Geometric Priors from Robot Vision in Deep Networks for 3D Object Classification

Jean-Baptiste Weibel, Timothy Patten and Markus Vincze

Abstract— **Handcrafted geometric features for object classification are heavily relied on in robot vision because of their demonstrated robustness. While modern deep learning approaches typically outperform classical methods, transferring this success to 3D data in a robust manner is still an open question because of the challenges introduced by the additional dimension and the relative lack of non-artificial large 3D datasets for classification. In this work, we demonstrate the benefits of using a deep network to improve on classical histogram-based descriptors. Our network uses shape features inspired by 3D object classification based on local and global geometry. Due to the geometric priors, our network does not require aligned data and is directly applicable to point clouds. Performance is evaluated on the ModelNet dataset and results show competitive accuracy and robustness, while being rotation invariant and using 10-100x less parameters than some competing methods.**

I. Introduction

Object classification is an intensively studied problem in computer vision that has applications in areas such as security, manufacturing and medicine. The now commonly available depth sensors, such as the Microsoft Kinect, have lead to improvements by allowing additional reasoning about object geometry. This has advanced the perception capabilities in important fields such as robotics and autonomous driving that heavily rely on spatial context.

The representation of 3D data as a point cloud has become the de facto standard for many robotic tasks. However, the difficulty of dealing with the unstructured representation of point clouds has prevented it from being widely adopted by state-of-the-art deep learning methods used in computer vision. Additionally, large datasets acquired by depth sensors are still rare, which limits the direct applicability of data hungry learning algorithms. As a result, classic methods for 3D object classification that use hand-crafted features based on local or global geometric features are still often used.

A common work around to generate large amounts of data is to use artificially computer generated models, however, data acquired by robotic platforms in the real-world are different in a few characteristic ways:

- unaligned objects
- occlusion
- outlier points
- sensor noise

*This research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 610532, SQUIRREL.

All authors are with the Vision4Robotics group (ACIN - TU Wien), Austria {weibel, patten, vincze}@acin.tuwien.ac.at

Objects in the real-world have arbitrary poses, therefore, either the data must be aligned, a challenging problem in itself, or rotation invariant features are required. Occlusion is caused by other objects or distractors that obstruct the field-of-view and outliers are often remnants from a segmentation step. Sensor noise is inevitable with any real sensor. They generate variable point density as well as non-linear noise profiles.

In this work, we present a novel method that is robust to imperfect data with a deep learning architecture. Building on the qualities of classical descriptors, that are typically resilient to various sources of noise, our network learns a probabilistic version of geometric feature histograms. Any histogram-based descriptor can be improved while maintaining their geometric properties and advantages such as rotation invariance (no alignment necessary) as well as robustness to occlusion and variable point density. Our proposed method uses PointNet [9] to learn, in an end-to-end fashion, a probabilistic histogram on the original feature space. The main advantage is that existing rotation invariant features are fed into the network to yield a rotationally invariant deep network for 3D classification. This has not been achieved by prior methods as they either require a separate alignment process (or network), or learn an approximately rotation-invariant network through specific optimization schemes.

Our contribution is the introduction of a general method to improve the descriptiveness of any histogram-based descriptor through learning. We demonstrate our method with two versions of an ESF-like global shape descriptor [19] (one using pairs, the other using pairs and triplets) coined L-ESF and a SHOT-like local descriptor [17] coined L-SHOT. Experiments are performed on the ModelNet dataset with the introduction of artificial occlusion, point density and sensor noise. The results show that our method achieves 83% accuracy and maintains robustness with increased data corruption.

II. Related Work

A. Deep Learning for 3D Classification

Deep learning techniques currently dominate the field of 3D classification. Due to the multiplicity of data representations available when dealing with 3D data, the approaches can differ drastically. We present a non-exhaustive list focusing on the most common and most promising architectures.

