Unsupervised Multi-View CNN for Salient View Selection of 3D Objects and Scenes

Ran Song[1][0000−0002−1344−4415], Wei Zhang*[1][0000−0001−5556−3896], Yitian Zhao[2][0000−0003−4357−4592], and Yonghuai Liu[3][0000−0002−3774−2134]

[1] School of Control Science and Engineering, Shandong University, China
{ransong,davidzhang}@sdu.edu.cn
[2] Cixi Institute of Biomedical Engineering, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo, China
yitian.zhao@nimte.ac.cn
[3] Department of Computer Science, Edge Hill University, UK
liuyo@edgehill.ac.uk

Abstract. We present an unsupervised 3D deep learning framework based on a ubiquitously true proposition named view-object consistency as it states that a 3D object and its projected 2D views always belong to the same object class. To validate its effectiveness, we design a multi-view CNN for the salient view selection of 3D objects, which quintessentially cannot be handled by supervised learning due to the difficulty of data collection. Our unsupervised multi-view CNN branches off two channels which encode the knowledge within each 2D view and the 3D object respectively and also exploits both intra-view and inter-view knowledge of the object. It ends with a new loss layer which formulates the view-object consistency by impelling the two channels to generate consistent classification outcomes. We experimentally demonstrate the superiority of our method over state-of-the-art methods and showcase that it can be used to select salient views of 3D scenes containing multiple objects.

Keywords: Unsupervised 3D Deep Learning, Multi-View CNN, View-Object Consistency, View Selection

1 Introduction

The success of Generative Adversarial Network (GAN) [8] demonstrates the great value and impact of a widely applicable unsupervised deep learning framework. One important reason is that data collection for training a deep network is laborious in many tasks. This is particularly the case for 3D tasks where data collection is generally more challenging than that in 2D tasks. Therefore, a widely applicable 3D deep learning framework is potentially of broad interest.

A simple but ubiquitously true proposition is that a 3D object and its projected 2D views always belong to the same object class no matter what taxonomy is applied to the classification. We name the proposition *view-object consistency*

* Corresponding author: Wei Zhang

and propose an unsupervised 3D deep learning framework based on it. Since it is not feasible for us to thoroughly explore the utility of the framework via various 3D tasks in one paper, we pick salient view selection of 3D objects to demonstrate its effectiveness for three reasons. First, salient view selection is challenging as it does not only rely on low-level geometric features but also involves high-level semantics of objects. Thus a data-driven method is naturally sound. Second, however, it is the particular task where collecting a large amount of accurately and consistently annotated data is notoriously difficult. We found that all existing datasets are very small (e.g. 68 objects in [7] and 16 objects in [23]) no matter whether the annotations were collected directly (e.g. by marking a viewpoint on a view sphere surrounding the object [7]) or indirectly (e.g. by selecting the preferred view from two views for multiple times [23]). Third, we shall further show the advantage of an unsupervised method by extending salient view selection to 3D scenes. Salient view selection of 3D scenes can hardly be addressed by a weakly supervised method relying on such annotation as a single class label as a scene often contains objects belonging to different classes.

The problem of salient view selection of 3D objects is arguably well defined. Besides related literatures in computer vision and graphics to be discussed in Section 2, researchers in psychology [6, 2] have revealed that for many classes of familiar objects, the preferred views are reasonably consistent among the human subjects. To make it clear, the most salient view of a 3D object herein is defined as the view that a human subject likes most for whatever reason. And we shall evaluate our method using the publicly available benchmark [7] where subjects were asked to rotate a 3D object to directly select the view that they preferred.

To instantiate the view-object consistency in the context of salient view selection, we develop a multi-view convolutional neural network (CNN). It formulates the view-object consistency through a two-channel architecture and a new loss function. It also integrates with an important heuristic of human's view preference via a specifically designed layer. The proposed multi-view CNN is trained end-to-end in an unsupervised manner using only a collection of 3D objects without any manual annotations and is thus named as Unsupervised Multi-View CNN (UMVCNN). Overall, it exploits both intra-view and inter-view knowledge via a multi-view representation of 3D objects for salient view selection.

The contribution of our work is hence threefold:

(1) We propose an unsupervised framework of 3D deep learning where the core idea is valid ubiquitously and thus potentially has a range of applications.
(2) We propose a multi-view CNN in accordance with this unsupervised framework to address the classical problem of salient view selection of 3D objects.
(3) By the unsupervised 3D deep learning framework, we extend salient view selection from individual 3D objects to scenes containing multiple objects.

2 Related work

We categorise the literatures into three groups. The first group is based only on handcrafted attributes of 3D objects; the second group is essentially shallow

learning of a certain model to combine multiple attributes while all attributes are not learned but still handcrafted; the third group is based on deep learning where some, if not all of the attributes, are learned via deep neural networks.

Handcrafted attributes. Polonsky et al. [21] explored general frameworks for view selection by analysing several handcrafted attributes associated to geometrical or statistical properties of a 3D object or its projected 2D views. Lee et al. [16] selected salient views using the attribute of mesh saliency computed via Gaussian-weighted mean curvatures. Yamauchi et al. [33] employed mesh saliency as the intra-view cue for finding salient views while taking into account such inter-view cue as the similarity of projected views. [17] computed a saliency measure based on both local geometrical and global topological attributes for salient view selection. However, most methods based on handcrafted attributes do not generalise well due mainly to the limited expressive capabilities of the attributes extracted by some fixed schemes for objects of different classes.

