

STUDENT : HAJBI EL HASSAN  
SUPERVISOR @SCHNEIDER : OLIVIER COULON



Schneider Electric

---

Report Internship PFE

# Multimodal AI For Business Productivity

---

Supervisor at Ensimag : Christophe Picard

Grenoble, February to August

# Contents

<b>1 Credits</b>	<b>4</b>
<b>2 Company information</b>	<b>5</b>
<b>3 Introduction</b>	<b>5</b>
3.1 Global Context . . . . .	5
3.1.1 Context of the mission . . . . .	5
3.1.2 Cross-Functional Collaboration . . . . .	6
3.2 Problem and vision . . . . .	6
3.3 Related UseCases . . . . .	7
3.4 Digital and AI for QVE/standardisation and productivity opportunities . .	7
3.5 3D search engine . . . . .	8
3.6 Roadmap . . . . .	9
3.7 Solution . . . . .	10
<b>4 Technical solutions studied</b>	<b>11</b>
4.1 Benchmark . . . . .	11
4.1.1 Internal Benchmark . . . . .	11
4.1.2 External Benchmark . . . . .	12
4.2 Metadata VS Embeddings similarity . . . . .	13
4.3 Internal APIs . . . . .	14
4.4 3D similarity . . . . .	16
4.4.1 Data formats . . . . .	16
4.4.2 Evaluation . . . . .	19
4.4.3 Non-Machine Learning algorithms . . . . .	19
4.4.4 Machine Learning algorithms . . . . .	25
4.4.5 Benchmark . . . . .	31
4.5 Textual semantic similarity . . . . .	32
4.6 Generation . . . . .	34
4.6.1 LLMs for analyzing products descriptions dissimilarities at scale . .	36
4.6.2 VLMs for analyzing products images dissimilarities at scale . . . .	36
4.7 Conclusion . . . . .	37
4.8 Limitations and further studies . . . . .	37
<b>5 Status of the advancement of the project and critical analysis</b>	<b>39</b>

<b>6</b>	<b>Environmental and social footprint</b>	<b>39</b>
<b>7</b>	<b>References</b>	<b>41</b>

## 1 Credits

I would like to express my heartfelt gratitude to everyone who supported and guided me during my apprenticeship at Schneider Electric. This experience has been immensely enriching and has contributed significantly to my personal and professional growth.

I am deeply thankful to Olivier Coulon (Should Cost Manager), my internship supervisor, and Virah, his Chief (V.P) for providing me with invaluable insights, mentorship, and continuous encouragement throughout my internship. Their expertise and guidance have been instrumental in shaping my understanding of what it is like to work inside Schneider, more importantly insights about Procurement, Finance, Quality Value Engineering, Should cost, buyers, raw materials market, marketing, design/engineering team ...

I extend my sincere appreciation to the entire Global Supply Chain and AI HUB at Schneider Electric - Grenoble (Electropole and Intencity sites) for their warm welcome and collaborative spirit. I am grateful for the opportunity to work alongside passionate professionals who generously shared their knowledge and experiences, which greatly enhanced my learning journey.

I also want to acknowledge my colleagues Loic, Benjamin, Virginie, Yi, Ludovic, Fanny, Sylvestre, Vincent, Claire and Elodie, whose support, camaraderie, and willingness to answer my questions made the work environment both enjoyable and productive.

My gratitude extends to the wider Schneider Electric community for fostering an environment of innovation and excellence, enabling me to contribute to projects that align with the team's mission of leveraging Schneider's supply chain and change management.

In conclusion, my internship at Schneider Electric has been a transformative experience that has equipped me with valuable skills, insights, and a deeper appreciation for the field of artificial intelligence, statistics and machine learning. I am truly honored to have been part of Schneider Electric's journey. Thank you all for making this internship an unforgettable and rewarding chapter in my professional development.

## 2 Company information

Schneider Electric is a global organization dedicated to optimizing energy usage on a worldwide scale. The company operates across three primary business segments: Energy Management, Industrial, and Innovation. Schneider Electric boasts an array of essential global functions, including Finance, Human Resources, Marketing, Supply Chain, Governance, Digital, Strategy, and Sustainability. With a widespread presence spanning Asia, Europe, Africa (including the Middle East), and the Americas, Schneider Electric embodies a multinational spirit. The company's diverse community fosters a rich multicultural environment, which not only enhances cultural understanding but also cultivates excellent communication skills.

## 3 Introduction

### 3.1 Global Context

#### 3.1.1 Context of the mission

In the global context of our mission, we aim to identify value-enhancing opportunities across our extensive range of products and parts, from QVE<sup>1</sup> initiatives to standardization projects, and beyond. Our objective is to benchmark these against competitors' offerings effectively. Currently, this analysis is a manual process, presenting significant challenges: extended completion timelines, often requiring a team of four up to a week to assess hundreds of products or parts, and limitations on the scope due to a limit of products we can process manually.

This Proof of Concept (POC) holds strategic significance, not only in addressing these operational inefficiencies but also in fostering the growth of our digital capabilities. We aspire to build a proficient digital team, promote the integration of digital and AI solutions within our business and procurement processes. The ultimate goal is to generate greater value for Schneider Electric without expanding our production volume, thereby enhancing margins intelligently. This initiative aligns with the company's financial objectives and underscores the importance of digital transformation. As a non-digital native company, we strive to raise awareness of the positive impact of digital technologies and AI, synergizing our efforts with those of internal communities, groups, and enthusiasts who champion this evolution.

---

<sup>1</sup>Quality value engineering: A process that involves identifying ways to enhance the value of a product or service. It aims to provide the best possible combination of function and cost.

### 3.1.2 Cross-Functional Collaboration

As the sole developer on this project, my job is to tackle the biggest challenges facing our business. First, identifying the key issues and their effects, then working with the team to brainstorm solutions. It's important to estimate how much value these solutions will bring before diving in. My hands-on role includes designing the technical plan, building the solution, and constantly refining it based on feedbacks.

I didn't work alone. Close collaboration is key. I teamed up with Marketing to understand customer needs, Design and engineering team, TSM<sup>2</sup>, experts like Quality Value Engineers, the AI specialists, and the PhD Procurement team to get feedbacks that helped me iterate and improve the solution. All this happened under the guidance of my manager Olivier Coulon, who ensures our work aligns with the company's goals.

Together, we form a team that turns complex problems into innovative solutions, always aiming to improve our operations and make sure our efforts lead to real gains. By combining individual expertise and shared vision, we strive for efficient and impactful results.

## 3.2 Problem and vision

We found out that having millions of products and parts, made Schneider' products portfolio very complex to navigate. Finding similar subassemblies<sup>3</sup> used in different line of businesses is not an easy task, while this helps experts launch project for standardisation (for example) as the main goal is reducing costs. Our vision then is to develop tools to navigate the complexity of our product and parts portfolio. Navigating does not only mean finding similarities, but it is a large spectrum of use cases from (retrieval, similarities, clustering, generation). Generation can be : given a product range, we want to generate a paragraph expliciting the main differences between our *time delay relays*.

Our vision is to support SE<sup>4</sup> equity journey<sup>5</sup> and keeps SE ahead of the competition, to scale up our Products competitiveness and efficiency and to Improve efficiency on technical productivity cycle (Process, People and Execution).

Today we need to increase our business productivity, and help our buyers navigate the complex product portfolio while taking into account the wide range of criterias and data

---

<sup>2</sup>Technology Standardization Management

<sup>3</sup>a smaller section of a product that is built separately and later combined with other subassemblies to form the final product.

<sup>4</sup>Schneider electric

<sup>5</sup>a project inside sheneider electric aiming to improve profitability by improving efficiency instead of increasing the volume of production.

attributes : from design and marketing to finance and procurement.

Thus, we can enhance our business productivity from the final Product level in the different lines of businesses (product lines) : Power Products and Home and distribution

From Base parts level : Starting with fasteners and wallbox.

### 3.3 Related UseCases

A POC that unlocks more than a usecase. We ended up discovering many ways to improve the productivity, *whether it is within the different Lines Of Business, in the plants or for the design/engineering team*. We selected 4 different usecases :

**Rationalization in early design Catalog and custom parts rationalisation** : I am a Standardization Engineer, scanning Symphony to identify so-called 'custom' parts that could be standardized to catalog parts within the CSM<sup>6</sup> library.

**Design Efficiency** : *Before/while designing in symphony new parts, searching in existing catalog parts (design support)* : I am a Designer, creating a part in Symphony. I check if this part (or a similar one) already exists as a Standard in the CSM library.

**Rationalization in mass productivity** : *Supply chain product's simplification for factory*: As a Standardization Engineer, starting from parts purchased by GSC, I search for commonalities to propose a rationalization of the different parts purchased, reducing their variety.

**Reinforce O2 practices** : *Given products from different regions and BU<sup>7</sup>* : find opportunities of rationalisations and standardisation and measure the potential savings.

During this internship project, we focused on reinforcing O2 practices by leveraging textual/visual similarities of products to retrieve standardisation opportunities..

### 3.4 Digital and AI for QVE/standardisation and productivity opportunities

Two major problems today when navigating products portfolio manually are **latency** (time to an impacting project) and **scope** (number of products we can process). QVE<sup>8</sup> is done following several major steps (from **information gathering** to **standardisation project implementation**), it is time consuming and the scope of products to analyse is

<sup>6</sup>internal database containing standardised parts

<sup>7</sup>business unit

<sup>8</sup>Quality value engineering

very large. This is the reason why, we propose using digital and AI to reduce the latency and increase the scope by tackling the first three steps :

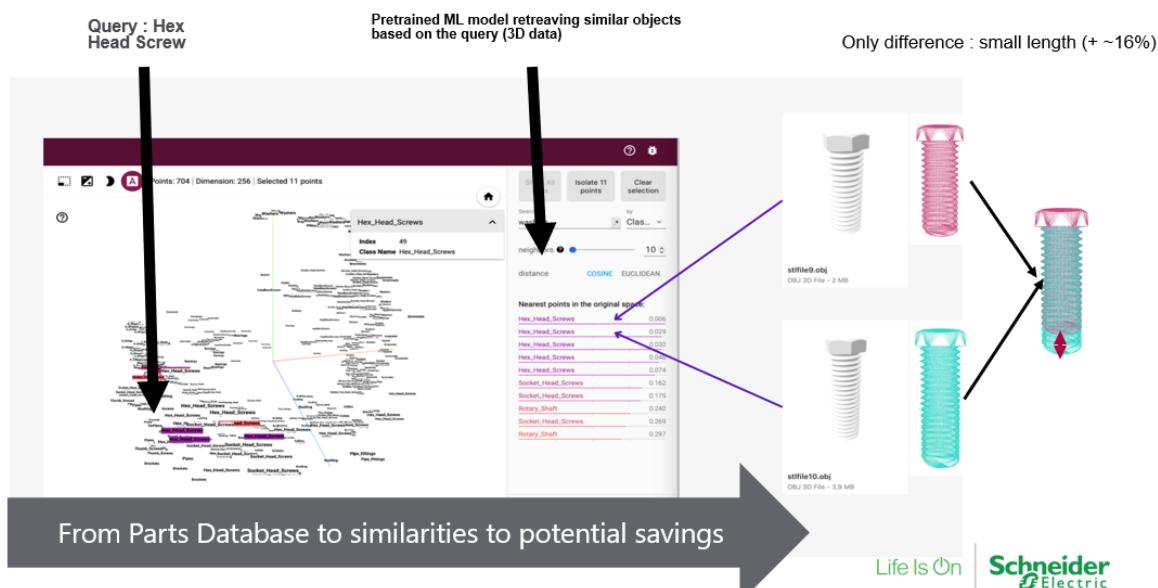
- Infos gathering by merging different data sources from marketing and design to finance and supply chain
- Brainstorming by using dedicated tool for Similarities retrieval
- Generation based on text and images. And then evaluating our potential savings.

### 3.5 3D search engine

To start with base parts, we developed an end to end AI search engine evaluated on Non-Schneider opensource dataset containing 24 categories of fasteners (approximately 4 thousands parts).

We designed a system where you can select the 3D data of your base part as the query and specify the parts database in which you wish to perform the search. Our pretrained machine learning model will then retrieve similar parts from this database and based on the results calculate the potential savings we can make by standardizing.

Here in the example below (figure 4.1), we have chosen a hexagonal head screw as a query and the database we are searching in is the one described before, containing 4K parts. The model we developed is able to retrieve similar hex screws and we show that the gap is really small, and can be quantified for further analysis.



### 3.6 Roadmap

#### Step 1: Non-ML for 3D Similarity

- Objective : Conducting a state-of-the-art analysis before implementing machine learning offers several advantages: understanding data formats, establishing a baseline for comparison, and building confidence in our approach by demonstrating existing results.
- In the end : Data formats to use, and evaluation of few approaches.

#### Step 3: a bit of engineering

- Objective : integrate expert rules and prior knowledge in our search engine, as filters before the design retrieval in latent space.
- In the end : a better retrieval that may correspond more to our experts' expectations.

#### Step 2: ML for Product Retrieval

- Objective : feature database using ML, a large dataset with parts but also subassemblies is a very good starting point for later doing multimodality with a joint representation.
- In the end : Part retrieval based on 3D similarity only + comparison with other methods from step1, clustering of latent space and standardisation proposal.

