Multimodal 3D Object Retrieval

Maria Pegia[1,2][0000−0003−2643−0028], Björn Þór Jónsson[2][0000−0003−0889−3491], Anastasia Moumtzidou[1][0000−0001−7615−8400], Sotiris Diplaris[1][0000−0002−9969−6436], Ilias Gialampoukidis[1][0000−0002−5234−9795], Stefanos Vrochidis[1][0000−0002−2505−9178], and Ioannis Kompatsiaris[1][0000−0001−6447−9020]

[1] Information Technologies Institute - Centre for Research and Technology Hellas, Thessaloniki, Greece
{mpegia,moumtzid,diplaris,heliasgj,stefanos,ikom}@iti.gr
[2] Reykjavik University, Reykjavik, Iceland {mariap22, bjorn}@ru.is

Abstract. Three-dimensional (3D) retrieval of objects and models plays a crucial role in many application areas, such as industrial design, medical imaging, gaming and virtual and augmented reality. Such 3D retrieval involves storing and retrieving different representations of single objects, such as images, meshes or point clouds. Early approaches considered only one such representation modality, but recently the CMCL method has been proposed, which considers multimodal representations. Multimodal retrieval, meanwhile, has recently seen significant interest in the image retrieval domain. In this paper, we therefore explore the application of state-of-the-art multimodal image representations to 3D retrieval, in comparison to existing 3D approaches. In a detailed study over two benchmark 3D datasets, we show that the MuseHash approach from the image domain outperforms other approaches, improving recall over the CMCL approach by about 11% for unimodal retrieval and 9% for multimodal retrieval.

Keywords: 3D retrieval · Supervised learning · 3D data.

1 Introduction

In recent years, advances in three-dimensional (3D) modeling tools [16], 3D scanning technology [10], and consumer devices with 3D sensors [3] have made it easier to create, share, and access 3D large collections of content, influencing various domains, from entertainment and gaming to healthcare [27], archaeology [1], computer-aided design (CAD) [8] and autonomous systems [7]. A large number of 3D models have now become available on the Web [17], where users can freely download, modify and build upon 3D models that suit their requirements, which not only saves costs and time in product design but also enhances product reliability and quality. Sifting through the vast number of available models to find the right one quickly and accurately is challenging, however. This is where 3D model retrieval techniques come into play, allowing users to retrieve models in a variety of ways, for example based on model class or model similarity.

A fundamental issue for efficient 3D model retrieval at scale is the data representation of the 3D models [6]. Figure 1 outlines the approaches considered in the literature: voxels (data points on a grid in the model space); point clouds (data points of interest in the model space); meshes (networks of triangles that approximate the shape); and multi-view images that visually represent the model. Of these, 3D mesh models stand out due to their capacity to capture intricate details and structural aspects. Recently, however, the CMCL method [11] combines the latter three representation modalities into a unified representation, leveraging the center information from each modality. This integration has resulted in improved retrieval accuracy and robustness. Nevertheless, challenges remain in representing and combining modalities due to the complexities inherent in 3D data.

Multimodal representation has also recently received extensive attention in the domain of image retrieval [19], [18]. While all images have a visual component, that can for example be described using semantic feature vectors, some collections may also have textual, temporal or spatial information that could be combined in various ways for more accurate retrieval, depending on the intended application. Early approaches considered cross-modal retrieval, typically attempting to learn a unified feature space for the visual and textual modalities, but more recently other approaches have considered the fusion of multiple modalities, such as Label-Attended Hashing (LAH) [30] and MuseHash [18]. It is therefore of interest to consider whether the approaches developed in the domain of image retrieval could be applicable in the related, yet significantly different, domain of 3D model retrieval.

The main contributions of this paper can be summarized as follows:

- We adapt state-of-the-art methods from the image retrieval domain to the 3D model retrieval domain.
- In a comparison with the state-of-the-art 3D retrieval methods using two class-based retrieval benchmarks from the literature, we show that the MuseHash approach generally performs best, improving recall over the CMCL approach by as much as 11% for unimodal retrieval and 9% for multimodal retrieval.
- Furthermore, we explore the performance of various combinations of the 3D mesh, point cloud, and visual (multi-view) modalities, showing that the combining 3D mesh and visual modalities improves average precision over 3D meshes alone, while using all three modalities gives the best accuracy.

The remainder of this paper is organized as follows: Section 2 provides an overview of the relevant state-of-the-art research in the domains of 3D model and image retrieval. Section 3 then details how image retrieval methods are adapted to 3D model retrieval. Section 4 presents and analyzes the experimental results for both unimodal and multimodal retrieval, before concluding in Section 5.

