M.S. THESIS

SPNet: Deep 3D Object Classification and Retrieval using Stereographic Projection

SPNet : 입체화법 투사을 이용한 3D 객체 분류 및 검색

BY

Mohsen Yavartanoo

AUGUST 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

SPNet: Deep 3D Object Classification and Retrieval using Stereographic Projection

SPNet : 입체화법 투사을 이용한 3D 객체 분류 및 검색

BY

Mohsen Yavartanoo

AUGUST 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

SPNet: Deep 3D Object Classification and Retrieval using Stereographic Projection

SPNet : 입체화법 투사을 이용한 3D 객체 분류 및 검색

지도교수 이 경 무
이 논문을 공학석사 학위논문으로 제출함

2019년 08월

서울대학교 대학원

전기 컴퓨터 공학부

야바타누모흐센

야바타누 모흐센의 공학석사 학위 논문을 인준함

2019년 08월

위 원 장: _____
부위원장: _____
위 원: _____

Abstract

We propose an efficient Stereographic Projection Neural Network (SPNet) for learning representations of 3D objects. We first transform a 3D input volume into a 2D planar image using stereographic projection. We then present a shallow 2D convolutional neural network (CNN) to estimate the object category followed by view ensemble, which combines the responses from multiple views of the object to further enhance the predictions. Specifically, the proposed approach consists of four stages: (1) Stereographic projection of a 3D object, (2) view-specific feature learning, (3) view selection and (4) view ensemble. The proposed approach performs comparably to the state-of-the-art methods while having substantially lower GPU memory as well as network parameters. Despite its lightness, the experiments on 3D object classification and shape retrievals demonstrate the high performance of the proposed method.

keywords: 3D object classification, 3D object retrieval, Stereographic Projection, Convolutional Neural Network, View Ensemble, View Selection.

student number: 2017-27494

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

In recent years, success of deep learning methods, in particular, convolutional neural network (CNN), has urged rapid development in various computer vision applications such as image classification, object detection, and super-resolution. Along with the drastic advances in 2D computer vision, understanding 3D shapes and environment have also attracted great attention.

Many traditional CNNs on 3D data simply extend the 2D convolutional operations to 3D, for example, the work of Wu et al. [36] which extends 2D deep belief network to 3D deep belief network, or the works of Maturana et al. [18] and Sedaghat et al. [25] where they extend 2D convolutional kernels to 3D convolutional kernels. Furthermore, Brock et al. [5] and Wu [35] proposed to build deeper 3D CNNs following the structures from inception-module, residual connections, and Generative Adversarial Network (GAN) to improve the generalization capability. However, these methods are based on 3D convolutions, thereby having high computational complexity and GPU memory consumption.

An alternate approach is based on projected 2D views of the 3D object to exploit established 2D CNN architectures. MVCNN [31] renders multiple 2D views of a 3D object and use them as an input to 2D CNNs. Some other works [28, 26, 1] propose to use the 2D panoramic views of a 3D shape. However, these methods can only observe

partial parts of the 3D object, failing to cover full 3D surfaces.

To address all these limitations, we introduce a novel 3D shape representation technique using stereographic mapping to project the full surfaces of a 3D object onto a 2D planar image. This 2D stereographic image becomes an input to our proposed shallow 2D CNN, thereby reducing substantial amount of network parameters and GPU memory consumption compared to the state-of-the-art 3D convolution-based methods, while achieving high accuracy.

By taking advantage of multiple projected views generated from a single 3D shape, we propose *view ensemble* to combine predictions of most discriminative views, which are sampled by our view selection network. On the contrary, Conventional methods [31, 28, 33, 20, 34, 1] simply aggregate the responses of all multiple views via max or average pooling.

Chapter 2

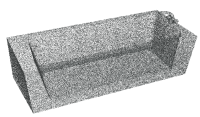
Related Work

In this section, we review recent deep learning methods for 3D feature learning. These methods are categorized in term of different feature representations; (1) point cloud-based representations, (2) 3D model-based representations, and (3) 2D and 2.5D image-based representations.

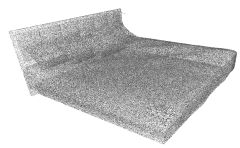
2.1 Point cloud-based methods

While previous works often combine hand-crafted features or descriptors with a machine learning classifier [11, 32, 4, 9], the point cloud-based methods operate directly on point clouds Fig. 2.1 in an end-to-end manner. In [6, 21, 16], the authors designed novel neural network architectures suitable for handling unordered point sets in 3D.

Features based on point clouds often require spatial neighborhood queries, which can be hard to deal for inputs with large numbers of points.

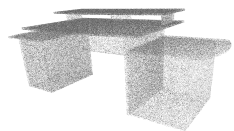


(a) Bathtub

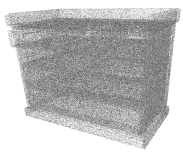


(b) Bed

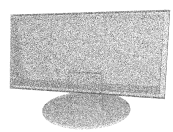
(c) Chair



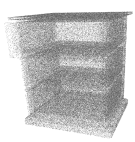
(d) Desk



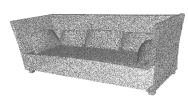
(e) Dresser



(f) Monitor



(g) Night_stand



(h) Sofa

(i) Table

(j) Toilet

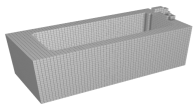
Figure 2.1: Point-cloud representation for sample shapes of ModelNet-10

2.2 3D model-based methods

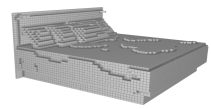
Voxel-based methods learn 3D features from voxels which represent 3D shape by the distribution of corresponding binary variables Fig. 2.2.

In 3D shapeNet [36], the authors proposed a method which learns global features from voxelized 3D shapes based on the 3D convolutional restricted Boltzmann machine. Similarly, Maturana and Scherer [18] proposed VoxNet which integrates a volumetric occupancy grid representation with a supervised 3D CNN. In a follow-up, Sedaghat et al. [25] extended VoxNet by introducing auxiliary task. They proposed to add orientation loss in addition to the general classification loss, in which the architecture predicts both the pose and class of the object. Furuya et al. [10] proposed Deep Local feature Aggregation Network (DLAN) which combines rotation-invariant 3D local features and their aggregation in a single architecture.

Sharma et al. [27] proposed a fully convolutional denoising auto-encoder to perform unsupervised global feature learning. In addition, 3D variational auto-encoders and generative adversarial networks have been adopted by Brock et al. [5] and Wu et al. [35], respectively. Furthermore, recent works[34, 22] exploit the sparsity of 3D input using the octree data structure to reduce the computational complexity and speed up the learning of global features.

