3D Indoor Scene Classification using Tri-projection Voxel Splatting

Kazuma Hamada* and Masaki Aono†
Computer Science and Engineering, Toyohashi University of Technology, Aichi, Japan
E-mail: *k-hamada@kde.cs.tut.ac.jp, †aono@tut.jp

***Abstract*—In this paper, we propose a new method for classifying a 3D scene. A 3D scene consists of a large collection of 3D shape objects. The complexity of a 3D scene makes it hard for us to classify which 3D scene we are dealing with. For instance, a 3D scene of a "room" may be an office, a kitchen, a living room, or a bed room. Here we propose a novel approach to classifying a 3D scene with Tri-projection Voxel Splatting (TVS), taking into account the voxel density along the depth direction. In TVS we first normalize a 3D scene in terms of position and size, followed by converting data into point clouds, via voxelization, and by projecting the scene on three perpendicular planes, reflecting the voxel density along the depth direction. Subsequently, we merge the three projected images, and we finally apply deep learning to predict the class of each 3D scene. To demonstrate the effectiveness of our proposed method (TVS), we conducted experiments with 3D indoor scene dataset extracted from Princeton University's SUNCG dataset. From the experiments, our proposed method outperformed the previous methods.**

I. Introduction

With the spread of VR technology, demand for applications using scenes (3D scenes) composed of a collection of 3D objects has been increasing. In recent years 3D scenes have been utilized not only for games and movies but also for educational systems and real estate businesses. It is expected that 3D scenes will be used for various potential applications in the future. Research on creating 3D scenes has begun [18]. Accordingly, more and more 3D scenes have spread and are available on the Internet.

Given a large amount of 3D scenes, it is highly important to automatically recognize what kind of 3D scenes they are. For instance, a 3D scene might be a "kitchen," it might be a "living room," or it might be "bathroom." This is where automatic labeling technology to an arbitrary 3D scene plays an important role. Indeed, in 2018, SHREC (Shape Retrieval Contest) has organized a track with 3D scenes retrieval [1][27], which demonstrates that 3D scene recognition is of interest academically.

Deep learning has attracted attention in recent years, and research using deep learning has been popular in classification and retrieval of 3D data. Quite a few researchers have begun using deep learning to classify and retrieve 3D data. However, to our knowledge, no previous methods have dealt with a 3D scene consisting of a collection of 3D objects when applying deep learning.

In this paper, we propose Tri-projection Voxel Splatting (TVS), which is a novel method for generating images that help to classify 3D scenes based on the depth density of 3D scenes via voxel representation. In addition, we describe indoor 3D scene classification applying deep learning to images generated by TVS.

In the following, we first survey related work in Section II, followed by introducing TVS in Section III. In Section IV, we describe experiments we have carried out, and conclude this paper in Section V.

II. Related Work

The classification of scenes represented by still pictures such as "kitchen" and "bedroom" has been studied for years by many researchers, where they employed hand-crafted features [13][16][30]. Recently end-to-end features with DNN has also been popular [7]. On the other hand, along with the spread of RGB-D cameras, research has also progressed, aiming at classifying scenes with RGB-D data [6][20].

3D shape classification and retrieval research has focused on feature extraction. Examples of 3D shape feature extraction include a method of utilizing histograms based on global features (e.g. [15]) and a method of computing feature vectors with multi-view rendering [28].

With the popularity of deep learning, research on 3D shape classification as well as 3D shape search has adopted deep learning approaches. Voxel-based methods include 3DShapeNets of the method applying Convolutional Deep Belief Network to 3D data [26], VoxNet of the method applying 3DCNN [14], OctNet of the method using octree space partitioning structures [17]. There is also a method [2] that applies deep networks such as ResNet [8] to 3D data. Image-based methods include MVCNN of the method learning CNN with multi-view depth images [23] and DeepPano of the method learning CNN with panoramic view images [19]. Point cloud-based methods include PointNet [3]. They have high classification performance for a single 3D object. However, since 3D scene has a property that it is larger in scale as compared with a single 3D object, contains a plurality of different objects, and is arranged in various layouts, it is unclear whether it is effective for classification of 3D scenes.

