Content-Based 3D Models Retrieval system

Ahmed Samady, Fahd Chibani, Mohamed Amine Fakhre-Eddine
IT Department, FST, Abdelmalek Essaadi University
ahmed.samady@etu.uae.ac.ma, fahd.chibani@etu.uae.ac.ma, contact@fakhreeddine.dev,
mohamedamine.fakhreeddine@etu.uae.ac.ma

***Abstract*—Technologies for creating digital 3D models of actual items and using them in a range of fields, including engineering, medical, and cultural heritage, have been more widely available in recent years. Within this context, content-based 3D object retrieval is emerging as a key area of study. Pottery [13] is one of the datasets under investigation, and it served as the foundation for this article. This article presents our two-week content-based 3D models retrieval system's theory, design concepts, implementation, and performance results. Additionally, we recommend the use of strong shape descriptors, such as Fourier descriptors and 3D Zernike moments, which are invariant under transformations, short, easy to index, and descriptive. The 3D content retrieval system performs well for certain categories like Abstract, Alabastron, and Amphora, but Mug and Native American - Effigy consistently underperform, indicating potential issues. Mesh simplification, particularly with larger cell sizes, introduces a trade-off between computational efficiency and retrieval performance. While simplification reduces model complexity, it also leads to a decline in precision, especially for categories already challenging for the system. Finally, we presented our implementation, confirming the theoretical findings.**

***Index Terms*—3D Model, Content-Based, Descriptors, Fourier Transform, Retrieval, Shape, Simplification, Zernike moments.**

I. Introduction

THese days, a significant portion of our data consists of 3D models, and their number is only growing over time. As a result, handling them is crucial, and this includes the duty of retrieving 3D models from massive databases in response to user requests, desires, and requirements. 3D models based on content Retrieval is a method that uses a number of descriptors and features to find models from a 3D model database that are semantically related to the user's query. The implementation of a content-based 3D model retrieval system with a basic search function will be the main goal of this paper.

The paper is structured as follows. We will define the descriptors used in our method and the underlying mathematics in Section III. The computation of the distances utilized in our method was presented in Section IV. The math underlying our search and its algorithm will be presented in Section V. In Sections VI and VII, respectively, we will conclude with an assessment of our approach, an implementation, and a demonstration.

II. Literature review

A. 3D Model Description and Retrieval systems

Numerous techniques to enhance retrieval performance have been highlighted by the thorough study of 3D retrieval systems from various angles. Various approaches have been researched and discussed.

In order to conduct similarity searches for polygonal mesh models, D. Vranic and D. Saupe presented a new method for describing 3D shapes in 2001. Using a coarse voxelization of a 3D model as the input for the 3D Discrete Fourier Transform (3D DFT), they presented a descriptor that is robust with respect to the level of detail and invariant with respect to translation, rotation, scaling, and reflection. The absolute values of the obtained (complex) coefficients are seen as components of the feature vector [1].

From a different angle, M. Novotni and R. Klein advocated the use of descriptors for content-based 3D shape retrieval that are known as 3D Zernike invariants. Their representation's basis polynomials make it easier to calculate invariants under scaling, translation, and rotation. They also talked about the impact of the algorithm's parameters, such as the number of used coefficients, conversion into a volumetric function, etc [2], [3].

Dejan V. Vranić built, analyzed, and tested new methods for describing the 3D shape of polygonal mesh models in his PhD thesis. In order to achieve the best 3D shape descriptors, he compared and examined a variety of feature vectors (descriptors), such as discrete Fourier transforms in 1D, 2D, and 3D, the Fourier transform on a sphere (spherical harmonics), and moments for representing the extent function. Additionally, he employed two methods for combining suitable feature vectors: crossbreeding (hybrid descriptors) and defining a complex function on a sphere. After carefully comparing his feature vectors with those of other authors, he came to the conclusion that a hybrid feature vector performs better than the state-of-the-art. [4].

D. Zarpalas, P. Daras, D. Tzovaras, and M. G. Strintzis suggested combining Fourier Transforms to create an effective descriptor that can be utilized for effective content-based search and retrieval of 3D models, following in the footsteps of D. Vranic and D. Saupe. Each model's volume is subjected to the suggested method after being broken down into rays and planes, to which the 1D and 2D Fourier Transforms are applied, respectively. The Spherical Fourier Transform receives the coefficients generated by these transformations as input, producing rotation-invariant descriptors with important shape features. Their tests demonstrated that the suggested approach outperforms the majority of the methods that were published at the time in terms of precision and recall [5].

From an alternative viewpoint, A. Del Bimbo and P. Pala offered a comparative study of a few distinct approaches for the description and retrieval of 3D models by similarity, which

Masters of Science and Technology in Artificial Intelligence and Data Science, January 23, 2025

are typical of the main categories of methods put forth [6]. They included Geometric moments [7] for Geometry-Based Solutions, Spin Image signatures [8], [9], and Light fields [10] for View-Based Solutions, as well as Statistics-Based Solutions, which include Curvature histograms [11] and Shape functions [12]. Curvature histograms A curvature index, which is composed of a histogram of the principal curvatures of each mesh face, a histogram of the distances between the faces, and a histogram of the volumes based on each face, was the method previously introduced by J.-P. Vandeborre, V. Couillet, and M. Daoudi. This approach described a three-dimensional model by three invariant descriptors and produced satisfactory results on a fifty-three-dimensional model database [11]. In order to calculate shape signatures for arbitrary 3D polygonal models, R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin proposed shape functions. In order to reduce the shape-matching problem to a comparison of probability distributions, the main idea is to represent an object's signature as a shape distribution sampled from a shape function measuring the object's global geometric properties. Despite the presence of arbitrary translations, rotations, scales, mirrors, tessellations, simplifications, and model degeneracies, they found that the differences between sampled distributions of simple shape functions offer a reliable way to distinguish between classes of objects in a moderately sized database [12]. J. Assfalg, G. D'Amico, A. Del Bimbo, and P. Pala used Spin Image to derive a description of both database and query objects that is independent of view. The process involves creating a set of spin images for each object and its constituent parts, evaluating a descriptor for each spin image, and then clustering the set of image-based descriptors for each object to produce a compact representation [8], [9].