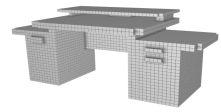


(a) Bathtub

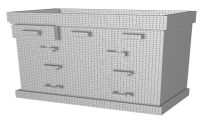


(b) Bed

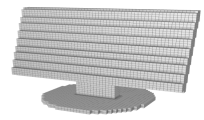
(c) Chair



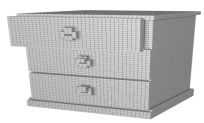
(d) Desk



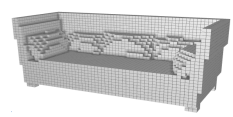
(e) Dresser



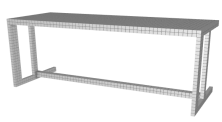
(f) Monitor



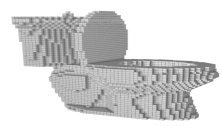
(g) Night_stand



(h) Sofa



(i) Table



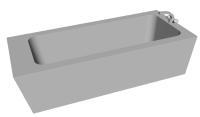
(j) Toilet

Figure 2.2: Point-cloud representation for sample shapes of ModelNet-10

2.3 2D/2.5D image-based methods

Image-based methods have been considered as one of the fundamental approaches in 3D object classification Fig.2.3. Light Field descriptor (LFD)[8] by Chen et al. used multiple views around a 3D shape, and evaluates the dissimilarity between two shapes by comparing the corresponding two view sets in a greedy way instead of learning global features by combining multi-view information. Bai et al. [3] used a similar approach but using the Hausdorff distance between the corresponding view sets to measure the similarity between two 3D shapes.

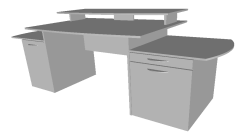
Su et al. [31] proposed a CNN architecture that aggregates information from multiple views rendered from a 3D object which achieves higher recognition performance compared to single view based architectures. By decomposing each view sequence into a set of view pairs, Johns et al. [14] classified each pair independently and learned an object classifier by weighting the contribution of each pair, which allows 3D shape recognition over arbitrary camera viewpoint. To perform pooling more efficiently, Wang et al. [33] proposed a dominant set clustering technique where pooling is performed in each cluster individually. Kanezaki et al. [15] proposed RotationNet which takes multi-view images of an object and jointly estimates its object category and poses. RotationNet learns viewpoint labels in an unsupervised manner. Moreover, it learns view-specific feature representations shared across classes to boost the performance.



(a) Bathtub

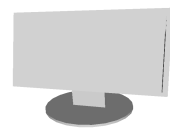
(b) Bed

(c) Chair



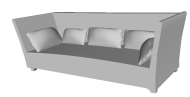
(d) Desk

(e) Dresser

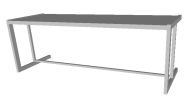


(f) Monitor

(g) Night_stand



(h) Sofa



(i) Table

(j) Toilet

Figure 2.3: Point-cloud representation for sample shapes of ModelNet-10

As an alternative approach, Gomez-Donoso et al. [12] proposed LonchaNet which uses three orthogonal slices from 3D point cloud as an input to three independent GoogLeNet networks, each network learning specific features for each slice. Cohen et al. [7] in Spherical CNNs proposed a definition for the spherical cross-correlation that is both expressive and rotation-equivariant. The spherical correlation satisfies a generalized Fourier theorem, which allows to compute it efficiently using a generalized Fast Fourier Transform (FFT) algorithm. Papadakis et al. [19] proposed PANORAMA 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. 2D Discrete Fourier Transform and the 2D Discrete Wavelet Transform are computed for each view. Shi et al. in DeepPano [28], projected each 3D shape into a panoramic view around its principal axis and used a CNN for learning the representations from these views. To make the learned representations invariant to the rotation around the principal axis a row-wise max-pooling layer is applied between the convolution and fully-connected layers. to achieve better feature descriptor for a 3D object in the training phase, Sfikas et al. [1] use three panoramic views corresponding to the major axes and taking average pooling over feature descriptor of each view for the training of an ensemble of CNNs.

Chapter 3

Proposed Stereographic Projection Network

In this section, we provide details of our proposed approach. We first describe how to transform a 3D object into a 2D planar image using stereographic projection. Then, we give the detailed description of the proposed shallow 2D CNN architecture, SPNet, followed by the procedures for view selection and view ensemble.

3.1 Stereographic Representation

Stereographic projection is a mapping that projects a 2D manifold onto a 2D plane. Such a technique is well developed in the field of Topology and Geography to project surface of the earth to a 2D planar map [30]. Since then, various projection functions have been proposed to improve the quality of mapping. In this work, we explore different types of projection functions showing that stereographic projection preserves the more detailed surface structure of a 3D object.

To construct the stereographic representation of a 3D object, we first normalize the 3D object such that a unit sphere can fully cover it. We then translate the origin of the sphere to the center of the object assuming that the orientation of the object is aligned. For each point p on the surface of the object, we denote e as a unit vector from the origin o to the point p as shown in Fig. 3.1(a). By assuming that the poles are aligned

with the z-axis, image coordinates in 2D mapped image can be determined by different types of projection functions as follow:

UV Projection [30]:

$$u = 0.5 + \frac{\lambda}{2\pi}, \tag{3.1}$$

$$v = 0.5 - \frac{\phi}{\pi}, \tag{3.2}$$

Kavrayskiy VII Projection [30]:

$$u = \frac{3\lambda}{2}\sqrt{\frac{1}{3} - (\frac{\phi}{\pi})^2}, \tag{3.3}$$

$$v = \phi, \tag{3.4}$$

Eckert IV Projection [30]:

$$u = 2\lambda\sqrt{\frac{4 - 3\sin|\phi|}{6\pi}}, \tag{3.5}$$

$$v = \sqrt{\frac{2\pi}{3}}(2 - \sqrt{4 - 3\sin|\phi|}), \tag{3.6}$$

Cassini Projection [30]:

$$u = 2\arcsin(\cos\phi\sin\lambda), \tag{3.7}$$

$$v = \arctan 2(\frac{\tan\phi}{\cos\lambda}), \tag{3.8}$$

where, $\lambda = \arctan 2(e_x, e_y)$ and $\phi = \arcsin e_z$ refer to the longitude and the latitude, respectively.

After determining the UV coordinates of the 3D object in the 2D mapped image, we set a value of each pixel with the distance of the corresponding point p from the origin in the 3D object as shown in Fig. 3.1(a). We discretize the 2D image to have a size of 128 x 128. As shown in Fig. 3.1(b)-(h). We note that the stereographic representations of 3D object preserve more details about the shape of the 3D object compared to other approaches such as panorama [28, 26, 1], slice [12], and multi-view [31, 15] representations.