In this research, we describe a new 3D scene classification method as described in Section III.

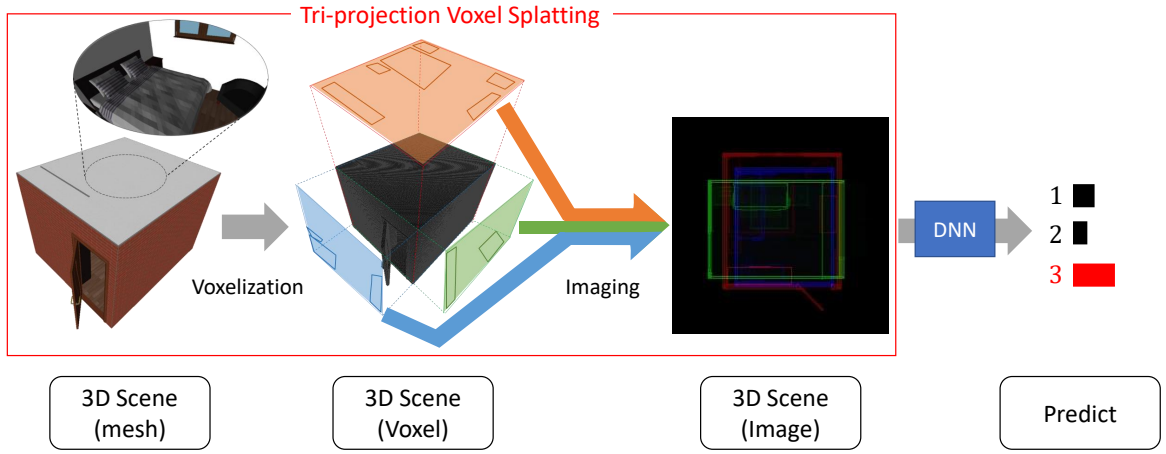

Fig. 1. Flow of 3D scene classification with TVS

III. Tri-projection Voxel Splatting

A. Overview of 3D scene classification with TVS

Fig. 1 illustrates the overall flow of our 3D scene classification with our proposed Tri-projection Voxel Splatting (TVS). First, the position and size of the 3D scene is normalized to generate voxel data. A voxel represents a value on a regular grid in three-dimensional space. In TVS, a shape is represented by binary voxels, where each voxel is expressing the presence or absence of an object with a value of 0 or 1. Next, the 3D scene is converted into voxel data by means of "imaging" with TVS, which will be elaborated later. Then, we train the DNN with images generated by TVS. Finally we classify 3D scenes using the trained DNN.

B. Normalization of 3D scene and voxelization

In this research, we deal with 3D scenes made of multiple objects defined by 3D meshes.

TVS needs to normalize the position and size of the 3D scene as preprocessing for voxelization. First, we apply normalization of the position by translating the 3D scene so that the center of gravity becomes the origin. Then, we apply normalization of the size, taking the division of the value of each vertex coordinate by the maximum distance from the origin to the vertex.

We apply voxelization of the normalized 3D scene. TVS first converts 3D scenes into point clouds by creating points on Osada's method [15] on each side of 3D scene. Then, by quantizing the coordinate value of each point in conformity with the size of the voxel expression, point clouds convert to voxel representation.