#### Step 4: Build the architecture of Multiomdality on top of the existing solution

Semaine	Février				Mars				Avril				Mai				Juin				Juillet				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Introduction																									
- Understanding the project																									
- 3D similarity review																									
I - use case studies																									
- standardisation engineering understanding																									
- qve process understanding																									
- studying related use cases																									
- validation of the use case 3d similarity																									
II - dataset construction																									
- finding schneider internal databases related to the use case																									
- fetching the data																									
- cleaning and understanding the data, modalities and formats																									
II - NON-ML for 3D Similarity																									
- data formats and conversion methods																									
- state of the art review of non-ml 3d similarity methods																									
- implementation and evaluation																									
II - ML for Product Retrieval																									
- find open source datasets of 3d models																									
- use a suitable machine learning architecture																									
- evaluation of similarity, retrieval and clustering performances																									
II - a bit of engineering																									
- gathering feedbacks from internal experts																									
- pitching to global procurement committee for strategic decisions																									
- pitching to the lines of businesses																									
III - Build the architecture of Multimodality on top of the existing solution																									
- creating three blocs: 3d similarity, textual semantic similarity and LLM/LVM generation																									
- creating the multimodal pipeline																									
- Report writing																									

### 3.7 Solution

$$\sum_{j=1}^n w_j x_j$$

Few steps, thousand products, similarities retrieval in less than 1 min.

What we suggested as a solution is a pipeline merging three blocks together : **Textual semantic similarity**<sup>9</sup>, **Visual similarity**<sup>10</sup> and **LLM/LVM generation**<sup>11</sup>. The pipeline should be **multimodal** (different data modalities), **handle different data sources**, **be value centered, reliable and easy to monitor**.

One main question would be : «why not use only datasheets»? «Why would we require visual data» ? The answer relies on user perspective : visual data is generic and universal, while textual descriptions are not (for example the nomination can differ from one person, country or company to another), and technically speaking visual data may contain some visual features/details that are not encoded into text. However we may lack 2D or 3D images <sup>12</sup>. Reason why, the pipeline should handle both of these scenarios : if we have 3D data we should use it to refine our analyses otherwise deal with the missing data by using only text.

Then we can move to products. We introduce an approach that leverages AI to handle thousands of products and retrieve similarities and analysis of products differences within few steps. Here we extract a big amount of products from the database we are interested

<sup>9</sup>refer to textual semantic similarity section

<sup>10</sup>refer to 3D similarity section

<sup>11</sup>refer to Generation section

<sup>12</sup>because of missing data

in (Power Products in our trial), we get textual descriptions from the marketing which also contain some technical attributes and visual data using either 2D or 3D images. Then we prefilter this database given a Commercial Reference<sup>13</sup> as a query to get all products that has a description semantically similar to it, and for deeper analysis on similarities we used Generative AI Large Language Models which can process a large amount of text and generate informative responses.

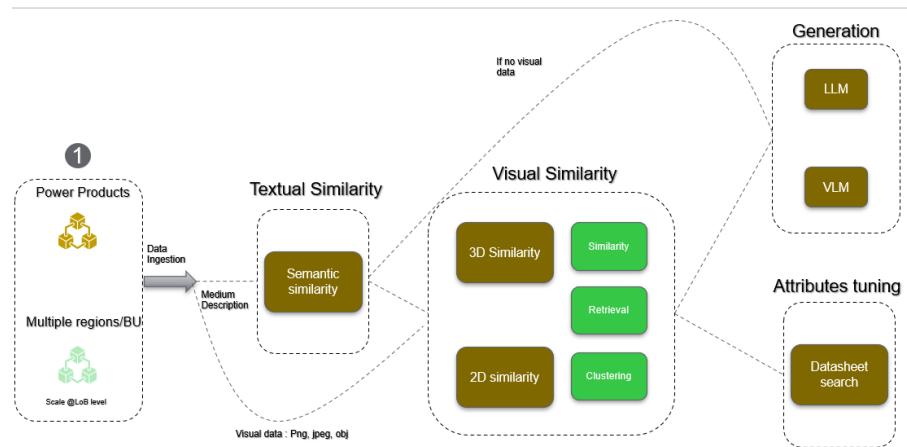


Figure 3.1: *Pipeline : from products database to visual/textual similarity and Generation*

## 4 Technical solutions studied

### 4.1 Benchmark

#### 4.1.1 Internal Benchmark

Looking at existing software : None of them is able to find similarities, analyse the differences at scale and provide value opportunities.

**CAP Tool**: is finding common parts with exact matches.

**Windchill**: is a design software, and one of its features is doing comparison one by one of subassemblies of two products (*overlapping the subassemblies and highlighting the excess volume*) : no global similarity at product level and no similarity measure.

**Marketing Software**: is not checking all informations. The most similar usecase is the Schneider search engine. After doing the semantic similarity by my own, I had the idea to try the existing search bar in Schneider ecommerce website, I discovered that it is

<sup>13</sup>standardized identifier or code assigned to a specific product or part, typically used in catalogues or databases to distinguish it from other items

not that accurate and sometimes the images of retrieved products for a given query are not similar at all, and I also found a beta feature consisting on a conversational GenAI for search assistant. To gather more information about relevant use cases, I contacted AIHub to obtain contact details. This allowed me to identify the datasets employed, target users, project objectives, and key contacts for potential collaboration. However it uses only product description and some technical attributes. . It is powered by a GPT3.5 model doing semantic search across SE database of product (PIM API), and using product description and some metadata information. Experts have to compare by themselves the visual and financial data.

#### Benchmark Evaluation Summary

Benchmarks	Multimodal <sup>a</sup>	Value Ctr. <sup>b</sup>	Generative <sup>c</sup>	<sup>d</sup> Sim. Quant.
CAP tool	No	Yes	No	No
Windchill	No	No	No	No
SE search	No	No	No	<sup>e</sup> No

<sup>a</sup>Can handle multimodal data (text, 2D, 3D, ...)

<sup>b</sup>Output: opportunity's value

<sup>c</sup>Generates content from prompts

<sup>d</sup>Quantifies the similarity between products, parts or subassemblies

<sup>e</sup>The semantic search engine can return the similarity, however the UI does not provide this information

#### 4.1.2 External Benchmark

EATON, GI are using the Dassault System software, and Siemens has developed their own. *They were developed respectively, before 2018 and 2022.*

A MOB (Make Or Buy) may be launched after this POC. In this perspective, the objective of the POC is dual : **baseline** (*for comparison to external tools*), **Informative** (*be aware of the complexity, the business value, data needed ...*)

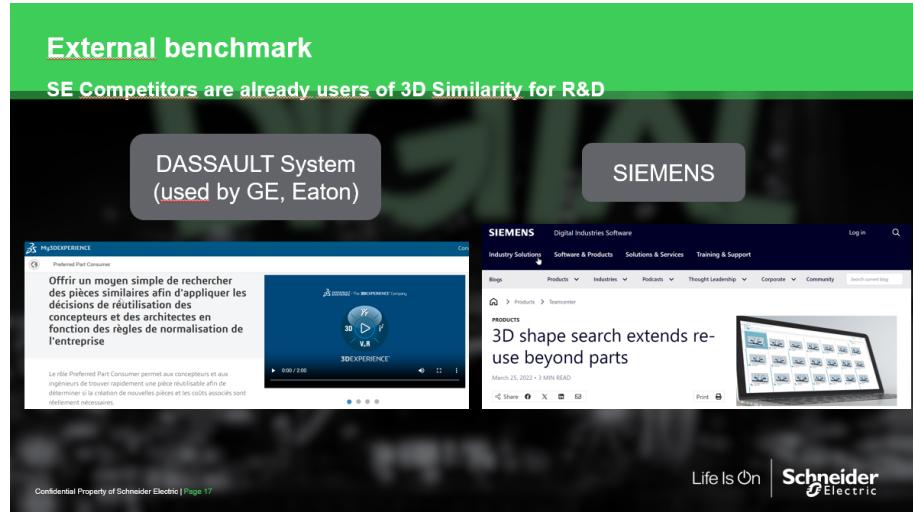


Figure 4.1: *External Benchmark*

## 4.2 Metadata VS Embeddings similarity

*We need to answer the following question : Which approach is more suited to our UC ?, is it **metadata-based search** or **principal data-based search** ?<sup>14</sup>* Metadata based approach consist on searching in a database using only metadata (keywords, title ...), this requires metadata to be very clean and enough for the search. However, when you are searching on Google, and since the huge number of websites couldn't all be categorized, a brilliant idea was to rank pages using their actual data and not only metadata (thus calling the approach principal data). And one of these methods is **semantic search** which requires **embedding the data** into a latent space and doing search between these representations.

To sum up, for similarity search, we can either use metadata (**product data sheets**) and return similar products given these attributes. Or encode the products' data (can be a joint representation of textual description and visual data) and **retrieve the nearest embeddings**.

<sup>14</sup><https://www.liip.ch/en/blog/metadata-based-search-vs-primary-data-search>

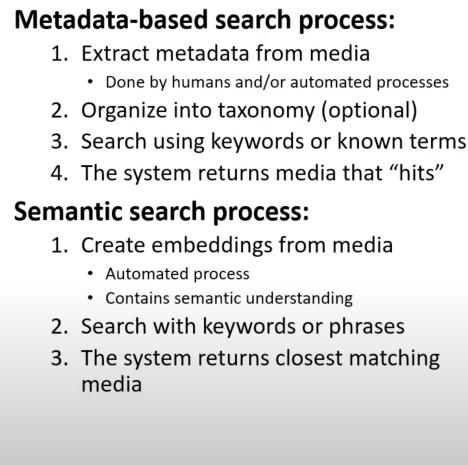


Figure 4.2: Metadata and embeddings search in video understanding from TwelveLabs

### 4.3 Internal APIs

We investigated four internal databases, to extract products’ data.

- PIM (Product Information Management): Connects images to EDMS, storing product characteristics and descriptions.
- EDMS (Enterprise Document Management System): Houses assets in a Digital Asset Management (DAM) section.
- Symphony: Offers technical attributes for customer view.
- i-Nego-RM: Manages raw materials and supplier details.

### Data and Features

1. *Visual data* accessed through PIM link to 3D files and 360° 2D images in EDMS.
2. *Supplier data*, including MPN<sup>15</sup> and technical references, are on Prism<sup>16</sup>.
3. Specifics like *generic/material types and surface treatments* are identified.

<sup>15</sup>Manufacturing Part Number

<sup>16</sup>Another internal database

## Key Elements and Connections

- HA Excel file<sup>17</sup> for raw materials and manufacturers at i-Nego-RM<sup>18</sup>.
- PIM provides an API, with Symphony detailing technical aspects.

## Conclusion and Integration Strategy

**Step 1.** PIM centralizes product data with *commercial references, descriptions, and primary attributes*.

**Step 2. Integrations:**

- Link PIM and Symphony via Commercial Reference (CR) and fetch *raw material alternatives and surface treatment*.
- Extract visual data from EDMS using CR as the common key.
- Access *RM and supplier data* from i-Nego-RM, matching on *RM Part*.

This strategy ensures a cohesive management system, integrating *product details, technical specifications, visual assets, and supply chain informations across platforms*, leveraging shared identifiers for smooth data flow.

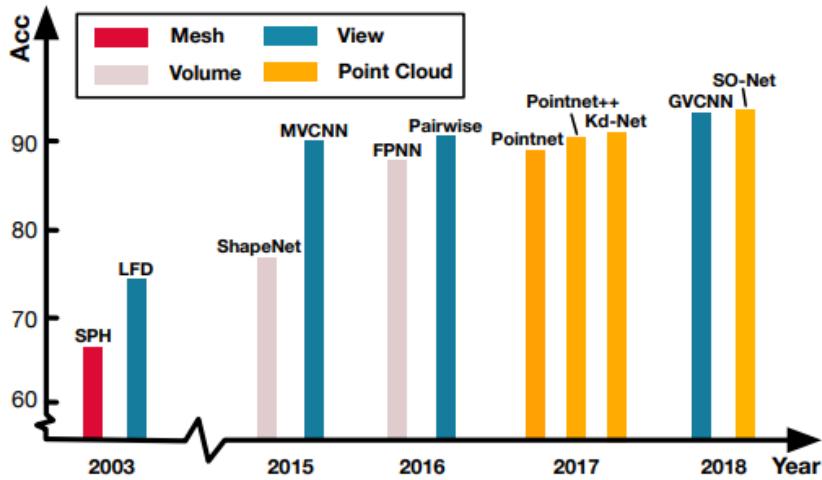
---

<sup>17</sup>An excel file managed by raw material buyers

<sup>18</sup>A database managed by buyer, containing usefull informations for negotiations and raw material prices

## 4.4 3D similarity

### 4.4.1 Data formats



**Figure 1: The developing history of 3D shape representation using different types of data.** The X-axis indicates the proposed time of each method, and the Y-axis indicates the classification accuracy.

In the field of 3D data representation, **various formats** are prevalent, serving **different purposes and domains**. The commonly used formats can be broadly categorized as follows:

**Mesh Formats:** Mesh representations describe the 3D shape using a collection of vertices, edges, and faces. Popular formats like .obj, .stl, and .ply are widely used in 3D printing, computer graphics, and visualization applications.