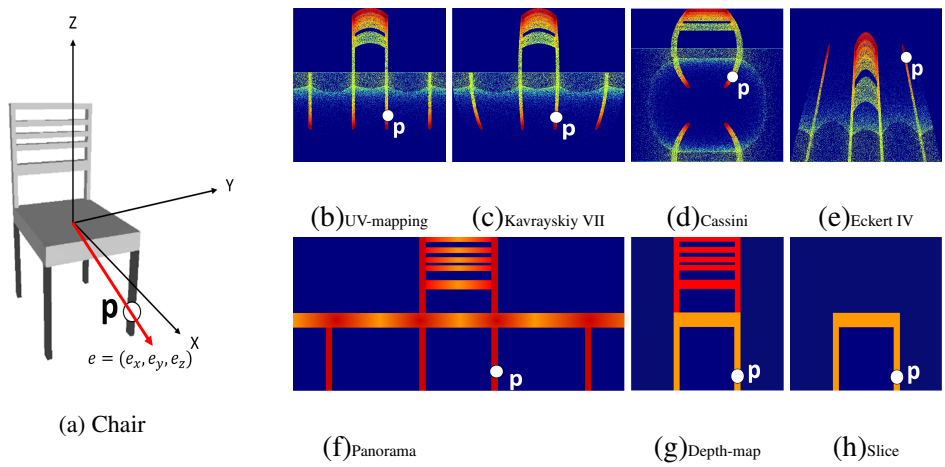


Figure 3.1: 2D representation of surface of 3D object. (a) 3D mesh model with a point p at the surface and its corresponding unit vector e from the origin 0. (b) (e) different types of stereographic projection functions. (f) Panoramic view [28, 26, 1]. (g) Depth-map [3, 14]. (h) Slice-based projection [12].

3.2 Network Architecture

We propose SPNet, a very shallow 2D CNN which consists of 4 convolutional layers and two fully connected layers. For each convolutional layer, we use a convolutional kernel of size 3x3 followed by tanh non-linearity and 2x2 max-pooling layers except for the last convolutional layer where we use global average pooling in place of max-pooling. Each side of inputs to all convolutional layers is zero-padded by 1 pixel to keep the feature map size unchanged. We also propose to add dropout after every layer except for the last fully connected layer to prevent over-fitting and for better generalization capability. The number of feature maps of our convolutional layers is 24, 32, 48, and 64, respectively. Details of the model are shown in Fig. 3.2.

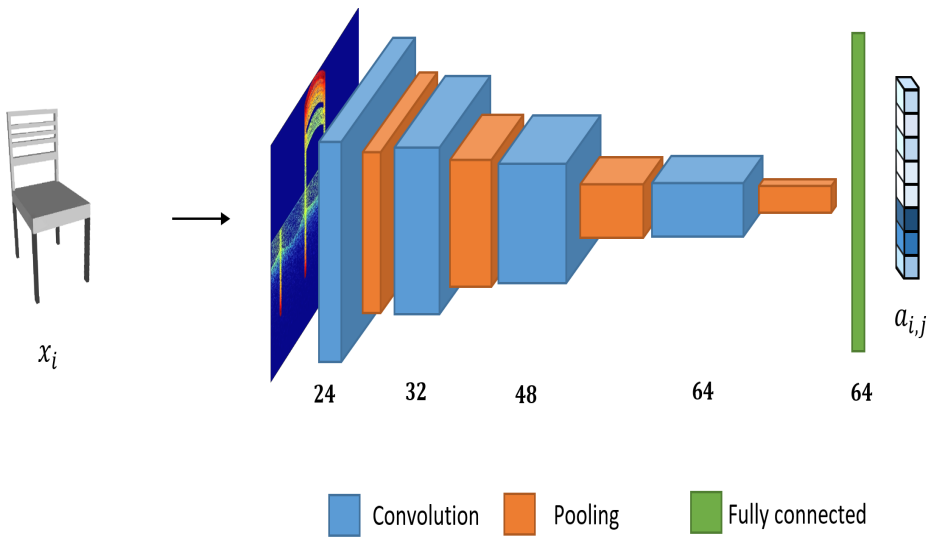


Figure 3.2: Illustration of proposed SPNet, a shallow 2D convolutional neural network architecture. $a_{i,j}$ denotes the output from the last fully connected layer.

3.3 View Selection

To construct multiple view stereographic representations from a 3D object, we augment the data with azimuth and elevation rotations. We first rotate the object along the gravity axis, each rotated 45° intervals. We further generate more views through elevation rotations with 45° intervals. Both angles are sampled uniformly from [0, 360°] to generate $N = 64$ views in total in Fig. 3.4. Let us denote generated views of the object x_i as $\{v_{i,j}\}_{j=1}^{N}$ where i refers to the instance of the 3D object and j refers to the rotated instance of the corresponding 3D object.

All views $v_{i,j}$ are fed into the trained SPNet in Fig. 3.2 to extract the view-specific feature response maps $a_{i,j}$. All N view-specific features are then passed through a one-by-one convolutional layer to perform weighted-average over all view-specific features. The output is then used as a final prediction score map. The overall process of view selection is visualized in Fig. 3.3. The one-by-one convolutional layer in our view selection learns the importance of each view-specific features, thereby indicating the degree of contributions of each view to the final prediction. Once our view selection converges, we select M most discriminative views $\{v_{i,j}^{*}\}_{j=1}^{M}$ where $M \leq N$ by observing the highest weight values in the one-by-one convolutional kernel.

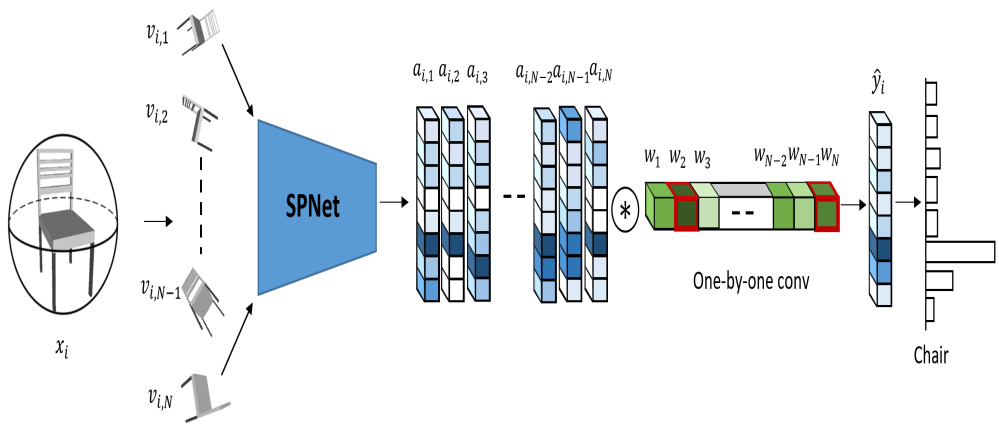


Figure 3.3: Illustration of view selection and view ensemble. Both view selection and view ensemble adopt the same architecture but with different numbers of views to train each model. $a_{i,j}$ is the output of SPNet for the corresponding view $v_{i,j}$. Darker colors on the view-specific features $a_{i,j}$ and on the weights of the one-by-one convolutional layer denote higher values. Red boxes on the weights of the one-by-one convolutional kernel indicate the selected views.