The most straightforward way to apply convolutional neural networks (CNNs) to 3D data is to extract 2D depth views from the full model. These depth maps can then be fed to any available CNN. Many mappings from single-channel

to three-channel representation have also been developed [3], [2], thus avoiding the issue of the lack of large datasets for training through the re-use of learned image features. Once a representation for each view is computed, different schemes for pooling them have been developed, from view pooling [16] to more complex view-set reasoning [18]. While performing very well in practice, these approaches lose all information between the views, and raise the problem of view selection since not all viewpoints are accessible. Another disadvantage is that these approaches uses significantly more parameters than other methods.

An alternative approach is to extend 2D CNNs to 3D CNNs by learning features over voxel grids [20], [7]. While this direction is a meaningful extension of CNNs, it has two main problems that are inherent to the design of the network. First, the additional dimension is an optimization burden, and because of the explosion of the number of parameters, they typically use coarse voxel grids, making the data much less informative. This explosion of parameters can be tackled with hierarchical representations [11], or more recently, by embracing the sparsity of the data at the convolution level [8]. Second, they are not rotation invariant by design. Consequently, they need either to learn to match the representation of each orientation of the object, as in [21] or [15], making learning more difficult, or align the model beforehand, which is in itself challenging to perform robustly.

The last direction is to operate directly with point clouds, however, this data representation loses all explicit neighborhood information. One way to compensate is to rely on the creation of a KD-Tree over the set of points [6]. However, the KD-Tree itself depends on the orientation of the model. Moreover, a reasonably large noise can also affect the structure of the KD-Tree. Another option is to rely only on the implicit information of the coordinates [9], [10]. The approach in [9] (PointNet) is to drastically expand the feature dimension of the coordinates (from 3 to 1024 in multiple steps). This enables the network to learn up to 1024 functions expressing a probability of presence in a certain area of the space. This approach requires aligned data, which is achieved using a spatial transformer network [5] to learn a data dependent alignment. Learning such an alignment over a whole object, however, is vulnerable to outliers and occlusion: they modify the data distribution that the alignment method relies on. The representation is then computed on an incomplete and potentially misaligned set of points. In [10] a similar path is taken, except that the models are assumed aligned. Improvement on the layers of the network is made by subtracting a weighted version of the maximum activation value over the whole set for each filter.

B. Robotic Classification

In robotic vision, handcrafted features were carefully developed to capture either local, regional or global geometric features. One of the best performing local handcrafted descriptor is the Signature of Histograms of OrienTations (SHOT) descriptor [17]. This descriptor first aligns the neighborhood along a local reference frame, which is computed as a repeatable representation of the statistical distribution of points. Once aligned, histograms of angles are computed on spatial bins spread in the sphere around the point of interest.

To capture local information, a whole family of descriptors have been created following variants of the scheme introduced by the Point Feature Histogram (PFH) and Fast Point Feature Histogram (FPFH) [13]. These histograms rely on a set of angles between the normal of a point of interest and its neighborhood. PFH/FPFH was extended in Viewpoint Feature Histogram (VFH) [12] to capture global information by considering angles of the vector made between the viewpoint and the centroid of the considered object.

The Ensemble of Shape Functions (ESF) [19] is a global descriptor that relies on simple randomly sampled geometrical entities, such as pairs and triangles. This captures global information in a viewpoint independent manner without relying on a direct neighborhood of a point of interest.

The advantage of these handcrafted descriptors is that they have proven their robustness to most types of noise typical in robotic tasks. However, due to their simple geometric features, randomization and the choice of pooling method (histograms), their descriptiveness is inferior to modern deep learning approaches.

III. Learned Descriptors for Robotic Classification

We propose to learn descriptors using the PointNet architecture as a histogram-like pooling solution. This builds on handcrafted features and thus provides robustness, while introducing learning to achieve good accuracy and generalization. In this section, we will first describe how we learn such a representation. We then provide two concrete applications, one for a global shape representation coined Learned-ESF (L-ESF), and one on a local shape representation coined Learned-SHOT (L-SHOT).

