15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

3D Convolutional Object Recognition using Volumetric Representations of Depth Data

Ali Caglayan, Ahmet Burak Can
Department of Computer Engineering
Hacettepe University, Ankara, Turkey 06800
{alicaglayan, abc}@cs.hacettepe.edu.tr

Abstract

Hand-crafted features are widely used in object recognition field. Recent advances in convolutional neural networks allow to extract features automatically and produce better results in object recognition without considering about feature design. Although RGB and depth data are used in some convolutional network based approaches, volumetric information hidden in depth data is not fully utilized. We present a 3D convolutional neural network based approach to utilize volumetric information extracted from depth data. Using a single depth image, a view-based incomplete 3D model is constructed. Although this method does not provide enough information to build a complete 3D model, it is still useful to recognize objects. To the best of our knowledge, the proposed approach is the first volumetric study on the Washington RGB-D Object Dataset and achieves results as competitive as the state-of-the-art works.

1 Introduction

Object recognition is one of the most fundamental problems in machine vision. Recognition systems deployed for particular entities, e.g. fingerprint, iris, optical character, license plate, and traffic sign, widely used in daily applications but recognising various types of objects is still a difficult task. Especially in the field of robotics, this kind of intelligence is needed to increase robot's interaction with the real world. Object recognition is a challenging task because (i) the same object class may contain many different instance types (intra-class variation) (ii) different object classes may form similar instance types (iii) the general challenges from the nature of the problem exist, such as environmental illumination challenges, viewpoint and scale variations, noise and distortions in the images.

Until recently, object recognition tasks were based on hand-crafted feature extraction. However, this kind of approach requires field expertise and also lacks the generic models that can be reused. The new trend in Convolutional Neural Networks (CNNs) [1] presents the ability of automatic feature learning and increases efficiency of recognition systems significantly. Since depth information provides relatively invariant information on color, illumination, and viewpoint changes, there has been increasing interest in object recognition using depth data after the invention of low-cost RGB-D sensors such as the Microsoft Kinect. In most of existing research efforts, depth data is used as an extra channel in addition to the RGB channels (e.g. [2, 3, 4]). However, the characteristics of RGB and depth images are different. While RGB data provides color and rich texture information, depth data has better ability of representing 3D structures of objects. Therefore, instead of using depth data as an additional channel, it would be better to use depth data to extract geometric structures of objects.

In this work, we propose a 3D CNN based approach to exploit 3D geometrical cues of objects using depth data. Two types of volumetric representations are constructed from depth images and objects are recognized using only depth data. Unlike the studies that use a complete 3D representation of objects [5, 6], the proposed approach is based on view-based incomplete 3D representations to recognize objects. Since these volumetric representations are constructed from only depth images, the approach can easily be used with an RGB-D sensor. An object can be recognized using only a single depth image without having a complete 3D model of the object. In summary, our contributions in this paper are: (i) We introduce two elegant and effective volumetric representations. (ii) We experimentally show that 3D CNNs are ingenious enough to learn objects from incomplete 3D object representations. (iii) To the best of our knowledge, this work is the first volumetric representation on the commonly used Washington RGB-D Object Dataset [7] and outperforms most state-of-the-art algorithms on this dataset.