C. Imaging

First, we generate a map expressing the depth density of the voxel data on the projection plane with the x, y and z axes as the depth (Fig. 2). For the sake of clarity, the size of voxel representation is $4\times4\times4$ in Fig. 2. YZ is the projection surface whose x axis is the depth, XZ is the projection surface whose y axis is the depth, assuming that the projection surface with the depth of z axis is XY, the map M generated on each projection plane is expressed as follows:

$$M_{YZ} = \begin{pmatrix} p_{YZ(1,1)} & p_{YZ(2,1)} & \cdots & p_{YZ(N,1)} \\ p_{YZ(1,2)} & p_{YZ(2,2)} & \cdots & p_{YZ(N,2)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{YZ(1,N)} & p_{YZ(2,N)} & \cdots & p_{YZ(N,N)} \end{pmatrix}$$

$$M_{XZ} = \begin{pmatrix} p_{XZ(1,1)} & p_{XZ(2,1)} & \cdots & p_{XZ(N,1)} \\ p_{XZ(1,2)} & p_{XZ(2,2)} & \cdots & p_{XZ(N,2)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{XZ(1,N)} & p_{XZ(2,N)} & \cdots & p_{XZ(N,N)} \end{pmatrix}$$

$$M_{XY} = \begin{pmatrix} p_{XY(1,1)} & p_{XY(2,1)} & \cdots & p_{XY(N,1)} \\ p_{XY(1,2)} & p_{XY(2,2)} & \cdots & p_{XY(N,2)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{XY(1,N)} & p_{XY(2,N)} & \cdots & p_{XY(N,N)} \end{pmatrix}$$

where, p represents the pixel value of each coordinate in map M, and N represents the resolution of projected images from voxels. Assuming that the abscissa of the projection plane is i and the ordinate is j, the pixel value p is expressed as follows:

$$p_{YZ(i,j)} = \mathrm{ceil}(\frac{255}{N}\sum_{x=1}^{N} v_{(x,i,j)})$$

$$p_{XZ(i,j)} = \mathrm{ceil}(\frac{255}{N}\sum_{y=1}^{N} v_{(i,y,j)})$$

$$p_{XY(i,j)} = \mathrm{ceil}(\frac{255}{N}\sum_{z=1}^{N} v_{(i,j,z)})$$

where, $v_{(x,y,z)}$ is the voxel value of 1 (object exists) or 0 (empty). Next, each map M generated on the three projection

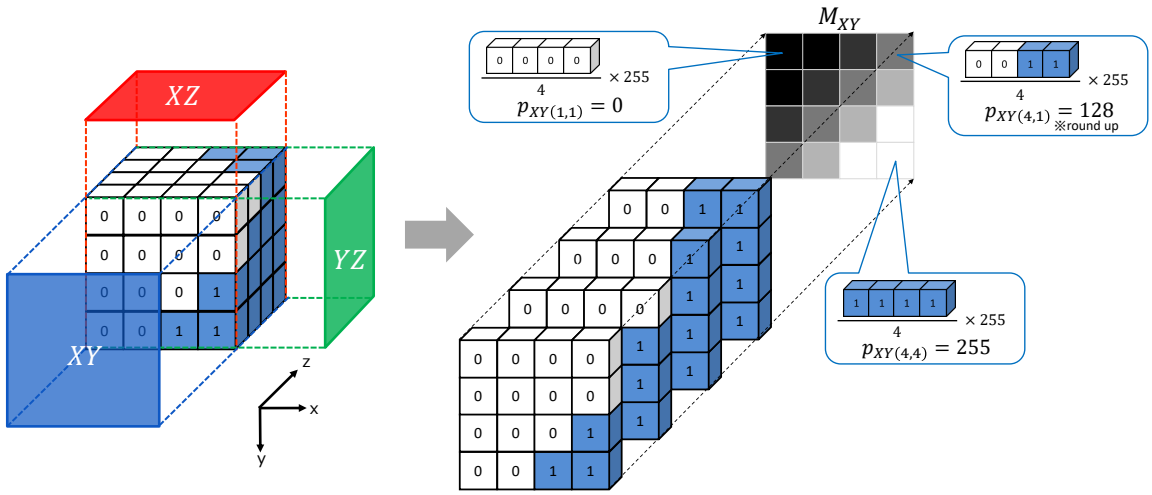

Fig. 2. Illustration of how a map is generated by considering the density along the depth axis. The figure on the right is an example of a projection plane with the z axis as the depth. In this example, $p_{XY(1,1)} = 0, p_{XY(4,1)} = 128, p_{XY(4,4)} = 255$.

