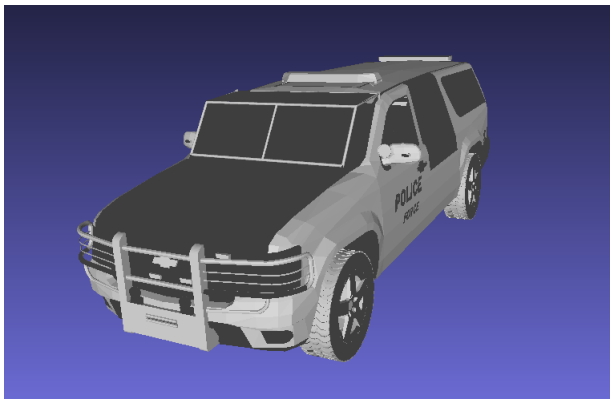
		vase_0231											
		Face 1	Face 2	Face 3	Face 4	Face 5	Face 6	Face 7	Face 8	Face 9	Face 10	Face 11	Face 12
vase_0196	Face 1	99,3999	99,3999	99,7246	99,7246	98,7093	98,7092	99,8825	99,8825	99,8743	99,8825	99,8825	99,8743
	Face 2	99,3999	99,3999	99,7246	99,7246	98,7093	98,7092	99,8825	99,8825	99,8743	99,8825	99,8825	99,8743
	Face 3	98,538	98,538	99,4134	99,4134	97,8473	97,8473	99,2555	99,2555	99,2638	99,2555	99,2555	99,2638
	Face 4	98,538	98,538	99,4134	99,4134	97,8473	97,8473	99,2555	99,2555	99,2638	99,2555	99,2555	99,2638
	Face 5	98,5966	98,5966	97,7211	97,7211	99,2872	99,2873	97,8791	97,8791	97,8708	97,8791	97,8791	97,8708
	Face 6	98,5966	98,5966	97,7211	97,7211	99,2872	99,2873	97,8791	97,8791	97,8708	97,8791	97,8791	97,8708
	Face 7	99,1521	99,1521	99,9725	99,9725	98,4614	98,4614	99,8696	99,8696	99,8779	99,8696	99,8696	99,8779
	Face 8	99,1521	99,1521	99,9725	99,9725	98,4614	98,4614	99,8696	99,8696	99,8779	99,8696	99,8696	99,8779
	Face 9	99,9912	99,9912	99,1157	99,1157	99,3182	99,3181	99,2737	99,2737	99,2654	99,2737	99,2737	99,2654
	Face 10	99,1521	99,1521	99,9725	99,9725	98,4614	98,4614	99,8696	99,8696	99,8779	99,8696	99,8696	99,8779
	Face 11	99,1521	99,1521	99,9725	99,9725	98,4614	98,4614	99,8696	99,8696	99,8779	99,8696	99,8696	99,8779
	Face 12	99,9912	99,9912	99,1157	99,1157	99,3182	99,3181	99,2737	99,2737	99,2654	99,2737	99,2737	99,2654

FIGURE 3.12 – Matrice de corrélation entre le modèle « vase_0196 » et « vase_0231 »

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Matrice générée par nos soins*

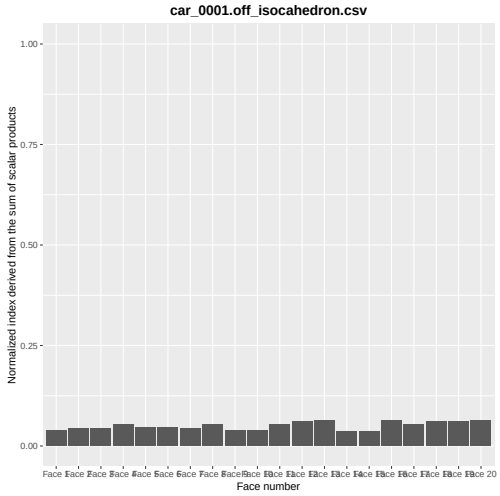
3.1.5 Échantillons rapportés à un icosaèdre

L'icosaèdre étant le modèle le plus détaillé dont nous disposons, nous avons mis à l'épreuve des objets assez complexes. Nous avons choisi des modèles de voitures, issus des nuages de points `car_0001.off` et `car_0002.off`, dont la représentation et l'histogramme sont visibles en figure 3.13 et figure 3.14.



