PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors

Simon Boche

Seminar Report: Recent Advances in 3D Computer Vision
Computer Vision Group - Technische Universität München

Abstract

This report presents *PPF-FoldNet*, a novel unsupervised approach for 3D local feature extraction. Based on previous works, such as *PointNet*, *FoldingNet* and *PPFNet* an autoencoder structure is built by adapting *PointNet's* architecture as encoding part of the network and using folding operations as a decoder alternative. Input data in the form of a point cloud is transformed into a local patch representation based on point pair features. Using point pair features as input to our network makes the approach invariant to 6DoF transformations. The new method is tested on the well-known *3DMatch Benchmark* Dataset and outperforms other state-of-the-art methods, learning-based and handcrafted ones, in terms of recall by at least 6%. This margin even increases if rotations are present in the input data. Compared to other methods, *PPF-FoldNet* yields a 20% higher recall on a randomly rotated dataset. Due to good generalization properties, *PPF-FoldNet* could be easily extended to other tasks, such as classification or object pose estimation.

1 Introduction

Many computer vision applications use local descriptors as tools in tasks as object detection, pose estimation or Simultaneous Localization and Mapping (SLAM). While methods are today well established for 2D problems, there are still a lot of issues to deal with in 3D local feature extraction. Most methods still lead to features that lack good discriminative power and repeatability [1].
Like in many other applications, current research tries to face these problems by using learning-based approaches. In 2D, learned descriptors are already able to outperform handcrafted feature extraction algorithms. But unlike in 2D, learning local features in 3D so far has still suffered from several shortcomings [1]. These are:

- being supervised and thus requiring an enormous amount of labeled training data
- sensititivity to 6DoF rotations
- requiring expensive pre-processing of data
- unsatisfactory performance

Haowen Deng, Tolga Birdal, and Slobodan Ilic, the authors of *PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors* introduce a new approach, *PPF-FoldNet*, to tackle the previously listed problems of 3D feature learning by using an unsupervised approach based on an autoencoder structure operating on rotation invariant point pair features as input to the neural network. This report will name related work that *PPF-FoldNet* is based on and give a short description of the introduced method before showing experimental results on a common benchmark dataset, *3DMatch Benchmark*. The report will conclude with a short summary and an outlook for possible extensions.

2 Related Work

The development of *PPF-FoldNet* is mainly based on three different related learning-based approaches for 2D feature extraction, namely *PointNet* (Stanford University 2017, [6]), *FoldingNet* (Carnegie Mellon University 2018, [10]) and *PPFNet* (TUM 2018, [2]). For the novel approach, the best attributes of these three have been combined and adapted to build up the structure of

PPF-FoldNet (TUM 2018). In the following, a brief overview of the basic concepts of these three approaches is given.

PointNet: *PointNet* is a supervised architecture processing input data in the form of unstructured 3D point clouds. Its architecture consists of a point-wise multi-layer perceptron (MLP) using max pooling operations to aggregate local features into a global one. Max poling effectively makes the network learn a set of criteria that selects interesting or informative points of the point cloud [6]. *PointNet* can be adapted to a wide range of tasks such as keypoint extraction, 3D segmentation and classification.

FoldingNet: Like *PointNet*, *FoldingNet* also works on point clouds as input but in contrast to *PointNet*, it constructs an unsupervised extension by setting up an autoencoder structure. Learning to reconstruct its input allows us to export a low-dimensional latent variable in the neural network as a local descriptor of a set of points. The novel approach in *FoldingNet* was the use of *folding* operations in the decoding part of the network. Instead of costly interpolations or voxelizations, folding tries to warp an underlying low-dimensional grid towards a desired set [1]. By the time *FoldingNet* was introduced it significantly outperformed other state-of-the-art unsupervised approaches [10].

