Deep Learning Advances on Various 3D Data Representations

Rohit Malhotra (s3801128) Lennart Faber (s2500253)

***Abstract*—This essay describes several deep learning based methods for 3D data representation. Several types of data will be distinguished, after which some recent well-performing systems will be discussed.**

***Index Terms*—3D Computer Vision, 3D data representations, 3D Deep Learning.**

I. Introduction

Deep learning methods have shown remarkable successes in the field of 2D computer vision. Their ability to learn highly discriminative features in the input domain has resulted in great advancements in the field. However, deep learning methods require large datasets and are computationally very expensive. This is applying them in the 3D domain has proven to be difficult in the past. More recently however, due to availability of high computational power and ease in acquisition and availability of large 3D datasets, various DL methods are being explored in the 3D domain.

We classify different 3D representations based on their structural and geometrical properties: 3D data representations that have an underlying grid structure that allows for global parametrization and a common coordinate system are categorised as Euclidean data. Euclidean data are useful for identification of rigid objects with only few deformations. On the contrary, a 3D non-Euclidean data representation has no underlying grid structure with no global parametrization. Extending traditional DL methods to non-Euclidean data is difficult as they lack vector space structure (we can not define $+$ and $.$ operations). For Euclidean data however, this is a straightforward task.

Based on the survey [1] by Ahmed et al., our essay will discuss the following:

- Section 2 discusses different 3D data representations and how they are classified as either Euclidean or non-Euclidean.
- Section 3 discusses state-of-the-art DL methods for a few different representations.
- Section 4 discusses some applications of various representations on computer vision tasks.

II. 3D Data Representations

3D data provides rich information about the geometry of the object, hence their adequate representation is of significant importance for computer vision tasks. Because of great advancements in 3D sensing technologies the amount of available 3D data has increased. This data comes in different forms, as discussed above, based on their structural properties and can be classified Euclidean and non-Euclidean data families.

A. Euclidean Data

Data structures that can have global parametrization such as voxel sizes in volumetric data, types of channels in RGB-D data and a location of origin in case of panoramic projections, belong to the family of Euclidean data. They can be further sub-classified into following categories.

1) Descriptors: Shape descriptors are simple representations of 3D objects, which helps in ease of processing, computation and comparison. For an object they represent various characteristics or their combinations, such as geometry, topology, texture and surface.

3D descriptors are further classified into local and global descriptors. Various handcrafted low-level descriptor fail to capture global discriminative features of an object and are therefore often combined with DL methods, which are known to be able to capture hierarchical discriminative features. Mostly unsupervised DL methods are more suitable for these tasks as supervised methods provide hierarchical abstractions of the data and low level descriptors are abstractions of a shape in itself, which would lead to undesirable abstractions of abstractions.

2) 3D data projections: 3D data is projected onto 2D space. Usually, projections are able to capture only some of the 3D properties. Types of preserved properties depend on the type of projections. For example, panoramic projections are invariant to rotation around the principal axis. This representation eases the processing. A pre-processing step is used to normalise data (like pose normalisation) and make a projection, after which 2D CNNs are used to extract features. These methods are simple and have shown better results than global descriptors based methods, but some geometric properties are lost due to projections.

3) RGB-D data: 3D data is represented as a 2D color image, with an additional layer representing a depth map. Due to the increased availability of inexpensive depth sensors, the number of available RGB-D datasets are larger than any other type of 3D datasets. DL methods started using CNNs to extract features from 2D images with a depth layer as additional channel, after which these representations were fed to multiple RNNs and combined into a single embedding. This evolved to processing of depth layer in a separate CNN, which further evolved to using a separate CNN for each channel and transferring the weights from each network.

4) Volumetric Data: We represent 3D data using a three dimensional grid. Voxels are used to define the distribution of object in the 3D space. This representation can also encode the viewpoint information, by assigning each voxel a value representing, visible, occluded or self-occluded. As voxel based representations store information about both occupied and non occupied voxels, they are not suitable for high resolution data. A solution to this problem is octree-based varying sized voxels. Both of these representations however are unable to preserve properties related to the shape and smoothness of the surface. In general, we use 3D-CNNs to extract features from a 3D grid, but 3D-CNNs are computationally very expensive. Methods like batch-normalisation were used in LightNet [5] for fast convergence.

