Computers & Graphics 70 (2018) 140–147

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Issue on CAD/Graphics 2017

Joint analysis of shapes and images via deep domain adaptation

Zizhao Wu[a,*], Yunhui Zhang[a], Ming Zeng[b], Feiwei Qin[c], Yigang Wang[a]

[a] School of Media and Design, Hangzhou Dianzi University, China
[b] Software School, Xiamen University, China
[c] School of Computer Science, Hangzhou Dianzi University, China

ARTICLE INFO

Article history:
Received 15 June 2017
Revised 7 July 2017
Accepted 9 July 2017
Available online 14 July 2017

Keywords:
3D shape recognition
Joint analysis
Cross-modal retrieval
Convolutional neural network
Domain adaption

ABSTRACT

3D shapes and 2D images usually contain complementary information for each other, and thus joint analysis of both of them will benefit some problems existing in different domains. Leveraging the connection between 2D images and 3D shapes, it's potential to mine lacking information of one modal from the other. Stemming from this insight, we design and implement a CNN architecture to jointly analyze shapes and images even with few training data guidance. The core of our architecture is a domain adaptation algorithm, which builds up the connection between underlying feature spaces of images and shapes, then aligns and correlates the intrinsic structures therein. The proposed method facilitates the recognition and retrieval tasks. Experiments on the shape recognition tasks show that our approach has superior performance under the difficult setting: zero-shot learning and few-shot learning. We also evaluate our method on the retrieval tasks, and demonstrate the effectiveness of the proposed method.

1. Introduction

With the rapid development of Internet, massive data in multiple modalities such as images, videos, and 3D models are emerging. These heterogeneous data are usually associated to represent the same entity. For example, one can generate images, videos, and 3D models respectively to depict an object in terms of the shape, the color, the texture or the motion. The explosive increase of multimedia data has brought the challenge of how to effectively recognize, retrieve and organize these resources. Significant efforts have been devoted for these tasks, however, most of such efforts handle these modalities of data separately, and do not take full advantage of the complementary information that exists in different domains. This is especially the case in the computer graphics and the computer vision communities, where the differences in properties of viewpoint, lighting, background, occlusion, as well as data representation are most prevalent, hence hinder cross-domain analysis.

In recent years, a few works [1–3] have been proposed to address the problem of joint analysis between 3D shapes and 2D images. These works have demonstrated the great potential for solving some difficult tasks in one domain by exploiting full advantage of the complementary information in the other, which include cross-view image retrieval, cross-modal retrieval, text based shape retrieval, 3D repository and 2D repository filtering, 3D model alignment, 3D shape recognition, etc. We note that most of these works employed the Convolutional Neural Network (CNN) to learn the feature vectors for shapes and images, and achieved remarkable performance in many fields. These algorithms usually represent each 3D shape as a set of rendered images from different views around the model, in order to overcome the gap of data representation between 3D shapes and CNN models. However, with the exception of Li et al. [3], they all treat the rendered images from shapes and 2D images without distinction, which will lead the problem of domain bias [4], due to the great discrepancy in appearance between them.

In this work, we present a novel architecture for joint analysis of shapes and images, our method overcomes the discrepancy between images of different domains by introducing a domain adaptation algorithm, which leverages knowledge across domains. Specifically, a joint source and target convolutional neural network architecture is introduced to learn the feature representations of different domains. Domain adaptation is carried out in the learning process, aiming at fusing the feature representations of different domains into a shared latent space. The shared feature space is semantically meaningful, i.e. data of nearby points hold similar semantic information, regardless the modalities they belong to.

An appealing feature of the proposed algorithm is its capacity of leveraging information from one domain to the other, which facilitates the task of dealing with insufficient training data in one domain. As a result, our architecture can be used to few-shot learning [5], when a small amount of target labeled data is available from

* Corresponding author.
E-mail address: wuzizhao@hdu.edu.cn (Z. Wu).

http://dx.doi.org/10.1016/j.cag.2017.07.013
0097-8493/

each category, and zero-shot learning [6], when a small amount of target labeled data is available from a subset of the categories. We first evaluated our approach for the recognition on some popular data sets, the results show that our method matches state-of-the-art performance while requiring less training data of the target. Furthermore, we demonstrate that our method has the ability to correctly predict object category labels for unseen categories, i.e. zero-shot classification, by leveraging the knowledge across domains, which is hard for the state-of-the-art recognition methods. We also evaluated the performance of our approach on cross-modal retrieval of shapes and images, and demonstrated its effectiveness.

The main contributions of this paper include (1) we introduce a novel algorithm for joint analysis of 3D shapes and 2D images; and (2) to the best of our knowledge, we are the first to address the few-shot learning and the zero-shot learning problems in the 3D shape recognition field.

The remainder of this paper is organized as follows. We review the related work in Section 2. We then present the overview of the proposed algorithm in Section 3. The details of our domain adaptation algorithm is described in Section 4. We show some experimental results on benchmark datasets in Section 5, followed by conclusions and future work in Section 6.

