MouldingNet: Deep-Learning for 3D Object Reconstruction

Tobias Burns, Barak A. Pearlmutter, and John B. McDonald

Department of Computer Science, Maynooth University, Ireland

August 21, 2019

Abstract

With the rise of deep neural networks a number of approaches for learning over 3D data have gained popularity. In this paper, we take advantage of one of these approaches, bilateral convolutional layers to propose a novel end-to-end deep auto-encoder architecture to efficiently encode and reconstruct 3D point clouds. Bilateral convolutional layers project the input point cloud onto an even tessellation of a hyperplane in the $(d+1)$-dimensional space known as the permutohedral lattice and perform convolutions over this representation. In contrast to existing point cloud based learning approaches, this allows us to learn over the underlying geometry of the object to create a robust global descriptor. We demonstrate its accuracy by evaluating across the shapenet and modelnet datasets, in order to illustrate 2 main scenarios, known and unknown object reconstruction. These experiments show that our network generalises well from seen classes to unseen classes.

1 Introduction

Robust algorithms for the analysis of a robot's environment are crucial for it to understand the 3D world, manipulate objects, and plan its actions. These algorithms rely on data captured through the robot's sensors as it moves through and interacts with the environment. A classic example are algorithms that solve the simultaneous localisation and mapping problem where a robot is required to construct a model of the world using data from onboard sensors, whilst simultaneously estimating its motion relative to this model. Although many solutions to this problem are available, the ability to reason about the resultant model is still an open research problem. With the advances made in deep learning in 2D, for problems such as semantic labelling and inpainting, a number of researchers have sought to use these approaches to perform learning and inference on 3D data. However, these advances have yet to be effectively scaled to 3D due to both the spatial and computational complexity of extending convolutions to 3D. For this reason existing approaches differ mainly in the representations of the data used including: voxels [Zhang et al., 2018, Dai et al., 2017, Liao et al., 2018, Stutz and Geiger, 2018], meshes, [Wang et al., 2018, Dai and Nießner, 2018], continuous representations [Park et al., 2019, Mescheder et al., 2018] or point clouds [Qi et al., 2016, Yang et al., 2017, Fan et al., 2016].

In this paper we utilise the encoder of SplatNet [Su et al., 2018] and the bilateral convolutional layers (BCL) defined in [Kiefel et al., 2015] to propose a novel decoder architecture to solve the problem of point cloud reconstruction. The use of BCL layers introduces an assumption on the structure of the data previously lacking from the state-of-the-art work in this area. This *relational inductive bias* [Battaglia et al., 2018] allows for each convolutional kernel to operate over a local structure of points in the point cloud just as traditional 2D convolutions operate over a neighbourhood of pixels in an image. Thus these BCL layers can be used in a hierarchical structure to find simple geometric cues such as edges, corners and curves which can be built into more complex objects. This leads to the network learning aspects of the geometry of the shape rather than networks based on the PointNet [Qi et al., 2016] architecture which learn a *critical point set* that describes the skeleton of the object. As such, the resultant compact encoding created by our approach provides a foundation for solving more complex problems in 3D such as inpainting.

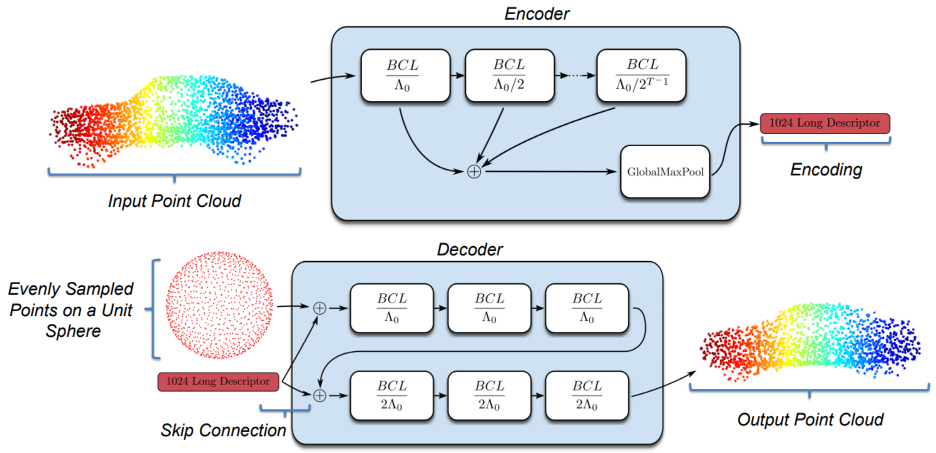

Figure 1: Our network architecture. The input is passed into the encoder to produce a descriptor, which is then appended to the co-ordinates of a unit sphere. The decoder uses this to mould the sphere into the output shape.