(a) Visualisation du nuage de points « car_0001 »

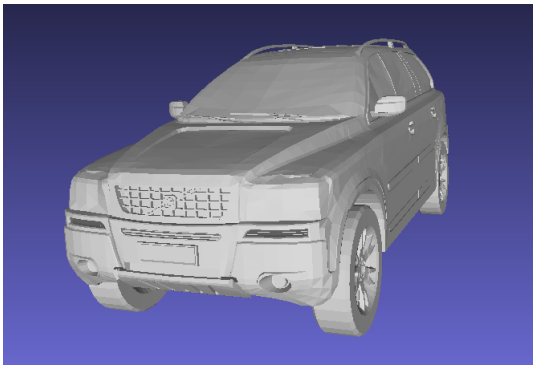
Source: *Base de données « ModelNet40 » de l'Université de Princeton / MeshLab*



(b) Histogramme du nuage de points « car_0001 » rapporté à un dodécaèdre

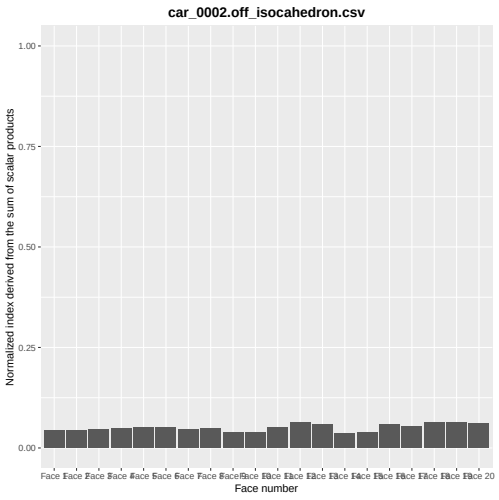
Source: *Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins*

FIGURE 3.13 – Nuage de points « car_0001 »



(a) Visualisation du nuage de points « car_0002 »

Source: Base de données « ModelNet40 » de l'Université de Princeton / MeshLab



(b) Histogramme du nuage de points « car_0002 » rapporté à un dodécaèdre

Source: Base de données « ModelNet40 » de l'Université de Princeton / Graphique généré par nos soins

FIGURE 3.14 – Nuage de points « car_0002 »

À l'aide de la matrice de corrélation de la figure 3.15, nous constatons que, majoritairement, les valeurs les plus élevées correspondent à la même face. On en déduit donc que ce volume de référence est adapté pour étudier ce type d'objet. Nous obtenons une similarité moyenne de 99.78 %.