2. Related work

Our method is related to prior work on shape descriptors, deep learning, domain adaptation and joint image-shape embedding, which we briefly discuss below.

2.1. Shape descriptor

Shape descriptor is an informative representation of the shape, aiming at facilitating tasks such as shape matching, recognition, retrieval, and so on. A large variety of shape descriptors has been developed in the computer graphics community. Most of the earlier shape descriptors focus on geometric properties of the shape such as shape contexts [7,8], shape distribution [9], local diameter [10], volume descriptors [11], spherical harmonics [12], conformal factors [13], Heat Kernel Signature (HKS) [14]. Some view-based descriptors also received widely attention, Cyr and Kimia [15] utilize multi-view projections to identify 3D objects and their poses. Chen et al. [16] propose Light Field Descriptor (LFD), which extracts Fourier descriptors from a set of 2D projections of views.

Instead of designing features according to human prior knowledge, discriminative feature learning provides an alternative way to characterize shapes. This especially benefits from the fast development of deep learning techniques [17], where the learned features lead to significant performance boost in classification and recognition tasks [18,19].

It is only very recent that a few works attempt to tackle 3D shape related problems via deep learning methods, such as classification, recognition and retrieval. Wu et al. [20] work with volumetric representation of 3D shapes, obtaining good results of shape classification on Princeton ModelNet [20]. Zhu et al. [21] utilize Auto-Encoder to learn 3D shape feature with multi-view depth images, leading to accurate 3D shape retrieval. One of the limitations of using 3D volumes as input is the loss in detail when shapes are voxelized. Su et al. [22] propose multi-view CNN(MVCNN) for 3D shape recognition where the features of multiple views are integrated with an extra CNN. Generating shape descriptors based on multiple views can be time-consuming and challenging for real-time retrieval. Bai et al. [23] propose real-time shape retrieval, using GPU acceleration and two inverted files (GIFT).

Our work on 3D shape recognition is similar to MVCNN in that we use deep CNN model to learn shape descriptors. It differs by the fact that our CNN model considers cross-domain input while MVCNN considers only one modality, which strengthens our method to handle with few-shot learning problem and even zero-shot learning problem.

2.2. Convolutional Neural Networks

Recently, Convolutional Neural Networks (CNNs) have been shown to be extremely effective for a variety of visual recognition tasks [18,19,24]. Though many CNN architectures have been proposed, such as AlexNet [18], GoogleNet [25], VGG [26], the network structure of AlexNet remains a popular structure, which consists of five convolutional layers with two fully-connected layers followed by a softmax layer to predict the class label. The network is capable of generating useful feature representations by learning low level features in early convolutional layers and accumulating them to high level semantic features in the latter layers.

There are several deep learning frameworks that efficiently implement the above popular networks, such as Berkeley Caffe [27] and Google Tensorflow [28]. We use Caffe in this paper.

2.3. Domain adaptation

Domain adaptation establishes knowledge transfer from the labeled source domain to the unlabeled target domain by exploring domain-invariant structures [29]. Recent studies have shown that deep neural networks can learn more transferable features for domain adaptation [30–32], which produce breakthrough results on some domain adaptation datasets. However these methods solve the problem of domain adaptation within the same modality. It is unclear how this can be done when moving across modalities. To address this issue, some notable approaches focus on the problem of jointly embedding or learning representations from multiple modalities into a shared feature space to improve learning [33] or enabling zero-shot learning [34,35].

Our work is primarily motivated by Tzeng et al. [36], that introduces a deep CNN model for the domain adaptation problem. We introduce this architecture to facilitate our task for joint analysis of shapes and images, and further make some optimizations to the original model.

2.4. Joint image-shape analysis

Multi-modal feature learning has been researched thoroughly over the past years [33], whereas only few works have addressed the problem in the computer graphics field in recent years. Herzog et al. [1] suggest a new method for structuring multi-modal representations of shapes and keywords, and adds images and sketches to the mix. This method builds the embedding mainly upon the class co-relation and hand-crafted descriptors. Hueting et al. [2] introduce a system for joint images and shapes processing to

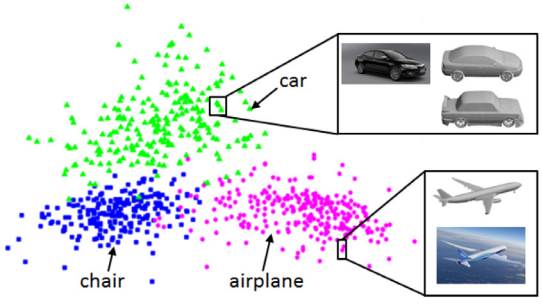

Fig. 1. Our method on joint analysis of shapes and images, and learn to obtain a joint representation in a shared latent space.

