Ensemble of PANORAMA-based Convolutional Neural Networks for 3D Model Classification and Retrieval

Konstantinos Sfikas[a,*], Ioannis Pratikakis[b], Theoharis Theoharis[a]

[a]*NTNU - Norwegian University of Science and Technology, Department of Computer Science, Trondheim, Norway*
[b]*Democritus University of Thrace, Department of Electrical and Computer Engineering, Xanthi, Greece*

ARTICLE INFO

Article history:
Received January 23, 2019

Keywords: 3D Object Classification, 3D Object Retrieval, Panoramic views, Convolutional Neural Network, Neural Network Ensemble

ABSTRACT

A novel method for the classification and retrieval of 3D models is proposed; it exploits the 2D panoramic view representation of 3D models as input to an ensemble of Convolutional Neural Networks which automatically compute the features. The first step of the proposed pipeline, pose normalization is performed using the SYMPAN method, which is also computed on the panoramic view representation. In the training phase, three panoramic views corresponding to the major axes, are used for the training of an ensemble of Convolutional Neural Networks. The panoramic views consist of 3-channel images, containing the Spatial Distribution Map, the Normals' Deviation Map and the magnitude of the Normals' Devation Map Gradient Image. The proposed method aims at capturing feature continuity of 3D models, while simultaneously minimizing data preprocessing via the construction of an augmented image representation. It is extensively tested in terms of classification and retrieval accuracy on two standard large scale datasets: ModelNet and ShapeNet.

1. Introduction

In the recent past, convolutional neural networks (CNN) have shown their superiority against humans in computing features, while they are very sensitive to the input representation. In this work an extension of the PANORAMA 3D shape representation, previously proposed by our team (Papadakis et al., 2010), is exploited as the input representation to a CNN for computing descriptor features for 3D object classification and retrieval.

The 3D models are initially pose normalized using the SYMPAN pose normalization algorithm, (Sfikas et al., 2014) which is based on the use of reflective symmetry on their panoramic view images. Next, an augmented panoramic view is created and used to train the convolutional neural network. This augmented panoramic view consists of the spatial and orientation components of PANORAMA, (see 3.1.1), along with the magnitude of the gradient image which is extracted from the orientation component. A reduction in the size of the augmented panoramic view representation is shown to benefit the training procedure.

The motivation behind the aforementioned method is that the PANORAMA representation is able to bridge the dimensionality gap between 3D object space and the 2D image input that is typically suitable for a convolutional neural network, in a very efficient manner. PANORAMA has already proven to be a successful hand-crafted 3D model descriptor that has achieved state-of-the-art 3D model retrieval performance in various implementations, (Papadakis et al., 2010; Sfikas et al., 2014, 2013a, 2016). It has also been used as input to a successful pose normalization method, SYMPAN (Sfikas et al., 2014) (briefly detailed in 3.1.2).

This work constitutes an extension of the method presented in (Sfikas et al., 2017). The novel elements are: (a) a new 3-channel input schema representation that contains the Spatial Distribution Map, the Normals' Deviation Map and the magnitude of the Normals' Devation Map Gradient Image; (b) an ensemble of Convolutional Neural Networks architecture along with an analysis of various parameters that were tested for evaluation purposes; (c) an extended evaluation of the proposed method on an additional large scale dataset, namely the ShapeNetCore 3D model dataset which is specifically aimed at machine learning; since many recent related works have been tested on this dataset, including the participants of the SHREC2017 and SHREC2016 *Large-scale 3D Shape Retrieval from ShapeNet Core55* tracks, (Savva et al., 2016, 2017), this

*Corresponding author: Tel.: +30-6944126968;
e-mail: ksfikas@di.uoa.gr (Konstantinos Sfikas), konstantinos.sfikas@ntnu.no (Konstantinos Sfikas)

strengthens the comparability of the proposed method; (d) the expansion of the Related Work section with additional works on machine learning for 3D model categorization and retrieval; this was necessary since several works recently appeared on the relevant evaluation datasets.

The performance of the proposed method is evaluated in terms of accuracy on both 3D model classification and retrieval. The datasets used for the evaluation are the publicly available Princeton ModelNet 3D CAD model dataset, (Wu et al., 2015) and the ShapeNet Core55 subset of the ShapeNet dataset, (Chang et al., 2015). These datasets are designed for machine learning algorithms, containing both training and testing partitions (the ShapeNet dataset also includes a validation partition).

The remainder of this paper is organized as follows: Section 2 briefly discusses recent works on 3D model classification and retrieval with emphasis on deep neural network methods. A brief review of recent pose normalization methods is also given. Section 3 details the proposed method. Section 4 presents the experimental procedure along with the corresponding results. Finally, in Section 5 conclusions are drawn and future work is discussed.

2. Related Work

One way of classifying 3D shape representation methods is based on the dimensionality of the representation: (a) 2D image-based representations (i.e. planar and panoramic projections) using global and/or local descriptors, (b) 3D model-based representations (i.e. 3D shapes, point clouds and voxels) and (c) higher levels of data representations (i.e. 3D videos, doxels etc). Recent works of the first two categories will be discussed in the sequel, as these are most relevant to the problem at hand.

2.1. 2D image-based representation Methods

One of the earliest methods for 3D object retrieval, based on the extraction of features from 2D representations of the 3D objects, was the Light Field descriptor, proposed by Chen et al. (Chen et al., 2003a). This descriptor comprises Zernike moments and Fourier coefficients computed on a set of projections taken at the vertices of a dodecahedron. Su et al. (Su et al., 2015) present a CNN architecture that combines information from multiple views of a 3D shape into a single and compact shape descriptor. They show that this descriptor is able to achieve higher recognition performance than single image recognition architectures. Papadakis et al. in (Papadakis et al., 2010) propose PANORAMA, a 3D shape descriptor that uses a set of panoramic views of a 3D object which describe the position and orientation of the object's surface in 3D space. For each view the corresponding 2D Discrete Fourier Transform and the 2D Discrete Wavelet Transform are computed. Shi et al. in (Shi et al., 2015), convert each 3D shape into a panoramic view, namely a cylinder projection around its principal axis. Then, a variant of CNN is used for learning the representations directly from these views. A row-wise max-pooling layer is inserted between the convolution and fully-connected layers, making the learned representations invariant to the rotation around a principal axis. In (Shi et al., 2015), the authors use panoramic views that feed a CNN for 3D model categorization and retrieval. Although similar to PANORAMA, the authors do not use the two different representations of PANORAMA (one distance based and one angle-based), nor the three standard projection axes. Furthermore, although rotation invariance on one axis is achieved through a specially designed layer of the proposed CNN architecture, it is not described how the key problem of pose normalization is solved. (Panoramic views change drastically as the orientation of a 3D model varies). Kanezaki et al. (Kanezaki, 2016)* propose RotationNet, a Convolutional Neural Network-based model that takes multiple views of an object as input and estimates both its pose and object category. The method treats the pose labels as latent variables, which are optimized to self-align in an unsupervised manner during the training using an unaligned dataset. The proposed pose alignment strategy enables one to obtain view-specific feature representations shared across classes. In (Bai et al., 2016), Bai et al. present a real-time 3D shape search engine based on the projective images of 3D shapes. The authors utilize efficient projection and view feature extraction using GPU acceleration. A first inverted file, referred as F-IF, is utilized to speed up the procedure of multi-view matching and a second inverted file (S-IF), which captures a local distribution of 3D shapes in the feature manifold, is adopted for efficient context-based reranking. As a result, for each query the retrieval task can be finished very fast, despite the necessary cost of IO overhead. The method is named GIFT, **G**PU acceleration and **I**nverted **F**ile **T**wice. The method of Tatsuma and Aono, as presented in (Savva et al., 2017), consists of feature extraction from a Convolutional Neural Network (CNN) with reduced number of filters for depth-buffer images and similarity calculation by an improved method of Neighbor Set Similarity (NSS), (Bai et al., 2015). The authors extract the feature vector of a 3D model by inputting the rendered depth-buffer images to a CNN. Initially, the translation, scale and then the rotation of the 3D models is normalized by using Point SVD, (Tatsuma and Aono, 2009). Next, the method renders 38 depth-buffer images at 224×224 resolution by setting the view point at each vertex of a unit geodesic sphere. Finally, the feature vector of a 3D model is obtained by averaging the CNN output vectors, which denote the classification probability, of each depth buffer image. For the dissimilarity between two feature vectors, the Euclidean distance is employed. Sedaghat et al. in (Sedaghat et al., 2016)* approach the category-level classification task as a multi-task problem, in which the network is forced to predict the pose of the object in addition to the class label. The authors show that this yields significant improvements in the classification results. They implement different network architectures for this purpose and test them on different datasets representing various 3D data sources: LiDAR data, CAD models and RGBD images.