**Point Cloud Formats:** Point cloud data consists of a set of points in 3D space, often capturing the surface geometry of an object through 3D scanning or LiDAR techniques. Formats like .pcd, .las, and .e57 are commonly used in industries like construction, manufacturing, and autonomous driving.

**Boundary Representation (BRep):** BRep formats, such as .step and .iges, are extensively used in computer-aided design (CAD) and manufacturing. They represent 3D objects as a combination of parametric surfaces, curves, and topological information, allowing for precise geometric modeling and data exchange.

**Multi-view Images:** Representing 3D objects using multiple 2D images from different viewpoints is a common approach in computer vision and deep learning applications.

Formats like .png and .jpg can be used to store these multi-view representations.

**Voxel Grid Formats:** Voxel grids represent 3D objects as a grid of 3D pixels (voxels), with each voxel storing information about occupancy or material properties. These formats are often used in medical imaging, scientific visualization, and volumetric data analysis.

*In research settings, mesh formats, point clouds, and multi-view images* are widely used for tasks such as **3D object recognition, reconstruction, and analysis**. On the other hand, *in industrial and business domains*, formats like **BRep (STEP, IGES) and CAD-native formats (e.g., .prt, .sldprt)** are more prevalent due to their **precise geometric representation and compatibility with CAD software**.

Mesh simplification algorithms are often employed to reduce the complexity of 3D mesh models, resulting in smaller file sizes and faster rendering or processing times. These algorithms aim to approximate the original mesh while minimizing geometric errors, making them suitable for applications like real-time rendering, web-based visualization, and mobile applications.<sup>19</sup>

Additionally, conversion between different 3D data formats is sometimes necessary for interoperability and compatibility across different software tools and applications. Various libraries and tools exist to facilitate format conversion, allowing users to leverage the strengths of different representations for specific tasks or workflows.<sup>20</sup>

Visual data of products can either be 2D images or 3D (of different formats). When we need to compare visual data to find similarities given a query, we need to develop *3D similarity models/algorithms*. The choice of these algorithms depends surely on the evaluation performances, but also on the exhibited *invariances*.

We choose to preprocess the 3D data into these formats :

**xyz (cloud points)** : for distribution/shape approach,

**Triangular meshes (obj, npz, wrl, ply)** : MRG<sup>21</sup>, Harmonic Spheres and machine learning based approach, also existing frameworks are adapted to this format : open3D for example, and research on mesh data is very rich,

**Voxel grids** : for naive volume similarity and skeleton thining and segmentation approach.

**Preprocessing : Mesh simplification** Meshes may need simplification as a preprocessing step. to normalise them to the same number of faces for example.

---

<sup>19</sup>The mesh simplification algorithm we have used is

<sup>20</sup>We have used obj to ply, mesh to volume : voxelisation, stp to mesh and mesh to point cloud : mesh sampling

<sup>21</sup>Mutli-resolutional Reeb Graph

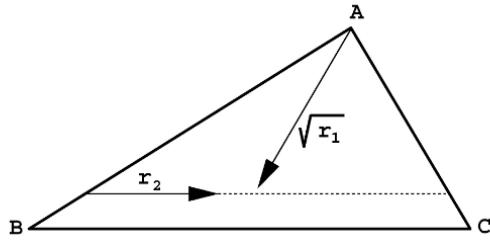


Figure 4.3: Sampling a random point in a triangle

### Preprocessing : Cloud points sampling

Although it would be simplest to sample vertices of the 3D model directly, the resulting shape distributions would be biased and sensitive to changes in tessellation. Instead, our shape functions are sampled from random points on the surface of a 3D model. Our method for generating unbiased random points with respect to the surface area of a polygonal model proceeds as follows. First, we iterate through all polygons, splitting them into triangles as necessary. Then, for each triangle, we compute its area and store it in an array along with the cumulative area of triangles visited so far. Next, we select a triangle with probability proportional to its area by generating a random number between 0 and the total cumulative area and performing a binary search on the array of cumulative areas. For each selected triangle with vertices (A, B, C), we construct a point on its surface by generating two random numbers,  $r_1$  and  $r_2$ , between 0 and 1, and evaluating the following equation:

$$P = (1 - \sqrt{r_1})A + \sqrt{r_1(1 - r_2)}B + \sqrt{r_1r_2}C \quad (1)$$

Intuitively,  $\sqrt{r_1}$  sets the percentage from vertex A to the opposing edge, while  $r_2$  represents the percentage along that edge (see Figure 4). Taking the square-root of  $r_1$  gives a uniform random point with respect to surface area.

(Taken from Shape distribution paper<sup>22</sup>)

---

<sup>22</sup>Shape distributions, Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin

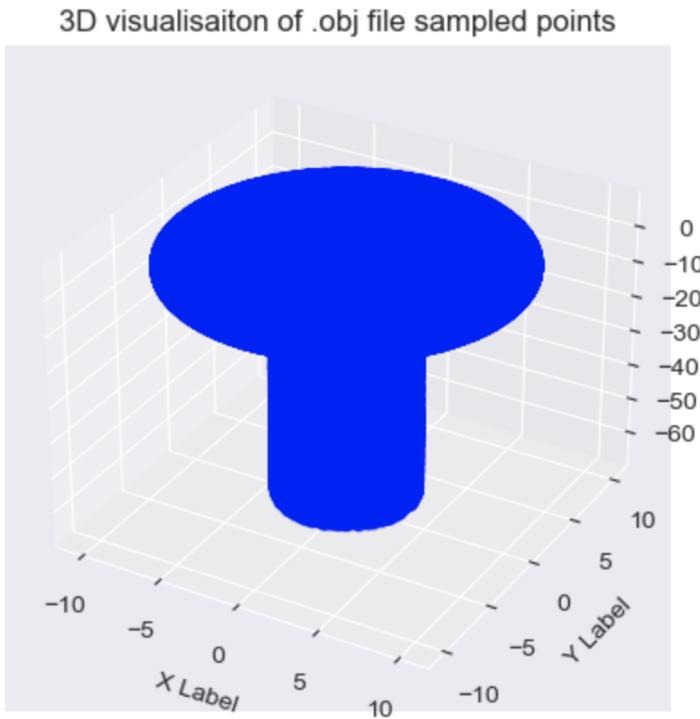


Figure 4.4: Mesh from our dataset sampled using the described approach - sampling size = 30

Let  $t$  be the number of triangles in the original mesh and  $s$  be the sampling size. The resulting cloud points will contain  $t*s$  3D points.

#### 4.4.2 Evaluation

Note that for evaluating these approaches we suggested : top ranked models based on a query<sup>23</sup> and Similarity matrix intra/inter class<sup>24</sup>. We can also evaluate the clustering performances of our search engine, using metrics such as the Dunn index<sup>25</sup>

The baseline we chose is a very simple naive 3D similarity algorithm based on voxel grids [Refer to Annexe B].

#### 4.4.3 Non-Machine Learning algorithms

The State of the art can be classified into : shape based (Shape Distribution/His-

<sup>23</sup>Retrieve k-nearest neighbors of a query using the choosen approach, and evaluate the retrieval performances

<sup>24</sup>Computes the similarities or distances between models of our dataset, and compare to a diagonal block matrix(meaning high similarity only between models of same class)

<sup>25</sup>Refer to annexe A on the information retrieval, similarity and clusterings metrics

tograms, Skeleton representation ..), **feature based** (DL embeddings, geometric features, SIFT features ..) and **topological methods** (Topological signatures, Multiresolutional Reeb Graph ..).

We have tried different algorithms, minimum one per category and the results are as below.

**Prior Knowledge** <sup>26</sup> As prior knowledge, we know that the algorithms will be used on electromechanical parts and products. For data distribution we can mention discriminative features between screws and washers for example ...

For **invariances** : translation and rotation invariance are important, while scale invariance is not necessary (we can have measurements for a better filtering)

Representation	Tr	Sc	Rot
Crease Histograms <sup>2</sup>	I	N	I
Shape Distributions <sup>3</sup>	I	N	I
Extend Gaussian Images <sup>4</sup>	I	N	N
Shape Histograms <sup>5</sup> (Shells)	N	N	I
Shape Histograms <sup>5</sup>	N	N	N
Spherical Extent Functions <sup>6</sup>	N	N	N
Wavelets <sup>7</sup>	N	N	N
Reflective Symmetry Descriptors <sup>8</sup>	N	N	N
Higher Order Moments <sup>9</sup>	N	N	N
Exponentiation EDT <sup>12</sup>	N	N	N

**Table 1:** A summary of a number of shape descriptors, showing if they are (N)ormalized or (I)nvariant to each of translation, scale and rotation.

**Dataset** SE <sup>27</sup>and opensource non SE existing datasets. for manufacturing parts, we have used two reliable opensource non SE datasets : ModelNet40 and a dataset of fasteners (both in obj format).

<sup>26</sup>Invariance and data distribution.

<sup>27</sup>Schneider Electric

**Shape based** Here we will introduce two papers, both suggests to find the 3D similarities using a shape based approach. The first one from a shape distribution<sup>28</sup> perspective and the second using skeleton shape representation.<sup>29</sup>

**Shape distribution** We implemented the shape distribution paper, which consist on generating shape distributions/histograms of a 3D model based on a shape function (A3, D2), then computing the distance between two distributions for a retrieval use.

Plotting the shape distributions generated based on A3 shape function, proves that it is able to discriminate between some different classes of objects while having similar distributions intra class.

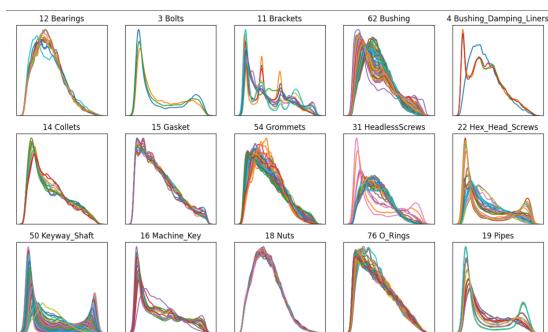


Figure 4.5: *Shape Distribution using A3 shape function on our dataset*

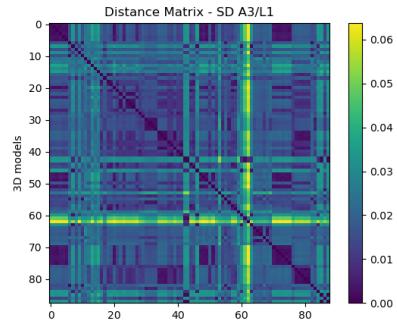


Figure 4.6: *Distance matrix of Shape distribution representation between models in the dataset - A3/L1*

While using D2 shape function, we got non-discriminative shape distributions on our dataset, which is confirmed by the distance matrix obtained using L1 distance.

<sup>28</sup>Shape distributions, Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin.

<sup>29</sup>Refer to annexe C

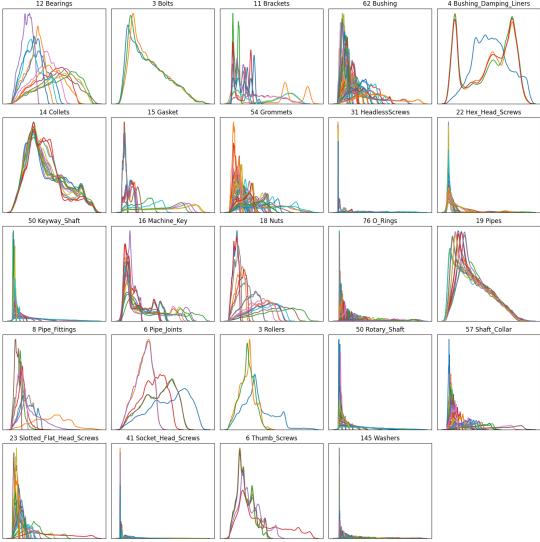


Figure 4.7: Shape Distribution using D2 shape function on our dataset

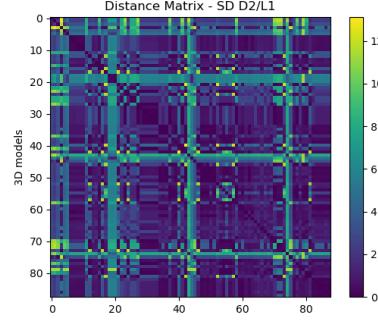


Figure 4.8: Distance matrix of Shape distribution representation between models in the dataset - D2/L1

**Geometric features based** For Geometric features extraction, we can mention circularity, eccentricity, or algebraic moments. Or SIFT features.

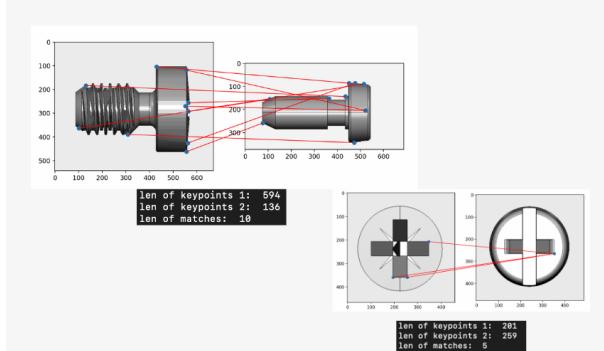


Figure 4.9: SIFT extracted features

For this experiment, we used multi-view data of few objects, and extracted SIFT features, then we tried matching them. The idea behind it is to evaluate the quality of the extracted features, thus conclude if it is discriminative enough to be used for retrieving similarities intra and inter class of objects. The results showed that this method is not adapted because the features extracted are not discriminative enough between objects which is proved by the bad quality of the matched points.

