Joint Embeddings of Shapes and Images via CNN Image Purification

Yangyan Li[1,*] Hao Su[1,*] Charles Ruizhongtai Qi[1] Noa Fish[2] Daniel Cohen-Or[2] Leonidas J. Guibas[1]
[1]Stanford University [2]Tel Aviv University

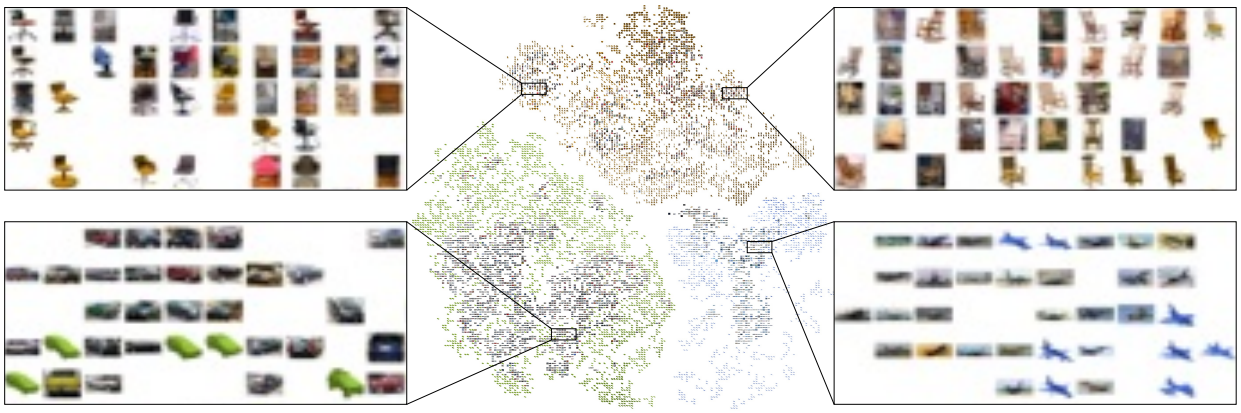

Figure 1: *We jointly embed shapes and images of three categories (chair, aeroplane and car) into a shared space. Distances between entities in the high-dimensional embedding space reflect object similarities between shapes and images (visualized by t-SNE here).*

Abstract

Both 3D models and 2D images contain a wealth of information about everyday objects in our environment. However, it is difficult to semantically link together these two media forms, even when they feature identical or very similar objects. We propose a *joint* embedding space populated by both 3D shapes and 2D images of objects, where the distances between embedded entities reflect similarity between the underlying objects. This joint embedding space facilitates comparison between entities of either form, and allows for cross-modality retrieval. We construct the embedding space using 3D shape similarity measure, as 3D shapes are more pure and complete than their appearance in images, leading to more robust distance metrics. We then employ a Convolutional Neural Network (CNN) to "purify" images by muting distracting factors. The CNN is trained to map an image to a point in the embedding space, so that it is close to a point attributed to a 3D model of a similar object to the one depicted in the image. This purifying capability of the CNN is accomplished with the help of a large amount of training data consisting of images synthesized from 3D shapes. Our joint embedding allows cross-view image retrieval, image-based shape retrieval, as well as shape-based image retrieval. We evaluate our method on these retrieval tasks and show that it consistently outperforms state-of-the-art methods, and demonstrate the usability of a joint embedding in a number of additional applications.

CR Categories: I.3.5 [Graphics]: Computational Geometry & Object Modeling—Object Representations I.4.7 [Image Processing & Vision]: Feature Measurement—Feature Representation;

Keywords: 3D Shapes, Embedding, Deep Learning

Figure 2: Image appearance vs. depicted object. *Images I_1 and I_2 are similar in general appearance while featuring different types of chairs. In contrast, images I_2 and I_3 have a different overall appearance, but both depict instances of the same type of chair.*

1 Introduction

Similar objects often appear in dissimilar images, but the necessity to recognize the object-based latent connection between these images exists in many Computer Vision and Computer Graphics applications. There are many factors that can hinder an attempt at object-based similarity estimation for images. Among these, dependencies on viewpoint, lighting, background differences, as well as partial occlusions, are most prevalent. In Figure 2, image I_1 features one object, while I_2 and I_3 feature a different one. As can be seen, although I_1 and I_2 do not portray the same object, they appear quite similar in their overall image ambiance. At the same time, I_2 and I_3 feature highly similar objects, but they greatly differ in overall image appearance. In contrast, 3D object representations are free from such impediments because they encode the entire object in a more pure form. Thus, object similarity measures for 3D shapes are inherently more robust than those for generic images.

To alleviate the problem, we present a method that embeds images and 3D shapes into a common space. In that space, both 3D model and image object similarities can be measured as if their 3D

*Joint first authors.

form was directly available. We utilize a set of 3D shapes to learn a collection-based similarity measure between objects in a given class. For each shape, this measure essentially defines a point in an embedding space with a metric that captures object similarity, as the coordinates of each point consist of a dimension-reduced form of the distances between the underlying shape and the entire set. Thus, two neighboring points in this embedding space are likely to represent similar shapes, as they agree, to a certain extent, on their similarities with all the other shapes.

We then consider the problem of mapping images to this shape embedding space. To overcome the limitations posed by the use of images outlined above, we leverage recent progress in the field of Deep Learning. A CNN can learn to map an image into the shape embedding space so that it lies near those of other images containing similar objects, as well as those of actual 3D models that are akin to the depicted object (see Figure 1). The process of *deep embedding* naturally enables cross-view image retrieval, image-based shape retrieval and shape-based image retrieval, in the unified embedding space. The unified space also supports various other applications that require 3D representations of objects observed in images [Xu et al. 2011; Zheng et al. 2012; Su et al. 2014; Kholgade et al. 2014; Lee et al. 2015; Huang et al. 2015].

An appealing quality of the trained CNN is its robustness to various kinds of nuisance factors or distractions that are common in real-world images [LeCun et al. 2015]. In our setting, this characteristic is advantageous as we aim to map images into the embedding space obliviously to these inconsistencies. However, to train such a CNN, a large amount of images along with their ground truth coordinates in the embedding space is required. These coordinates are necessarily high dimensional real values, therefore humans will find this annotation task to be quite difficult. Another alternative for obtaining the necessary links between images and their embedding, is to manually link images to similar 3D models. However, this task is highly time-consuming and error-prone. To overcome this difficulty we synthesize the image training set based on rendering a rather modest set of annotated shapes from ShapeNet [Su et al. 2015b]. We show that a large amount of effective and diverse annotated training images can be generated from a controlled synthesis procedure [Su et al. 2015a], requiring a minimal amount of manual labor, already performed while setting up ShapeNet.

3D shapes and 2D images are both important visual forms representing, among others, the objects around us. To date, however, these two forms have not been effectively linked together, due mostly to the great variation and inconsistency that is characteristic of real-world images. Our deep embedding is capable of "purifying" these images by peeling off their distracting layers. It then maps them into the 3D shape embedding space, where the two domains are inter-linked by their shared object content. Such a linking is key in making 3D shapes and 2D images comparable to each other and thereby also cross-retrievable, regardless of differences in overall image appearance. We show that our deep embedding universally supports cross-view image retrieval (section 6.1), image-based shape retrieval (section 6.2) and shape-based image retrieval (section 6.3). We evaluate the performance of our *deep retrieval*, and show that it out-performs state-of-the-art methods, most notably when dealing with real-world images with cluttered backgrounds. We also show that our method can be an important building block in several Computer Graphics applications, such as 3D-aware image manipulation and image-based 3D modeling.

2 Related Work

This work revolves around bridging the gap between 3D shapes and 2D images, by linking them together in a common space. As such, it touches on central aspects in both domains, including shape similarity, image similarity, and retrieval of both shapes and images. The proposed method makes use of the CNN known as AlexNet [Krizhevsky et al. 2012] for the task of image embedding and we report our performance based on it. More recent CNNs, such as GoogLeNet [Szegedy et al. 2014], can also be used, with potential performance improvements. Chatfield et al. [2014] provide a profound review of deep learning and CNNs. We therefore focus this Section on discussing the most relevant approaches for shape signatures, image retrieval, shape retrieval, and multi-modal embeddings — the key tools of our construction.

