

# Using Learned Visual and Geometric Features to Retrieve Complete 3D Proxies for Broken Objects

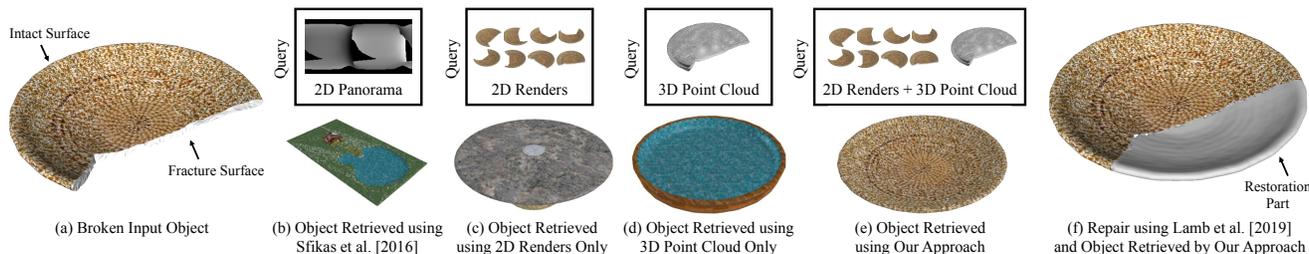
Nikolas Lamb  
lambne@clarkson.edu  
Clarkson University  
Potsdam, New York, USA

Noah Wiederhold  
wiedern@clarkson.edu  
Clarkson University  
Potsdam, New York, USA

Benjamin Lamb  
bt15104@rit.edu  
Rochester Institute of Technology  
Rochester, New York, USA

Sean Banerjee  
sbanerje@clarkson.edu  
Clarkson University  
Potsdam, New York, USA

Natasha Kholgade Banerjee  
nbanerje@clarkson.edu  
Clarkson University  
Potsdam, New York, USA



**Figure 1:** In this work we provide an approach to retrieve complete proxies to broken query object models for the purpose of object repair. Given (a) a 3D model of a broken query object, (b) prior approaches retrieve inaccurate proxies [Sfikas et al. 2016], (c) using 2D renders alone provides outputs that are geometrically dissimilar, and (d) using 3D point clouds only eliminates discriminative information such as surface color. Our approach combines the advantages of 2D renders and 3D point clouds to retrieve (e) complete proxies that demonstrate visual and geometric similarity to the original input. (f) Our approach can be leveraged to generate restorations for objects using automated restoration approaches [Lamb et al. 2019].

## ABSTRACT

3D printing offers the opportunity to perform automated restoration of objects to reduce household waste, restore objects of cultural heritage, and automate repair in medical and manufacturing domains. We present an approach that takes a 3D model of a broken object and retrieves proxy 3D models of corresponding complete objects from a library of 3D models, with the goal of using the complete proxy to repair the broken object. We input multi-view renders and point cloud representations of the query to neural networks that output learned visual and geometric feature encodings. Our approach returns complete proxies that are visually and geometrically similar to the broken query object model by searching for the learned encodings in the complete models library. We demonstrate results for retrieval of complete proxies for broken object models with breaks generated synthetically using models from the ShapeNet dataset, and from publicly available datasets of

scanned everyday objects and cultural heritage objects. By combining visual and geometric features, our approach shows consistently lower Chamfer distance than when either feature is used alone. Our approach outperforms the existing state-of-the-art method in retrieval of proxies for broken objects in terms of the Chamfer distance. The 3D proxies returned by our approach enable understanding of object geometry to identify object portions requiring repair, to incorporate user preferences, and to generate 3D printable restoration components. Our code to perform broken object model generation, feature extraction, and object retrieval is available at <https://git.io/JuKaJ>.

## CCS CONCEPTS

• Computing methodologies → Shape analysis; Mesh models; Feature selection; • Information systems → Multimedia and multimodal retrieval.

## KEYWORDS

Broken Object Retrieval, Computational Fabrication, Object Restoration, Learned Features

## ACM Reference Format:

Nikolas Lamb, Noah Wiederhold, Benjamin Lamb, Sean Banerjee, and Natasha Kholgade Banerjee. 2021. Using Learned Visual and Geometric Features to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SCF '21, October 28–29, 2021, Virtual Event, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9090-3/21/10...\$15.00

<https://doi.org/10.1145/3485114.3485118>

Retrieve Complete 3D Proxies for Broken Objects. In *Symposium on Computational Fabrication (SCF '21), October 28–29, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3485114.3485118>

## 1 INTRODUCTION

3D printing technology facilitates repair of damaged objects to mitigate environmental waste, restore cultural heritage objects [Gregor et al. 2014; Papaioannou et al. 2017], enable rapid repair in remote locations with limited access to spare parts, and facilitate domain-specific repair in dentistry, manufacturing, and medicine [Arbace et al. 2013; Singare et al. 2008]. Traditional restoration of objects has been largely manual, with users utilizing computer-aided design tools to create ‘fill-in’ objects using 3D scans of damaged objects. The manual restoration process is painstaking and requires expert knowledge of computer-aided design, which falls outside the scope of the average consumer. Users are most likely to throw out objects if they are required to generate repair parts by hand. Some automated approaches avoid 3D printing and provide the user with assembly instructions to recombine the broken object parts into a complete object [Mavridis et al. 2015; Zhang et al. 2018]. These approaches cannot complete the object if parts have been lost or destroyed. Approaches have been proposed to democratize object repair using 3D printing by automatically completing the object structure from a 3D scan of the broken object, and extracting the restoration as a subtraction [Gregor et al. 2014; Lamb et al. 2019; Papaioannou et al. 2017; Sipiran 2018]. Some approaches require that the objects have obvious axes of symmetry and asymmetrical fractures [Gregor et al. 2014; Papaioannou et al. 2017; Sipiran 2018], which severely limits the types of objects that can be processed. The approach of Lamb et al. [2019] requires users to provide a 3D scan of a complete object corresponding to the broken object. While their approach is more generalizable than symmetry-based methods, a user may not have ready access to a complete object scan, especially if the user only had a single instance of the object that is now broken. Given a broken object model such as the one shown in Figure 1(a), we provide an approach that automatically searches in a library of publicly available 3D models for a complete proxy 3D model, as shown in Figure 1(e). The complete proxy may be used to generate a restoration part for the broken object as shown in Figure 1(f). Our method reduces the complexity of object repair, enabling an average user to perform repairs in the field, restore delicate cultural heritage objects, and mend single-instance or discontinued objects while reducing material waste.

Current work in automated object model retrieval addresses searching for complete models in a library given a complete or incomplete object model as input by encoding the object models into a shared feature space [Aono and Iwabuchi 2020; Avetisyan et al. 2019a,b; Esteves et al. 2018; Iwabuchi and Aono 2018; Nie et al. 2019; Qi et al. 2016; Uy et al. 2020; Wang et al. 2014; Zhu et al. 2015]. Incomplete object models correspond to representations that observe a part of the object, such as single view images and depth maps. Broken objects are more challenging to encode as they contain an additional fracture surface, as shown in Figure 1(a), and as a result any encoding must be applied selectively or be robust enough to mitigate the effect of the fracture. A number of approaches address the difference between incomplete and

complete object geometry or appearance by training feature encoders directly on incomplete object depth scans [Esteves et al. 2018; Uy et al. 2020; Zhu et al. 2015], incomplete object point clouds generated from depth scans [Avetisyan et al. 2019a,b], or single RGB images of incomplete objects [Nie et al. 2019]. Adapting these methods to the problem considered in our work requires training with a large dataset of broken object models, which makes them difficult to rapidly propagate to new objects. To eliminate a training phase involving incomplete object models, some methods use handcrafted features [Funkhouser et al. 2004; Lavoué 2012; Li et al. 2014; Rock et al. 2015], which lack the accuracy of learned features, while other approaches use learned feature encoders that are trained on representations extracted from complete object models, e.g., 3D representations such as point clouds [Aono and Iwabuchi 2020; Iwabuchi and Aono 2018; Uy et al. 2020], voxels [Qi et al. 2016; Wang et al. 2014] or 2D or 2.5D representations such as multi-view renders [Qi et al. 2016]. These approaches compress the entire set of representations into a single feature vector, and are likely to encode the fracture surface into the feature space, thereby introducing a large mismatch between the broken object and its complete counterpart. To the best of our knowledge, only a single approach, i.e., that of Sfikas et al. [2016], performs retrieval of proxies on broken objects, using models of cultural heritage objects with synthetically generated breaks. Their approach uses bag of features (BoF) with handcrafted features, known to provide poor performance in contrast to machine learned features [Avetisyan et al. 2019a].

In this work, we provide an approach that leverages a combination of learned visual and geometric features extracted by running pre-trained neural network encoders on multi-view renders and point cloud representations of broken object models to perform retrieval of complete proxy models from a dataset. Querying with multiple independent renders minimizes the impact of the break on visual features, as viewpoints that do not observe the fracture surface on the broken object are mapped closer to similar viewpoints of visually comparable complete proxies in feature space. Encoding multi-view renders separately overcomes the limitations that methods which integrate a full set of representations into a single feature vector [Aono and Iwabuchi 2020; Iwabuchi and Aono 2018; Qi et al. 2016; Uy et al. 2020; Wang et al. 2014] encounter when using broken query object models as input. Using multi-view renders also incorporates surface texture, which provides a visual indicator of the fracture surface if the object interior is colored differently from its surface, as is the case in Figure 1(a). However, performing retrieval using only visual features may not encode crucial geometric information, such as geometry on the reverse side of the object, and may retrieve objects that are geometrically dissimilar to the broken object, as shown in Figure 1(c). While using geometric features encodes shape-related information, geometric features alone disregard discriminative information such as surface color. They may retrieve objects that belong to a different class from the broken object or that have a different texture, as shown Figure 1(d). Our approach combines the strengths of visual and geometric features to retrieve complete proxies with significantly closer geometric similarity in terms of Chamfer distance and normal consistency and a higher mean average precision than using either feature independently, as demonstrated by our results in

Section 5. We use VGG16 [Simonyan and Zisserman 2014] to extract visual features. We evaluate PointNet++ [Qi et al. 2017] and histograms created from a 3D local feature generator provided by Choy et al. [2019] to represent geometric features. The complete proxy object model returned by our approach can be combined with an automated restoration approach, e.g. the approach of Lamb et al. [2019], to repair the object, as shown in Figure 1(f), without requiring the user to scan a corresponding complete object. We summarize our contributions below.

