NPM3D project : Distinctive 3D local deep descriptors

Salma ELGHOURBAL

Master IASD, March 2021

Contents

Introduction

In this document, we review the paper [5] written by Fabio Poiesi and Davide Boscaini in 2020 and accepted in the IEEE International Conference on Pattern Recognition 2020. We will first discuss the context of the paper. Next, we will summarize the main results of the paper then we will present some experiments that we did using the model presented in the paper. Finally, we will conclude and express our personal opinion about the paper and our implementation.

1 Previous related work

Many previous papers study how to encode local 3D geometric information that can be used for a task such as registration or object recognition.. Existing solutions belong to 2 categories : one-stage and two-stage methods. The former encode geometric information of a patch directly from Its points (spin images, the perfect match, SHOT). The latter first estimate a local reference frame to canonicalise the patches and then compute local descriptors (3DMatch, PPFNet, Fully Convolutional Geometric Features, D3Feat, FPFH). The studied paper is a two-stage method that learns the feature descriptors in an end-to-end manner.

2 Brief summary of the paper

2.1 End to end learning method

2.1.1 Training

The authors invent a new method called DIPs (Distinctive 3D local deep descriptors) which is a deep learning approach that aims at learning rotation-invariant compact 3D local descriptors. DIPs can be used to apply rigid point cloud registration. Furthermore, It doesn't require an initial alignment. Given a selected patch $\mathcal{X}$ of n points in a point cloud, a neural network applied to this patch outputs a vector of size d that generates DIPs $f = \phi_\theta(\mathcal{X})$.
In more details, here are the steps of the end to end learning method :

1. The input is two point clouds $\mathcal{P}$ and $\mathcal{P}'$ that overlap in regions $\mathcal{O} \in \mathcal{P}$ and $\mathcal{O}' \in \mathcal{P}'$. During the training procedure, we know the ground truth transformation $\mathcal{T}$ between $\mathcal{P}$ and $\mathcal{P}'$. The alignment between the two regions is done using a nearest neighbourhood search after applying $\mathcal{T}$ to $\mathcal{P}'$
2. $b = 256$ points are sampled from $\mathcal{O}$ following the farthest point sampling method. Then their corresponding neighbors in $\mathcal{O}'$ are retrieved.
3. Around each point c sampled with FPS in $\mathcal{O}$, we build a local patch of radius τ_r. The same is done to each corresponding nearest neighbor c' in $\mathcal{O}'$. The points in each local patch are used to compute a local reference frame LRF. The equations of the three axis of LRF are given in the paper [3]
4. From each patch $\mathcal{Y}$ and $\mathcal{Y}'$ from $\mathcal{O}$ and $\mathcal{O}'$, $n = 256$ points are sampled for computation constraints. These points are centered and normalized and then aligned to the local reference frame.
5. A network is trained following a Siamese approach that processes pairs of patches concurrently using two branches with shared weights. Each branch contains the following operations :
    - A **transformation network** (TNET) similar to that used in FullPointNet that we implemented in the course. It learns an affine matrix A that is applied to each point x among the n points of a patch $\mathcal{X}$. Here, the produced matrix A is not constrained to be orthogonal.
    - **A first Multilayer Perceptron** (MLP1) block consisting of three **shared** MLP layers of size (256,512,1024).
    - A **bottleneck** γ that computes max pooling (which is claimed to be better than average pooling).γ predicts how informative DIP is and produces permutation-invariant outputs called global signature $\gamma = (g_1, g_2, .., g_m)$ with m is the number of channels of a previous layer. α represents the vector of indices that produced γ in the patch. If we take two corresponding patches $\mathcal{X}$ and $\mathcal{X}'$ from overlapping point clouds, then the corresponding values of α and α' can be interpreted as the correspondences between points in $\mathcal{X}$ and $\mathcal{X}'$. The norm of γ can be effectively used to quantify the reliability of γ . The authors found that the more structured the surface enclosed in a patch is, the higher the value of $\rho = \|\gamma\|_2$. Patches with flat surface and incomplete information have low value of ρ. Thus, a threshold τ_p can be chosen to discard patches with flat surfaces for example. Here the bottleneck dimension is 1024.
    - **A second Multilayer Perceptron** (MLP2) block consisting of three MLP layers of size (512,256,d).The block learns the embeddings of descriptors.

- Finally, a **Local Response Normalisation layer** performs L2 normalization to have unitary length descriptors.

2.1.2 Loss functions

The aforementioned siamese network is trained by combining two loss functions.

- **The Chamfer loss** is used to train the TNET. It aims at reducing the distance between the TNET output for each point x in $\mathcal{X}$ (Ax) and the TNET output for Its neighbor in $\mathcal{X}'$ (Ax')
- **The Hardest-contrastive loss** is applied to the siamese network's output. Given a pair of anchors, the computation of this loss mines the hardest negatives for these pairs. Minimizing this loss results in bringing close together the anchor descriptors and enforcing hard negatives to be as far as we want from anchors.