Shape Signatures A shape signature is a concise description of the shape, aimed at facilitating central tasks such as shape matching, organization, and retrieval. Designing a robust signature that can discriminate certain important shape characteristics while being oblivious to others has been extensively studied in Computer Graphics and Computer Vision. One type of shape signature focuses on geometric properties of the shape such as volume, distance and curvature [Osada et al. 2002; Ohbuchi et al. 2005; Gal et al. 2007], spherical harmonics [Kazhdan et al. 2003] and 2D projections [Chen et al. 2003]. Another type of method uses graph representations in order to obtain a topological description of the shape [Hilaga et al. 2001; Chen and Ouhyoung 2002; Sundar et al. 2003]. We design our shape distance metric based on shape signatures, specifically using the *LightField Descriptor* principles [Chen et al. 2003]. In this approach, shapes are indexed via a set of 2D projections or views — a representation that is directly related to the manner of appearance of an object in an image. A view-based distance metric can be derived from these descriptors, which can be leveraged in the construction of an embedding space that captures object similarities. Our signature for each shape essentially becomes its embedding point. As we will show later, there is no need to generalize the shape signature computation approach for real-world images, as there is a more effective way of inferring a compatible signature for an image, and its embedding into the robust space designed primarily for shapes.

Image Retrieval The task of content-based image retrieval (CBIR) involves searching for an image that is similar to the query, either in low-level aspects such as color and texture, or in high-level semantics such as objects that appear in the image. The difference between these two levels is referred to as the *semantic gap* [Smeulders et al. 2000], and is the main focus of CBIR methods, as surveyed in [Liu et al. 2007].

A recent work by [Bell and Bala 2015] bears important similarity to our method. In their work, visual search in interior design is addressed by learning a joint embedding for two types of images depicting home decor objects — iconic and in-situ. Similarly to our method, this approach also makes use of a CNN to learn the joint embedding. The input to the system is a collection of similar as well as dissimilar pairs of images collected through crowd sourcing. A Siamese network [Hadsell et al. 2006] is then adopted to learn the joint embedding. Our method differs in several respects, the most central of which is that our scheme is aimed at embedding both images and shapes into the same space where they can be easily compared. Secondly, the Siamese network approach solves for both the discovery of the embedding space, as well as the mapping of images into that space. We construct the joint embedding space by leveraging the above robust distance metric for 3D shapes, and then train a CNN for purifying and mapping real-world images into this pre-built embedding space. This separation between the construction of the embedding space and the image embedding process makes the overall task more tractable. We show that our embedding space induced by 3D shape similarities provides high

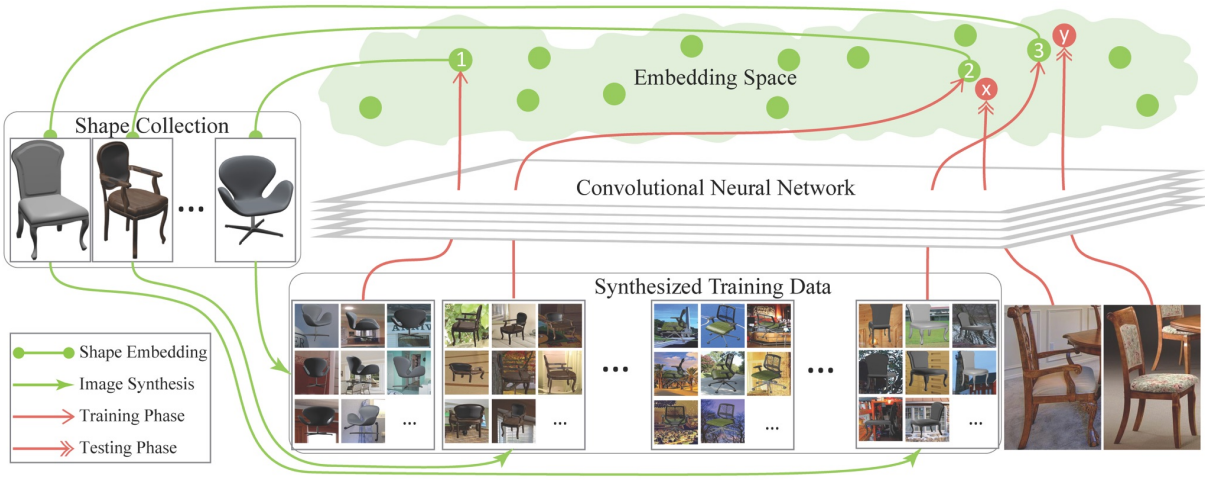

Figure 3: Our method is composed of four major components: embedding space construction, training image synthesis, training phase, and testing phase. In embedding space construction, 3D shapes are embedded into a space where the distance between the shapes reflects the similarity between them (green links). In training image synthesis, a large amount of training data is synthesized from the shape collection (green arrows). In the training phase, the CNN is trained to map the synthesized images to the embedding points of their source shapes (red arrows). In the testing phase, the network is shown to be capable of mapping real-world images into the same embedding space (red double-ended arrows).

quality guidance for training the purifying CNN, out-performing the Siamese-network-based approach. Finally, we collect training data by synthesizing a large amount of annotated images using a collection of 3D shapes, thereby eliminating the need for manual work while gaining annotations of a higher quality and finer detail.

Shape Retrieval Many notable content-based shape retrieval approaches have been proposed in recent years. These methods mostly operate on an input query in the 3D domain [Tangelder and Veltkamp 2008]. Other methods support view-based retrieval, where a query containing a sketch or an image is used to retrieve 3D models that have similar views to the query [Loffler 2000; Cyr and Kimia 2001; Funkhouser et al. 2003; Chen et al. 2003]. However, requiring projections or sketches with a clean background, these methods do not extend well to real-world image queries. With the ever growing real-world images readily available to everyone, it is quite clear that facilitating a real-world-image-based shape retrieval will help render this task more accessible. Our approach achieves this goal by embedding both images and shapes into the same space, thereby enabling a comparison between them.

Recently, Aubry et al. [2014] proposed an exemplar part-based method focusing on detecting image regions that match parts in 3D models of chairs in a large shape database. A star model is designed to combine discriminative patches for measuring the similarity. In our case, the similarity measurements are performed through a robust 3D shape distance metric, while image feature extraction is left as a task for the CNN. Our system features an end-to-end solution for similarity learning, thereby minimizing the effort involved in tuning the right parameters for similarity measurement.

Multi-modal Embeddings Multi-modal embeddings have been used in Computer Vision [Weston et al. 2010; Weston et al. 2011] to establish image-word relationships, but they have not been a common practice in the Computer Graphics domain. A recent graphics contribution is [Herzog et al. 2015] which starts by a creating a common embedding space for 3D models and keywords, and adds images and sketches to the mix. In that work the embedding is obtained by using feature vectors from all modalities simultaneously, mixing both informative and noisy data. Instead, our embedding space is computed from clean 3D models alone and therefore better reflects ideal object similarities. Furthermore, our 2D shape view HoG-based similarity metric is better suited to matching shapes to images, as compared to their local histograms of principal curvature directions. Most importantly, the key to robust comparison of real-world images to 3D models is image invariance learning — a difficult task when using linear classifiers as in LeSSS, but one that is handled exceptionally well by the CNN component of our work for removing image nuisance factors. The superior image to shape matching results achieved by this approach justify the choices we have made.

3 Overview

The input to our method is a set of 3D shapes from one or multiple classes (Figure 1), and a set of images, each featuring a prominently-displayed object from a known class. The images can be of varying characteristics, and there are no restrictions on viewpoint, lighting or background properties.

The approach consists of four major components: embedding space construction, training image synthesis, CNN training phase, and the final testing phase (see Figure 3 for a full illustration). In the embedding space construction, a collection of 3D shapes is embedded into a common space (green links) that serves as a platform for comparison between shapes and images. In the training image synthesis step, 3D shapes are used in a rendering process to obtain a large amount of annotated training data (green arrows). In the network training phase, a CNN is trained to learn the mapping between images and the 3D shape induced embedding space (red arrows). Finally, in the testing phase, the trained network is applied on new images to obtain an embedding into the space, where comparisons can be carried out (red double-ended arrows). The first component describing the construction of the embedding space is discussed in Section 4. The final three components are discussed in Section 5.

New images can be embedded into the space at any time, simply by feeding them as input to the trained purification CNN, and retrieving the output (Section 5.3). Introducing a new shape, how-

ever, is not as straight-forward, since the embedding space is constructed based on information obtained from the initial collection of 3D shapes. However, incremental addition of new shapes can be supported by solving an optimization problem designed to preserve the pairwise distances between the added shape and the existing shapes within the embedding space (Section 4.3).