		car_0002																		
		Face 1	Face 2	Face 3	Face 4	Face 5	Face 6	Face 7	Face 8	Face 9	Face 10	Face 11	Face 12	Face 13	Face 14	Face 15	Face 16	Face 17	Face 18	Face 19	Face 20
car_0001	Face 1	99,5183	99,9825	99,9934	99,207	99,739	99,7365	99,9947	99,2072	99,4322	99,4333	99,2107	98,2293	98,14	99,0763	99,0785	98,1381	99,2094	98,23	98,2622	98,0695
	Face 2	99,7104	99,8254	99,8145	99,015	99,547	99,5444	99,8132	99,0151	99,6242	99,6254	99,0187	98,0372	97,9479	99,2683	99,2706	97,946	99,0174	98,038	98,0701	97,8775
	Face 3	99,4323	99,8966	99,9075	99,293	99,825	99,8224	99,9088	99,2932	99,3462	99,3473	99,2967	98,3153	98,2259	98,9903	98,9925	98,224	99,2954	98,316	98,3482	98,1555
	Face 4	99,0745	99,5388	99,5496	99,6508	99,8172	99,8197	99,5509	99,651	98,9884	98,9895	99,6545	98,6731	98,5838	98,6325	98,6347	98,5819	99,6532	98,6738	98,706	98,5133
	Face 5	98,9755	99,4397	99,4506	99,7499	99,7181	99,7207	99,4519	99,75	98,8894	98,8905	99,7536	98,7721	98,6828	98,5335	98,5357	98,6809	99,7523	98,7729	98,805	98,6124
	Face 6	98,951	99,4152	99,4261	99,7744	99,6936	99,6962	99,4274	99,7745	98,8648	98,866	99,7781	98,7966	98,7073	98,5089	98,5112	98,7054	99,7767	98,7974	98,8295	98,6368
	Face 7	99,3913	99,8556	99,8665	99,334	99,866	99,8634	99,8678	99,3342	99,3052	99,3063	99,3377	98,3563	98,2669	98,9493	98,9515	98,265	99,3364	98,357	98,3892	98,1965
	Face 8	99,0478	99,5121	99,523	99,6775	99,7905	99,7931	99,5243	99,6777	98,9617	98,9628	99,6812	98,6998	98,6104	98,6058	98,608	98,6085	99,6799	98,7005	98,7327	98,54
	Face 9	99,974	99,5097	99,4988	98,6993	99,2313	99,2287	99,4975	98,6994	99,9399	99,9411	98,703	97,7215	97,6322	99,584	99,5862	97,6303	98,7017	97,7223	97,7544	97,5618
	Face 10	99,9552	99,4909	99,48	98,6805	99,2125	99,2099	99,4787	98,6806	99,9587	99,9599	98,6842	97,7027	97,6134	99,6028	99,605	97,6115	98,6829	97,7035	97,7356	97,543
	Face 11	98,7415	99,2057	99,2166	99,9838	99,4842	99,4867	99,2179	99,384	98,6554	98,6565	99,9875	99,0061	98,9168	98,2995	98,3017	98,9149	99,3862	99,0069	99,039	98,8463
	Face 12	97,7301	98,1944	98,2053	99,0048	98,4728	98,4754	98,2065	99,0046	97,644	97,6451	99,0011	99,9825	99,9281	97,2881	97,2903	99,9263	99,0024	99,9818	99,9496	99,8577
	Face 13	98,0774	98,5416	98,5525	99,3521	98,8201	98,8226	98,5538	99,3519	97,9913	97,9924	99,3484	99,6702	99,5809	97,6354	97,6376	99,579	99,3497	99,671	99,7031	99,5104
	Face 14	99,7545	99,2902	99,2793	98,4798	99,0118	99,0092	99,278	98,4799	99,8406	99,8395	98,4835	97,5021	97,4127	99,8035	99,8057	97,4108	98,4822	97,5028	97,535	97,3423
	Face 15	99,7694	99,3051	99,2943	98,4947	99,0267	99,0242	99,293	98,4949	99,8555	99,8544	98,4984	97,517	97,4277	99,7886	99,7908	97,4258	98,4971	97,5177	97,5499	97,3572
	Face 16	98,0495	98,5137	98,5246	99,3242	98,7922	98,7947	98,5259	99,324	97,9634	97,9645	99,3204	99,6981	99,6088	97,6075	97,6097	99,6069	99,3218	99,6989	99,731	99,5383
	Face 17	98,6942	99,1584	99,1693	99,9689	99,4369	99,4394	99,1706	99,9687	98,6081	98,6092	99,9652	99,0534	98,9641	98,2522	98,2544	98,9622	99,9665	99,0542	99,0863	98,8936
	Face 18	97,7	98,1643	98,1752	98,9747	98,4427	98,4453	98,1765	98,9745	97,6139	97,615	98,971	99,9524	99,9582	97,258	97,2602	99,9563	98,9723	99,9517	99,9195	99,8878
	Face 19	97,624	98,0882	98,0991	98,8986	98,3666	98,3692	98,1004	98,8985	97,5378	97,539	98,8949	99,8764	99,9657	97,182	97,1842	99,9676	98,8962	99,8756	99,8435	99,9639
	Face 20	97,9708	98,4351	98,446	99,2455	98,7135	98,7161	98,4473	99,2454	97,8847	97,8858	99,2418	99,7767	99,6874	97,5288	97,531	99,6855	99,2431	99,7775	99,8097	99,617

FIGURE 3.15 – Matrice de corrélation entre le modèle « car_0001 » et « car_0002 »

Source: Base de données « ModelNet40 » de l'Université de Princeton / Matrice générée par nos soins

3.2 Reconnaissance automatique des objets

Dans la partie précédente, nous avons vu comment nous pouvons comparer des modèles entre eux à l'aides des histogrammes que nous avons généré. À présent, l'objectif est de réussir à identifier à quelle classe d'objet appartient chacun des modèles de test de la base de données, et surtout de voir à quel point cela est possible.