A. Learning Histogram-like Features

Most of the descriptors used in robotics rely on histograms, which is a crucial contribution to their proven robustness. Approaching noisy data as a set is a good trade-off between the amount of information discarded and the robustness of the description, and a histogram is the most straightforward way for set pooling. The PointNet architecture, by working on one data point at a time but optimizing over the whole set, provides a structure to learn functions that activate when a data point is present nearby. The functions behave like probabilistic bins over the Euclidean space and the global optimization ensures that the functions are optimally spread. An ensemble of such functions can therefore be seen as a probabilistic histogram that is trainable end-to-end. This idea can be extended to any other space.

Our proposed architecture is shown in Figure 1. In this setup, the spatial transformer network of the original PointNet architecture is not necessary as we no longer rely on Euclidean coordinates. Instead, four one-dimensional

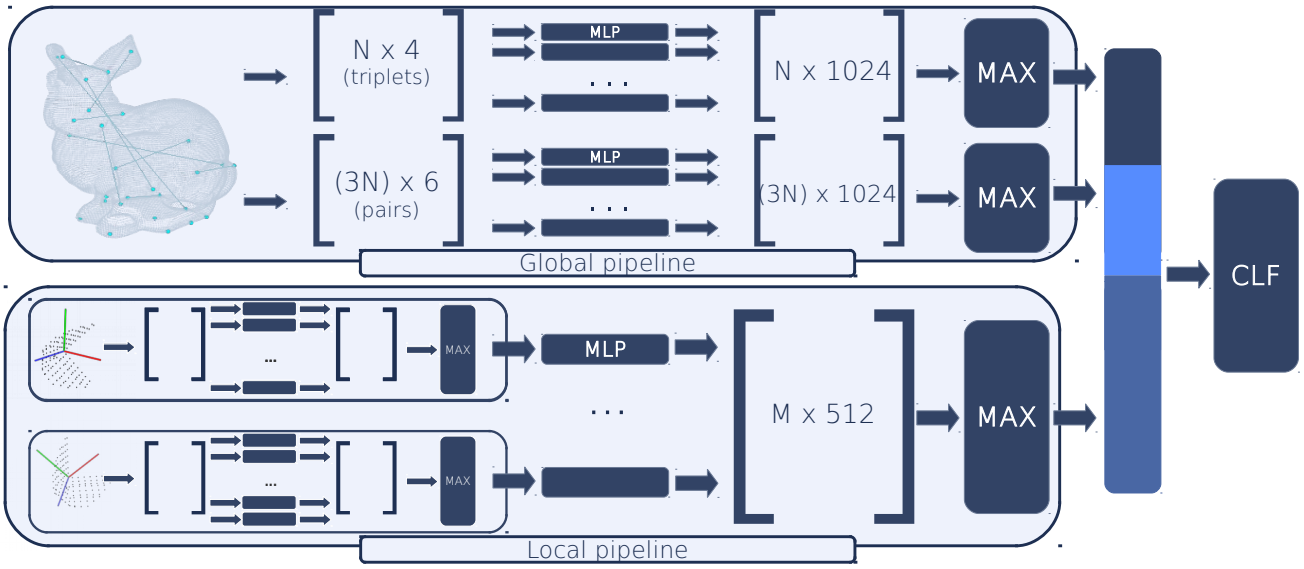

Fig. 1. System overview of our proposed method. The classifier layer is composed of two fully connected layers. N is the number of pairs sampled, M is the number of local descriptors sampled.

convolutional layers are used, with a kernel size of 1. A progressively increasing number of filters are then used before a max-pooling layer to implement the histogram-like feature learning scheme. Each of the layers include a batch normalization step [4]. We also subtract a weighted version of the maximum value of a given filter over the whole set to each output, as described in [10]. ReLU is used as an activation function.

B. Global Pipeline - Learned ESF

We first extract robust global shape information with a pipeline that is inspired by the ESF descriptor [19]. This descriptor was chosen because it is a good performing viewpoint independent handcrafted global shape descriptor. The ESF descriptor first samples random pairs and triplets of points and extracts handcrafted features for each. Various histograms are then created. For a pair of points, the features chosen are their spatial distance and the percentage of the length of the line between the points that is filled with surface points. This quantity is determined by tracing the line in a voxel grid of size 64x64x64 and counting the percentage of filled voxels. For a triplet of points, the angle of the triangle and the area covered by the triangle are computed.