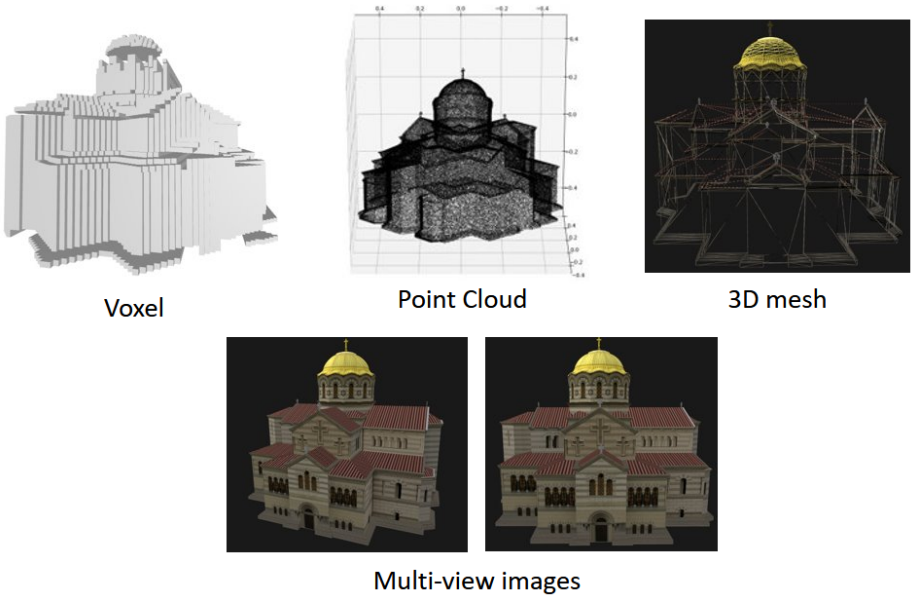

Fig. 1. Examples of different 3D data representations.

2 Related Work

2.1 Unimodal retrieval

Volumetric data representations (Figure 1) have been crucial in 3D data analysis and retrieval, prompting the development of diverse techniques:

Voxels Divide 3D space into a grid, assigning values to voxels [12,23]. Common in medical design, but computationally demanding for large data. Maturana et al. [14] introduced a volumetric occupancy network called VoxNet to achieve robust 3D object recognition. Wang et al. [24] proposed an Octree-based CNN for 3D shape classification.

Multi-view images Capture 2D images for 3D reconstruction [21,22]. Useful for diverse viewpoints, but quality relies on captured views. Lin et al. [13] proposed two self-attention modules, View Attention Module and Instance Attention Module for building the representation of a 3D object as the aggregation of three features: original, view-attentive, and instance-attentive.

Point Clouds Depict objects using individual points, which is particularly applicable in robotic [21,22]. Sparse and irregular points pose challenges. Qi et al. [2] introduced PointNet, a network architecture that effectively harnesses unordered point clouds and offers a comprehensive end-to-end solution for classification/retrieval tasks. DGCNN [25] employs dynamic graph convolution for point cloud processing, though challenges persist due to point cloud sparsity and irregularity.

3D meshes Describe surface geometry with vertices, edges, faces [21,22]. While they find applications in graphics and design, they also come with computational and storage complexities. MeshNet [6] transforms mesh data into a list of faces, calculating two types of information for each face: a spatial vector based on center data and a structural vector using center, normal, and neighbor information. These features are then merged using a multi-layer perceptron. In contrast, MeshCNN [9] applies convolution and pooling operations to mesh edges and the edges of connected triangles. When pooling is needed, it collapses edges while preserving the overall mesh structure.

Each representation method has its strengths and drawbacks. Voxels offer a structured approach for occupancy and property representation, but can be resource-intensive. Multi-view images leverage multiple perspectives for reconstruction, but accuracy depends on captured views. Point Clouds are storage-efficient and precise, but sparsity and irregularity pose challenges. 3D Meshes capture complex shapes and details, yet computational intensity and storage demands are notable. Deciding between the aforementioned representations methods, depends on the factors like accuracy, efficiency, and suitability.

In our research, we empasize on mesh data, because they perform best due to its rich information representation [11]. Specifically, for our unimodal experimentation, we chose the most recent methods, MeshNet and MeshCNN as our prefered candidates based on research [11].

2.2 Multimodal 3D Object Retrieval

The evolution of 3D data retrieval has spurred the exploration of multimodal approaches, which leverages diverse views to enhance accuracy and versatility. An important development is the fusion of multiple 3D views, capitalizing on their respective strengths in geometric precision and surface details. This integration answers the call for more potent retrieval systems capable of capturing intricate object traits while preserving precise shapes. By harmonizing these distinct viewpoints, retrieval techniques gain proficiency in object recognition, proving effective across an array of practical applications.

In a recent approach called Cross-Modal Center Loss (CMCL) [11], point clouds, meshes, and multi-view images are integrated into a single unified framework. Within this framework, multiple 3D modalities are combined to collectively train representations and identify optimal features. Various loss functions, including cross-entropy and mean-square-error, are employed to refine and enhance the performance of this framework. However, it can be computationally intensive due to the integration of multiple 3D modalities into a single framework, potentially requiring significant computational resources. Moreover, Additionally, CMCL's performance can vary depending on the dataset, as it is sensitive to the central characteristics of each modality.