- We present the first approach to use learned visual and geometric features to retrieve complete proxy object models using broken models as queries. We show that our approach outperforms previous work in complete proxy model retrieval using broken queries. Our approach does not require a training phase involving broken objects, and can be expanded to arbitrarily many object classes without re-training.
- We perform extensive evaluation of retrieval in terms of Chamfer distance, normal consistency, and mean average precision for class retrieval using synthetically broken object models from three datasets—ShapeNet [Chang et al. 2015], A Large Dataset of Scanned Objects [Choi et al. 2016], and cultural heritage objects from the Hampson Museum dataset [Payne et al. 2009].

We generate evaluation datasets containing 5,552 broken and 28,832 complete object models spanning 31 common object classes derived from the ShapeNet [Chang et al. 2015] dataset, 94 broken and complete object models spanning 7 classes derived from the dataset of Choi et al. [2016], and 25 broken and 122 complete object models derived from the Hampson Museum dataset [Payne et al. 2009]. We ensure that each object model is waterproofed and we apply textures where applicable. We generate breaks synthetically by subtracting a random spherical or cubical geometric primitive from the object. Our work outperforms the existing state-of-the-art method on retrieval of proxies for broken objects [Sfikas et al. 2016] in terms of the Chamfer distance using synthetically broken objects from ShapeNet, Choi et al. [2016], and the Hampson Museum dataset. We share our code to replicate our experiments at <https://git.io/JuKaJ>.

## 2 RELATED WORK

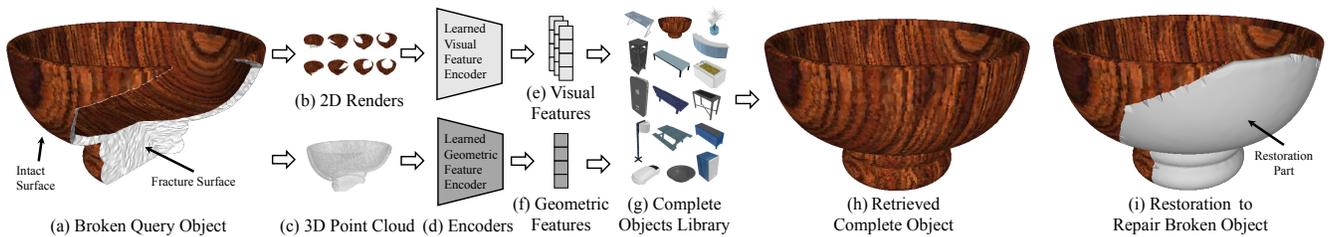
### 2.1 Retrieval of Complete Proxies for Complete and Incomplete Objects

Early work in object retrieval uses handcrafted features to retrieve proxies for input objects whose representations may either be complete, e.g., a full 3D model or a set of multi-view renders spanning the object, or incomplete, e.g., a single-view depth map or RGB image, or a partial 3D model. Funkhouser et al. [2004] perform complete shape retrieval by encoding a partial 3D model as a single feature that approximates the the Euclidean distance but is faster to compute. When used with broken models rather than partial models, approaches that encode the entire query object into a single vector will include the broken model’s fracture surface in the encoding, causing them to return less relevant complete proxies. Bronstein et al. [2011] provide a complete shape retrieval method based on bag of features (BoF) that incorporates the context in which features appear by defining each object using feature

pairs. Litman et al. [2014] take this concept further by grouping local features and then performing a sparse coding and pooling operation on these groups to get the final encoding. In the case of retrieving proxies for incomplete object models, the approach of Lavoue [2012] uses a BoF approach that encodes regions of incomplete object models rather than the object as a whole. Johan et al. [2014] create a hybrid shape descriptor for incomplete object model matching that includes information about local and global object features, and uses particle swarm optimization to intelligently combine the feature distances. Toldo et al. [2010] create geometric features that represent segments of a query object and train a support vector machine to categorize the complete or incomplete query model without comparing it to the entire object library. Rock et al. [2015] perform incomplete object retrieval on depth maps using handcrafted features based on object silhouettes and voxel representations. Other retrieval approaches use descriptors such as Reeb graphs [Barra and Biasotti 2013; Biasotti et al. 2008], eigenfunctions [Wang and Lin 2020], and covariance matrices [Tabia and Laga 2015] for retrieval of proxies from incomplete objects [Barra and Biasotti 2013; Biasotti et al. 2008; Tabia and Laga 2015] and complete objects [Tabia and Laga 2015; Wang and Lin 2020].

Recent approaches have used machine learning to extract expressive features for object retrieval and classification or to augment existing handcrafted features. Many approaches address proxy retrieval for complete objects by using 2D convolutional neural networks (CNNs) to extract features from multi-view renders [Bai et al. 2016; Han et al. 2019; He et al. 2018; Jiang et al. 2019; Su et al. 2015], stereographic renders [Yavartanoo et al. 2018], or panoramic renders [Sfikas et al. 2018; Shi et al. 2015]. These methods integrate information from multi-view renders into a compressed feature vector that encodes information from the entire object. If applied directly to broken object models, the methods are likely to provide compressed vectors that are dissimilar from those of their counterpart complete objects, limiting reliable retrieval. Work exists on complete object retrieval using 3D features, including methods that train 3D CNNs [Fu et al. 2020; Furuya and Ohbuchi 2016; Qi et al. 2016] and spherical CNNs [Esteves et al. 2018], though these approaches have historically performed worse than those based on multi-view rendering [Esteves et al. 2018], and, similar to multi-view rendering work, may provide encoded vectors for broken objects that are distinct from their complete counterparts.

In the area of using machine learning to retrieve proxies for incomplete objects, Nie et al. [2019] train a network to perform object retrieval using a synthetically generated real image of the query object. Zhu et al. [2015] train a Siamese Network to perform 3D object retrieval from a depth map, and Avetisyan et al. [2019b] train a network for simultaneous point cloud cropping and retrieval. Avetisyan et al. [2019a] present an approach for incomplete object proxy retrieval that requires every encoded library object to be compared to every depth scan, which is infeasible with a large set of objects. Using these methods for broken objects requires re-training the networks on a dataset containing broken object models, which limits their adaptability to novel objects. To avoid including the incomplete object in the training phase, Wang et al. [2014] encode local features in a conditional random field and embed geometric structures into a regression tree field. Iwabuchi et al. [2018] construct a voxel representation from local features,



**Figure 2: Overview of our approach.** (a) Given the 3D model of a query broken object as input, our approach generates (b) multiple 2D renders and (c) a point cloud representation (with normals) for the object, (d) using neural network-based encoders to extract learned (e) visual and (f) geometric features, matches the features to those extracted from (g) a library of complete objects, and (h) retrieves the complete object model with the closest match. We use the complete object model to generate (i) a repair part for the broken object using the automated restoration approach of Lamb et al. [2019].

and use a 3D CNN to encode incomplete objects. Aono et al. [2020] partition complete object point clouds and encode them into a library. Uy et al. [2020] train an encoder on sets of similar objects such extracted features are invariant to deformation. Qi et al. [2016] train 3D CNNs on complete object voxels and test their method on incomplete depth scans. Similar to work on retrieval for complete objects, these approaches encode the object’s entire information into a single feature vector. If applied to broken object models they are likely to encode the fracture region and generate feature vectors that are dissimilar from the complete object. We address match discrepancy due to the fracture region by using a learned visual encoder based on VGG16 [Simonyan and Zisserman 2014] to encode multiple renders of the object independently, and selecting the best matching view as per the distance between visual features. Since the best matching view may lack geometric information for high accuracy, we incorporate matches from features extracted using geometric encoders [Chen et al. 2021; Choy et al. 2019].

## 2.2 Proxy Retrieval for Broken Objects

Despite its importance in the object repair pipeline, work on broken object proxy retrieval is limited. To the best of our knowledge, the only work in this domain is that of Sfikas et al. [2016], who demonstrate results using the cultural heritage objects from the Hampson Museum dataset [Payne et al. 2009]. They evaluate broken object models generated by subjecting the dataset objects to synthetic fractures. They render cultural heritage objects as panoramic depth images, extract handcrafted features in the form of SIFT [Lowe 2004] and local depth histograms for randomly selected keypoints on the image, and encode the features using BoF. Our approach uses learned feature encoders due to their demonstrated superiority over hand-crafted features [Avetisyan et al. 2019a]. Learned visual features from 2D renders of the broken query object enable retrieval of high-quality complete objects even if the query is not easily identifiable from its geometry due to large fractures. Our approach outperforms the work of Sfikas et al. in terms of Chamfer distance, normal consistency, and mean average precision.