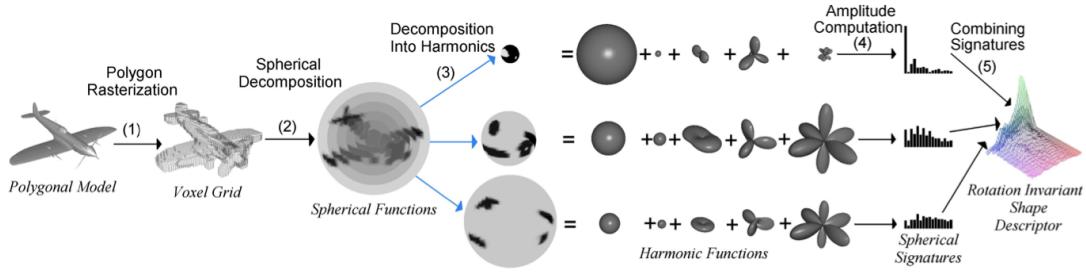


Figure 4.10: Harmonic Spheres algorithm

**Harmonic Spheres Descriptor** Developing a rotation invariant descriptor is one of our constraints. Thus we implemented Harmonic Spheres Descriptor<sup>30</sup> which computes a 2D descriptor of a 3D model of shape  $R^*(m+1)$  where  $R$  is the number of spheres over which the 3D Gaussian-EDT is to be sampled and  $m$  the number of spherical frequencies that are used to define the descriptor.

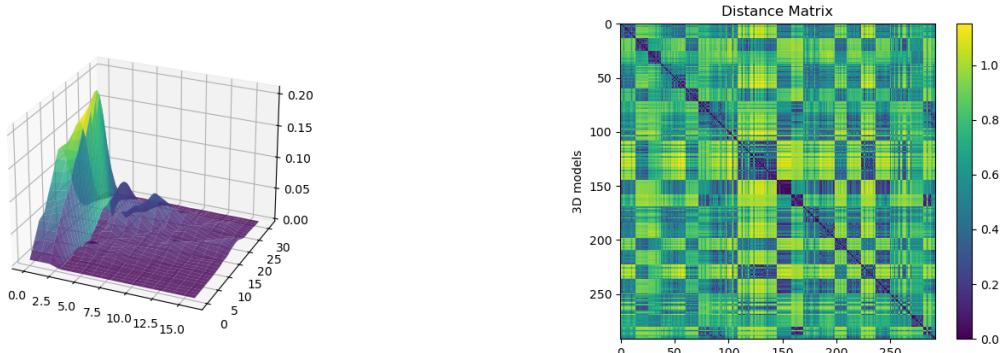


Figure 4.11: HS Descriptor from our dataset

Figure 4.12: Distance matrix - HS Descriptor

The paper suggests an indexing and matching procedure, however for our experiment we computed the L2 euclidian norm between two descriptors and retrieved the nearest representations for a given query. Here is the distance matrix Figure 4.19 we computed from our dataset:

<sup>30</sup><https://www.cs.jhu.edu/~misha/MyPapers/SGP03.pdf> and <https://www.cs.princeton.edu/~funk/tog03.pdf>

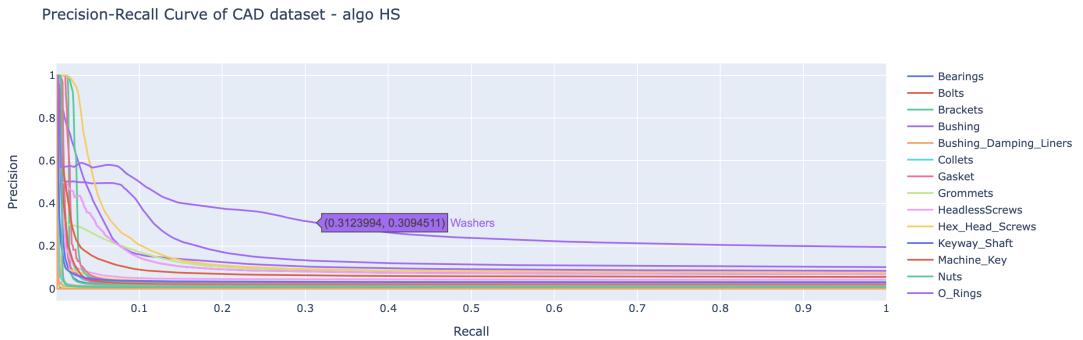


Figure 4.13: Precision-recall curve *Harmonic Spheres - all classes*

**Topology based** Topological methods are algorithms that calculate the similarity between two 3D images based on their respective topological representations usually encoded in a graph. Here we will talk about the MRG (Multi-resolutional Reeb Graph).

$$Sim(obj1, obj2) = Sim(MRG1, MRG2)$$

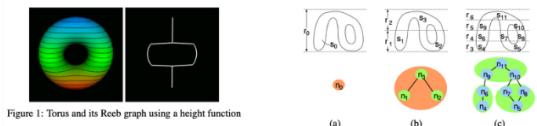


Figure 4.14: *Multi-resolutional Reeb Graph*

Query Model	Most Similar Model	Similarity Value
models/plastic_2.wrl	models/tolerie.wrl	0.4343532348879353
models/plastic_2.wrl	models/vis_1.wrl	0.4148966833710717
models/plastic_2.wrl	models/vis_2.wrl	0.3395055257257435
models/tolerie.wrl	models/vis_1.wrl	0.6423869367467379
models/tolerie.wrl	models/vis_2.wrl	0.6076140709266459
models/vis_1.wrl	models/tolerie.wrl	0.6423869367467379
models/vis_1.wrl	models/vis_2.wrl	0.5939105435893113
models/vis_2.wrl	models/tolerie.wrl	0.6076140709266459
models/vis_2.wrl	models/vis_1.wrl	0.5939105435893113

Figure 4.15: *Results of Multi-resolution Analysis*

The intuition behind using a topological similarity is its theoretical ability to discriminate very well between certain class of objects topologically, such as screws, washers and electrical outlets : for example in a screw we have one connected component, while in a washer, the hole in the middle, creates other components (the number of connected components will depend on the resolution) ...

However, the conclusion is that the complexity in time is very high, and the results for our dataset (see figure 4.15) are not as promising as the ones shown in the paper.<sup>31</sup>

<sup>31</sup>Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes

#### 4.4.4 Machine Learning algorithms

2D Convolutional Neural Networks (CNNs) excel at extracting meaningful features from 2D images due to their hierarchical structure and translation invariance<sup>32</sup>. Pooling layers, a key component of CNNs, encode this invariance by downsampling feature maps, making them robust to small shifts in the input image. This property aligns well with the geometric nature of visual data, positioning CNNs as powerful tools for representation learning in geometric deep learning. While 2D CNNs are effective for 2D images, capturing 3D similarity requires different approaches. Contrastive learning or combining a strong embedder with a classifier can be explored to learn meaningful 3D representations. Additionally, ML models offer advantages like interpretability through activation or attention maps, providing insights into learned visual discriminators. Continuous learning enables adaptation to new objects, enhancing the model's generalisation ability for specific use cases.

To address the challenge of 3D similarity search, we adopted a two-stage approach. Initially, a 3D CNN-based architecture was pre-trained on a downstream classification task involving 40 categories. Subsequently, the classifier was removed to isolate the learned embeddings, which were then visualized to assess their quality as a foundation for a similarity search engine operating on 3D simplified triangular mesh data. The promising results from this evaluation suggest that the pre-trained embeddings provide a solid starting point.

To further refine the embeddings for similarity search, a contrastive learning stage can be introduced. By employing the triplet loss, the model can be trained to embed similar samples closer together in the latent space while pushing dissimilar samples apart. This process is expected to enhance the discriminative power of the embeddings for effective similarity retrieval.

Beyond similarity search, the trained model exhibits versatility. Its classification capabilities can be leveraged to relabel missing or noisy data within the dataset. Additionally, by integrating vision-language models, textual descriptions can be generated for data points lacking such information. The effectiveness of these downstream tasks is contingent upon the extent of missing data, which would require a quantitative analysis to assess the potential impact.

**Demonstration** We visualised the precision-recall curve on MeshNet [Figure 4.17], and the result shows that Washers have higher performances, however for all classes of objects the precision performances dump very rapidly with respect to recall. Refer to benchmark

---

<sup>32</sup>Geometric Deep Learning

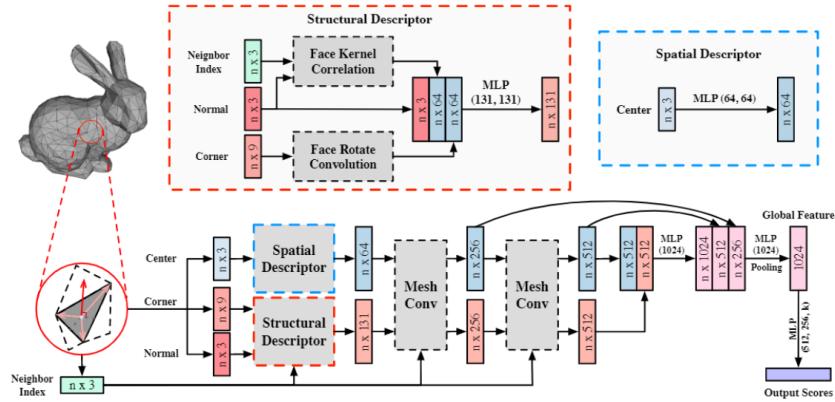


Figure 4.16: *MeshNet* architecture

with other approaches and to the conclusion.