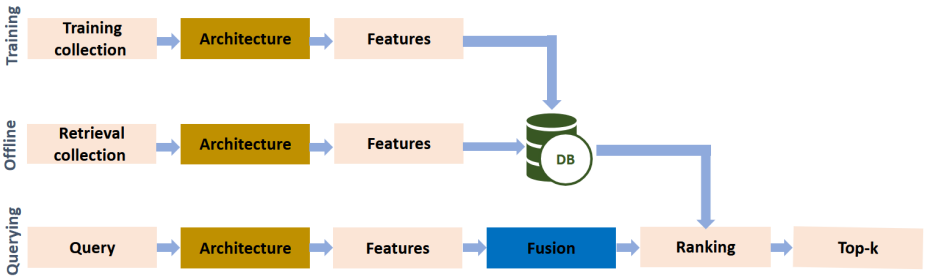

Fig. 2. Overview of abstract framework.

2.3 Multimodal Image Retrieval

Within the rich landscape of multimodal image retrieval methods for retrieval, various strategies combining different modalities have been explored in the literature with more emphasis on hashing methods due to fast queries and less memory consumption. For instance, there are methods like Discrete Online Cross-modal Hashing (DOCH) [31], which creates high-quality hash codes for different data types by using both the likeness between data points and their detailed meanings. Fast Cross-Modal Hashing (FCMH) [26] adds an extra element to estimate the binary code, making it better by reducing mistakes. Label-Attended Hashing (LAH) [30] initially generates embeddings for images and label co-occurrence separately. Following this, it employs a graph convolutional network (GCN) to combine label features with image features, enhancing the model's capabilities. Nevertheless, it's worth noting that LAH learns hash functions from specific real data samples.

Based on a recent research [18], we choose to adapt MuseHash as the state-of-the-art (SOTA) method and its competitor LAH [30] more suitable for 3D retrieval due its effectiveness in other domain-specific datasets. MuseHash estimates semantic probabilities and statistical properties during the retrieval process, enhancing its performance in capturing meaningful relationships within the data. It not only demonstrates its prowess in multimodal retrieval but also aligns perfectly with the complexities of 3D data.

In our exploration, we compare MuseHash with CMCL to evaluate their performance in multimodal query scenarios, ultimately solidifying MuseHash as the prime candidate for adapting to 3D retrieval. Our study focuses on the potential of volumetric retrieval, leveraging the flexibility and detail offered by different representations of data. By exploring the challenges and opportunities of 3D retrieval, we aim to advance the field using multiple modalities.

3 Methodology

To formally address the problem, we define the following scenario: Given a query object denoted as $\mathcal{Q}$ and a database $\mathcal{DB}$ comprising a collection of 3D objects

represented using varying views, such as images and meshes, the fundamental objective is to perform effective retrieval. This retrieval process aims to identify objects within $\mathcal{DB}$ that share similarities with the query $\mathcal{Q}$. The process involves a meticulous analysis of the distinctive features characterizing $\mathcal{Q}$ and the subsequent comparison of these features with corresponding attributes of objects within $\mathcal{DB}$ to ascertain pertinent matches.

Figure 2 provides a visual representation of the conceptual framework that underlies our research. This framework comprises three distinct phases: training, offline, and querying. During the training phase, data is input into a specific architecture, resulting in the generation of feature vectors. In the offline phase, features are extracted from a retrieval set and subsequently stored in a database for future reference. In the online phase, the architecture is applied to queries, and relevant results are retrieved from the database. To visually distinguish the areas where each studied model was applied, we've used an ochre color. Additionally, the presence of a blue color indicates instances where the model has a multimodal capability; otherwise, it is absent.

When it comes to 3D retrieval methods, both MeshNet and MeshCNN fall under the category of unimodal mesh-based techniques. CMCL, on the other hand, stands out as a cross-modal 3D retrieval method. It adopts a different approach by concurrently learning a shared space for various features from different sources, utilizing MeshNet, DGCNN, and ResNet for mesh, point-cloud, and image modalities, respectively.

In our adaptation of image retrieval techniques for 3D retrieval, we apply supervised hashing methods within the architecture to generate hash code features. Specifically, we've chosen LAH and MuseHash due to their proven effectiveness across various data types, as emphasized in recent research by [18]. LAH, initially designed for unimodal image retrieval, learns hash codes by applying a non-linear hash function to these mesh features using features from MeshNet. MuseHash, on the other hand, leverages the same models as CMCL to extract features from all modalities. Subsequently, MuseHash employs Bayesian ridge regression to learn hash functions, mapping feature vectors to the Hamming space, thereby enabling both unimodal and multimodal queries.

In our experiments, we transformed the different modalities into feature vectors to prepare them for the hashing methods. For the visual modality, we conducted an averaging process involving 180 multi-view image feature vectors extracted from ResNet50's fc-7 layer, generating a 2048-D vector. As for the point cloud and mesh modalities, we obtained 256-D vectors directly from the final layers of DGCNN and MeshNet, respectively.

4 Experiments

In this section, we begin by describing the datasets used for evaluation and providing an overview of the experimental setup. We then present detailed experimental results for a variety of modalities and hash code lengths.

4.1 Datasets

The evaluation of our method and the comparison with existing SOTA methods is done on the two publicly available datasets (Table 1):

BuildingNet_v0 The BuildingNet_$v0$ [20] provides high-quality annotations and diverse building types (like church, palace etc.)