Pour ce faire, nous avons réalisé un algorithme en Python. Celui-ci fait le tour de tous les modèles de la base de données et calcule un indice de similarité entre chaque modèle et celui dont on cherche à déterminer l'appartenance.

3.2.1 Calcul de l'indice de similarité

Notre indice de similarité est assez simple à calculer. Il s'agit simplement de la somme de l'écart face à face entre deux modèles :

$$I = \sum_{k} |F_{k\text{ref}} - F_{k\text{test}}|$$

(avec F_k l'indice de la face k sur l'histogramme)

3.2.2 k plus proches voisins

Par la suite, nous avons implémenté un algorithme des « k plus proches voisins ». On sélectionne les k modèles avec l'indice de similarité le plus élevé, et on cherche quel classe revient le plus dans ceux-ci.

Dans le contexte de notre programme, on a $k = 5$, c'est-à-dire qu'on prend en compte les cinq modèles les plus proches de celui à analyser.

3.2.3 Résultats

Afin de traiter les résultats avec facilité, nous avons réalisé quelques scripts en Bash qui nous permettent notamment d'extraire des statistiques.

Par exemple, le script `find_best.sh` permet d'obtenir le meilleur résultat pour chaque classe d'objet, c'est-à-dire le polyèdre qui identifie le plus souvent la classe de l'objet en question et le pourcentage de réussite. Par exemple, le fichier de la figure 3.16 a été généré par ce script.

```
airplane compared to icosahedron -> 96.00 %
bathtub compared to cube -> 56.00 %
bed compared to icosahedron -> 70.00 %
bench compared to cube -> 25.00 %
bookshelf compared to icosahedron -> 62.00 %
bottle compared to icosahedron -> 64.00 %
bowl compared to dodecahedron -> 50.00 %
car compared to icosahedron -> 76.00 %
chair compared to icosahedron -> 73.00 %
cone compared to dodecahedron -> 60.00 %
cup compared to cube -> 15.00 %
curtain compared to cube -> 55.00 %
desk compared to icosahedron -> 38.37 %
door compared to cube -> 75.00 %
dresser compared to icosahedron -> 55.81 %
flower_pot compared to octahedron -> 5.00 %
glass_box compared to icosahedron -> 92.00 %
guitar compared to icosahedron -> 49.00 %
keyboard compared to octahedron -> 45.00 %
lamp compared to cube -> 15.00 %
laptop compared to cube -> 75.00 %
mantel compared to icosahedron -> 85.00 %
monitor compared to cube -> 88.00 %
night_stand compared to icosahedron -> 39.53 %
person compared to icosahedron -> 30.00 %
piano compared to dodecahedron -> 43.00 %
plant compared to cube -> 24.00 %
radio compared to icosahedron -> 10.00 %
range_hood compared to icosahedron -> 59.00 %
sink compared to icosahedron -> 15.00 %
sofa compared to icosahedron -> 86.00 %
stairs compared to cube -> 10.00 %
stool compared to dodecahedron -> 30.00 %
table compared to icosahedron -> 74.00 %
tent compared to icosahedron -> 25.00 %
toilet compared to icosahedron -> 67.00 %
tv_stand compared to icosahedron -> 25.00 %
vase compared to icosahedron -> 70.00 %
wardrobe compared to icosahedron -> 40.00 %
xbox compared to cube -> 45.00 %

Overall: 50.44 %
```

FIGURE 3.16 – Meilleurs résultats pour chaque classe

Source: *Base de données « ModelNet40 » de l'Université de Princeton / Fichier généré par nos soins*

Les résultats sont très bons pour certaines classes d'objet (souvent celles contenant les objets les plus complexes, comme les voitures ou les avions), en général lorsqu'on les compare à des formes très détaillées, telles le dodécaèdre ou l'icosaèdre. Il arrive que le cube soit la forme la plus efficace pour certaines comparaisons, notamment lorsque l'objet a une forme cubique (ordinateur portable, écran, porte).

Les résultats sont tout de même mitigés, puisqu'on arrive à peine au dessus de la moitié des objets reconnus.