Ohbuchi et al. in (Ohbuchi et al., 2008), based on multi-scale local visual features, describe a shape-based 3D model retrieval method. Features are extracted from 2D range images of 3D models viewed from uniformly sampled locations on a sphere. The method is view-based, and is able to handle all models that

	Methods evaluated on the ModelNet dataset(s)	Methods evaluated on the ShapeNetCore dataset
2D	MVCNN, (Su et al., 2015)	RotationNet, (Kanezaki, 2016)*
	DeepPano, (Shi et al., 2015)	GIFT, (Bai et al., 2016)
	LFD, (Chen et al., 2003a)	ReVGG, (Savva et al., 2017)
	PANORAMA, (Papadakis et al., 2010)	MVCNN, (Su et al., 2015)
	GIFT, (Bai et al., 2016)	MVCNN Multires, (Qi et al., 2016a)*
	ORION, (Sedaghat et al., 2016)*	
	Fusion-Net, (Hegde and Zadeh, 2016)*	
	MVCNN Multires, (Qi et al., 2016a)*	
	PANORAMA-NN, (Sfikas et al., 2017)	
3D	3D ShapeNets, (Wu et al., 2015)	DLAN, (Furuya and Ohbuchi, 2016)
	Geometry Image, (Sinha et al., 2016)	MVCNN Multires, (Qi et al., 2016a)*
	SPH, (Kazhdan et al., 2003)	
	Set-Convolution, (Ravanbakhsh et al., 2016)*	
	3D-GAN, (Wu et al., 2016)*	
	VRN Ensemble, (Brock et al., 2016)*	
	Fusion-Net, (Hegde and Zadeh, 2016)*	
	VoxNet, (Maturana and Scherer, 2015)	
	PointNet-Garcia, (Garcia-Garcia et al., 2016)	
	MVCNN Multires, (Qi et al., 2016a)*	
	FPNN, (Li et al., 2016)*	
	Klokov & Lempitsky, (Klokov and Lempitsky, 2017)*	
	Xu & Todorovic, (Xu and Todorovic, 2016)*	

Table 1: Method categorization based on evaluation dataset and dimensionality of the descriptor (2D or 3D). Methods indicated by an (*) are arXiv versions, at the time of writing

can be rendered as a range image. For each range image, a set of 2D multi-scale local visual features is computed by using the SIFT algorithm. To reduce the cost of distance computation and feature storage, a set of local features that describe a 3D model is integrated into a histogram, using the Bag-Of-Features approach.

In (Lian et al., 2013), Lian et al., propose a visual similarity-based 3D shape retrieval method (CM-BOF) using Clock Matching and Bag-of-Features. Initially, pose normalization is applied to each 3D model to generate its canonical pose, then the normalized object is represented by a set of depth-buffer images captured on the vertices of a geodesic sphere. Each image is described as a word histogram obtained by the vector quantization of the corresponding salient local features. Finally, a multi-view shape matching scheme is employed to measure the dissimilarity between two models.

In (Sfikas et al., 2016), the authors present a method for partial matching and retrieval of 3D objects based on range image queries. The proposed methodology addresses the retrieval of complete 3D objects using range image queries that represent partial views. The base method relies upon Bag-of-Visual-Words modelling and enhanced Dense SIFT descriptor computed on local features of PANORAMA views and range image queries.

2.2. *3D model-based representation Methods*

In (Kazhdan et al., 2003), Kazhdan proposes the Spherical Harmonic Representation, a rotation invariant representation of spherical functions in terms of the energies at different frequencies. This descriptor is a volumetric representation of the Gaussian Euclidean Distance Transform of a 3D object, expressed by norms of spherical harmonic frequencies. Wu et al. (Wu et al., 2015) propose to represent a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid, using a Convolutional Deep Belief Network. Sinha et al. (Sinha et al., 2016) propose an approach of converting the 3D shape into a 'geometry image' so that standard CNNs can directly be used to learn 3D shapes, thus bridging the associated representation gap. Geometry images using an authalic parametrization are created on a spherical domain. This spherically parameterized shape is then projected and cut to convert the original 3D shape into a flat and regular geometry image. The algorithm proposed in (Furuya and Ohbuchi, 2016) aims at extracting 3D shape descriptors that are robust against geometric transformations including translation, uniform scaling, and rotation of 3D models. The algorithm is called Deep Local feature Aggregation Network (DLAN). DLAN takes as its input a set of low-level 3D geometric features having invariance against 3D rotation. It produces a compact, high-level descriptor per 3D model for efficient and effective matching among 3D shapes. The DLAN pipeline consists of the following steps: Generating oriented point set, Extracting rotation invariant local features, Aggregating local features and Comparing aggregated features. In (Ravanbakhsh et al., 2016)*, Ravanbakhsh et al. introduce a simple permutation equivariant layer for deep learning with set structure. This type of layer, obtained by parameter-sharing, has a simple implementation and linear-time complexity in the size of each set. Deep permutation-invariant networks are used to perform point-could classification and MNIST-digit summation, where in both cases the output is invariant to permutations of the input. In (Wu et al., 2016)* Wu et al., study the problem of 3D object generation. They propose 3D Generative Adversarial Network (3D-GAN), which generates 3D objects from a probabilistic space by leveraging recent advances in volumetric convolutional networks and generative adversarial nets. The proposed model uses an adversarial criterion, instead of traditional heuristic criteria. Brock et al. (Brock et al., 2016)* explore voxel-based models, and present evidence for the viability of voxelized representations in applications including shape modeling and object classification. The contributions are methods for training voxel-based variational autoencoders, a user interface for exploring the latent space learned by the autoencoder, and a deep convolutional neural network architecture for object classification. In (Hegde and Zadeh, 2016)*, Hegde and Zadeh, tackle the object recognition problem by using Convolutional Neural Networks on two different data representations: a volumetric representation and a pixel representation. Their aim is to bridge the gap between the efficiency of the above two representations. They combine both representations and exploit them to learn new features, which yield a significantly better classifier than using either of the representations in isolation. To this end, they introduce the Volumetric CNN (V-CNN) architecture. In (Maturana and Scherer, 2015), Maturana and Scherer propose VoxNet, an architecture for tackling the problem of robust object recognition by integrating a volumetric Occupancy Grid representation with a supervised 3D Convolutional Neural Network (3D CNN). In (Garcia-Garcia et al., 2016), Garcia et al. propose PointNet, an approach inspired by VoxNet and 3D ShapeNets, as an improvement over existing methods by using density occupancy grid representations for the input data, and integrating them into a supervised Convolutional Neural Network architecture. Qi et al. in (Qi et al., 2016a)* aim to improve both volumetric CNNs and multi-view CNNs by introducing two distinct network architectures of volumetric CNNs. In addition, the authors examine multi-view CNNs, where they introduce multi-resolution filtering in 3D. In (Li et al., 2016)*, Li et al. represent 3D spaces as volumetric fields, and propose a novel design that employs field probing filters to efficiently extract features from them. Each field probing filter is a set of probing points - sensors that perceive space. Their learning algorithm optimizes the weights associated with the probing points and also their locations, which deforms the shape of the probing filters and adaptively distributes them in 3D space. Klokov and Lempitsky present a new deep learning architecture (called Kd-network) that is designed for 3D model recognition tasks and works with unstructured point clouds (Klokov and Lempitsky, 2017)*. The new architecture performs multiplicative transformations and shares parameters of these transformations according to the subdivisions of the point clouds imposed onto them by Kd-trees. Unlike the current convolutional architectures that usually require rasterization on uniform two-dimensional or three-dimensional grids, Kd-networks do not rely on such grids and thus exhibit better scaling behaviour. To address the issue of efficiently recognizing voxelized 3D shapes of large magnitude, (Xu and Todorovic, 2016)* formulates CNN learning as a beam search aimed at identifying an optimal CNN architecture, namely, the number of layers, nodes, and their connectivity in the network, as well as estimating parameters of such an optimal CNN. Each state of the beam search corresponds to a candidate CNN. Two types of actions are defined to add new convolutional filters or new convolutional layers to a parent CNN, and thus transit to children states. The utility function of each action is efficiently computed by transferring parameter values of the parent CNN to its children, thereby enabling an efficient beam search.

A categorization of the aforementioned methods based on the representation dimensionality and the dataset that they have been evaluated on (see section 4.1) is presented in Table 1. Methods indicated by an (*) are arXiv versions, at the time of writing.

The view-based method presented in our previous work (Sfikas et al., 2017) is based on the successful hand-crafted PANORAMA descriptor representation, extending its usage based on CNNs. The method, in a manner similar to, but in many ways extending (Shi et al., 2015), feeds a CNN with the PANORAMA representation (both spatial and orientation) for the three principal projection axes. In addition, it uses a PANORAMA-based pose normalization method (Sfikas et al., 2014). In (Shi et al., 2015) the spatial component of a single panoramic view was used and pose normalization was apparently not performed (except for rotation invariance with respect to the 2D panoramic image projection axis). Table 2 summarizes the differences between the proposed method, /hldenoted as **PANORAMA-ENN**, (**ENSEMBLE NEURAL NETWORK**) for the remainder of this paper, and the method of (Shi et al., 2015), (denoted as DeepPano).

	PANORAMA-ENN	DeepPano
2D Image Representation	**spatial, orientation**	spatial
Projection Axes	$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$	one axis
Pose Normalization Axes	$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$	one axis

Table 2: Differences between the proposed PANORAMA-(E)NN and DeepPano (Shi et al., 2015).

2.3. Pose Normalization Methods

The best-known approach for computing the alignment of 3D models is Principal Component Analysis (PCA) or Karhunen - Loeve transformation (Paquet et al., 2000; Shilane et al., 2004; Theodoridis and Koutroumbas, 1999; Vranić et al., 2001; Zaharia and Prêteux, 2004). The PCA algorithm, based on the computation of 3D model moments estimates the principal axes of a 3D model that are used to determine its orientation. In its original form, PCA can be imprecise and often the principal axes of 3D models that belong to the same class produce poor alignments (Chen et al., 2003b). To alleviate these problems, Vranic introduces an improvement to the original method, the Continuous PCA (CPCA) algorithm (Vranic, 2004; Vranić et al., 2001; Vranic, 2005). Based on the continuous triangle set of a 3D model, CPCA computes the principal axes. Similar to the CPCA method, Papadakis et al. propose the Normal PCA (NPCA) algorithm (Papadakis et al., 2007, 2008). NPCA computes the principal axes of the 3D model based on its surface normal set. Related to PCA is the use of Singular Value Decomposition (SVD) for alignment (Theodoridis and Koutroumbas, 1999).