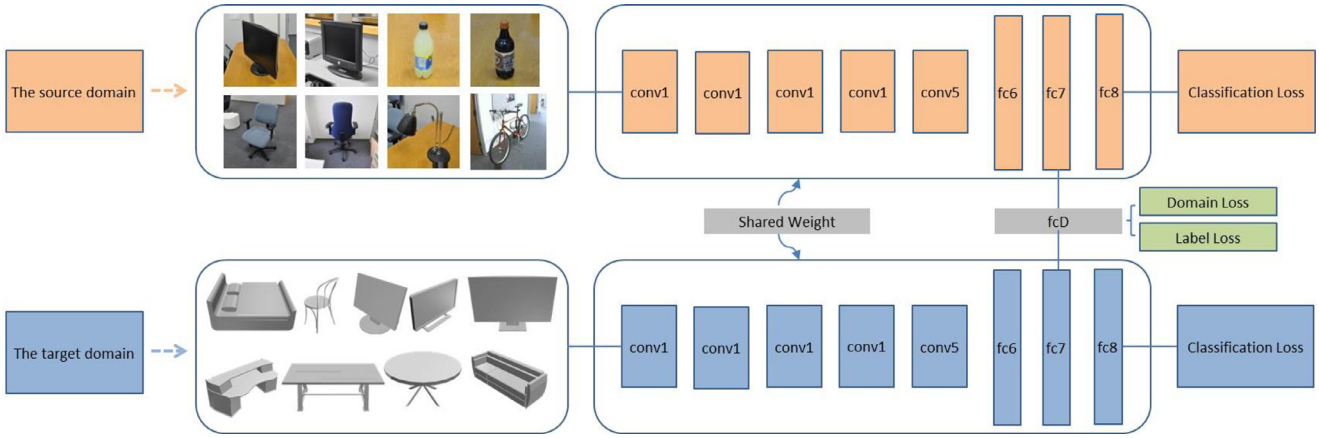

Fig. 2. The full pipeline of our CNN architecture. We build our CNN architecture with two CNNs, i.e. the source CNN model and the target CNN model, the two CNNs are trained jointly. Each CNN contains five convolutional and pooling layers, two fully connected layers, an additionally fully connected confusion layer and an output layer. The key component of our CNN is the confusion layer, which is devised to measure the cross-entropy loss for domain adaptation.

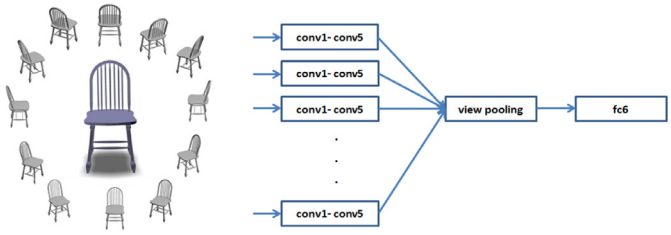

Fig. 3. We generate 12 rendered views for each model, these rendered images are taken as input to our CNN separately, and have been aggregated in a pooling layer to form a compact description for a single shape, which is located between conv5 and fc6.

facilitate analysis and exploration. However, their method does not consider the discrepancy between the different domains (images and rendered views), i.e. to compute CNN features without distinction. Li et al. [3] learns joint embedding of shapes and images via CNN image purification, they suggested a synthesize scheme, which generates images from shapes with rich variation in lighting, viewpoint, and backgrounds, to alleviate the domain discrepancy problem. However, they construct the embedding space using 3D shape similarity measure, which lead the embedding lack of semantic information. Shape2Vec [37] suggests to embed multimodal data, such as 3d shapes, images, scans, sketches and so on, into a semantic space, with the guidance of Word2Vec [38]. Instead, our method considers more compact representation, neglecting other redundant semantic information than theirs.

3. Algorithm overview

Our focus in this paper is to joint analysis of 3D shapes and 2D images with few training data guidance. Fig. 2 shows the full pipeline of our approach. The input to our method is a set of sparsely labeled 3D shapes from multiple categories, and a set of labeled images.

3.1. Pre-processing

We start by pre-processing all the 3D shapes to 2D rendered views generated by a rendering engine. We first assume that the given 3D shapes are upright oriented, since most models in modern online repositories like ModelNet [20], satisfy such requirements, for others do not have consistent orientation, we suggest the approach of [39] to deal with them.

12 virtual cameras are placed around each shape, these cameras are pointed toward to the centroid of the shape, and are elevated 30° from the plane, which is perpendicular to the upright axis. Furthermore, we set the rendering environment with the phong-reflections model [40], and scale the shape uniformly to fit into the view's volume. At last, after rendering the shapes from these cameras, 12 rendered images are generated for each shape of the input set. Since these rendered views have pure illumination, background and texture, which differ greatly from real-world images in the appearance. Therefore directly taken the labeled images as training data to train a classifier to recognize the target views will cause the problem of domain bias, which is also regarded as the domain shift [4].

To address this, we then introduce a following CNN architecture to adapt the knowledge from one domain to the other, i.e. from the real image domain to the phong-rendered image domain.

3.2. CNN architecture

Inspired by Krizhevsky and coauthors [18,27], we build our CNN architecture with two CNNs, i.e. the source CNN model and the target CNN model, the two CNNs are trained jointly. Furthermore, each CNN contains five convolutional and pooling layers, two fully connected layers, an additionally fully connected confusion layer and an output layer. The convolutional layer aims to learn feature representations of the inputs, and the pooling layer aims to achieve shift-invariance by reducing the resolution of the feature maps.