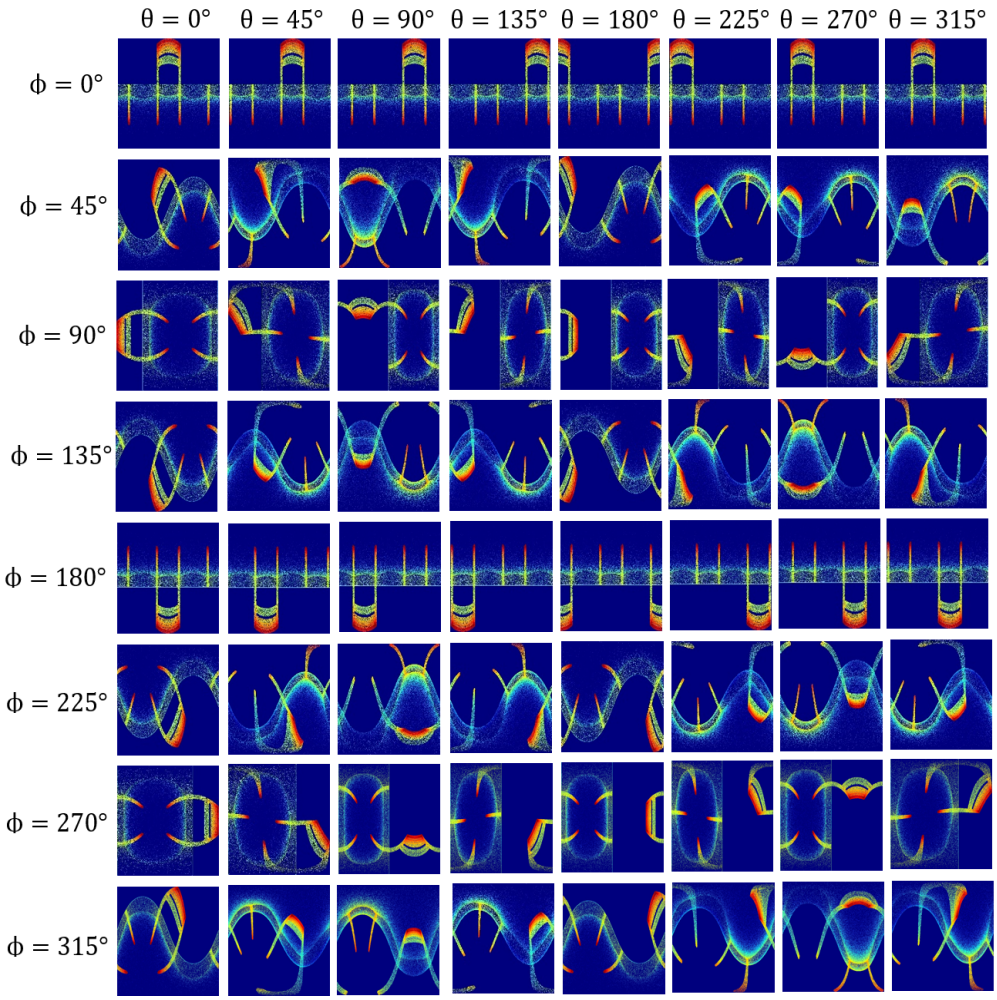


Figure 3.4: Illustration of UV-mapping for 64 different views of a chair. θ and ϕ indicate the rotation angle around the Z-axis and Y-axis, respectively.

3.4 View Ensemble

Many recent works [23, 13, 17] have shown that the use of ensemble technique provides a significant boost to the classification performance. Thus, we also exploit the weighted-average over predictions of M selected views $\{v^*_{i,j}\}^M_{j=1}$.

We train our view ensemble model in Fig. 3.3 by using only the selected most important M views $\{v^*_{i,j}\}^M_{j=1}$. Moreover, we examine different types of aggregation for the predictions of M selected views:

Max-pooling:

$$\hat{y}^*_i = \max_j \{a^*_{i,j}\}, \tag{3.9}$$

Avg-pooling:

$$\hat{y}^*_i = \frac{1}{M} \sum_{j=1}^{M} a^*_{i,j}, \tag{3.10}$$

Weighted-average:

$$\hat{y}^*_i = \sum_{j=1}^{M} w^*_j a^*_{i,j}, \tag{3.11}$$

Where, $\hat{y}^*_i$ denotes the estimate of the object category label for each object x_i.

We have tested these three ensemble methods empirically and found that by learning the weights $\{w^*_j\}^M_{j=1}$ of the one-by-one convolutional layer properly, the weighted-average produces superior performance over the max-pooling and the average-pooling [31, 28, 33, 20, 34, 1], as shown in Table 4.3.

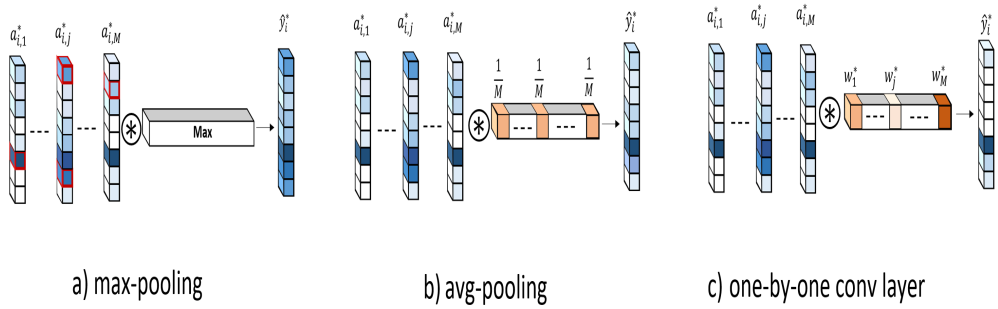


Figure 3.5: Comparison of different types of ensemble. Darker colors on each view-specific features and weights of the one-by-one convolutional kernel indicate higher values.

Chapter 4

Experimental Evaluation

4.1 Datasets

We have evaluated our method on the two subsets of the Princeton ModelNet large-scale 3D CAD model dataset [36] and the ShpeNet Core55, a subset of the ShapeNet dataset [24].

ModelNet-10 includes ten categories of 3991 and 908 models into training, and testing partitions, respectively. The dataset provides objects of same orientations.

ModelNet-40 contains 12,311 CAD models split into 40 categories that provides objects of same orientations. The training and testing subsets consist of 9843 and 2468 models, respectively.