Another major category of normalization methods exploits symmetry features that are present in a large number of 3D models. Kazhdan et al. (Kazhdan et al., 2002a) defines a reflective symmetry descriptor that represents a measure of reflective symmetry for an arbitrary 3D voxel model, for all planes through the model's center of mass. This descriptor is used for finding the main axes of symmetry or to determine that none of them exist in a 3D model. The descriptor is defined on the unit sphere and describes the global characteristics of a 3D shape. In (Podolak et al., 2006), Podolak et al. extend this work and introduce a Planar Reflective Symmetry Transform (PRST) that computes a measure of the reflective symmetry of a 3D shape with respect to all possible planes. This measure is used to define the center of symmetry and the principal symmetry axes of the global coordinate system. Rustamov improves this approach with the augmented symmetry transform in (Rustamov, 2007). Martinet et al. (Martinet et al., 2006) use generalized moments to detect perfect symmetries in 3D shapes. The authors perform an analysis of the extrema values, as well as the components of the spherical harmonics and compute the parameters of the symmetries that characterize a 3D model. The algorithm operates incrementally, thus enabling the determination of symmetries in larger models, based on existing symmetries of their parts. Mitra et al. (Mitra et al., 2006) compute partial and approximate symmetries in 3D models. The method is based on the matching of simple local characteristics, in pairs, and the use of these matchings for the augmentation of information about the existence of symmetries in the corresponding space transformations. A segmentation step extracts potential significant symmetries of the 3D model. Using both PCA-alignment and planar reflective symmetry, Chaouch and Verroust - Blondet (Chaouch and Verroust-Blondet, 2009a) compute a 3D model's principal axes and then, using a Local Translational Invariance Cost (LTIC), make a selection of the most suitable ones.

Using a rectilinearity measure, Lian et al. (Lian et al., 2010) compute a 3D model's best rotation by estimating the maximum ratio of its surface area to the sum of its three orthogonal projected areas. Similar to the previous approach, (Chaouch and Verroust-Blondet, 2009a), a selection between the rectilinearity measure and a PCA-based alignment is made. In (Axenopoulos et al., 2011) Axenopoulos et al. combine the properties of plane reflection symmetry and rectilinearity for achieving alignment. In this paper both CPCA and reflective symmetry are used, in order to achieve alignment. Rectilinearity is utilized to improve the alignment results.

Sfikas et al. (Sfikas et al., 2011a) propose a 3D model pose normalization method based on the similarity between a 3D model and its symmetric model across a plane of symmetry, thus determining the optimal plane of symmetry of the model. Initially, the axis-aligned minimum bounding box of a rigid 3D model is modified by requiring that the 3D model is also in minimum angular difference with respect to the normals to the faces of its bounding box. To estimate the modified axis-aligned bounding box, a set of predefined planes of symmetry are used and a combined spatial and angular distance, between the 3D model and its symmetric model, is calculated. By minimizing the combined distance, the 3D model is fitted inside its modified axis-aligned bounding box and alignment with the coordinate system is achieved.

In (Sfikas et al., 2013b, 2014) a pose normalization method, SYMPAN, based on reflective symmetry computed on PANORAMA-based views, is presented. Initially, through an iterative procedure, the symmetry principal plane of a 3D model is estimated, thus computing the first axis of the model. This is achieved by iteratively rotating the 3D model and computing reflective symmetry scores on panoramic view images. The other principal axes of the 3D model are estimated by computing the pixel variance on the 3D model's panoramic views.

The SYMPAN method has been incorporated in a hybrid scheme, that serves as the pose normalization procedure in a 3D object retrieval system. The effectiveness of this system, is evaluated in terms of retrieval accuracy and the results showed improved performance against previous approaches. This performance increase justifies the use of SYMPAN as the pose normalization method that complements the PANORAMA descriptor, due to its close integration with the PANORAMA representation (based on the same panoramic views).

3. Methodology

3.1. Background

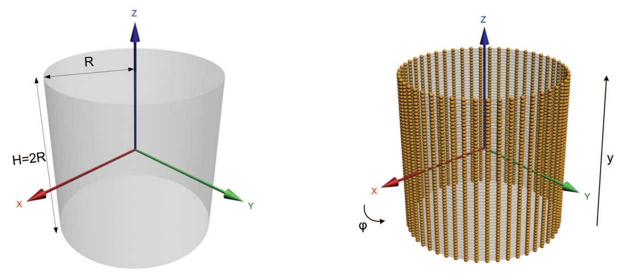
3.1.1. PANORAMA Representation Extraction

The *panoramic view* of a 3D model is obtained by projecting its surface onto the lateral surface of a cylinder of radius R and height $H = 2R$, centered at the origin, with its axis parallel to one of the principal axes of space (Papadakis et al., 2010), see Fig. 1a. The value of R is set to $2 * d_{max}$ where d_{max} is the maximum distance of the model's surface from its centroid.

Assuming the cylinder axis to be the z axis, the lateral surface of the cylinder is parameterized using a set of points $s(\phi, y)$ where $\phi \in [0, 2\pi]$ is the angle in the XY plane, $y \in [0, H]$ and the ϕ and y coordinates are sampled at rates $2B$ and B, respectively (B is set to be equal to 360). The ϕ dimension is sampled at twice the rate of the y dimension to account for the difference in length between the perimeter of the cylinder's lateral surface and its height. Although the perimeter of the specific cylinder's lateral surface is $2\pi \simeq 3$ times its height, the sampling rates are set at $2B$ and B, respectively, as these values were experimentally found to give good results. Thus, the set of points $s(\phi_u, y_v)$ are obtained, where $\phi_u = u * 2\pi/(2B)$, $y_v = v * H/B$, $u \in [0, 2B - 1]$ and $v \in [0, B - 1]$. These points are shown in Fig. 1b.

Next, the value at each point $s(\phi_u, y_v)$ of the panoramic view must be determined. The computation is carried out iteratively for $v = 0, 1, ..., B - 1$, each time considering the set of coplanar $s(\phi_u, y_v)$ points, i.e. a cross section v of the cylinder at height y_v and for each such cross section casting rays from its center c_v in the ϕ_u directions.

The cylindrical projections are used to capture two different characteristics of a 3D model's surface; (i) the position of the



(a) (b)

Fig. 1: (a) A projection cylinder for the acquisition of a 3D model's panoramic view and (b) the corresponding discretization of its lateral surface to the set of points $s(\phi_u, y_v)$

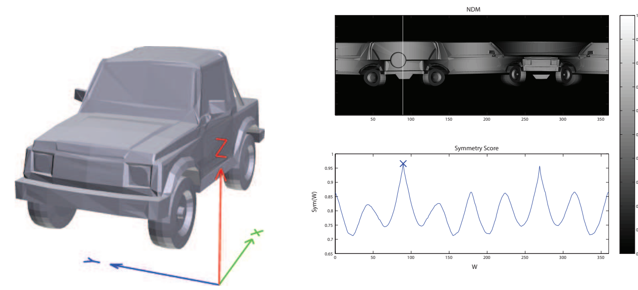


Fig. 2: Sample 3D model with the corresponding panoramic view and symmetry plane estimation, as these are employed in the SYMPAN pose normalization method.

model's surface in 3D space, (referred to as **Spatial Distribution Map** or **SDM**), and (ii) the orientation of the model's surface, (referred to as **Normals' Deviation Map** or **NDM**). To capture these characteristics two kinds of cylindrical projections $s_1(\phi_u, y_v)$ and $s_2(\phi_u, y_v)$ are used.

To capture the position of the model's surface, for each cross section at height y_v, the distances from c_v of the intersections of the model's surface are computed with the rays at each direction ϕ_u. Let $pos(\phi_u, y_v)$ denote the distance of the furthest from c_v point of intersection between the ray emanating from c_v in the ϕ_u direction and the model's surface; then $s_1(\phi_u, y_v) = pos(\phi_u, y_v)$. This value lies in the interval $[0, R]$, where R is the radius of the cylinder.

To capture the orientation of the model's surface, for each cross section at height y_v, the intersections of the model's surface with the rays at each direction ϕ_u are computed and the angle between a ray and the normal vector of the triangle that is intersected is measured. The value stored in $s_2(\phi_u, y_v)$ is a function of the cosine of the angle between the ray and the normal vector of the furthest from c_v intersected triangle of the model's surface. If $ang(\phi_u, y_v)$ denotes the aforementioned angle, then $s_2(\phi_u, y_v) = |\cos(ang(\phi_u, y_v))|^n$.

The nth power of $|\cos(ang(\phi_u, y_v))|$ is taken, where $n \geq 2$, since this setting enhances the contrast of the produced cylindrical projection. It has been experimentally found that setting n to a value in the range $[4, 6]$ gives the best results (Papadakis et al., 2010). Also, taking the absolute value of the cosine is necessary to deal with inconsistently oriented triangles along the model's surface due to e.g. concavities.

A cylindrical projection can be viewed as a 2D gray-scale image where pixels correspond to the (ϕ_u, y_v) values normalized to $[0, 1]$, in a manner reminiscent of cylindrical texture mapping.