We note that an additional pooling layer [22] is introduced to aggregate multiple views in order to synthesize information from all views into a single 3D shape description.

This layer is placed between the convolutional layer (conv5) and the fully connected layer (fc6), which are closely related to the max-pooling layers and maxout layers. It differs in the dimension of their pooling operation is carried out.

After several convolutional and pooling layers, the followed two fully-connected layers aim to extract high-level global semantic features. The confusion layer is used to measure the cross-entropy loss so as to constrain the feature maps, which is the key component for our domain adaptation task. The last output layer is devised to predict class label distributions of the input. Note that we devise the source CNN and the target CNN with sharing weights, such a weight sharing mechanism has several advantages such as reducing the model complexity and constraining separate feature representation into a common space.

3.3. Feature extraction and its application

We extract features from the fully connected layer (fc7) in our case, which we empirically denote it to be the most discrimination one. The feature vectors of shapes and images are inherently lied in a common space due to the constraints of our domain adaptation.

Our architecture facilitates the tasks of shape recognition and cross-modal retrieval. The recognition results can be directly obtained through our CNN pipeline, where a softmax function is applied to the final layer to predict the class labels. For retrieval tasks, we simply compute the k-nearest neighbors for a given query input in the feature space, based on the L_2 distance.

To visualize the joint embedding of shapes and images in 2D, we first compute a distance matrix of pairwise distances between the feature vectors, then non-linear Multi-dimensional Scaling [41] is performed to learn the 2D embedding, which is the standard method for mapping the high-dimensional vectors to 2D for the visualization purpose.

4. CNN architecture for domain adaptation

In this section, we describe the detail implementation of our domain adaptation. Inspired by prior work on deep domain adaptation [36], we extend their ideas to our setting. The purpose of our domain adaptation is to learn a deep representation which should consider both the classification loss and the domain confusion loss, according to their semantic information.

Let X_S and X_T be the source data and the target data respectively, since X_T is sparsely labeled, we also denote the labeled examples both exist in X_S and X_T as X_L. Suppose there are C categories defined on X_L, and let Y be the class labels of examples. The goal of our CNN model is to produce a category classifier $f(\cdot)$ and a shared feature extractor $\phi(\cdot)$, which operates on source data examples $x_s \in X_S$, and target data examples $x_t \in X_T$. Normally, CNN models contain millions of parameters, as they have to approximate high dimensional and non-linear function. The output of a CNN model is largely affected by these parameters. Hence, we measure the mapping error of the CNN model by defining the standard softmax loss function:

$$\mathcal{L}_S(X_L, Y) = -\sum_k \mathbf{1}[y = k] log \rho_k, \tag{1}$$

where ρ is the softmax of the classifier activations, y is the discrete label of a given example.

As is mentioned previously, directly training a classifier using only the source data often leads to overfitting to the source distribution, causing reduced performance at test time when recognizing in the target domain. As a result, the feature representation of the source data and target data will lie in distinct parts of the feature space, as is illustrated in Fig. 4. To alleviate this problem, we need to learn a representation that is domain invariant.

Draw inspiration from the work [42], we consider the notion of the Maximum Mean Discrepancy (MMD) as the domain confusion loss constraint to the standard objective functional, and directly optimize our representation by minimizing the discrepancy between the source and the target distributions. Formally, the domain confusion loss can be defined as follows:

$$\mathcal{L}_D(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|. \tag{2}$$

The domain confusion loss seeks to minimize the distance between the marginal distributions of the two domains, which guarantees that the two domains share a common feature space.

Moreover, since the domain confusion loss acts to align the feature distribution of the two domains, there are no guarantees

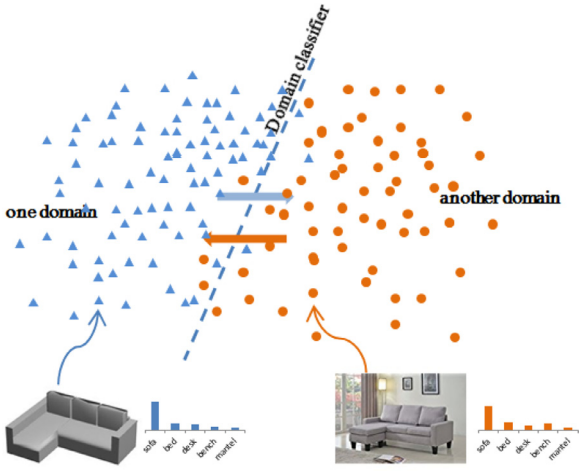

Fig. 4. We maximize domain confusion by making the marginal distributions of the two domains as similar as possible, meanwhile the class structures of different domains are preserved via the soft label functional constraints.