PPFNet: *PPFNet* is again a supervised approach but instead of working on pure point clouds it combines point pair features with points and their normals within a certain local vicinity. It proposes to learn local features informed by the global context of the scene. To achieve this, it is seeking to find correspondences between all patches of two fragments. The authors of [2] were able to show that their novel globally aware 3D descriptor was performing better than state-of-the art feature extraction methods, especially under challenging conditions, e.g. in presence of rotations. However, *PPFNet* had one big downside, it showed a significant memory bottleneck.

3 Method description

The method to be presented, *PPF-FoldNet* aims to combine the best properties of the aforementioned developments and therefore is using only point pair features as input and is operating without supervision.

3.1 Construction of Point Pair Features (PPFs)

We usually obtain our input point cloud as a set of oriented points

$$\mathbf{X} = \left\{\mathbf{x}_i : \mathbf{x}_i = \{\mathbf{p}_i, \mathbf{n}_i\} \in \mathbb{R}^6\right\}$$

including the 3D coordinates $\mathbf{x}_i$ of each point and its corresponding surface normal $\mathbf{n}_i$. From that, we create subsets, so-called local patches, containing all points that are within a certain vicinity of a reference point $\mathbf{x}_r$. Each local patch can then be encoded as a collection of PPFs by computing point pairs between any point and the central reference point. Each of these point pairs is then uniquely defined by 4 parameters F_i. These parameters are illustrated in Figure 1 for an arbitrary point pair $\{\mathbf{m}_1, \mathbf{m}_2\}$ and are: the distance $||\mathbf{d}||_2$ between the two points (F_1), the angles between the distance vector and each normal (F_2, F_3) and the angle between the two normals (F_4). This representation is invariant to rigid body transformations.

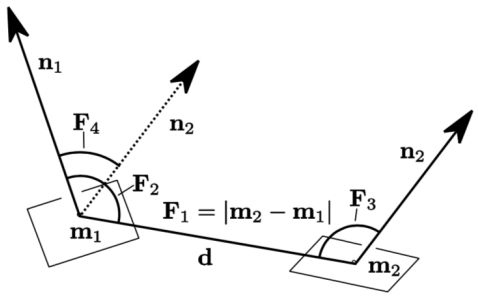

Figure 1: Visualization of point pair features [8].

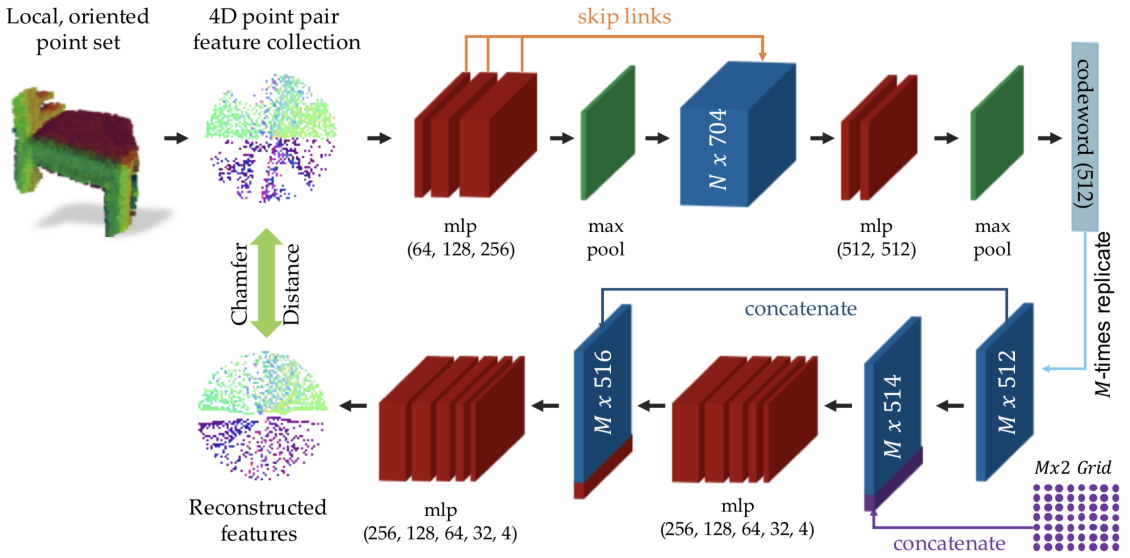

Figure 2: Network architecture of PPF-FoldNet [1].