3.1.2. SYMPAN: PANORAMA-based Pose Normalization

Pose normalization is performed using the SYMPAN method (Sfikas et al., 2014) which uses the **SDM** and the **NDM** extracted in PANORAMA. Pose normalization is significant in order to maintain integrity between the corresponding panoramic view representations of the 3D models. The choice of SYMPAN as the pose normalization method is due to its close integration with the PANORAMA representation and the fact that the majority of CAD 3D models and 3D models of non-artificial entities (e.g. furniture, vehicles, humans and animals, etc) actually exhibit reflective symmetry, to a certain degree. Methods that exploit symmetries have exhibited high performance, both in terms of pose normalization and retrieval accuracy, see (Sfikas et al., 2011b; Kazhdan et al., 2002b; Chaouch and Verroust-Blondet, 2009b).

Initially, a 3D model with arbitrary pose is normalized in terms of translation and scaling. Translation normalization is achieved though the extraction of the 3D model's centroid and the displacement of this centroid to the coordinate system origin. Consecutively, the 3D model is scaled so that it is inscribed within the unit sphere.

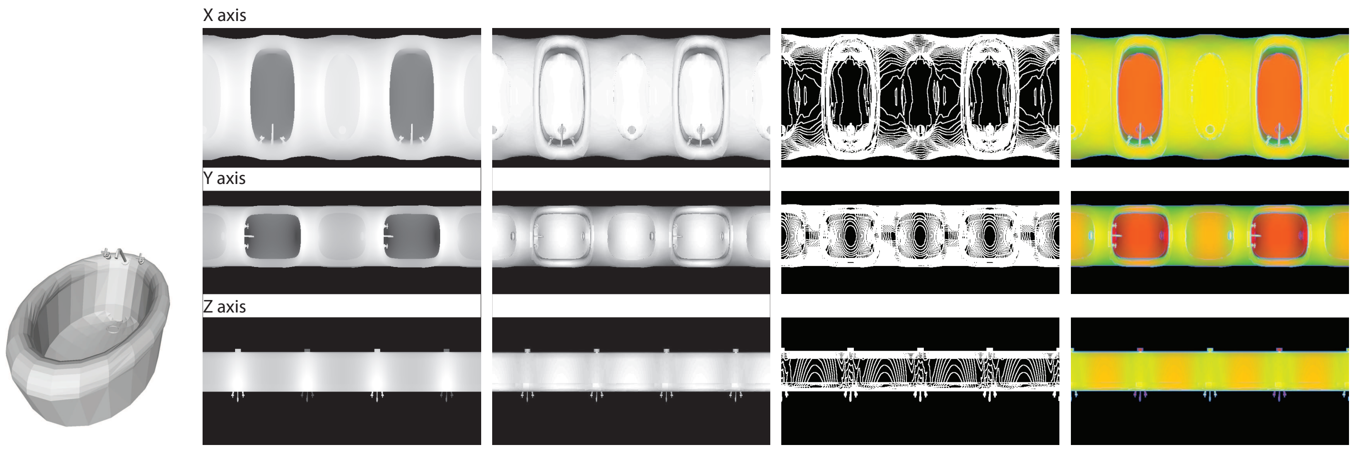
The estimation of a plane of symmetry of a 3D model corresponds to the detection of a line of reflective symmetry in its panoramic view. Since translation normalization has been performed, the plane of symmetry of the 3D object will pass through the origin of the coordinate system. The aim is to rotate the symmetry plane so that it includes the axis of the cylindrical projection (i.e. the z axis); then the plane of symmetry will be detectable in the panoramic image.

Once a plane of symmetry is defined, the first principal axis of the model is set to be the normal to that plane of symmetry (see Fig. 2). The remaining two principal axes have yet to be estimated. The 3D model can thus be rotated so that its symmetry plane coincides with one of the principal planes of space (e.g. the XY plane).

To complete the rotation normalization task, the 3D model is projected onto the surface of a projection cylinder whose axis is one of the principal axes of space, perpendicular to the symmetry plane's normal. The 3D model is iteratively rotated around the normal axis to the symmetry plane and the corresponding **SDM** images are calculated. For each **SDM** image, the variance of its pixel values is computed and the rotation that minimizes this variance, is defined as the rotation which aligns the principal axis of the 3D model with the axis of the projection cylinder.

3.2. Augmented Panoramic View Construction

In order to efficiently train an artificial neural network using the PANORAMA representation, an augmented schema is employed based on the panoramic views produced with respect to the three principal axes.



(a) 3D Model (b) SDM Image (c) NDM Image (d) Magnitude of Gradient Image (e) 3-channel Augmented Panoramic View Image

Fig. 3: Sample augmented panoramic view of a 3D model. (a) illustrates the original sample 3D model. (b) illustrates the SDM image for the three principal axes, (c) illustrates the NDM image for the three principal axes, (d) illustrates the magnitude of the gradient image computed from NDM, (e) illustrates the combined 3-channel image that is used as input to the convolutional neural network. The principal axes order is (from top to bottom): X axis, Y axis, Z axis.

More specifically, for each principal axis, the SDM (Fig. 3b) and NDM (Fig. 3c) cylindrical view representations are computed. On the NDM cylindrical view representation the magnitude of the gradient image is also computed, augmenting the initial PANORAMA representation (Fig. 3d). It should be noted that taking the magnitude of the gradient image on the SDM also increased performance, however the NDM gradient magnitude gave significantly better results.

Half of each panoramic view (in terms of width) is appended to its end. This, ensures a continuous representation with no 'wrap-around' gaps.

Thus, for each 3D model, the result is a total of three cylindrical view representations (corresponding to the three principal axes), each comprised of 3 separate channels (Fig. 3e). The three 3-channel representations are then stacked together in the following order: NDM(X) - SDM(X) - GradM(X), NDM(Y) - SDM(Y) - GradM(Y), NDM(Z) - SDM(Z) - GradM(Z) (see Fig. 3). This augmented representation defines the input of the convolutional neural network. The total size of a 3D model's augmented representation is 1.5 * 720 = 1080 pixels width by 360 * 3 = 1080 pixels height, for each channel.

Once the augmented representation has been montaged, its size is reduced to 10% of its original size, namely 3-channels $\times$ 108 $\times$ 108 pixels using bicubic interpolation. Although an amount of detail of the original representation is lost, it has been experimentally found that the minimized representation is sufficient to achieve high performance on the classification task while maintaining feasible neural network training times (see Section 4).

3.3. Convolutional Neural Network Architecture

The convolutional neural network architecture selected in the proposed implementation is based on a standard scheme, namely an input layer followed by a set of convolutional layers and finally by the fully connected layers of the output. The architecture proposed in (Krizhevsky et al., 2012) has been chosen, which has demonstrated state-of-the-art performance in image classification.

Three convolutional layers are used and the corresponding feature maps are 64, 256 and 1024 respectively. The kernel size is respectively set to 5, 5, 3 and the padding is set to 2 for all the layers. After each convolutional layer both a ReLU and a 2×2 max-pooling layer are inserted.

The output of the architecture consists of two fully connected layers, each composed of number of neurons equal to the number of image categories for the specific task. The two fully connected layers are followed by a dropout layer, (Srivastava et al., 2014), used to reduce overfitting. Finally, a softmax layer outputs class probabilities for a given input 3D model. The class with the highest probability is considered as the predicted class for the 3D model.

The network is trained using the stochastic gradient descent method (SGDM) with momentum set to 0.9.

3.4. Ensemble of CNNs

As many recent works have shown, the use of (convolutional) neural network ensembles provides a significant boost to the classification performance of a corresponding pipeline, (Russakovsky et al., 2015; Huang et al., 2016; Kumar et al., 2017). Hence, the PANORAMA-NN network presented in (Sfikas et al., 2017) is extended to an ensemble. The goal is to create a branched pipeline that divides according to the 3 axes of the panoramic views. To simplify the processing routine, each Augmented Panoramic View is divided into 3 regions (each consisting of 3-channels: one for SDM, one for NDM and one for the magnitude of gradient image of the NDM) along the vertical dimension and given as input to the corresponding pipeline path. Region #1 for projection axis X, region #2 for projection axis Y and region #3 for projection axis Z.

For the classification task the combined probability vector is assembled by taking the mean of all three individual probability

Fig. 4: Illustration of the proposed method pipeline, including the convolutional neural network architecture.

vectors.

Figure 4 illustrates the complete ensemble convolutional neural network pipeline, indicating how input data are divided to the pipeline paths and, correspondingly, how the probability vectors are combined for the final output.

Another way of dividing the input data would be according to the input image channels (NDM, SGM, gradient) and/or in combination with the aforementioned division according to the axes. Since the input to the convolutional neural network is 3-channel images, this can be considered to have been done implicitly by the input schema, since each channel is fed to different input neurons.

4. Experiments

4.1. Datasets

The datasets used for evaluating the proposed method are the Princeton ModelNet large scale 3D CAD model dataset, (Wu et al., 2015) and the ShapeNet Core55 subset of the ShapeNet dataset, (Chang et al., 2015).

ModelNet is comprised of 127,915 CAD models split into 662 object categories and is split into two subsets, ModelNet-10 and ModelNet-40, both of which contain training and testing partitions. ModelNet-10 comprises 4,899 CAD models split into 10 categories. The models have been manually cleaned and pose normalized in terms of translation and rotation. The training and testing subsets of ModelNet-10 consist of 3,991 and 908 models respectively. ModelNet-40 comprises 12,311 CAD models split into 40 categories. The models have been manually cleaned but are not pose normalized. The training and testing subsets of ModelNet-40 consist of 9,843 and 2,468 models respectively.