4 Embedding Space Construction

It is a challenging task to design an embedding space where both real-world images and shapes co-exist. The difficulty lies in the requirement that the space captures similarities between heterogeneous media forms based on the underlying object they represent. To alleviate this, we focus our efforts on obtaining a robust embedding space solely based on the set of 3D shapes, for two reasons. First, unlike images, 3D models are generally less afflicted by distracting or nuisance factors, rendering pairwise comparison between them more reliable. In addition, 3D models are a more pure and complete representation of objects, and as such are easier to map to each other globally and locally. A pairwise comparison between them is therefore also more informative and precise. We then rely on the competence of a CNN for facilitating the embedding of real-world images into the space that we obtain (see Section 5). We shall first describe our shape similarity estimation, followed by the construction of the embedding space and the manner in which it utilizes shape similarities.

4.1 Shape Similarity

Our shape distance metric is based on the principles of the Light-Field Descriptor [Chen et al. 2003], where the similarity between two shapes is measured by the aggregate of similarities among corresponding views. Denote by $\mathcal{S} = \{S_i\}_{i=1}^{n}$ our input set of 3D shapes. The instances in the set are jointly aligned by a global rigid transformation [Huang et al. 2013], and then projected from k viewpoints to generate projection images: $\mathcal{I}_i = \{I_{i,v}\}_{v=1}^{k}$ for each S_i. We set $k = 20$ in all of our experiments and viewpoints are evenly sampled around the common up-orientation on the viewsphere. For each $I_{i,v}$ we compute a feature vector $H_{i,v}$ based on Histogram of Gradients (HoG) [Dalal and Triggs 2005]. $H_{i,v}$ is composed of a 3-level pyramid of HoG computed from images scaled to resolution 120×120, 60×60 and 30×30, and has $10,188$ dimensions. A comparison between two HoG descriptors is an effective and discriminative approach to estimate the similarity between two projections in our setting, since these projections are all well aligned. We therefore opted to use HoG rather than the combination of a region shape descriptor (Zernike moments) and a contour shape descriptor (Fourier) as suggested by Chen et al. [2003], which is more robust than HoG when applied on 3D models in-the-wild, when no global alignment is provided.

The feature vector F_i of shape S_i is obtained by concatenating its viewpoint feature vectors ,i.e., $F_i = (H_{i,1}; H_{i,2}; ...; H_{i,k}) \in \mathbb{R}^{203,760}$. The distance between S_i and S_j is then the L_2 distance between their feature vectors: $d_{i,j} = \|F_i - F_j\|_2$.

4.2 Embedding Space

The construction of the embedding space requires attention to two aspects. First, recalling that we aim to use the constructed space as a platform for estimating similarity between multi-modal entities, the most basic requirement is that the distances between the embedded shapes within the space reflect the similarity between them. Additionally, it is advantageous to restrict the dimension of the space for computational reasons. The CNN parameter space can then be bounded to avoid overfitting, and fast distance computations can be performed within the space. A lower dimension also guides the embedding to better respect shared structure among the shapes. The space spanned by F_i, $\mathbb{F} = \text{span}(\{F_i\}) \in \mathbb{R}^{203,760}$, satisfies the first criterion, but not the second one. Principal Component Analysis (PCA) can be applied to obtain a compact version of $\mathbb{F}$, denoted by $\mathbb{F}^-$, of a significantly lower dimension while still preserving pairwise distances well.

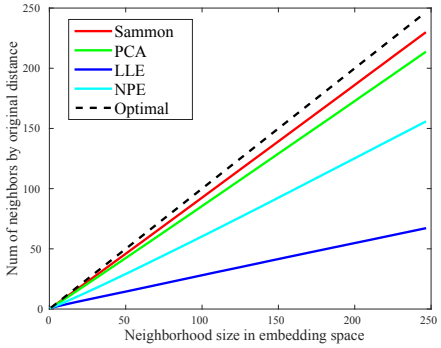

Figure 4: Recall curve of $\mathbb{F}^-$ (from PCA, LLE, and NPE) and $\mathbb{D}^-$ (from Sammon mapping). *Note that $\mathbb{D}^-$ preserves local neighborhoods better than $\mathbb{F}^-$.*

A central observation on similarity between objects is that distances between dissimilar instances are not as meaningful and reliable as distances between similar ones. In our case, as can be expected, a set of shapes contains many more dissimilar than similar pairs of shapes. However, the most straight forward embedding space construction $\mathbb{F}^-$ does not respect this distinction between the greater importance of the small distances and the lesser one of the larger ones, as it is obtained by PCA that optimizes only for a minimization of the reconstruction error.

With that in mind, we propose an alternative option for space construction as follows. Consider $\mathcal{D}_{n\times n}$, a pairwise distance matrix such that $\mathcal{D}(i,j) = d_{i,j}$. The space spanned by the rows of $\mathcal{D}$, $\mathbb{D} \in \mathbb{R}^{\|\mathcal{S}\|}$, also satisfies the first criterion of similarity preservation, as a similarity between the distances of S_i to all other shapes in $\mathcal{S}$, and the distances of S_j to all other shapes in $\mathcal{S}$, implies a direct similarity between S_i and S_j. Despite the lower dimension of $\mathbb{D}$, it is still too high to be practical when dealing with large sets of shapes. Differently than $\mathbb{F}$, $\mathbb{D}$ is a distance matrix, for which we can apply dimensionality reduction methods that respect the distinction between the greater importance of the small distances and the lesser one of the larger ones. We opted to obtain $\mathbb{D}^-$ from $\mathbb{D}$ by performing non-linear Multi-Dimensional Scaling (MDS) [Kruskal 1964] with Sammon mapping [Sammon 1969], as it encourages the preservation of the structure of local neighborhoods while embedding the original space into a Euclidean space of a lower dimension. More specifically, we estimate the coordinate of each shape in $\mathbb{D}^-$ by minimizing the following Sammon error during MDS:

$$E = \frac{1}{\sum_{i<j} \mathcal{D}(i,j)} \sum_{i<j} \frac{(\mathcal{D}(i,j) - \mathcal{D}^-(i,j))^2}{\mathcal{D}(i,j)}, \qquad (1)$$

where $\mathcal{D}^-(i,j)$ denotes the Euclidean distance between S_i and S_j in $\mathbb{D}^-$. The Sammon type error is a weighted sum of differences between the original pairwise distances and the embedding pairwise distances. Intuitively, dissimilar shape pairs are weighted down.

We use the same dimensionality m for both $\mathbb{F}^-$ and $\mathbb{D}^-$, such that $\mathbb{F}^-, \mathbb{D}^- \in \mathbb{R}^m$. In all our experiments, we set $m = 128$. This value is chosen heuristically to cater to both criteria — faithful dis-

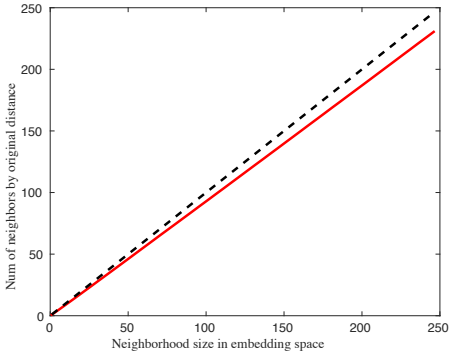

Figure 5: Adaptation to shape database growth. *The set of chairs (6778 instances in total) is divided into two parts — a training set containing 5000 shapes and a validation set containing the remaining 1677 shapes. An embedding space is constructed solely based on the training set shapes. The validation set shapes are then added post-construction (Section 4.3). The red solid line denotes the recall and the diagonal line represents the optimum.*

tance preservation and space-size compactness. In particular, for $\mathbb{F}^-$ obtained through PCA-based dimensionality reduction, we experimentally observe that 98% of shape descriptor energy is preserved for the "chair" category. Note that such high energy preservation is made possible by the clean and nuisance-free traits of the 3D-shape-based rendered images. We applied a similar dimensionality reduction procedure on n "chair" images from ImageNet, and found that only 76% energy is preserved following a reduction to the same dimension. Experimental results also show that $\mathbb{D}^-$ is superior to $\mathbb{F}^-$, allowing for a better performance by discriminating smaller distances attributed to similar pairs of objects. Henceforth, we proceed with the $\mathbb{D}^-$ as a reference to our embedding space.