ModelNet40 The ModelNet40 [28] is a large-scale 3D CAD model dataset, offering a wide range of object categories (like car, bottle, etc.).

Table 1. Two benchmark datasets used in experiments.

	Ground Truth	Collection Sizes			
Dataset	**Labels**	**Whole**	**Retrieval**	**Training**	**Test**
BuildingNet_v0	60	2000	1900	500	100
ModelNet40	40	12311	11696	4843	615

4.2 Experimental Settings

In our experiments, we evaluate the performance of two different types of methods using various metrics. Specifically, we consider the hashing methods, MuseHash [18] and the LAH[3] [30], where we examine the impact of different hash code lengths (d_c = 16, 32, 64, 128). For each volumetric method, we vary the number of epochs (epochs = 10, 50, 100, 150) used for computing each metric. We use the proposed training and testing size by the authors [28,20] for each dataset as suggested by the authors.

We compare our approach with two state-of-the-art 3D mesh methods MeshNet[4] [6], MeshCNN[5] [9], one cross-modal 3D retrieval method and CMCL[6] [11] in terms of mean Average Precision (mAP), precision at k (prec@k), recall at k (recall@k), f-score at k (fscore@k), accuracy and training time.

In the paper, a 5-fold cross-validation methodology was employed in all of the experiments for a more robust evaluation. We measured runtime of the experiments per epoch or per hash code length. Additionally, all 3D retrieval implementations used big amounts of memory, while MuseHash uses a small amount of memory (only some bits). In the following tables, the symbol '*' indicates that MuseHash, has demonstrated statistical significance compared to the other methods, as determined by a t-test.

[3]https://github.com/IDSM-AI/LAH
[4]https://github.com/iMoonLab/MeshNet
[5]https://github.com/ranahanocka/MeshCNN
[6]https://github.com/LongLong-Jing/Cross-Modal-Center-Loss

Table 2. MAP results for ModelNet40 and BuildingNet_v0 with different code lengths or number of epochs and mesh modality.

Dataset	No. Epochs	MeshNet [6]	MeshCNN [9]	CMCL [11]	Hash Length	LAH [30]	MuseHash [18]
ModelNet40	10	0.6801*	0.6726*	0.7097*	16	0.7811*	**0.8010**
	50	0.6954*	0.6900*	0.7099*	32	0.7889*	**0.8056**
	100	0.7091*	0.6711*	0.7103*	64	0.8001*	**0.8101**
	150	0.6654*	0.6502*	0.6695*	128	0.8058*	**0.8122**
BuildingNet_v0	10	0.6201*	0.6007*	0.6511*	16	0.7629*	**0.7723**
	50	0.6350*	0.6226*	0.6520*	32	0.7701	**0.7791**
	100	0.6552*	0.6449*	0.6670*	64	0.7754*	**0.7834**
	150	0.6550*	0.6501*	0.6623*	128	0.7821*	**0.7883**

4.3 Unimodal Retrieval Results

To study the performance of the multimodal approaches in unimodal situations, we compare all the aforementioned methods using only the mesh queries over the mesh modality. The results of those methods over the two datasets are given in Table 2 and Table 3 for mAP and accuracy, respectively. In this scenario, MuseHash outperforms all state-of-the-art methods.

Table 3. Accuracy results for ModelNet40 and BuildingNet_v0 with different code lengths or number of epochs and mesh modality.

Dataset	No. Epochs	MeshNet [6]	MeshCNN [9]	CMCL [11]	Hash Length	LAH [30]	MuseHash [18]
ModelNet40	10	0.8091*	0.7511*	0.7916*	16	0.9221*	**0.9431**
	50	0.8363*	0.8002*	0.8001*	32	0.9278*	**0.9488**
	100	0.8422*	0.8101*	**0.9791**	64	0.9312*	0.9500
	150	0.8490*	0.8091*	**0.9895**	128	0.9401	0.9510
BuildingNet_v0	10	0.7882*	0.7716*	0.7910*	16	0.9189*	**0.9323**
	50	0.8025*	0.7922*	0.8001*	32	0.9207*	**0.9344**
	100	0.8337*	0.8267*	0.8510*	64	0.9255*	**0.9390**
	150	0.8405*	0.8373*	0.8601*	128	0.9345*	**0.9401**

The MuseHash algorithm performs better than other methods for different hash lengths and epochs on both datasets. It shows the highest mAP and accuracy scores, highlighting its effectiveness for 3D retrieval tasks using only the mesh view. Particularly on the BuildingNet_v0 dataset, MuseHash achieves the best accuracy, surpassing other methods. Additionally, the CMCL approach achieves top accuracy on the ModelNet40 dataset with more epochs, yet its mAP

performance lags behind. This implies CMCL's proficiency in classification but potential challenges in organizing relevant retrieval results. Apart from that, image retrieval methods can perform better in 3D retrieval task from current SOTA 3D retrieval methods.

4.4 Multimodal Retrieval Results

For the multimodal case, we consider the combined utilization of point clouds, meshes, and multi-view images. The results of these techniques for both mAP and accuracy are detailed in Table 4 for ModelNet40 and BuildingNet_v0 dataset.