ShapeNetCore is comprised of approximately 51,300 3D models made up of 55 common categories. Each category is divided into several subcategories. ShapeNetCore offers two dataset versions: (a) consistently aligned 3D models and (b) models that are perturbed by random rotations. From the complete dataset are created three splits of 70%, 10% and 20% for training, validation and testing respectively.

4.2. 3D Model Classification

The proposed method, **PANORAMA-ENN**, is evaluated on the task of classification of the test subset of both ModelNet-10 and ModelNet-40. The performance is measured via the average binary categorical accuracy (a value of 1 corresponds to the case where the category of the test 3D model is correctly predicted, otherwise 0).

Participating in the comparison are the original Light Field (Chen et al., 2003a) (LFD, 4,700 dimensions) and Spherical Harmonics (Kazhdan et al., 2003) (SPH, 544 dimensions) descriptors that do not use machine learning in order to set a baseline for the evaluation. Also included are recent methods that use machine learning: PANORAMA-NN (Sfikas et al., 2017), 3D ShapeNets (V) (Wu et al., 2015), the DeepPano descriptor (Shi et al., 2015), Multi-view Convolutional Neural Networks (V) (Su et al., 2015) (MVCNN) and the Geometry Image descriptor (Sinha et al., 2016). In addition to the above competing methods that were also reported in (Sfikas et al., 2017), the results are extended to include the following techniques: GIFT (Bai et al., 2016), ORION (V) (Sedaghat et al., 2016), Set-convolution (Ravanbakhsh et al., 2016), 3D-GAN (V) (Wu et al., 2016), VRN Ensemble (V) (Brock et al., 2016), FusionNet (V) (Hegde and Zadeh, 2016), VoxNet (V) (Maturana and Scherer, 2015), the PointNet method by (Garcia-Garcia et al., 2016) (PointNet-Garcia), the PointNet method by (Qi et al., 2016b) (PointNet-Qi), MVCNN-MultiRes (V) (Qi et al., 2016a), FPNN (V) (Li et al., 2016), the method by Klokov and Lempitsky (Klokov and Lempitsky, 2017) and the method

Method	ModelNet-10	ModelNet-40
PANORAMA-ENN	0.9685	**0.9556**
PANORAMA-NN	*0.9112*	*0.9070*
PANORAMA-NN + GradM	0.9345	0.9201
VRN Ensemble (V)*	**0.9714**	0.9554
Klokov & Lempitsky*	0.9400	0.9180
MVCNN-MultiRes (V)*	N/A	0.9140
Fusion-Net (V)*	0.9311	0.9080
MVCNN (V)	N/A	0.9010
Set-Convolution*	N/A	0.9000
PointNet-Qi	N/A	0.8920
FPNN (V)*	N/A	0.8840
Geometry Image	0.8840	0.8390
3D-GAN (V)*	0.9100	0.8330
GIFT	0.9235	0.8310
VoxNet(V)	0.9200	0.8300
DeepPano	0.8866	0.8254
Xu & Todorovic (V)*	0.8800	0.8126
3D ShapeNets (V)	0.8354	0.7732
ORION (V)*	0.9380	N/A
PointNet-Garcia	0.7760	N/A
LFD (NON-ML)	0.7987	0.7547
SPH (NON-ML)	0.7979	0.6823

Table 3: Classification accuracies on the ModelNet-10 and ModelNet-40 datasets. Methods indicated by an (*) are arXiv versions, at the time of writing. Methods that employ voxel representations are indicated by (V) while those that do not involve machine learning are indicated by (NON-ML).

by Xu and Todorovic (V) (Xu and Todorovic, 2016). The scores of the aforementioned competing methods are those reported by the authors in the respective papers. Table 3 summarizes the scores of the above methods.

The proposed method outperforms all aforementioned methods in the challenging ModelNet-40 dataset while in ModelNet-10 it is only surpassed by the VRN Ensemble method (Brock et al., 2016) by a small margin. It is evident that methods employing voxel representations generally perform better than methods using image representations. This can be justified by the richer information contained in 3D volumetric data with respect to the 2D representations. However, the proposed method is able to successfully outperform previous methods despite the use of an image representation.

In order to compare all key aspects of the extended approach to the original PANORAMA-NN (Sfikas et al., 2017), a version of the PANORAMA-NN that in addition to the SDM and NDM representation views, also includes the magnitude of the gradient image (but does not use the ensemble architecture) has also been added to Table 3 (referred as PANORAMA-NN + GradM). In this version the data for all three projection axes are fed to the same network. It appears that the addition of the new image and the ensemble architecture contributed to the gain in performance in similar portions.

4.3. 3D Model Retrieval

Another evaluation of the proposed method was performed on the task of 3D model retrieval.

Method	ModelNet-10	ModelNet-40
PANORAMA-ENN	**0.9328**	**0.8634**
PANORAMA-NN	*0.8739*	*0.8345*
GIFT	0.9112	0.8194
DeepPano	0.8418	0.7681
Geometry Image	0.7490	0.5130
3D ShapeNets	0.6826	0.4923
MVCNN	N/A	0.7950
PANORAMA (NON-ML)	0.6032	0.4613
LFD (NON-ML)	0.4982	0.4091
SPH (NON-ML)	0.4405	0.3326

Table 4: Retrieval accuracies measured in mean Average Precision (mAP) on the ModelNet-10 and ModelNet-40 datasets.

The performance of the proposed method was measured on the ModelNet-10 and ModelNet-40 datasets, against the methods that offer retrieval results i.e., (Sfikas et al., 2017) and the GIFT method (Bai et al., 2016). On the ShapeNetCore dataset, the proposed method was compared against a number of methods that competed on the SHREC2016 and SHREC2017 *Large-scale 3D Shape Retrieval from ShapeNet Core55* tracks (Savva et al., 2016, 2017). More specifically, RotationNet (Kanezaki, 2016), GIFT (Bai et al., 2016), ReVGG (Savva et al., 2017) and DLAN (Furuya and Ohbuchi, 2016). Also the SHREC2016 versions of the GIFT (Bai et al., 2016) and MVCNN (Su et al., 2015) are included. Finally, we include the performance of the original (non-ML) PANORAMA descriptor (Papadakis et al., 2010) The scores of the aforementioned competing methods are those reported by the authors in the respective papers.

On the ModelNet datasets, retrieval accuracy is measured via the *mean Average Precision* (mAP) metric and the Precision-Recall plots. On the ShapeNetCore dataset, retrieval accuracy is measured via the mAP metric, as well as the F-score and the Normalized Discounted Cumulative Gain (NDCG) metrics, to be directly comparable with the SHREC *Large-scale 3D Shape Retrieval from ShapeNet Core55* track results.

The descriptor for the retrieval task is composed of the activations of the last fully connected layer of the convolutional neural network. Each 3D model descriptor is compared against the rest of the 3D model descriptors using the L_1 distance metric. L_1 distance is used due to its linearity, which emphasizes the difference between components of the descriptor vectors.

For the ModelNet datasets, Table 4 and Fig. 5 show the results of the retrieval experiment where the proposed method outperforms the competition in both datasets.

Fig. 6 illustrates the confusion matrix for the 3D models of the ModelNet-10 dataset. Lower values indicate higher similarity between corresponding models. It is evident that higher similarity is exhibited between 3D models that belong to the same class than 3D models of different classes. Furthermore, it can be seen that 3D models of different classes that, however, have similar structure (i.e., `night_stand` and `dresser`, or `table` and `desk`) show higher similarity than classes of different structure (i.e., `table` and `bathtub`). The proposed method, able to distinguish between different classes, is also capable of determining if two 3D models have similar structure in an efficient

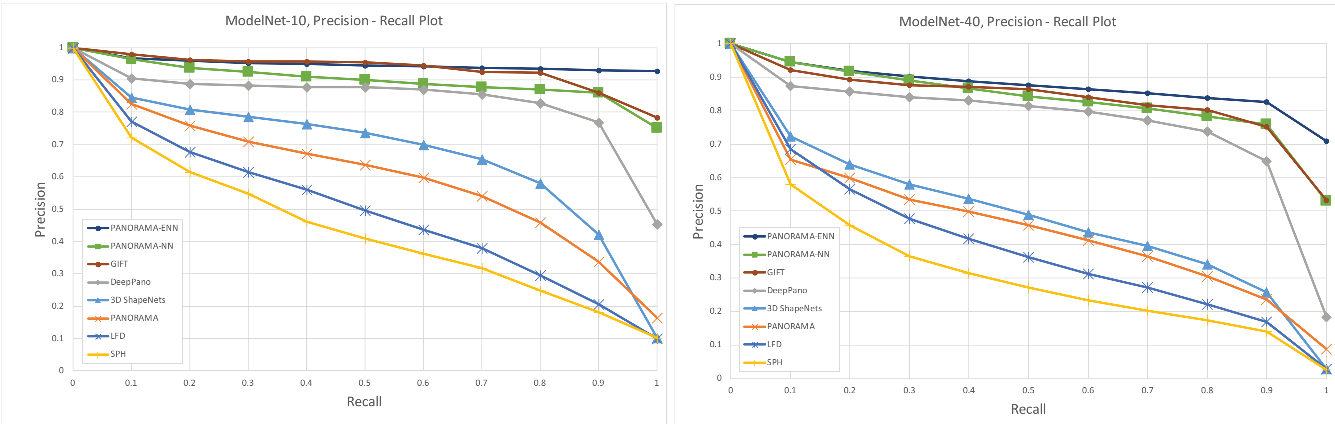


Fig. 5: Precision-Recall plots for ModelNet-10 (left) and ModelNet-40 (right) datasets. Illustrated are the proposed method (PANORAMA-ENN) compared to the previous version of the method (PANORAMA-NN) and six other retrieval methods.