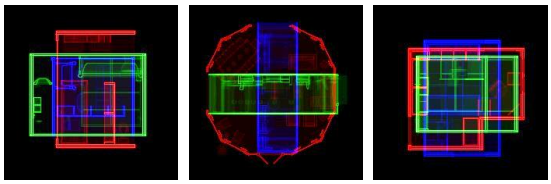

Fig. 3. An example of the images generated by TVS. For the sake of clarity, we show brighter images here than those of the original.

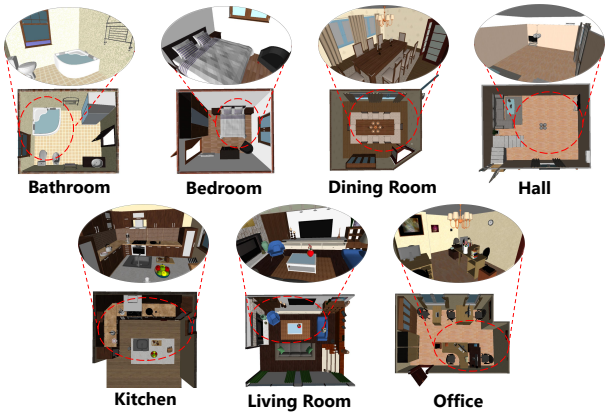

Fig. 4. 7 categories of 3D indoor scene benchmark dataset

planes is fitted to 3ch (R, G, B) of the color image, and it is combined into one image. Finally, the image I generated by TVS is expressed as follows:

$$I = \{M_{YZ}, M_{XZ}, M_{XY}\}$$

Fig. 3 is an example of the images generated by TVS.

IV. Experiments

In this section, we first describe the 3D indoor scene dataset we have used for experiments, followed by a DNN architecture comparison. We then describe the comparison of our proposed method with previous methods. Finally, we describe the results of the comparisons, respectively.

A. Dataset

In this experiment, we have divided the 3D house scene published in the SUNCG dataset [22] into rooms. We removed scenes of 10 or fewer elements constituting a scene such as a floor and a desk. Among the rooms after noise removal, we extract 7 categories (Bathroom, Bedroom, Dining Room, Hall, Kitchen, Living Room, Office), each of which is divided into 3,800 pieces of training data and 1,000 pieces of test data, taken from the benchmark dataset of the indoor 3D scene. The total number of training data is 26,600 and the total number of test data is 7,000. Fig. 4 shows an example of data included in the benchmark dataset.

B. DNN Architecture Comparison

We conducted experiments to see which DNN architecture best fitted TVS. Specifically, we compared 9 different neural network architectures including AlexNet [12], ResNet50 [8], Xception [4]. We adopted the method with the highest accuracy as the proposed method. Next, we compared our proposed method with the previous methods such as VoxNet [14] and PointNet [3].

In the training of the model, the size of the input image was unified with 224×224 and 80 epoch learning was done. At this time, Adam [11] was used as an optimization algorithm, and the learning rate was set to 0.01. Cross entropy was applied

to the loss function during training.

$$E = -\sum_{k} x_k \log(y_k)$$

Data augmentation was performed on training data and the training was performed using extended data. In this experiment, we generated 80 pieces from one image by randomly shifting horizontally and vertically in the range of 10% the length of the image, and in addition, randomly inverted it in the horizontal direction.

