4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Marc, Brede

Department of Informatics, Technische Universität München

The paper "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks" by the authors Christopher Choy, JunYoung Gwak and Silvio Savarese generalizes the concept of sparse convolutions to arbitrary dimensions and specifically shows that state-of-the-art 3D and 4D perception can be improved using such convolutions. To overcome the challenges of perception in higher dimensions, the authors use different kernel shapes and present a hybrid kernel that can yield a performance increase. Finally, to enforce consistent predictions over the time and space dimensions, the paper presents an approach to applying conditional random fields to higher dimensional data.

1 INTRODUCTION

The paper addresses the perception of the 3D dimensional space over time using convolutions, i.e. 4D Spatio-temporal convolutions. Nowadays, LIDAR or other depth-sensing sensors are being used widely and there is a variety of domains that capture 4D data. There is a lot of value to be gained in perceiving such data for these domains, which include AR/VR applications or robotics. However, the authors claim that at the time of publication, the performance of 3D convolutional neural networks was worse or just on-pair with 2D convolutions and there was a lack of shared open-source libraries that tackle the problem of large-scale 3D perception. The paper 4D Spatio-Temporal ConvNets generalizes the concept of sparse convolutions introduced by [1] to arbitrary dimensions and applies them to the perception of higher-dimensional data. Compared to traditional dense convolutions, sparse convolutions are only applied to the coordinates in space that are non-zero. Additionally, sparse tensors only keep track of values for coordinates that contain values. This makes sparse convolutions more computational and memory-efficient for sparse data input. As the concept of the curse of dimensionality states, data in higher dimensions tends to get sparse, therefore, the authors present a way to perceive data in such dimensions more efficiently. Additionally, the authors introduce a new kernel shape for convolutions and show that a hybrid kernel can yield a performance increase compared to previously used kernel shapes. A final contribution of the paper was applying the concept of conditional random fields to higher dimensions to force the neural network to make consistent predictions.

2 RELATED WORK

At the time of publication, there was no previous work on 4D perception using neural networks. This paper's related work is, therefore, on 3D perception. One can divide the related work on 3D perception into work that uses 3D convolutions and work based on other concepts. Using 3D convolutions before often meant using them in a traditional dense representation [2, 3]. The increase of complexity here was dealt with by concepts like OctNets. OctNets make use of Octree structures that easily allow marking large parts of the space as empty. OctNets introduced a method to convolve a space represented by such an Octree. Some other work already used a sparse representation of convolutions with different quantization methods [4, 5]. Work without 3D convolutions includes an approach that uses 2D convolutions for 3D perceptions [6]. The authors of this paper realized that in 3D space it is often just the surfaces of 3D objects that are to be perceived, therefore, their method applies 2D convolutions to them. Another important branch of 3D perception was introduced by the PointNet approaches [7, 8]. Rather than perceiving the space itself, these methods make use of the set of coordinates and use them as features in a multi-layer perceptron.

3 TECHNICAL SECTION

As already stated, a major contribution is the generalization of convolutions, specifically sparse convolutions, to any dimension. The equation for generalized dense convolutions can be seen in equation 1. Here, the input features $\boldsymbol{x}_u^{in} \in \mathbb{R}^{N^{in}}$ are being convolved with the Kernel $\mathcal{V}^D(K)$. For this, the weight tensor $W \in \mathbb{R}^{K^D * N^{out} * N^{in}}$ is iterated over and applied to $\boldsymbol{x}_u^{in}$ to obtain the values for the output coordinates. This is being done for all output coordinates $u \in \mathbb{Z}^D$. In contrast to this, equation 2 shows the formula for the generalized sparse convolution. There are two differences: As data in a sparse representation only keeps track of non-zero values, the output is only computed for coordinates for which this is the case. Here, these output coordinates are denoted with $\mathcal{C}^{out}$. It was already mentioned that the work of this paper also comprises the introduction of new kernel shapes. The second difference in equation 2 is, therefore, how the kernel shape is being described. $\mathcal{V}^D(K)$ of equation 1 refers to the D-dimensional hyperkernel that is evenly sized in every dimension. $\mathcal{N}^D(u, \mathcal{C}^{in})$ on the other hand, describes a more generalized shape that can have different sizes in different dimensions.