#### Precision-Recall Curve of CAD dataset - algo MeshNet

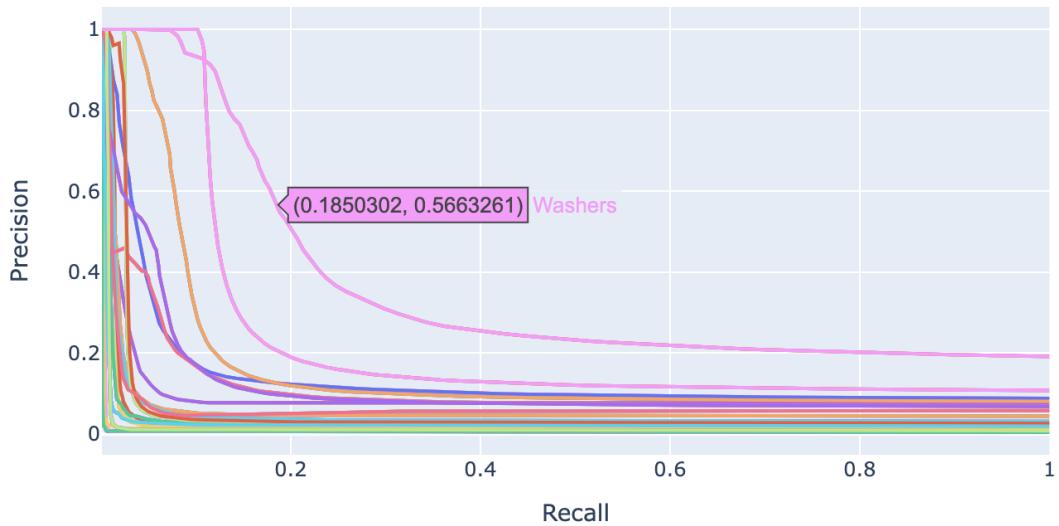


Figure 4.17: Precision-recall curve - *MeshNet* on our dataset of fasteners

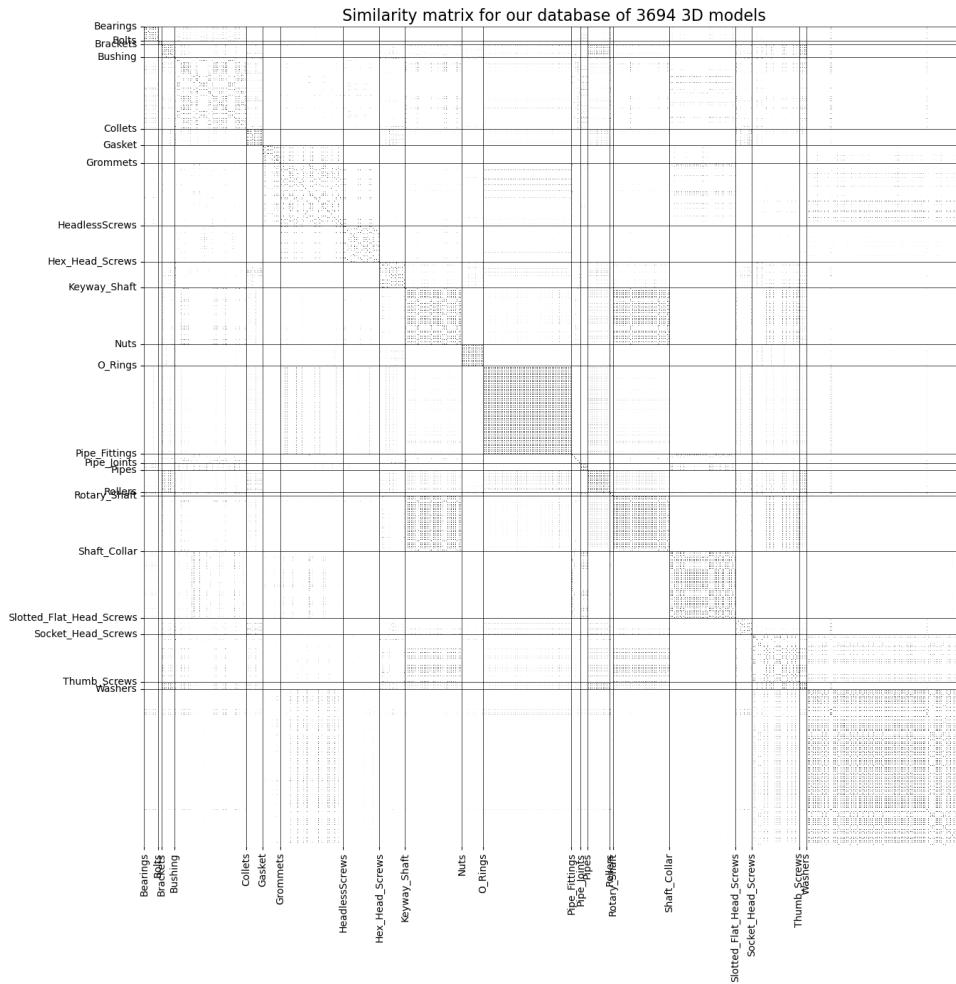


Figure 4.18: *similarity matrix inter/intra class on CAD fasteners Dataset*

The similarity matrix shows a high degree of intra-class similarity, meaning that models within the same class are generally quite similar to each other. This is evident by the darker blocks along the diagonal of the matrix, which correspond to comparisons between models of the same type. However, the matrix also shows some degree of inter-class similarity, particularly between certain pairs of classes. For example, there is some overlap in the similarity between "Rotary Shaft" and "Keyway Shaft", which suggests that the algorithm may struggle to distinguish between these two classes at times. Overall, the matrix suggests that the algorithm is capable of capturing some degree of geometric similarity between 3D models, but it may still be susceptible to confusion between certain classes of objects.

We can also visualise the similarity matrix for the pretraining dataset. Since the model has never seen fasteners before, extracting some features may be hard. Especially some fasteners look similar (grommets and washers). We will then evaluate the retrieval per-

formances of a ML model pretrained on the same concepts, on the classification task.

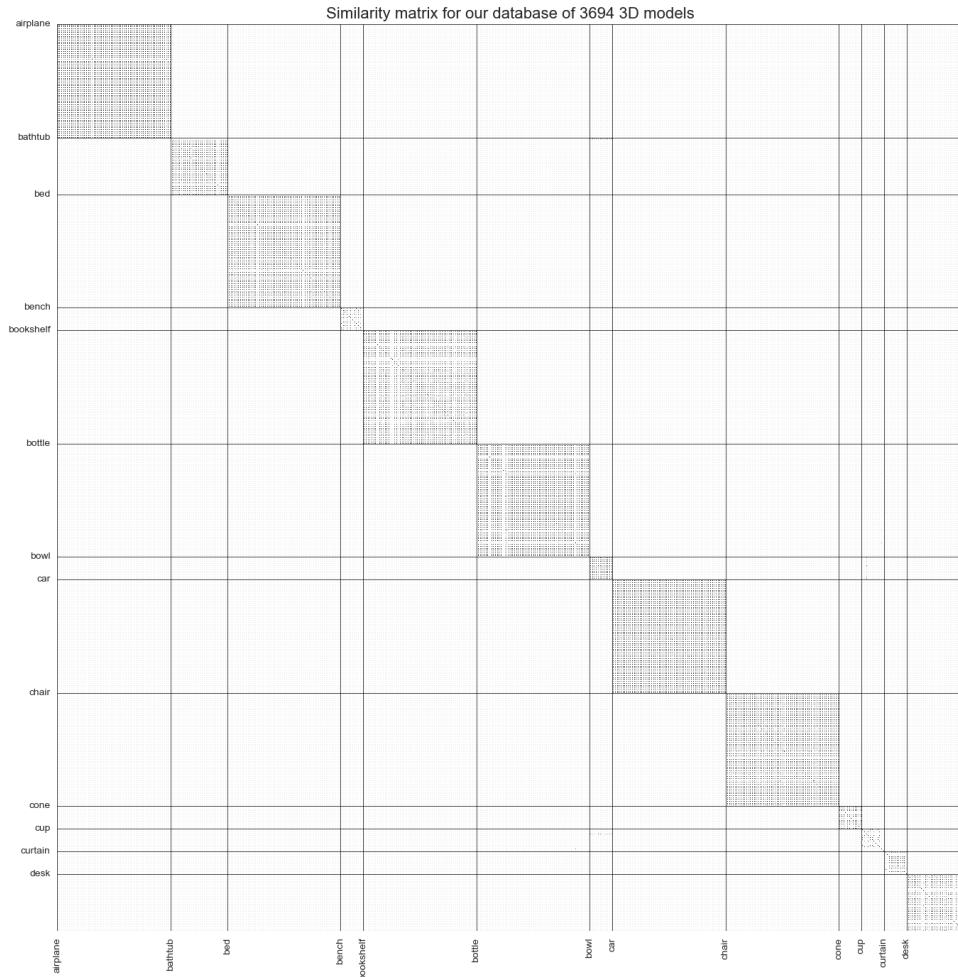


Figure 4.19: *similarity matrix inter/intra class on ModelNet40 Dataset*

We conclude that the similarity retrievals of the MeshNet model are much better when pretrained on a classification task on the same concepts : ModelNet40 dataset. This helps the model extract useful features to separate the different concepts.

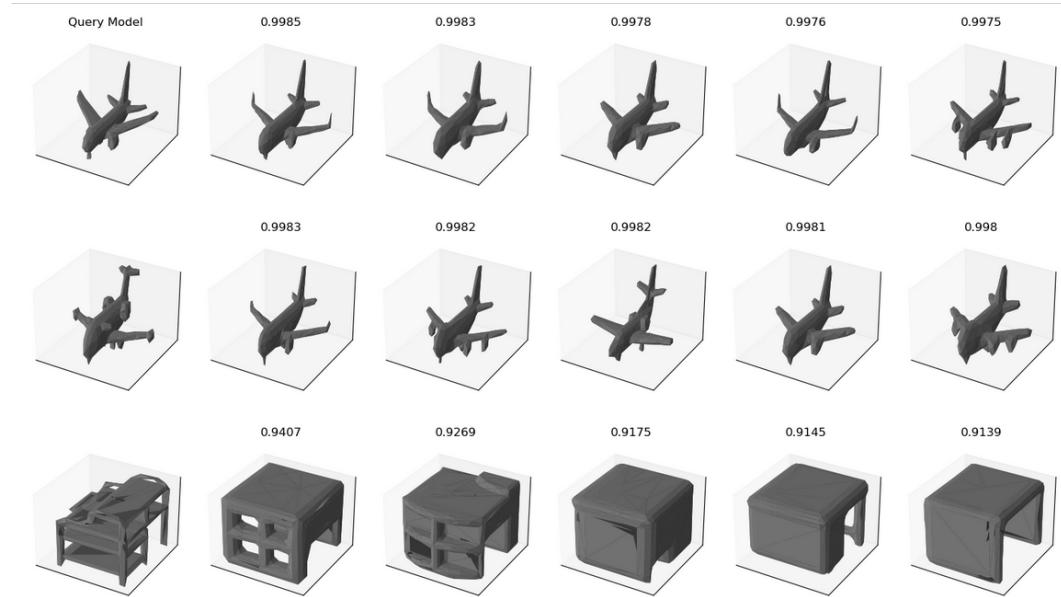


Figure 4.20: *Top-ranked model based on a query*

The top-ranked evaluation, demonstrates strong retrieval performances on the modelNet40 dataset.

**Data** A custom dataset class was implemented to efficiently store and access essential mesh attributes such as label, file path, vertex coordinates, neighbor information, and centroids.

The original .obj data was preprocessed in two steps: *mesh simplification to standardize the face count*, followed by *conversion to the .npz format for optimized model ingestion*.

**The .npz training data:** In our experiment, we converted our dataset (initially containing meshes in obj format) to npz format. NPZ files offer a trade-off between efficiency and memory usage. On one hand, they efficiently store multiple NumPy arrays in a single compressed file, and they only load the specific data required when you access individual arrays. This lazy loading behavior is beneficial for large datasets. *However, NPZ files have a limitation: they cannot be directly mapped to memory, which can hinder performance when dealing with extremely large arrays. In such cases, alternative file formats that support memory mapping, like HDF5 might be more suitable.*

The attributes we stored in npz files are : `faces` (`center`, `corners` and `face normal`), `neighbors`

**Data augmentation** During the training, we augment the data by jittering the positions of vertices by a Gaussian noise with zero mean and 0.01 standard deviation.

**Opensource Datasets** Here's a table formatted for comparing open-source 3D CAD datasets with these columns: Dataset Name, Size, Available Formats, and Concepts.

Table 4.1: *Comparison of Open-Source CAD Datasets*

Dataset	Size	Formats	Concepts
ShapeNetCore <sup>[?]</sup>	51.300	Meshes	57 common object categories
Thingi10K <sup>[?]</sup>	10K	Meshes	Wide range
ABC Dataset <sup>[?]</sup>	1M	OBJ, STEP, STL	Most models are mechanical parts
ModelNet40 <sup>[?]</sup>	12.311	OBJ <sup>33</sup> , Point cloud <sup>34</sup>	airplane, car, plant, lamp ...
DeepCAD	180K	OBJ, STEP	Electromechanical parts

**Ablation study** refer to MeshNet paper<sup>35\*</sup>

**Visualisation on tensorflow projector** This visualisation is done using **Tensorflow Embedding Projector**, for a given file of embeddings, it computes a dimension reduction (PCA, T-SNE ...) for a 3D visualisation of the embeddings and retrieves the nearest neighbors of a given query in the original space. The visualisation is very good to assess the clustering performances and the quality of the embeddings in terms of similarity and retrieval.

<sup>35</sup>MeshNet: Mesh Neural Network for 3D Shape Representation, Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, Yue Gao.

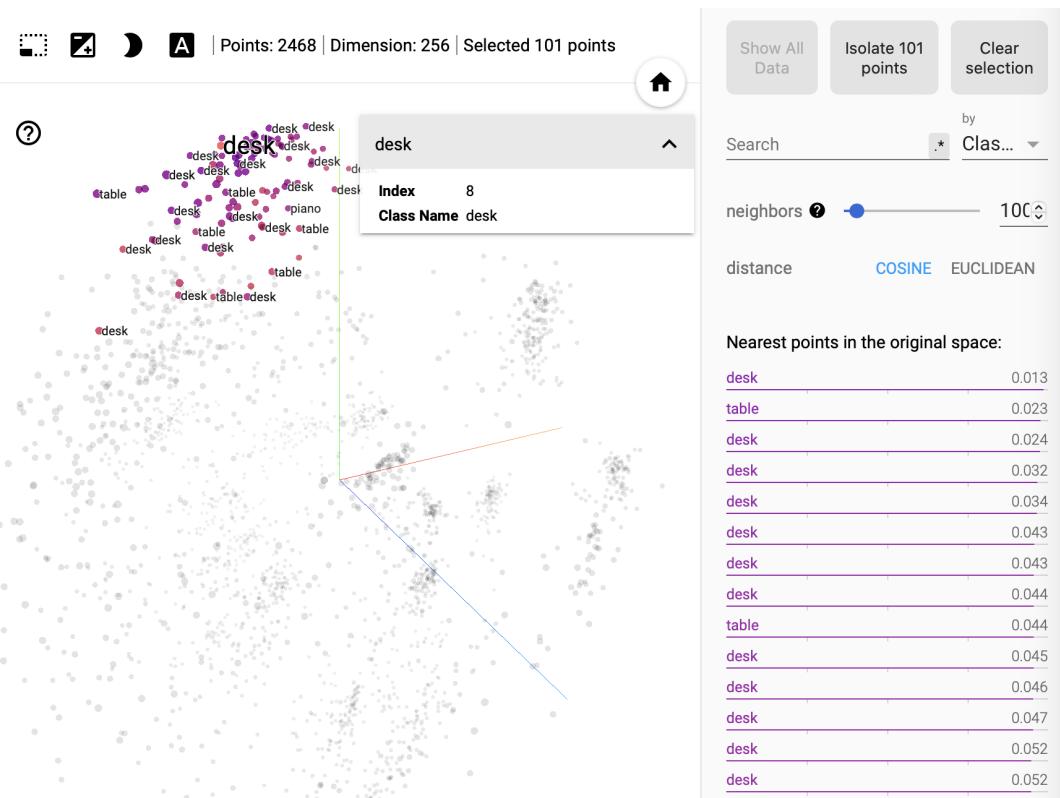


Figure 4.21: Visualisation of MeshNet embeddings in a 3D space using PCA, and retrieval performances - Embedding Projector from Tensorflow