C. Comparison with Previous Methods

In the previous methods for comparison, VoxNet [14] as a voxel-based method, MVCNN [23] as an image-based method, PointNet [3] as a point cloud-based method were chosen. VoxNet [14] converts 3D data to voxel representation and accepts 3DCNN as input. In the experiment, we set the input size of VoxNet to $64 \times 64 \times 64$. MVCNN [23] is a method for extracting the features from the multi-view depth images with the trained CNN, integrating all the views with the view-pooling layer. In the experiment, we set the depth image rendered in 18 directions from the center of gravity of the 3D scene for the MVCNN. For the CNN of feature extraction from the MVCNN, we applied the pre-trained model with ImageNet [5] and used the features extracted from the second to the last layer. At this time, the model to be applied is the same as the model used in the proposed method. PointNet [3] is a method that learns by inputting 3D data represented by a point cloud as an input. In the experiment, we used the method of Osada et al. [15] to represent the 3D scene of PointNet expressed by 2,048 points as input.

D. Results

1) Result of DNN Architecture Comparison: The result of comparison among different neural network architectures is shown in TABLE I. TABLE I is a summary of the F-measures of the 7 categories with the average F measures across all the categories. In TABLE I, the largest value in each category is indicated in bold. From TABLE I, it is confirmed that among the compared models, Xception [4] has the highest accuracy in all categories. On the other hand, it is confirmed that the learning is not well performed on AlexNet [12] and VGG19 [21] with relatively few layers. We speculate that the DNN with fewer layers is not appropriate for classifying 3D scenes with TVS.

2) Result of Comparison of Our Proposed Method with Previous Methods: The results of experiments comparing our method with the previous methods are shown in TABLE II. From TABLE II, we can confirm that the proposed method outperforms the previous methods in all categories. It is observed that VoxNet [14] cannot learn 3D scenes, yielding single class collapse.

With the 3D scene voxelize whose is a feasible size for training, the shape information of each object in a 3D scene is likely to be lost . Therefore, it is presumed that the method of directly inputting voxels such as with VoxNet [14], into the classification of 3D scenes is not suitable.

Since the proposed method maintains detailed shape information of the objects to be kept in the voxels, which are converted to the image by TVS, it is possible to train the DNN.

An example of successful classification of the proposed method is shown in Fig. 5. In Fig. 5, the matched categories between the predicted and the ground truth are denoted by bold letters such as **Bedroom** and **Office**.

From Fig. 5, we have confirmed the proposed method correctly classified 3D scenes that could not be properly classified by previous methods. In addition, we have confirmed that the images generated by TVS emphasize the shape of 3D scenes and objects in the scene such as chairs and shelves.

When the training is performed using these images as input, we speculate that the DNN learns the shape of the 3D scene and the tendency of the type and the number of the objects through the contours of the objects. As far as our experiment goes, since there is no tendency peculiar to categories in the outline of the 3D scene, we believe that it is important to observe the tendency of the type and the number of objects in order to recognize the 3D scene.

Since MVCNN [23] extracts the features for each image for multi-view depth images, and performs pooling, it may fail to capture the tendency of the number of objects. PointNet [3] can grasp the presence or absence of an object by looking at the density of point clouds, but it may fail to specify the type of the object included in the scene if the object is represented by a sparse point. On the other hand, our proposed method grasps the tendency of the type and the number of objects by training the images generated by TVS, so that it is possible to accurately classify 3D scenes where previous methods fail to do so.

3) Result of Comparison in terms of Category: Fig. 6 shows the confusion matrix of our proposed method. From Fig. 6 it is observed that the misclassification of Bathroom and Bedroom categories is small. This is because there are specific objects such as bathtubs in Bathrooms and beds in Bedrooms. On the other hand, there are observed a few misclassifications of Hall and Living Room. This is because the tendency of the type of the object is similar and the scene categories are determined by the position of the object, it is necessary to grasp not only the kind of object but also the position of the object.

V. Conclusion

In this paper, we proposed Tri-projection Voxel Splatting (TVS), which is a novel method for generating images that help to classify 3D scenes based on the depth density of 3D scenes via voxel representation. In addition, we described indoor 3D scene classification applying deep learning to images generated by TVS.