Following this model, and after scaling our point cloud to the unit sphere, our global shape descriptor randomly samples triplets of points and extracts a number of features using the three points. Flat triangles are rejected as they are uninformative in our pipeline. Both the triplet and pair features are rotation invariant, making the whole pipeline rotation invariant as well.

1) Triplets: For a triangle with sides a, b and c, we extract the three angles and the square root of the area of the triangle using the Heron formula,

$$A = \sqrt{s(s-a)(s-b)(s-c)}, \quad \text{where } s = \frac{a+b+c}{2}. \tag{1}$$

This creates a four-dimensional feature vector for each triangle.

2) Pairs: In addition to the features extracted for the ESF descriptor, we also draw inspiration from the Point Pair Feature [1], and Fast Point Feature Histogram [13] descriptors. For the two sampled points, $\overrightarrow{p}_1$ and $\overrightarrow{p}_2$, and their respective normals (which are assumed to be normalized), $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$, we extract the distance d between the points

$$d = \left\| \overrightarrow{p}_1 - \overrightarrow{p}_2 \right\|. \tag{2}$$

The cosine similarity between the normals (3) and the absolute value of the cosine similarity between the vector $\overrightarrow{p}_1 - \overrightarrow{p}_2$ and each of the normals (4) are also computed

$$cos(\angle(\overrightarrow{n}_1, \overrightarrow{n}_2)) = \overrightarrow{n}_1 \cdot \overrightarrow{n}_2, \tag{3}$$

$$\left| cos(\angle(\overrightarrow{p}_1 - \overrightarrow{p}_2, \overrightarrow{n}_{\{1,2\}})) \right| = \left| \frac{(\overrightarrow{p}_1 - \overrightarrow{p}_2).\overrightarrow{n}_{\{1,2\}}}{\left\| \overrightarrow{p}_1 - \overrightarrow{p}_2 \right\|} \right|. \tag{4}$$

Finally, as in the ESF descriptor, we consider the percentage of $\overrightarrow{p}_1 - \overrightarrow{p}_2$ that is on the surface of the object.

Following the PointNet [9] architecture for the classification of the global pipeline, each set of features is fed to its own convolutional neural network, always operating on a single element (pair or triplet) at a time. The feature dimension is enlarged and the set is then max-pooled. Due to the intractable number of potential pairs and triangles,

information about fine structure can be lost. For this reason, we introduce a local descriptor pipeline.

C. Local Pipeline - Learned SHOT

Finer structure needs to be captured to improve the descriptiveness of our approach. However, computing local descriptor densely would be wasteful. As such, we introduce a method to guide the sampling of local patches.

1) Attention Model: We use an attention model that is based on the statistical consistency of the normal orientations over the whole model. Finer structures are characterized by a larger angle between neighboring normals. However, local variations of the angle between normals cannot be relied on solely, otherwise, any rounded surface would be considered salient. Consider the example of a flower pot: local descriptors on the leaf are probably more informative than redundant local descriptors on the rounded pot itself.

Therefore, a histogram of angles between each point's normals and its neighbors' normal is created. Each point contributes to the same histogram, thus capturing globally the frequency of each angle value on the whole model. The histogram is normalized such that the sum of the bins is equal to one. With $\|P\|_0$ the number of non-zeros elements in the histogram and P_k the k-th entry in the histogram, the following transformation is applied

$$\tilde{P}_k = \begin{cases} \|P\|_0 - P_k \text{ if } P_k \leq \|P\|_0 , \\ 0 \text{ otherwise } . \end{cases} \tag{5}$$

This allows statistically significant angles to be captured while removing all highly recurrent angles.

In a second step, the saliency value for each point is computed. This is done by checking the saliency value of the transformed histogram bin that corresponds to the angle between the point's normal and the normal of its neighbors. The values for each neighborhood are summed.

Sampling salient points is done according to a Poisson distribution. The attention model gives a zero probability to many points, therefore, we draw from the average between the distribution from the attention model and a uniform distribution.

