The descriptiveness of feature descriptors with reduced dimensionality

Dániel Varga[1], János Márk Szalai-Gindl[1], and Sándor Laki[1]

Eötvös Loránd University, Pázmány Péter sétány 1/C, H-1117, Budapest

Abstract. Nowadays, depth data has an important role in many applications. The sensors which can capture depth data became essential parts of autonomous vehicles. These sensors record a huge amount of 3D data (point clouds with x, y, and z coordinates). Furthermore, for many point cloud processing applications, it is important to calculate feature vectors that aim at describing the neighborhood of each point. Usually, a feature vector has high dimensionality, and storing it in a database is a difficult task. One of the most common operations on feature descriptors is the nearest neighbor search. However, earlier works show that nearest neighbor search with spatial index structures in high dimensions could be outperformed by sequential scan. In this work, we investigate how dimensionality reduction on 3D feature descriptors affects the descriptiveness.

Keywords: feature descriptor · PCA · nearest neighbor search

1 Introduction

One important sub-area of point cloud processing is registration, where the task is to align two point clouds to each other, i.e., to make a transformation which moves one point cloud to another in such a way as to minimize the distance between identical parts of the surfaces defined by the clouds. For larger point clouds, feature-based registration methods are the most common [5]. These processes have the following steps: (1) detect the keypoints to select such points for which the neighborhood is distinguishable from the rest of the point cloud, (2) compute the feature vectors to create descriptors that represent the geometric properties of the neighborhoods, (3) determine the correspondences to match the key points of two point clouds based on the distance between their feature vectors, (4) compute the transformation based on the previous step. The feature descriptors mentioned in the second step are usually elements of a high-dimensional vector space which are needed to store in a database due to various data management and data analysis tasks.

Suppose that we examine sequences of data distributions where the dimensionality increases along the sequences. For example, such a sequence can consist of uniform distibutions of different dimensions. Beyer et al. showed in their classic paper [1] that it is true for a wide range of the aforementioned sequences of data distributions that the distances of a query point from the nearest and the farthest neighbors tend to be equal as the number of dimensions increases. In these cases, the concept of the nearest neighbor becomes meaningless in a data distribution with sufficiently high-dimensionality. However, even if the dimensionality is not so high, it can still occur that the indexing techniques are outperformed by sequential scan. They also point out that there are situations in which high-dimensional data vectors spread in a subset of the space such that their intrinsic dimensionality is significantly lower than the original dimensionality of the data space. (Intrinsic dimensionality is the minimum number for attributes needed to describe data [2] at least with a good approximation.) In these cases, the nearest neighbor queries and indexing may make sense. Further works addressed different data distributions with different dimensions and analyzed the performance of indexing techniques such as R-trees for the nearest neighbor search (e.g. [15], [7]). For example, if the data set has the self-similar property, then the performance depends on the intrinsic ("fractal") dimensionality of it [7]. As Samet writes in his book [13, Chapter 4], one of the reasons is the intrinsic dimensionality of a data set why dimension reduction techniques are used for it before storing in a database. Another reason mentioned in this book is that the possible number of child nodes decreases at a given page capacity as the dimensionality increases for some disk-based search trees (e.g. the R-trees) because, for example, the space requirement for the representation of a minimum bounding box also depends on the dimensionality. However, if the number of child nodes decreases, the height of the search tree increases, leading to a decrease in performance. Because of the above, on one hand, we apply Principal Component Analysis (PCA) on the feature descriptor data; on the other hand, we modify the internal parameters of the feature descriptors to reduce the dimensionality. We examine how their performance changes.

Based on Guo et al. [4], one of the most important attributes of a feature descriptor is descriptiveness, which shows the capability to encapsulate the information of the underlying surface. There are existing works in which the authors compared 3D point feature descriptors. Guo et al. compute precision and recall values based on point pair correspondences, obtained from nearest neighbor searches [14] [4]. Usually, these papers compare feature descriptors using only their default number of dimensions. Prakhya [10] compares the descriptiveness of feature descriptors, using PCA to reduce the dimensionality of the descriptors. They use RRR metric to measure descriptiveness, which is slightly different from the previously mentioned methods [14].

2 Method

In this work we investigate three well-known 3D point feature descriptors: Point Feature Histogram (PFH) [11], Fast Point Feature Histogram (FPFH) [12] and Spin Image (SI) [6]. Their default dimensions are 125, 33 and 153, respectively. Each feature descriptor has a parameter that can define the resulting number of bins. In the case of PFH, it is the number (n) of subdivisions on each axis, and the resulting number of dimensions is n^3. For FPFH the number of subdivisions also defines the number of dimensions, but the bins on each axis are concatenated. The resulting number of dimensions is $n * 3$. For SI the parameter which defines the length of the descriptor is *image_width*. The resulting number of dimensions is $(image\_width + 1) * (2 * image\_width + 1)$. More information can be found in the cited articles. We reduce the dimensionality of feature descriptors in two ways: 1) modify their internal parameters, and 2) perform PCA on a computed feature with its original dimensionality.