2 Related Work

With the advent of affordable RGB-D sensors, an increasing number of papers have focused on object recognition using depth images [2, 3, 4, 5, 6, 7, 8, 9, 10]. Most of these approaches use depth data as an additional channel to the RGB channels [2, 3, 4, 7, 8, 9, 10], which are considered as 2.5D recognition approaches. Socher et al. [2] propose a convolutional-recursive neural network model (CNN-RNN) which learns color and depth features separately and then combines them for the softmax classifier. In [3], depth kernel descriptors are proposed to capture size, shape, edges, pixel orientations in a unified way. Hierarchical kernel descriptors [4] extend [3] in a layer-wise fashion. Both [4] and [3] are based on hand-crafted feature extraction. Cheng et al. [8] propose a semi-supervised learning method in which they use CNN-RNN model [2] to construct RGB and depth features along with a co-training algorithm to make use of unlabeled data. In [9], the authors extend this work by considering grayscale images and surface normals in addition to the RGB and depth images. They also adapt CNN-RNN model [2] for arbitrary image sizes by replacing the first step of CNN-RNN with a spatial pyramid pooling layer. In [10], a non-automatic subset based patch extraction for convolutional feature learning is presented. In [11], sparse coding is used to learn hierarchical feature representations of RGB-D data in an unsupervised way. More recently, an interesting algorithm is proposed by Zaki et al. [12]. The authors embed depth data and point cloud data into the RGB domain to allow knowledge transfer from a pre-trained CNN model. Thus, their method is taking advantage of large annotated datasets like ImageNet [13]. They also use hypercube pyramids to encode locally-activated features in the earlier CNN layers.

Volumetric approaches have started with the introduction of the 3D ShapeNets [5] which represents 3D shapes as a probability distribution of binary voxels. Maturana et al. [6] take ShapeNets one step further by reducing the number of model parameters up to *12x* and increasing the classification accuracy significantly. Inspired by these works, we present a 3D CNN model on the Washington RGB-D Object Dataset [7] which is one of the most used benchmarks. Both [5] and [6] are built on using the full sphere of viewpoints over an object whereas our approach is based on single view of an object. Furthermore, unlike our work, they represent 3D models as a probabilistic approach of spatial occupancy on voxel grid maps. While ShapeNets [5] and VoxNet [6] augment the dataset by copying each input instance rotated around z axis (i.e. 12 poses per model for ShapeNets and 12 or 18 poses per model for VoxNet) to acquire complete 3D object models, our method is trained on view-based incomplete models but still successful results are achieved.

3 Method

The proposed method starts with a preprocessing step to increase the data quality. The input of our pipeline is a raw depth image. The raw depth data obtained with the Kinect is noisy and has missing depth values (holes) due to reasons such as reflections, transparency of surfaces, etc [14]. We apply an iterative process to fill out the missing zero values with the mean of 5×5 pixel neighbourhood of the target value. After converting depth image to the point cloud data, we apply the denoising method in [15] to remove noise from the point cloud. The overview of the proposed method is illustrated in Fig. 1.

3.1 Volumetric Representation

The depth data is generally used as an additional channel in the literature (e.g. [2, 3]). Since the characteristics of RGB and depth images are different, geometric structure information hidden in depth data may not be fully utilized with this approach. On the other hand, volumetric representations have advantages in CNN architectures such as simplicity, convenience to convolutional approaches and good representation of 3D geometrical cues. Within this context, we propose two elegant and effective volumetric representations. Depth images are converted to point cloud data. Then, our volumetric representations are constructed based on projection of point cloud data to 3D matrix space in which each cell represents a voxel. There is no need to resize the depth images since the projection operation does not require equal image sizes. Therefore, the use of volumetric representations prevents potential performance degradation by cropping and warping input images.

3.1.1 Volumetric Binary Grid

Binary grid represents the existence of a surface point in a voxel. *1* indicates the presence of a point whereas *0* specifies the absence. For a given $m \times 3$ point cloud data where m denotes the number of points; $X = \{x_1, x_2, ..., x_m\}$, $Y = \{y_1, y_2, ..., y_m\}$, and $Z = \{z_1, z_2, ..., z_m\}$ are the column vectors of all the values of x-axis, y-axis and z-axis respectively. Then the transformation is done as follows:

$$\begin{aligned} X^{'} &= \left(\frac{X - x_{min}}{(x_{max} - x_{min}) + \epsilon} \right) (l_{max} - l_{min}) + l_{min} \\ Y^{'} &= \left(\frac{Y - y_{min}}{(y_{max} - y_{min}) + \epsilon} \right) (l_{max} - l_{min}) + l_{min} \\ Z^{'} &= \left(\frac{Z - z_{min}}{(z_{max} - z_{min}) + \epsilon} \right) (l_{max} - l_{min}) + l_{min} \end{aligned} \tag{1}$$