3.2.4 Base de données réduite

Afin de potentiellement augmenter ces statistiques, nous avons réalisé des essais sur le jeu de données « ModelNet10 », qui est une version réduite à dix classes d'objets soigneusement sélectionnés et dont l'orientation a été corrigée. Dès lors, les résultats sont bien meilleurs, comme le montre le fichier de la figure 3.17.

```
bathtub compared to cube -> 66.00 %
bed compared to cube -> 81.00 %
chair compared to icosahedron -> 83.00 %
desk compared to icosahedron -> 40.69 %
dresser compared to icosahedron -> 62.79 %
monitor compared to icosahedron -> 90.00 %
night_stand compared to icosahedron -> 51.16 %
sofa compared to cube -> 88.00 %
table compared to cube -> 86.00 %
toilet compared to icosahedron -> 79.00 %

Overall: 72.76 %
```

FIGURE 3.17 – Meilleurs résultats pour chaque classe

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

De plus, pour la base de données réduite à 10 classes, il nous est possible de faire des matrices de confusion – une par polyèdre – afin de mieux pouvoir observer l'efficacité de chaque solide de référence dans la description d'un nuage de points.

Efficacité du tétraèdre

	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet	?
bathtub	4,00%	22,00%	44,00%	0,00%	2,00%	8,00%	0,00%	4,00%	0,00%	6,00%	10,00%
bed	0,00%	56,00%	20,00%	0,00%	0,00%	0,00%	3,00%	1,00%	2,00%	1,00%	17,00%
chair	0,00%	10,00%	55,00%	0,00%	1,00%	0,00%	1,00%	0,00%	3,00%	10,00%	20,00%
desk	0,00%	8,14%	4,65%	31,40%	0,00%	2,33%	4,65%	17,44%	17,44%	2,33%	11,63%
dresser	0,00%	1,16%	15,12%	4,65%	26,74%	4,65%	2,33%	8,14%	1,16%	3,49%	32,56%
monitor	0,00%	0,00%	5,00%	0,00%	2,00%	82,00%	1,00%	3,00%	0,00%	2,00%	5,00%
night_stand	0,00%	8,14%	19,77%	6,98%	3,49%	0,00%	6,98%	11,63%	10,47%	11,63%	20,93%
sofa	0,00%	5,00%	1,00%	2,00%	2,00%	0,00%	1,00%	83,00%	0,00%	1,00%	5,00%
table	0,00%	2,00%	2,00%	9,00%	0,00%	0,00%	2,00%	0,00%	74,00%	0,00%	11,00%
toilet	0,00%	3,00%	37,00%	0,00%	3,00%	0,00%	2,00%	0,00%	0,00%	39,00%	16,00%

FIGURE 3.18 – Matrice de confusion avec le tétraèdre en polyèdre de référence

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

Dans le cas présent, on voit bien que, dans la certains cas, les valeurs les plus élevées sont situées dans la diagonale. Cependant, les pourcentages de reconnaissance fructueuse sont assez souvent faibles, à cause du peu de détails véhiculés par les faces du tétraèdre. Par exemple, dans le cas des baignoires, seules 4 % d'entre elles sont correctement reconnues, mais 44 % sont reconnues en tant que chaises !

Efficacité du cube

	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet	?
bathtub	64,00%	6,00%	8,00%	0,00%	4,00%	2,00%	0,00%	6,00%	2,00%	2,00%	6,00%
bed	0,00%	79,00%	8,00%	1,00%	0,00%	0,00%	3,00%	0,00%	3,00%	2,00%	4,00%
chair	0,00%	6,00%	74,00%	1,00%	0,00%	0,00%	1,00%	0,00%	2,00%	10,00%	6,00%
desk	1,16%	4,65%	9,30%	32,56%	3,49%	0,00%	6,98%	5,81%	17,44%	1,16%	17,44%
dresser	4,65%	3,49%	1,16%	2,33%	51,16%	5,81%	5,81%	5,81%	1,16%	5,81%	12,79%
monitor	1,00%	0,00%	0,00%	0,00%	4,00%	88,00%	0,00%	3,00%	0,00%	1,00%	3,00%
night_stand	0,00%	5,81%	4,65%	2,33%	8,14%	2,33%	31,40%	11,63%	6,98%	9,30%	17,44%
sofa	1,00%	3,00%	0,00%	3,00%	0,00%	0,00%	1,00%	86,00%	1,00%	0,00%	5,00%
table	0,00%	1,00%	4,00%	5,00%	0,00%	0,00%	1,00%	0,00%	83,00%	0,00%	6,00%
toilet	0,00%	4,00%	20,00%	0,00%	0,00%	0,00%	2,00%	0,00%	0,00%	69,00%	5,00%