## 2.3 Object Completion

Another approach to obtain a complete object given an incomplete or broken object as input is to generate the complete object

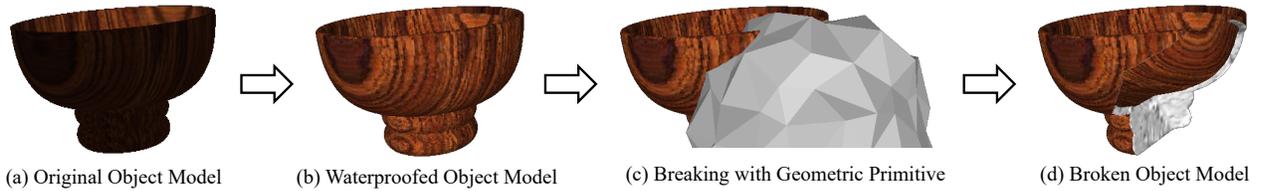
directly, without the use of a library. Historically, object completion algorithms have made use of object symmetry to complete an object [Gregor et al. 2014; Sipiran 2018]. These approaches require that the damage to the object is asymmetric, and fail if the object to be repaired does not exhibit symmetry or if the damage is great enough that a plane of symmetry cannot be found. Other approaches use data-driven techniques to generate a complete object directly from images [Groueix et al. 2018; Mescheder et al. 2019] or partial depth maps [Dai et al. 2018; Park et al. 2019; Sarmad et al. 2019; Son and Kim 2020]. However, data-driven approaches fail to generate satisfactory completions for atypical objects, and must be re-trained entirely using pairs of complete and incomplete or broken objects, limiting their extensibility to new object classes. Our approach performs well for atypical broken objects as these objects are more easily matched to similar library objects, and can be scaled to arbitrarily many new classes without training.

## 3 RETRIEVING COMPLETE PROXIES FOR BROKEN OBJECTS

The goal of our approach is to retrieve a complete proxy object model from a library of object models that can be used to generate a restoration part for a broken query object model. Given a broken query such as the one shown in Figure 2(a), we render the object model from multiple viewpoints as shown in Figure 2(b), extract a point cloud with point normals from the model as shown in Figure 2(c), and encode the renders and point cloud using learned feature encoders as shown in Figure 2(d) to create visual and geometric features as shown in Figures 2(e) and 2(f). We discuss our learned visual and geometric feature extraction approach in Section 3.1. We use the encoded features to query a library of features extracted from complete object models, and we extract the closest object model in the library as shown in Figure 2(h). Section 3.2 discusses our retrieval approach. Figure 2(i) shows a restoration part generated by applying the approach of Lamb et al. [2019] to the retrieved proxy object model and the broken query object model.

### 3.1 Extracting Learned Features

Given a 3D model of a broken object, our approach generates 8 multi-view renders at 640x640 pixel resolution for visual feature extraction. We uniformly space the multi-view render viewpoints



**Figure 3: (a) Given the original object model, (b) we use the approach of Stutz and Geiger [2020] to waterproof the model. (c) We randomly position a primitive break object and perform Boolean subtraction to generate (d) the broken model.**

around the object to increase the chance of having viewpoints that observe the object’s intact surface. We orient the viewpoints  $35^\circ$  downwards to represent typical object viewing angles. We run the VGG16 [Simonyan and Zisserman 2014] neural network on each multi-view render and extract the features of the second-to-last layer to create a set of 8 visual feature vectors, one per multi-view render. We create a point cloud representation of the object augmented with normals, in order to generate geometric features. We evaluate two types of geometric features in this work. For the first type of feature, we run PointNet++ [Qi et al. 2017] on the object’s point cloud and normals, and extract features at the second-to-last layer to obtain a global feature descriptor for the object. For the second feature type, we run the 3D CNN from Choy et al. [2019] on the point cloud and extract the features at the last layer to generate a set of local feature vectors. Given the local vectors, we use BoF to generate a single vector by clustering the local vectors into a 256 cluster-codebook using  $k$ -means, and binning counts of the local vectors in each cluster into a histogram. We compute the codebook over a subset of 10% of the objects in the library. Given a library of complete object models, we repeat these steps on each complete model to get visual and geometric features.

### 3.2 Proxy Retrieval Using Learned Features

For each visual feature from the 8 multi-view renders of the query broken object model, we compute the  $L_2$  distance to its  $K$  nearest neighbor visual features in the library where  $K = 2,048$ . We obtain the library object index to which the feature belongs. We perform nearest neighbor searches using Faiss [Johnson et al. 2019]. The nearest neighbor feature retrieval provides  $8K$  distances and library object indices for the visual features of the query. To eliminate repeated library object indices, we retain all unique object indices in the set  $\mathcal{I}_v$ . We store the smallest distance for each unique object index in the tuple-set  $\mathcal{I}_{dv}$ , with each tuple containing a library index from  $\mathcal{I}_v$  and the corresponding smallest distance. For each geometric feature from the point cloud of the object, we obtain  $K$  nearest neighbor geometric features from the library. We record the nearest neighbor distance and library object index for each geometric feature. The process provides  $K$  library object indices stored in set  $\mathcal{I}_g$  and  $K$  corresponding distances stored in the tuple-set  $\mathcal{I}_{dg}$ . Each tuple contains a library object index from  $\mathcal{I}_g$  and the corresponding distance. Since there is a single geometric feature per object, each object index is unique for the geometric features. To combine the visual and geometric feature distances, we extract common library object indices by obtaining the set  $\mathcal{I} = \mathcal{I}_v \cap \mathcal{I}_g$ . For each object index  $i \in \mathcal{I}$ , we locate its corresponding tuples in

$\mathcal{I}_{dv}$  and  $\mathcal{I}_{dg}$ . We extract the distances  $d_{vi}$  and  $d_{gi}$  from the tuples. We obtain the total match for the library object index  $i$  as

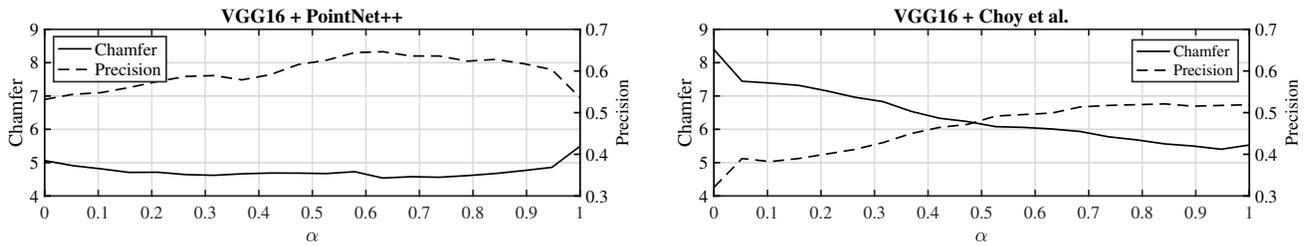
$$d_i = \alpha d_{vi} + (1 - \alpha) d_{gi}, \quad (1)$$

where  $\alpha$  is a user-defined parameter used to weight relative the importance of the visual and geometric features. We return the library object index with the smallest value of  $d_i$  over all indices in  $\mathcal{I}$  as the closest matching object model to the broken query object. Our approach also enables returning the top  $n$  library objects by ranking the distances over all indices in  $\mathcal{I}$  in increasing order, and returning the  $n$  indices with the smallest distances. We evaluate various values of  $\alpha$  in Section 5 and present results with the value of  $\alpha$  that provides the lowest Chamfer distance.

## 4 BROKEN AND COMPLETE DATASETS

We use three datasets to evaluate our object retrieval approach—ShapeNet [Chang et al. 2015], A Large Dataset of Object Scans [Choi et al. 2016], and the Hampson Museum dataset of cultural heritage (CH) objects [Payne et al. 2009]. ShapeNet contains object classes for household objects such as jars, mugs, bowls, pots, and toys, as well as large objects such as cars, tables, and chairs. The dataset of Choi et al. [2016] contains scanned models of objects such as chairs, tables, trashcans, benches, and motorcycles. The scanned models contain extraneous information such as the ground plane and walls. We clean each scan by scaling the scan to fit inside a unit cube, computing a plane corresponding to the ground plane using random sample consensus [Fischler and Bolles 1981], and using the ground plane to orient the object upright. To isolate the object, we retain all points that belong to the largest connected component, lie above the ground plane, and are contained in a cube of side 0.6 units at the scan centroid. We manually inspect objects, and discard those that are incorrectly isolated. The CH dataset contains objects such as clay pots, bowls, and jars. While the object models contain some damage due to being archaeological items, we subject them to additional breaks to generate queries, similar to Sfikas et al. [2016].

We require waterproofed object models to generate synthetic breaks using Boolean subtraction. Given an object model from each dataset, we use the approach of Stutz and Geiger [2020] to generate waterproofed models as shown in Figure 3(b). To retain high-fidelity texture after waterproofing, we iterate through the vertices in the waterproofed mesh, and transfer the interpolated texture value at the vertex with the closest aligned normal from the two nearest neighbor vertices in an unsampled version of the original mesh. Ensuring that the normals are aligned reduces the



**Figure 4: Left: Chamfer distance and precision over varying  $\alpha$  for ‘VGG16 + PointNet++’. Right: Chamfer distance and precision over varying  $\alpha$  for ‘VGG16 + Choy et al.’ We demonstrate results using the value of  $\alpha$  that provides the lowest Chamfer distance. We observe that this value of  $\alpha$  also provides high mAP. Chamfer distance shown has been multiplied by 100.**

chance of transferring vertex colors from inverted faces in the non-waterproofed models. Figure 3(c) demonstrates our approach to generate a break by subtracting a geometric primitive from the waterproofed object model. We use a synthetically generated convex geometric primitive cube, icosphere, or a subdivided icosphere to fracture objects, as we observe that broken objects typically have concave or shear fractures. For each object, we select a geometric primitive and randomly translate the vertices of the primitive in the range  $[-0.05, 0.05]$ , picked from a uniform distribution. We apply a random rotation and translation to the object and subdivide the surface of the object to a max edge length of 0.025. To simulate the complexity of real fractures, we randomly translate the vertices of the subdivided primitive in the range  $[-0.00125, 0.00125]$ , picked from a uniform distribution. We subtract the primitive from the object using Boolean subtraction. If a break removes more than 50% or less than 30% of an object, the primitive is moved or regenerated a maximum of 15 times before the object is discarded. We discard any models that are not waterproof after being synthetically broken. The fracture surface is colored uniformly white. Figure 3(d) shows the broken object model generated as output.

