Seminar 3D Vision: PointNet

MAARTEN BUSSLER, Technische Universität München, Germany

Efficient semantic segmentation and classification algorithms are an important aspect of computer vision. These algorithms work on 3D data and can perceive and interpret the surroundings of a scene, which is crucial to the fields of autonomous driving, virtual reality or indoor navigation. With modern advancements in sensor technology and computational power, it is today possible to apply deep learning techniques on these 3D data specific tasks. Point clouds is a versatile data format that is especially important for the field of autonomous driving. Because of their irregular structure, point clouds are notoriously hard to handle for neural networks and are thus often converted into regular and more voluminous representations like voxel grids. This, however, limits the size of the analyzable data and exposes the input data to quantization errors. This paper aims to give an overview of the PointNet architecture by Qi et. al. [7], a highly versatile and efficient deep learning network that works directly on points clouds without the need of previous data format transformations and forms the basis for many modern state of the art algorithms on 3D data classification and segmentation. Furthermore, this paper discusses different areas of application for PointNet, but also limitations and advancements of the original PointNet architecture.

1 INTRODUCTION

3D object detection and scene segmentation tasks are essential for the success of many real-world applications, such as virtual reality, robotics or autonomous driving. In order to track objects and perceive surroundings in real time, these applications rely on several 3D sensors, such as LIDAR or depth sensors that produce large data sets in the form of irregular point clouds [12]. With respect to these technological challenges and backed by new advances in applying deep learning methods for computer vision tasks, recent research efforts have investigated how deep learning and neural networks could be applied to 3D geometric data. The efficiency and success of many deep learning object detection methods, such as 2D or 3D convolutional neural networks, rely on highly regular and structured underlying data. Since most initial sensor data in the form of point clouds or 3D meshes do not offer a regular format, numerous algorithms first transform these original data formats to structured formats, such as 3D voxel grids, prior to consuming them by a neural network. However, transformation tasks on the initial data bears the risk of introducing quantization errors and renders the resulting data unnecessarily voluminous, which can pose a big obstacle for handling large data sets. In order to perform fast and reliable object detection and semantic segmentation in real time, the industry requires a way to perform deep learning directly on the output point clouds of the 3D sensors. Point clouds consist of a set of points that are defined in 3D space by their (x, y, z) components. While these point clouds are very exact representations of the original sensor data, the single points in the set possess no inherent ordering and the cloud is invariant to the permutations of the points, which makes it difficult to build a learning-based algorithm purely on point clouds.

Qi et al. propose a solution for this problem by introducing PointNet [7]. PointNet is a neural network structure that directly consumes point clouds and offers a structured and learnable representation of point clouds that can be used for 3D object classification and segmentation tasks. Essential for the success of the PointNet network is the usage of a max pooling function in order to counter the irregular structure of the point cloud. In its initial steps the network extracts per point features out of the point cloud and learns a spatial encoding of each point. Subsequently, a max pooling operator selects the dominant points of the point cloud and aggregates the per point features to a global feature descriptor that describes the whole point cloud. Eventually, the global feature descriptor can be fed to a classifier network that then produces labels for object classification, part segmentation or scene semantic segmentation tasks, as seen in Figure 1. Furthermore the authors introduce alignment networks to the PointNet structure in order to canonicalize the input and feature data and further improve the network performance.

This paper is structured as follows: First related research and the importance of point clouds as a data structure are discussed. Then the architecture and properties of the PointNet neural network are presented. Lastly, the influence of PointNet to modern research methods are reviewed.

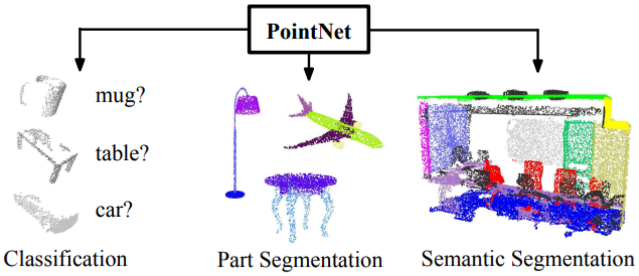

Fig. 1. Utilization of PointNet. PointNet works directly on point clouds and can be used to produce labels for object classification, as well as semantic segmentation tasks. Taken from [7].