A. Koutsoudis, G. Pavlidis, V. Liami, D. Tsiafakis, and C. Chamzas presented a novel approach to 3D pottery processing and retrieval. The approach consists of feature extraction, descriptor encoding, and 3D mesh pre-processing (scale and pose normalization). A vessel is scaled in a unit bounding sphere as part of the 3D mesh preprocessing, and principal components analysis is used to coarsely detect the vessel's axis of symmetry. The vessel is divided into the sections that make up its main body and its appendages using mesh contouring. The identified axis of symmetry is then fine-tuned and aligned with the Y-axis of a right-handed 3D Cartesian coordinate system using only the major body parts. The vessel is then rotated to position its top and appendages in predetermined locations. After contouring the normalized mesh once more, the objects that appear in each contouring level undergo circular regression to start feature extraction. Lastly, the suggested compact shape descriptor is encoded with features like the radius and the center coordinates of the best-fit circles. As anticipated, the suggested descriptor outperformed the general-purpose MPEG-7 SSD descriptor in every scenario [13].

III. Descriptors

The approach to extracting valuable information from 3D models uses 3D Zernike moments and the 3D Fourier Transform. These measures have several features, including descriptive power, conciseness, ease of indexing, and invariance under transformations. The similarity measure should deliver a similarity ordering close to the application-driven notion of resemblance. The descriptor should be compact, minimizing storage requirements and accelerating the search by reducing the problem's dimensionality. Additionally, the computed descriptor values must be invariant under specific application-dependent transformations, such as similarity transformations or deformations, to further accelerate the search process [1]–[3], [5].

A. Zernike moments

Zernike moments are shape descriptors that are obtained by a sequence of mathematical transformations from geometric moments.

1) Geometric Moments: For a 3D mesh model with vertices $\mathbf{V}$ and faces $\mathbf{F}$, the geometric moments are calculated using:

$$G_{ijk} = \sum_{f \in \mathbf{F}} V_f \cdot S_{ijk}(\mathbf{C}_f)$$

where:

- V_f is the volume of facet f
- $\mathbf{C}_f$ represents the monomial coefficients for facet f
- S_{ijk} is the term combining monomial coefficients

The facet volume is computed as:

$$V_f = \det(\mathbf{V}_f)$$

where $\mathbf{V}_f$ is the matrix of vertex coordinates for facet f.

2) Monomial Coefficients: The monomial coefficients for each vertex are calculated as:

$$C_{ijk}(x, y, z) = \binom{i+j+k}{i, j, k} x^i y^j z^k$$

where $\binom{i+j+k}{i,j,k}$ is the trinomial coefficient:

$$\binom{i+j+k}{i, j, k} = \frac{(i+j+k)!}{i!j!k!}$$

3) Zernike Moment Transformation: The transformation from geometric moments to Zernike moments involves several steps:

a) First Transformation (V):

$$V_{abc} = \sum_{\alpha=0}^{a+c} (i)^{\alpha} \binom{a+c}{\alpha} G_{2a+c-\alpha,\alpha,b}$$

b) Second Transformation (W):

$$W_{abc} = \sum_{\alpha=0}^{a} (-1)^{\alpha} 2^{a-\alpha} \binom{a}{\alpha} V_{a-\alpha,b,c+2\alpha}$$

c) Third Transformation (X):

$$X_{abc} = \sum_{\alpha=0}^{a} \binom{a}{\alpha} W_{a-\alpha,b+2\alpha,c}$$

d) Fourth Transformation (Y):

$$Y_{l\nu m} = \sum_j Y_{ljm} X_{\nu+j, l-m-2j, m}$$

where:

$$Y_{ljm} = \frac{(-1)^j \sqrt{2l+1}}{2^l} \frac{\binom{m+j+l-m-2j}{m,j,l-m-2j}\binom{2(l-j)}{l-j}}{\sqrt{\binom{m+m+l-m}{m,m,l-m}}}$$

e) Final Zernike Moments (Z):

$$Z_{nlm} = \frac{3}{4\pi} \sum_\nu Q_{kl\nu} Y^*_{l\nu m}$$

where:

$$Q_{kl\nu} = \frac{(-1)^{k+\nu}}{4^k} \sqrt{\frac{2l+4k+3}{3}} \frac{\binom{\nu+k-\nu+l+\nu+1}{\nu,k-\nu,l+\nu+1}\binom{2(l+\nu+1+k)}{l+\nu+1+k}}{\binom{2(l+\nu+1)}{l+\nu+1}}$$

4) Feature Extraction: The final descriptors are computed as:

$$F_{nl} = \|\mathbf{Z}_{nl}\|_2$$

where $\mathbf{Z}_{nl}$ is the vector of Zernike moments for fixed n and l.

5) Implementation Constraints: For numerical stability and computational efficiency:

- Input models are centered and normalized to unit sphere
- Maximum order N is typically set to 3-5
- Only moments satisfying $(n-l) \mod 2 = 0$ are computed
- Complex conjugates are handled appropriately for numerical stability

B. Fourier Descriptors

The system generates shape descriptors by converting 3D mesh models into voxel representations and using the 3D Discrete Fourier Transform (DFT).