2 Approach

Our network architecture follows a similar pattern of traditional auto-encoder networks. However as in SplatNet [Su et al., 2018] we replace the convolutions with bilateral convolutions [Kiefel et al., 2015] which embed the point cloud in a $(d+1)$-dimensional permutohedral lattice in order to efficiently perform convolutions over an unordered point set. The structure of the network is shown in Figure 1 with the encoder similar to the design proposed in SplatNet. We use 5 BCL layers consecutively with a scale factor Λ halving at each layer and input features P_{out} of the previous layer being used as the P_{in} to the next. All layers use the XYZ co-ordinates of the input point cloud as the lattice features L_{in} and L_{out}. Unlike SplatNet rather than keeping the feature vector of each point in the point cloud we use global max-pooling to generate a single 1024 long encoding which describes the global features of the object.

Our decoder works by taking a set of points P_{sphere} of size m, evenly distributed across the surface of the unit sphere and passing them through consecutive BCL layers to mould them into the required shape. In order to do this the 1024 long encoding is first replicated m times. This m-by-1024 matrix is used as the L_{in} of our first BCL layer while P_{sphere} is used as the P_{in}. This is repeated 3 times with the P_{out} of previous layers again being used as the P_{in} to the next as in the encoder. However rather than altering Λ, we instead keep it fixed at a low value to encourage the decoder to produce only the low level features of the object. The output of these 3 layers is an m-by-3 matrix which we then use as our new L_{in} to another 3 BCL layers but using $\Lambda * 2$ in order to reconstruct the finer details of the object.

The loss function we use for point set comparison between $S_1, S_2 \subseteq \mathbb{R}^3$ is the Chamfer Distance (CD), defined as:

$$d(S_1,S_2) = \sum_{x\in S_1} \min_{y\in S_2} \|x-y\|^2 + \sum_{y\in S_2} \min_{x\in S_1} \|x-y\|^2 \tag{1}$$

To compute Equation 1 for each point in S_1 a nearest neighbour in S_2 is found and used to calculate the sum of the L_2 distances. As the nearest neighbour in the CD is not unique this could result in all of the predicted points converging to one position. To combat this the distance is calculated from S_2 to S_1.

3 Experiments

We conduct 2 main experiments in order to demonstrate MouldingNet's ability to not only embed an object's important geometric details but also to generalise to unseen object classes. We provide an evaluation and comparison to the existing methods at the tasks of 1) Representing known 3D shapes shown in Table 1 and 2) Representing unknown 3D shapes shown in Table 2.

	air	bat	bed	ben	boo	bot	bow	car	cha	con	cup	cur	des	doo	dre	flo	gla	gui	key	lam	**AVG**
FoldingNet	0.021	0.029	0.030	0.025	0.033	0.021	0.034	0.030	0.030	0.026	0.036	0.022	0.032	0.021	0.030	0.040	0.032	0.015	0.020	0.032	**0.030**
Ours	0.029	0.044	0.043	0.036	0.050	0.035	0.054	0.043	0.043	0.046	0.052	0.032	0.048	0.029	0.048	0.054	0.046	0.021	0.028	0.054	**0.044**

	lap	man	mon	nig	per	pia	pla	rad	ran	sin	sof	sta	sto	tab	ten	toi	tvs	vas	war	xbo	**AVG**
FoldingNet	0.023	0.032	0.029	0.032	0.027	0.036	0.043	0.024	0.034	0.030	0.030	0.037	0.031	0.021	0.029	0.040	0.032	0.032	0.026	0.029	**0.030**
Ours	0.035	0.051	0.043	0.049	0.039	0.054	0.055	0.041	0.051	0.047	0.044	0.055	0.042	0.029	0.049	0.057	0.047	0.050	0.043	0.047	**0.044**

Table 1: In order to demonstrate our networks ability to represent known 3D shapes we first train over the ShapeNet [Chang et al., 2015] dataset before testing over ModelNet40 [Wu et al., 2014]. We evaluate our results using the Chamfer Distance and compare to the FoldingNet [Yang et al., 2017] architecture.

To train our network 2048 points were evenly sampled from the surface of each object model. The data was then normalised to be scaled within a unit sphere and additionally augmented as described in FoldingNet [Yang et al., 2017] with random axis aligned rotations. For all experiments the ADAM optimizer was used with a cyclical learning rate [Smith, 2015] with values ranging between 1e-5 and 1e-8. The networks were trained for 100 epochs with a batch size of 10 on a single Nvidia GTX 1080ti and Intel Xeon Bronze 3104.