2 RELATED WORK

Point clouds are very exact and are often captured directly from LIDAR or depth sensors. As such their handling is especially important for the fields of autonomous driving and virtual reality. However, the irregularity of point clouds poses a nontrivial problem for point cloud-based 3D object detection and segmentation algorithms. Many existing algorithms try to overcome these challenges by either projecting the point clouds to 2D images [6], [13] or use quantization to convert the clouds to regular voxel grids [8], [15]. Subsequently, the efficiency of convolutional networks and 2D detection frameworks can be leveraged. This data representation transformation, however, is not optimal and bears the risks to lose information during quantization, to obscure natural invariances of the original data and to render the resulting data unnecessarily voluminous. Instead of first

Author's address: Maarten Bussler, maarten.bussler@tum.de, Technische Universität München, Germany.

transforming point cloud data to voxels or other regular data formats for feature learning, PointNet directly consumes point clouds in order to perform point cloud classification and segmentation.

2.1 Geometric Data Structures

The visualization and handling of 3D geometric data is a central part of the field of computer vision and multiple modern real life applications. 3D data is highly versatile and can be accessed in various different formats, with some of the most prominent structures being point clouds, 3D meshes or voxel grids. Each format poses unique opportunities and challenges when applied to deep learning algorithms. Voxel grids are highly structured and easy to process for a neural networks, but their memory footprint also grows cubically with their input resolution [1]. 3D meshes encode not only geometry but also topology data and excel at the compact representation of 3D shapes. However, their non-canonical and irregular structure make even simpler image operations a highly non-trivial task [10], which makes it hard to use meshes for training of neural networks. Point clouds are another irregular data structure. This data format represents an object or a scene by a set of unstructured 3D points that are defined by their (x, y, z) components. Although point clouds are a very exact and easy to render data format, they usually perform bad for deep learning related tasks. This is because point clouds possess no canonical ordering, as such the ordering of the points within the cloud is not defined. Thus, a neural network has to be invariant to $N!$ permutations as well as possible rotations and transformations of the input set in order to work directly on point clouds.

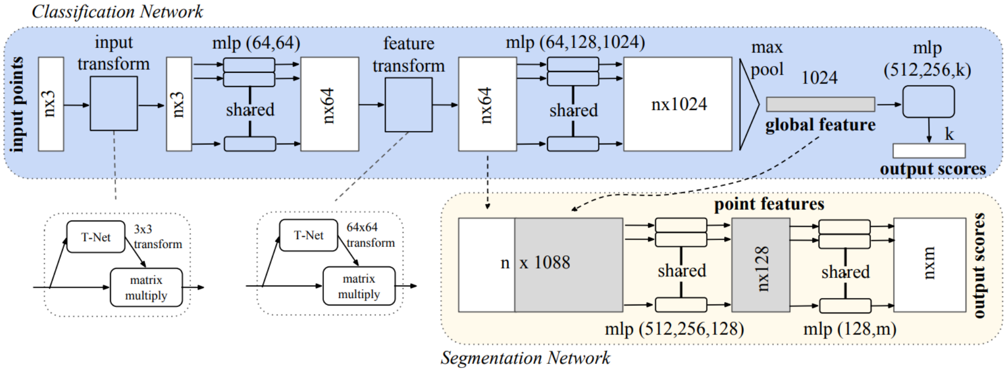

Fig. 2. Pipeline of PointNet. The network takes all n points of the input set and their properties directly as input and performs pose normalization before using multi-layered perceptrons to extract local features from the singular points. The resulting feature space is again aligned with the help of a second transformation network. A max pooling operation then selects the most relevant points of the input to form a global feature vector that describes the whole point cloud. For classification tasks, this global feature descriptor is then fed to a classification network that produces k output scores for k candidate classes. For segmentation tasks, the global and local per point feature information are combined to produce for each input point m output scores for m parts of the object. Taken from [7].