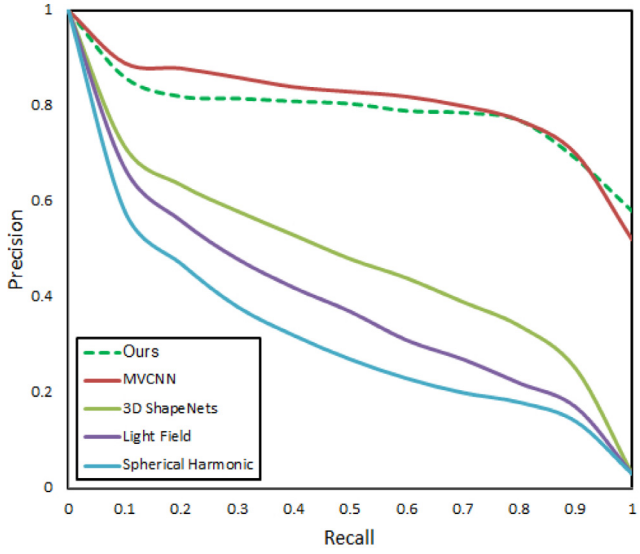

Fig. 5. Precision-recall curves of different approaches on ModelNet40.

about the alignment of the class structures between them. In other words, we need to consider not only the feature distribution between the two domains, but also the inter-class correlations between them. In our former setting, Eq. (1) treats the label all-or-none thinking, which is also regarded as the hard 0/1 labels [43]. According to some prior works [36,43], fine-tuning with hard labels limits the impact of few training examples, and makes difficulties for the network to generalize to full examples. A solution is to replace the hard labels with soft labels, which is measured through the conditional probabilities they belong to. Following them, we define a soft label loss function to ensure that the relationships between classes are preserved across the source and the target:

$$\mathcal{L}_L(X_T, Y) = -\sum_k l_k log \rho_k, \tag{3}$$

where l_k is the average over the softmax of all activations of the source examples in category k, ρ_k denotes the soft activation of the target data. This soft label loss will adjust our network parameters to produce a soft label activation that matches the average output distributions of the source examples, and guarantees the output label distributions of the target test data are non-zero probabilities, as shown in Fig. 4.

Table 1
Statistical evaluation of the average accuracy by applying few-shot recognition on the ModelNet40.

Method	Views	Samples	Accuracy	Samples	Accuracy	Samples	Accuracy
AlexNet [18]	12	1	32.5%	5	51.0%	10	58.3%
MVCNN [22]	12	1	38.4%	5	52.4%	10	67.4%
OURS:domain	12	1	56.8%	5	64.5%	10	74.2%
OURS:full	12	1	59.7%	5	68.2%	10	74.6%

Finally, the total joint loss of our optimization can be described as follows:

$$\mathcal{L} = \mathcal{L}_S(X_L, Y) + \alpha \mathcal{L}_D(X_S, X_T) + \beta \mathcal{L}_L(X_T, Y), \quad (4)$$

where the weights α and β are introduced to control how strongly the domain confusion and the label confusion influence the optimization.

5. Experimental results

In this section, we first present the implementation details of our CNN model, and then conduct a few experiments and comparisons to evaluate the efficiency and performance of our architecture. These experiments are divided into two components: shape recognition and cross-modal retrieval.

Implementation. Training CNN is a problem of global optimization. By minimizing the loss function, we can find the best fitting set of parameters. In our setting, we initialize the parameters of conv1-fc7 using the released Caffe weights. We set the weight parameter of domain confusion loss to $\alpha = 0.01$, and the weigh parameter of label confusion loss to $\beta = 0.1$. At first only the labeled examples are used to compute the classification loss, while all data are used from both domains to compute the domain confusion loss. The training procedure periodically evaluates the cross-entropy objective function (Eq. (4)) on the given training set and the validation set. The optimization procedure usually converges after around 2000 iterations. We use a model trained for 5000 iterations for testing.

5.1. Shape recognition

Data set. We evaluate our approach on the ModelNet40 [20], which is a subset of ModelNet, and contains 12,311 models in 40 categories. Since the dataset contains only 3D shapes, we have to create an additional real-image datasets for the supervision. The image dataset is developed and contains 40 categories that match the categories encountered in the ModelNet40 dataset. We create the image datasets from the ImageNet dataset, the Office dataset and the Google search. Formally, we regard the datasets of images as the source, and the datasets of render views of 3D shapes as the target.

Few-shot learning. Since in this work, we are most interested in the setting with limited target data, we first experiment our approach with the few-shot learning problem, where only a few training data are provided for the recognition tasks. We subsample the training set of the target into smaller sets with only few examples per category, and perform evaluation separately. Table 1 shows the evaluation results. Note that the accuracy arises per category as we increase the number of labeled examples in the target, this suggests that a more reasonable number of training examples are desirable. Table 1 also indicates that our approach has superior performance of few-show learning when comparing with the MVCNN and the AlexNet. Taken the 5-shot case as an example, our model achieves 68.2% classification accuracy, while the MVCNN and the AlexNet achieve 51.0% and 52.4% separately. This validates our contribution for the ability to handle with limited training data.