We conducted classification experiments with 3D indoor scene dataset extracted from SUNCG dataset [22], where the problem was formulated by a 7 class classification problem. As

TABLE I
COMPARISON RESULT WITH DIFFERENT DNN ARCHITECTURES

Method	F-measure							
	Bathroom	Bedroom	Dining Room	Hall	Kitchen	Living Room	Office	Average
AlexNet [12]	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.04
VGG19 [21]	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.04
ResNet50 [8]	0.91	0.89	0.76	0.62	0.84	0.69	0.79	0.79
InceptionV3 [25]	**0.92**	0.90	0.77	0.63	0.83	0.72	0.81	0.80
Inception ResNetV2 [24]	0.90	0.89	0.77	0.62	0.83	0.71	0.80	0.78
Xception [4]	**0.92**	**0.91**	**0.79**	**0.65**	**0.85**	**0.74**	**0.83**	**0.81**
MobileNet [9]	0.90	0.89	0.75	0.64	0.83	0.71	0.78	0.79
DenseNet201 [10]	0.90	0.88	0.76	0.59	**0.85**	0.72	0.80	0.79
NASNet (Mobile) [29]	0.88	0.86	0.75	0.60	0.83	0.69	0.77	0.77

TABLE II
COMPARISON RESULT WITH PREVIOUS METHODS

Method	F-measure							
	Bathroom	Bedroom	Dining Room	Hall	Kitchen	Living Room	Office	Average
VoxNet [14]	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.04
MVCNN [23]	0.80	0.70	0.70	0.48	0.73	0.60	0.57	0.65
PointNet [3]	0.78	0.70	0.58	0.45	0.82	0.72	0.47	0.65
Proposed (TVS + Xception [4])	**0.92**	**0.91**	**0.79**	**0.65**	**0.85**	**0.74**	**0.83**	**0.81**

the result, our proposed TVS method outperformed previous methods. Specifically, on the average TVS outperformed the best previous method by 16%.

As far as our experiments, TVS can consider objects inside each 3D scene such as bed, bathtub, and desk. However, since TVS needs normalization to fit each 3D scene into a voxel, if the 3D scene becomes large enough, TVS might fail to recognize tiny objects. In the future, we plan to investigate the above issue with 3D outdoor scenes.

ACKNOWLEDGMENT

A part of this research was carried out with the support of the Grant-in-Aid for Scientific Research (B) (issue number 17H01746).

REFERENCES

[1] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh-Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. 2D Image-Based 3D Scene Retrieval. In Alex Telea, Theoharis Theoharis, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2018.

[2] Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J. Weston. Generative and discriminative voxel modeling with convolutional neural networks. pages 1–9, 12 2016. Workshop contribution; Neural Inofrmation Processing Conference : 3D Deep Learning, NIPS ; Conference date: 05-12-2016 Through 10-12-2016.

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

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[5] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[6] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, June 2013.

[7] M. Hayat, S. H. Khan, M. Bennamoun, and S. An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, Oct 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[9] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[10] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 12 2014.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