2) Local Descriptor: The learned local descriptor follows the design of the SHOT descriptor [17]. Firstly, a robust and repeatable local reference frame (LRF) is computed for alignment. Then, the sphere of all neighbors is divided into 32 bins with a histogram of normal angles computed for each.

In contrast to the PointNet architecture [9] that uses a learned alignment, our architecture transforms all neighboring points in the LRF, where the sampled salient point is used as the origin. The coordinates of the points are then fed into a smaller version of PointNet. The removal of the spatial transformer network drastically reduces the number of parameters.

TABLE I
CLASSIFICATION ACCURACY ON THE MODELNET DATASET [20]

	MN10	MN40	Input	Rot. Inv.
3DShapeNets [20]	83.5	77	Voxel grid	✗
VoxNet [7]	92	83	Voxel grid	✗
PointNet [9]		89.2	Point Cloud	~
KD-Networks [6]	94	91.8	KD-Tree	✗
MVCNN [16]		90.1	Views	~
SHOT+PointNet	83.3	73.9	Point Cloud	✓
ESF	81.1	70.4	Point Cloud	✓
SHOT+ESF	85.2	76.9	Point Cloud	✓
L-ESF	84.6		Point Cloud	✓
L-SHOT+L-ESF (Pairs)	86.3	83.0	Point Cloud	✓
L-SHOT+L-ESF	87.0	83.0	Point Cloud	✓

IV. EXPERIMENTS

A. Classification Accuracy - ModelNet

An evaluation is performed on the ModelNet dataset [20]. This is a CAD model dataset that has two variants, ModelNet40 with 40 classes, and ModelNet10, which is a 10 class subset of ModelNet40. Experiments in this section are performed with clean data, and when necessary, with the aligned version of ModelNet10 and ModelNet40.

Our method was designed to be used with point clouds, so as a pre-processing step, point clouds from the original CAD models were extracted by re-sizing the CAD model to the unit sphere and then sampling points on the surface.

For the global pipeline, 2000 triplets (or 5000 pairs if working only on pairs) are sampled for ModelNet40 and 1500 triplets (or 3200 pairs) are sampled for ModelNet10. For the local pipeline, 50 salient points are drawn using Poisson sampling from a distribution that is the average of a uniform distribution and our attention model. For all experiments, the accuracy is averaged over 5 runs on the test set due to the random nature (sampling) of our algorithm. For the SHOT, ESF and SHOT+ESF results, the learned versions of the descriptors are replaced with the original descriptors as implemented in [14]. The same classification layers are maintained. We classify the set of SHOT descriptors with a PointNet architecture for a fair comparison on the robustness analysis (thus improving its accuracy compared to a more classical bag of words approach). The accuracy results can be found in Table I for ModelNet.

B. Invariance and Robustness

Our proposed network was designed with robust features. To demonstrate their intrinsic power, we devised a set of transformations applied to the input point cloud and report the corresponding accuracy. It is important to note that, unless specified otherwise, **the experiments are performed without re-training the network, which is only trained on clean data.** The following experiments demonstrate that the network carries over the robustness of the chosen features. We report the results of our proposed architecture in comparison to the original descriptors, whose robustness has already been proven. No experiments are required regarding