1) Voxelization Process: Given a 3D mesh model with vertices $\mathbf{V} \in \mathbb{R}^{n \times 3}$ and faces $\mathbf{F}$, the voxelization process maps the model into a binary voxel grid $\mathcal{V} \in \{0,1\}^{N \times N \times N}$.

a) Normalization: The vertices are normalized using: $\mathbf{V}_{norm} = \frac{\mathbf{V}-\min(\mathbf{V})}{s} - 0.5$
where:

- $s = \max(\max(\mathbf{V}) - \min(\mathbf{V}))$ is the maximum dimension of the bounding box
- The subtraction of 0.5 centers the model in the voxel grid

b) Voxel Mapping: The normalized vertices are mapped to voxel coordinates:

$$\mathbf{V}_{voxel} = \text{clip}(\lfloor (\mathbf{V}_{norm} + 0.5)N \rfloor, 0, N-1)$$

where:

- N is the voxel grid resolution
- $\text{clip}(x, a, b)$ constrains values to range $[a, b]$

c) Face Rasterization: For each triangular face $f \in \mathbf{F}$ with vertices (v_1, v_2, v_3), we set:
$\mathcal{V}[x, y, z] = 1$ for all (x, y, z) in the bounding box:

$$x \in [\min(v_{1x}, v_{2x}, v_{3x}), \max(v_{1x}, v_{2x}, v_{3x})]$$

$$y \in [\min(v_{1y}, v_{2y}, v_{3y}), \max(v_{1y}, v_{2y}, v_{3y})]$$

$$z \in [\min(v_{1z}, v_{2z}, v_{3z}), \max(v_{1z}, v_{2z}, v_{3z})]$$

2) 3D Discrete Fourier Transform:

a) Forward Transform: The 3D DFT of the voxel grid is computed as:

$$\mathcal{F}(u, v, w) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \mathcal{V}(x, y, z) e^{-2\pi i (\frac{ux}{N} + \frac{vy}{N} + \frac{wz}{N})}$$

where:

- (u, v, w) are the frequency coordinates
- i is the imaginary unit

b) Frequency Shift: The zero frequency is shifted to the center:

$$\mathcal{F}_{shifted}(u, v, w) = \mathcal{F}((u + \frac{N}{2}) \bmod N, (v + \frac{N}{2}) \bmod N, (w + \frac{N}{2}) \bmod N)$$

c) Coefficient Selection: Given a bandwidth parameter K, we select coefficients:

$$\{|\mathcal{F}_{shifted}(u, v, w)| : u, v, w \in [-K, K]\}$$

subject to lexicographic ordering constraint:

$$(u > 0) \lor (u = 0 \land v > 0) \lor (u = v = 0 \land w \geq 0)$$

This results in a feature vector $\mathbf{f} \in \mathbb{R}^m$ where $m = \frac{(2K+1)^3+1}{2}$

3) Implementation Considerations:

- The voxelization resolution N is typically set to 128 for a balance between accuracy and computational efficiency
- The bandwidth parameter K is usually set to 2, yielding 63 coefficients
- Only unique Fourier coefficients are stored due to conjugate symmetry
- **FFT** is used for efficient computation of the **DFT**
- The descriptor is invariant to translation due to the normalization step
- Scale invariance is achieved through the normalization to unit bounding box

IV. Distances

To make use of these features, we have to compute the distance between a query model and an indexed model for each feature.

- Zernike Moments (M_z)
- Fourier Descriptors (F_c)

For every feature, two distance metrics are supported:

- L_1 Distance

$$d_{L_1}(\mathbf{d}_1, \mathbf{d}_2) = \sum_i |d_{1i} - d_{2i}|$$

- L_2 Distance

$$d_{L_2}(\mathbf{d}_1, \mathbf{d}_2) = \sqrt{\sum_i (d_{1i} - d_{2i})^2}$$

where $\mathbf{d}$ is the respective feature we want to compute the distance for.

V. Search system

Our retrieval system utilizes a dual-descriptor approach combining Zernike moments and Fourier descriptors, followed by a weighted similarity calculation pipeline.

A. Feature Extraction Pipeline

For a 3D model $\mathcal{M}$ with vertices $\mathbf{V}$ and faces $\mathbf{F}$, the system computes:

$$\mathcal{D}(\mathcal{M}) = \{\mathbf{z}, \mathbf{f}\}$$

where:

- $\mathbf{z} \in \mathbb{R}^{d_z}$ is the Zernike moment descriptor
- $\mathbf{f} \in \mathbb{R}^{d_f}$ is the Fourier descriptor

B. Distance Computation

For query model $\mathcal{Q}$ and database model $\mathcal{D}_i$, distances are computed:

$$d_z(\mathcal{Q}, \mathcal{D}_i) = d_{L_2}(\mathcal{Q}, \mathcal{D}_i)$$

$$d_f(\mathcal{Q}, \mathcal{D}_i) = d_{L_2}(\mathcal{Q}, \mathcal{D}_i)$$

where:

- d_z is the L_2 distance between Zernike descriptors
- d_f is the L_2 distance between Fourier descriptors

C. Distance Normalization

For each descriptor type $k \in \{z, f\}$, distances are normalized using min-max scaling:

$$\hat{d}_k(\mathcal{Q}, \mathcal{D}_i) = \frac{d_k(\mathcal{Q}, \mathcal{D}_i) - \min_j d_k(\mathcal{Q}, \mathcal{D}_j)}{\max_j d_k(\mathcal{Q}, \mathcal{D}_j) - \min_j d_k(\mathcal{Q}, \mathcal{D}_j)}$$

Special case for constant distances : $\hat{d}_k(\mathcal{Q}, \mathcal{D}_i) = 0$ if $\max_j d_k(\mathcal{Q}, \mathcal{D}_j) = \min_j d_k(\mathcal{Q}, \mathcal{D}_j)$