#### 4.4.5 Benchmark

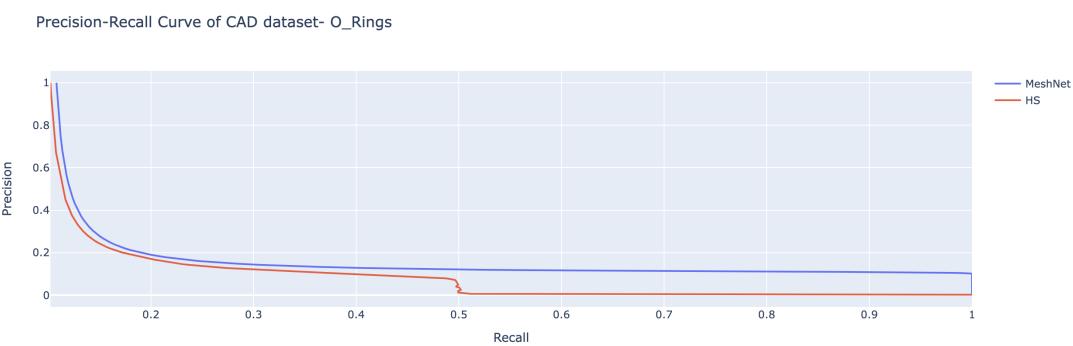


Figure 4.22: Slightly better precision-recall curve for MeshNet - O-Rings class

Precision-Recall Curve of CAD dataset- Washers

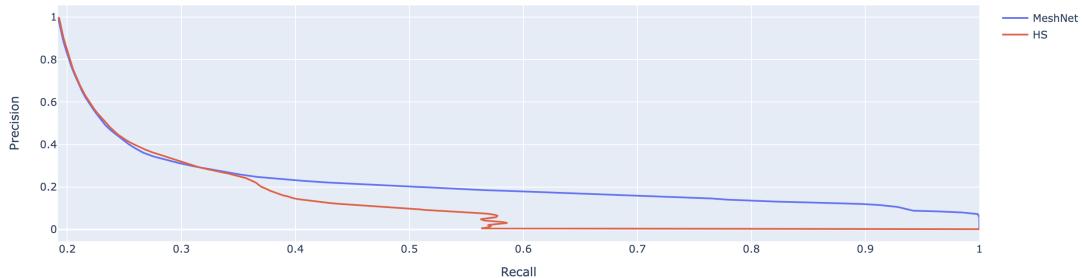


Figure 4.23: Slightly better precision-recall curve for MeshNet - Washers class

We then tried to balance our dataset, since it theoretically influences the precision-recall results, and in our dataset there are some very similar classes.

Precision-Recall Curve of CAD dataset - algo HS

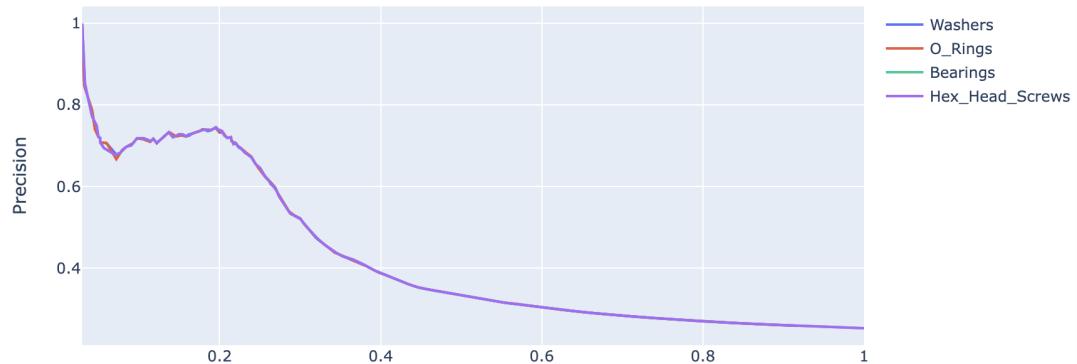


Figure 4.24: Precision-recall curve for HS algorithm - balanced dataset with fewer distinct class

## 4.5 Textual semantic similarity

This block serves as a prefilter, generating a shortlist of potentially similar objects based on their textual descriptions from a vast dataset.

Product and part descriptions within this dataset primarily consist of *labels and a few technical attributes* (such as measurements, voltage, or adjustability). Given the marketing-oriented nature of the data and potential regional or company-specific variations in terminology, a semantic representation becomes crucial. Semantic similarity allows us to capture the underlying meaning, effectively treating synonyms as equivalents.

To quantify textual semantic similarity, we employed semantic textual embedding models followed by cosine similarity calculations between sentence representations. The

chosen sentence length [*approximately ten words*], was determined by the available data and the information richness of the descriptions. It's worth noting that the optimal context length might influence the performance of the embedding model.

Description Size	Data Availability	Informativeness <sup>a</sup>	Context Length <sup>b</sup>
Short	NAA <sup>c</sup>	Concise but possibly incomplete	4
Medium	A <sup>d</sup>	Balances detail and brevity	15
Long	NAA	Detailed, risk of redundancy	50

<sup>a</sup>Quantity of information describing the product/part

<sup>b</sup>Number of tokens (we will consider number of words)

<sup>c</sup>Not always available

<sup>d</sup>available

**Tools Used** Hugging Face Transformers has emerged as a leading platform for natural language processing (NLP), offering pre-trained models and tools that accelerate development. These transformer-based models, such as BERT, RoBERTa, and GPT, have demonstrated exceptional performance in various NLP tasks, including semantic similarity. Key advantages of using Hugging Face Transformers:

**Pre-trained models:** These models are trained on massive datasets, providing a strong foundation for downstream tasks.

**Transfer learning:** Fine-tuning pre-trained models on specific datasets often leads to rapid performance improvements.

**Community and resources:** The Hugging Face community offers extensive support, tutorials, and datasets.

**State-of-the-art performance:** Transformer-based models consistently achieve top results on benchmark tasks.

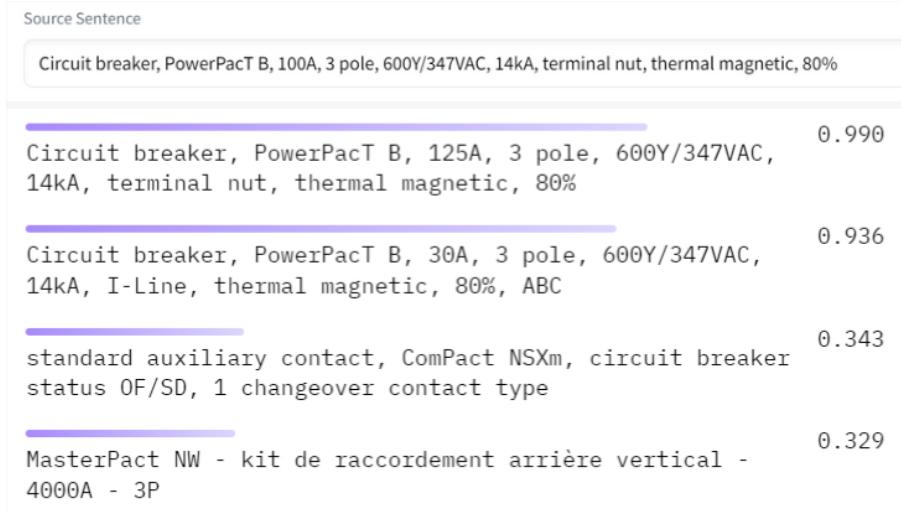


Figure 4.25: Semantic textual similarity on a subset of our dataset

As seen in *Figure 4.25*, the semantic model is retrieving similar product descriptions based on both *product class* (*Circuit Breaker*) and *technical attributes* (100A, 3 pole ...).

## 4.6 Generation

The generation block aims to generate insights based on the user prompt, processing both visual and textual data, and acting on a list of potentially similar objects returned by the semantic filter. Reason enough to introduce both Large Language Models (LLM) and Vision Language Models (VLM) and test how we can benefit from them.

Our experiment was based on a list of potentially similar objects, with few differences in their technical attributes. We aimed to test how good are LLMs dealing with a list of raw product descriptions, finding all differences, and generating informative call-to-action (Highlighted in red on the figure).

Prompt	Generated Response
<p>You are a Quality value engineer, you are working for a company that has a lot of products. Your job is to find opportunities of standardisation or value engineering in your product portfolio.</p> <p>[Non-adjustable time delay relay for voltage release MN - 100/130 V AC/DC . Adjustable time delay relay for voltage release MN - 48/60 V AC/DC.</p> <p>Non-adjustable time delay relay - for MN undervoltage release - 100/130 V AC/DC.</p> <p>Non-adjustable time delay relay - for MN undervoltage release - 200/250 V AC/DC.</p> <p>Adjustable time delay relay - for MN under voltage release - 100/130V AC/DC - sp.</p> <p>Non-adjustable time delay relay - MN undervoltage release - 200/250 V AC/DC - sp.</p> <p>Adjustable time delay relay - for MN under voltage release - 380/480 V AC - sp.</p> <p>Adjustable time delay relay - for MN undervoltage release - 100/130 V AC/DC.</p> <p>Adjustable time delay relay - for MN undervoltage release - 200/250 V AC/DC.</p> <p>Adjustable time delay relay - for MN undervoltage release - 48/60 V AC/DC.</p> <p>Adjustable time delay relay - for MN undervoltage release - 200/250 V AC/DC - sp]</p>	<p>As a Quality Value Engineer, my approach to finding opportunities for standardisation and value engineering in the given product portfolio would involve the following steps:</p> <p>1 - <b>Product Categorization:</b> First, categorize the products based on their primary functions and specifications:</p> <ul style="list-style-type: none"> <li>*Non-adjustable relays (fixed voltage thresholds)</li> <li>*Adjustable relays (variable voltage thresholds)</li> <li>*Undervoltage release specific relays</li> <li>*Voltage ranges (100/130 V AC/DC, 48/60 V AC/DC, 200/250 V AC/DC, 380/480 V AC)</li> <li>*Special versions (indicated by "sp")</li> </ul> <p>2 - <b>Commonality Analysis:</b> Identify shared components, technologies, and design features across all products. For example, many use similar time delay mechanisms and operate within the same voltage categories. Look for common platforms or base designs that can be utilised across different models.</p> <p>3 - <b>Standardisation Opportunities:</b></p> <p>Voltage Ranges: Consolidate offerings by creating a modular design that supports multiple voltage ranges with interchangeable or easily configurable modules. This could reduce the number of unique SKUs and simplify production.</p> <p>4 - <b>Adjustability:</b> Develop a universal adjustable mechanism that can cover all identified voltage ranges, reducing the need for separate adjustable and non-adjustable models.</p> <p>- <b>Special Features:</b> Standardise special features like "sp" into modular add-ons, making them optional across all base models. ...</p>

Figure 4.26: Prompt engineering experiment

#### 4.6.1 LLMs for analyzing products descriptions dissimilarities at scale

Large Language Models are very good at processing large textual context, and generating informative responses.



Figure 4.27: *LLM pipeline*

Starting with a big similar product list, we are now checking all the dissimilarities through LLM Generated response. This unlocks analysing products descriptions dissimilarities at scale

**Tools Used and why** At the time of writing this report, Qwen models are ranked among the top large models on the leaderboard. Additionally, since the data we have used so far is not private, we did not face any data privacy issues. Otherwise, we could have used a dedicated cloud instance for Schneider's internal use to preserve data privacy, or employed internal LLMs or an on-device local LLM/SLM.

#### 4.6.2 VLMs for analyzing products images dissimilarities at scale

Using product descriptions was only the beginning of our experience, we then used 2D images on VLM, which are GenAI models able to handle both text and images. We concluded that The textual comparaison can be completed with product images, since the models are able to describe the products images efficiently and even propose the minimal changes to converge two designs into one.

Some usefull prompts we have tried are : «Describe the product given its image» «Given the two images I provided you, what are the differences. What are the minimal changes to reduce design cost?» «Whom company is this product ?»

**Tools Used and why :** Vision arena is a huggingface public space with a research purpose. It offers a wide range of VLMs for free and the UI enables comparing their

*outputs and choosing the most adequate ones.*

## 4.7 Conclusion

*Existing tools often fall short in efficiently identifying valuable Quality Value Engineering (QVE) opportunities.* To address this, we developed a comprehensive framework encompassing 3D and textual similarity retrieval, as well as Large Language Model (LLM) and Large Vision Model (LVM) generation. While our initial focus was on 3D similarity and retrieval, future research will explore 3D-to-3D translation and text-conditioned 3D generation for design reuse.

Our experiments on an electromechanical parts dataset demonstrated the effectiveness of our search engines in distinguishing between objects within and across classes. However, the mAP results were lower than 0.3, likely due to the specialized nature of our fasteners dataset, which includes highly similar object classes. From a user's perspective, the goal is not necessarily to recall all objects within a specific class but to **cluster similar ones based on the design or/and description, even if they belong to different classes.**

Regarding generation capabilities, we successfully demonstrated rapid generation of informative insights from our product database, which combines textual and 2D visual data. This empowers users to navigate the extensive database efficiently while tailoring their search queries. In practice, we found this feature to be valuable for **competitive product comparisons.**<sup>36</sup>

## 4.8 Limitations and further studies

**Mesh Conversion and Simplification:** While the conversion of point clouds to meshes and subsequent simplification were essential preprocessing steps, a deeper evaluation of these algorithms could potentially identify performance bottlenecks. It's worth noting that mesh simplification, although generally effective, can introduce artifacts, particularly in detailed components like grommets and washers. Given the satisfactory performance of the ML models in this study, a more in-depth analysis of these preprocessing steps was deferred.