Handcrafted attributes with shallow learning. Vieira et al. [29] learned good views via an SVM classifier where the candidate views were represented by a collection of handcrafted attributes. To investigate human view preference, Secord et al. [23] collected a small dataset to learn a regression model combining a list of handcrafted attributes. Mezuman and Weiss [18] leveraged Internet images to learn the view from which we most often see the object, where the handcrafted GIST descriptor was employed to measure view similarity. He et al. [10] proposed a multi-view learning framework exploiting both 2D and 3D handcrafted attributes to recommend viewpoints for photographing architectures.

Deep learning. Apart from the psychological work [27, 6, 9], in computer vision, there is also evidence [32, 26, 19] of the relation between view selection and object recognition where view-dependent attributes were extracted via deep networks for 3D object recognition. Kim et al. [14] and Song et al. [25] leveraged CNNs for view selection instead of improving recognition accuracy. Our work is inspired by them but fundamentally different for two reasons: 1) both [14] and [25] require annotated data for training while our work is unsupervised; 2) both of them cannot be trained end-to-end where the former trains two CNNs and a Random Forest classifier separately and the latter trains a CNN and a Markov Random Field individually while our UMVCNN is trained fully end-to-end.

3 Salient view selection via UMVCNN

In this section, we first describe each component of our method in a piecewise manner. We then elaborate the implementation as a whole where each component is situated in the context of the complete pipeline.

3.1 Multi-view representation of a 3D object

Multi-view CNNs have been used to adapt image-based deep networks to 3D objects where an object is represented as a set of its projected views. Compared with other methods which generalise deep learning to non-Euclidean domains,

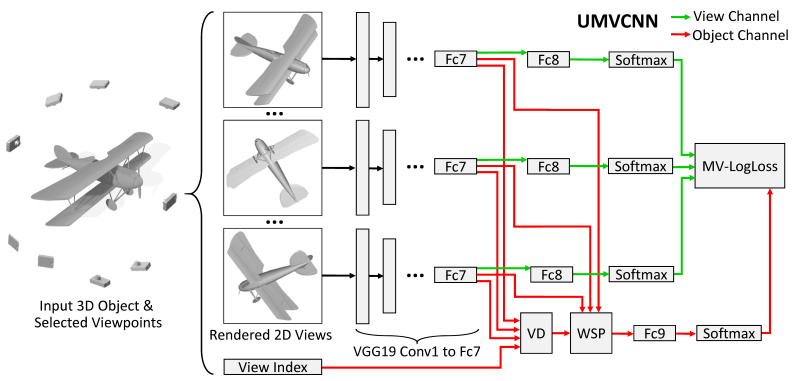

Fig. 1. Overview of the proposed UMVCNN containing two channels. The green and the red arrows denote the view channel and the object channel respectively. "VD" and "WSP" denote the view distinction and the weighted sum pooling layers respectively.

multi-view CNNs showed state-of-the-art performance in various 3D shape understanding tasks [26, 22, 13, 12]. One consensus is that we should avoid using the very 'bad' views usually defined as the ones that cause misunderstanding of objects. We propose a scheme considering two low-level attributes to ensure that the selected 2D views for representing a 3D object are at least 'not very bad'.

We start with an icosahedron to uniformly sample a view sphere surrounding the input object. Then we iteratively subdivide the icosahedron to produce more viewpoints on the view sphere. We end with a polyhedron with 162 vertices. Next, we rank the views taken from these viewpoints based on the attributes of view area and silhouette length. View area is calculated as the area of the projection of the object as seen from a particular viewpoint. Silhouette length is the length of the outer contour of the silhouette of the object as seen from a particular viewpoint. We collect the top $N = 20$ views with the highest ranks on average based on the two attributes as the multi-view representation of the 3D object.

3.2 UMVCNN architecture

Overview. Fig. 1 illustrates the architecture of the UMVCNN. It starts with VGG-19 [3] as the backbone and then branches off the view and the object channels after the Fc7 layers. Through the view distinction (VD) layer, it generates an inter-view heuristic using the deep features extracted from the 2D views. A weighted sum pooling (WSP) layer is then employed to incorporate this heuristic and multiple intra-view features derived from each individual view into a single tensor encoding the information corresponding to the entire 3D object. These two layers and the newly added fully connected layer Fc9 followed by a Softmax normalisation form the object channel. It outputs to the loss layer a vector composed of the probabilities of the 3D object belonging to a certain class. On the other hand, we still keep the original Fc8 layer of VGG-19 in the view channel that generates a vector for each view predicting which class the view belongs

to. Every VGG-19 layer from Conv1 to Fc8 in the UMVCNN shares the same weights for all views. Finally, the outputs of the view and the object channels converge at the newly designed Multi-View Logistic Loss (MV-LogLoss) layer that formulates the view-object consistency to enable an unsupervised learning.

View distinction (VD) layer. Existing work [33, 23, 34] showed that humans subjects find a good view by not only scrutinising its own intra-view content, but also comparing it with other views of the same object. Note that a limitation of most previous work is the lack of the consideration of such inter-view knowledge in their algorithms. In this work, we propose a heuristic mechanism to formulate the inter-view knowledge via paired comparisons of views. Previous work [31, 15] in psychology pointed out that a basic principle in human visual system is to suppress the response to frequently occurring features, while at the same time it remains sensitive to features that deviate from the norm. We thus propose the VD layer as a heuristic method to formulate this principle where the view most different from all other views are regarded as the most distinct one. The VD layer takes as input the outputs of all Fc7 layers. Since one 3D object is represented as N views, the input of the VD layer is a matrix of size $4096 \times N$ for a given object. Each of its columns can be regarded as a feature descriptor of one view. The VD layer outputs an N-dimensional vector to the WSP layer. Each element of the vector corresponds to the distinction of a particular view.