D. Similarity Score Computation

1) Weighted Fusion: Given weights w_z and w_f where $w_z + w_f = 1$, the final similarity score is:

$$s(\mathcal{Q}, \mathcal{D}_i) = 1 - (w_z \hat{d}_z(\mathcal{Q}, \mathcal{D}_i) + w_f \hat{d}_f(\mathcal{Q}, \mathcal{D}_i))$$

2) Ranking: Models are ranked by decreasing similarity:

$$\mathcal{R}(\mathcal{Q}) = \{\mathcal{D}_i : s(\mathcal{Q}, \mathcal{D}_i) \geq s(\mathcal{Q}, \mathcal{D}_{i+1})\}$$

3) Retrieval Algorithm:

Algorithm 1 Retrieval Algorithm

```
Input: Query model $\mathcal{Q}$, number of results $n$
Extract descriptors $\mathcal{D}(\mathcal{Q})$
for each database model $\mathcal{D}_i$ do
    Compute $d_z(\mathcal{Q}, \mathcal{D}_i)$ and $d_f(\mathcal{Q}, \mathcal{D}_i)$
end for
Normalize distances to obtain $\hat{d}_z$ and $\hat{d}_f$
Compute similarity scores $s(\mathcal{Q}, \mathcal{D}_i)$
Return top $n$ models by similarity
```

4) Complexity Analysis: For database size N:

- Feature extraction: $O(V)$ for V vertices
- Distance computation: $O(N(d_z + d_f))$
- Normalization: $O(N)$
- Sorting: $O(N \log N)$

Total retrieval complexity: $O(N \log N)$

VI. Evaluation

In this section, we evaluate the performance of the 3D content retrieval system using a benchmark of **39 models** from **14 different categories**: Abstract, Alabastron, Amphora, Bowl, Hydria, Lekythos, Modern-Bottle, Mug, Native American - Bottle, Native American - Bowl, Native American - Effigy, Native American - Jar, Oinochoe, and Psykter. These categories were chosen based on the dataset owner's recommendation, as stated on their website:

”For implementing retrieval experiments and shape descriptor performance evaluation, we have used five generic shape categories (Ancient Greek Alabastron, Amphora, Psykter, Lekythos, and Hydria) due to their relatively large population in the dataset and geometrical coherence. Other classes found in the dataset can also be used. No part of the dataset has been defined as a training set. In our experiments, we have used all models of a class as queries against the whole dataset.” [13]

However, we expanded the evaluation to include additional categories to ensure a more comprehensive analysis. It is important to note that the benchmarking process was conducted manually. This was necessary because the 3DPottery dataset's initial classification, while semantically consistent, does not always reflect similar geometrical properties across models within the same category. As a result, manual evaluation was required to ensure accurate and meaningful retrieval performance metrics.

A. No Simplification

The first configuration, without any mesh simplification, serves as the baseline for our evaluation. Figure 1 shows the average precision per category for this configuration.

- **High Precision Categories**: Categories such as **Abstract**, **Alabastron**, and **Amphora** consistently achieve perfect precision (P@5 = 1) across all k values. This indicates that the retrieval system performs exceptionally well for these categories.
- **Moderate Precision Categories**: Categories like **Bowl** and **Hydria** show high precision at lower k values but experience a gradual decline as k increases. For example, **Bowl** has P@5 = 1 but drops to P@20 = 0.7.
- **Low Precision Categories**: Categories such as **Mug** and **Native American - Effigy** exhibit poor performance, with P@5 = 0.2 and P@20 = 0.05, respectively. This suggests that the retrieval system struggles with these categories.

The overall average precision across all categories is 0.85 for P@5, 0.79 for P@10, 0.75 for P@15, and 0.70 for P@20. This indicates that the system performs well for most categories but has room for improvement in others.

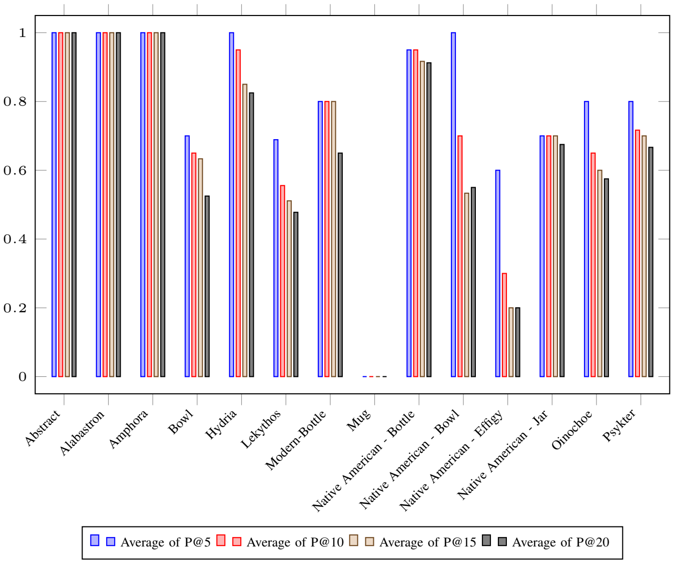


Fig. 1. Average Precision per Category (No Simplification)

B. Vertex Clustering at 0.03 Cell Size

The second configuration applies vertex clustering with a cell size of 0.03, which simplifies the mesh while attempting to preserve the overall shape. Figure 2 shows the average precision per category for this configuration.