ShapeNet Core55 contains 51,300 3D models in 55 categories and several subcategories. Two versions of ShapeNet Core55 exist (a) consistently aligned 3D models and (b) models that are perturbed by random rotations. This dataset split into three subsets of 70%, 10% and 20% for training, validation, and testing respectively. We trained and evaluated our 3D retrieval method on the training set and test set of the aligned 3D models, respectively.

4.2 Training

The baseline architecture of our CNN is shown in Fig. 3.2 which is smaller than the VGG-M network architecture that MVCNN [31] used. Table 4.1 shows the comparison of classification accuracy on the ModelNet-10 [36] of our baseline architecture and some famous Convolutional Neural Network architectures. To train SPNet, we used SGD optimizer with a learning rate of 0.01.

Table 4.1: Classification accuracy on ModelNet-10 with various network architectures for a single view

Architectures	SPNet (ours)	VGG-16	ResNet-32	ResNet-50	ResNet-101
Accuracy	**93.39%**	83.92%	91.19%	92.18%	91.41%

4.3 Choice of Stereographic Projection

We have evaluated several stereographic projection models for the 3D classification task including UV, Kavrayskiy VII, Eckert IV, and Cassini [30]. Table 4.2 shows the test results on ModelNet-10 [36], where we can clearly observe that the UV-mapping outperforms the others. Since the UV-mapping is proven to be the best, we will use this mapping function in all subsequent experiments.

Table 4.2: Classification accuracy on ModelNet-10 with various mapping functions

mapping function	accuracy
UV [30]	**93.39%**
Kavrayskiy VII [30]	93.17%
Eckert IV [30]	89.76%
Cassini [30]	92.51%
Depth-map (YZ-plane)	85.02%
Panorama (around Z-axis)	92.07%

4.4 Test on View Selection Schemes

We consider three view selection setups for the ensemble of the multi-view 2D stereographic representation to demonstrate the preferences of our view selection approach.

Case (i): Major axes

In this case, we set the viewpoints along three axes, x-axis, y-axis, and z-axis. The objects have same orientation namely that the viewpoint is along the x-axis. To obtain the two other viewpoints, each time we rotate the objects by $\theta = 90^\circ$ and $\phi = 90^\circ$ around z-axis and y-axis, respectively.

Case (ii): 12 MVCNN

In this case, we fix z-axis as the rotation axis. We place the viewpoints at $\phi = 30^\circ$ from the ground plane and each time rotate the objects by $\theta = 30^\circ$ around the z-axis to obtain 12 views for the object.

Case (iii): View Selection

Our view selection method which learns the view's influence by a one-by-one convolutional layer. We used the method on 64 different rotations by rotating the objects around z-axis and y-axis and then selected the views with the highest influence.

We compared the classification accuracy for these three view setup on the ModelNet-10 [36] with our view ensemble neural network architecture named SPNet_VE. Table 4.3 shows the comparison of classification accuracy on the ModelNet-10 [36] of plain and ensemble with the Max-pooling, Avg-pooling, and one-by-one convolutional layer as a weighted-average over the score features of the multi-view 2D representations. From these results, we observe that our learned weighted averaging of 5 views gives the best performance over other schemes, so that we use this ensemble model for our experiments.

Table 4.3: Classification accuracy on ModelNet-10 with various view selection schemes

View setup	#views	Max-pool	Avg-pool	one-by-one conv
Plain	1	93.39%	93.39%	93.39%
Major axes	3	95.15%	95.59%	96.26%
MVCNN	12	91.63%	92.51%	92.40%
View Selection	1	93.39%	93.39%	93.39%
	2	95.82%	96.15%	96.15%
	3	95.59%	95.59%	96.26%
	4	95.15%	95.48%	96.58%
	5	94.05%	95.93%	**97.25**%
	6	94.16%	95.15%	97.03%
	64	90.64%	91.74%	91.52%

4.5 3D Object Classification

We have first evaluated our baseline method SPNet in classification on both ModelNet-10 [36] and ModelNet-40 [36]. The performance of our model is measured by the average binary categorical accuracy.

We have compared our method with recent sate-of-the-art methods including 3D ShapeNet [36], GIFT [3], DeepPano [28], Multi-view Convolutional Neural Networks (MVCNN) [31], Geometry Image descriptor [29]. In addition to above methods the results are extended to include the following voxel based methods: ORION [25], 3D-GAN [35], VoxNet [18], O-CNN [34] and OctNet [22]. Table 4.4 summarizes the comparative results of classification on ModelNet-10 and ModelNet-40 in terms of GPU memory usage and the number of parameters during the training phase, and classification accuracy.

We note that in our approach, the view-ensemble model (SPNet_VE) boosts significant performance improvement over the baseline model (SPNet) by 3.9% and 4.0% on ModelNet-10 and ModelNet-40, respectively. Moreover, SPNet_VE achieved comparable results to those of the state-of-the-arts RotationNet [15], while requiring much less memory (2%) and network parameters (0.2%), respectively. Note also that there is a large gap between the average (94.82%) and maximum (98.46%) accuracy of the RotationNet [15] which shows this method is not stable while our method showed consistent performances (97.25%) for each trial of training process.

Table 4.4: Classification results and comparison to state-of-the-art methods on ModelNet-10 and ModelNet-40. Also the number of parameters and GPU memory usage. VE indicates view ensemble

InputModality	Method	GPU memory	Parameters	ModelNet	
				class 10	class 40
Point Clouds	PointNet [6]	-	3.5M	-	89.2%
	PointNet++ [21]	-	-	-	91.9%
3D Volume	ShapeNet [36]	60.5MB	15M	83.50%	77.00%
	LightNet [2]	2MB	0.3M	93.39%	86.90%
	ORION [25]	4.5MB	0.91M	93.80%	-
	VRN [5]	129MB	18M	93.60%	91.33%
	VoxNet [18]	4.5MB	0.9M	92.00%	83.00%
	FusionNet [2]	548MB	118M	93.10%	90.80%
	3D-GAN [35]	56MB	11M	91.00%	83.30%
	OctNet [22]	-	-	90.42%	-
	O-CNN [34]	-	-	-	90.6%
Others	Spherical CNNs [7]	-	1.4M	-	-
	LonchaNet [12]	-	15M	94.37%	-
2D Represen.	MVCNN [31]	331MB	42M	-	90.10%
	MVCNN-MultiRes [20]	-	180M	-	91.40%
	RotationNet [15]	731MB	42M	**98.46%**	**97.37%**
2.5D Represen.	DeepPano [28]	9.8MB	3.27M	85.45%	77.63%
	PANORAMA-NN [26]	6.77MB	2.86M	91.10%	90.70%
	PANORAMA-ENN [1]	42MB	8.6M	96.85%	95.56%
	GIFT [3]	-	-	92.35%	83.10%
	Pairwise [14]	-	42M	92.80%	90.70%
	SPNet (ours)	3MB	86K	93.39%	88.61%
	SPNet_VE (ours)	15MB	86K	97.25%	92.63%

4.6 Shape Retrieval

We have evaluated the view ensemble version, SPNet_VE with the learned five views for the 3D object retrieval task under three datasets, ModeNet-10 [36], ModelNet-40 [36] and ShepeNet Core 55 [24] by three different metrics.

mean Average Precision

Average Precision compute the average value of the precision $p(k)$ over the recall r that is the area under the precision-recall curve. For a finite number samples the area is a finite sum over every position in the ranked sequence of the samples:

$$AP = \sum_{k=1}^{n} P(k)\Delta r(k) \tag{4.1}$$

where n is the number of retrieved samples, $P(k)$ is the precision at level k in the list, and $\Delta r(k)$ is the change in recall from level $k-1$ to k.