As can be seen in Figure 2 the network is capable of accurately representing and reconstructing complex 3D objects over a range of classes. However Table 1 shows that although we have not yet reached state-of-the-art, overall accuracy on seen classes is encouraging. For unknown classes Table 2 shows that we out perform the full GenRe pipeline, although this is arguably solving a harder problem of predicting the point cloud from a single rgb image, and falling behind GenRe_ORACLE when the ground truth depth is given to it at a mid point. We believe our results could be improved by solving the clustering of points in certain circumstances such as at the tail of the airplane in Figure 2.

	airplane	car	chair	**AVG SEEN**	bench	rifle	cabinet	sofa	table	telephone	loudspeaker	vessel	display	lamp	**AVG UNSEEN**
OURS	0.043	0.054	0.056	**0.051**	0.070	0.052	0.087	0.070	0.091	0.081	0.092	0.062	0.085	0.120	**0.081**
GenRe				**0.064**	0.089	0.112	0.116	0.082	0.096	0.107	0.115	0.092	0.130	0.124	**0.106**
GenRe_ORACLE				**0.034**	0.032	0.021	0.044	0.044	0.038	0.037	0.045	0.030	0.040	0.031	**0.036**

Table 2: In order to demonstrate our networks ability to represent unknown 3D shapes we train over a small subset of ShapeNet [Chang et al., 2015] before testing across unseen classes. We evaluate our results using the Chamfer Distance and compare to the GenRe [Zhang et al., 2018] architecture both with and without an oracle.

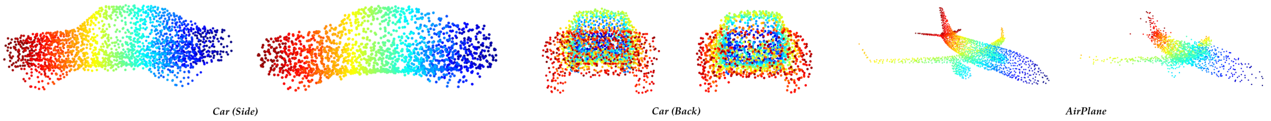

Figure 2: Example inputs (left) and outputs (right) of our architecture. The accuracy of our reconstruction demonstrates mouldingNet is creating a representative descriptor.

4 Conclusion

We have presented MouldingNet, a novel auto-encoder architecture for the task of point cloud reconstruction. First results from our network show promising performance when compared to state of the art, however importantly we believe our network follows a more principled approach to learning by utilising the geometry implicitly present in a point cloud. Our overall objective with this research is to exploit these properties of the architecture to provide efficient and accurate generative point cloud architectures which could be applied to higher level tasks such as 3D inpainting.

Acknowledgments

This research was supported by the Maynooth University John and Pat Hume Scholarship Scheme 2017, Science Foundation Ireland grant 16/RI/3399 and grant 13/RC/2094 to Lero - the Irish Software Research Centre (www.lero.ie).

References

[Battaglia et al., 2018] Battaglia, P. W. et al. (2018). Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

[Chang et al., 2015] Chang, A. X. et al. (2015). ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR].

[Dai and Nießner, 2018] Dai, A. and Nießner, M. (2018). Scan2mesh: From unstructured range scans to 3d meshes. *CoRR*, abs/1811.10464.

[Dai et al., 2017] Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., and Nießner, M. (2017). Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. *CoRR*, abs/1712.10215.

[Fan et al., 2016] Fan, H., Su, H., and Guibas, L. J. (2016). A point set generation network for 3d object reconstruction from a single image. *CoRR*, abs/1612.00603.

[Kiefel et al., 2015] Kiefel, M., Jampani, V., and Gehler, P. V. (2015). Permutohedral lattice cnns. In *3rd International Conference on Learning Representations, ICLR, Workshop Track Proceedings*.

[Liao et al., 2018] Liao, Y., Donné, S., and Geiger, A. (2018). Deep marching cubes: Learning explicit surface representations. *2018 IEEE/CVF CVPR*, pages 2916–2925.

[Mescheder et al., 2018] Mescheder, L. M., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2018). Occupancy networks: Learning 3d reconstruction in function space. *CoRR*, abs/1812.03828.

[Park et al., 2019] Park, J. J., Florence, P., Straub, J., Newcombe, R. A., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. *CoRR*, abs/1901.05103.

[Qi et al., 2016] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593.

[Smith, 2015] Smith, L. N. (2015). No more pesky learning rate guessing games. *CoRR*, abs/1506.01186.

[Stutz and Geiger, 2018] Stutz, D. and Geiger, A. (2018). Learning 3d shape completion under weak supervision. *CoRR*, abs/1805.07290.

[Su et al., 2018] Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M., and Kautz, J. (2018). Splatnet: Sparse lattice networks for point cloud processing. *CoRR*, abs/1802.08275.

[Wang et al., 2018] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y. (2018). Pixel2mesh: Generating 3d mesh models from single RGB images. *CoRR*, abs/1804.01654.

[Wu et al., 2014] Wu, Z., Song, S., Khosla, A., Tang, X., and Xiao, J. (2014). 3d shapenets for 2.5d object recognition and next-best-view prediction. *CoRR*, abs/1406.5670.

[Yang et al., 2017] Yang, Y., Feng, C., Shen, Y., and Tian, D. (2017). Foldingnet: Interpretable unsupervised learning on 3d point clouds. *CoRR*, abs/1712.07262.

[Zhang et al., 2018] Zhang, X., Zhang, Z., Zhang, C., Tenenbaum, J. B., Freeman, W. T., and Wu, J. (2018). Learning to reconstruct shapes from unseen classes. *CoRR*, abs/1812.11166.