Where $X^{'}$, $Y^{'}$, $Z^{'}$ are the projection vectors corresponding to X, Y, Z point cloud data; (x_{max}, x_{min}), (y_{max}, y_{min}) and (z_{max}, z_{min}) are the maximum and minimum pair values in X, Y, Z vectors respectively; l_{max} and l_{min} are maximum and minimum projection values. In our case, $l_{max} = 30$ and $l_{min} = 1$. The small constant value $\epsilon \approx 0$ in denominator is to prevent division by zero in the case when max and min values are equal. Then, the values in $X^{'}$, $Y^{'}$, $Z^{'}$ vectors are rounded to closest integer value in order to obtain discrete values between l_{min} and l_{max}. Finally, for a given (x'_k, y'_k, z'_k) voxel in the grid, volumetric binary value is assigned as follows:

$$(x'_k, y'_k, z'_k) = \begin{cases} 1, & \text{if } x'_k \in X', y'_k \in Y', z'_k \in Z' \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Where $k = \{1, 2, ..., m\}$ and since the values are scaled and rounded, $x'_k = \{l_{min}, ..., l_{max}\}$, $y'_k = \{l_{min}, ..., l_{max}\}$, $z'_k = \{l_{min}, ..., l_{max}\}$. Last to mention, as shown in the equations (1), the column vectors X, Y and Z are calculated separately within themselves. This allows voxels to maintain the relative positions to each other.

3.1.2 Volumetric Intensity Grid

In binary grids, scaled point cloud values are projected into voxel points to represent whether there is a surface point in each of voxels. However, in this representation, many point cloud values might be represented with the same voxel value. As long as a point cloud value is projected into a voxel, its value is always 1 no matter how many point cloud values are represented by the voxel. Instead of keeping trace of presence/absence of a point in a voxel, the objective of our volumetric intensity grid is to keep how many points a voxel represents. Each voxel has an intensity value according to the number of point cloud values projected into that voxel. To do this, the voxel value is incremented by one for each projected point cloud value in the equation (2).

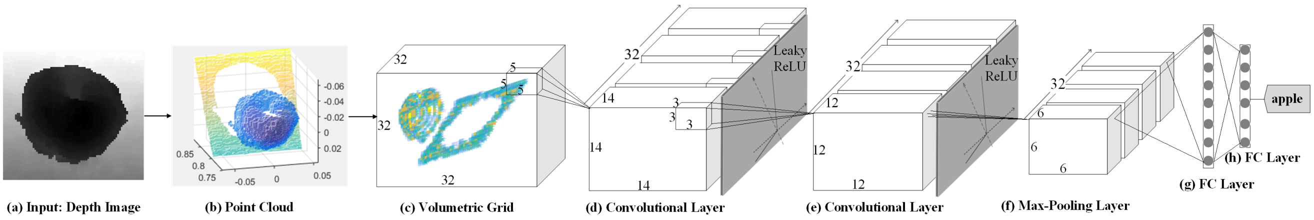

Figure 1. An overview of our model. The input depth image is converted to a point cloud after passing through the first preprocessing step. The volumetric representation is obtained after denosing the point cloud. The convolutional layers have *32* filters with *5* × *5* × *5* and *3* × *3* × *3* sizes followed by a leaky ReLU. The third layer is a pooling layer which downsamples the input volume. The last two layers are fully-connected layers with 128 and 51 unit numbers respectively.