The mean Average Precision for Q number of queries is the mean of the defined average precision score for each query.

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{4.2}$$

F-score

$F - score$ is the weighted harmonic mean of precision and recall:

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{4.3}$$

Discounted Cumulative Gain

DCG uses a graded relevance scale of samples from the result set to evaluate the usefulness, or gain, of a sample based on its position in the result list. The DCG at the position p is defined as:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{4.4}$$

Since result set may vary in size among different queries or systems, to compare performances the normalized version of DCG uses an ideal DCG. Therefore, it sorts documents of a result list by relevance, producing an ideal DCG at position p ($IDCG_p$), which normalizes the score:

$$NDCG_p = \frac{DCG_p}{IDCG_p}. \tag{4.5}$$

The average over the $NDCG$ values for all queries obtains a measure of the average performance.

Table 4.6 shows the results of our retrieval experiment on the test sets of ModelNet-10 and ModelNet-40 with mean Average Precision (mAP) in comparison with other state-of-the-art methods.

We used the learned global features of our ensemble network before the last $\tanh$ activation function. Then, we applied the function to create the best feature descriptors for all 3D objects. We sorted the most relevant 3D objects for each query from the test set by using both L_1 and L_2 distance metrics. Our SPNet_VE with L_1 achieved the best performance on ModelNet-10 and the second best on ModelNet-40. Furthermore, in Table 4.5 we evaluated SPNet_VE with different metrics on both ModelNet-10 and ModelNet-40 datasets. Note that the complexity of our model is much lighter than PANORAMA-ENN [1]; only 36% and 1% of the memory and parameters of PANORAMA-ENN are used, respectively.

Table 4.5: Retrieval results measured in $F-score$, mean Average Precision (mAP) and Normalized Discounted Gain ($NDCG$) on the ModelNet-10 and ModelNet-40

Dataset	Distance metrics	Micro-averaged			Macro-averaged		
		F-score	mAP	NDCG	F-score	mAP	NDCG
ModlelNet-10	L1	96.19%	94.20%	98.40%	95.90%	93.91%	98.30%
	L2	95.85%	92.94%	96.94%	95.43%	92.45%	96.94%
ModlelNet-40	L1	90.64%	85.21%	94.70%	83.42%	75.48%	89.37%
	L2	90.00%	84.68%	94.10%	82.23%	73.99%	88.34%

Table 4.6: Comparison of retrieval results measured in mean Average Precision (mAP) on the ModelNet-10 and ModelNet-40 datasets

Method	GPU memory	Parameters	ModelNet(mAP)	
			class 10	class 40
MVCNN [31]	331MB	42M	-	79.5%
Geometry Image [29]	-	-	74.9%	51.3%
GIFT [3]	-	-	91.12%	81.94%
DeepPano [28]	9.8MB	3.27M	84.18%	76.81%
3D ShapeNets [36]	-	-	68.3%	49.2%
PANORAMA-ENN [1]	42MB	8.6M	93.28%	**86.34%**
SPNet_VE (L2)	15MB	86K	92.94%	84.68%
SPNet_VE (L1)	15MB	86K	**94.20%**	85.21%

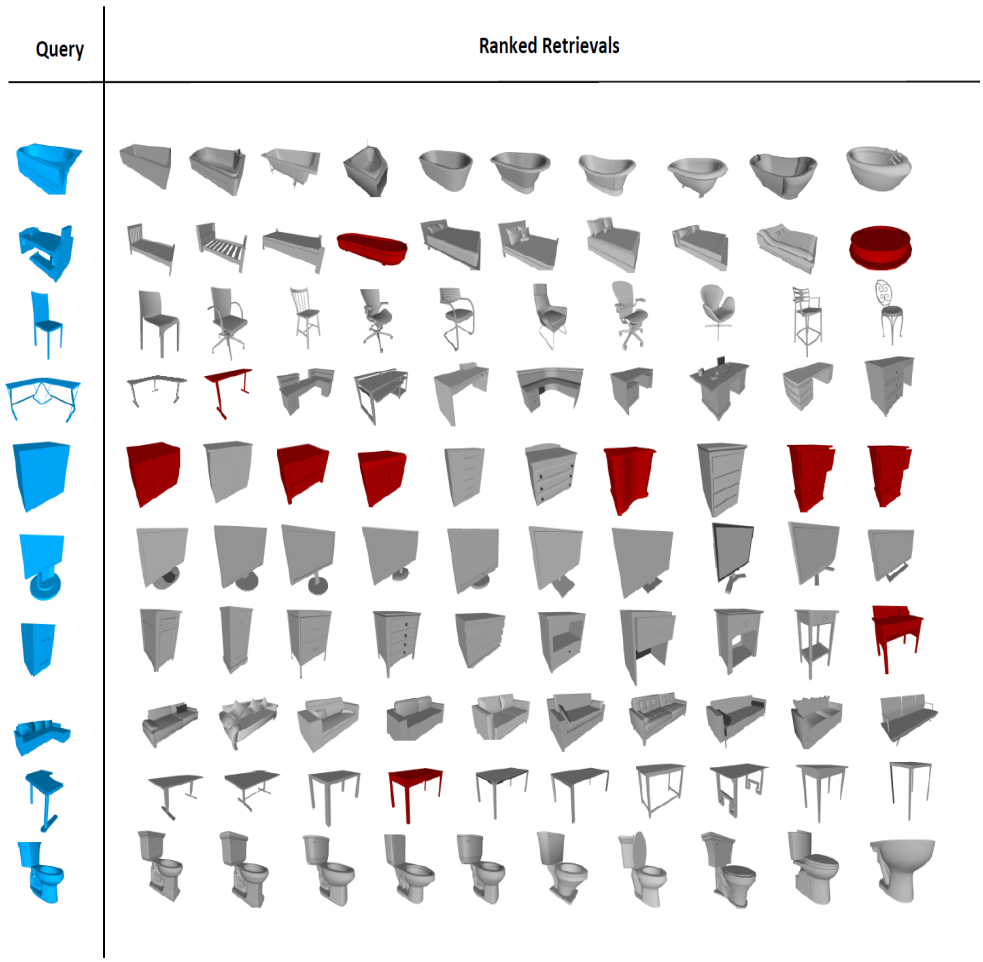


Figure 4.1: Retrieval examples by the proposed SPNet_VE on the test set of the ModelNet-10 dataset. The first column illustrates the queries and the remaining columns show the corresponding retrieved models in rank order. Retrieved objects with blue and red colors are queries and failure cases, respectively.