3 ARCHITECTURE OF POINTNET

Figure 2 depicts an overview of the full PointNet structure. The network consists of three main components:

(1) A single symmetric function that aggregates extracted per point features to a global feature descriptor, which encapsulates the whole input point cloud.
(2) Transformation networks that canonicalize input and feature spaces.
(3) An optional segmentation network that combines local and global semantic information.

The following paragraphs discuss the structure of PointNet in more detail.

3.1 Symmetric Function

In order to deploy a successful neural networks that works directly on point clouds, the authors of PointNet have to introduce structure to the original unstructured input data. This is done with the help of a single symmetric function. A function f of n variables is symmetric, if the output value of f is the same for all possible $n!$ permutations of its input. The element-wise max operator on vectors is used as a symmetric function by PointNet to render the network invariant to the input order of the point set.
Figure 3 depicts this core structure of PointNet. The basic concept of PointNet aims to approximate any function $f(\{x_1, ..., x_n\})$ on a set of points by applying a max pooling operation on the transformed elements of the input set:

$$f(\{x_1, ..., x_n\}) \approx \gamma(MAX(h(x_1), ..., h(x_n))) \qquad (1)$$

The function h acts as a feature extractor of the input points, while γ behaves as a classifier. PointNet approximates these functions with the help of multi-layered perceptrons (MLP) that are trained during the learning phase of the network. With a collection of different functions h, different functions f can be realised and thus different properties of the input set can be captured.

3.2 Feature Extraction

PointNet uses feature extraction to learn the local spatial encoding of every point. This is done with the help of multi-layered perceptrons that apply one dimensional convolution kernels on the

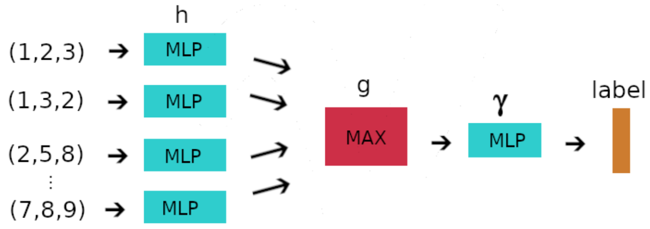

Fig. 3. Basic structure of PointNet. The function h projects each input point to a higher dimension embedding space and extracts their per point features. Then g uses a maximum operator to aggregate the extracted features to a global descriptor that is finally used by γ to generate the final prediction label. Adapted from [2].

(x, y, z) components of the input points until enough point features are extracted and the dimension of the feature vectors grows to a sufficient size. The feature extraction kernels are learned during the training phase of the network. Each kernel captures a specific region of the point cloud. This means that in the exemplary case of an airplane, specific kernels would be activated by the points in the head, body or wings of the plane and corresponding features that would lead to the classification of the point cloud as an airplane would be extracted [14].

3.3 Transformation Networks

PointNet aligns the consumed input sets and their feature spaces to a canonical space in order to relax the constraints of the feature extraction and classification process, as well as to make the semantic labeling invariant to rigid transformations applied to the input point cloud. This is done by introducing another mini network to the larger PointNet structure, the T-Net. These MLPs are trained with the rest of the network and are tasked with predicting affine transformation matrices for each input point. As shown in Figure 2.1, these matrices are then used in the steps of input transformation and feature transform to align different input sets and perform pose normalization, or to align the feature spaces of different input clouds.

3.4 Local and Global Information Aggregation

The global feature descriptor built by the max pooling operator summarizes the input point cloud by a sparse set of key points and can be further used for object classification tasks when fed to a classification network. However, this purely global descriptor does not suffice for the tasks of part- or scene semantic segmentation. For example when performing a segmentation task on a point cloud that encapsulates a plane, it is not enough to identify the geometric labeling of the input, but the network is also tasked with producing labels for each point that identify the part of the object (like wings, or passenger seats) to where the point belongs. As such, a combination of global and local knowledge is needed. As depicted in Figure 2.1, PointNet uses a separate segmentation network to handle this task. Here, each local per point feature is concatenated with the global feature vector. These combined feature vectors can then be used to generate new features and point quantities that link local and global semantics and can then be used by the network to perform segmentation tasks.