3.2 3D CNN Model

CNNs trained on large databases such as ImageNet [13] have broken new ground in visual recognition. Unlike conventional handcrafted models, deep learning methods provide reusable models. Recently, VoxNet [6] presents an effective 3D CNN model in terms of runtime performance, memory requirements and accuracy based on The Lasagne framework [16]. In this work, the 3D CNN architecture of VoxNet is used for modelling the constructed volumetric representations. The CNN architecture is composed of two convolutional layers followed by the leaky ReLU [17], a max pooling layer after the second convolutional layer and two fully-connected layers as the last layers. The input layer accepts *32* × *32* × *32* volumetric data. The convolution layers perform *5* × *5* × *5* and *3* × *3* × *3* convolutions with stride size *2* and *1* respectively. Each creates *32* feature maps by convolving the inputs with *32* learned filters. The outputs of both convolutional layers pass through a leaky ReLU with parameter *0.1*. The max pooling layer downsamples the input volume by a factor of *2* for each dimension with maximum values. The fully connected layers have *128* and *51* (number of classes) unit numbers respectively. Stochastic Gradient Descent (SGD) is used for minimizing the objective function with *L2* regularization and momentum optimization forms. There are dropout layers after the first convolution layer, the pooling layer and the first FC layer with *0.2*, *0.3* and *0.4* parameters respectively. Originally, VoxNet uses n views of each input to obtain rotationally invariant volumetric model. Our approach uses only one view of an object to perform classification. Despite this limitation, our approach produces promising results.

Table 1. Category recognition accuracy (%) with the complete Washington RGB-D Object Dataset using depth data

Volumetric Grid	with Mask	without Mask
Binary	89.9	93.2
Intensity	93.0	96.1

4 Experiments

To test performance of our approach, we use one of the most commonly used RGB-D benchmarks, the Washington RGB-D Object Dataset [7]. This dataset has *300* object instances in *51* object categories. The dataset contains a total of *207.662* depth images taken from different view angles and sizes. We evaluate our method according to two testing scenarios: (i) We use the approach of volumetric works on ShapeNets [5] and VoxNet [6], in which *80%* of data is used as training split, the rest *20%* is used as testing split. However, these works use ModelNet dataset [5] while we use the Washington RGB-D Object Dataset [7]. We use the entire dataset in this testing scenario. (ii) As a second scenario, we use the commonly used setup on the Washington RGB-D Object Dataset in literature [2, 3, 4, 7, 8, 9, 10, 11, 12]. We sub-sample the dataset by taking every fifth depth image in order to have around *41.500* images. Then, we randomly leave one instance out from each category for testing and train on the rest of the remaining objects. For this testing scenario, the experiments are run *10* times and average results are given. All the tests are performed on a Tesla K40c GPU.

In the first scenario, we investigate recognition performance using both binary and intensity volumetric grids. We also conduct experiments in which no segmentation masks are used in the preprocessing step, showing that our approach is able to deal with the background clutter. Table 1 shows the results obtained by the first scenario. We can see that despite the partial 3D shape views, our approach conducts superior performance. The volumetric intensity grid improves the results significantly. Instead of considering presence/absence of a point projected into a voxel, point intensity in a voxel gives more information for a better classification. Another interesting result is that using segmentation masks negatively affects classification performance. We think that this is due to imperfections in object masks provided with the dataset. In the experiments, we realized that some erroneous masks crop important parts of objects. However, as it can be seen in Table 1, the proposed method handles the background problems without using masks and provides superior performance in the presence of background.

Considering the results in Table 1, we only evaluate the best performing volumetric intensity grid with background combination for the second testing scenario. Table 2 gives the performance comparison of our approach with the previous works. Comparing to Table 1, the decrease in recognition rate is due to test-

Table 2. Performance comparison of category recognition on the Washington RGB-D Object Dataset using depth data