Table 4.7: Retrieval results measured in $F-score$, mean Average Precision (mAP) and Normalized Discounted Gain ($NDCG$) on the normalized ShapeNet Core55. VE indicates View Ensemble

Method	Micro-averaged			Macro-averaged		
	F-score	mAP	NDCG	F-score	mAP	NDCG
Kanezaki	**79.8%**	**77.2%**	86.5%	**59.0%**	**58.3%**	65.6%
Zhou	76.7%	72.2%	82.7%	58.1%	57.5%	65.7%
Tatsuma	77.2%	74.9%	82.8%	51.9%	49.6%	55.9%
FUruya	71.2%	66.3%	76.2%	50.5%	47.7%	56.3%
Thermos	69.2%	62.2%	73.2%	48.4%	41.8%	50.2%
Deng	47.9%	54.0%	65.4%	16.6%	33.9%	40.4%
Li	28.2%	19.9%	33.0%	19.7%	25.5%	37.7%
Mk	25.3%	19.2%	27.7%	25.8%	23.2%	33.7%
SHREC16-Su	76.4%	73.5%	81.5%	57.5%	56.6%	64.0%
SHREC16-Bai	68.9%	64.0%	76.5%	45.4%	44.7%	54.8%
SPNet_VE	78.9%	69.2%	**89.0%**	53.5%	39.2%	**69.5%**

Table 4.7 shows our results of the retrieval experiment on the large-scale normalized ShapeNet Core55 dataset. We tested our ensemble model by $F - score$, mean Average Precision (mAP) and Normalized Discounted Gain ($NDCG$) metrics in comparison to [3, 10]. The Macro-averaged is an unweighted average over the entire dataset while the Micro-averaged gives an average over category. The proposed method outperformed the other methods by $NDCG$ metric on both the Macro and Micro averaged.

Fig. 4.1 shows some of the retrieval cases on the test set of the ModelNet-10. The first column in the figure illustrates the queries and the remaining columns illustrate the corresponding retrieved objects in rank order. The red models indicate that the retrieved objects are in a wrong class with the queries. In other cases, the queries and the retrieved objects have the same classes. For instance, in the class of the dresser, the retrieved objects are so similar to the query while they are from different classes. The reason for these failure cases is that some objects from two different classes are hard to distinguish. Note that our approach does not have any failure cases in the class of Chair and Toilet of the ModelNet-10. Fig. 4.2 shows the confusion matrix for all 3D objects on the test set of ModelNet-10. The similarity is measured by L1 distance. Therefore, so lower values indicate higher similarities between pairs of objects.

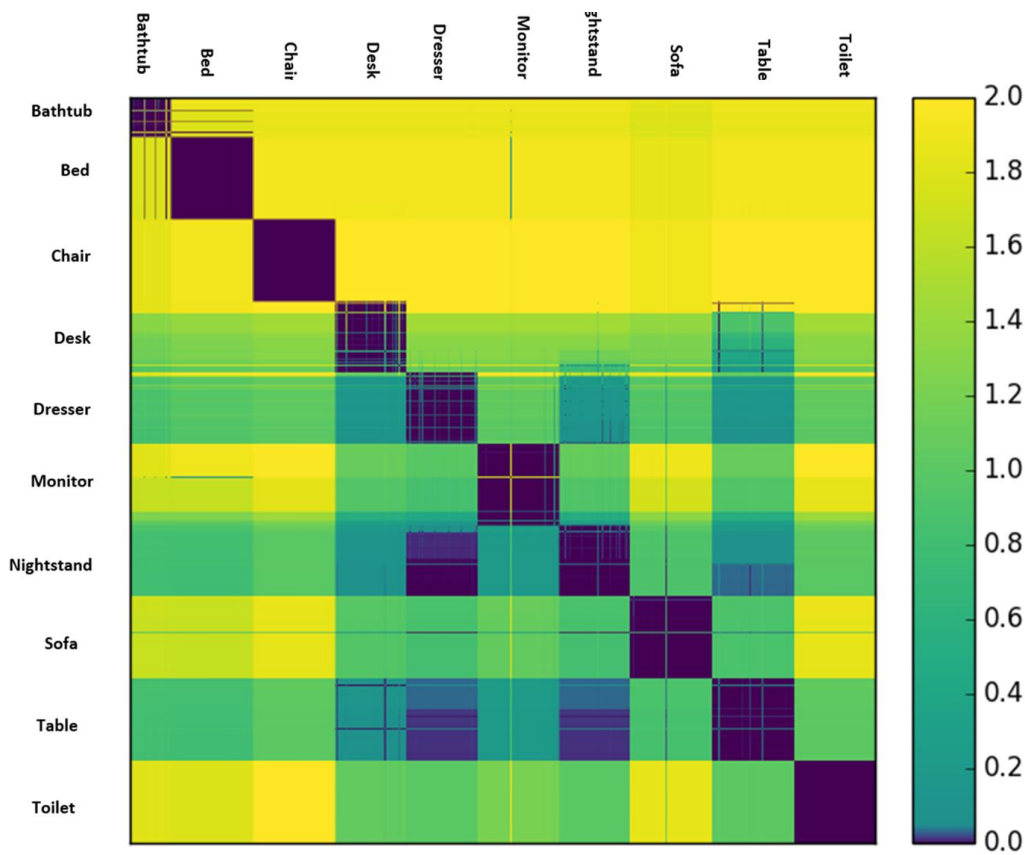


Figure 4.2: Confusion matrix for 3D objects on the test set of the ModelNet-10. Values of the matrix show the similarity between pairs of 3D objects. Higher values indicate the two 3D objects have fewer similarities; see the color bar.

4.7 Implementation

We have evaluated the proposed method SPNet on an Intel (R) Core (TM) i5 @ 3.4GHz CPU system, with 32GB RAM and NVIDIA (R) GTX 1080 Ti GPU with 12GB RAM. The system was developed in Python 3.5.2, and the network was implemented using TensorFlow-1.4.0 via CUDA instruction set on the GPU. The runtime of our SPNet and the prepossessing per each object are 2.5ms and 120ms, respectively.