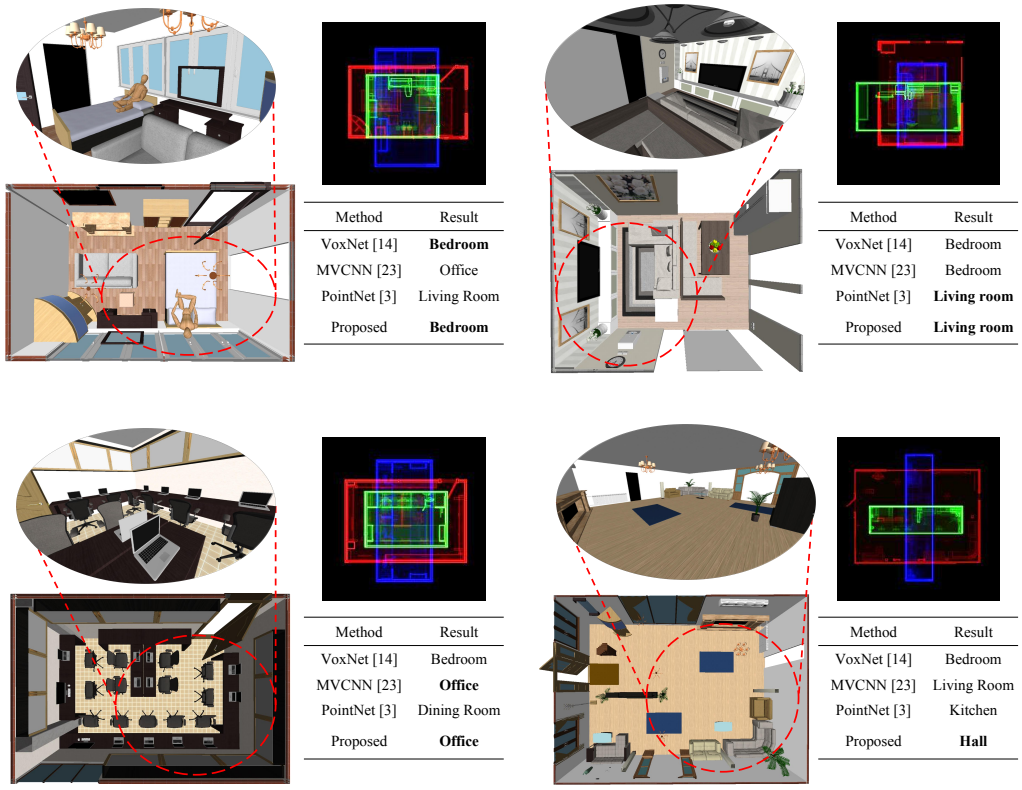

Fig. 5. An example of successful classification of the proposed method. The matched categories between the predicted and the ground truth are denoted in bold. The upper right figure of each example is an image generated by TVS based on the 3D scene of a classification example. For the sake of clarity, we show brighter images here than those of the original image generated by TVS.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006.

[14] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, Sept 2015.

[15] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, October 2002.

[16] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, June 2009.

[17] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Manolis Savva, Angel X. Chang, and Maneesh Agrawala. Scenesuggest: Context-driven 3d scene design. *CoRR*, abs/1703.00061, 2017.

[19] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. 22:2339–2343, 12 2015.

[20] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 945–953, Washington, DC, USA, 2015. IEEE Computer Society.

[24] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[26] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015.

[27] Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh-Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. 2D Scene Sketch-Based 3D Scene Retrieval. In Alex Telea, Theoharis

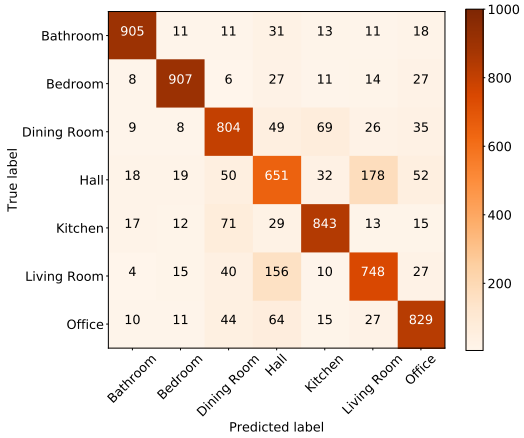

Fig. 6. Confusion matrix of our proposed method

Theoharis, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2018.

[28] Chen Ding Yun, Tian Xiao Pei, Shen Yu Te, and Ouhyoung Ming. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232.

[29] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.

[30] Zhen Zuo, Gang Wang, Bing Shuai, Lifan Zhao, Qingxiong Yang, and Xudong Jiang. Learning discriminative and shareable features for scene classification. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 552–568, Cham, 2014. Springer International Publishing.