Followed Guo et al. [4], to evaluate the descriptiveness of feature descriptors we use the Precision-recall curve (PRC). According to them, the PRC is more suitable for evaluating 3D point descriptors than widely used ROC. Creating a PRC consists of the following steps. We select *sample_num* points from both clouds as keypoints, with uniform sampling. It is important to note that there are many keypoint detectors, which are designed to detect keypoints that are highly distinctive (unique), repeatable, and stable [9]. With using keypoint detection algorithms instead of randomly selected keypoints, the descriptiveness evaluation of the feature descriptors results may vary. We think that by using randomly selected keypoints we avoid any bias. We compute the feature descriptors for the selected keypoints (when we use PCA, we compute the feature descriptor for all point in all point cloud, to obtain the dimensionality reduction transformation). To compute the feature descriptors we need to set a radius that defines the neighborhood of a point. As the radius increases the computational time increases too. We used the same radius with each feature descriptor. In the next step, we build a k-d tree to find the nearest neighbors of the features of the selected keypoints from the two clouds. We use the nearest neighbor distance ratio to decide if a point-pair considered is a match [8]. If the ratio between a feature's nearest and second nearest neighbor is less than a threshold τ, we consider the feature and its first neighbor a match. We use the dataset's ground truth transformations to decide if a match is really correct or not. If the two points are closer than a given ϵ threshold after the transformation, it is counted as a correct match. Finally, we compute the *precision* and *recall* of the matches. To generate the PCR we increase the threshold τ from 0.5 to 1 with 0.1 steps. It starts from 0.5 because there are only a small number of feature pairs that satisfy the condition that the difference between the point and its nearest neighbor and the point and its second neighbor is at least twice as much. In the case of real-world data, outlier points like this are usually considered as noise. The AUC denotes the area under the Precision-recall curve, which shows how descriptive is a feature descriptor, using different τ values to filter the matches, in a very compact way.

3 Evaluation

To evaluate the feature descriptors with different number of dimensions we used the 7-Scenes RGB-D dataset [3]. The dataset consists of point cloud fragments, with approx. 250000 points in each cloud. Before the evaluation, we performed the following preprocessing steps on the point clouds. 1) We downsampled the

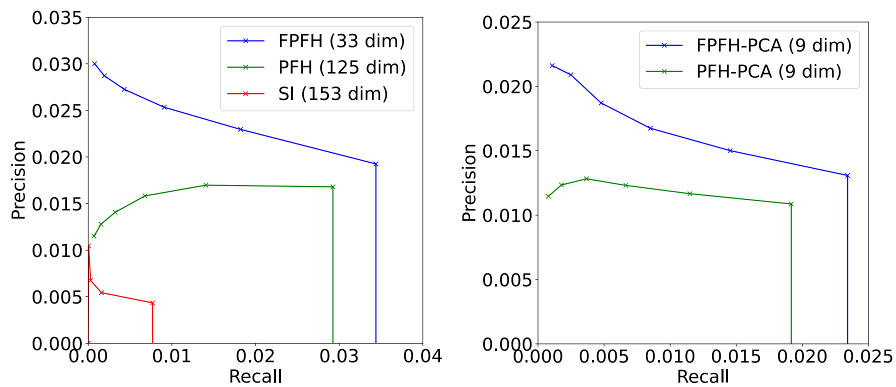

Fig. 1: Left: Precision-recall curve of 3 feature descriptor: FPFH, PFH, SI. Right: Precision-recall curve of FPFH and PFH. Both features were reduced to 9 dimensions.

point cloud using a voxel grid filter with 0.01 voxel leaf size for faster processing time. The filter reduced the number of points in each cloud fragment to approx. 100000 points. 2) We filtered out the outlier points using a radius-based filtering method: a point was removed if it did not have 10 neighbors in a 0.04 radius. This step removed approx. 100 points from each fragment at most. 3) Finally, we estimated the normal vectors at each point, because it is needed to compute the feature descriptors. For normal estimation we used 0.04 as radius. The runtimes of computing feature descriptors are different from each other and it depends on the feature radius. In this work, we do not investigate computational times. For every feature descriptor, we used the same radius (0.06). For evaluation, we worked on 10 overlapping point cloud fragment pairs. The *sample_num* was 5000. Because of the randomness of keypoint selection, every point cloud fragment pair was evaluated 10 times. The PRC and AUC values on the figures show aggregated results.

First, we evaluated the feature descriptors with their original number of dimensions. Figure 1 (left) shows the PRC of three feature descriptor: FPFH, PFH, SI. Based on our evaluation the Spin Image descriptor is significantly worse than the other two. Note that SI has the most number of dimensions. The FPFH has a higher AUC than PFH, but the difference between the two descriptors is decreasing as the value of τ increases. Because the Spin Image descriptor has significantly smaller AUC value, we decided to investigate the dimensionality reduction on PFH and FPFH only.