Specifically, the Table 4 highlights the mAP and accuracy results for ModelNet40 dataset across different hash lengths, epochs, and query modalities. MuseHash, demonstrates competitive performance in the majority of scenarios. MuseHash exhibits a distinct advantage in accuracy when queries involve both visual and point cloud modalities. While CMCL also exhibits competitive results, MuseHash's efficacy in handling diverse query modalities showcases its adaptability and robustness across different data representations.

In addition, the multimodal variant of MuseHash, which incorporates both mesh and image modalities, demonstrates substantial performance improvements with longer code lengths (from 16 to 32), particularly for larger code lengths (64 and 128). However, further increasing the code length does not lead to significant performance gains. This observation highlights an optimal code length range where MuseHash excels in capturing intricate multimodal relationships.

According BuildingNet_v0 dataset (Table 4), MuseHash outperforms in all multimodal cases the CMCL approach. In general, MuseHash has higher value on mAP as the code length increases and reaches the highest value when it uses visual and mesh view as a query and for code length 128.

4.5 Analysis of Runtime Requirements

The Figure 3 represents the training time (in minutes) for various methods, including MeshNet, MeshCNN, CMCL, LAH and three variants of MuseHash, across different training epochs or code lengths. Each line in the plot corresponds to a specific method, and the x-axis represents the training time in minutes, while the left y-axis the number of training epochs and the right y-axis the code length used in the training process. The black and grey dotted lines correspond to the values of each method for a specific epoch or code length, respectively. Particularly, there are three variants of the MuseHash method (MuseHash1, MuseHash2, and MuseHash3) evaluated for different code lengths, which correspond to the use of mesh, mesh and visual, and mesh, visual and point cloud view, respectively.

MeshNet and MeshCNN exhibit relatively shorter training times compared to CMCL and MuseHash variants, with CMCL requiring significantly more time for the same number of epochs. Among the MuseHash variants, 'MuseHash1' consistently shows the shortest training time across different code lengths, making it the most computationally efficient option. As the code length increases,

Table 4. MAP and accuracy results for ModelNet40 and BuildingNet_v0 with different code lengths or number of epochs and query modalities.

Dataset	Query	No. Epochs	CMCL[11] mAP	CMCL[11] Accuracy	Hash Length	MuseHash [18] mAP	MuseHash [18] Accuracy
ModelNet40	**Visual**	10	0.6911*	0.9012*	16	**0.8184**	**0.9501**
	Mesh	50	0.7010*	0.9045*	32	**0.8201**	**0.9578**
		100	0.7122*	0.9091*	64	**0.8234**	**0.9601**
		150	0.7415*	0.9129*	128	**0.8212**	**0.9525**
	Visual	10	0.6710*	0.7661*	16	**0.7712**	**0.9423**
	Point Cloud	50	0.6912*	0.7712*	32	**0.7821**	**0.9489**
		100	0.7010*	0.7891*	64	**0.7823**	**0.9510**
		150	0.7122*	0.7922*	128	**0.7840**	**0.9517**
	Mesh	10	0.6910*	0.8992*	16	**0.7882**	**0.9345**
	Point Cloud	50	0.7039*	0.9042*	32	**0.7910**	**0.9422**
		100	0.7128*	0.9188*	64	**0.7900**	**0.9577**
		150	0.7231*	0.9201*	128	**0.7854**	**0.9611**
	Visual	10	0.7097*	0.7916*	16	**0.8051**	**0.9611**
	Mesh	50	0.7099*	0.8001*	32	**0.7976**	**0.9601**
	Point Cloud	100	0.7103*	**0.9791**	64	**0.7923**	0.9583
		150	0.6695*	**0.9895**	128	**0.7911**	0.9550
BuildingNet_v0	**Visual**	10	0.6911*	0.8011*	16	**0.7810**	**0.9423**
	Mesh	50	0.7010*	0.8091*	32	**0.7912**	**0.9455**
		100	0.7122*	0.8123*	64	**0.8010**	**0.9589**
		150	0.7415*	0.8231*	128	**0.8091**	**0.9610**
	Visual	10	0.6801*	0.7938*	16	**0.7701**	**0.9244**
	Point Cloud	50	0.6881*	0.7957*	32	**0.7734**	**0.9301**
		100	0.6910*	0.8010*	64	**0.7791**	**0.9451**
		150	0.7001*	0.8139*	128	**0.7801**	**0.9510**
	Mesh	10	0.6761*	0.7810*	16	**0.7610**	**0.9181**
	Point Cloud	50	0.6810*	0.7910*	32	**0.7691**	**0.9201**
		100	0.6902*	0.8031*	64	**0.7701**	**0.9221**
		150	0.6971*	0.8091*	128	**0.7688**	**0.9200**
	Visual	10	0.6511*	0.7910*	16	**0.7790**	**0.9021**
	Mesh	50	0.6520*	0.8001*	32	**0.7800**	**0.8900**
	Point Cloud	100	0.6670*	0.8510*	64	**0.7881**	**0.8991**
		150	0.6623*	0.8601*	128	**0.7912**	**0.8920**

all variants of MuseHash experience longer training times due to more complex computations and increased memory demands. In summary, the Figure 3 compares training times among different methods. MuseHash1 exhibits the shortest training times initially, but as the code length increases, the training times become longer.