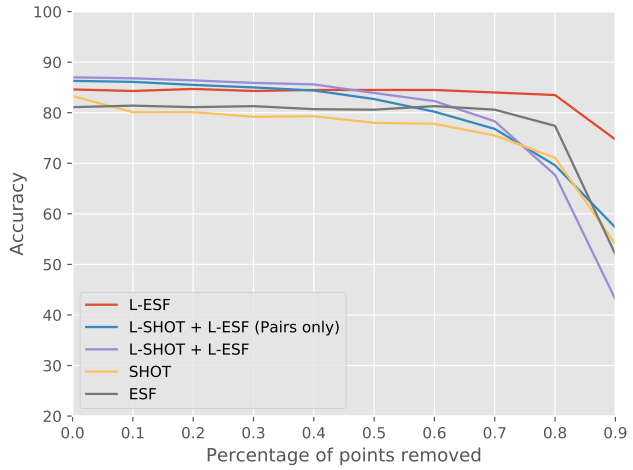

Fig. 2. Influence of the point density on the accuracy

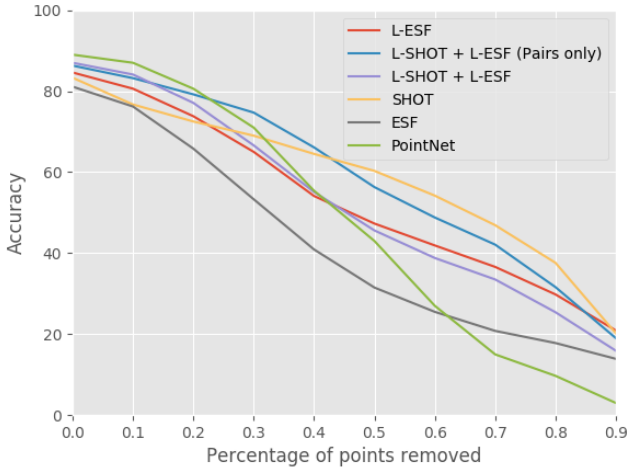

Fig. 3. Influence of occlusion on the accuracy

data that is not aligned as all features fed to the network are rotation invariant.

1) Point Density: We first evaluate how well the network behaves depending on the point density of the point cloud. Point density is reduced by randomly removing points from the original point clouds. Results of the experiments are reported in Figure 2.

2) Occlusion: Artificial occlusion is introduced by removing a neighborhood around a randomly sampled point. The size of the neighborhood corresponds to the percentage of the points being removed. Results are reported in Figure 3. Experiments with PointNet and artificial occlusion were performed with ModelNet40.

3) Sensor Noise: Full models prevent a true analysis of realistic sensor noise. However, generic noise can be modeled by adding Gaussian noise. The standard deviation of the Gaussian noise is chosen as a proportion of the longest distance between points in the point cloud. Results

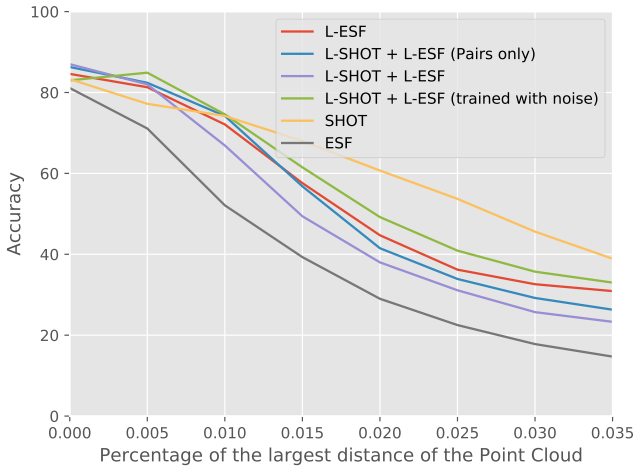

Fig. 4. Influence of sensor noise on the accuracy

are reported in Figure 4. We also show the improvement of retraining with noise for L-SHOT+L-ESF.

C. Discussion

Our approach achieves results that are better than or on par with the methods based on vanilla voxel grids and classical descriptors but worst than the view-based methods. It should, however, be noted that view-based methods use architectures with significantly more parameters (10-100 million compared to around 1 million for ours). Compared to the point cloud based methods, KD-Networks performs best but requires aligned models, and was therefore evaluated on the aligned version of ModelNet40, so it is not as robust. PointNet also performs well but learns an alignment which is itself sensitive to occlusion and outliers.

In the case of ModelNet10, most of the misclassifications are caused by the confusion between `night_stand` and `dresser`, and between `table` and `desk`. In the case of ModelNet40, there is additional confusion between `flower_pot` and `plant`. A deeper observation of the CAD models in each of these classes shows that the mistakes are quite reasonable. Overall our architecture shows promising potential as a robust generic shape feature.