5) Multi-view Data: A 3D object is represented as a combination of multiple 2D images, each representing a different view point. This representation is robust to noise, occlusion and incompleteness. A major question is how many views are sufficient for representations: too few will lead to overfitting and too many will lead to redundant data. It has been observed that multi-view data representations perform better than volumetric data on various tasks, mainly object classification. Multi-view data can use various established 2D DL paradigms very effectively, hence we don't need to tailor our architectures for 3D data. An effective DL architecture is MVCNN [6], which introduced the concept of a view pooling layer.

B. Non-Euclidean Data

Also referred to as geometric data. It is mainly categorised into two sub categories: point clouds and 3D meshes and graphs. This representation is important because most Euclidean representations introduce quantization artifacts, which do not capture invariances in the data.

1) Point Clouds: Point clouds represent the geometry of 3D objects as an unordered set of points. They are simple structures that avoid combinatorial irregularities, hence they are easier to learn from. Any method using point clouds should be invariant to permutations and rigid motions. Point clouds are very easy to capture, but show ambiguity about the surface information.

We will discuss the DL method PointNet [2] in section 3.

2) 3D meshes and graphs: They are represented as a set of vertices in 3D space and a connectivity list that describes how each vertex is connected to the other. Due to the irregular structure of meshes, extending DL methods to these representations is a challenging task. However, meshes can also be presented as graph data structures, with nodes as vertices and connections between them as edges. Hence, DL methods for graphs are applicable to meshes. These methods of convolution on graphs (GCNN) are mainly categorized into spectral and spacial filtering methods.

III. Deep Learning Architectures

This section we will describe DL methods using different data representations: PointNet using point clouds, DeepPano using 3D data projections and ShapeNet using volumetric data representations.

A. PointNet

It is a deep learning framework that directly takes an unordered set of 3D points as input. Each point P_i is set of (x, y, x) coordinates plus extra features such as color, normals etc. For a classification task it outputs k scores for k classes for each point. This framework is invariant towards permutations of order of points in a set. This property is achieved by the use of max pooling. The framework captures the local and global interactions between points by concatenating the global features that are obtained by pooling and local features that are obtained by feature transformations. It is invariant towards transformations such as rotation, translation, rigid motions. This is achieved by aligning the input set to a canonical space before doing feature extraction. An affine transformation matrix is predicted for this alignment.

B. 3D ShapeNet

The method that is considered to be the first method to exploit the the full geometry of 3D object represented as voxels was introduced in [4] by Wu et al. and is goes by the name of 3D ShapeNet. All three dimensions were used by the convolutional kernel in a Convolutional Deep Belief Net (CDBN), resulting in increased performance in comparison to earlier models that merely processed 2D representations of the 3D shapes. However, as mentioned before, volumetric representations of data by 3D-CNNs are computationally very expensive.

C. DeepPano

DeepPano, introduced by Shi et al in [3], works by extracting representations of 3D shapes from 2D images. This is done by converting each 3D shape into cylinder projection around its principle axis, resulting in a panoramic view of the object. DeepPano has been made invariant to rotation around a principle axis by applying a row-wise max-pooling layer in the CNN. Shi et al. showed that performance of methods exploiting the full geometry of 3D objects can be surpassed by computationally less demanding models as they outperformed the 3D Shapenet model on the ModelNet-10 and ModelNet-40 datasets by a large margin.

IV. Analysis and Conclusion

For the task of 3D object classification, multi-view techniques are state-of-the-art, achieving 98.6% accuracy on ModelNet10. They also outperform other methods in retrieval tasks. On the other hand, for correspondence tasks, non-Euclidean DL techniques offer the best performance.

With the ongoing advancements in scanning technologies, there is huge increase in the amount of available 3D data. Also there are great advancements in the field of DL. Hence it is natural to combine both and achieve success in various tasks.

REFERENCES

[1] Ahmed, E., Saint, A., Shabayek, A.E., Cherenkova, K., Das, R., Gusev, G., Aouada, D., Ottersten, B.E. (2018). Deep Learning Advances on Different 3D Data Representations: A Survey. ArXiv, abs/1808.01462.

[2] Qi, C. R., Su, H., Mo, K., Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 652-660).

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

[4] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1912-1920).

[5] Zhi, S., Liu, Y., Li, X., Guo, Y. (2017, April). LightNet: A Lightweight 3D Convolutional Neural Network for Real-Time 3D Object Recognition. In 3DOR.

[6] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision (pp. 945-953).