Given two views V_i and V_j, their difference can be measured as the Euclidean distance between their feature descriptors F_i and F_j output by the Fc7 layer (with ReLU activation). However, this measure is insufficient as a view tends to have similar content with its neighbouring views. If a view is even very different from its neighbouring views, it is likely to contain some unique content and thus be considered confidently distinct from the other views. Hence, the dissimilarity of two views should be proportional to the difference computed as the Euclidean distance between their feature descriptors and inversely proportional to the geodesic distance between their corresponding viewpoints. Such a heuristic also computationally holds for symmetric objects. For symmetric views, the dissimilarity is always 0 as $F_i = F_j$ and thus has nothing to do with the geodesic distance between them. Besides the N projected views, the UMVCNN also requires as input the view index $\text{VInd}_i \in \{1, 2, ..., 162\}$ generated as a byproduct when creating the multi-view presentation of the object (see Section 3.1).

Let $\text{Geod}(\text{VInd}_i, \text{VInd}_j)$ be the geodesic distance between the viewpoints corresponding to V_i and V_j, the dissimilarity between them is defined as:

$$D_{ij} = \frac{\|F_i - F_j\|}{1 + \alpha \cdot \text{Geod}(\text{VInd}_i, \text{VInd}_j)}, \quad \text{s.t. } i, j \in \{1, 2 \ldots, N\} \text{ and } i \neq j \tag{1}$$

where $\alpha = 2$ in our implementation. The distinction of V_i is then computed as the sum of its pairwise dissimilarity to all the other views.

$$S_i = \sum_{j \neq i} D_{ij}. \tag{2}$$

Eqs. (1) and (2) are both differentiable. Thus for back-propagation, given that the gradient passed to the VD layer is an N-dimensional vector $\mathcal{S}$, according to

the chain rule, the gradient $\mathcal{F}$ with regard to its input can be computed as

$$\mathcal{F}_i = \mathcal{S}_i \frac{\partial S_i}{\partial F_i} \tag{3}$$

Considering Eqs. (1) and (2) and the partial derivative of the Euclidean distance function $\frac{\partial \|x\|}{\partial x_i} = \frac{x_i}{\|x\|}$, it can be computed as

$$\frac{\partial S_i}{\partial F_i} = \sum_{j \neq i} \frac{F_i - F_j}{(1 + \alpha \cdot \mathrm{Geod}(\mathrm{VInd}_i, \mathrm{VInd}_j)) \cdot \|F_i - F_j\|}. \tag{4}$$

Weighted sum pooling (WSP) layer. To implement the view-object consistency through the loss layer requiring the outputs of the view and the object channels to have the same dimensions, we need to pool to aggregate the learned knowledge across all 2D views to create a single descriptor for the 3D object. Also, we need to consider how to cast the influence of view distinction into this aggregation process where distinct views should have larger weights. Thus instead of the popular element-wise max pooling [26, 13] in multi-view CNNs, we carry out a WSP to incorporate view distinction as the weights into the pooling

$$P = \sum_{i=1}^{N} F_i S_i \tag{5}$$

where F_i is the column vector of the output of the Fc7 layer F which denotes the feature descriptor of V_i. S_i is its distinction output by the VD layer. The output of the WSP layer P regarded as the feature descriptor of the 3D object is thus estimated as the weighted sum of the feature descriptors of all views where the weights are their distinctions. Eq. (5) can be expressed in a bilinear form as $P = FS$. Thus with the gradient $\mathcal{P}$ passed to the WSP layer, the gradients $\mathcal{F}$ and $\mathcal{S}$ with regard to the inputs F and S respectively can be computed as

$$\mathcal{F} = \mathcal{P} S^T, \quad \mathcal{S} = F^T \mathcal{P}. \tag{6}$$

MV-LogLoss Layer. We propose the MV-LogLoss layer to formulate the view-object consistency enabling an unsupervised learning. No matter what the taxonomy is, the outcome of the classification based on each 2D view should be consistent with that based on the entire 3D object. As illustrated in Fig. 1, either of the view and the object channels alone is specifically designed to have the architecture of a classification network, which facilitates the formulation of the view-object consistency. Moreover, such a design benefits salient view selection as the features vital for object classification are usually important for the selection of a salient view. Psychological studies [27, 6, 9] validated that a good view of an object can significantly help people to correctly recognise it.

The MV-LogLoss simply adapts the log loss in a multi-view scenario. This loss layer first computes the individual log loss of the softmax-normalised output of each Fc8 layer, $\mathcal{V}(i)$ with regard to that of the Fc9 layer, $\mathcal{O}$, which represent

the final outputs of the view channel and the object channel respectively. The multi-view loss is then computed as the sum of all individual log losses:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} \mathcal{O}_c \cdot \log\left(\mathcal{V}_c(i)\right) \tag{7}$$

where for simplicity, we write the output of the view channel $\mathcal{V}_c(V_i)$ as $\mathcal{V}_c(i)$. Through training, Eq. (7) is minimised by impelling $\mathcal{O}$ to be consistent with $\mathcal{V}(i)$ and the view-object consistency is thus realised.