We evaluated the quality of different embedding space constructions by the recall rate obtained for each construction (see Figure 4). Aside from the aforementioned PCA and MDS with Sammon mapping methods, we also evaluated the embedding space obtained by applying Local Linear Embedding (LLE) [Roweis and Saul 2000] and Neighborhood Preserving Embedding (NPE) [He et al. 2005] on $\mathbb{F}$. It is clear from the recall curve that MDS with Sammon mapping is a better fit for our task than the alternatives.

4.3 Mapping New Shapes

The embedding space is constructed based on a set of 3D shapes. The instances within this set are therefore jointly embedded into the space as part of the process. Introducing a new shape S_* into the system therefore requires special attention, and calls for the retrieval of an embedding point $P_{S_*} \in \mathbb{D}^-$. This retrieval process can be derived from the manner in which the embedding space was originally constructed. First, a LightField HoG feature vector F_* is computed. Next, pairwise distances between S_* and all $S_i \in \mathcal{S}$ are computed and set to be $d_{S_*,S_i} = \|F_* - F_i\|_2$. We denote the distance between S_* and any S_i in the embedding space as $d^-_{S_*,S_i} = \|P_{S_*} - P_{S_i}\|_2$. P_{S_*} can be solved by L-BFGS [Liu and Nocedal 1989] while minimizing the Sammon type error in a similar manner to that which is featured in Equation 1:

$$P_{S_*} = \arg\min_{P_{S_*}} \sum_i^n \frac{(d_{S_*,S_i} - d^-_{S_*,S_i})^2}{d_{S_*,S_i}}. \quad (2)$$

Figure 5 presents an experiment to evaluate the quality of our approach for introducing new shapes. After constructing an embedding space $\mathbb{D}^-$ based on a shape collection $\mathcal{S}_1$, a disjoint set $\mathcal{S}_2$ is embedded into $\mathbb{D}^-$. We compute the k-nearest neighbors (for varying k) of each shape in $\mathcal{S}_2$, from within the set $\mathcal{S}_1$, in two different ways. First, under distances between the original HoG feature vectors, and second, under distances within the embedding space $\mathbb{D}^-$. The two sets of nearest neighbors are compared to reflect the extent of preservation of the original distances.

5 CNN for Image Embedding

The embedding space $\mathbb{D}^-$ associates each 3D shape S_i with a point $P_{S_i} \in \mathbb{D}^-$. Aiming for a joint space shared not only by shapes but also by images, we are looking to embed the latter group into $\mathbb{D}^-$.

Due to its convolutional structure, a CNN is able to separate an image into various layers of abstraction, capturing different features and elements. It is this characteristic of the network that allows it to be utilized for many different learning tasks, each requiring a different focus. Leveraging this adaptive ability, we train a CNN to map an image I depicting an object similar to S_i, to a point $P_I \in \mathbb{D}^-$ such that P_I is close to P_{S_i}. Our CNN is essentially required to learn the latent connection that exists between an image and the object it features.

An important characteristic of the CNN is its ability to generalize. A CNN trained to perform a certain task can, in many cases, be adapted to perform various other tasks. This property is highly advantageous as it is more efficient to re-purpose a well-trained network. In our setting, we require the network to learn a high-dimensional space. This is not a very common endeavor in the field, as most learning tasks revolve around different forms of classification. Therefore, re-purposing an existing network, AlexNet in our case, necessitates a few adaptations which will be discussed below. First, we describe our training data generation technique.

5.1 Training Image Synthesis

Training a CNN requires a large amount of diverse annotated data to form a general system that can perform well on unseen data. In our setting, we seek to learn a mapping between images and their corresponding points in the embedding space. Therefore, the point coordinates are the annotations, in the form of a long vector of real values. As such, obtaining manual annotations for the data is impractical.

An advantage of our scheme for placing both images and shapes in the same space is that it allows us to leverage the abundant information contained in the set of 3D shapes for training data generation. The instances in the set are represented as clean and complete meshes, allowing control and flexibility. As such, a shape $S_i \in \mathcal{S}$ can be used in a rendering process to generate a plethora of images R_i, each $R_{i,r} \in R_i$ featuring S_i in an image setting. As we have described previously, our set of 3D shapes facilitates the construction of an embedding space $\mathbb{D}^-$. In this space, a shape S_i is mapped to a corresponding point $P_{S_i} \in \mathbb{D}^-$. Hence, for each $R_{i,r}$, its annotation is automatically known to be P_{S_i}. The collection of all pairs $(R_{i,r}, P_{S_i})$ is the training data for our CNN.

CNN models are capable of approximating high dimensional and non-linear functions, as they infer millions of parameters. The representation power of the CNN can be unleashed only under training by a large amount of rich and diverse data. If the network is trained improperly, it tends to settle too closely to the training data in a process known as overfitting, rather than attempting to learn the general latent patterns that exist within the data. We follow [Su et al. 2015a] for generating training images that are resistant to overfitting. The images are rendered from the 3D shapes with rich variation in lighting and viewpoint, and then superimposed on random backgrounds.

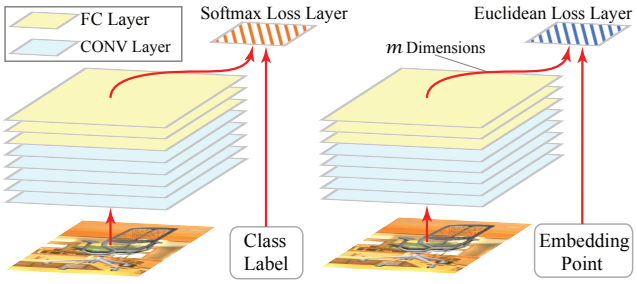

Figure 6: Network architecture. *We adapt AlexNet (left) for our image purification CNN (right)*[1]. *The last fully-connected layer (fc8) of AlexNet is set to output m (the dimensionality of the embedding space) real values, and the softmax loss layer is replaced by a Euclidean loss layer.*

In practice, we synthesize $\sim$ 1 million training images per category. Please refer to [Su et al. 2015a] for more details.

5.2 Network Architecture and Training

The training data consists of a collection of image-embedding pairs $(R_{i,r}, P_{S_i})$, where i indexes a shape and r indexes a synthesis configuration. The CNN is trained to map $R_{i,r}$ to its corresponding P_{S_i}. Disregarding any differences in viewpoint, lighting or background, all images $R_{i,r}$ generated from shape S_i are necessarily assigned the same embedding point, namely, P_{S_i}. For that reason we can think of our CNN as a "purifying" agent. It acts as a proxy between the original image and the object it contains by stripping the image of its distracting factors, such as lighting, viewpoint and background characteristics, and mapping it to a point in the embedding space corresponding to the object. It is this action which facilitates the comparison between images and shapes, as well as between images of varying appearances. Transferring an image to the shape embedding space and conducting any comparisons there, essentially simulates a comparison between pure 3D shapes.

Formally, our CNN is a function f that receives as input an image $R_{i,r}$ and is expected to output P_{S_i}. The actual output of f depends on the parameters of the network θ that are tuned during training. Hence, $f(R_{i,r};\theta) \rightarrow P_{S_i}$. We measure the mapping error with a Euclidean loss function:

$$L(\theta) = \sum_{i,r} \|f(R_{i,r};\theta) - P_{S_i}\|_2^2. \quad (3)$$

A discussion about the optimal CNN architecture for minimizing the given loss function is out of the scope of this paper. We adopt AlexNet [Krizhevsky et al. 2012] for our task and report our performance based on it. The input to AlexNet are 227×227 images, thus we render the 3D shapes such that their bounding boxes in the rendered images contain approximately 227×227 pixels. We modify the last fully-connected layer (fc8) to output an m-dimensional vector, which is the dimension of our embedding space $\mathbb{D}^-$, and switch the Softmax loss layer to an Euclidean loss layer (Eq. 3). More advanced CNNs, such as GoogLeNet [Szegedy et al. 2014], can also be adapted and used here, with potential performance improvements.