Type	Method	Accuracy(%)
*2.5*D (Hand-crafted)	Kernel SVM [7]	64.7 ± 2.2
	HKDES [4]	75.7 ± 2.6
	KDES [3]	78.8 ± 2.7
*2.5*D	SSL [8]	77.7 ± 1.4
	CNN-RNN [2]	78.9 ± 3.8
	HMP [11]	81.2 ± 2.3
	Subset-RNN [10]	81.8 ± 2.6
	CNN-SPM-RNN [9]	83.6 ± 2.3
	Hypercube [12]	85.0 ± 2.1
Volumetric	**This work**	82.0 ± 2.3

ing on unseen category instances as well as reducing the dataset size by *1/5*. Because the key to success of multi-layered convolutional deep architectures comes from using large datasets. Nonetheless, our approach outperforms all methods except that of Zaki et al. [12] and Cheng et al. [9]. Zaki et al. [12] achieved their success with the help of additional use of outputs of earlier layers in CNN as described in Section 2. They also make use of pre-trained CNNs in depth data to take advantage of large annotated datasets like ImageNet. Besides that, they also take advantage of different data modalities among RGB images, depth maps and point clouds to capture object features. Their separate experiment using depth images and point clouds in isolation gives 79.4% and 70.3% accuracies respectively. Similarly, Cheng et al. [9] make use of surface normals along with depth data as in [11].

5 Conclusion

We have presented a 3D convolutional object recognition approach based on two volumetric representations using depth maps. Although depth maps do not give enough information to build a complete 3D model of objects, the constructed view-based incomplete 3D model is still useful to recognize objects. Our experiments show that the proposed model has achieved higher accuracy than many state-of-the-art approaches on the commonly used Washington RGB-D Object Dataset. To the best of our knowledge, the proposed model is the first volumetric approach on this dataset.

We have demonstrated that 3D CNNs on volumetric representations make it possible to learn rich 3D structural information of objects. We believe this work opens up possibilities for learning rich 3D geometrical features. We plan to explore other possibilities of volumetric learning in the future.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105. ↑1

[2] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, “Convolutional-recursive deep learning for 3d object classification,” in *Advances in Neural Information Processing Systems*, 2012, pp. 665–673. ↑1, ↑2, ↑3, ↑4

[3] L. Bo, X. Ren, and D. Fox, “Depth kernel descriptors for object recognition,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2011, pp. 821–826. ↑1, ↑2, ↑3, ↑4

[4] L. Bo, K. Lai, X. Ren, and D. Fox, “Object recognition with hierarchical kernel descriptors,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 1729–1736. ↑1, ↑3, ↑4

[5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920. ↑1, ↑2, ↑3

[6] D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 922–928. ↑1, ↑2, ↑3

[7] K. Lai, L. Bo, X. Ren, and D. Fox, “A large-scale hierarchical multi-view rgb-d object dataset,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1817–1824. ↑1, ↑2, ↑3, ↑4

[8] Y. Cheng, X. Zhao, K. Huang, and T. Tan, “Semi-supervised learning for rgb-d object recognition.” in *ICPR*, 2014, pp. 2377–2382. ↑1, ↑3, ↑4

[9] Y. Cheng, X. Zhao, K. Huang, and T. Tan, “Semi-supervised learning and feature evaluation for rgb-d object recognition,” *Computer Vision and Image Understanding*, vol. 139, pp. 149–160, 2015. ↑1, ↑3, ↑4

[10] J. Bai, Y. Wu, J. Zhang, and F. Chen, “Subset based deep learning for rgb-d object recognition,” *Neurocomputing*, vol. 165, pp. 280–292, 2015. ↑1, ↑3, ↑4

[11] L. Bo, X. Ren, and D. Fox, “Unsupervised feature learning for rgb-d based object recognition,” in *Experimental Robotics.* Springer, 2013, pp. 387–402. ↑1, ↑3, ↑4

[12] H. F. Zaki, F. Shafait, and A. Mian, “Convolutional hypercube pyramid for accurate rgb-d object category and instance recognition,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on.* IEEE, 2016, pp. 1685–1692. ↑2, ↑3, ↑4

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. ↑2, ↑3

[14] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with microsoft kinect sensor: A review,” *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013. ↑2

[15] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3d point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008. ↑2

[16] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma *et al.*, “Lasagne: First release,” *Zenodo: Geneva, Switzerland*, 2015. ↑3

[17] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. ICML*, vol. 30, no. 1, 2013. ↑3