It can be clearly seen that the MV-LogLoss defined as Eq. (7) does not rely on any annotations as $\mathcal{O}_c$ and $\mathcal{V}_c(i)$ are internally generated by the object channel and the view channel of the UMVCNN respectively. C in Eq. (7) is a picked integer defining the output dimension of the Fc8/Fc9 layer when building the UMVCNN. And the influence of varying C will be studied in Section 4.4.

3.3 Salient view selection

In the deployment, given an object represented as a set of N views, we first feed the views into the UMVCNN and hijack the output of the Softmax layer connected with the Fc9 layer during the forward-propagation to predict its object class $\mathcal{C}$. Then, we back-propagate a C-dimensional one-hot vector where only the entry of index $\mathcal{C}$ is 1 from this Softmax layer to the input views with all network weights fixed. It leads to a per-pixel saliency map I_i for all pixels in each view V_i based on their influence on the predicted class $\mathcal{C}$. I_i can be interpreted as a measure of pixel importance with regard to the recognition of the object. Like previous methods [16, 23, 17] and also to facilitate evaluations, we are keen to obtain the goodness of any viewpoint, which requires to generate a per-vertex saliency map. We thus employed the 2D-to-3D saliency transfer scheme proposed in [25] to derive a 3D saliency map H_i from I_i. Finally, we hijack the output of the VD layer S_i as the weighting parameters which represent the learned view distinction to aggregate multi-view saliency maps H_is into a single one:

$$H = \sum_{i=1}^{N} S_i H_i. \tag{8}$$

We then select the viewpoint that maximises the sum of the saliency map H for the visible regions of the 3D object as the salient viewpoint:

$$v_s = \arg\max_{v}\Big(\sum_{m\in B(v)} H(m)\Big) \tag{9}$$

where $B(v)$ is the set of the vertices visible from the viewpoint v and $H(m)$ denotes the saliency of the vertex m. $M(v) = \sum_{m\in B(v)} H(m)$ can be regarded as the saliency map of the viewpoints. Fig. 2 shows the 2D representation of the unwarped viewpoint saliency map on a view sphere normalised to the interval of $[0, 1]$. It is generated via the Mercator projection where the x and the y axes correspond to the latitude and the longitude respectively. Note that initially the model is not up oriented in the view sphere.

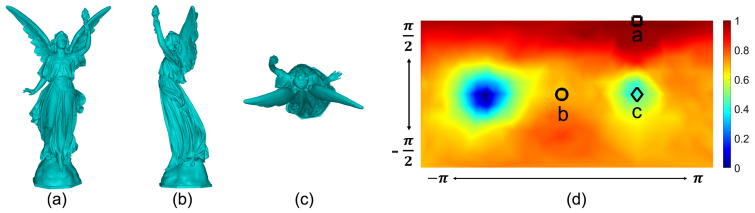

Fig. 2. Viewpoint saliency map. (a)–(c) are the projected views of the Lucy model. (d) is the viewpoint saliency map where the black square, circle and diamond mark the locations of the viewpoints corresponding to the views shown in (a)–(c) respectively.

3.4 Implementation

The proposed method is fully unsupervised. All we need to do is to pick an integer C for defining the output dimension of the Fc8/Fc9 layer.

We first render each 3D object as 20 views as described in Section 3.1 using a standard OpenGL renderer with perspective projection mode. The strengths of the ambient light, the diffuse light and the specular reflection are set to 0.2, 0.6 and 0.1 respectively. We apply flat shading to the meshed object. Using different illumination models or shading coefficients does not affect our method due to the invariance of the learned convolutional filters to illumination changes, as observed in image-based CNNs. All of the 20 views are then printed at 200 dpi, also in the OpenGL mode, and further resized to the resolution of 224×224. Then for training we feed these views into the UMVCNN wherein the convolutional layers and the first fully connected layer Fc6 are initialised with the weights pre-trained on ImageNet while other fully connected layers Fc7, Fc8 and Fc9 are all initialised with random weights using the popular method proposed in [11]. The UMVCNN is trained end-to-end by stochastic gradient descent with the learning rate of 10^{-5}. As we observed, the training always converged within 50 epochs for all of the variants of the UMVCNN that we shall discuss in Sections 4.4. When deploying the UMVCNN to select the salient view of a given 3D object, we again render the object as 20 views with the same rendering settings and then use the scheme described in Sections 3.3 to output the salient viewpoint.

4 Experimental results

In this section, we first introduce the datasets used in the experiments and evaluate our method qualitatively. Then, we show that our method can be directly used to select the salient view of a 3D scene to attract further interest. Finally, we evaluate both the proposed UMVCNN and its variants via quantitative comparisons for demonstrating its superiority and better understanding it.