CNNs trained on ImageNet [Russakovsky et al. 2014] for classification tasks have been shown to generalize well by solving several other tasks when fine-tuned accordingly. In our setting, the training data is synthesized but the trained system is expected to handle and perform well on real-world images. Given the visual differences in appearance between real and synthesized images, it is unlikely that a CNN trained solely on synthesized data can successfully process real-world data [Yosinski et al. 2014]. Thus, rather than train a brand new network with all of its layers, we opt to fix the five pre-trained convolutional layers of AlexNet. These layers are responsible for detecting various features within the image. By reusing layers that were trained on a large and diverse set of real-world images such as ImageNet, we make sure that our system is able to process non-synthesized data correctly. We fine-tune the top three fully-connected layers of AlexNet, which are the more task-specific layers, and adapt them to fit our task.[2] The optimization usually converges after around $5,000$ iterations. We use a model trained for $10,000$ iterations for testing, and report its performance through various experiments in Section 6.

5.3 Mapping New Images

A trained purification CNN model maps images to the embedding space. To obtain the embedding for any image I, one must simply pass it through the network and retrieve the output vector as the embedded point $P_I = f(I;\theta) \in \mathbb{D}^-$. This process is fast, and can typically complete in under a second on a CPU, and even less on a GPU (tens of milliseconds). The same process cannot be applied on a new 3D shape that we seek to map to the space, since our CNN model is designed for images. However, our embedding space is robust enough to support our scheme of embedding new shapes into it as described in Section 4.3.

6 Experimental Results

To evaluate the efficacy and performance of our method, we conducted a few experiments and comparisons. These experiments are divided into three types, as dictated by the three central applications of our approach – cross-view image retrieval (Section 6.1), image-based shape retrieval (Section 6.2) and shape-based image retrieval (Section 6.3). The 3D shapes used in the experiments are from chair, airplane, and car category of ShapeNet 2015 summer release (v1.0), with 6778, 4045, and 7497 shapes, respectively.

6.1 Cross-view Image Retrieval

Given a query image of an object, our system can be used to retrieve images depicting similar objects. Since the similarity is computed in the 3D domain, it is inherently view invariant. The retrieved images can be quite diverse in terms of image appearance. They can be from a different viewpoint, on a variety of backgrounds, and in different colors and textures. However, all of them should feature objects that are similar, or even identical to that in the query image. Such a capability can cater to the intentions of a user in searching for images while focusing on object content.

More specifically, all the images I_i already mapped to the embedding space are associated with an embedding point P_{I_i}. Given a query image I_q, we first map it to a point P_{I_q} in the embedding space by feeding it into the network, as described in Section 5.3. Image retrieval can then be simply acquired by computing the neighborhood of P_{I_q} in $\{P_{I_i}\}$. Since $P_{I_q}, P_{I_i} \in \mathbb{R}^m$, where m is small, the nearest neighbor search can be efficiently computed.

[1] For simplicity, the pooling, local response normalization (LRN), rectified linear unit (ReLU) and dropout layers are not shown. Please refer to [Krizhevsky et al. 2012] for a full network definition.

[2] Fine-tuning the entire network registers an improvement of $\sim$1% in accuracy, at the cost of a substantial decrease in speed (x40). We opted to forgo this minor accuracy improvement to facilitate our experiments.

Figure 7: Example results for cross-view image retrieval. *The retrieved images share a similar type and style to that of the query image. However, they are often from very different viewpoints.*

	HoG	BoW	LLC	FisherVector	AlexNet fc7 (ImageNet)	AlexNet fc7 (fine tune)	Siamese (64 neighbors)	Siamese (0 neighbor)	Ours
Chair-clutter	0.698	0.681	0.690	0.665	0.706	0.724	0.691	0.701	**0.765**
Chair-clean	0.710	0.678	0.717	0.675	0.744	0.757	0.724	0.723	**0.801**
Car	0.278	0.280	0.283	0.270	0.287	0.293	0.285	0.259	**0.312**

Table 1: Performance comparison on the cross-view image retrieval task. *We compare the performance of our method on the cross-view image retrieval task against several other methods: HoG, BoW, LLC, FisherVector, AlexNet fc7 trained on ImageNet, AlexNet fc7 fine-tuned on the classification task with synthetic images from shapes belonging to the same class, Siamese network with positive image pairs sampled from* 64 *neighboring shapes and from only within same-shape renders (*0*-neighbor). We report the AUC value in this table.*

To evaluate the performance of our joint embedding on the cross-view image retrieval task, we compare our results to several other alternative methods, namely HoG [Dalal and Triggs 2005], BoW [Csurka et al. 2004], LLC [Wang et al. 2010], FisherVector [Sánchez et al. 2013], and AlexNet [Krizhevsky et al. 2012] (facilitated by the feature vector given by the layer fc7, trained on ImageNet for the 1000-class image classification task and fine-tuned on the same task with synthetic images rendered from shapes belonging to the same class). We also compare our result to that of a Siamese network trained on our synthesized data. The training data of a Siamese network consists of positive (similar) and negative (dissimilar) image pairs. The positive image pairs are sampled with a Gaussian distribution from synthesized images of neighboring shapes, i.e., more pairs are sampled from more synthesized images of more similar shapes. The k-closest shapes are considered to be neighboring shapes. The negative image pairs are randomly sampled from synthesized images of distant shapes. We compare the performance for both $k = 0$ (only images from the same shape are considered as positive pairs), as well as $k = 64$. We sampled $400,000$ positive pairs and $8,000,000$ negative pairs for the training of the Siamese network.

The comparison is conducted on a benchmark collected from ImageNet under the "chair" and "car" categories. It contains $1,309$ chair images with few distractions ("Chair-clean"), $5,874$ chair images with a large amount of clutter ("Chair-clutter"), and $5,758$ car images. All of the images are assigned category-level annotations. The pairwise distances between testing images can be computed by the aforementioned methods, for each of which we can compute a ranking list per image. We evaluate the precision and recall based on those ranking lists, by counting the images with a matching category to that of the query image that are in the top-k neighbors. By averaging through all the images for varying values of k, we obtain a precision-recall curve and compute the area under the curve as a performance indicator. We report the performances in Table 1.

Note that our joint embedding based approach consistently outperforms the other methods on all three datasets, clearly demonstrating the image purification capabilities of our purification CNN, and the advantage of the view-invariant embedding. The gap in performance between the Siamese network and our approach is of special interest. This gap may be potentially attributed to the inherent strength of an embedding space, designed and built based on 3D shape data, providing guidance for the CNN to learn the purification procedure. Conversely, in the Siamese network, the CNN must simultaneously learn two dependent functions — an embedding process and an image distraction removal process. This likely results in a significantly harder task.

We present our cross-view image retrieval results in Figure 7. Note that the retrieved images differ, to a varying extent, from the query image in their overall appearance. However, all of them feature a similar, and at times even identical, object to that which is featured in the query image.

6.2 Image-based Shape Retrieval

As mentioned, image-based shape retrieval has mostly been restricted to queries containing sketches or projections with a clean background. Our system extends this by allowing real-world images as queries, increasing ease-of-use for novice users.

We manually assemble a benchmark dataset for evaluating image-based shape retrieval. The benchmark dataset contains 105 shapes and 315 images. Each shape is associated with 3 images, each of

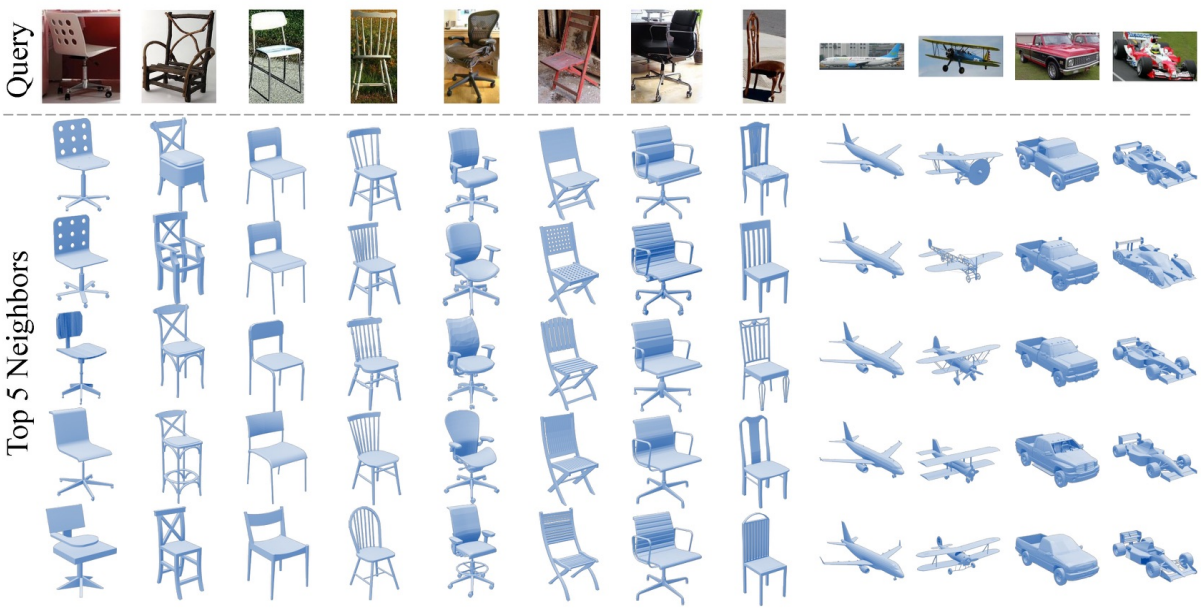

Figure 9: Example results for image-based shape retrieval. *Twelve real-world image queries are featured in the top-most row, followed by the* 5*-nearest models per query as retrieved by our method. The retrieved models share a similar type and style to that of the object depicted in the query.*