3.2 Network Architecture

PPF-FoldNet builds up an autoencoder structure by adapting and using *PointNet* in the Encoder and *FoldingNet* in the Decoder. The overall architecture is shown in Figure 2. After having computed the 4D PPFs, our input of dimensions $N \times 4$, with N being the number of samples per local patch, is passing a three-layer point-wise neural network with ReLu activation functions followed by a max pooling layer to create a global feature out of the local features. Via skip links in each layer, our global features are concatenated with low-level ones to obtain a more powerful representation. Another two-layer perceptron and an additional max pooling layer finally lead to our latent variable, the *codeword*, a vector of 512 elements which will be our encoded descriptor for the local patch.
The decoding part of the architecture basically consists of two folding operations, each followed by a five-layer perceptron. More precise, the codeword is replicated M times and concatenated with an $M \times 2$ grid before passing a five-layer perceptron. For the second folding operation, the output of the first MLP is then again concatenated with the replicated codeword and fed into another five-layer MLP. With the dimension of the last layer being 4, this leads to an output of dimensions $M \times 4$ which represents our reconstructed point pair features. To finally train our network, we need to set up a loss function that we can evaluate between the input PPFs ($N \times 4$) and the reconstructed ones ($M \times 4$). To do so for sets of unequal cardinality, we use the *Chamfer Loss* which is given by:

$$d\left(\mathbf{F}, \hat{\mathbf{F}}\right) = \max\left\{ \frac{1}{|\mathbf{F}|} \sum_{\mathbf{f} \in \mathbf{F}} \min_{\hat{\mathbf{f}} \in \hat{\mathbf{F}}} \left|\left|\mathbf{f} - \hat{\mathbf{f}}\right|\right|_2 , \frac{1}{|\hat{\mathbf{F}}|} \sum_{\hat{\mathbf{f}} \in \hat{\mathbf{F}}} \min_{\mathbf{f} \in \mathbf{F}} \left|\left|\mathbf{f} - \hat{\mathbf{f}}\right|\right|_2 \right\} \tag{1}$$

for two sets $\mathbf{F}$ and $\hat{\mathbf{F}}$. This can be interpreted as calculating the average euclidean distance between each point of the first set to its nearest neighbor in the second set and vice versa. We, then take the maximum of those two distances.

4 Experiments and results

The presented algorithm has been implemented using Tensorflow framework accelerated on GPU. All parameters have been initialized randomly by Xavier's algorithm. For minimization, an ADAM optimizer with exponentially decaying learning rate has been used on batches of size 32 [1].

4.1 Data preparation

For evaluation of experimental results and comparison against other methods, the *3D Match Benchmark*, provided by Princeton University, is used. This benchmark contains a total of 62 different scenes with fragments fused from 50 consecutive depth frames [11]. Out of these scenes, 54 are reserved for training and validation and the remaining 8 are used for testing. Additionally, the color information is omitted such that the network becomes insensitive to illumination changes.
To construct PPFs for the network, the fragments are downsampled with spatial uniformity and local patches are formed by all points within a 30 cm vicinity of the reference points. The corresponding normals are computed in a 17-point neighborhood according to the approach by *Hoppe et al.* [3] based on signed distance functions. For a fair comparison with state-of-the-art methods, the local patches are limited to 2048 points, although it can be easily extended as it does not suffer from the memory bottleneck of *PPFNet*. Therefore, the developers of *PPF-FoldNet* also provide a version using 5000 points.