If the input data is represented densely, the mapping from input to output coordinates can be easily inferred. However, for a sparse representation, the mapping is non-trivial and has to be defined. This mapping is called the kernel maps and is denoted with $M = \{(I_i, O_i)\}_i \; for \; i \in \mathcal{N}^D$, as a set of tuples of an input and an output coordinate.

$$x_u^{out} = \sum_{i \in \mathcal{V}^D(K)} W_i x_{u+i}^{in} \; for \; u \in \mathbb{Z}^D \qquad (1)$$

$$x_u^{out} = \sum_{i \in \mathcal{N}^D(u, \mathcal{C}^{in})} W_i x_{u+i}^{in} \; for \; u \in \mathcal{C}^{out} \qquad (2)$$

The generalized kernels that are being introduced, can have arbitrary sizes over the dimensions. This is being done because the authors realized that the previously used hyperkernels are overparametrized. This overparameterization is because the surfaces of 3D objects are to be perceived. While these surfaces scale linearly over time and quadratically to the spatial dimension, the number of parameters in hyperkernels scale exponentially. By relaxing the condition of evenly shaped kernels the authors introduce generalized kernel shapes (Figure 1) and subsequently show that hybrid kernels can yield an efficiency and performance improvement.

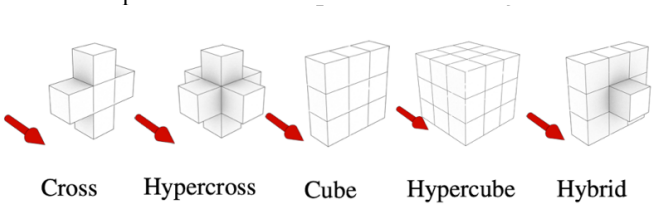

Figure 1: Different 3D Spatio-temporal Kernel Shapes. The red Arrow indicated the Time Dimension and its Direction.

A final contribution of the authors is taking the concept of conditional random fields and applying it to higher-dimensional spaces. This concept forces the model to make consistent predictions of the spatial and time dimensions.

4 RESULTS

At the time of publication, a model implementing all the previously described methods was able to beat all state-of-the-art models on different 3D perception benchmark tasks. Table 1 shows the model's performance compared to other methods

for the 3D semantic label benchmark dataset of ScanNet. Particularly interesting at the time was the margin by which this paper's model beat previously best-performing methods like PointNet.

Table 1: MinkowskiNet Performance on 3D Semantic Label Benchmark on ScanNet

Method	mIOU
ScanNet [9]	30.6
SurfaceConv [6]	44.2
PointNet++ [8]	33.9
PointNet++SW	52.3
MinkowskiNet42 (2cm)	**73.4**

5 CONCLUSION

With the introduction of generalized sparse convolutions, the authors have extended the scope of where convolutions can be applied to higher dimensions. On typical sparse high dimensional data input, sparse convolutions create a memory and computational improvement that make convolving such data feasible. As the significant performance increase compared to previous 3D perception shows, this comes with all the advantages that convolutions already showed on lower-dimensional data like images.

REFERENCES

[1] Benjamin Graham. Spatially-sparse convolutional neural networks. arXiv preprint arXiv:1409.6070, 2014.

[2] Lyne P Tchapmi, Christopher B Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richlyannotated 3d reconstructions of indoor scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.

[4] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Vangelis Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. arXiv preprint arXiv:1802.08275, 2018.

[5] Ben Graham. Sparse 3d convolutional neural networks. British Machine Vision Conference, 2015.

[6] Hao Pan, Shilin Liu, Yang Liu, and Xin Tong. Convolutional neural networks on 3d surfaces using parallel frames. arXiv preprint arXiv:1808.04952, 2018.

[7] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593, 2016.

[8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, 2017.

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richlyannotated 3d reconstructions of indoor scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017.