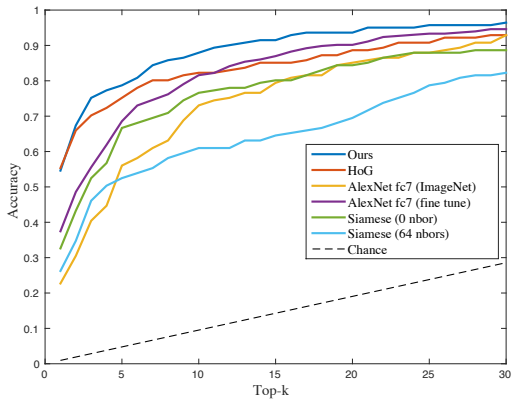

Figure 8: Comparison of top-k accuracy on image-based same-instance shape retrieval.

which features an object that is identical (or nearly identical) to that which its associated shape represents. The benchmark construction is time consuming. The user first browses Trimble 3D Warehouse, and identifies shapes containing a product name. The product name is then used as a text query in Google image search to retrieve relevant images. Finally, the user selects three images (of differing general appearance) featuring the shape object, if such exist. It took 20 human hours to assemble this benchmark.

We evaluate the image-based shape retrieval performance by the top-k instance retrieval accuracy on the benchmark. Note that the 105 exact matching shapes are excluded from the training data, to avoid them being simply "remembered" by the CNN models. We compare our joint embedding approach against HoG, AlexNet (pool5 and fc7 features), and Siamese. Since these models do not naturally support image-to-shape comparison, we create an intermediate layer in order to make them comparable to our method. We render the 3D shapes from 100 distinct viewpoints, and compute the HoG, AlexNet and Siamese features on these rendered images. The retrieval of these models is thus still computed in the image domain. The shape is considered to be retrieved if one of its 100 view images is retrieved. The comparison results are presented in Figure 8.

Our method is computationally more efficient compared to other approaches, since the comparison takes place in a rather low-dimensional space and typically takes tens of milliseconds to complete. Conversely, when using features such as the HoG descriptor, multiple views must be compared, summing up to a running time of several seconds.

Visual examples of our image-based shape retrieval results are presented in Figure 9. Note that the images are highly cluttered. Our CNN purifies them so that they are directly comparable to shapes.

6.3 Shape-based Image Retrieval

A joint embedding of shapes and images also facilitates the task of shape-based image retrieval. Given a 3D model, we can locate images featuring an object that is similar to the model, by searching the embedding space for images that are close to the model.

We utilize the same benchmark dataset mentioned in Section 6.2 for the evaluation of the shape-based image retrieval task. However, recall that in this dataset there is a many-to-one relationship between images and 3D shapes, therefore, there are multiple correct answers per query. Thus, instead of using top-k accuracy, which is not well defined in this case, we use median ranking of correct image retrieval as our evaluation metric. Specifically, for each of the 105 models, we sort the 315 test images according to their distances to the model in embedding space (in ascending order). We then locate the first and last correct image (of this model) in the sorted list and mark their ranking as a quantitative measure of our system performance on this query (the lower the better). To obtain the overall performance measure, we compute the median ranking of both the first and last matches. A low median ranking of the first

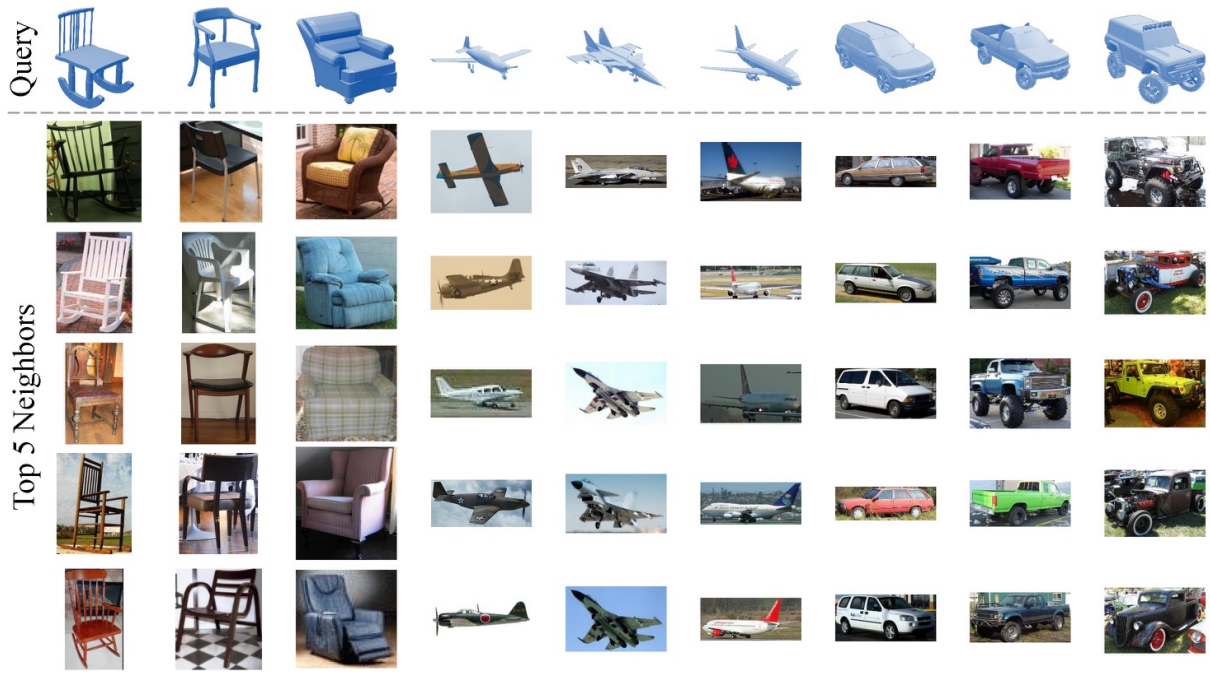

Figure 10: Example results for shape-based image retrieval. *3D shape queries from three categories are shown in the top row, each followed by the top-5 retrieved images. These images respect the shape of the query model in terms of the object contained in them, but they differ in overall appearance, as well as stylistic object characteristics, such as color. This property makes for an interesting application, as a user may wish to be shown the various real-world possibilities corresponding to a certain 3D model.*

Median rank of	HoG	AlexNet fc7 (ImageNet)	AlexNet fc7 (fine tune)	Siamese (64 nbors)	Siamese (0 nbor)	Ours
first matched	**1**	7	5	3	3	**1**
last matched	32	84	71	94	49	**5**

Table 2: Comparison of performance on shape-based same-instance image retrieval. *For each model there are multiple images with the same instance. We sort the retrieved images of a model and compute the rankings of the first and last image matches. Median ranking (the lower the better) is the evaluation metric.*

image match (rank $= 1$ is optimal) reflects a high "accuracy" score, and a low median ranking of the last image match implies that the retrieval system has good "recall" (all the images corresponding to the query model appear within the top ranked results).

Quantitative results of shape-based image retrieval are shown in Table 2. The testing procedure is similar to that in Section 6.2. While the HoG approach and our embedding approach share a similar median ranking for the first image match, the embedding approach outperforms on the last image match median ranking. This is likely due to the association between images and shapes in our embedding space, which is oblivious to differences in image appearance. HoG features on the other hand, do not support such an invariance, and as a result, are unlikely to recognize cluttered images as a good match, despite the nature of the object contained in them. Visual results are presented in Figure 10.