FIGURE 3.19 – Matrice de confusion avec le cube en polyèdre de référence

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

Le cube donne des résultats bien meilleurs. La plupart des pourcentages dans les diagonales sont supérieurs à 50 %. En revanche, dans le cas particulier des bureaux et des tables de nuit, les résultats sont bien plus bas : en effet, il semblerait que les modèles de bureaux soient très souvent confondus avec les modèles de tables (dans près de 20 % des cas), ce qui est logique car ce sont des objets aux formes assez proches.

Efficacité de l'octaèdre

	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet	?
bathtub	54,00%	12,00%	0,00%	0,00%	0,00%	4,00%	2,00%	8,00%	0,00%	6,00%	14,00%
bed	0,00%	62,00%	21,00%	1,00%	1,00%	0,00%	3,00%	0,00%	0,00%	3,00%	9,00%
chair	0,00%	13,00%	70,00%	1,00%	0,00%	0,00%	2,00%	0,00%	2,00%	4,00%	8,00%
desk	0,00%	5,81%	9,30%	30,23%	4,65%	0,00%	12,79%	9,30%	15,12%	1,16%	11,63%
dresser	4,65%	1,16%	0,00%	2,33%	45,35%	6,98%	5,81%	17,44%	1,16%	0,00%	15,12%
monitor	1,00%	1,00%	0,00%	0,00%	4,00%	88,00%	1,00%	4,00%	1,00%	0,00%	0,00%
night_stand	1,16%	2,33%	2,33%	3,49%	9,30%	1,16%	30,23%	10,47%	16,28%	0,00%	23,26%
sofa	1,00%	2,00%	0,00%	2,00%	5,00%	0,00%	1,00%	84,00%	0,00%	0,00%	5,00%
table	0,00%	1,00%	2,00%	7,00%	0,00%	0,00%	5,00%	1,00%	80,00%	0,00%	4,00%
toilet	2,00%	9,00%	21,00%	0,00%	1,00%	0,00%	1,00%	0,00%	1,00%	49,00%	16,00%

FIGURE 3.20 – Matrice de confusion avec l'octaèdre en polyèdre de référence

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

Concernant l'octaèdre, les résultats sont similaires à ceux du cube. On observe toujours les mêmes soucis avec les bureaux et les tables de nuits. On régresse même un peu sur certaines valeurs.

Efficacité du dodécaèdre

	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet	?
bathtub	52,00%	10,00%	8,00%	0,00%	4,00%	4,00%	2,00%	4,00%	2,00%	4,00%	10,00%
bed	0,00%	77,00%	10,00%	1,00%	0,00%	0,00%	1,00%	0,00%	3,00%	2,00%	6,00%
chair	0,00%	7,00%	75,00%	1,00%	0,00%	0,00%	2,00%	0,00%	2,00%	7,00%	6,00%
desk	1,16%	4,65%	6,98%	31,40%	4,65%	0,00%	9,30%	11,63%	17,44%	1,16%	11,63%
dresser	4,65%	2,33%	0,00%	0,00%	45,35%	5,81%	9,30%	9,30%	2,33%	3,49%	17,44%
monitor	1,00%	0,00%	0,00%	0,00%	4,00%	88,00%	2,00%	4,00%	0,00%	0,00%	1,00%
night_stand	0,00%	6,98%	2,33%	8,14%	9,30%	1,16%	32,56%	6,98%	6,98%	5,81%	19,77%
sofa	1,00%	3,00%	0,00%	3,00%	2,00%	0,00%	1,00%	85,00%	1,00%	1,00%	3,00%
table	0,00%	3,00%	2,00%	7,00%	0,00%	0,00%	2,00%	0,00%	78,00%	0,00%	8,00%
toilet	0,00%	5,00%	22,00%	0,00%	0,00%	0,00%	0,00%	0,00%	1,00%	59,00%	13,00%

FIGURE 3.21 – Matrice de confusion avec le dodécaèdre en polyèdre de référence

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

On arrive encore aux mêmes conclusions. L'algorithme a toujours des difficultés à discerner les objets très similaires. Encore une fois, les valeurs ne progressent pas avec l'augmentation du nombre de faces. On perd même 10 points dans le cas des baignoires !