2.1.3 Testing method

In this section, we explain how the trained model is applied on a test set and particularly on two overlapping cloud points.
According to the code provided by the authors in the demo file "demo-3dmatch.py" available in the github `https://github.com/fabiopoiesi/dip`, random points are sampled from both provided overlapping point clouds. Next, for each point in either clouds, the local reference frame is computed and each local patch is extracted and set to contain 256 points which are then aligned to the LRF. Each patch is processed with PointNet to compute a DIP feature vector of size d. After that, RANSAC is used for registration, in each RANSAC iteration, random points are picked from the source point cloud. Their corresponding points in the target point cloud are detected by querying the nearest neighbor in the d-dimensional DIP feature space. The method finally estimates the rigid transformation to apply to source point cloud in order to be aligned with the target cloud.

2.1.4 Performance metrics

Two evaluation metrics are used to compare the method proposed in this paper to alternative descriptors :

- **Feature-matching recall** : This metric is computed accross matching point clouds having over 30% overlap. It doesn't require RANSAC since It directly averages the number of correctly matched point clouds across datasets.The FMR represents the percentage of successful alignment whose inlier ratio is above some threshold which measures the matching quality during pairwise registration.
- **Registration recall** : This metric corresponds to the percentage of successful alignment whose transformation error is below some threshold. In others words, It quantifies the miss-rate by measuring the distance between corresponding points for each point cloud pair using the estimated transformation based on ground-truth point correspondence information.

2.2 Experiments presented in the paper

The authors train their model on the 3DMatch dataset which is an RGB-D dataset. They provide the values of the different parameters (e.g the number of epochs, the patch radius, the number of iterations per epoch, the learning rate, etc). The resulting model is tested on 3D match, on 3DMatchRotated so as to assess the rotation-invariance ability of the model, on the ETH dataset that is a laser scanner dataset in order to assess the ability of DIPs to generalise across sensor modalities and finally on a

new dataset called VigoHome dataset created by the authors which is a smartphone RGB dataset.

The authors compare their method with 14 state-of-the-art methods on the aforementioned datasets and carry out an ablation study. The results of their experiments can be summarized as follows :

- The DIPs outperforms all 14 state-of-the-art methods in terms of feature-matching recall on the two datasets 3DMatch and 3DMatchRotated. It is better than PPFNet [2] which is also a method based on a PointNet. Also, the FMR values for DIP for the two datasets are so close (0.958 and 0.955). This shows that DIPs descriptors are rotation invariant. Moreover, the computation of the registration recall on different scenes (e.g Kitchen, home,..) shows that DIP has higher recall than other methods except for the lab scene. In fact, this is because there are a lot of flat surfaces in the scene and not much geometric information in contrast to the kitchen scene where DIPs are more distinctive descriptors.
- Regarding the ablation study, the authors study the effect of the TNet, LRF and LRN modules on the feature-matching recall.This study is carried out on 3DMatch and 3DMatchRotated and the main results are : the LRF is essential for DIPs to be rotation invariant. In addition, the LRN layer improves the FMR. Finally, increasing the size of the feature descriptor also improves the FMR.
- On the ETH dataset, the DIP method has a higher FRM on average in comparison with other methods. Comes second, the 3Dsmoothnet [3] : It uses smoothed density value voxelization as input representation for the deep neural network that outputs the local feature descriptors whereas DIPs uses the raw point representation. Hence, DIPs are more generalizable accross sensors and are more robust. The performance of learned DIPs on VigoHome dataset is quite good even if the dataset is very different from 3D Match.

3 Our experiments

The training of the model from scratch using the official implementation of the paper fails at iteration 1150 out of 99600 due to a limit in the quota of ressources in Colab. So instead, we tested a pretrained DIPs model provided by the authors on different kinds of datasets. The model was trained on the 3D Match Dataset and uses 32 as the dimension of the feature descriptor. To do so, We adapted the file "demo-3Dmatch.py" published in the official github of the paper to the datasets we want to use and instead of directly visualizing the registration results with Open3d which could have been done if Colab supported displaying 3D point clouds, we export the clouds after transformation to .ply format and visualize them in CloudCompare. The results are discussed below.

- Bunny original and bunny perturbed : We used the two 3D clouds for bunny original and bunny perturbed provided in the 2nd TP. We fixed the LRF kernel to the value of $0.03\sqrt{3}$. The patches and LRFs calculation uses 5000 points sampled from both point clouds. As the figure 1 shows, the DIPs method successfully registrates the two clouds, i.e the clouds in pink and white coincide.