---

<sup>36</sup>Comparing Schneider products to similar ones from the competitors

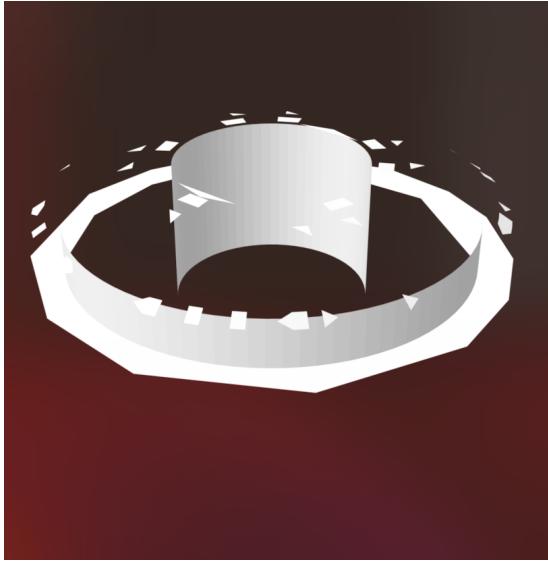


Figure 4.28: Illustration of a badly simplified triangular mesh from a Grommet object

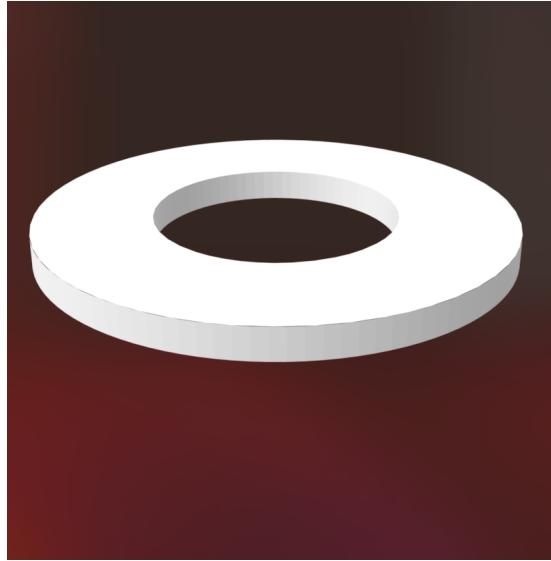


Figure 4.29: Illustration of a well simplified triangular mesh from a washer object

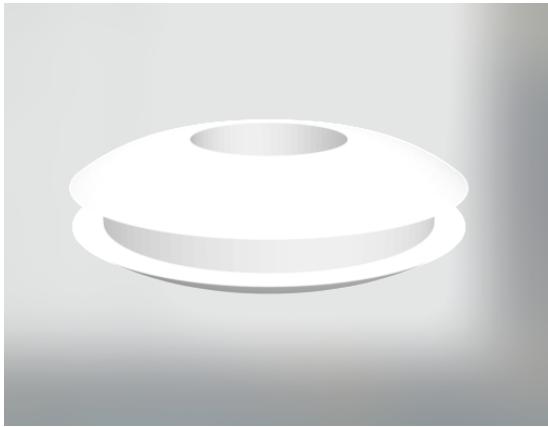


Figure 4.30: Illustration of the original triangular mesh from a Grommet object



Figure 4.31: Illustration of the original triangular mesh from a washer object

**Prompt Engineering and Language Models:** The preliminary exploration of prompt engineering techniques demonstrated the potential of LLMs and VLMs in enhancing the system's capabilities. However, more sophisticated prompting strategies, such as the use of chain-of-thought reasoning, reinforcement learning from human feedback (RLHF), and Mixture-of-Experts (MoE) architectures, warrant further investigation. Additionally, the evaluation of hallucination risks within the generative components of the system is crucial for ensuring AI safety and reliability.

**Data Quality and Monitoring:** The presence of unclean data, including incorrect descriptions, labels, and missing information, poses a significant challenge. To mitigate the "garbage in, garbage out" problem, a dedicated data and ML monitoring pipeline should be implemented to validate data quality prior to ingestion into the core pipeline. This will be a critical area for future research.

## 5 Status of the advancement of the project and critical analysis

We successfully produced high-quality content, tested and evaluated different approaches. The presentation to the Global Procurement Committee generated significant interest, particularly from the PP and HD LOBs as well as Quality Value Engineers. We were able to define several concrete and impactful use cases.

Personally, I am satisfied with overcoming the challenges encountered during the project, which made this experience particularly enriching. While we achieved satisfactory results, I would be curious to explore multimodal models for similarity in more depth and compare them to our cascade approach.

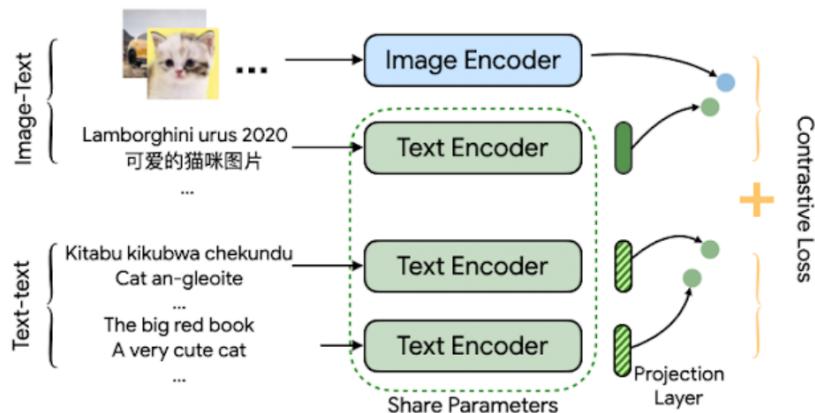


Figure 5.1: Multimodal embeddings

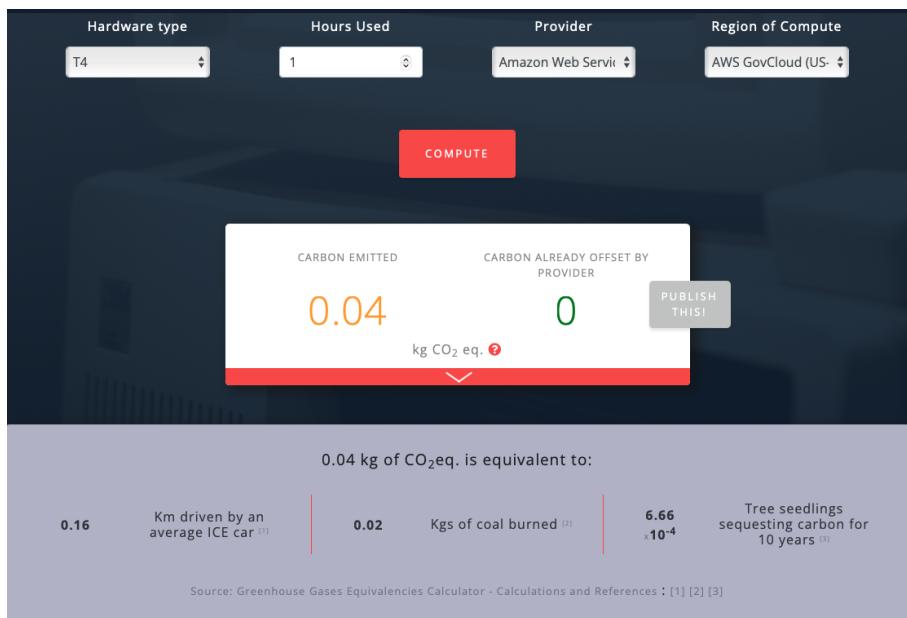
## 6 Environmental and social footprint

This project is closely linked to critical AI concerns such as bias, safety, and the environmental impact associated with AI and high-performance computing. However,

our proof-of-concept demonstrated that it's possible to achieve promising results while minimizing resource consumption. By leveraging Hugging Face Spaces, a pre-trained 3D ML model, and GPU processing, we significantly reduced our computational footprint.

At the end of the experiment, we saw that it is possible to display CO<sub>2</sub> emission of the huggingface spaces/models, see their official page <sup>37</sup>

Since for ML model, we built a private huggingface space running on a T4 medium GPU for about 1 hour, and referring to a CO<sub>2</sub> emission calculator <sup>38</sup>, we got this result :



To estimate the CO<sub>2</sub> emissions related to the equipment I use, we convert the electricity consumption into CO<sub>2</sub> emissions according to GreenIT <sup>39</sup>, which states that 1 kWh corresponds to 100 g CO<sub>2</sub>e. Daily, I use a laptop and a monitor, except on remote work days when I do not use an independent monitor and work solely with the laptop. This represents 40 hours (35 working hours and 5 hours of breaks where the computer remains on) per week of laptop usage, totaling 960 hours over the entire internship, and 576 hours for the monitor, excluding remote work days (960h - 48d x 8h). The energy consumption of the laptop and monitor is 65W and 20W respectively, which results in a total of 7.392 kg CO<sub>2</sub>e.<sup>40</sup>

<sup>37</sup><https://huggingface.co/docs/hub/model-cards-co2>

<sup>38</sup><https://mlco2.github.io/impact/compute>

<sup>39</sup><https://www.greenit.fr/20de-co2-degage-un-1-kwh-electrique/>

<sup>40</sup><https://avenirclimatique.empreinte-carbone-trajet/>

## 7 References

- **A Retrieval Method for 3D CAD Assembly Models Using 3D Radon**, Kaoru Katayama.
- **A Survey of Moment-Based Techniques for Unoccluded Object Representation and Recognition**, Richard J. Prokop and Anthony P. Reeves.
- **Content based CAD assembly model retrieval**, Katia Lupinetti, Jean-philippe Pernot, Marina monti, Franca Giannini.
- **Content based CAD assembly model retrieval**, Katia Lupinetti, Jean-philippe Pernot, Marina monti, Franca Giannini.
- **Distinctive Image Features for Scale-Invariant Keypoints**, David G. Lowe.
- **Hamilton-Jacobi Skeletons**, Kaleem Siddiqi, Sylvain Bouix, Allen Tanenbaum, Steven W. Zucker.
- **A database system of mechanical components based on geometric and topological similarity. Part I: representation**, Mohamed El-Mehalawi, R. Allen Miller.
- **A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment**, Mohamed El-Mehalawi, R. Allen Miller.
- **Indexing and Matching for View-Based 3-D Object Recognition Using Shock Graphs**, Diego Alejandro Macrini.
- **Indexing Hierarchical Structures Using Graph Spectra**, Ali Shokoufandeh, Member, IEEE; Diego Macrini, Student Member, IEEE Computer Society; Sven Dickinson, Member, IEEE; Kaleem Siddiqi; and Steven W. Zucker, Fellow, IEEE.
- **Matching 3D Models with Shape Distributions**, Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin.
- **Measuring 3D Shape Similarity by Matching the Medial Scaffolds**, Ming-Ching Chang, Benjamin B. Kimia.
- **Similarity comparison of mechanical parts to reuse existing designs**, Taesik Hong\*, Kunwoo Lee, Sungchan Kim.
- **Recognition of Shapes by Editing Their Shock Graphs**, Thomas B. Sebastian, Member, IEEE, Philip N. Klein, and Benjamin B. Kimia.
- **Retrieving Articulated 3-D Models Using Medial Surfaces and Their Graph Spectra**, Juan Zhang, Kaleem Siddiqi, Diego Macrini, Ali Shokoufandeh, and Sven Dickinson.
- **Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors**, Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz.

- **Shape distributions**, Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin.
- **Shape retrieval using 3D Zernike descriptors**, Marcin Novotni, Reinhard Klein, Institute of Computer Science II, University of Bonn, Romerstr. 164, D-53117 Bonn, Germany.
- **Shock Graphs and Shape Matching**, Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson and Steven W. Zucker.
- **Similarity and scaling assessments of mechanical parts using adjacency relation matrices**, Adem Çiçek.
- **Similarity assessment of 3D mechanical components for design reuse**, Chih-Hsing Chu, Yung-Chang Hsu.
- **The power crust, unions of balls, and the medial axis transform**, Nina Amenta, Sunghee Choi, Ravi Krishna Kolluri.
- **The Shock Scaffold for Representing 3D Shape**, Nina Amenta, Sunghee Choi, Ravi Krishna Kolluri.
- **The Shock Scaffold for Representing 3D Shape**, Frederic F. Leymarie and Benjamin B. Kimia.
- **Topological segmentation of discrete surfaces**, Grégoire Malandain, Gilles Bertrand, Nicholas Ayache.
- **Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes**, Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, Tosiyasu L. Kunii.
- **MeshNet: Mesh Neural Network for 3D Shape Representation**, Yutong Feng, Yifan Feng, Haoxuan You, Xibin Zhao, Yue Gao.
- **Revisiting Training Strategies and Generalization Performance in Deep Metric Learning** Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, Joseph Paul Cohen.