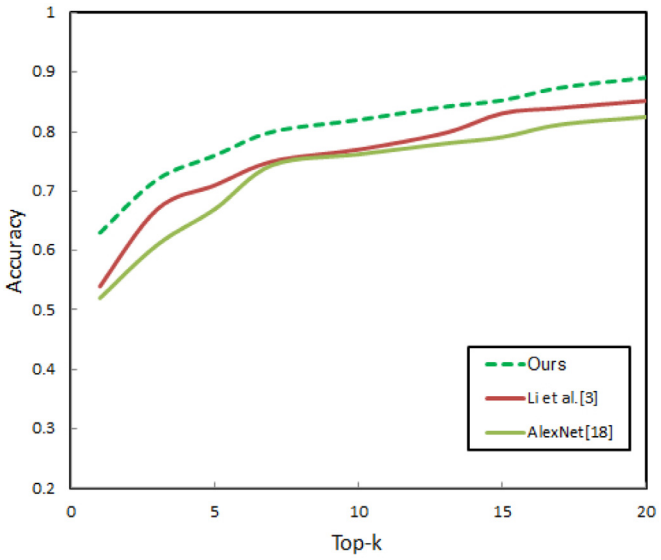

Fig. 6. Comparison of top-k accuracy on image-based shape retrieval.

We also compare our CNN model with the soft label constraint and without the soft label constraint of the algorithm. With the soft label constraint, we achieve high accuracy results.

Zero-shot learning. We then conduct experiment with the zero-shot learning setting, we consider the case in which no labeled training data are available of a subset of categories of ModelNet40. We are interesting in whether we can transfer label information from the source domain to the target.

To do this, we manually split the target dataset into two groups, one group has the training data and the other does not have. In our setting, we pick out the Bottle, the Guitar, the Lamp, the Piano, and the Sofa datasets to padding the test set of our target domain, while throwing away the training data of these datasets.

We conduct the experiment, according to the results, with our domain adaptation, even though there is no training examples of the target categories, we obtain an average precision of 51% for the test categories. In comparison, some other methods like MVCNN [22] and AlexNet [18] cannot deal with this case.

Sufficient training data. We also compare the performance of our method to some state-of-the-arts for 3D shape recognition with sufficient training data. Recent work in [22] has provided their evaluation results, which was achieved 88.3% accuracy based on sufficient training data of the same domain. In order to make a comparison, we weaken the impaction of the source network of our full architecture by decreasing the number of training images, and conduct experiments with sufficient training data of the target network. Qualitative results are shown in Table 2. The results show that we achieve comparable results to the method of [22]. This is reasonable since both methods are sharing similar setting, the main difference is that our method takes images as an additional input. We additionally emphasize that the main contribution of this work is the ability to deal with limited training data for recognition.

Fig. 7. Example results for shape-based image retrieval. The query shape are shown in the first row, followed by the 5-nearest images per query as retrieved by our method.

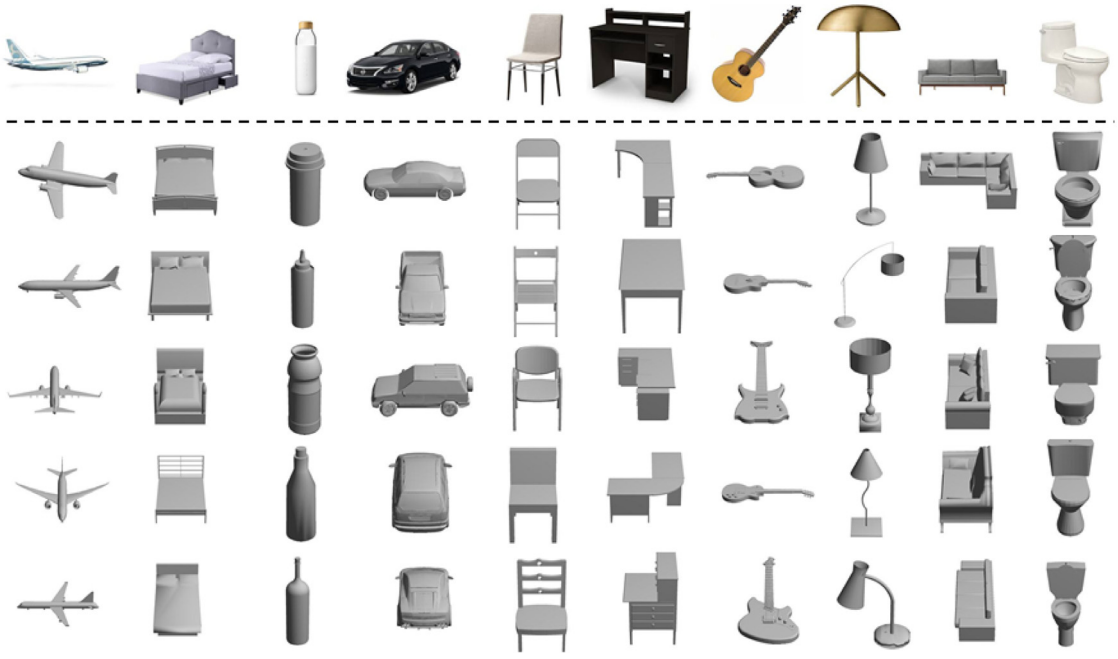

Fig. 8. Example results for image-based shape retrieval. The query image are shown in the first row, followed by the 5-nearest shapes per query as retrieved by our method.