4.1 Datasets

We create a new dataset containing 2747 3D models downloaded from the Princeton ModelNet dataset [32], the Schelling dataset [4] and the Trimble 3D Ware-

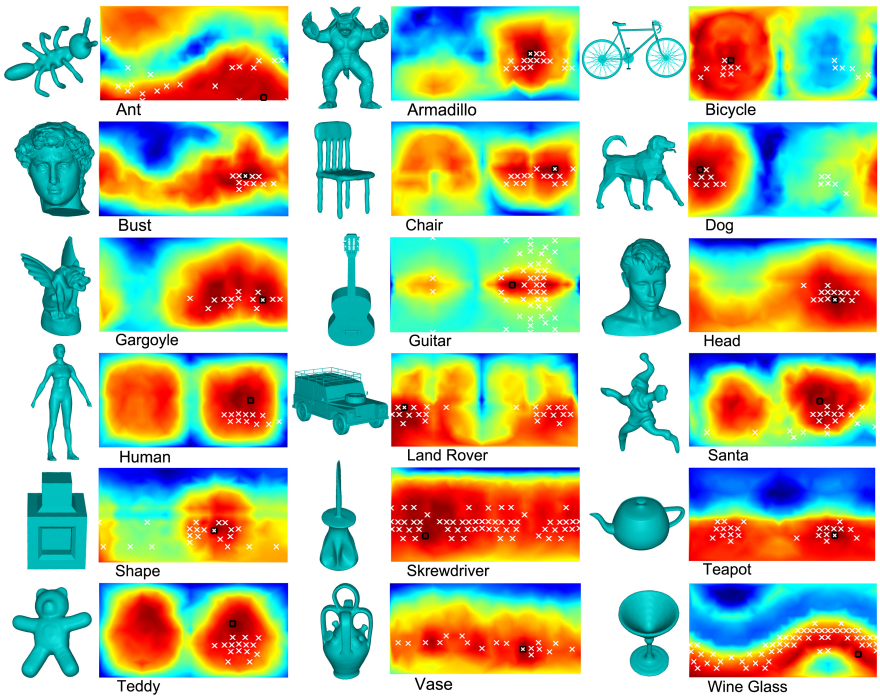

Fig. 3. Qualitative results of the salient views and the viewpoint saliency maps generated by our method. The black square corresponds to the salient viewpoints selected by our method. The white "X"s correspond to the ground truth best viewpoints picked by 26 human subjects (including their symmetric viewpoints) provided by [7].

house [30]. These models are originally from 30 object categories while in this work, all categorical annotations are removed in training and validation for an unsupervised learning. We use the same data split as in [32] where 80% of the objects in each category are used for training and 20% are used for validation.

We test our method on the Best View Selection benchmark [7] which might be the only publicly available benchmark suitable for quantitatively evaluating view selection methods. It contains 68 3D objects of various classes including some that do not belong to any of the 30 categories from the perspective of human recognition. It provides a quantitative benchmarking measure, the ground truth best viewpoints picked by 26 people and the results of 7 competing methods. We also used objects from the Stanford 3D Scanning Repository [5], the Princeton Shape Benchmark [24] and the Watertight Track of SHREC'07 for evaluations. Data and codes are available at `https://github.com/rsong/UMVCNN`.

4.2 Qualitative results

Fig. 3 shows our results for a variety of 3D objects with the ground truth best viewpoints supplied by [7]. The ground truth best viewpoints could be more

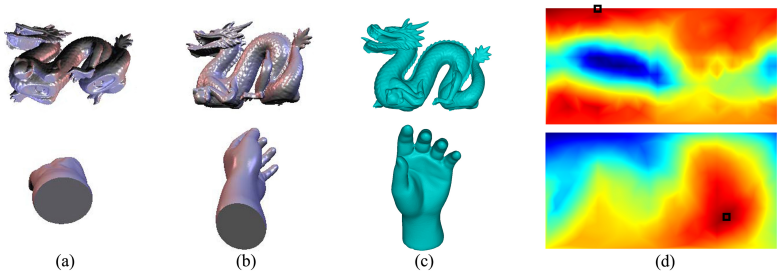

Fig. 4. Qualitative comparisons with [16] and [33]. (a) The best views selected by [16] (as implemented and shown in [33]). (b) The best views selected by [33]. (c) The best views selected by our method. (d) The viewpoint saliency maps generated by our method where the black squares mark the most salient viewpoints.

or less than 26 as 1) several participants could select the same viewpoint and 2) the symmetry of each object is taken into account and thus the symmetric viewpoints of those picked by the participants are also included. It can be seen that the consistency of human preferred viewpoints varies over different objects. Even though, most ground truth best viewpoints fall into the red or orange areas in the viewpoint saliency maps, which demonstrates that our method is good at predicting human's viewpoint preference over various objects. Also, for most objects, the salient viewpoint found by our method is, or at least very close to, a ground truth viewpoint picked by a participant. Due to the default distortion of the Mercator projection, for the Ant, the viewpoints on the bottom boundary of the viewpoint saliency map that look distant from each other are actually very close on the view sphere since they are both close to its bottom pole.

We next compare our method with some state-of-the-art methods. Since some of them require tuning of parameters and some are not open-sourced, we used our method to select salient views for the same objects used in the papers where the methods were reported. Fig. 4 compared our method with [16] and [33]. It can be seen that the our method is less influenced by some local geometric features such as the sharp edges at the bottom of the hand model if semantically they do not help the recognition of the object. Similarly, Fig. 5 shows that [17] chose a back view of the lamp containing many local details such as wires and screws. In comparison, for both the lamp and the jeep, our method tends to select views natural and good for recognising the objects. Fig. 5 also shows that our method outperforms [25] over a helicopter and a horse while more convincing quantitative comparisons using a variety of 3D objects are provided in Section 4.3. Note that [25] is essentially based on a weakly supervised deep learning framework where the class labels of the objects are available during training.