## Annexe A : Information Retrieval, similarity and clustering metrics

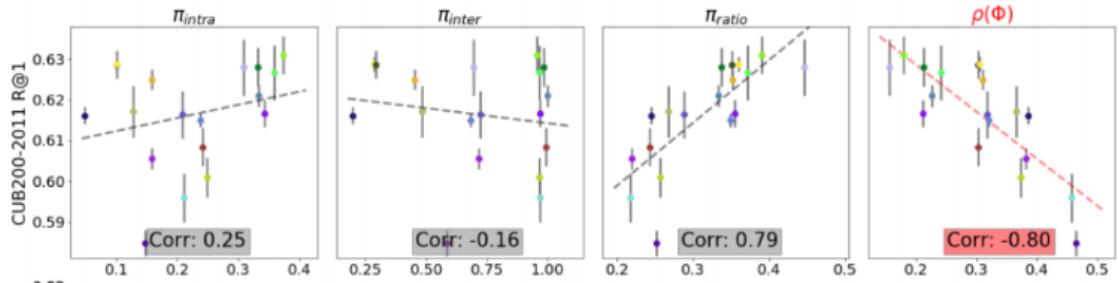


Figure 7.1: Some Clustering Intra/Inter class metrics

**Dunn Index** The Dunn Index is a metric used to evaluate the quality of clustering results. It measures the compactness of clusters (intra-cluster distance) relative to the separation between clusters (inter-cluster distance). A higher Dunn Index indicates better clustering results, with well-separated and compact clusters.

The Dunn Index aims to maximize the inter-cluster distance and minimize the intra-cluster distance.

The Dunn Index is calculated using the following formula:

$$D = \frac{\min_{1 \leq i < j \leq n} d(i, j)}{\max_{1 \leq k \leq n} d'(k)}$$

Here are some other clustering metrics :

**Silhouette Coefficient:** Measures how close each point is to its own cluster compared to other clusters. It ranges from -1 to 1, with higher values indicating better clustering.

**Advantage:** Provides an individual score for each data point, which can help in understanding cluster assignment quality.

**Disadvantage:** It can be sensitive to outliers and performs less well with clusters of very different densities.

**Calinski-Harabasz Index (CH Index):** Also known as the Variance Ratio Criterion, it measures the ratio of between-cluster dispersion to within-cluster dispersion.

**Advantage:** Easy to compute and emphasizes the importance of between-cluster separation.

**Disadvantage:** It may favor larger clusters and may not perform well with clusters of significantly different sizes or densities.

**Rand Index / Adjusted Rand Index (ARI):** These compare the actual clustering with a random clustering, measuring the agreement between the two. The Adjusted Rand Index corrects for chance agreement.

**Advantage:** They are more general, not just applicable to hierarchical clustering, and useful when you have a ground truth for comparison.

**Disadvantage:** They require a predefined ground truth for accurate assessment, which is often not available in real-world clustering tasks.

**Mutual Information (MI):** Definition: Measures the mutual dependence between the true class labels and the cluster assignments.

**Advantage:** It's a non-parametric method that can handle arbitrary cluster shapes and sizes.

**Disadvantage:** Like ARI, it needs ground truth labels for meaningful evaluation.

## Annexe B : Baseline Naive algorithm for 3D similarity

### Baseline Naive similarity based on voxel grid

A naive baseline can consist on overlapping voxel grids from each object, and calculating the percentage of overlap. However this method is not invariant (translation, rotation and scaling). Thus, we thought of either finding the "matching" transformation  $T_i$  : a composition of rotation and translation in 3D.

You can get  $T_i$  by : RANSAC then system solving on descriptors given by SIFT for example. Or any method of registration : ICP, Fast Point Feature Histograms (FPFH) descriptors...<sup>41</sup>

Or in case of having the same scale, and normalising the voxel grids to the same referential (translating to the origin), we simply need to find the rotation that matches the two objects.

For our experiment, we used an implementation of Shape Matching and Anisotropy paper.<sup>42</sup>

---

<sup>41</sup><https://stackoverflow.com/questions/70160183/how-can-i-align-register-two-meshes-in-open3d-python>

<sup>42</sup><https://www.cs.jhu.edu/~misha/MyPapers/SIG04.pdf>

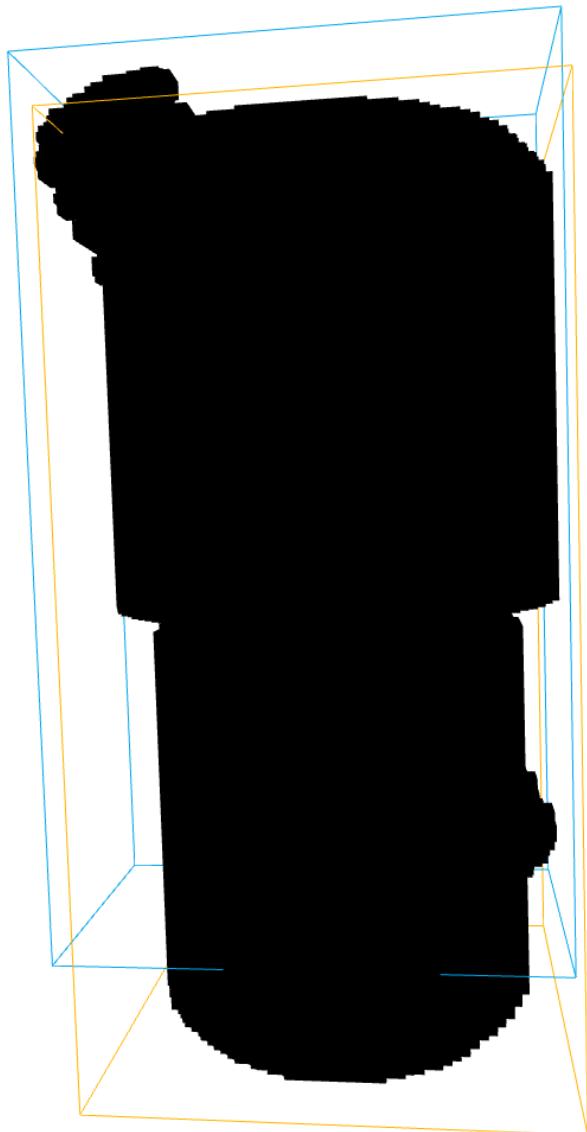


Figure 7.2: *Two aligned voxelised objects and their respective boundary box : blue and orange*

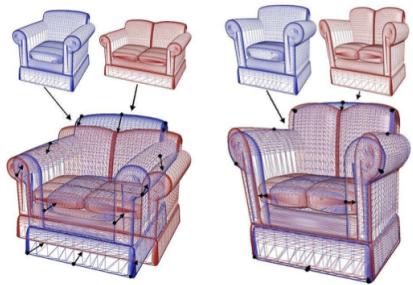


Figure 7.3: *Illustration from the Shape Matching and Anisotropy paper*

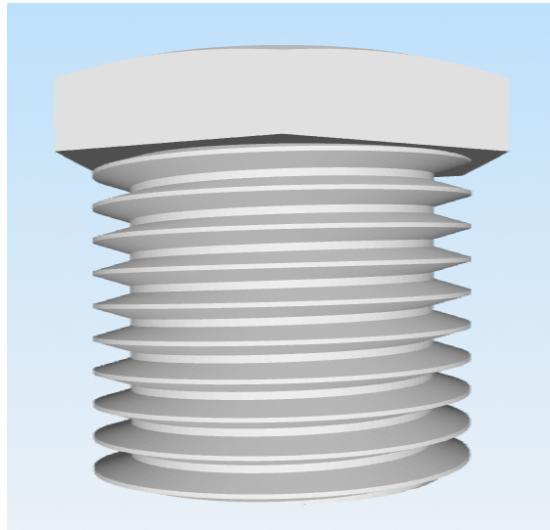


Figure 7.4: *Visualisation of the target mesh*



Figure 7.5: *Visualisation of the source mesh*

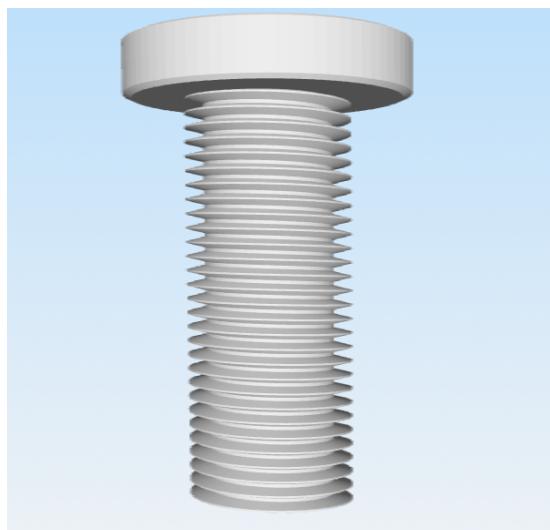


Figure 7.6: *Aligned mesh to target-output of shape matching algorithm*

$$sim(obj1, obj2) = sim(Ti(VoxelGrid(obj1)), VoxelGrid(obj2))$$

## Annexe C : skeleton approach, DAG<sup>43</sup> creation and implementation of indexing algorithm

For the skeleton approach, We consider the use of medial surfaces to represent symmetries of 3-D objects. This allows for a qualitative abstraction based on a directed acyclic graph of components and also a degree of invariance to a variety of transformations including the articulation and deformation of parts.

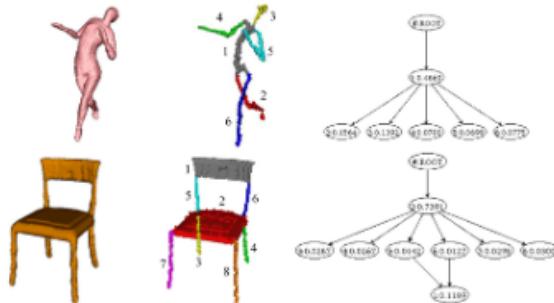


Figure 7.7: From 3D model to Skeleton to DAG

The approach described in the paper follows three steps : Topology preservation thinning algorithm (iteratively remove some voxels while preserving the topology of the 3D model, based on topological classification of surface points<sup>44</sup>, the result is a skeleton : when aligned with the 3D model we call it the medial surface) - Skeleton segmentation (based on connected manifolds, junction points and curvature points ...) - DAG (creating a directed acyclic graph from the segmented skeleton based on manifolds saliences). Then we implemented an indexing algorithm<sup>45</sup>, which computes a topological signature of a DAG, and uses a vote accumulation strategy. A matching algorithm can be used after the indexing results.

$$Sim(obj1, obj2) = Sim(DAG1, DAG2)$$

### Average Outward Flux : AOF First algorithm

<sup>43</sup>Directed Acyclic Graph

<sup>44</sup>Topological segmentation of discrete surfaces, Grégoire Malandain, Gilles Bertrand, Nicholas Ayache.

<sup>45</sup>ndexing Hierarchical Structures Using Graph Spectra Ali Shokoufandeh, Member, IEEE, Diego Macrini, Student Member, IEEE Computer Society, Sven Dickinson, Member, IEEE, Kaleem Siddiqi, and Steven W. Zucker, Fellow, IEEE

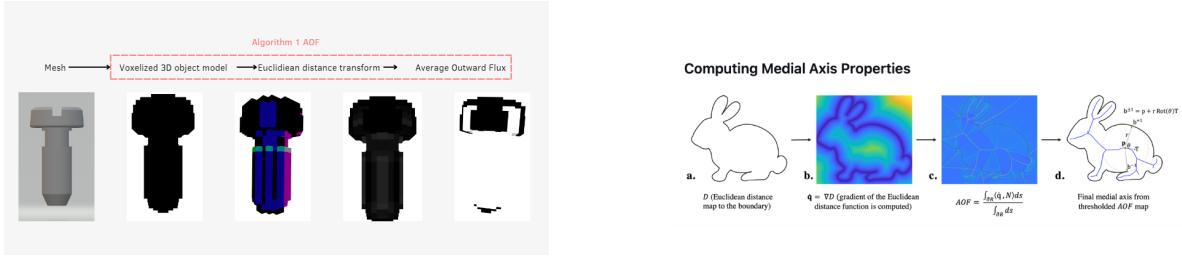


Figure 7.8: *Algorithm 1: AOF*

## Thining Algorithm    Second algorithm

### Algorithm 2: Topology Preserving Thinning

```

Data : 3-D Object Model, Average Outward Flux Map.
Result : 3-D Skeleton (Medial Surface).
for (each point  $x$  on the boundary of the object) do
    if ( $x$  is simple) then
        | insert( $x$ , maxHeap) with  $AOF(x)$  as the sorting key for insertion;
    while ( $maxHeap.size > 0$ ) do
         $x = \text{HeapExtractMax}(maxHeap);$ 
        if ( $x$  is simple) then
            | if ( $x$  is an end point) and ( $AOF(x) < Thresh$ ) then
                | | mark  $x$  as a medial surface (end) point;
            else
                | Remove  $x$ ;
                | for (all neighbors  $y$  of  $x$ ) do
                    | | if ( $y$  is simple) then
                        | | | insert( $y$ , maxHeap) with  $AOF(y)$  as the sorting key for insertion;

```

Figure 7.9: *Illustration of the algorithm from the paper*