Chapter 5

Conclusions

We proposed a novel ensemble architecture to learn 3D object descriptors based on the Convolutional Neural Networks. We used stereographic transformation to project 3D objects into a 2D planar followed by 2D CNNs to give confidence scores for multiple views. A one-by-one convolutional layer learns the importance of each view and selects the best views ordinary. To improve the performance, we proposed an ensemble CNN which combines the responses from the chosen views by weighted-averaging with learned weights. We evaluated our network on two large-scale datasets, ModelNet, and ShapeNet Core55. We showed that the performance of the proposed method for the classification task is par to those of the state-of-the-art approaches, while outperforms most existing works in the retrieval task. Moreover, our proposed model is most efficient regarding GPU memory usage and the number of parameters compared to existing networks.

In the future works, the ensemble neural network can be extended. Moreover, The datasets that we used do not contain texture and color information. The one channel 2D plane represented by our stereographic representation could be extended to more channels if this information existed.

Bibliography

[1] Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. Computers Graphics **71**, 208 – 218 (2018)

[2] Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning. Computers Graphics **71**, 199 – 207 (2018)

[3] Bai, S., Bai, X., Zhou, Z., Zhang, Z., Latecki, L.J.: Gift: A real-time and scalable 3d shape search engine. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5023–5032 (2016)

[4] Behley, J., Steinhage, V., Cremers, A.B.: Performance of histogram descriptors for the classification of 3d laser range data in urban environments. 2012 IEEE International Conference on Robotics and Automation pp. 4391–4398 (2012)

[5] Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. CoRR **abs/1608.04236** (2016)

[6] Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 77–85 (2017)

[7] Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical CNNs. In: ICLR (2018)

[8] Ding-Yun, C., Xiao-Pei, T., Yu-Te, S., Ming, O.: On visual similarity based 3d model retrieval. Computer Graphics Forum **22**(3), 223–232

[9] Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors (2004)

[10] Furuya, T., Ohbuchi, R.: Deep aggregation of local 3d geometric features for 3d model retrieval. In: BMVC (2016)

[11] Golovinskiy, A., Kim, V.G., Funkhouser, T.: Shape-based recognition of 3D point clouds in urban environments. International Conference on Computer Vision (ICCV) (Sep 2009)

[12] Gomez-Donoso, F., Garcia-Garcia, A., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M.: Lonchanet: A sliced-based cnn architecture for real-time 3d object recognition. In: IJCNN. pp. 412–418 (2017)

[13] Huang, H., Lu, Z., Wang, L., Liu, L., Jia, Z., Huang, Q., Peng, X., Pan, Y.: Dynamical waveforms and the dynamical source for electricity meter dynamical experiment. In: 2016 Conference on Precision Electromagnetic Measurements (CPEM 2016). pp. 1–2 (2016)

[14] Johns, E., Leutenegger, S., Davison, A.J.: Pairwise decomposition of image sequences for active multi-view recognition. In: Savva:2016:LSR:3056462.3056479. pp. 3813–3822 (2016)

[15] Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: CVPR (2018)

[16] Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: ICCV. pp. 863–872 (2017)

[17] 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 **21**(1), 31–40 (2017)

[18] Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. pp. 922–928 (2015)

[19] Papadakis, P., Pratikakis, I., Theoharis, T., Perantonis, S.: Panorama: A 3d shape descriptor based on panoramic views for unsupervised 3d object retrieval. IJCV **89**(2), 177–192 (Sep 2010)

[20] Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: CVPR. pp. 5648–5656 (2016)

[21] Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)

[22] Riegler, G., Ulusoy, A.O., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: CVPR. pp. 6620–6629 (2017)

[23] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (Dec 2015)

[24] Savva, M., Yu, F., Su, H., Aono, M., Chen, B., Cohen-Or, D., Deng, W., Su, H., Bai, S., Bai, X., Fish, N., Han, J., Kalogerakis, E., Learned-Miller, E.G., Li, Y., Liao, M., Maji, S., Tatsuma, A., Wang, Y., Zhang, N., Zhou, Z.: Large-scale 3d shape retrieval from shapenet core55. In: Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval. pp. 89–98. 3DOR '16 (2016)

[25] Sedaghat, N., Zolfaghari, M., Brox, T.: Orientation-boosted voxel nets for 3d object recognition. CoRR **abs/1604.03351** (2016)

[26] Sfikas, K., Theoharis, T., Pratikakis, I.: Exploiting the panorama representation for convolutional neural network classification and retrieval. In: 3DOR (2017)

[27] Sharma, A., Grau, O., Fritz, M.: Vconv-dae: Deep volumetric shape learning without object labels. In: ECCV Workshops (2016)

[28] Shi, B., Bai, S., Zhou, Z., Bai, X.: Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters **22**(12), 2339–2343 (2015)

[29] Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: ECCV (2016)

[30] Snyder, J.P.: Flattening the earth : two thousand years of map projections / John P. Snyder. University of Chicago Press Chicago (1993)

[31] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV. pp. 945–953 (2015)

[32] Teichman, A., Levinson, J., Thrun, S.: Towards 3d object recognition via classification of arbitrary object tracks. 2011 IEEE International Conference on Robotics and Automation pp. 4034–4041 (2011)

[33] Wang, C.: Dominant set clustering and pooling for multiview 3 d object recognition . In: BMVC (2017)

[34] Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. **36**, 72:1–72:11 (2017)

[35] Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NIPS. pp. 82–90 (2016)

[36] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR. pp. 1912–1920 (2015)

초 록

본 논문에서는 3D 물체분류 문제를 효율적으로 해결하기위하여 입체화법의 투사를 활용한 모델을 제안한다. 먼저 입체화법의 투사를 사용하여 3D 입력 영상을 2D 평면 이미지로 변환한다. 또한, 객체의 카테고리를 추정하기 위하여 얕은 2D 합성곱신셩망(CNN)을 제시하고, 다중시점으로부터 얻은 객체 카테고리의 추정값들을 결합하여 성능을 더욱 향상시키는 앙상블 방법을 제안한다. 이를위해 (1) 입체화법투사를 활용하여 3D 객체를 2D 평면 이미지로 변환하고 (2) 다중시점 영상들의 특징점을 학습 (3) 효과적이고 강인한 시점의 특징점을 선별한 후 (4) 다중시점 앙상블을 통한 성능을 향상시키는 4단계로 구성된 학습방법을 제안한다. 본 논문에서는 실험결과를 통해 제안하는 방법이 매우 적은 모델의 학습 변수와 GPU 메모리를 사용하는과 동시에 객체 분류 및 검색에서의 우수한 성능을 보이고있음을 증명하였다.

주요어: 3D 객체 분류, 3D 객체 검색, 입체화법 투사, 합성곱신경망, 다중시점 선별 및 앙상블

학번: 2017-27494

ACKNOWLEGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. NRF-2017R1A2B2011862).