4.2 Accuracy evaluation

Given a pair of fragments $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}$ and $\mathbf{Q} = \{\mathbf{q}_i \in \mathbb{R}^3\}$ that can be aligned via a rigid body transformation $\mathbf{T} \in SE(3)$, we can define a set of ground-truth matches $\mathbf{M}_{GND}$ by setting an inlier distance threshold τ_1 such that:

$$\mathbf{M}_{GND} = \{\{\mathbf{p}_i, \mathbf{q}_i\} : (\mathbf{p}_i, \mathbf{q}_i) \in \mathbf{M}, ||\mathbf{p}_i - \mathbf{T}\mathbf{q}_i||_2 < \tau_1\} \tag{2}$$

The set of feature matches obtained from the network $\mathbf{M}$ is obtained by using nearest neighbor search $NN(\cdot, \cdot)$ in the feautre space.

$$\mathbf{M} = \{\{\mathbf{p}_i, \mathbf{q}_i\} : g(\mathbf{p}_i) = NN(g(\mathbf{q}_i), g(\mathbf{P})), g(\mathbf{q}_i) = NN(g(\mathbf{p}_i), g(\mathbf{Q}))\} \tag{3}$$

where $g(\cdot)$ is the learned mapping function from input space to feature space (encoding part of the network). Furthermore, we define the inlier ratio r_{in} as percentage of true matches in $\mathbf{M}$.

$$r_{in} = \frac{|\mathbf{M}_{GND}|}{|\mathbf{M}|} \tag{4}$$

We demand r_{in} to be higher than a specified inlier ratio threshold τ_2 for a correct match of two fragments. With these definitions, we introduce the recall R as a measure for the quality of features. For a set of fragment pairs $\mathbf{S} = \{\mathbf{P}, \mathbf{Q}\}$, assumed to match under ground-truth alignment, the recall is given by:

$$R = \frac{1}{|\mathbf{S}|} \sum_{i=1}^{|\mathbf{S}|} \mathbb{1}(r_{in}(\mathbf{S}_i = (\mathbf{P}_i, \mathbf{Q}_i)) > \tau_2) \tag{5}$$

In equation (5) the number of fragment pairs that have a sufficiently large inlier ratio are counted and divided by the total number of fragment pairs such that the recall R finally yields the percentage of correctly detected matches out of all true matches.

4.3 Results

The performance of features obtained from *PPF-FoldNet* is evaluated and compared against a variety of other methods including state-of-the-art learning based methods (*3DMatch* [11], *CGF* [5], *PPFNet* [2] and *FoldingNet* [10]) as well as handcrafted features (*Spin Images* [4], *SHOT* [9] and *FPFH* [7]). Initially, we fix the inlier distance threshold to $\tau_1 = 10$ *cm* and the inlier ratio threshold to $\tau_2 = 5$ %.
The results show, that in this setup for single scenes other methods might have a higher recall but on average, *PPF-FoldNet* yields a recall of approximately 68 % which outperforms all other methods by at least 6 %. Additionally, using the extended version with patches of 5000 points, one can obtain a further improvement of 3 %. To investigate the impact of the parameters τ_1 and τ_2 on the performance of all methods, the experiments are once repeated with varying inlier distance threshold τ_1 (while fixing $\tau_2 = 5$ %) and once with varying inlier ratio threshold τ_2 (while fixing