7 Applications

The retrieval tasks discussed in Section 6 are in and of themselves important and powerful applications. In this section, we present several more applications that further utilize the high-performance

Figure 11: Shape-guided image editing. *Our image-based shape retrieval can boost an image editing process by shape proxies. The retrieved 3D shape is used here for computing a shadow casting of the object in the image (chair), while being lit by an additional light source (floor lamp).*

retrieval abilities of our method. The joint embedding of both shapes and images serves as an essential building block in all of these. Moreover, some of the following applications are completely new and only made possible by this joint embedding.

7.1 Shape-Guided Image Editing

Images are the most popular media form for capturing the 3-dimensional world around us. Manipulating an image in a manner that respects and preserves the real-world characteristics of the image, is naturally desirable. However, pixel/patch-level image manipulation operators are often insufficient, as it tediously falls to the user to follow and simulate the underlying real-world phenomena. This impediment can be alleviated with the help of our method. The retrieval of a 3D shape that approximates an object depicted in an image under edit, can provide assistance in the form of 3D clues to guide real-world-based editing operations. Figure 11 presents an example where a retrieved model assists in computing a shadow

Figure 12: Interactive image-based scene modeling. *Starting from an image with user-marked object bounding boxes (upper left), our system retrieves similar 3D models. Together with 3D poses estimated by [Su et al. 2015a], a 3D scene (lower left) can be modeled with a layout reasoning module from [Choi et al. 2013]. The model retrieval and pose estimation are conducted in real-time, and the user is given instant feedback once a bounding box is drawn. Equipped with such a system, 2D scene images can be easily lifted into 3D and rendered from novel views (right).*

casting that can then be applied to the original image. This application shares the same merit as that in [Zheng et al. 2012]. However, our 3D shape proxies are automatically retrieved, and are of a finer-grained level of detail to that which is supplied by human annotated cuboid proxies.

7.2 Interactive Image-based Modeling

The image to shape connection provided by our joint embedding brings scene images and 3D scenes closer together. We propose an interactive image-based modeling approach based on our joint embedding (see Figure 12). In the modeling session, the user is first asked to mark object bounding boxes. Our system then retrieves a list of similar 3D models and presents it to the user to select and add to the scene. Object poses and scene layout can be automatically estimated by recent, relevant work [Su et al. 2015a; Choi et al. 2013]. Together with our joint embedding based retrieval, user effort is mostly reduced to object bounding box annotation and scene refinements.

7.3 Text-to-Shape via Image Proxy

Text-to-image retrieval is a commonly used application in our daily lives. A text query is used to search image databases for instances containing labels that match the text query. A plethora of images available on the Internet are retrievable to everyone, thanks to meaningful and descriptive labels and annotations that are attached to the images. Unfortunately, the amount of 3D shapes that are accompanied by descriptive labels is limited, making it difficult to directly infer a mapping between the two modalities. Our system bridges the gap between images and 3D shapes, thereby enabling text-to-shape queries by going through images serving as proxies. To retrieve shapes from a text query, we first retrieve relevant images from a public image search engine. The top-N retrieved images then jointly vote in our shape-image embedding space, and shapes are ranked by their votes. We compare the text-to-shape retrieval provided by Trimble 3D Warehouse (possibly with labels associated to shapes), and the text-to-shape retrieval powered by our joint embedding via image proxies in Figure 13. Averbuch-Elor et al. [2015] propose an approach for generating 3D shapes from a text query, but the final 3D shape is generated from limited views of the

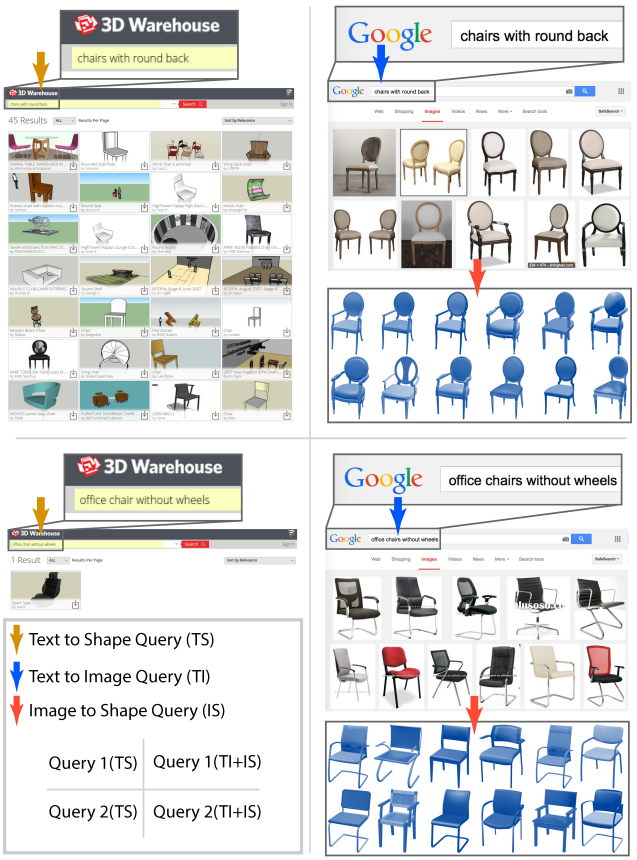

Figure 13: Text-based shape retrieval. *Our joint shape-image embedding space maps images to shapes. This mapping can be concatenated to the established text-to-image mapping, obtaining a text-to-shape mapping. We show a text-to-shape retrieval provided by Trimble 3D Warehouse, and powered by our joint embedding via image proxies.*

query object, and thus lacks detail. In our approach, the quality of the retrieved shapes is defined by the large shape collection, which often contains close approximations to real-world objects.

7.4 Limitations and Future Work

Shape Database Purification. The 3D shape collections used in our experiments originate from Trimble 3D Warehouse, with annotations from ShapeNet. The 3D models are created by artists or students of varying levels of skill, and for diverse purposes. This results in 3D models that may not accurately capture real-world objects. This is often a setback for Computer Graphics and Computer Vision tasks that rely on shape proxies of real-world objects. Despite the ever-growing availability of 3D models, it is still surpassed by several orders of magnitude by the availability of image data. Thus, statistic information computed on image data is far more stable and accurate than similar statistics computed for 3D shape data. Such statistics, after transferred into the shape space, via the joint embedding space, can be important for purifying 3D shape databases by filtering out shapes that are not well associated with real-world images.

Online Dynamic Embedding Space. Our current scheme dictates that a constructed embedding space is fixed. New shapes and

images can be projected into this space and compared against any other embedded entity. However, these new shapes do no contribute any additional and potentially enriching information to the embedding space. In order to take advantage of new shapes and images, the embedding space has to be rebuilt, and the CNN has to be further fine-tuned to capture the new embedding space. It is interesting to explore an online dynamic embedding space, where new shapes and new images can be continuously integrated into the embedding space, evolving it progressively.

Multi-modal Entities Sharing an Embedding Space. In our approach, both 3D shapes and 2D images are represented as points in a joint embedding space. However, 3D shapes belong to the 3D domain, while images belong to the 2D domain, and include projections of 3D shapes. From this perspective, if images are represented as points in an embedding space, one may consider representing a 3D shape as a collection of points, or a hyperplane, in the embedding space. That way, the status of an image entity as a projection of a 3D shape can be transferred and preserved in the embedding space as well.

Multi-faceted Similarity Measurement. Similarity is a multi-facet concept. The degree of similarity between two entities may vary under different criteria as specified by different applications. In our HoG-based similarity measure, the similarity between two shapes is captured by a real value scalar. Extending the measure to a vector may achieve a better approximation of the similarity phenomenon, catering to various criteria and reducing the subjectivity that similarity estimation naturally involves.

What Distance Metrics Can CNNs Learn? The structure of a CNN strongly utilizes and relies on the spatial formation between pixels in an image. The HoG descriptors used for similarity estimation are computed in a spatial manner, potentially aligning with the effectiveness with which the CNN has learned the embedding space induced by them. Distance metrics such as the persistent barcodes [Carlsson et al. 2004] can be purely topology-based, and it is interesting to explore whether a CNN is capable of successfully learning a function that is based on a distance metric with properties that are not necessarily spatial.