Table 2
Statistical evaluation of the average accuracy by our algorithm with sufficient training examples.

Method	Fine-tune	Views	Accuracy
SPH [12]	–	–	68.2%
LFD [16]	–	–	75.5%
AlexNet [18]	–	12	87.5%
AlexNet [18]	ModelNet40	12	88.6%
MVCNN [22]	–	12	88.1%
MVCNN [22]	ModelNet40	12	89.9%
OURS:domain	ModelNet40+ImageSet	12	87.1%
OURS:full	ModelNet40+ImageSet	12	88.3%

5.2. Cross-modal retrieval

Our approach facilitates the task of cross-modal retrieval, i.e. shape-based image retrieval and image-based shape retrieval. Given a query 3D shape with their rendered views, we first compute its feature vectors through our CNN model, then k-nearest neighbors search is performed in the embedding space to find the most resemblance images. Fig 7 shows some examples of the retrieval results. Note that the query shapes and the retrieval results share the same object category, even they differ in appearance. Similarly, for each query image, we also retrieve similar 3D shapes according to their embedding, Fig 8 shows the example. To better illustrate the distribution of the joint embedding of shapes and images, we sub-sample three categories: the airplane, the car and the chair from the original ModelNet40, and perform the non-linear MDS in the feature space, Fig. 1 shows the visualization results, where different colors denoting different categories of the points belonging to.

We make a comparison to several other approaches, such as the AlexNet [18] and the joint embedding of Li et al. [3]. The comparison is conducted on the ModelNet10, the images of corresponding

Table 3
The ranking values of the first and the last matched correct images in the sorted list.

Median rank	Ours	AlexNet	Li et al. [3]
First match	2	4	3
Last match	8	21	19

categories are collected from the ImageNet dataset and the Google search. 100 shapes and 300 images are selected as the test set for evaluation, where each test shape is associated with 3 images manually assigned by the users. The performance is evaluated using the top-k accuracy criteria. Fig. 6 shows the results, our method has higher accuracy score comparing with the others.

As for the shape-based image retrieval case, we utilize the same dataset. Since the relationship between images and shapes is many-to-one, we resort to the median ranking measure instead of the top-k accuracy, following the work of Li et al. [3]. The median ranking by it means measures the first and last ranking values as the quantitative measure. A low ranking value of the first match implies a high accuracy score, and a low ranking of the last match represents a high recall value. Table 3 shows the quantitative results. Our approach outperforms the previous work of Li et al. [3]. In our experiments, the previous work [3] constructs the initial embedding space based on the appearance similarities between 3D shapes, which performs poorly when the shapes of different categories are similar in their appearance. Differently, our method builds the joint embedding based on their semantic information under the initial supervision of the users, which leads our embedding more semantically meaningful than theirs.

6. Conclusions

In this paper, we presented a novel algorithm for joint analysis of shapes and images. We accomplish this through a deep domain adaptation which transfers knowledge between domains in the form of a cross-entropy loss, and learn to obtain a joint representation of the shapes and images. The proposed method effectively deals with the few-shot learning and the zero-shot learning problems, which aim to learn information from only a few or zero training shapes of itself. To the best of our knowledge, this is the first work to address these problems in the 3D shape recognition domain. We evaluated our method on several benchmark datasets, and demonstrated the effectiveness of the proposed approach.

There are many limitations of our algorithm, which suggest many avenues for future work. First, we pre-process all 3D shapes with their rendered views, which will loss geometry information their own. Thus, directly take the 3D shape as the input of deep model is expecting. Second, our model considers only the domain adaptation, which limits our model to handle problems like viewport estimation, dataset filtering and so on, thus more proper optimizations over our model is appreciated. Finally, it is a worth trying direction that use the proposed method on multi-modal data across more different domains beyond image and shape.

Acknowledgment

Thanks to the anonymous reviewers for their valuable comments. This work was partially supported by the National Natural Science Foundation of China (grant nos. 61602139, 61402387, 61502129 and 61502130).