- **High Precision Categories**: Similar to the no simplification case, **Abstract**, **Alabastron**, and **Amphora** maintain high precision, with P@5 = 1 and P@20 = 0.975.
- **Moderate Precision Categories**: Categories like **Hydria** and **Native American - Bottle** show a slight drop in precision compared to the no simplification case. For example, **Hydria** has P@5 = 0.85 and P@20 = 0.69.
- **Low Precision Categories**: Categories such as **Mug** and **Native American - Effigy** continue to perform poorly, with P@5 = 0.2 and P@20 = 0.05, respectively.

The overall average precision for this configuration is 0.75 for P@5, 0.68 for P@10, 0.64 for P@15, and 0.60 for P@20. This indicates that while the system maintains high precision for some categories, the simplification process slightly reduces performance for others.

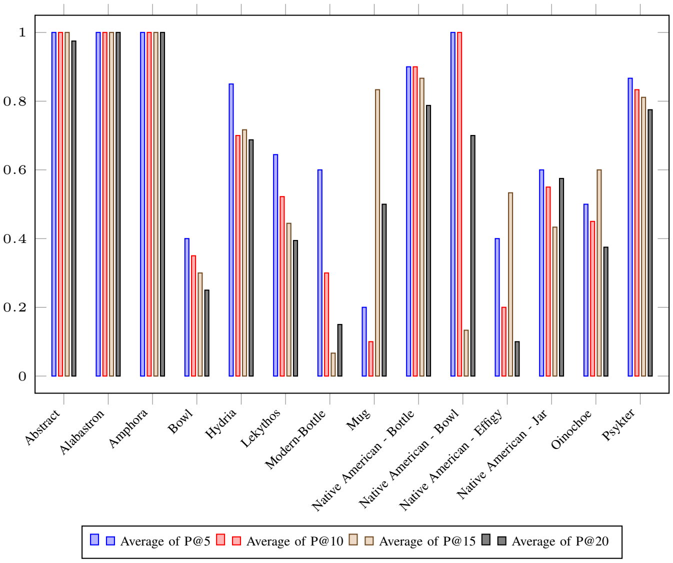


Fig. 2. Average Precision per Category (VC @ 0.03 Cell Size)

C. Vertex Clustering at 0.1 Cell Size

The third configuration applies vertex clustering with a larger cell size of 0.1, resulting in more aggressive mesh simplification. Figure 3 shows the average precision per category for this configuration.

- **High Precision Categories**: **Abstract**, **Alabastron**, and **Amphora** still achieve perfect precision at P@5 = 1, but **Abstract** drops to P@20 = 1, indicating a slight decline in performance.
- **Moderate Precision Categories**: Categories like **Bowl** and **Hydria** show further declines in precision compared to the previous configurations. For example, **Bowl** has P@5 = 0.7 and P@20 = 0.53.
- **Low Precision Categories**: **Mug** and **Native American - Effigy** continue to perform poorly, with P@5 = 0 and P@20 = 0, respectively. This suggests that aggressive simplification significantly impacts retrieval performance for these categories.

The overall average precision for this configuration is 0.81 for P@5, 0.74 for P@10, 0.70 for P@15, and 0.67 for P@20. While the system maintains relatively high precision for some categories, the more aggressive simplification leads to a noticeable drop in performance for others.

D. Summary

The evaluation results demonstrate that the 3D content retrieval system performs well for certain categories, such as

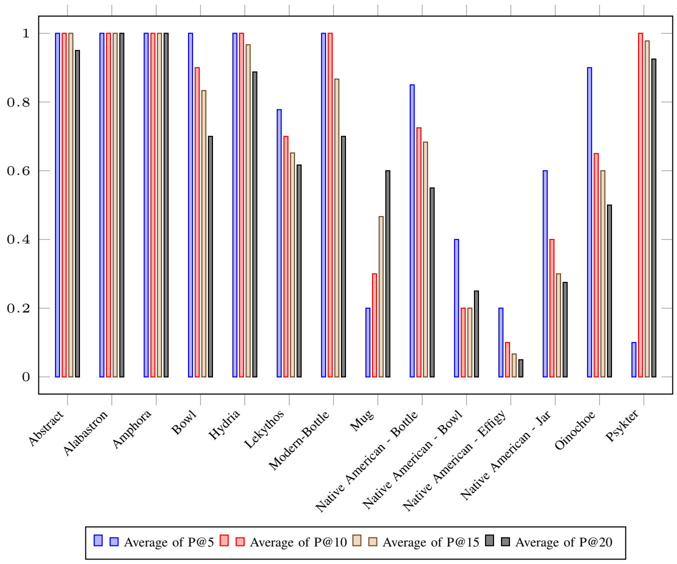


Fig. 3. Average Precision per Category (VC @ 0.1 Cell Size)

Abstract, **Alabastron**, and **Amphora**, across all configurations. However, categories like **Mug** and **Native American - Effigy** consistently underperform, indicating potential issues with the retrieval system for these types of models.

Mesh simplification, particularly with a larger cell size, introduces a trade-off between computational efficiency and retrieval performance. While simplification reduces the complexity of the models, it also leads to a decline in precision, especially for categories that are already challenging for the system.

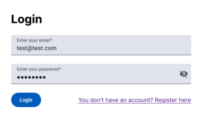
VII. Implementation and Demonstration

A. Description

We created a session-based application that uses web technologies for user accessibility in order to test our methods; no previous search or usage data is saved or stored. It has many different features.

B. Features

1) Basic login and registration:


Login

Enter your email*: test@test.com

Enter your password*: ●●●●●●●●●

Fig. 4. Login page

2) Providing images: The user has total control over the creation of a dataset of 3D models, including the ability to add and remove data.