We use waterproofed complete and broken models from the three datasets to conduct four sets of experiments. Our first experiment tests retrieval success when a replica of the query object is present in the library. The experiment uses a complete objects library subset of ShapeNet objects with 28,832 models, and a broken objects query subset of 5,552 models that are present in the library subset. Our second, third, and fourth experiments test retrieval success when an exact replica of the query object is not present in the library. For the second experiment, the broken objects query subset remains the same, i.e., 5,552 models from ShapeNet, however, the complete objects library subset now contains 28,832-5,552 or 23,280 models that are mutually exclusive of the query subset. For the third experiment, we use the 94 scanned objects from the dataset of Choi et al. [2016] as the query subset, and we compare the query against the original subset of 28,832 ShapeNet models as the complete objects library. The fourth experiment compares a subset of 25 CH models as query against a mutually exclusive library subset of 97 CH models.

## 5 RESULTS

To evaluate the performance of the approaches tested in this work we use (a) the Chamfer distance as defined by Park et al. [2019],

(b) the normal consistency as defined by Mescheder et al. [2019], and (c) the mean Average Precision (mAP) over all query object classes. Unless otherwise specified, we compute each metric with respect to the top-1 object retrieved by each approach. For all experiments we use the VGG16 architecture and weights included in TensorFlow, with weights obtained through training on ImageNet [Deng et al. 2009]. We use the PointNet++ architecture and weights provided by Yan [2019] with training performed using ModelNet [Wu et al. 2015]. For the approach of Choy et al. [2019], we use the architecture and weights provided by Choy and Lee [2019] with training performed on the 3DMatch dataset [Zeng et al. 2017]. When constructing the BoF used to aggregate features from Choy et al., we use 10% of the objects in the library subsets obtained from ShapeNet, and all the objects in the library subset obtained from the CH dataset. We test two configurations of our approach, one that combines visual features using VGG16 with geometric features using PointNet++, which we refer to as ‘VGG16 + PointNet++’, and one that combines VGG16 features with geometric features generated by the approach of Choy et al. [2019], which we refer to as ‘VGG16 + Choy et al.’ We determine  $\alpha$  by performing a line search in the range  $[0, 1]$  and select the value that returns the smallest Chamfer distance between each query object and the top ranked library object. We perform this search on a subset of data containing 10% of the objects from the second ShapeNet library and query subsets, and find our model achieves the lowest Chamfer distance at  $\alpha = 0.64$  for ‘VGG16 + PointNet++’ and  $\alpha = 0.95$  for ‘VGG16 + Choy et al.’ as shown in Figure 4. We observe that the value of  $\alpha$  with lowest Chamfer distance also provides high mAP over all classes in the ShapeNet query subset. We also demonstrate results from using VGG16, PointNet++, and Choy et al. independently, which we term ‘VGG16’, ‘PointNet++’, and ‘Choy et al.’ We perform testing on a 40-core Intel Xeon server with 2 NVIDIA 3090s.

### 5.1 Retrieval for ShapeNet Query Objects using ShapeNet Objects in Library

We show results of our first and second experiments on complete object retrieval from the ShapeNet library subsets for broken object models that are present and absent from the library in the second and third super-columns of Table 1. As shown in the table, we report the Chamfer distance, the normal consistency, and the mAP. For both cases, i.e., when the query is present in the library and when the query is absent from the library, we achieve the best results



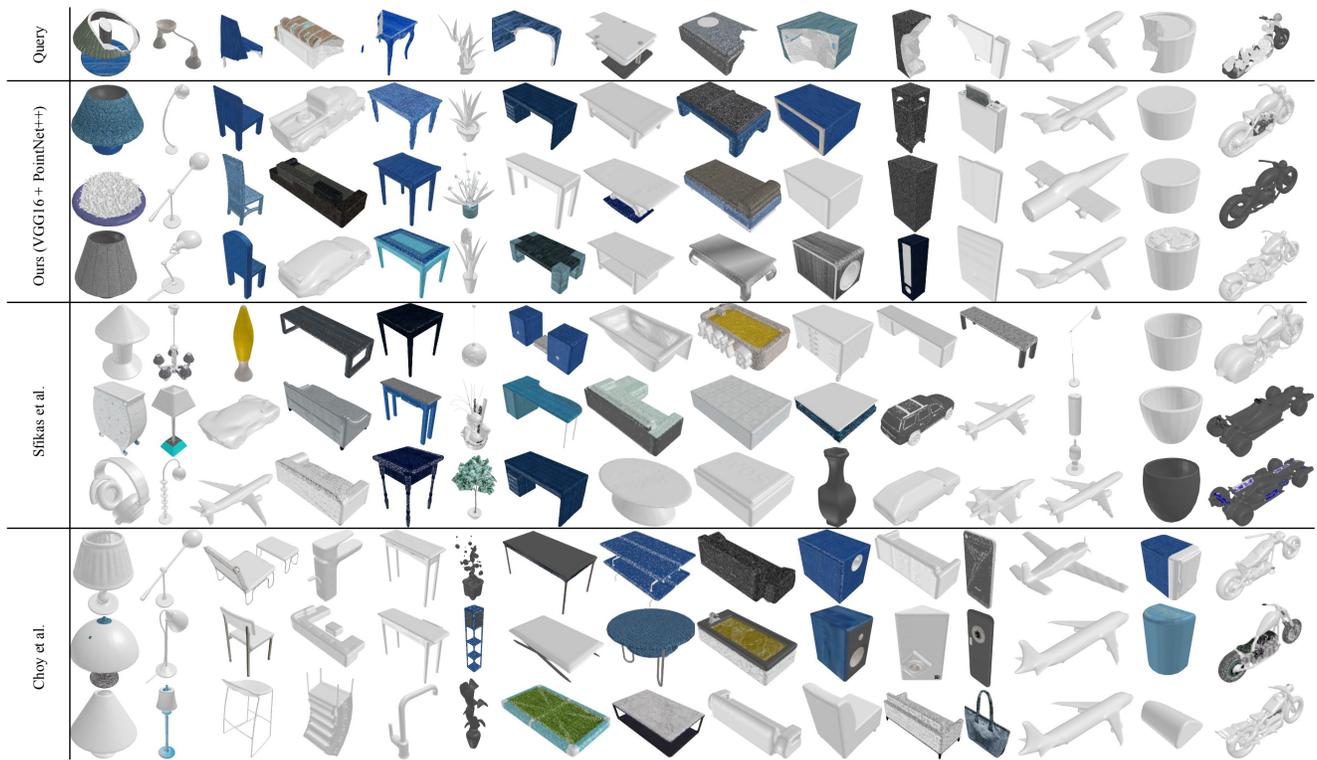


Figure 6: Retrieval where the complete version of the ShapeNet query object is absent from the library of ShapeNet objects.

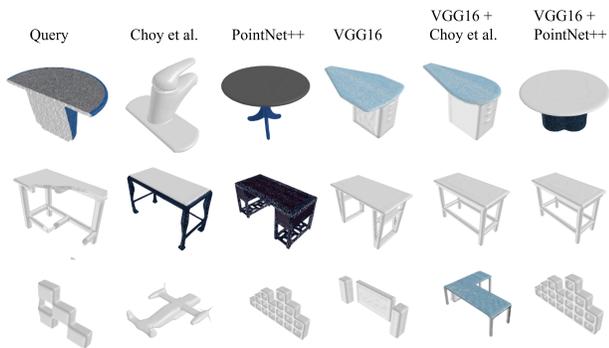


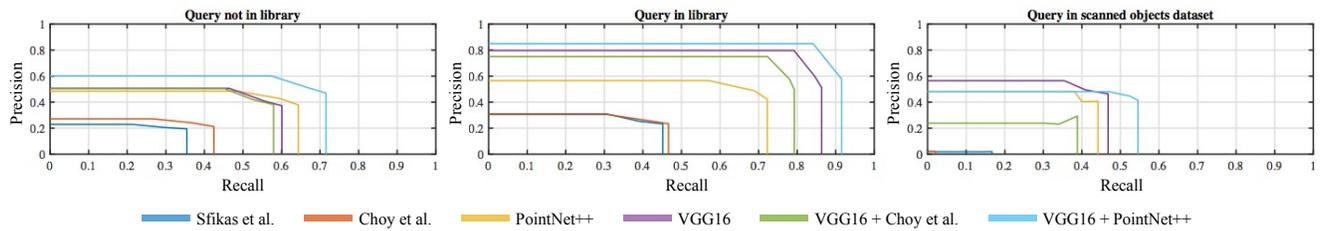
Figure 7: Retrieval results for each feature configuration using the first ShapeNet subset, where the query is not present in the library. ‘VGG16 + PointNet++’ returns objects that are a strong visual and geometric match to the query object.

details on a larger scale are not represented correctly in the retrieved object model, e.g., the base of the first query. Combining VGG16 and PointNet++ enables addressing matches at coarse and fine scales. When we analyze the method of Choy et al. [2019] with local features accumulated using BoF, we find that while it models smaller details well, its use of BoF eliminates spatial consistency, causing a reduction in match accuracy at a global scale. Combining VGG16 and Choy et al. appears to improve results at a global scale,