References

[1] Herzog R, Mewes D, Wand M, Guibas LJ, Seidel H. Lesss: Learned shared semantic spaces for relating multi-modal representations of 3d shapes. Comput Graph Forum 2015;34(5):141–51.
[2] Hueting M, Ovsjanikov M, Mitra NJ. Crosslink: joint understanding of image and 3d model collections through shape and camera pose variations. ACM Trans Graph 2015;34(6). 233:1–233:13
[3] Li Y, Su H, Qi CR, Fish N, Cohen-Or D, Guibas LJ. Joint embeddings of shapes and images via CNN image purification. ACM Trans Graph 2015;34(6). 234:1–234:12
[4] Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. Machi Learn 2010;79(1–2):151–75.
[5] Miller EG, Matsakis NE, Viola PA. Learning from one example through shared densities on transforms. In: 2000 Conference on computer vision and pattern recognition (CVPR); 2000. p. 1464–71.
[6] Larochelle H, Erhan D, Bengio Y. Zero-data learning of new tasks.. In: AAAI. AAAI Press; 2008. p. 646–51.
[7] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell. 2002;24(4):509–22.
[8] Körtgen M, Park G-J, Novotni M, Klein R. 3d shape matching with 3d shape contexts. In: The 7th central European seminar on computer graphics, 3; 2003. p. 5–17.
[9] Osada R, Funkhouser TA, Chazelle B, Dobkin DP. Shape distributions. ACM Trans Graph 2002;21(4):807–32.
[10] Shapira L, Shalom S, Shamir A, Cohen-Or D, Zhang H. Contextual part analogies in 3d objects. Int J Comput Vision 2010;89(1–2):309–26.
[11] Knopp J, Prasad M, Willems G, Timofte R, Gool LJV. Hough transform and 3d SURF for robust three dimensional classification. In: European conference on computer vision; 2010. p. 589–602.
[12] Kazhdan MM, Funkhouser TA, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In: SGP; 2003. p. 156–64.
[13] Ben-Chen M, Gotsman C. Characterizing shape using conformal factors. In: 3DOR; 2008. p. 1–8.
[14] Sun J, Ovsjanikov M, Guibas LJ. A concise and provably informative multi-scale signature based on heat diffusion. Comput Graph Forum 2009;28(5):1383–92.
[15] Cyr CM, Kimia BB. 3d object recognition using shape similarity-based aspect graph. In: ICCV; 2001. p. 254–61.
[16] Chen D-Y, Tian X-P, te Shen Y, Ouhyoung M. On visual similarity based 3d model retrieval. Comput Graph Forum 2003;22:223–32.
[17] Bengio Y, Courville AC, Vincent P. Unsupervised feature learning and deep learning: A review and new perspectives. CoRR 2012. abs/1206.5538.
[18] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1106–14.
[19] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10, 000 classes. In: Conference on computer vision and pattern recognition, CVPR; 2014. p. 1891–8.
[20] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, et al. 3d shapenets: a deep representation for volumetric shapes. In: IEEE Conference on computer vision and pattern recognition CVPR; 2015. p. 1912–20.
[21] Zhu Z, Wang X, Bai S, Yao C, Bai X. Deep learning representation using autoencoder for 3d shape retrieval. CoRR 2014. abs/1409.7164.
[22] Su H, Maji S, Kalogerakis E, Learned-Miller EG. Multi-view convolutional neural networks for 3d shape recognition. In: IEEE International conference on computer vision ICCV; 2015. p. 945–53.
[23] Bai S, Bai X, Zhou Z, Zhang Z, Latecki LJ. GIFT: A real-time and scalable 3d shape search engine. In: IEEE conference on computer vision and pattern recognition CVPR; 2016. p. 5023–32.
[24] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. CoRR 2015. abs/1512.07108.
[25] Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, et al. Going deeper with convolutions. CoRR 2014. abs/1409.4842.
[26] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR 2014. abs/1409.1556.
[27] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick RB, et al. Caffe: Convolutional architecture for fast feature embedding. CoRR 2014. abs/1408.5093.
[28] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR 2016. abs/1603.04467.
[29] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowledge Data Eng 2010;22(10):1345–59.
[30] Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning ICML; 2011. p. 513–20.
[31] Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, et al. Decaf: A deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31st international conference on machine learning ICML; 2014. p. 647–55.
[32] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?. In: Advances in neural information processing systems; 2014. p. 3320–8.
[33] Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY. Multimodal deep learning. In: Proceedings of the 28th international conference on machine learning, ICML; 2011. p. 689–96.
[34] Socher R, Ganjoo M, Manning CD, Ng AY. Zero-shot learning through cross–modal transfer. In: Advances in neural information processing systems; 2013. p. 935–43.
[35] Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Ranzato M, et al. Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems; 2013. p. 2121–9.

[36] Tzeng E, Hoffman J, Darrell T, Saenko K. Simultaneous deep transfer across domains and tasks. CoRR 2015. abs/1510.02192.
[37] Tasse FP, Dodgson NA. Shape2vec: semantic-based descriptors for 3d shapes, sketches and images. ACM Trans Graph 2016;35(6). 208:1–208:12
[38] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. CoRR 2013. abs/1301.3781.
[39] Liu Z, Zhang J, Liu L. Upright orientation of 3d shapes with convolutional networks. Graph Models 2016;85:22–9.
[40] Phong BT. Illumination for computer generated pictures. Commun ACM 1975;18(6):311–17.
[41] Kruskal J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 1964;29(1):1–27.
[42] Sejdinovic D, Sriperumbudur BK, Gretton A, Fukumizu K. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. CoRR 2012. abs/1207.6076.
[43] Ba J, Caruana R. Do deep nets really need to be deep?. In: Advances in neural information processing systems; 2014. p. 2654–62.