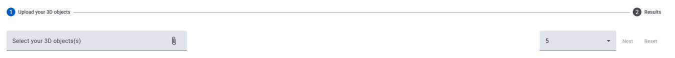
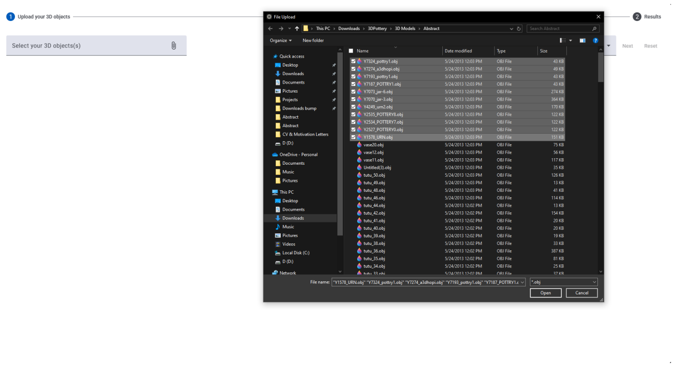


Fig. 5. Dataset building page

The user can upload multiple 3D models at once without their thumbnails for convenience.


File Upload

This PC › Downloads › 3DPottery › 3D Models › Abstract

Name	Date modified	Type	Size
☑ Y7324_pottry1.obj	5/24/2013 12:03 PM	OBJ File	43 KB
☑ Y7274_a3dhopi.obj	5/24/2013 12:03 PM	OBJ File	49 KB
☑ Y7193_pottry1.obj	5/24/2013 12:03 PM	OBJ File	43 KB
☑ Y7187_POTTRY1.obj	5/24/2013 12:03 PM	OBJ File	43 KB
☑ Y7073_jar-6.obj	5/24/2013 12:03 PM	OBJ File	274 KB
☑ Y7070_jar-3.obj	5/24/2013 12:03 PM	OBJ File	364 KB
☑ Y4249_urn2.obj	5/24/2013 12:03 PM	OBJ File	170 KB
☑ Y2535_POTTERY8.obj	5/24/2013 12:03 PM	OBJ File	122 KB
☑ Y2534_POTTERY7.obj	5/24/2013 12:03 PM	OBJ File	122 KB
☑ Y2527_POTTERY0.obj	5/24/2013 12:03 PM	OBJ File	122 KB
☑ Y1578_URN.obj	5/24/2013 12:03 PM	OBJ File	151 KB
vase20.obj	5/24/2013 12:03 PM	OBJ File	75 KB
vase12.obj	5/24/2013 12:03 PM	OBJ File	56 KB
vase11.obj	5/24/2013 12:03 PM	OBJ File	117 KB
Untitled(3).obj	5/24/2013 12:03 PM	OBJ File	35 KB
tutu_50.obj	5/24/2013 12:03 PM	OBJ File	126 KB
tutu_49.obj	5/24/2013 12:03 PM	OBJ File	13 KB
tutu_48.obj	5/24/2013 12:03 PM	OBJ File	41 KB
tutu_46.obj	5/24/2013 12:03 PM	OBJ File	114 KB
tutu_44.obj	5/24/2013 12:02 PM	OBJ File	13 KB
tutu_42.obj	5/24/2013 12:02 PM	OBJ File	154 KB
tutu_41.obj	5/24/2013 12:02 PM	OBJ File	20 KB
tutu_40.obj	5/24/2013 12:02 PM	OBJ File	20 KB
tutu_39.obj	5/24/2013 12:02 PM	OBJ File	19 KB
tutu_38.obj	5/24/2013 12:02 PM	OBJ File	33 KB
tutu_36.obj	5/24/2013 12:02 PM	OBJ File	387 KB
tutu_35.obj	5/24/2013 12:02 PM	OBJ File	81 KB
tutu_34.obj	5/24/2013 12:02 PM	OBJ File	25 KB
tutu_33.obj	5/24/2013 12:02 PM	OBJ File	37 KB

File name: "Y1578_URN.obj" "Y7324_pottry1.obj" "Y7274_a3dhopi.obj" "Y7193_pottry1.obj" "Y7187_POTTRY1.c

*.obj

Fig. 6. 3D models upload demonstration

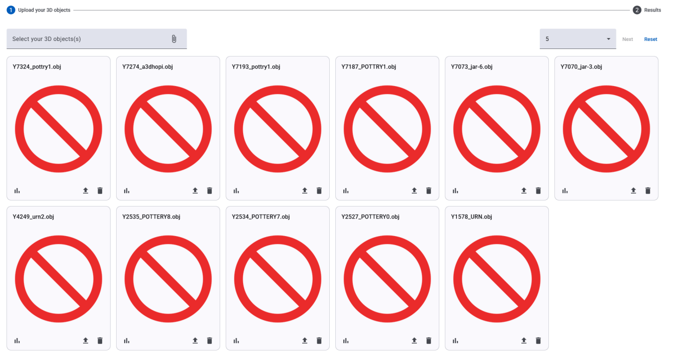


Fig. 7. Dataset building page with multiple 3D models uploaded without thumbnails

An uploaded model's thumbnail can be updated by the user for improved display if desired.