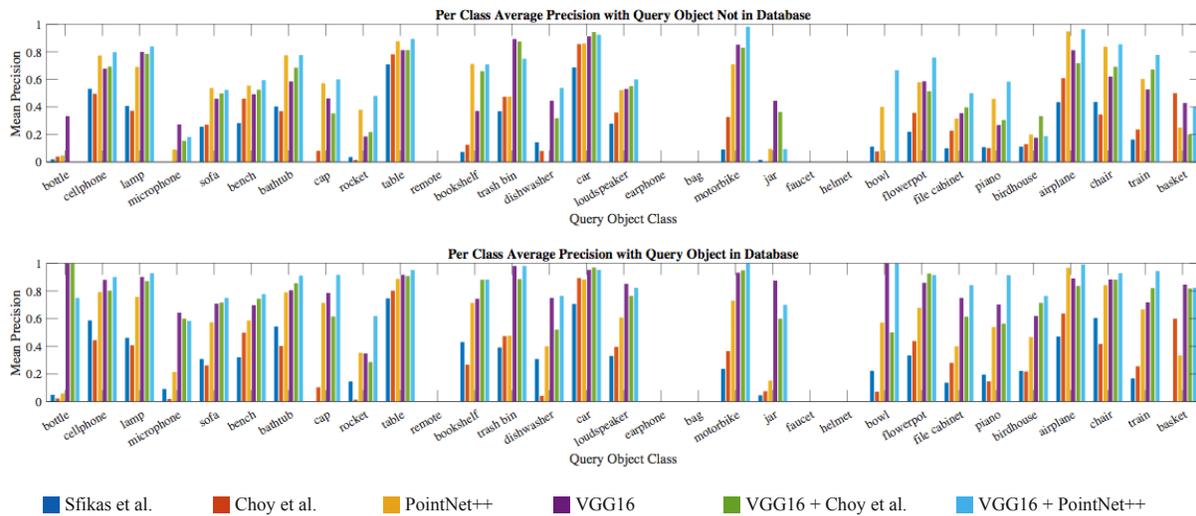
however, we find that visually, objects retrieved using ‘VGG16 + PointNet++’ demonstrate a closer match. Our approach outperforms the approach of Sfikas et al. [2016] with respect to all metrics, when the query is present in the library and when the query is absent from the library, due to our use of learned features and our incorporation of visual features unlike Sfikas et al. [2016] who only use geometric features. Figure 8 shows the precision-recall curve generated by returning the top 1 to top 20 objects from the library subset, and computing the mean precision and mean recall over all classes. As shown by the figure, using ‘VGG16 + PointNet++’ shows highest performance, demonstrating that the classes are most relevant when compared with other methods. As expected, the performance is higher when the query is present in the library. Even when performance is lower, as in the case of the query object not being present in the library, Figure 6 demonstrates that the results are geometrically similar, enabling them to be usable in repair. As shown in Figure 9, cars, airplanes, motorbikes, tables, and chairs show high precision, while helmets, faucets, bags, remotes, earphones, microphones, and bottles show low or 0 precision. The negligible precision may be attributed to their generic structure, which may induce high matches with geometrically similar objects from a different class.

## 5.2 Retrieval for Scanned Objects as Query using ShapeNet Objects in the Library

To analyze the effectiveness of our method in comparing data from real scans against synthetically created objects, we provide results



**Figure 8: Precision versus recall curves for each of the encoder combinations, compared to the approach of Sfikas et al. [2016] and features generated using the approach of Choy et al. [2019]. We compute the mean precision and recall over all classes for the top 1-20 retrieved objects. Left: query object not present in library. Center: query object present in library. Right: query object in scanned object library.**



**Figure 9: Average precision of each of the encoder combinations, compared to the approach of Sfikas et al. [2016] and features generated using the approach of Choy et al. [2019] across all classes in the ShapeNet [Chang et al. 2015] dataset.**

of the third experiment on complete object retrieval using the query subset from the Large Dataset of Object Scans [Choi et al. 2016] and the first ShapeNet library subset containing all 28,832 complete ShapeNet objects. We show quantitative results in the third super-column of Table 1, and qualitative results in Figure 10. As shown in Table 1, our approach using ‘VGG16 + PointNet++’ outperforms the remaining methods in terms of Chamfer distance and normal consistency, demonstrating that our approach provides geometrically similar object proxies. VGG16 demonstrates a higher mAP than our approach, indicating that the geometric features from the scanned objects may not be as successful for class retrieval. This may be attributed to slight alterations in the geometric structure during cleaning and waterproofing, when holes left by extraneous geometry removal are incorrectly filled. Similar to Subsection 5.1, our approach outperforms Sfikas et al. [2016]. Figure 8 demonstrates a similar trend as Table 1, i.e., that VGG16 shows a higher performance than ‘VGG16 + PointNet++’ in terms of class retrieval. The left of Figure 12 provides a breakdown of precision in terms of classes. We find that including geometric features does not perform

as well as using VGG16 alone for trash bins and sofas, likely due to the lack of fine geometric detail. Including geometric features improves performance for motorbikes and chairs, exceeding VGG16 for motorbikes. This may be attributed to the success of geometric features in modeling the fine detail of motorbikes.

### 5.3 Retrieval of Cultural Heritage Objects

We show quantitative results for the fourth experiment on proxy retrieval using objects from the Hampson Museum dataset [Payne et al. 2009] in Table 2. In this case, query objects are absent from the library. Table 2 provides Chamfer distance and normal consistency. We do not show mAP as the Hampson Museum dataset lacks class labels. As shown in Figure 11, our approach consistently returns relevant CH objects and performs comparably to the approach of Sfikas et al. [2016] and features generated by the approach of Choy et al. [2019]. In terms of geometric similarity between the query objects and proxy results, our method produces results that are competitive with those of Sfikas et al. [2016] and Choy et al. [2019], shown in Table 2.

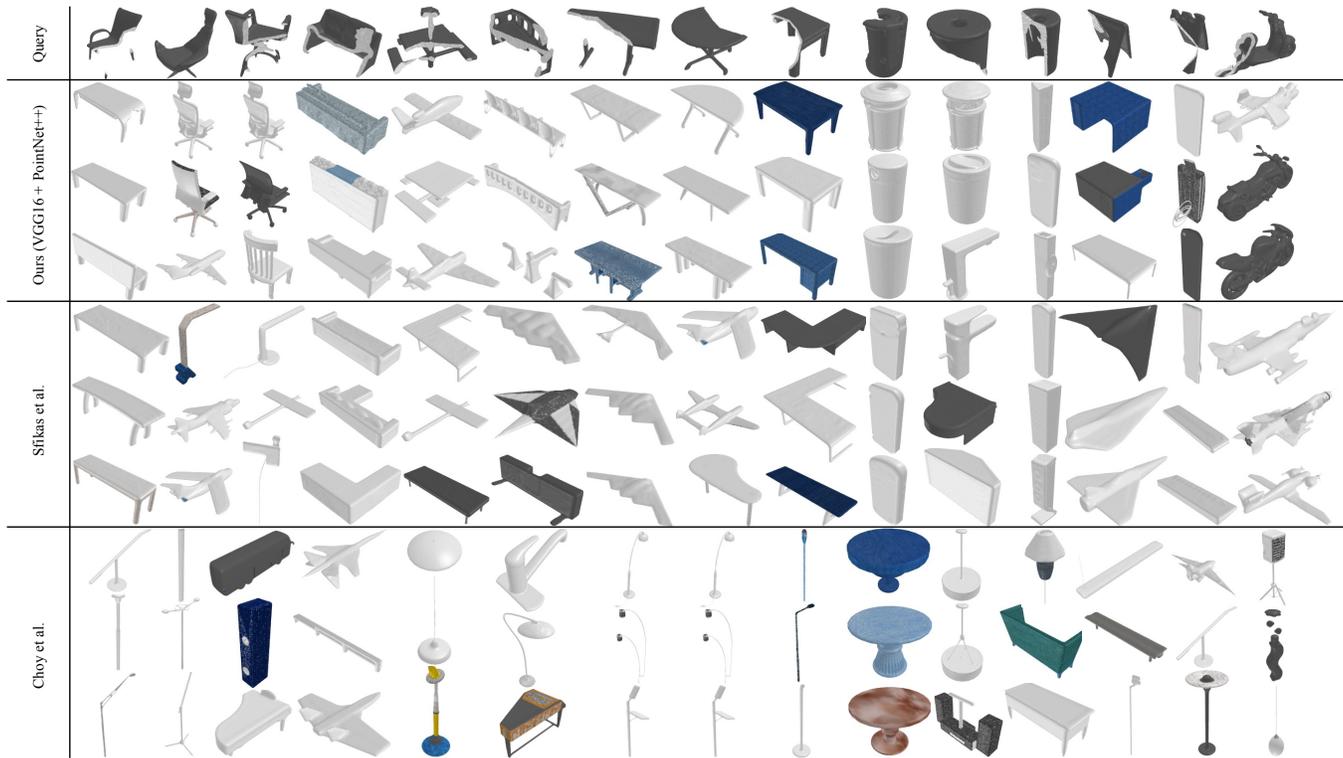


Figure 10: Retrieval for objects from the scanned object dataset [Choi et al. 2016] with ShapeNet object models as the library.

Table 2: Chamfer distance, normal consistency, and mAP of each encoder combinations, compared to the approach of Sfikas et al. [2016] and features generated using the approach of Choy et al. [2019] for the CH library. Low values are optimal for Chamfer distance and high values are optimal for normal consistency. Chamfer distance shown has been multiplied by 100.