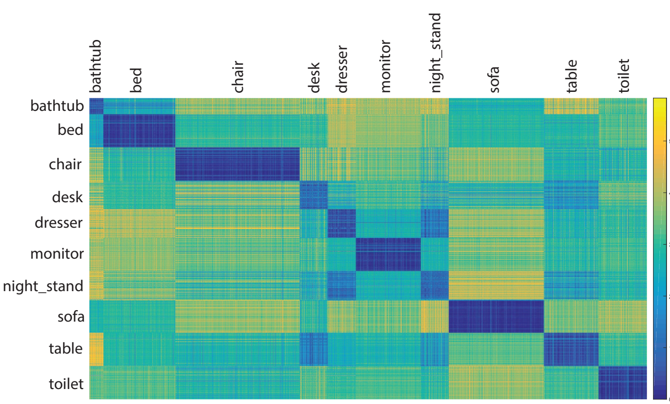


Fig. 6: Confusion matrix for the 3D models of the ModelNet-10 dataset classes. Values indicate similarity between 3D models; see colour map on the right.

manner.

Fig. 7 illustrates qualitative retrieval results for 10 sample query models. The first column indicates the query and the remaining columns (left-to-right in retrieval order) indicate the top 10 retrieved 3D models from the ModelNet-10 dataset. Note that the first retrieved 3D model is the query model itself while all the retrieved 3D models belong to the same class as the query.

Table 5 and Table 6 show the results of the retrieval experiment on the ShapeNetCore dataset, respectively for the pose normalized and perturbed versions. The methods compared are those that exhibited higher performance in terms of retrieval accuracy, in the SHREC2016 and SHREC2017 *Large-scale 3D Shape Retrieval from ShapeNet Core55* tracks. Also, for reference purposes, the PANORAMA-NN method, (Sfikas et al., 2017), is also included.

Macro-averaged versions of the metrics are used to give an unweighted average over the entire dataset. The retrieval scores for all the models are averaged with equal weights. In the micro-averaged versions, each query and retrieval results are treated equally across classes, and therefore the results are averaged without reweighting based on category size. This gives a representative performance metric average across categories, see (Savva et al., 2017). The micro- and macro- averaged versions of the metrics have been computed using the evaluation code of the SHREC2017 track.

On the normalized 3D models dataset, the proposed method outperforms the other methods on the F-score and mAP metric on the Macro-averaged version, while being surpassed only by a small margin on the NDCG metric. On the Micro-averaged version the proposed method can be placed among the best methods, based on the aforementioned metrics, surpassed by a small margin. On the perturbed 3D models dataset, the proposed method outperforms the other methods on the F-score of the Macro-averaged version and on the mAP metric of the Micro-averaged version. It is always very close to the best results on this dataset.

4.4. Failure Cases

Fig. 8 qualitatively illustrates four of the worst retrieval failure cases. The first column indicates the query and the remaining columns (left-to-right in retrieval order) indicate the top 4 retrieved 3D models from the ModelNet-10 dataset. As can be seen, although the retrieved models do not belong to the same class as the query model, their structure is highly similar. In the second row the query is from the `desk` class and the results from the `table` class, while in the fourth row, the query originates from the `dresser` class and the results from the `night_stand` class. These classes contain models whose structure is very similar, but are separate classes mainly due to utilitarian reasons and are hard to distinguish purely on geometric grounds.

One insight that can be gained from the failure cases is that when the objects exhibit similarities or patterns along one or more of their principal axes they are less distinguishable by the proposed method.

4.5. Implementation

The proposed method was tested on an Intel (R) Core (TM) i7 @ 3.60GHz CPU system, with 32GB of RAM and a discrete NVIDIA (R) TITAN X GPU with 12GB RAM. The system run Matlab R2016b. The PANORAMA representation extraction method was developed in a hybrid Matlab/C++/OpenGL architecture while the pose normalization procedure was developed

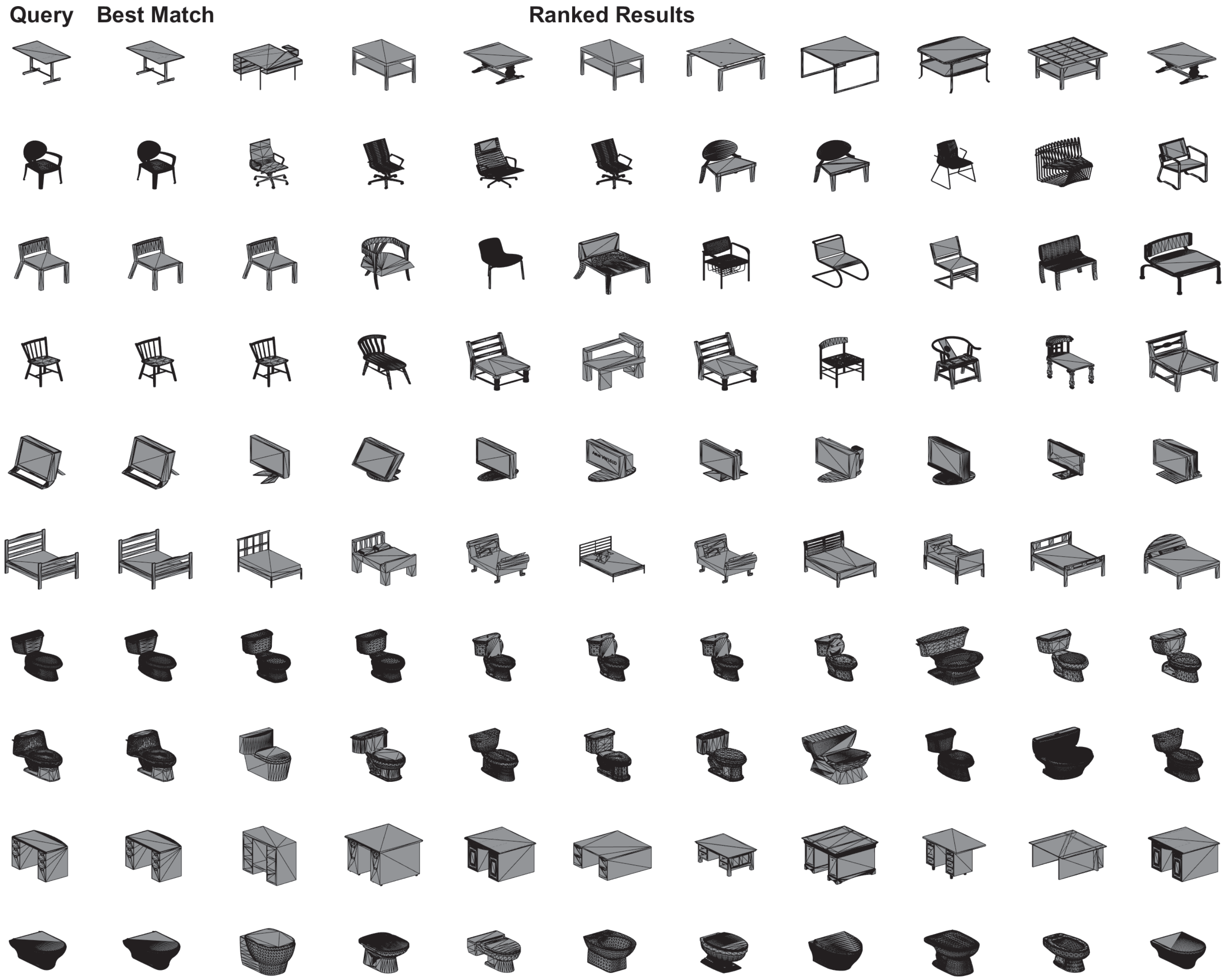


Fig. 7: Retrieval examples for the proposed method on the ModelNet-10 dataset. First column illustrates the queries while the remaining columns illustrate the corresponding retrieved models in rank order. Note that the first retrieved model is the query model in all cases.

in Matlab. The artificial neural network was implemented using the Matlab Deep Neural Network toolbox and accelerated via the CUDA instruction set on the GPU.

The approximate PANORAMA representation extraction for a 10,000 face 3D model is 350 ms. The approximate pose normalization time for the same typical model is 1,850ms. The artificial neural network training procedure requires approximately 16 minutes to converge. When image representations of higher resolution were used (reduction to 20% of the original size, i.e. 216×216 pixels, instead of reduction to 10%) the performance gain was considered insignificant (approximately +0.005%) while the training process doubled in time (to approximately 30 minutes).

Note that although the architecture of the proposed method has been extended to an ensemble of convolutional neural networks, the three pipeline paths can easily be parallelized with minimum overhead for data division (one I/O read for all pipeline paths for each 3D model, as this is performed in memory) and results combination (one simple addition of memory values).