Since the UMVCNN does not rely on the knowledge about object classes, our method can be directly used to select the salient view of a 3D scene which usually contains objects of different classes and thus is unlikely to be reliably categorised in most datasets. According to the results shown in Fig. 6, our method successfully selects good views for various 3D scenes. The viewpoint saliency

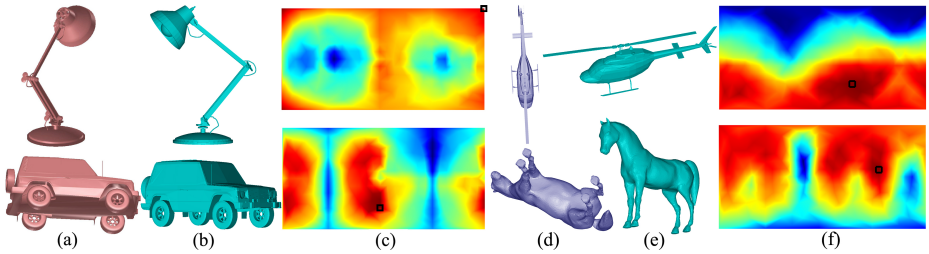

Fig. 5. Qualitative comparisons with [17, 25]. (a) and (d) The best views selected by [17] and [25] respectively. (b) and (e) The best views selected by our method. (c) and (f) The viewpoint saliency maps with black squares marking the most salient viewpoints.

maps of 3D scenes generated by our method are also informative. For instance, by observing the corresponding locations of the best and the worst views in the viewpoint saliency maps of most scenes, we find that the views with positive elevation angles are generally much more salient than those with negative ones, which is consistent with human's viewpoint preference. We also observed that the best view of a scene is not necessarily the best view of each individual object in it. For example, in the living room scene, the best view of the entire scene is not that of one of the three sofas. Similarly, in the work site scene, the best view of the scene is not that of the person in the middle and some chairs.

Please refer to the supplemental material for more qualitative results.

4.3 Quantitative results

We tested our method on the benchmark supplied by [7] which contains 68 objects using a computer with an Intel i7-4790 3.6GHz CPU and 32GB RAM without any GPU acceleration. The salient views of most objects can be computed within 1 minute where the vertex visibility to each viewpoint is precomputed.

Table 1 gives the statistics of the View Selection Error (VSE) of 9 automatic view selection methods over all of 68 objects. The VSE proposed by [7] measures the geodesic distance between the viewpoint found by a method and the ground truth supplied by a human subject on a unit view sphere and is averaged over the choices of all subjects, with the consideration of object-specific symmetry.

According to Table 1, our method yields the best performance in terms of the mean VSE, the median VSE and the number of objects for which a method gave the lowest VSE among all competing methods. Here UMVCNN-30 refers to the UMVCNN with C set to 30. As mentioned at the end of Section 3.2, this means that the output dimension of the Fc8 and Fc9 layers is set to 30 when we build the UMVCNN, which indicates that either of the view and the object channels categorises the objects into 30 classes. As shown in Fig. 3, due to the inconsistency of the ground truth choices of human subjects over the same object, reaching a zero mean VSE is impossible and improving the VSE is very challenging if it is already low. In most cases, a viewpoint with a mean VSE lower

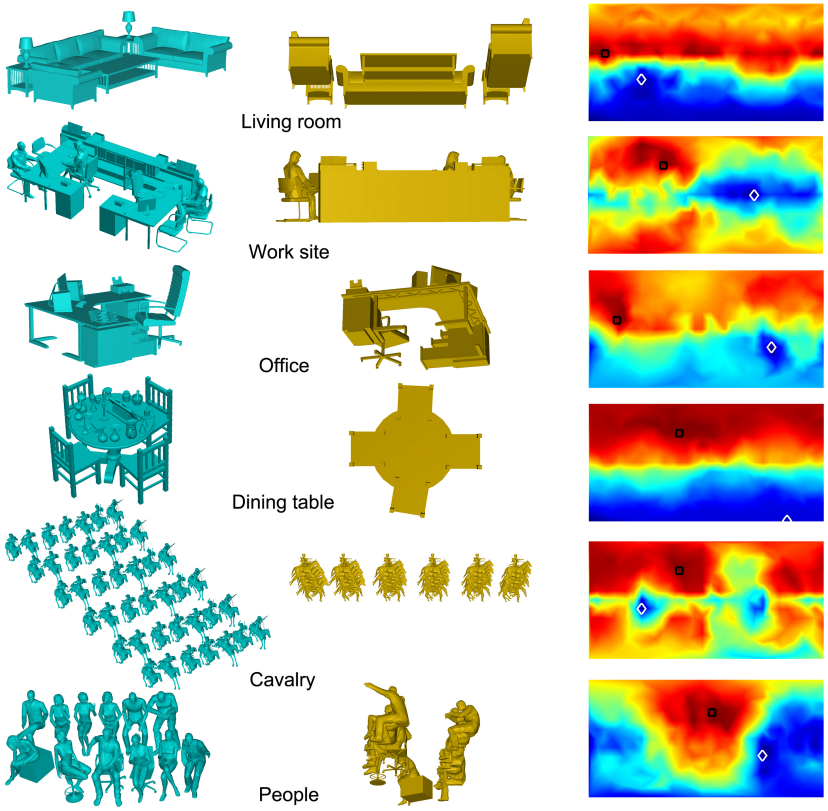

Fig. 6. Salient and non-salient views of 3D scenes (courtesy of Trimble 3D Warehouse [30]) selected by our method. Left column: the most salient views; Middle column: the least salient views; Right column: the viewpoint saliency maps where the black square marks the most salient view and the white diamond marks the least salient view.

than 0.3 corresponds to a good view. Even though, our method outperforms [25] by 3.4%, 2.9%, 4.6% and 24.8% in terms of the mean, the median, the standard deviation and the interquartile range of the VSE respectively. Note that their method is also based on deep learning but trained, in a weakly supervised manner, on a large dataset with the annotations of object class membership.