In the study of the robustness of our model, we can see that most of the desirable properties of the classical descriptors are retained. Only the robustness to Gaussian noise seems worse. This is due to the setup of our experiment: this study focuses on the intrinsic properties of the features used, rather than on the already demonstrated learning capabilities of neural networks. For the classical descriptors, the robustness to Gaussian noise mostly comes from the use of a histogram rather than the features themselves, and it can be seen that training with some noise improves the robustness to noise. A final observation is that introducing more geometric priors in the network improves results with clean data. The experiment with PointNet on artificial occlusions shows that the performance steeply degrades as the amount of occlusion

increases.Training with occlusions could mitigate this issue, but would make training harder, potentially needing more parameters than with an appropriate prior.

Our learning scheme transfers the robustness of the original features, as can be seen by the individual evaluation of the priors, and is overall rotation invariant. However, the simple scheme for fusing each pipeline is suboptimal regarding the robustness, and a noise-adaptive scheme would be necessary to make the most of each pipeline. Introducing more geometric information also allows us to more efficiently use the parameters, and use a more compact network, i.e. no spatial transformer and a rotation invariant representation. Through the use of randomization, fewer parameters and batch normalization, our model is less likely to overfit because every representation of a given instance is slightly different, making it possible to train on smaller datasets.

V. Conclusion and Future Work

We have presented a novel architecture that provides robust yet descriptive shape features through the use of geometric priors. Moreover, the scheme devised in this paper can be used to adapt any local or global histogram-based handcrafted feature into a learned descriptor, thus facilitating better task specific performance and end-to-end learning. Our first evaluation shows promising results but indicates a further investigation of a better fusion scheme for the priors is still needed.

The flexible nature of the architecture also allows its use in both a single and multi-view scenario. In the case of multiple views, it can perform efficiently, as information already computed over previous sets can easily be included to the next step through the max-pooling layer over the whole set, while still preserving inter-view information.

Finally, the geometric priors make the features more interpretable. By keeping track of the indices used during the max-pooling step, it is possible to extract the contributing pairs, triplets and/or local structures, which allows insight to be gained about choices made by the network. As an extension, such local structures could be used for correspondence problems, such as pose estimation, which can be done by sampling pairs as demonstrated in [1].

References

[1] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model Globally, Match Locally: Efficient and Robust 3d Object Recognition," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[2] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," *arXiv:1507.06821 [cs]*, 2015.

[3] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," in *Proc. of European Conference on Computer Vision (ECCV)*, 2014.

[4] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proc. of International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[5] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2017–2025.

[6] R. Klokov and V. Lempitsky, "Escape From Cells: Deep KD-Networks for the Recognition of 3D Point Cloud Models," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[7] D. Maturana and S. Scherer, "Voxnet: A 3D Convolutional Neural Network for Real-time Object Recognition," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[8] A. Notchenko, Y. Kapushev, and E. Burnaev, "Large-scale shape retrieval with sparse 3d convolutional neural networks," in *Analysis of Images, Social Networks and Texts*, W. M. van der Aalst, D. I. Ignatov, M. Khachay, S. O. Kuznetsov, V. Lempitsky, I. A. Lomazova, N. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, A. V. Savchenko, and S. Wasserman, Eds. Cham: Springer International Publishing, 2018, pp. 245–254.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] S. Ravanbakhsh, H. Su, J. Schneider, and B. Poczos, "Deep Learning with Sets and Point Clouds," in *Proc. of International Conference on Learning Representations (ICLR)*, 2017.

[11] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2155–2162.

[13] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1848–1853.

[14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[15] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3d object recognition," in *British Machine Vision Conference (BMVC)*, 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2017/SZB17a

[16] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015.

[17] F. Tombari, S. Salti, and L. Di Stefano, "Unique Signatures of Histograms for Local Surface Description," in *Proc. of European Conference on Computer Vision (ECCV)*, 2010, pp. 356–369.

[18] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition," in *Proc. of British Machine Vision Conference (BMVC)*, 2017.

[19] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2011, pp. 2987–2992.

[20] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.

[21] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.