5. Conclusions and Future Work

A novel convolutional neural network based method for the creation of 3D model descriptors has been proposed. A complete pipeline is given, defining the input representation as well as the parameters and structure of the CNN employed. Initially, the 3D models of the dataset are pose normalized using the SYMPAN algorithm. This is a crucial step since not all dataset 3D models are guaranteed to be pose normalized (e.g. the ModelNet-40 and ShapeNetCore perturbed datasets are not pose normalized). Next, for each 3D model, an augmented panoramic representation is extracted consisting of 9 parts (3 for the major axes times 3 for the data contents). This representation is then resized to 10% if its original size and used as input to a convolutional neural network ensemble; the ensemble divides the input into 3 parts based on the major axes.

	Micro-averaged			Macro-averaged		
Method	**F-score**	**mAP**	**NDCG**	**F-score**	**mAP**	**NDCG**
PANORAMA-ENN	0.789	0.739	0.845	**0.591**	**0.588**	0.656
PANORAMA-NN	*0.776*	*0.723*	*0.815*	*0.580*	*0.557*	*0.630*
RotationNet*	**0.798**	0.722	**0.865**	0.590	0.583	0.656
ReVGG	0.772	**0.749**	0.828	0.519	0.496	0.559
GIFT	0.767	0.722	0.827	0.581	0.575	**0.657**
DLAN	0.712	0.663	0.762	0.505	0.477	0.563
SHREC 2016 GIFT	0.689	0.640	0.765	0.454	0.447	0.548
SHREC 2016 MVCNN*	0.764	0.735	0.815	0.575	0.566	0.640

Table 5: Retrieval accuracies measured by F-score, mean Average Precision (mAP) and Normalized Discounted Cumulative Gain (NDCG) on the normalized 3D models ShapeNetCore dataset. Methods indicated by an (*) are arXiv versions, at the time of writing

	Micro-averaged			Macro-averaged		
Method	**F-score**	**mAP**	**NDCG**	**F-score**	**mAP**	**NDCG**
PANORAMA-ENN	0.715	**0.703**	0.759	**0.510**	0.462	0.554
PANORAMA-NN	*0.701*	*0.687*	*0.720*	*0.476*	*0.447*	*0.522*
RotationNet*	0.636	0.606	0.702	0.333	0.327	0.407
ReVGG	**0.719**	0.696	**0.783**	0.434	0.418	0.479
GIFT	0.643	0.567	0.701	0.437	0.406	0.513
DLAN	0.706	0.656	0.754	0.503	**0.476**	**0.560**
SHREC 2016 GIFT	0.661	0.607	0.735	0.423	0.412	0.518
SHREC 2016 MVCNN*	0.612	0.535	0.653	0.416	0.367	0.459

Table 6: Retrieval accuracies measured by F-score, mean Average Precision (mAP) and Normalized Discounted Cumulative Gain (NDCG) on the perturbed 3D models ShapeNetCore dataset. Methods indicated by an (*) are arXiv versions, at the time of writing

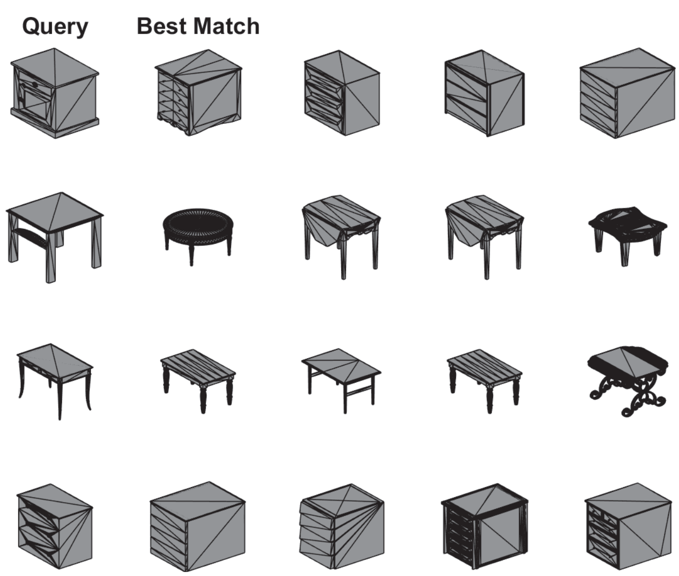


Fig. 8: Sample retrieval failure cases for the proposed method. First column illustrates the queries while the remaining columns illustrate the corresponding retrieved models in rank order.

PANORAMA, in addition to being a good shape descriptor, bridges the gap between the initial 3D model representation and the 2D input that is typically more suitable for convolutional neural networks. The SYMPAN pose normalization method works with reflective symmetries and this could partially explain the high accuracy achieved on both the ModelNet and ShapeNetCore datasets. These datasets consist of both CAD and 'real-life' entities that contain several such symmetries. The ModelNet-10 and ModelNet-40 as well as the ShapeNetCore datasets used for evaluation were specifically designed for deep neural network classification applications.

The descriptors created by the proposed method were compared against a number of published works on the tasks of 3D model classification and retrieval and achieve performance above or comparable to the state-of-the-art. The superiority of the proposed method compared to the competitive ones is that its data representation preserves feature continuity of the 3D models, whereas other image representation techniques (i.e. planar projections) do not.

Future work should include the exploration of additional channels of information regarding the 2D image representation. The 3-channel scheme could be extended, e.g, by surface color information. Unfortunately, none of the datasets that we experimented with possessed such information and the training of deep networks is dependent on the existence of suitable large training datasets.

References

Papadakis, P, Pratikakis, I, Theoharis, T, Perantonis, S. PANORAMA: a 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. International Journal of Computer Vision 2010;89(2-3):177–192.

Sfikas, K, Theoharis, T, Pratikakis, I. Pose normalization of 3D models via reflective symmetry on panoramic views. The Visual Computer 2014;30(11):1261–1274.

Sfikas, K, Theoharis, T, Pratikakis, I. 3D object retrieval via range image queries in a bag-of-visual-words context. The Visual Computer 2013a;29(12):1351–1361.

Sfikas, K, Pratikakis, I, Koutsoudis, A, Savelonas, M, Theoharis, T. Partial matching of 3D cultural heritage objects using panoramic views. Multimedia Tools and Applications 2016;75(7):3693–3707.

Sfikas, K, Theoharis, T, Pratikakis, I. Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval. In: Pratikakis, I, Dupont, F, Ovsjanikov, M, editors. Eurographics Workshop on 3D Object Retrieval. The Eurographics Association. ISBN 978-3-03868-030-7; 2017,doi:10.2312/3dor.20171045.

Savva, M, Yu, F, Su, H, Aono, M, Chen, B, Cohen-Or, D, et al. SHREC16 Track Large-Scale 3D Shape Retrieval from ShapeNet Core55. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval. 2016,.

Savva, M, Yu, F, Su, H, Kanezaki, A, Furuya, T, Ohbuchi, R, et al. SHREC17 Track Large-Scale 3D Shape Retrieval from ShapeNet Core55. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval. 2017,.

Wu, Z, Song, S, Khosla, A, Yu, F, Zhang, L, Tang, X, et al. 3D ShapeNets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, p. 1912–1920.

Chang, AX, Funkhouser, T, Guibas, L, Hanrahan, P, Huang, Q, Li, Z, et al. ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR]; Stanford University — Princeton University — Toyota Technological Institute at Chicago; 2015.

Su, H, Maji, S, Kalogerakis, E, Learned-Miller, E. Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 945–953.

Kanezaki, A. RotationNet: Learning Object Classification Using Unsupervised Viewpoint Estimation. CoRR 2016;abs/1603.06208. URL: http://arxiv.org/abs/1603.06208.

Shi, B, Bai, S, Zhou, Z, Bai, X. Deeppano: Deep panoramic representation for 3D shape recognition. IEEE Signal Processing Letters 2015;22(12):2339–2343.

Bai, S, Bai, X, Zhou, Z, Zhang, Z, Jan Latecki, L. Gift: A real-time and scalable 3D shape search engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 5023–5032.

Chen, DY, Tian, XP, Shen, YT, Ouhyoung, M. On visual similarity based 3D model retrieval. In: Computer graphics forum; vol. 22. Wiley Online Library; 2003a, p. 223–232.

Qi, CR, Su, H, Nießner, M, Dai, A, Yan, M, Guibas, LJ. Volumetric and Multi-View CNNs for Object Classification on 3D Data. CoRR 2016a;abs/1604.03265. URL: http://arxiv.org/abs/1604.03265.

Sedaghat, N, Zolfaghari, M, Brox, T. Orientation-boosted Voxel Nets for 3D Object Recognition. CoRR 2016;abs/1604.03351. URL: http://arxiv.org/abs/1604.03351.

Hegde, V, Zadeh, R. FusionNet: 3D Object Classification Using Multiple Data Representations. CoRR 2016;abs/1607.05695. URL: http://arxiv.org/abs/1607.05695.

Furuya, T, Ohbuchi, R. Deep Aggregation of Local 3D Geometric Features for 3D Model Retrieval. In: Wilson, RC, Hancock, ER, Smith, WAP, editors. Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016. BMVA Press; 2016,URL: http://www.bmva.org/bmvc/2016/papers/paper121/index.html.

Sinha, A, Bai, J, Ramani, K. Deep learning 3D shape surfaces using geometry images. In: European Conference on Computer Vision. Springer; 2016, p. 223–240.

Kazhdan, M, Funkhouser, T, Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Symposium on geometry processing; vol. 6. 2003, p. 156–164.

Ravanbakhsh, S, Schneider, J, Poczos, B. Deep Learning with Sets and Point Clouds. ArXiv e-prints 2016;arXiv:1611.04500.