No method is consistently the best over all 68 objects although our method accomplishes the best results for 20 objects, the most over all competing methods. This is in agreement with the conclusions in [1, 23] which argued that human's view preference is driven by a variety of attributes. In general, the methods based on low-level attributes perform significantly worse than the two based on deep neural networks which learn high-level attributes of 3D objects.

In particular, Table 1 shows that our method significantly outperforms [7] based on view area and [21] based on silhouette length in terms of the VSE. This demonstrates that the improvement of the VSE does come from the UMVCNN

Table 1. Statistics of the VSE of 9 methods over 68 objects. SD and IQR represent the standard deviation and the interquartile range respectively. n gives the number of objects for which a method gave the lowest VSE among all competing methods.

View Selection Method	mean VSE	median VSE	SD of VSE	IQR of VSE	n
View area [7]	0.517	0.539	0.186	0.306	6
Ratio of visible area [21]	0.473	0.473	0.196	0.338	1
Surface area entropy [28]	0.396	0.386	0.144	0.195	8
Silhouette length [21]	0.446	0.445	0.172	0.275	7
Silhouette entropy [20]	0.484	0.469	0.153	0.241	5
Curvature entropy [20]	0.474	0.466	0.139	0.239	8
Mesh saliency [16]	0.430	0.395	0.165	0.233	2
Deep mesh distinction [25]	0.380	0.346	0.173	0.314	11
UMVCNN-30	0.367	0.336	0.165	0.236	20

Table 2. Mean View Selection Error of the variants of the UMVCNN over 68 objects

UMVCNN Variants	$C = 10$	$C = 15$	$C = 20$	$C = 25$	$C = 30$	$C = 30$, max-pooling	$C = 30$, 30 views	$C = 35$	$C = 40$
mean VSE	0.379	0.373	0.382	0.381	0.367	0.384	0.366	0.377	0.380

rather than the handcrafted features, i.e. view area and silhouette length that we use for the multi-view representation of a 3D object (see Section 3.1).

4.4 Evaluations over the variants of UMVCNN

Effect of varying C. Table 2 gives the mean VSE of the UMVCNN variants. It can be seen that redesigning the UMVCNN by varying C from 30 leads to an insignificant degradation of performance. As mentioned in Section 4.1, the 3D objects used for training are originally from 30 categories while we removed all categorical annotations for an unsupervised learning. Presumably, that $C = 30$ is indeed a good choice for the UMVCNN can be interpreted by the fact that salient view selection is highly related to classification as we observe that the objects of the same class tend to have analogous salient viewpoints while it is not the case the other way round. However, we cannot observe any obvious rule that suggests a way for deciding C. In a supervised learning, the network is forced to adopt the taxonomy of object classification consistent with human annotations while there is no guarantee that this taxonomy is optimal to the particular task such as salient view selection. Thus in different tasks, C might need to be tuned, but not necessarily fine-tuned as the UMVCNN is not very sensitive to it.

Ablation study for validating VD and WSP. We are interested in the heuristic component of the UMVCNN, i.e. the VD and the WSP layers. To validate its effectiveness, we replace the VD and the WSP layers with the popular element-wise max pooling which have demonstrated state-of-the-art performances in various 3D shape understanding tasks such as classification [26], retrieval [26] and segmentation [14]. The variant corresponds to '$C = 30$, max

Fig. 7. Limitation. Our method tends to select views good for recognition but not necessarily "natural". Left: the view selected by a subject; Middle: the view selected by our method; Right: the viewpoint saliency map where the diamond and the square mark the views selected by the subject and our method respectively.

pooling' in Table 2. To aggregate the multi-view 3D saliency maps H_i in Eq. (8), we set S_i to 1 as it is not available via this variant. Table 2 shows that the performance of the UMVCNN is significantly worse without the VD and the WSP layers. This demonstrates the effectiveness of the view distinction heuristic. It also suggests that the unsupervised learning based on the view-object consistency is likely to benefit from some heuristics introduced for the specific task.

Effect of the number of views. We tested the variant corresponding to '$C = 30$, 30 views' in Table 2 where a 3D object is projected into 30 instead of 20 views. All the other variants in Table 2 used the 20-view setup. It can be seen that using 30 views merely reduces the mean VSE slightly from 0.367 to 0.366. Using more or different views is trivial, however, we found that a 20-view setup is already enough to achieve high performance.

5 Conclusions

This work reveals that the view-object consistency is promising for the establishment of an unsupervised framework of 3D deep learning. We validate its effectiveness on the challenging task of salient view selection of 3D objects through the relatively naive design of a multi-view deep architecture. While the performance of our method is impressive, it has some limitations as shown in Fig. 7. Our method tends to select a view good for recognising the object, such as the view that better shows some features important for recognising the airplane (e.g. the wings and the engines). However, most subjects prefer a "natural" side view.