	Sfikas et al.	Choy et al.	PointNet++	VGG16	VGG16 + Choy	VGG16 + PointNet++
Chamfer	6.970	4.280	5.018	<b>3.730</b>	4.049	3.996
Consistency	0.223	<b>0.389</b>	0.259	0.342	0.331	0.334

## 5.4 Object Repair

To demonstrate the usability of our approach for object repair, we use two synthetically fractured objects from ShapeNet as queries and retrieve the closest matching library ShapeNet object model when the queries are not present in the library, as shown on the right of Figure 12. We use the returned ShapeNet object model to perform repair by using the automated restoration approach of Lamb et al. [2019]. While the retrieved object model is not a perfect replica of the query object, as shown by the restorations generated in Figure 12, the retrieval is able to generate plausible restorations.

## 5.5 Statistical Testing

We perform a two-sided paired Student’s  $t$ -test to determine if the Chamfer distances, per class mAP, and normal consistency of our approach (‘VGG16 + PointNet++’) are significantly different than the approaches of Sfikas et al. [2016], Choy et al. [2019], ‘PointNet++’, ‘VGG16’, and ‘VGG16 + Choy et al.’. We use the Student’s  $t$ -test when the query is not in the library, when the query is in the

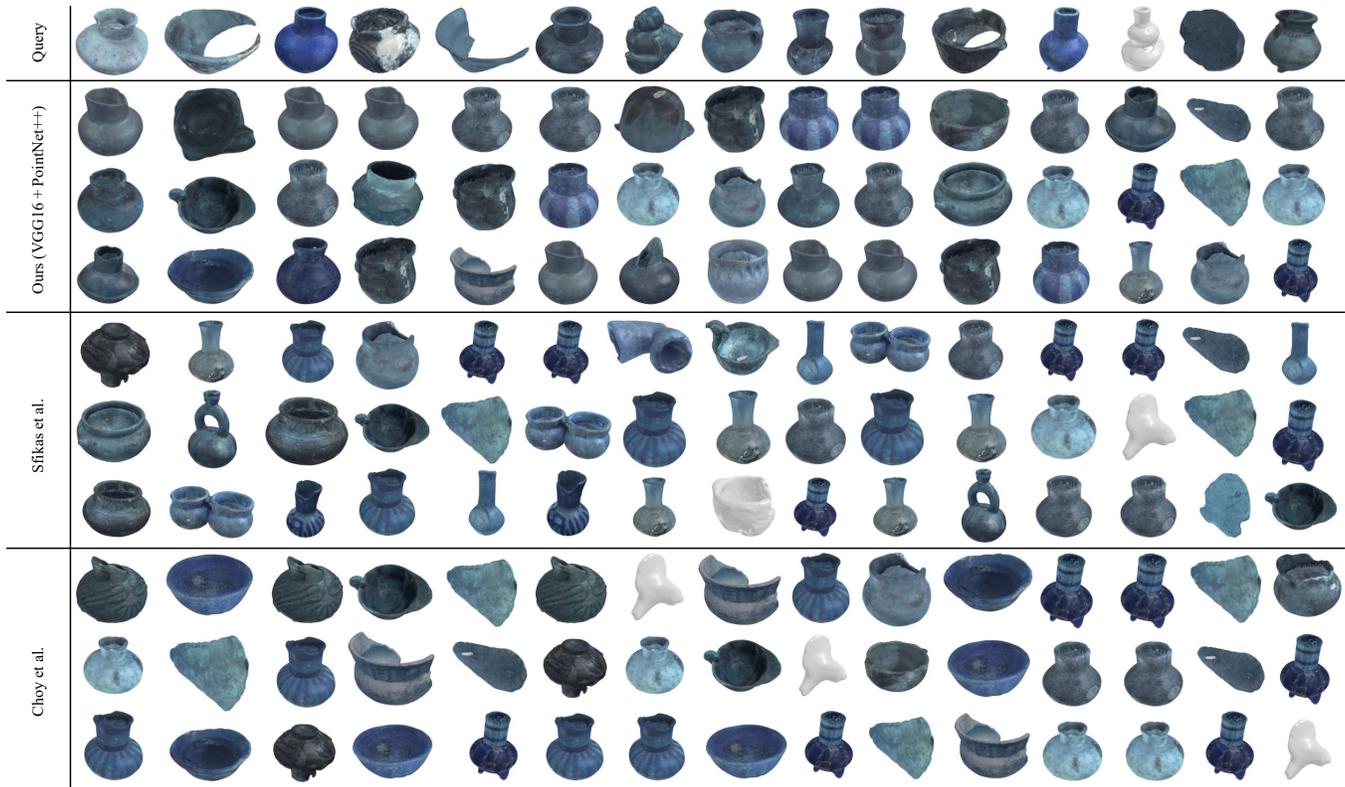
library, and when the query is a scanned object. Since we have less than 30 CH objects we use a two-sided paired Wilcoxon signed-rank test instead of the two-sided paired Student’s  $t$ -test. We exclude a statistical significance test for mAP for the scanned query object as we have less than 10 classes. We apply a Bonferroni correction and reject the null hypothesis if  $p < 0.01$ . We state the null and alternate hypotheses as follows:

**Null:** There is no difference in the [metric] value between our approach and [comparative] approach.

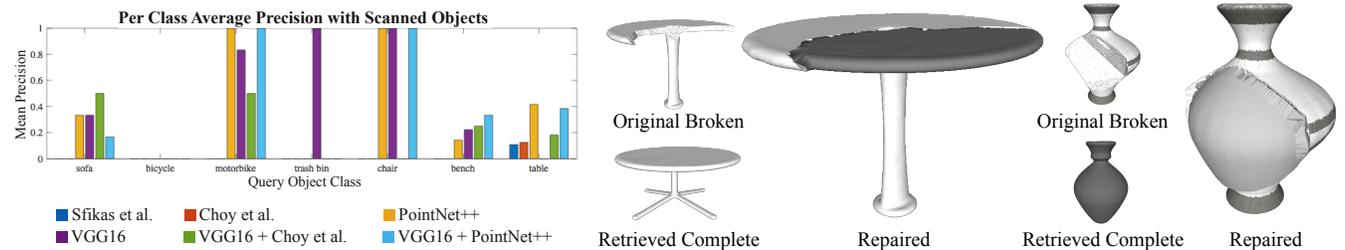
**Alternate:** The [metric] value obtained by our approach is not equal to [comparative] approach.

Above, [metric] refers to Chamfer distance, normal consistency, and mAP, and [comparative] refers to one of Sfikas et al. [2016], Choy et al. [2019], ‘PointNet++’, ‘VGG16’, or ‘VGG16 + Choy et al.’. For Chamfer distance our approach is superior if we obtain a lower value. For mAP and normal consistency our approach is superior if our value is greater.

We summarize the results of our statistical tests in Tables 3, 4, 5, and 6. As shown in Table 3, using ‘VGG16 + PointNet++’ when



**Figure 11: Retrieval results using query broken object models and the library scanned object models from the CH dataset. Query objects are not present in the library. While all object models demonstrate a degree of damage, the query models are subjected to further breaks using the object breaking method discussed in Section 4.**



**Figure 12: Left: Average precision of each encoder combination, compared to the approach of Sfikas et al. [2016] and features generated using the approach of Choy et al. [2019] across all classes in the scanned objects dataset. The precision for bicycles is zero as bicycles are not present in our ShapeNet subset. Right: Examples of broken objects, proxies retrieved using our method, and repair performed using Lamb et al. [2019], provided in the top-left, bottom-left, and right respectively for each object.**

the query is not in the library outperforms all other approaches with differences being statistically significant, except in the case of mAP for VGG16. However, our overall mAP score is higher as shown in Table 1. As shown in Table 4 using ‘VGG16 + PointNet++’ outperforms all other approaches when the query object is in the library, with differences being statistically significant. As shown in Table 5, using ‘VGG16 + PointNet++’ we obtain a lower Chamfer distance that is statistically significant when compared to all other approaches except for PointNet++. However, our Chamfer distance

score is lower when compared to PointNet++ as shown in Table 1. The consistency score with ‘VGG16 + PointNet++’ is statistically different from Sfikas et al. [2016]. The score is not statistically different from the other methods, likely since normal consistency is more successful at capturing high frequency differences, while the scanned models, being bulbous, demonstrate higher mid-level smoothness. As shown in Table 6 using ‘VGG16 + PointNet++’ we obtain a lower Chamfer distance that is statistically significant when compared to Sfikas et al. [2016]. This reflects the observation

**Table 3: When the query object is not in the library, our approach provides statistically significant lower Chamfer distance, higher normal consistency, and higher mean average precision when compared to other approaches. An italicized value indicates a difference that is not statistically significant, however, our average mAP score is higher.**

	VGG16	Choy et al.	Sfikas et al.	VGG16 + Choy et al.	PointNet++
Chamfer	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$
Consistency	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$
mAP	$p = 0.02615$	$p = 2.01E-09$	$p = 4.12E-11$	$p = 0.004626$	$p = 7.87E-05$

**Table 4: When the query object is in the library, our approach provides statistically significant lower Chamfer distance, higher normal consistency, and higher mean average precision when compared to other approaches.**

	VGG16	Choy et al.	Sfikas et al.	VGG16 + Choy et al.	PointNet++
Chamfer	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$	$p < 2.20E-16$
Consistency	$p = 3.46E-08$	$p < 2.20E-16$	$p < 2.20E-16$	$p = 1.91E-14$	$p < 2.20E-16$
mAP	$p = 0.008492$	$p = 1.96E-15$	$p < 2.20E-16$	$p = 0.000386$	$p = 8.09E-10$

**Table 5: For scanned query objects, we show a statistically significant lower Chamfer distance when compared to all approaches except for PointNet++. However, as shown in Table 1, our approach retrieves complete proxies with the lowest Chamfer distance of all approaches. For normal consistency we show a statistically significant higher score when compared to Sfikas et al. [2016]. However, as shown in Table 1, our approach retrieves complete proxies with the highest normal consistency score of all approaches. An italicized value indicates a difference that is not statistically significant.**