Wu, J, Zhang, C, Xue, T, Freeman, WT, Tenenbaum, JB. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. CoRR 2016;abs/1610.07584. URL: http://arxiv.org/abs/1610.07584.

Brock, A, Lim, T, Ritchie, JM, Weston, N. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks. CoRR 2016;abs/1608.04236. URL: http://arxiv.org/abs/1608.04236.

Maturana, D, Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2015, p. 922–928. doi:10.1109/IROS.2015.7353481.

Garcia-Garcia, A, Gomez-Donoso, F, Garcia-Rodriguez, J, Orts-Escolano, S, Cazorla, M, Azorin-Lopez, J. PointNet: A 3D Convolutional Neural Network for real-time object class recognition. In: 2016 International Joint Conference on Neural Networks (IJCNN). 2016, p. 1578–1584. doi:10.1109/IJCNN.2016.7727386.

Li, Y, Pirk, S, Su, H, Qi, CR, Guibas, LJ. FPNN: Field Probing Neural Networks for 3D Data. CoRR 2016;abs/1605.06240. URL: http://arxiv.org/abs/1605.06240.

Klokov, R, Lempitsky, V. Escape from Cells: Deep Kd-Networks for The Recognition of 3D Point Cloud Models. ArXiv e-prints 2017;arXiv:1704.01222.

Xu, X, Todorovic, S. Beam Search for Learning a Deep Convolutional Neural Network of 3D Shapes. CoRR 2016;abs/1612.04774. URL: http://arxiv.org/abs/1612.04774.

Bai, X, Bai, S, Wang, X. Beyond diffusion process: Neighbor set similarity for fast re-ranking. Information Sciences 2015;325:342 – 354. URL: http://www.sciencedirect.com/science/article/pii/S0020025515005150. doi:https://doi.org/10.1016/j.ins.2015.07.022.

Tatsuma, A, Aono, M. Multi-Fourier Spectra Descriptor and Augmentation with Spectral Clustering for 3D Shape Retrieval. Vis Comput 2009;25(8):785–804. URL: http://dx.doi.org/10.1007/s00371-008-0304-2. doi:10.1007/s00371-008-0304-2.

Ohbuchi, R, Osada, K, Furuya, T, Banno, T. Salient local visual features for shape-based 3D model retrieval. In: 2008 IEEE International Conference on Shape Modeling and Applications. 2008, p. 93–102. doi:10.1109/SMI.2008.4547955.

Lian, Z, Godil, A, Sun, X, Xiao, J. CM-BOF: Visual Similarity-based 3D Shape Retrieval Using Clock Matching and Bag-of-Features. Mach Vision Appl 2013;24(8):1685–1704. URL: http://dx.doi.org/10.1007/s00138-013-0501-5. doi:10.1007/s00138-013-0501-5.

Paquet, E, Rioux, M, Murching, AM, Naveen, T, Tabatabai, AJ. Description of shape information for 2D and 3D objects. Sig Proc: Image Comm 2000;16(1-2):103–122. doi:10.1016/S0923-5965(00)00020-5.

Shilane, P, Min, P, Kazhdan, MM, Funkhouser, TA. The princeton shape benchmark. In: SMI. IEEE Computer Society. ISBN 0-7695-2075-8; 2004, p. 167–178. doi:10.1109/SMI.2004.63.

Theodoridis, S, Koutroumbas, K. Pattern recognition. Academic Press; 1999. ISBN 978-0-12-686140-2.

Vranić, DV, Saupe, D, Richter, J. Tools for 3D-object retrieval: Karhunen-loeve transform and spherical harmonics. In: IEEE MMSP 2001. 2001, p. 293–298.

Zaharia, TB, Prêteux, FJ. 3D versus 2D/3D shape descriptors: a comparative study. In: Dougherty, ER, Astola, J, Egiazarian, KO, editors. Image Processing: Algorithms and Systems; vol. 5289 of *SPIE Proceedings*. SPIE; 2004, p. 47–58. doi:10.1117/12.533092.

Chen, DY, Tian, XP, Shen, YT, Ouhyoung, M. On visual similarity based 3D model retrieval. Comput Graph Forum 2003b;22(3):223–232. doi:10.1111/1467-8659.00669.

Vranic, DV. 3D model retrieval. Ph.D. thesis; 2004.

Vranic, DV. DESIRE: a composite 3D-shape descriptor. In: ICME. IEEE; 2005, p. 962–965. doi:10.1109/ICME.2005.1521584.

Papadakis, P, Pratikakis, I, Perantonis, SJ, Theoharis, T. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. Pattern Recognition 2007;40(9):2437–2452. doi:10.1016/j.patcog.2006.12.026.

Papadakis, P, Pratikakis, I, Theoharis, T, Passalis, G, Perantonis, SJ. 3D object retrieval using an efficient and compact hybrid shape descriptor. In: Perantonis, SJ, Sapidis, NS, Spagnuolo, M, Thalmann, D, editors. 3DOR. Eurographics Association. ISBN 978-3-905674-05-7; 2008, p. 9–16. doi:10.2312/3DOR/3DOR08/009-016.

Kazhdan, MM, Chazelle, B, Dobkin, DP, Finkelstein, A, Funkhouser, TA. A reflective symmetry descriptor. In: Heyden, A, Sparr, G, Nielsen, M, Johansen, P, editors. ECCV (2); vol. 2351 of *Lecture Notes in Computer Science*. Springer. ISBN 3-540-43744-4; 2002a, p. 642–656. doi:10.1007/3-540-47967-8_43.

Podolak, J, Shilane, P, Golovinskiy, A, Rusinkiewicz, S, Funkhouser, TA. A planar-reflective symmetry transform for 3D shapes. ACM Trans Graph 2006;25(3):549–559. doi:10.1145/1141911.1141923.

Rustamov, RM. Augmented symmetry transforms. In: Shape Modeling International. IEEE Computer Society; 2007, p. 13–20. doi:10.1109/SMI.2007.6.

Martinet, A, Soler, C, Holzschuch, N, Sillion, FX. Accurate detection of symmetries in 3D shapes. ACM Trans Graph 2006;25(2):439–464. doi:10.

1145/1138450.1138462.
Mitra, NJ, Guibas, LJ, Pauly, M. Partial and approximate symmetry detection for 3D geometry. ACM Trans Graph 2006;25(3):560–568. doi:10.1145/1141911.1141924.
Chaouch, M, Verroust-Blondet, A. Alignment of 3D models. Graphical Models 2009a;71(2):63–76. doi:10.1016/j.gmod.2008.12.006.
Lian, Z, Rosin, PL, Sun, X. Rectilinearity of 3D meshes. International Journal of Computer Vision 2010;89(2-3):130–151. doi:10.1007/s11263-009-0295-0.
Axenopoulos, A, Litos, G, Daras, P. 3D model retrieval using accurate pose estimation and view-based similarity. In: Natale, FGBD, Bimbo, AD, Hanjalic, A, Manjunath, BS, Satoh, S, editors. ICMR. ACM. ISBN 978-1-4503-0336-1; 2011, p. 41. doi:10.1145/1991996.1992037.
Sfikas, K, Theoharis, T, Pratikakis, I. ROSy+: 3D object pose normalization based on PCA andreflective object symmetry with application in 3D object retrieval. International Journal of Computer Vision 2011a;91(3):262–279. doi:10.1007/s11263-010-0395-x.
Sfikas, K, Pratikakis, I, Theoharis, T. SymPan: 3D Model Pose Normalization via Panoramic Views and Reflective Symmetry. In: Castellani, U, Schreck, T, Biasotti, S, Pratikakis, I, Godil, A, Veltkamp, RC, editors. 3DOR. Eurographics Association. ISBN 978-3-905674-44-6; 2013b, p. 41–48.
Sfikas, K, Theoharis, T, Pratikakis, I. ROSy+: 3D Object Pose Normalization Based on PCA and Reflective Object Symmetry with Application in 3D Object Retrieval. Int J Comput Vision 2011b;91(3):262–279. URL: http://dx.doi.org/10.1007/s11263-010-0395-x. doi:10.1007/s11263-010-0395-x.
Kazhdan, M, Chazelle, B, Dobkin, D, Finkelstein, A, Funkhouser, T. A reflective symmetry descriptor. In: European Conference on Computer Vision. Springer; 2002b, p. 642–656.
Chaouch, M, Verroust-Blondet, A. Alignment of 3D models. Graphical Models 2009b;71(2):63–76.
Krizhevsky, A, Sutskever, I, Hinton, GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 2012, p. 1097–1105.
Srivastava, N, Hinton, GE, Krizhevsky, A, Sutskever, I, Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 2014;15(1):1929–1958.
Russakovsky, O, Deng, J, Su, H, Krause, J, Satheesh, S, Ma, S, et al. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 2015;115(3):211–252. doi:10.1007/s11263-015-0816-y.
Huang, HK, Chiu, CF, Kuo, CH, Wu, YC, Chu, NNY, Chang, PC. Mixture of deep CNN-based ensemble model for image retrieval. In: 2016 IEEE 5th Global Conference on Consumer Electronics. 2016, p. 1–2. doi:10.1109/GCCE.2016.7800375.
Kumar, A, Kim, J, Lyndon, D, Fulham, M, Feng, D. An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification. IEEE Journal of Biomedical and Health Informatics 2017;21(1):31–40. doi:10.1109/JBHI.2016.2635663.
Qi, CR, Su, H, Mo, K, Guibas, LJ. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CoRR 2016b;abs/1612.00593. URL: http://arxiv.org/abs/1612.00593.