Future work will focus on implementing the unsupervised learning framework in more applications to demonstrate that it is amenable to a wide range of 3D shape understanding tasks. Particularly interesting applications might be some 3D scene understanding tasks hindered by the difficulty of collecting large amounts of accurately and consistently annotated data for training.

Acknowledgements. We acknowledge the support of the Young Taishan Scholars Program of Shandong Province tsqn20190929 and the Qilu Young Scholars Program of Shandong University 31400082063101, the National Natural Science Foundation of China under Grant 61991411 and U1913204, the National Key Research and Development Plan of China under Grant 2017YFB1300205, and the Shandong Major Scientific and Technological Innovation Project 2018CXGC1503.

References

1. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological review **94**(2), 115 (1987)
2. Blanz, V., Tarr, M.J., Bülthoff, H.H.: What object attributes determine canonical views? Perception **28**(5), 575–599 (1999)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: Proc. BMVC (2014)
4. Chen, X., Saparov, A., Pang, B., Funkhouser, T.: Schelling points on 3d surface meshes. ACM Trans. Graph. (Proc. SIGGRAPH) **31**(4), 29 (2012)
5. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proc. SIGGRAPH 1996. pp. 303–312 (1996)
6. Cutzu, F., Edelman, S.: Canonical views in object representation and recognition. Vision Research **34**(22), 3037–3056 (1994)
7. Dutagaci, H., Cheung, C.P., Godil, A.: A benchmark for best view selection of 3d objects. In: Proc. ACM workshop on 3DOR. pp. 45–50 (2010)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
9. Hayward, W.G.: Effects of outline shape in object recognition. Journal of experimental psychology: human perception and Performance **24**(2), 427 (1998)
10. He, J., Wang, L., Zhou, W., Zhang, H., Cui, X., Guo, Y.: Viewpoint assessment and recommendation for photographing architectures. IEEE Trans. Vis. Comput. Graph (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. ICCV. pp. 1026–1034 (2015)
12. Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V.G., Yumer, E.: Learning local shape descriptors from part correspondences with multiview convolutional networks. ACM Trans. Graph. **37**(1), 6 (2018)
13. Kalogerakis, E., Averkiou, M., Maji, S., Chaudhuri, S.: 3D shape segmentation with projective convolutional networks. In: Proc. CVPR. vol. 1, p. 8 (2017)
14. Kim, S.h., Tai, Y.W., Lee, J.Y., Park, J., Kweon, I.S.: Category-specific salient view selection via deep convolutional neural networks. In: Computer Graphics Forum. vol. 36, pp. 313–328. Wiley Online Library (2017)
15. Koch, C., Poggio, T.: Predicting the visual world: silence is golden. Nat. Neurosci. **2**, 9–10 (1999)
16. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. ACM Trans. Graph. (Proc. SIGGRAPH) **24**(3), 659–666 (2005)
17. Leifman, G., Shtrom, E., Tal, A.: Surface regions of interest for viewpoint selection. IEEE Trans. Pattern Anal. Mach. Intell **38**(12), 2544–2556 (2016)
18. Mezuman, E., Weiss, Y.: Learning about canonical views from internet image collections. In: Proc. NIPS. pp. 719–727 (2012)
19. Novotny, D., Larlus, D., Vedaldi, A.: Learning 3d object categories by looking around them. In: Proc. ICCV (Oct 2017)
20. Page, D.L., Koschan, A.F., Sukumar, S.R., Roui-Abidi, B., Abidi, M.A.: Shape analysis algorithm based on information theory. In: Proc. ICIP. vol. 1, pp. I–229 (2003)
21. Polonsky, O., Patané, G., Biasotti, S., Gotsman, C., Spagnuolo, M.: What's in an image? The Visual Computer **21**(8-10), 840–847 (2005)

22. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.: Volumetric and multi-view cnns for object classification on 3d data. In: Proc. CVPR. pp. 5648–5656 (2016)
23. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Trans. Graph. **30**(5), 109 (2011)
24. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Proceedings of Shape modeling applications (2004)
25. Song, R., Liu, Y., Rosin, P.L.: Distinction of 3D objects and scenes via classification network and markov random field. IEEE Trans. Vis. Comput. Graph. (2018)
26. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3d shape recognition. In: Proc. ICCV. pp. 945–953 (2015)
27. Tarr, M.J., Pinker, S.: Mental rotation and orientation-dependence in shape recognition. Cognitive psychology **21**(2), 233–282 (1989)
28. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: VMV. vol. 1, pp. 273–280 (2001)
29. Vieira, T., Bordignon, A., Peixoto, A., Tavares, G., Lopes, H., Velho, L., Lewiner, T.: Learning good views through intelligent galleries. In: Computer Graphics Forum. vol. 28, pp. 717–726. Wiley Online Library (2009)
30. Warehouse, D.: https://3dwarehouse.sketchup.com
31. Wolfe, J.M.: Guided search 2.0 a revised model of visual search. Psychonomic Bulletin & Review **1**(2), 202–238 (1994)
32. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D shapenets: A deep representation for volumetric shapes. In: Proc. CVPR. pp. 1912–1920 (2015)
33. Yamauchi, H., Saleem, W., Yoshizawa, S., Karni, Z., Belyaev, A., Seidel, H.P.: Towards stable and salient multi-view representation of 3d shapes. In: IEEE International Conference on Shape Modeling and Applications (2006)
34. Zhao, S., Ooi, W.T.: Modeling 3d synthetic view dissimilarity. The Visual Computer **32**(4), 429–443 (2016)