	VGG16	Choy et al.	Sfikas et al.	VGG16 + Choy et al.	PointNet++
Chamfer	$p = 0.008734$	$p = 1.92E-12$	$p < 2.20E-16$	$p = 7.72E-06$	$p = 0.2598$
Consistency	$p = 0.378$	$p = 0.01605$	$p = 1.74E-13$	$p = 0.5599$	$p = 0.9145$

**Table 6: For CH objects we show a statistically significant difference only in the Chamfer distance compared to Sfikas et al. [2016].**

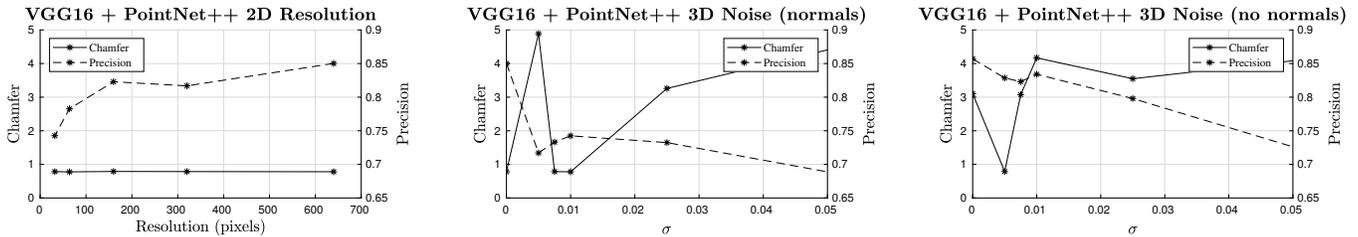
	VGG16	Choy et al.	Sfikas et al.	VGG16 + Choy et al.	PointNet++
Chamfer	$p = 0.7257$	$p = 0.4908$	$p = 0.000162$	$p = 0.9368$	$p = 0.07548$
Consistency	$p = 0.8127$	$p = 0.432$	$p = 0.03573$	$p = 0.8127$	$p = 0.2708$

in Figure 11 that the proxies returned by Sfikas et al. [2016] are less similar to the broken queries than the proxies returned by our method. The Chamfer distances and normal consistency scores are not statistically different for all remaining methods, indicating that they perform comparably. This may be attributed to the objects being largely uniform and non-diverse due to which visual and geometric features are likely to perform well.

## 5.6 Input Noise

We demonstrate the effect of using low quality 2D renders and varying degrees of point cloud noise with and without point normals as input to our approach in Figure 13. Experiments are performed using the first query and library subsets, i.e. where the query object models are present in the library. To study the effect of 2D render resolution on object retrieval, we re-render the query and library objects from 8 viewpoints over 5 resolutions: 640x640, 320x320, 160x160, 64x64, and 32x32 pixels, and encode these renders using

VGG16. As shown on the left of Figure 13, the precision improves exponentially with resolution, though the Chamfer distance remains relatively constant, showing that the quality of 2D renders impacts the relevance of retrieved proxies but has minimal effect on their geometric similarity. To study the effect of 3D point cloud noise on object retrieval, we apply zero-mean Gaussian noise to the point clouds with 6 standard deviations—0.0 or no noise, 0.005, 0.0075, 0.01, 0.025, and 0.05. We encode each noisy point cloud using PointNet++. When normals are encoded the Chamfer distance increases and the precision decreases rapidly as noise increases, as shown in the center of Figure 13. When normals are not encoded the Chamfer distance increases rapidly and the precision decreases slowly as noise increases, as shown on the right of Figure 13. When point cloud noise is 0, our approach shows lower Chamfer distance and higher average precision with normals than without normals. The decay of Chamfer distance and precision due to point cloud noise indicate that our approach returns geometrically suitable proxies if the models have normals and are scanned with high fidelity, but



**Figure 13: Chamfer distance and precision of ‘VGG16 + PointNet++’ on noisy inputs. Left: Chamfer distance and precision over varying 2D square resolutions. Center: Chamfer distance and precision over varying amount of Gaussian noise, with point normals. Right: Chamfer distance and precision over varying amount of Gaussian noise, without point normals. Gaussian noise applied to the point clouds is given by the standard deviation  $\sigma$ . Chamfer distance shown has been multiplied by 100.**



**Figure 14: Scale of query and returned proxy may differ.**

is negatively impacted by point cloud noise and impacted more negatively by point normal noise.

## 6 DISCUSSION

In this paper, we introduce a novel method to incorporate textural information and learned visual and geometric shape features using pre-trained neural networks in order to query a library with a broken object model and find a corresponding complete proxy model. We have demonstrated that our method provides reasonable complete proxy models for broken object queries even if the corresponding complete object is not present in the library. Our approach is able to robustly return a relevant set of complete object models in the case of inadequate visual or geometric data, such as missing or additional fracture geometry. Our method requires no training and consequently does not require any broken object training sets or time to train. The use of pre-trained networks in our work enables it to be adaptable to diverse datasets while still yielding reasonable complete proxy matches. Our approach forms an important component of the repair pipeline by automating the process of proxy retrieval. The object repair results we demonstrate in Figure 12 leverage complete proxies that are dissimilar from the original object with convincing results even without mesh deformation. The joint seamlessness is likely to degrade if the proxy model deviates significantly from the query object. However, the 3D models returned by our approach can be combined with deformation techniques [Botsch and Sorkine 2007; Sorkine et al. 2004] to align the object closely to the query for fully automated seamless repair. Complete automation of the repair pipeline enables restoration of discontinued, unique, handmade, personalized, and CH objects for which it is difficult or impossible to find a replacement.

## 6.1 Limitations

**6.1.1 Misclassification Due to Scale Invariance.** A key limitation of our approach is that it disregards object scale, as all models in the ShapeNet dataset are normalized with respect to a unit cube. As a result our system may return object models in the library that belong to the incorrect class even though they appear visually and geometrically similar to the query object model. For instance, our method returns a bathtub for the broken basket in Figure 14. While a characteristic of all retrieval methods that lack scale-awareness, the returned object may still be used to perform repair if the generated repair part does not cause the repaired object to deviate significantly from its original intended functionality. In the event that scale awareness is desired, our work can be extended to build in scale awareness by pairing it with a recognition step that automatically identifies the scale of the broken object model, and narrows the search to library objects that are within a certain scale range. While ShapeNet models are normalized, the models contain dimensional data that can be leveraged to perform scale-based searches.

**6.1.2 Lack of Consensus Between Visual and Geometric Features.** Another limitation is that if  $\mathcal{I} = \emptyset$ , the visual and geometric features do not reach a consensus on what library object best represents the query. While rare, this may occur if the dataset contains many objects that may be similar to the query so that the consistent closest matches in visual and geometric space are not found in the top returned features. The problem may be mitigated by increasing the number of returned features.

**6.1.3 Repair Infeasibility.** Our approach suffers from the limitations of database-driven repair algorithms, which can only repair broken objects that have suitable complete proxies in the database. Though our approach is able to generate high quality repairs such as those shown in Figure 12 for many common household objects that have generic geometry, such as vases, cups, and bowls, our approach is ill-suited to repair objects that are unique, such as statues or busts. Our approach may fail to return suitable complete proxies for object classes that require precise tolerances or geometric symmetry, such as gears and mechanical parts, unless an exact complete copy of the object exists in the database. However, our approach can be adapted to work with precise mechanical parts by deforming the retrieved proxy to the input broken model using existing deformation techniques [Botsch and Sorkine 2007; Sorkine et al. 2004] and enforcing constraints of symmetry and planarity.

## 6.2 Future Work

While our approach plays a key role in enabling repair automation, there remains a substantial scope for research in enabling full end-to-end automation of the repair pipeline. For objects where an entire part is missing, future work can investigate performing retrieval by leveraging parts annotations in datasets such as ShapeNet [Chang et al. 2015] and PartNet [Mo et al. 2019] to provide proxy restoration parts. Parts-based searches can also be used for locating parts on an object that require fixing, and performing retrieval for the parts rather than the entire object. This may improve fidelity of retrieval and repair by eliminating complex intact regions of the object that may hinder search, e.g., if the top of a swivel chair is broken, the complex geometry of the chair wheels need not be included in the search. At the output end, complete repair automation requires automatic identification of the material type needed to fix the object, in order to preserve internal and surface material properties and end-use functionality. Future work should investigate performing identification of the material composition by incorporating visual material sample databases into the retrieval process. Future work may also apply methods in proxy-based repair to modular 3D printing in order to repair large objects in remote locations where repair parts may not be immediately available.

While our approach leverages appearance and shape-based features to perform retrieval, our work is readily extensible to perform matching based on semantic information in the form of manually provided descriptions. Future work can investigate methods to facilitate user-directed repair by prompting the user for an input and tailoring the repair according to the user's preferences. At the input end, automation requires rapid strategies to scan 3D models of broken objects that overcome the limitations of traditional scanners, which includes users having to repeatedly re-position objects to model the base or hollow interiors of the object, a task that can prove arduous to the average user and hazardous for fragile or sharp objects. As part of future work, we are interested in exploring the use of depth cameras on robotic manipulators to perform full-range scanning of objects without requiring re-positioning of the object. We are also interested in investigating the use of multiple robotic manipulators to automatically perform repair-related tasks such as joint-bond application and join strength in order to enable a full end-to-end automation of the repair pipeline.

## ACKNOWLEDGMENTS

This work is supported by the NSF Graduate Research Fellowship under grant 1945954, and by NSF grant IIS-2023998.