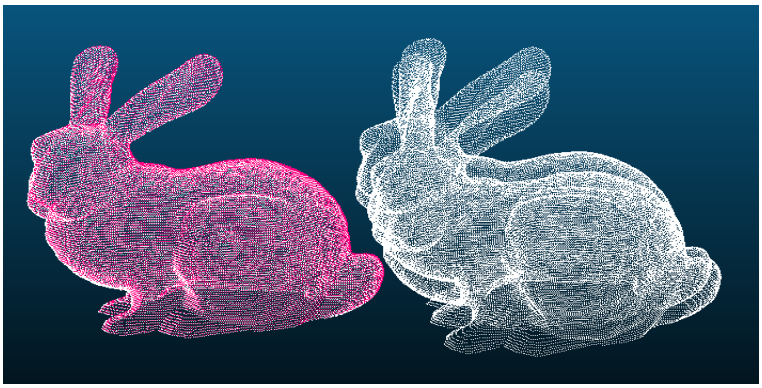

Figure 1: Right : before applying DIPs, Left : after applying DIPs

- Two fragments from 3D match : We used the same parameters as specified by the authors : LRF kernel of size $0.3\sqrt{3}$ and 5000 points sampled from each fragment. The result is shown in the figure 2. The registration time was around 1 min. DIPs performs well on this indoor RGBD dataset.

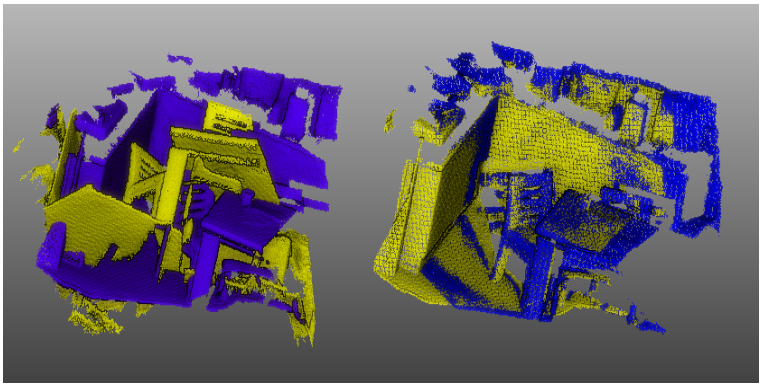

Figure 2: 3Dmatch dataset, left : before applying DIPs, right : after applying DIPs

- Synthetic data : We generated a point cloud in the shape of a cylinder using Open3D and created another point cloud by applying a rotation on that cylinder. Each cloud contains 30000 points. In this setting, the LRF kernel is fixed to the value of $0.03\sqrt{3}$ and as before, the computation of patches uses 5000 sampled points from each cylinder. Here, we know the ground truth transformation, so It is possible to compare It to the transformation retrieved by DIPs + RANSAC. The figure 3 shows the result of the registration. There is a slight error in the estimation of the transformation matrix. Nonetheless, the result of the registration looks acceptable.

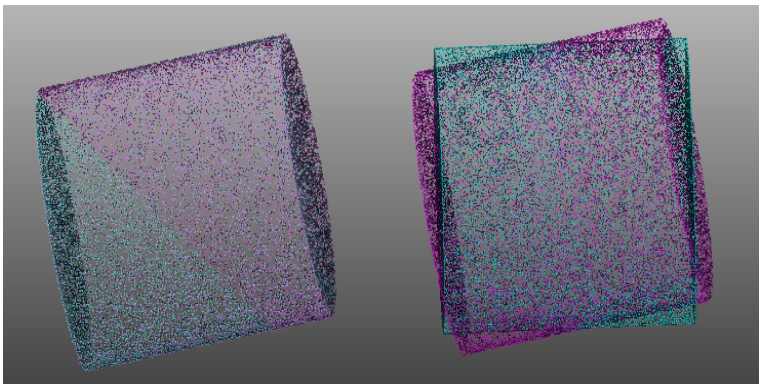

Figure 3: Cylinders, right : before applying DIPs, left : after applying DIPs

- Stanford buddha : We used two overlapping clouds for Buddha from the Stanford 3D scanning Repository available on the address : `http://graphics.stanford.edu/data/3Dscanrep/`. We applied the pretrained model with the same parameters as for the bunny. The figure 4 shows the registration output. Here, the DIPs + RANSAC was able to align both clouds even if there is just a little overlapping between them and that there are missing points in the clouds.