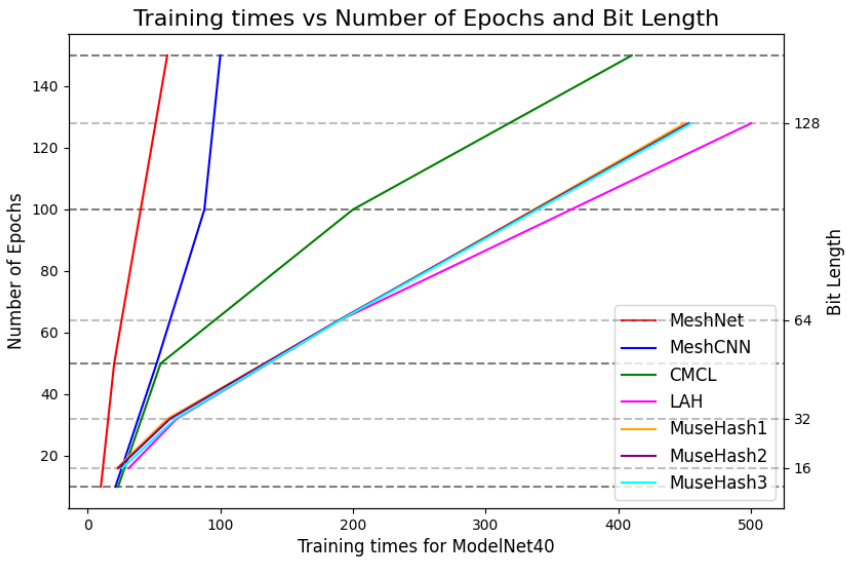

Fig. 3. Comparison of training times for all methods in minutes.

4.6 Discussion

Table 5 presents a comprehensive comparison of various methods based on Precision@k, Recall@k, and Fscore@k for k = 10, 25, 50 and for different code lengths or number of epochs on the ModelNet40 dataset. Thus, the table provides a detailed understanding of how well the retrieval methods rank and retrieve relevant items. Moreover, the selected metrics shed light on the methods' ranking mechanisms and their capacity to capture pertinent data points among the top-k results.

The methods are categorized into two groups: MeshNet, MeshCNN, and CMCL, each evaluated for different numbers of epochs. Additionally, there are the three predefined variants of the MuseHash (MuseHash1, MuseHash2, MuseHash3) method and the LAH method evaluated for different code lengths.

Therefore upon careful observation, it is evident that multimodal approaches excel in comparison to MeshCNN and MeshNet, revealing limitations in the architecture or feature representation of the latter two methods when operating exclusively within the mesh view.While CMCL occasionally achieves superior outcomes, considering the trade-off between performance gains and training time, MuseHash emerges as the more efficient choice. Additionally, MuseHash's capability to incorporate multiple modalities (e.g., mesh and image) into a unified hash code enhances its retrieval accuracy and diversity. The efficiency of MuseHash becomes particularly valuable in scenarios with extensive datasets and resource constraints, where fast and accurate similarity searches are paramount.

Table 5. Comparison of all methods based on Precision at k (k = 10, 25, 50) for different number of epochs or code lengths on ModelNet40 dataset