4 PROPERTIES OF POINTNET

The authors of PointNet show that the presented neural network is able to function as an universal approximator for any continuous set function [7]. They further note that as a whole, the semantic expressiveness of PointNet is strongly coupled to the number of neurons in the max pooling layer and the size of the global descriptor vector K. Because PointNet uses max pooling to aggregate the local per point features to a global feature vector, only at most K points can influence this global feature descriptor and the network learns to summarize the input point cloud by a sparse set of key points: the *critical set*. The critical set often corresponds to the skeleton of the object contained in the point cloud and defines an upper bound shape of the point cloud. Loosing non critical points or perturbating points between the critical set and the upper bound shape results in the exact same global feature vector and does thus not change the semantic labeling of the input. Thus the max pooling operation and the resulting critical set act as a measure against perturbated or corrupted input sets and provides an inherent robustness to the PointNet network structure.

5 CONCLUSION AND EXTENSIONS OF POINTNET

The PointNet network is a widespread approach for handling deep learning directly on the point cloud data format. The network uses a single symmetric function to structure the input in order to produce labels for object classification and semantic segmentation tasks. Today, PointNet provides the basis for many modern and state of the art applications on point clouds.

However, PointNet is not without it's faults and drawbacks and can still be improved. One limitation of the original PointNet structure is that the neural network does not consider distances to interesting neighbouring points as relevant information when extracting point features. Furthermore, for real life applications, like autonomous driving, high accuracy decisions have to be made in seconds on large data sets consisting of millions of points. This is hard to achieve for the original PointNet, since particularly efficient image analysis methods, like 2D convolutions, are not used. The original PointNet does also not consider the sparsity of real life point clouds that are captured by modern sensors. If PointNet is used on a region of the cloud where the density of points is low, the efficiency of the network also declines. PointNet++ [9] and PointPillars [4] try to solve these problems by improving on local feature extraction with the help of a grouping layer, or pillar encoder respectively, that leverage PointNet on point neighbourhoods of different sizes. In the case of PointPillars these pillar feature maps can then be converted to pseudo 2D images that can then be used by efficient 2D convolution feature extraction techniques. Furthermore PointNet can also be used as a tool for pose estimation to track and align point clouds [5], [11], or to learn hand poses directly from depth images and point clouds [3].

REFERENCES

[1] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Bjorn Ottersten. 2018. A survey on deep learning advances on different 3D data representations. *arXiv preprint arXiv:1808.01462* (2018).

[2] ComputerVisionFoundationVideos. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. https://www.youtube.com/watch?v=Cge-hot0Oc0. Accessed: 2022-01-13.

[3] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. 2018. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8417–8426.

[4] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12697–12705.

[5] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. 2021. PointNetLK Revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12763–12772.

[6] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. 2018. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 641–656.

[7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[8] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. 2016. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5648–5656.

[9] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* (2017).

[10] Yi-Ling Qiao, Lin Gao, Jie Yang, Paul L Rosin, Yu-Kun Lai, and Xilin Chen. 2019. LaplacianNet: Learning on 3D meshes with Laplacian encoding and pooling. *arXiv preprint arXiv:1910.14063* (2019).

[11] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. 2019. PCRNet: Point cloud registration network using PointNet encoding. *arXiv preprint arXiv:1908.07906* (2019).

[12] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 770–779.

[13] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 7652–7660.

[14] Binbin Zhang, Shikun Huang, Wen Shen, and Zhihua Wei. 2019. Explaining the PointNet: What Has Been Learned Inside the PointNet?. In *CVPR Workshops*. 71–74.

[15] Yin Zhou and Oncel Tuzel. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4490–4499.