## REFERENCES

- Masaki Aono and Wataru Iwabuchi. 2020. Part-In-Whole Type 3d Partial Shape Retrieval Based On Connected Faces With Pointnet Features. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, Piscataway, NJ, 1207–1215.
- Lucia Arbace, Elisabetta Sonnino, Marco Callieri, Matteo Dellepiane, Matteo Fabbri, Antonio Iaccarino Idelson, and Roberto Scopigno. 2013. Innovative uses of 3D digital technologies to assist the restoration of a fragmented terracotta statue. *Journal of Cultural Heritage* 14, 4 (2013), 332–345.
- Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. 2019a. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 2614–2623.
- Armen Avetisyan, Angela Dai, and Matthias Nießner. 2019b. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Piscataway, NJ, 2551–2560.
- Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. 2016. Gift: A real-time and scalable 3d shape search engine. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 5023–5032.
- Vincent Barra and Silvia Biasotti. 2013. 3D shape retrieval using kernels on extended Reeb graphs. *Pattern Recognition* 46, 11 (2013), 2985–2999.
- Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. 2008. Reeb graphs for shape analysis and applications. *Theoretical computer science* 392, 1–3 (2008), 5–22.
- Mario Botsch and Olga Sorkine. 2007. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics* 14, 1 (2007), 213–230.
- Alexander M Bronstein, Michael M Bronstein, Leonidas J Guibas, and Maks Ovsjanikov. 2011. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)* 30, 1 (2011), 1–20.
- Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* 1, 1 (2015), 1–11.
- Yang Chen, Guanlan Liu, Yaming Xu, Pai Pan, and Yin Xing. 2021. PointNet++ Network Architecture with Individual Point Level and Global Features on Centroid for ALS Point Cloud Classification. *Remote Sensing* 13, 3 (2021), 472.
- Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. 2016. A Large Dataset of Object Scans. *arXiv:1602.02481* 1, 1 (2016), 1–7.
- Christopher Choy and Junha Lee. 2019. Open Universal Correspondence Network. <https://github.com/chrischoy/open-ucn>.
- Christopher Choy, Jaesik Park, and Vladlen Koltun. 2019. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Piscataway, NJ, 8958–8966.
- Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. 2018. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 4578–4587.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 248–255.
- Carlos Esteves, Christine Allen-Blanchette, Amesh Makadia, and Kostas Daniilidis. 2018. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, Berlin, Germany, 52–68.
- Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
- Rao Fu, Jie Yang, Jiawei Sun, Fang-Lue Zhang, Yu-Kun Lai, and Lin Gao. 2020. RISA-Net: Rotation-Invariant Structure-Aware Network for Fine-Grained 3D Shape Retrieval. *arXiv preprint arXiv:2010.00973* 1, 1 (2020), 1–23.
- Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. 2004. Modeling by example. *ACM transactions on graphics (TOG)* 23, 3 (2004), 652–663.
- Takahiko Furuya and Ryutarou Ohbuchi. 2016. Deep Aggregation of Local 3D Geometric Features for 3D Model Retrieval. In *BMVC*, Vol. 7. The British Machine Vision Association and Society for Pattern Recognition, Durham, UK, 8.
- Robert Gregor, Ivan Sipiran, Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, and Pavlos Mavridis. 2014. Towards Automated 3D Reconstruction of Defective Cultural Heritage Objects. In *EUROGRAPHICS Workshops on Graphics and Cultural Heritage*. EUROGRAPHICS Association, Geneva, Switzerland, 135–144.
- Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 216–224.
- Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. 2019. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation. *IEEE Transactions on Image Processing* 28, 8 (2019), 3986–3999.
- Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. 2018. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 1945–1954.
- Wataru Iwabuchi and Masaki Aono. 2018. 3d cnn based partial 3d shape retrieval focusing on local features. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, Piscataway, NJ, 1523–1529.
- Jianwen Jiang, Di Bao, Ziqiang Chen, Xibin Zhao, and Yue Gao. 2019. MLVCNN: Multi-loop-view convolutional neural network for 3D shape retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, Palo Alto, California, 8513–8520.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 1, 1 (2019), 535–547.

- Nikolas Lamb, Sean Banerjee, and Natasha Kholgade Banerjee. 2019. Automated reconstruction of smoothly joining 3D printed restorations to fix broken objects. In *Proceedings of the ACM Symposium on Computational Fabrication*. ACM, New York, NY, 3.
- Guillaume Lavoué. 2012. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer* 28, 9 (2012), 931–942.
- Bo Li, Afzal Godil, and Henry Johan. 2014. Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3D model retrieval. *Multimedia tools and applications* 72, 2 (2014), 1531–1560.
- Roe Litman, Alex Bronstein, Michael Bronstein, and Umberto Castellani. 2014. Supervised learning of bag-of-features shape descriptors using sparse coding. In *Computer Graphics Forum*, Vol. 33. Wiley, Hoboken, NJ, 127–136.
- David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
- Pavlos Mavridis, Anthousis Andreadis, and Georgios Papaioannou. 2015. Fractured Object Reassembly via Robust Surface Registration. In *Eurographics (Short Papers)*. Eurographics, Geneva, Switzerland, 21–24.
- Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 4460–4470.
- Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 909–918.
- Weizhi Nie, Weijie Wang, Anan Liu, Jie Nie, and Yuting Su. 2019. HGAN: Holistic Generative Adversarial Networks for Two-dimensional Image-based Three-dimensional Object Retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 4 (2019), 1–24.
- Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, Pavlos Mavridis, Robert Gregor, Ivan Sipiran, and Konstantinos Vardis. 2017. From reassembly to object completion: A complete systems pipeline. *Journal on Computing and Cultural Heritage (JOCCH)* 10, 2 (2017), 1–22.
- Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 165–174.
- Angela Payne, Keenan Cole, Katie Simon, Christopher Goodmaster, and Fredrick Limp. 2009. Designing the next generation virtual museum: Making 3D artifacts available for viewing and download. In *Making History Interactive: Proceedings of the 37th Annual International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA)*, Vol. 3. CAA, 1–6.
- Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. 2016. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 5648–5656.
- Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* 1, 1 (2017), 1–14.
- Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. 2015. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 2484–2493.
- Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. 2019. RL-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, 5898–5907.
- Konstantinos Sfikas, Ioannis Pratikakis, Anestis Koutsoudis, Michalis Savelonas, and Theoharis Theoharis. 2016. Partial matching of 3D cultural heritage objects using panoramic views. *Multimedia Tools and Applications* 75, 7 (2016), 3693–3707.
- Konstantinos Sfikas, Ioannis Pratikakis, and Theoharis Theoharis. 2018. Ensemble of PANORAMA-based convolutional neural networks for 3D model classification and retrieval. *Computers & Graphics* 71 (2018), 208–218.
- Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. 2015. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters* 22, 12 (2015), 2339–2343.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* 1, 1 (2014), 1–14.
- Sekou Singare, Yaxiong Liu, Dichen Li, Bingheng Lu, and Sanhu He. 2008. Individually prefabricated prosthesis for maxilla reconstruction. *Journal of Prosthodontics* 17, 2 (2008), 135–140.
- Ivan Sipiran. 2018. Completion of cultural heritage objects with rotational symmetry. In *EUROGRAPHICS Workshop on 3D Object Retrieval*. EUROGRAPHICS Association, Geneva, Switzerland, 87–93.
- Hyeontae Son and Young Min Kim. 2020. SAUM: Symmetry-Aware Upsampling Module for Consistent Point Cloud Completion. In *Proceedings of the Asian Conference on Computer Vision*. ACM, New York, NY, 1–17.
- Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, New York, NY, 175–184.
- David Stutz and Andreas Geiger. 2020. Learning 3d shape completion under weak supervision. *International Journal of Computer Vision* 128, 5 (2020), 1162–1181.
- Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Piscataway, NJ, 945–953.
- Hedi Tabia and Hamid Laga. 2015. Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification. *IEEE transactions on multimedia* 17, 9 (2015), 1591–1603.
- Roberto Toldo, Umberto Castellani, and Andrea Fusiello. 2010. The bag of words approach for retrieval and categorization of 3D objects. *The Visual Computer* 26, 10 (2010), 1257–1268.
- Mikaela Angelina Uy, Jingwei Huang, Minhyuk Sung, Tolga Birdal, and Leonidas Guibas. 2020. Deformation-aware 3d model embedding and retrieval. In *European Conference on Computer Vision*. Springer, Berlin, Germany, 397–413.
- Yan Wang, Jie Feng, Zhixiang Wu, Jun Wang, and Shih-Fu Chang. 2014. From low-cost depth sensors to cad: Cross-domain 3d shape retrieval via regression tree fields. In *European Conference on Computer Vision*. Springer, Berlin, Germany, 489–504.
- Zihao Wang and Hongwei Lin. 2020. 3D shape retrieval based on Laplace operator and joint Bayesian model. *Visual Informatics* 4, 3 (2020), 69–76.
- Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 1912–1920.
- Xu Yan. 2019. Pointnet/Pointnet++ Pytorch.
- Mohsen Yavartanoo, Eu Young Kim, and Kyoung Mu Lee. 2018. Spnet: Deep 3d object classification and retrieval using stereographic projection. In *Asian Conference on Computer Vision*. Springer, Berlin, Germany, 691–706.
- Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 2017. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, 1802–1811.
- Yuhe Zhang, Kang Li, Xiaoxue Chen, Shunli Zhang, and Guohua Geng. 2018. A multi feature fusion method for reassembly of 3D cultural heritage artifacts. *Journal of Cultural Heritage* 33 (2018), 191–200.
- Jing Zhu, Fan Zhu, Edward K Wong, and Yi Fang. 2015. Learning pairwise neural network encoder for depth image-based 3d model retrieval. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, New York, NY, 1227–1230.