Method	Variable	Precision@k			Recall@k			Fscore@k		
	Epochs	**10**	**25**	**50**	**10**	**25**	**50**	**10**	**25**	**50**
MeshNet [6]	10	0.6510	0.6560	0.6410	0.6802	0.6533	0.6602	0.6653	0.6546	0.650
	50	0.6810	0.6712	0.6678	0.7011	0.7051	0.7187	0.6909	0.6877	0.6923
	100	0.6901	0.6854	0.6802	0.7029	0.7011	0.7089	0.6964	0.6932	0.6943
	150	0.7010	0.6910	0.6824	0.7091	0.7123	0.7189	0.7055	0.7015	0.7002
MeshCNN [9]	10	0.5822	0.5701	0.5623	0.5791	0.5607	0.5689	0.5806	0.6011	0.6178
	50	0.6001	0.5803	0.5734	0.5998	0.6011	0.6183	0.5999	0.5905	0.5950
	100	0.6245	0.6183	0.6002	0.6011	0.6190	0.6189	0.6126	0.6186	0.6094
	150	0.6221	0.6112	0.6009	0.6005	0.6123	0.6230	0.6111	0.6117	0.6118
CMCL [11]	10	0.8290	0.7679	0.7142	**0.9985**	0.9943	0.9968	0.9011	0.8666	0.8321
	50	0.8291	0.7687	0.7147	0.9883	0.9943	**0.9968**	0.9018	0.8671	0.8325
	100	**0.8298**	0.7687	0.7149	0.9884	**0.9944**	**0.9968**	**0.9019**	0.8671	0.8326
	150	0.8283	0.7677	0.7142	0.9865	**0.9944**	**0.9968**	0.9013	0.8665	0.8322
	Code Length	**10**	**25**	**50**	**10**	**25**	**50**	**10**	**25**	**50**
LAH [30]	16	0.6190	0.6179	0.6242	0.9215	0.9243	0.9268	0.7405	0.7407	0.7460
	32	0.6202	0.6287	0.6347	0.9383	0.9343	0.9461	0.7468	0.7516	0.7597
	64	0.6298	0.6287	0.6349	0.9584	0.9444	0.9468	0.7601	0.7549	0.7601
	128	0.6281	0.6271	0.6242	0.9265	0.9344	0.9468	0.7487	0.7505	0.7524
MuseHash [18] 1	16	0.6412	0.6501	0.6623	0.9567	0.9689	0.9781	0.7454	0,7781	0.7898
	32	0.6589	0.6620	0.6778	0.9612	0.9723	0.9612	0.7818	0.7877	0.8018
	64	0.6601	0.6789	0.6801	0.9667	0.9712	0.9789	0.7845	0,7992	0.8026
	128	0.6791	0.7123	0.7256	0.9701	0.9734	0.9601	0.7989	0.8229	0.8265
MuseHash [18] 2	16	0.6571	0.6810	0.7020	0.9612	0.9723	0.9865	0.7806	0.8010	0.8203
	32	0.6910	0.7001	0.7112	0.9546	0.9612	0.9667	0.8017	0.8101	0.8195
	64	0.7662	0.7405	0.7156	0.9712	0.9781	0.9801	0.8566	0.8429	0.8272
	128	0.8010	**0.8588**	**0.8423**	0.9865	0.9902	0.9923	0.8841	**0.9198**	**0.9112**
MuseHash [18] 3	16	0.6480	0.6501	0.6589	0.9523	0.9678	0.9621	0.7712	0.7778	0.7821
	32	0.6510	0.6678	0.6781	0.9678	0.9698	0.9512	0.7784	0.7910	0.7918
	64	0.6782	0.6789	0.6834	0.9701	0.9700	0.9634	0.7983	0.7988	0.7996
	128	0.7012	0.6910	0.6901	0.9701	0.9623	0.9603	0.8140	0.8044	0.8038

5 Conclusion

In this paper, we have leveraged the state-of-the-art methods from image retrieval to the domain of 3D object retrieval. In particular, we have adapted the recently proposed multimodal MuseHash method to support queries within volumetric data. The MuseHash method exploits the inner relations between different modalities. Our experiments show that MuseHash outperforms in most cases three state-of-the-art methods in both unimodal and multimodal queries across two different domain-specific benchmark image collections.

Acknowledgment

This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreement H2020-101070250 XRECO.

References

1. Brutto, M. L., Meli, P.: Computer Vision Tools for 3D Modelling in Archaeology. International Journal of Heritage in the Digital Era. https://doi.org/https://doi.org/10.1260/2047-4970.1.0.1, (2012).
2. Charles, R., Qi, H.S., Kaichun, M., Leonidas, J. G.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), openaccess.thecvf.com, (2017).
3. Dummer, M., Johnson, K., Rothwell, S., Tatah, K., Brenner, H. M.: The role of VCSELs in 3D sensing and LiDAR. SPIE 11692, Optical Interconnects XXI, https://doi.org/https://doi.org/10.1117/12.2577885, (2021).
4. Garland, M., Heckbert, P.S.: Simplifying surfaces with color and texture using quadric error metrics, Proceedings Visualization '98, https://doi.org/https://doi.org/10.1109/VISUAL.1998.745312, (1998).
5. Feng, Y., Feng, Y., You, H., Zhao, X., Gao, Y.: MeshNet: Mesh Neural Network for 3D Shape Representation, AAI Conference on Artificial Intelligence, vol. 33, 8279-8286, ojs.aaai.org, (2019).
6. Gezawa, A., Zhang, Y., Wang, Q., Yunqi, L.: A Review on Deep Learning Approaches for 3D Data Representations in Retrieval and Classifications, IEEE Access, https://doi.org/https://doi.org/10.1109/ACCESS.2020.2982196, (2020).
7. Ha, Q. P., Yen, L., Balaguer, C.: Robotic autonomous systems for earthmoving in military applications. Automation in Construction. https://doi.org/https://doi.org/10.1016/j.autcon.2019.102934, (2019).
8. Han, Y.-S., Lee, J., Lee, J., Lee, W., Lee, K.: 3D CAD data extraction and conversion for application of augmented/virtual reality to the construction of ships and offshore structures. International Journal of Computer Integrated Manufacturing. https://doi.org/https://doi.org/10.1080/0951192X.2019.1599440, (2019).
9. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: MeshCNN: a network with an edge. ACM Transactions on Graphics, vol. 38, 1-12, https://doi.org/https://doi.org/10.1145/3306346.3322959, (2019).
10. Javaid, M., Haleem, A., Singh, R.P., Suman, R.: Industrial perspectives of 3D scanning: Features, roles and it's analytical applications. Sensors International, vol. 2, https://doi.org/https://doi.org/10.1016/j.sintl.2021.100114, (2021).
11. Jing, L., Vahdani, E., Tan, J., Tian, Y.: Cross-Modal Center Loss for 3D Cross-Modal Retrieval, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3142-3151, openaccess.thecvf.com, (2021).
12. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. IEEE/CVF International Conference on Computer Vision (ICCV), openaccess.thecvf.com, (2017).
13. Lin, D., Li, Y., Cheng, Y., Prasad, S., Nwe, T.L., Dong, S., Guo, A: Multi-view 3d object retrieval leveraging the aggregation of view and instance attentive features, Knowl. Based Syst., vol. 247, https://doi.org/https://doi.org/10.1016/j.knosys.2022.108754, (2022).
14. Maturana, D., Scherer, S.: A 3D convolutional neural network for real-time object recognition, IEEE/RSJ International Conference on Intelligent Robots and Systems, 992-928, https://doi.org/https://doi.org/10.1109/IROS.2015.7353481, (2015).
15. Maglo, A., Lavoué , G., Dupont, F., Hudelot, C.: 3D Mesh Compression: Survey, Comparisons, and Emerging Trends, ACM Computing Surveys, vol. 47, 1-41, https://doi.org/https://doi.org/10.1145/2693443, (2015).