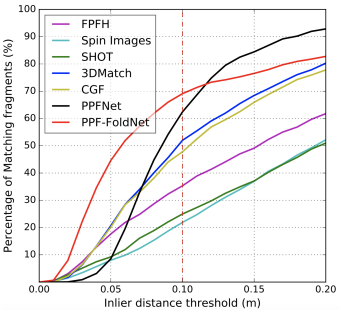

Figure 3: Variation of τ_1

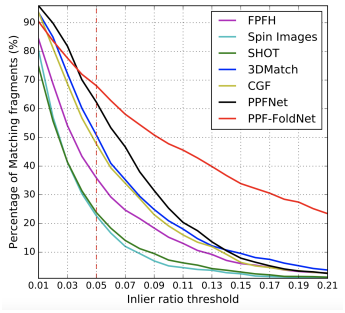

Figure 4: Variation of τ_2

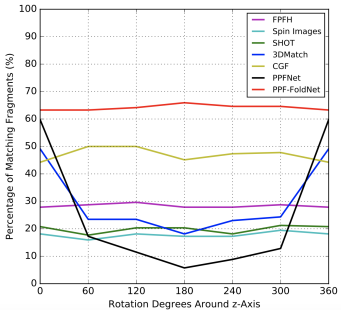

Figure 5: Rotated fragments

$\tau_1 = 10\ cm$). The results are shown in Figure 3 and 4. One can observe, that for strict inlier distance requirements (below 12 cm), *PPF-FoldNet* yields the highest recall of all methods that have been compared. For higher choices of τ_1, only *PPFNet* produced better results. The variation of τ_2 in a range of 0 - 20 % shows that when increasing the threshold for the inlier ratio above 5 %, *PPF-FoldNet* outperforms all other methods and the gap between its recall and the other methods is even growing the larger τ_2 becomes. Especially, for the maximum of 20 %, *PPF-FoldNet* is still able to match more than 20 % of fragments whereas all other approaches are hardly able to match any fragments.
As the goal of the authors of [1] is to learn rotation invariant local descriptors, two more experiments have been executed to test the robustness of the introduced approach against rigid body transformations. In a first test, random fragments have been taken and gradually rotated around the z-axis from 60° to 360° in steps of 60°. The resulting recall values are shown in Figure 5. One can observe two things. First, while some methods like *PPFNet* or *3DMatch* more or less fail completely, *PPF-FoldNet* yields an almost constant recall over the whole range of rotations. And second, it outperforms all methods, even those who are also rotation invariant, by a large margin.
A second test on rotation invariance is done by introducing a new benchmark, the *Rotated 3D Benchmark*, which is obtained from regular *3D Benchmark* by rotating all fragments around randomly sampled axes over the whole rotation range. On this benchmark, some of the previously mentioned approaches fail completely. For example, *3DMatch* and *PPFNet* have recall values around 1 % or even lower. In contrast, *PPF-FoldNet* is barely impacted and achieves the best results on all scenes with an average recall of 69 % (73 % for extended 5K network). These results are nearly identical to the results on the standard benchmark.

5 Conclusion

With *PPF-FoldNet*, a novel unsupervised approach for learning of 3D local descriptors has been introduced. It combines the best attributes of previous developments, especially *PointNet*, *FoldingNet* and *PPFNet*. Building blocks towards its outstanding performance are the ability to operate on sparse input data, a property inherited from *PointNet*, folding operations in the decoder and the use of PPFs which in the end make the approach invariant to 6DoF transformations.
In several experiments, it has been shown that the introduced method outperforms state-of-the art feature extraction methods, learning based as well as handcrafted ones, on standard benchmark datasets under challenging conditions and varying point cloud density. Compared to other methods that also operate on point pair features, *PPF-FoldNet* is able to result in a far higher recall. Besides of achieving a superior performance, the network is interpretable as the point pair features can visualized during the training progress. Therefore, a geometrical projection of the 4D PPFs into a 2D space based on polar coordinates is used. Some further benefits are a better computational performance in terms of time compared to other methods [1] and the fact that it can be easily extended to larger patches without running into memory issues. As it can furthermore be shown that *PPF-FoldNet* has good generalization properties, it offers a broad range of applications for possible extensions in the futures. Exemplary for this, the features could be adapted to tasks like classification or pose estimation.

References

[1] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[2] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the ACM*, 1992.

[4] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. In *IEEE Transactions on pattern analysis and machine intelligence*, 1999.

[5] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltuni. Learning compact geometric features. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[6] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation*, 2009.

[8] Carsten Steger. Lecture Slides: "Bildverstehen 2: Robot Vision", 2019.

[9] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. In *Computer Vision and Image Understanding*, 2014.

[10] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Andy Zeng, Shuran Song, Matthias Nießner, Fisher Matthew, Xiao Jianxiong, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.