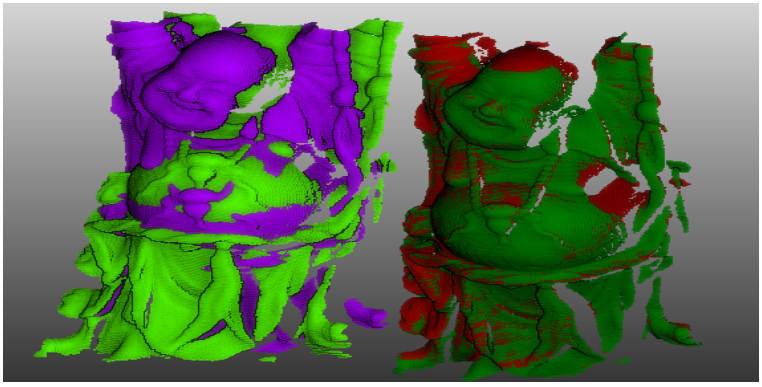

Figure 4: Buddha : left, before DIPs and right : after DIPs

- Notre dame des champs : We applied DIPs on the two clouds of Notre dame that were provided in the 2nd TP. Here, we increased the number of points sampled to 20000 because the cloud points are huge (1M and 4M points). We first tried 200000 but the computation time was over 1h and did not finish computing the LRFs. The kernel size is fixed to the value of $5\sqrt{3}$. We show the registration results in the figure below. We can see that the original clouds are so close (small translation between them). But, this transformation was not well computed by DIPs. Instead, we got a completely false registration. We also observed that the computation time is very high (20 min).

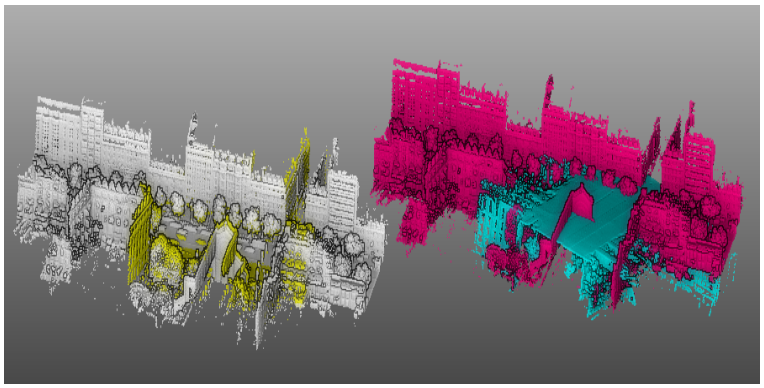

Figure 5: Notre dame des champs, left : before applying DIPs, right : after applying DIPs

4 Contributions and limitations

To conclude, we discuss the contributions and the limitations of this paper. First of all, DIPs is a data driven method that computes rotation-invariant compact descriptors and that doesn't require any handcrafted preprocessing after patch canonicalisation. They can generalise well across different sensor modalities because they are learnt end-to-end from locally and randomly sampled points. The method is robust to noise and missing points. The experiments of the researchers show that DIPs outperforms 14 state of the art methods on ETH dataset and generalizes to a new type of dataset. Furthermore, based on our experiments, the DIPs generalises well to synthetic data (cylinder) and to stanford models (bunny and buddha).
Regarding limitations, from our experiments we noticed that the pretrained DIPs model doesn't work well on a point cloud with a lot of points (like the Notre dame des champs cloud points). Also, the LRF computation takes a lot of time (93% of the execution time). Second, if the scene contains a lot of partially reconstructed objects and flat surfaces and lacks structured elements, the DIPs performance drops a little.

5 Posterior related papers

The paper which was published in 2020 has been cited twice so far. The first citation is by Sheng Ao et al. in the paper [1]. This paper presents another new registration end to end method called SpinNet that is different then DIPs but shares the same properties : descriptiveness, rotation invariance and not including any hand crafted features. Even though the authors cite DIPs, they do not compare SpinNet's performance to DIPs. Instead, they use the same metrics (FMR) to compare SpinNet with other methods such as PPFNet, 3DMatch, PerfectMatch, FCGF..etc on the same datasets presented in this paper in addition to KITTI dataset.
The second paper that cites the DIPs paper is written by Sofiane Horache, Jean-Emmanuel Deschaud and François Goulette. They present a novel method for registration called MS-SVConv. According to this paper, Dips has the following limitation "[...] patch-based methods are very slow. Moreover, because each patch is treated individually, their design is usually not flexible enough to add new capabilities (keypoint detector, end-to-end extension, feature pre-training for semantic segmentation)." [4]. Instead of using patches, the authors use a deep neural network based on sparse voxel convolutions which are much faster.

References

[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *arXiv preprint arXiv:2011.12149*, 2021.

[2] H. Deng, T. Birdal, and S. Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[3] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Wieser Andreas. The perfect match: 3d point cloud matching with smoothed densities. In *International conference on computer vision and pattern recognition (CVPR)*, 2019.

[4] Sofiane Horache, Jean-Emmanuel Deschaud, and Franccois Goulette. 3d point cloud registration with multi-scale architecture and self-supervised fine-tuning. 2021.

[5] Fabio Poiesi and Davide Boscaini. Distinctive 3d local deep descriptors, 2020.