16. Mohr, E., Thum, T., Bär, C.: Accelerating cardiovascular research: recent advances in translational 2D and 3D heart models. European Journal of Heart Failure, vol. 24, https://doi.org/https://doi.org/10.1002/ejhf.2631, (2022).
17. Pal, P., Ghosh, K. K.: Estimating digitization efforts of complex product realization processes. The International Journal of Advanced Manufacturing Technology, vol. 95, https://doi.org/https://doi.org/10.1007/s00170-017-1442-3, (2018).
18. Pegia, M., Jónsson, B. Þ., Moumtzidou, A, Gialampoukidis, I., Vrochidis, S., Kompatsiaris, I.: MuseHash: Supervised Bayesian Hashing for Multimodal Image Representation, ACM International Conference on Multimedia Retrieval (ICMR), https://doi.org/https://doi.org/10.1145/3591106.3592228, (2023).
19. Rahate, A., Walambe, R., Ramanna, S., Kotecha, K.: Multimodal Co-learning: Challenges, applications with datasets, recent advances and future directions. Information Fusion. 203-209. https://doi.org/https://doi.org/10.1016/j.inffus.2021.12.003, (2022).
20. Selvaraju, P., Nabail, M., Loizou, M., Maslioukova, M., Averkiou, M., Andreou, A., Chaudhuri, S., Kalogerakis, E.: BuildingNet: Learning to Label 3D Buildings, IEEE/CVF International Conference on Computer Vision (ICCV), openaccess.thecvf.com, (2021).
21. Su, H., Maji, S., Kalogerakis, E., Learnined-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition, IEEE Internation Conference on Computer Vision (ICCV), 945-953, cv-foundation.org, (2015).
22. Su, J.-C., Gadelha, M., Wang, R., Maji, S.: A deeper look at 3d shape classifiers. European Conference on Computer Vision (ECCV), openaccess.thecvf.com, (2018).
23. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convulutional architectures for high-resolution 3d outputs. IEEE Internation Conference on Computer Vision (ICCV), 945-953, openaccess.thecvf.com, (2017).
24. Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X.: O-CNN: Octree-based convolutional neural networks for 3D shape analysis, ACM Transactions on Graphics (TOG), vol. 72, https://doi.org/https://doi.org/10.1145/3072959.3073608, (2018).
25. Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J.: Dynamic graph cnn for learning on point clouds, ACM Transactions on Graphics (TOG), vol. 38, 1-12, https://doi.org/https://doi.org/10.1145/3326362, (2019).
26. Wang, Y., Chen, Z.-D., Xin, L., Li, R., Xu, X.-S.: Fast Cross-Modal Hashing With Global and Local Similarity Embedding, IEEE Transactions on Cybernetics, https://doi.org/https://doi.org/10.1109/TCYB.2021.3059886, (2021).
27. Willemink, M. J., Koszek, W. A., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., Folio, L. R., Summers, R. M., Rubin, D. L., Lungren, M. P.: Preparing Medical Imaging Data for Machine Learning. Radiology. https://doi.org/https://doi.org/10.1148/radiol.2020192224, (2020).
28. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A Deep Representation for Volumetric Shapes, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1912-1920, cv-foundation.org, (2015).
29. Wu, W., Qi, Z., Fuxin, L.: PointConv: Deep Convolutional Networks on 3D Point Clouds, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9621-9630, openaccess.thecvf.com, (2019).
30. Xie, Y., Liu, Y., Wang, Y., Gao, L., Wang, P., Zhou, K.: Label-attended hashing for multi-label image retrieval, IEEE Transactions on Cybernetics, https://doi.org/https://doi.org/10.1109/TCYB.2021.3059886, (2020).
31. Zhan, Y.-W., Wang, Y., Sun, Y, Wu, X.-M., Luo, X., Xu, X.-S.: Discrete online cross-modal hashing, Pattern Recognition, vol. 122, https://doi.org/https://doi.org/10.1016/j.patcog.2021.108262, (2022).