Efficacité de l'icosaèdre

	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet	?
bathtub	58,00%	6,00%	2,00%	0,00%	0,00%	4,00%	0,00%	6,00%	4,00%	14,00%	6,00%
bed	0,00%	77,00%	8,00%	1,00%	1,00%	0,00%	3,00%	0,00%	3,00%	0,00%	7,00%
chair	0,00%	5,00%	81,00%	1,00%	0,00%	0,00%	2,00%	0,00%	2,00%	5,00%	4,00%
desk	1,16%	6,98%	3,49%	33,72%	3,49%	0,00%	8,14%	11,63%	13,95%	0,00%	17,44%
dresser	2,33%	2,33%	0,00%	2,33%	55,81%	4,65%	9,30%	6,98%	2,33%	1,16%	12,79%
monitor	2,00%	0,00%	0,00%	0,00%	2,00%	89,00%	1,00%	5,00%	0,00%	0,00%	1,00%
night_stand	0,00%	6,98%	1,16%	10,47%	6,98%	1,16%	44,19%	9,30%	8,14%	0,00%	11,63%
sofa	1,00%	3,00%	0,00%	3,00%	1,00%	0,00%	1,00%	87,00%	1,00%	1,00%	2,00%
table	0,00%	2,00%	2,00%	9,00%	0,00%	0,00%	1,00%	0,00%	79,00%	0,00%	7,00%
toilet	0,00%	3,00%	17,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	74,00%	6,00%

FIGURE 3.22 – Matrice de confusion avec le icosaèdre en polyèdre de référence

Source: *Base de données « ModelNet10 » de l'Université de Princeton / Fichier généré par nos soins*

Les comparaisons des objets avec l'icosaèdre améliore cette fois-ci grandement les valeurs, et réduit les confusions possibles. On gagne quelques points sur chacun des objets et cela en fait, si il ne fallait en choisir qu'un, le meilleur polyèdre de référence pour identifier les objets de la base de données.

4

Conclusion

L'objectif du projet était de décrire, d'analyser et de comparer des objets en trois dimensions, à partir d'une base de données. En l'occurence, il s'agissait d'utiliser des méthodes normalisées pour confronter différents nuages de points en trois dimensions entre eux.

Pour ce faire, nous avons eu recours à des polyèdres réguliers de référence. Pour chacune de leurs faces, nous avons calculé leur vecteur normal. En les comparant aux vecteurs normaux de chaque face des nuages de points, nous avons pu obtenir des histogrammes permettant de modéliser chaque échantillon en fonction d'un solide de référence.

La réalisation de ce projet nous a aussi permis de nous initier au traitement de modèles et de données, d'appréhender de nouveaux logiciels et langages de programmation (MeshLab, R, ...). De plus, nous avons ainsi pu appliquer quelques notions de géométrie en trois dimensions (produit scalaire, produit vectoriel, ...) et comprendre leurs applications diverses. Nous avons également découvert un outil très utilisé en traitement de données : la matrice de confusion. Dans notre cas, les matrices de confusion ont servi à témoigner de la fiabilité et des capacités de reconnaissance de notre algorithme.

Nous avons également pu approfondir et appliquer certaines connaissances acquises cette année. Par exemple, nous avons utilisé les principes de la programmation orientée objets en C++ pour réaliser le programme en charge de l'analyse des fichiers OFF. Les compétences en Bash acquises en cours de *Systèmes Unix* nous ont également été utiles à de nombreuses reprises !

Si nous avions disposés de plus de temps, il aurait été intéressant d'ajuster plus précisément notre algorithme des k plus proches voisins, en adaptant la valeur de k en fonction des résultats obtenus. Par ailleurs, afin de tester les capacités de la méthode retenue et de notre algorithme, nous aurions pu effectuer des tests de reconnaissance sur des modèles externes, soit trouvés en ligne, soit scannés à l'aide d'un LiDAR par exemple. Enfin, notre algorithme de reconnaissance actuel ne tient pas compte du fait que les objets peuvent être orientés différemment. La face i du premier modèle est comparée avec la face i du deuxième dans l'état actuel des choses. Par conséquent, si l'objet est tourné de 90° selon un axe, l'algorithme le reconnaîtra beaucoup plus difficilement.