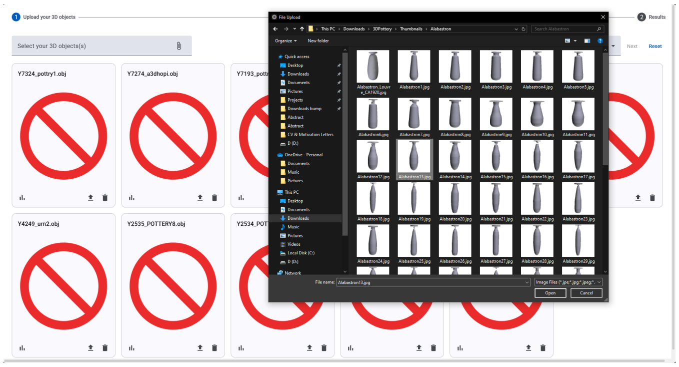


Fig. 8. 3D model thumbnail update demonstration

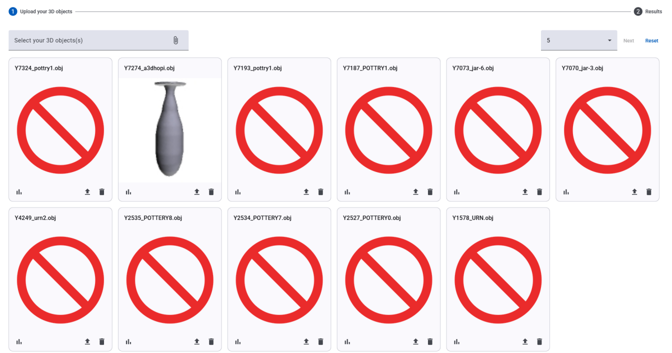
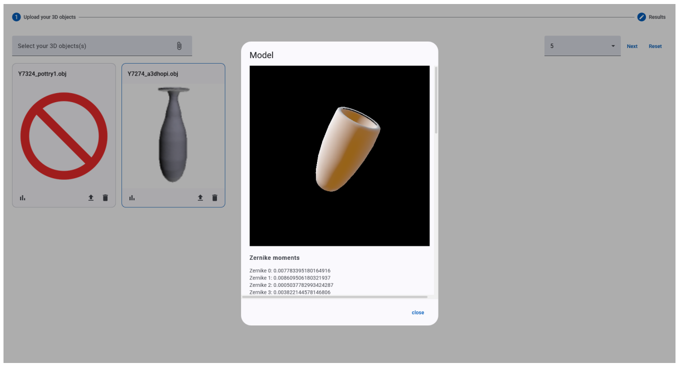


Fig. 9. 3D model with thumbnail updated

3) Consult descriptors: The user can readily obtain the Zernike moments and Fourier descriptors for a chosen model. as well as the model itself, allowing the user to explore and examine it in three dimensions.



Zernike moments

Zernike 0: 0.007783395180164916
Zernike 1: 0.008609506180321937
Zernike 2: 0.0005037782993424287
Zernike 3: 0.003822144578146806

Fig. 10. 3D model view and descriptors

4) Search: The user can select the number of desired results and use our search method to get the results after selecting a model.

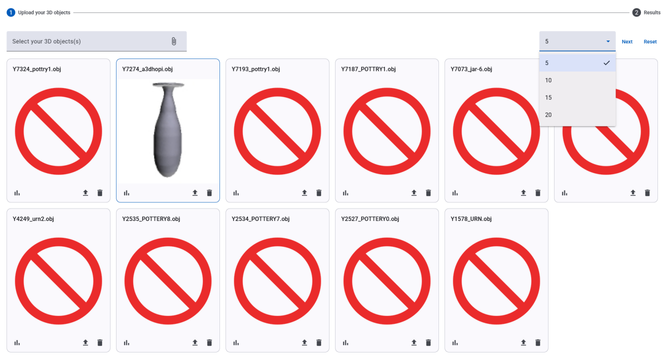
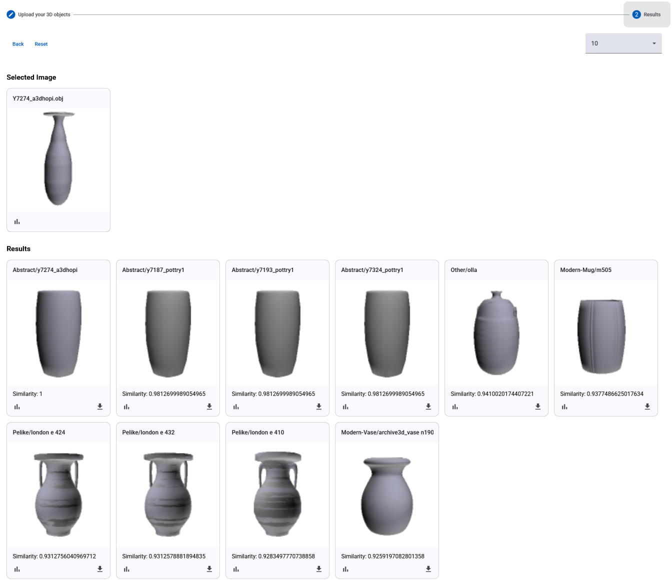


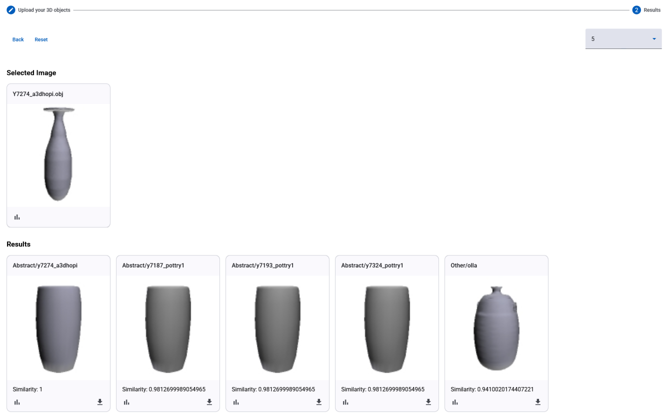
Fig. 11. Number of results choice



Similarity: 1
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9410020174407221
Similarity: 0.9377486625017634
Similarity: 0.9312756040969712
Similarity: 0.9312578881894835
Similarity: 0.9283497770738858
Similarity: 0.9259197082801358