Figure 2 shows the AUC values of FPFH and PFH descriptors with a varying number of dimensions. The left columns denote dimension reduction with parameters of the descriptors. The right columns denote the dimension reduction with PCA. We used PCA on the feature descriptor with its original dimension. Therefore the first column has only one value. In the case of FPFH we could

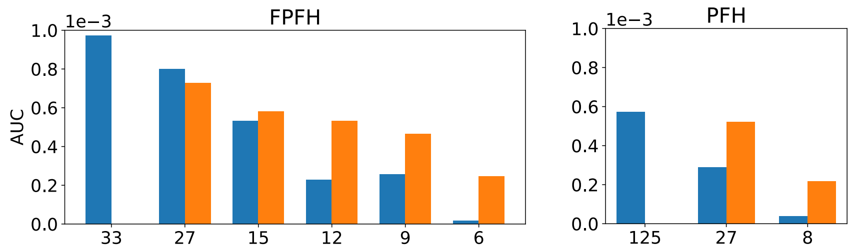

Fig. 2: Area Under Curve (AUC) of FPFH (left) and PFH (right) feature descriptors with different dimensions. The left columns denote the descriptors with reduced dimensionality using their internal parameters. The right columns denote dimensionality reduction using PCA.

compute the descriptor with many different dimensions: 33, 27, 15, 12, 9, and 6. Figure 2 (left) shows that from 15 dimensions FPFH-PCA gives better results. The difference between the AUC values becomes significant from 12 dimensions. If we need the use less than 15 dimensions for the FPFH, the PCA reduction is a better choice. Note that PCA computation adds time to the total runtime of computing and storing descriptors. Therefore, if the speed of runtime is critical, we should consider using FPFH without PCA with 15 dimensions despite FPFH-PCA is slightly better in this case also. With the parameters of PFH, we could only compute the feature descriptor with 3 different dimensions: 125, 27, and 8. In Figure 2 (right), we want to highlight that the original descriptor with 125 dimensions has almost the same AUC value as the FPFH-PCA with 27 dimensions. The reason is, that for an ordinary point a PFH descriptor with 125 dimensions contains many zero values. In the case of PFH the PCA versions give significantly better results with 27 dimensions already. On Figure 1 (right) we computed both descriptors with 9 dimensions. This figure is very similar to Figure 1 (left). Based on our evaluation the FPFH-PCA gives better results than PFH-PCA with every number of dimensions.

4 Conclusion

Storing and indexing feature descriptors are difficult problems, especially when nearest neighbor searches are important. The biggest challenge is to handle descriptors with a high number of dimensions. In this work, we investigated how the descriptiveness of feature descriptors changes when we reduce their dimensionality. We reduced the dimensionality in two ways: modifying the descriptor's internal parameters and apply PCA dimension reduction technique. Based on our evaluation it is better to reduce the dimensionality with PCA, and FPFH is the more descriptive even with a low number of dimensions.

Acknowledgements The authors thank the support of project "Application Domain Specific Highly Reliable IT Solutions" that has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

1. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: International conference on database theory. pp. 217–235. Springer (1999)
2. Fukunaga, K., Olsen, D.R.: An algorithm for finding intrinsic dimensionality of data. IEEE Transactions on Computers **100**(2), 176–183 (1971)
3. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time rgb-d camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 173–179 (2013)
4. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.: A comprehensive performance evaluation of 3d local feature descriptors. International Journal of Computer Vision **116**, 66–89 (04 2015)
5. Holz, D., Ichim, A.E., Tombari, F., Rusu, R.B., Behnke, S.: Registration with the point cloud library: A modular framework for aligning in 3-d. IEEE Robotics Automation Magazine **22**(4), 110–124 (2015)
6. H'roura, J., Roy, M., Mansouri, A., Mammass, D., Juillion, P., Bouzit, A., Méniel, P.: Salient spin images: A descriptor for 3d object recognition. In: Mansouri, A., El Moataz, A., Nouboud, F., Mammass, D. (eds.) Image and Signal Processing. pp. 233–242 (2018)
7. Korn, F., Pagel, B.U., Faloutsos, C.: On the" dimensionality curse" and the" self-similarity blessing". IEEE Transactions on Knowledge and Data Engineering **13**(1), 96–111 (2001)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**, 91–110 (11 2004)
9. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision **89**, 348–361 (09 2010)
10. Prakhya, S., Liu, B., Lin, W., Li, K., Xiao, Y.: On creating low dimensional 3d feature descriptors with pca. In: TENCON 2017 - 2017 IEEE Region 10 Conference. pp. 315–320 (11 2017)
11. Rusu, R., Marton, Z., Blodow, N., Beetz, M.: Persistent point feature histograms for 3d point clouds. Proc 10th Int Conf Intel Autonomous Syst (IAS-10) **16** (01 2008)
12. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3212–3217 (2009)
13. Samet, H.: Foundations of multidimensional and metric data structures. Morgan Kaufmann (2006)
14. Spezialetti, R., Salti, S., Di Stefano, L.: Performance evaluation of learned 3d features. In: Image Analysis and Processing – ICIAP 2019. pp. 519–531 (2019)
15. Weber, R., Schek, H.J., Blott, S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB. vol. 98, pp. 194–205 (1998)