8 Conclusion

At the center of our approach is a joint embedding space for both 3D shapes and 2D images. The distances between embedded entities in this space reflect the similarities between their underlying objects. Therefore, this joint embedding creates a computational link between the 3D shape domain and the 2D image domain, facilitating any comparison between multi-modal entities. A robust distance metric designed for the 3D domain, and applied on a 3D shape collection, is at the core of the embedding space construction, contributing to the robustness and reliability of the space. The mapping of images to the shape-based embedding space is accomplished by a purifying CNN trained on a comprehensive collection of synthesized images, attenuating the distracting properties that are prevalent in real-world images. Our approach thus bridges the computational gap between the two domains, and serves as an essential building block for many existing Computer Graphics applications, as well as supporting and inspiring new ones.

Backed by the strong performance observed in various experiments using our robust embedding and its surrounding modules, we believe that this approach is the first step in the important undertaking involved in establishing a strong link between the 3D shape and 2D image domains. As such, we trust in the potential of this approach to lead to new research directions and exploration of relevant and novel techniques.

For full reproducibility of our method and to encourage further development based upon it, we open source all our data and code at `http://shapenet.github.io/JointEmbedding/`.

Acknowledgements

We are grateful to Matthias Nießner and Matthew Fisher for insightful discussions, Amit Bermano for proofreading the paper and the reviewers for invaluable comments and suggestions.

We would like to acknowledge the support of NSFC grant 61202221, NSF grant DMS 1228304, CCF 1161480 and IIS 1528025, ONR MURI grant N00014-13-1-0341, a Google Focused Research Award, the Max Planck Center for Visual Computing and Communications, K40 GPU donations from NVIDIA Corporation, and a gift from the Apple Corporation.

References

AUBRY, M., MATURANA, D., EFROS, A. A., RUSSELL, B. C., AND SIVIC, J. 2014. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, IEEE, 3762–3769.

AVERBUCH-ELOR, H., WANG, Y., QIAN, Y., GONG, M., KOPF, J., ZHANG, H., AND COHEN-OR, D. 2015. Distilled collections from textual image queries. *Computer Graphics Forum 34*, 2, 131–142.

BELL, S., AND BALA, K. 2015. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph. 34*, 4 (July), 98:1–98:10.

CARLSSON, G., ZOMORODIAN, A., COLLINS, A., AND GUIBAS, L. 2004. Persistence barcodes for shapes. In *SGP*, ACM, 124–135.

CHATFIELD, K., SIMONYAN, K., VEDALDI, A., AND ZISSERMAN, A. 2014. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, British Machine Vision Association.

CHEN, D.-Y., AND OUHYOUNG, M. 2002. A 3d object retrieval system based on multi-resolution reeb graph. In *Proc. of Computer Graphics Workshop*, vol. 16.

CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3d model retrieval. *Computer Graphics Forum 22*, 3, 223–232.

CHOI, W., CHAO, Y.-W., PANTOFARU, C., AND SAVARESE, S. 2013. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, IEEE, 33–40.

CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J., AND BRAY, C. 2004. Visual categorization with bags of keypoints. In *ECCV*, vol. 1, IEEE, 1–2.

CYR, C. M., AND KIMIA, B. B. 2001. 3d object recognition using shape similiarity-based aspect graph. In *ICCV*, vol. 1, IEEE, 254–261.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, vol. 1, IEEE, 886–893.

FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3d models. *ACM Trans. Graph. 22*, 1, 83–105.

GAL, R., SHAMIR, A., AND COHEN-OR, D. 2007. Pose-oblivious shape signature. *TVCG 13*, 2, 261–271.

HADSELL, R., CHOPRA, S., AND LECUN, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, vol. 2, IEEE, 1735–1742.

HE, X., CAI, D., YAN, S., AND ZHANG, H.-J. 2005. Neighborhood preserving embedding. In *ICCV*, vol. 2, IEEE, 1208–1213.

HERZOG, R., MEWES, D., WAND, M., GUIBAS, L., AND SEIDEL, H.-P. 2015. Lesss: Learned shared semantic spaces for relating multi-modal representations of 3d shapes. *Computer Graphics Forum 34*, 5, 141–151.

HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. of SIGGRAPH '01*, ACM, 203–212.

HUANG, Q.-X., SU, H., AND GUIBAS, L. 2013. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. Graph. 32*, 6 (Nov.), 190:1–190:10.

HUANG, Q., WANG, H., AND KOLTUN, V. 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph. 34*, 4.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *SGP*, vol. 6, ACM.

KHOLGADE, N., SIMON, T., EFROS, A., AND SHEIKH, Y. 2014. 3d object manipulation in a single photograph using stock 3d models. *ACM Trans. Graph. 33*, 4 (July), 127:1–127:12.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*. Curran Associates, Inc., 1097–1105.

KRUSKAL, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1, 1–27.

LECUN, Y., BENGIO, Y., AND HINTON, G. 2015. Deep learning. *Nature 521*, 436–444.

LEE, J., KIM, Y., LEE, S., KIM, B., AND NOH, J. 2015. High-quality depth estimation using an exemplar 3d model for stereo conversion. *IEEE TVCG PP*, 99, 1–1.

LIU, D. C., AND NOCEDAL, J. 1989. On the limited memory bfgs method for large scale optimization. *Math. Program. 45*, 3 (Dec.), 503–528.

LIU, Y., ZHANG, D., LU, G., AND MA, W.-Y. 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition 40*, 1, 262–282.

LOFFLER, J. 2000. Content-based retrieval of 3d models in distributed web databases by visual shape information. In *InfoVis*, IEEE, 82–87.

OHBUCHI, R., MINAMITANI, T., AND TAKEI, T. 2005. Shape-similarity search of 3d models by using enhanced shape functions. *International Journal of Computer Applications in Technology 23*, 2, 70–85.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph. 21*, 4, 807–832.

ROWEIS, S. T., AND SAUL, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500, 2323–2326.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., ET AL. 2014. Imagenet large scale visual recognition challenge. *IJCV*, 1–42.

SAMMON, J. W. 1969. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers 18*, 5, 401–409.

SÁNCHEZ, J., PERRONNIN, F., MENSINK, T., AND VERBEEK, J. 2013. Image classification with the fisher vector: Theory and practice. *IJCV 105*, 3, 222–245.

SMEULDERS, A. W., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *TPAMI 22*, 12, 1349–1380.

SU, H., HUANG, Q., MITRA, N. J., LI, Y., AND GUIBAS, L. 2014. Estimating image depth using shape collections. *ACM Trans. Graph. 33*, 4 (July), 37:1–37:11.

SU, H., QI, C. R., LI, Y., AND GUIBAS, L. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, IEEE.

SU, H., YI, E., SAVVA, M., CHANG, A., SONG, S., YU, F., LI, Z., XIAO, J., HUANG, Q., SAVARESE, S., FUNKHOUSER, T., HANRAHAN, P., AND GUIBAS, L., 2015. Shapenet: An ongoing effort to establish a richly-annotated, large-scale dataset of 3d shapes. `http://shapenet.org`.

SUNDAR, H., SILVER, D., GAGVANI, N., AND DICKINSON, S. 2003. Skeleton based shape matching and retrieval. In *Shape Modeling International, 2003*, IEEE, 130–139.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2014. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*.

TANGELDER, J. W., AND VELTKAMP, R. C. 2008. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications 39*, 3, 441–471.

WANG, J., YANG, J., YU, K., LV, F., HUANG, T., AND GONG, Y. 2010. Locality-constrained linear coding for image classification. In *CVPR*, IEEE, 3360–3367.

WESTON, J., BENGIO, S., AND USUNIER, N. 2010. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning 81*, 1, 21–35.

WESTON, J., BENGIO, S., AND USUNIER, N. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *International Joint Conference on Artificial Intelligence*, AAAI Press, 2764–2770.

XU, K., ZHENG, H., ZHANG, H., COHEN-OR, D., LIU, L., AND XIONG, Y. 2011. Photo-inspired model-driven 3d object modeling. *ACM Trans. Graph. 30*, 4 (July), 80:1–80:10.

YOSINSKI, J., CLUNE, J., BENGIO, Y., AND LIPSON, H. 2014. How transferable are features in deep neural networks? In *NIPS*. Curran Associates, Inc., 3320–3328.

ZHENG, Y., CHEN, X., CHENG, M.-M., ZHOU, K., HU, S.-M., AND MITRA, N. J. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph. 31*, 4 (July), 99:1–99:11.