Fig. 12. Obtained results

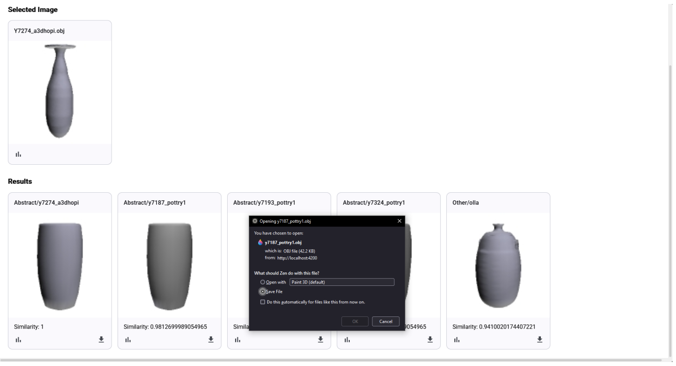
It is simple and convenient for the user to change the quantity of results in the results tab.



Similarity: 1
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9410020174407221

Fig. 13. Obtained results with a different number of results

Additionally, the user can download any returned model in the results.



Opening y7187_pottry1.obj

You have chosen to open:

y7187_pottry1.obj

which is: OBJ file (42.2 KB)

from: http://localhost:4200

What should Zen do with this file?

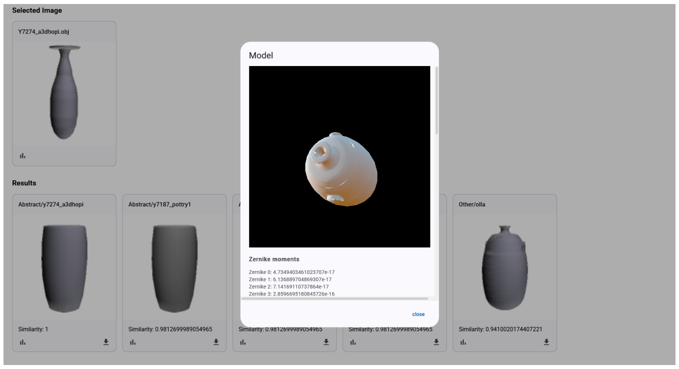
○ Open with Paint 3D (default)

● Save File

□ Do this automatically for files like this from now on.

Fig. 14. Downloading a model from the results

Similar to query models, you can also view the model in three dimensions and consult its descriptors.


Selected Image
Y7274_a3dhopi.obj
Results
Abstract/y7274_a3dhopi
Abstract/y7187_pottry1
Other/olla
Model
Similarity: 1
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9812699989054965
Similarity: 0.9410020174407221
close

Zernike moments

```
Zernike 0: 4.7349403461023707e-17
Zernike 1: 6.136889704869307e-17
Zernike 2: 7.14169110737864e-17
Zernike 3: 2.8596695180845726e-16
```

Fig. 15. Consulting a model's descriptors from the results

VIII. Conclusion and future work

In order to retrieve 3D models, we used the Fourier descriptors and 3D Zernike moments in this paper. We discussed our work's theoretical foundations. Since 3D Zernike moments are derived from geometrical moments, which are known for their instability, we also conducted a comparative evaluation of our system using various simplification techniques applied to the models. This evaluation showed that our system performed well in some categories but poorly in others. This could be because of the implementation. In order to find a solution, future research could address the issue from this angle.

References

[1] Dejan V. Vranic and Dietmar Saupe. 3d shape descriptor based on 3d fourier transform. 2001.

[2] Marcin Novotni and Reinhard Klein. Shape retrieval using 3d zernike descriptors. *Computer-Aided Design*, 36(11):1047–1062, 2004. Solid Modeling Theory and Applications.

[3] Marcin Novotni and Reinhard Klein. 3d zernike descriptors for content based shape retrieval. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, SM '03, page 216–225, New York, NY, USA, 2003. Association for Computing Machinery.

[4] Dejan V. Vranić. *3D Model Retrieval*. PhD thesis, University of Leipzig, June 2004. Ph.D. Dissertation, submitted on December 22, 2003.

[5] Dimitrios Zarpalas, Petros Daras, Dimitrios Tzovaras, and M Strintzis. 3d shape descriptors based on fourier transforms. 04 2005.

[6] Alberto Del Bimbo and Pietro Pala. Content-based retrieval of 3d models. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):20–43, February 2006.

[7] Michael Elad, Ayellet Tal, and Sigal Ar. Content based retrieval of vrml objects — an iterative and interactive approach. *the sixth Eurographics workshop on Multimedia*, 2001, 01 2001.

[8] J. Assfalg, A. Del Bimbo, and P. Pala. Spin images for retrieval of 3d objects by local and global similarity. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 906–909 Vol.3, 2004.

[9] J. Assfalg, G. D'Amico, A. Del Bimbo, and P. Pala. 3d content-based retrieval with spin images. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 2, pages 771–774 Vol.2, 2004.

[10] Wei-Chao Chen, Lars Nyland, Anselmo Lastra, and Henry Fuchs. Acquisition of large-scale surface light fields. In *ACM SIGGRAPH 2003 Sketches & Applications*, SIGGRAPH '03, page 1, New York, NY, USA, 2003. Association for Computing Machinery.

[11] J.-P. Vandeborre, V. Couillet, and M. Daoudi. A practical approach for 3d model indexing by combining local and global invariants. In *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pages 644–647, June 2002.

[12] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, October 2002.

[13] Anestis Koutsoudis, George Pavlidis, Vassiliki Liami, Despoina Tsiafakis, and Christodoulos Chamzas. 3d pottery content-based retrieval based on pose normalisation and segmentation. *Journal of Cultural Heritage*, 11(3):329–338, 2010.