POINT DENSITY-INVARIANT 3D OBJECT DETECTION AND POSE ESTIMATION

Su-A Kim

Intel Visual Computing Institute
Saarland Informatics Campus
Germany

Kuk-Jin Yoon

Gwangju Institute of Science and Technology
School of Electrical Engineering and Computer Science
Republic of Korea

ABSTRACT

For 3D object detection and pose estimation, it is crucial to extract distinctive and representative features of the objects and describe them efficiently. Therefore, a large number of 3D feature descriptors has been developed. Among these, Point Feature Histogram RGB (PFHRGB) has been evaluated as showing the best performance for 3D object and category recognition. However, this descriptor is vulnerable to point density variation and produces many false correspondences accordingly. In this paper, we tackle this problem and propose an algorithm to find the correct correspondences under the point density variation. Experimental results show that the proposed method is promising for 3D object detection and pose estimation under the point density variation.

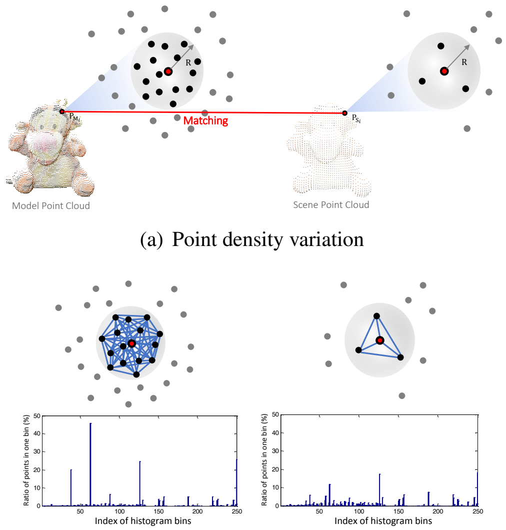
***Index Terms*—** 3D object detection and pose estimation

1. INTRODUCTION

A task of 3D object detection and pose estimation is finding the object instances in a scene and calculating transformation which best aligns them into the scene. Typical approach for 3D object detection and pose estimation is 3D local feature-based approach which describes the local surfaces around the particular keypoints and finds the correspondence points by pair-to-pair feature matching.

Several researches have been devoted to the study of 3D local feature descriptors [1, 2, 3] because how to describe the object well is a crucial part. According to [2] which comparatively evaluated both 3D local and global feature descriptors for 3D object and category recognition, Point Feature Histogram RGB (PFHRGB) [4] consequentially obtained the best results in a common dataset. Besides, it is one of the 3D local feature descriptors that many researchers have used for various purposes [5, 6, 7, 8].

However, this feature histogram is vulnerable to density variation of point clouds. Point density variation is caused by two factors: different distances from the sensor and viewpoint variation. Under the point density variation, even true correspondences can have different feature histograms because of different distributions of neighboring points. Thus, the increased difference between two histograms makes the possibility of being a false correspondence and adversely affects performance of 3D object detection and pose estimation. But handling the point density variation between the model point cloud and the scene point cloud has been received relatively little attention so far.



(a) Point density variation

(b) At dense point cloud (c) At sparse point cloud

Fig. 1. Point density variation makes the difference of the feature histogram. (a) Two points which are at the same position but having different density, (b, c) top: the lines between all pairs of neighbors, bottom: the histograms on the two points

In this paper, we propose an algorithm to find the correct correspondences under the point density variation for 3D object detection and pose estimation. We generate multi-scale features on the model point cloud and select a right scale by measuring the similarity of the density about each point at the matching stage. We evaluate the proposed method using the dataset with the point density variation. Our method gives high accuracy in the phase of feature matching, therefore, the overall results of pose estimation and detection show stable and low errors. To the best of our knowledge, it is the first pa-

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A01005455). Since April 2017 the first author has been supported by the European Commission through the H2020-MSCA Distributed 3D Object Design (Grant No. 642841).

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

per that focuses on the point density variation of the 3D local feature descriptor.

2. POINT DENSITY VARIATION

We describe the features of local surface around the keypoints using Point Feature Histogram RGB (PFHRGB). PFHRGB is an extended version of Point Feature Histogram (PFH) [4] with color information. It consists of angular and photometric features. The features which combined color and shape information achieved the best performance compared to the features which only used shape information [2].

PFHRGB is one of the best 3D feature descriptors, however, it is vulnerable to 'point density variation'. The point density is defined by the number of the points inside a fixed sphere when selecting a set of *k*-neighbor points around the keypoints. Even though two point clouds are from the same objects, the density of the point clouds varies with two different situations; First of all, let two point clouds be *captured from different distances from the sensor*. Since the point density gets sparse as distance from the sensor increases, we get two point clouds with the point density variation as a result. In a general situation, a model object is captured in a very short distance from the sensor, and a target object in the scene is further than the model object. Moreover, when we *capture the same object with different viewpoint* and then turn a point cloud as the same viewpoint with the other, we can observe the different density in the same points.

PFHRGB defines the neighbors around keypoints using a radius search, and then creates the feature histograms considering the relationship between all bi-directional pairs of the neighbors. Fig. 1 represents that even two points which are at the same position can have different feature histograms because of their different neighboring points. To handle the density variation, the increments of the histogram are normalized by the number of bi-directional combinations of the neighbors $2 \cdot \binom{k}{2}$, where k is the number of the neighbors. However, this is not sufficient to completely handle the problem because the sparse point cloud has little information in comparison with the dense point cloud. Under the point density variation, some distinctive neighbor points, which the dense point cloud has but the sparse point has not, make the difference of the feature histograms. Thus, the increased distance between two histograms makes the possibility of being a false correspondence and adversely affects performance of 3D object detection and pose estimation.

3. PROPOSED METHOD

3.1. Multi-scale Feature Representation

To reduce the influence of the point density variation, we have to make the neighbors between two point clouds having the different density similar. With respect to doing this, two approaches, up-sampling or down-sampling, can be considered. Up-sampling is to make virtual points from sparse points using interpolation techniques. Accordingly, there is no guarantee to make the virtual points same or similar with the dense point cloud. This uncertainty can generate more adversely effects than before. On the contrary, down-sampling is to filter out the points, therefore, we can make the dense point cloud to get sparse. For down-sampling the 3D point cloud, a voxel-grid down-sampling method creates a 3D voxel grid such as a tiny 3D box over the point cloud and then all the points in each voxel are approximated with their centroid. Consequently, the point density gets sparse as the voxel grid size increases. However, since we do not have any prior information about the point density of an target object in the scene point cloud, we cannot guess appropriate scale for down-sampling the dense point cloud.

Algorithm 1 Proposed Algorithm

```
Input: $\mathcal{M} = \{P_M,\ K_M\}, \mathcal{S} = \{P_S,\ K_S\}, N_{scale}$
Output: A set of initial correspondences $\mathcal{C}_{\mathcal{S}\rightarrow\mathcal{M}}$
Initialization Step:
    $\{H_M,\ \varphi_M\} \leftarrow$ MultiScalePFHRGB($P_M,\ K_M$)
    $\{H_S,\ \varphi_S\} \leftarrow$ PFHRGB($P_S,\ K_S$)
for $j \leftarrow 1$ to $N_S$ do
    for $i \leftarrow 1$ to $N_M$ do
        for $k \leftarrow 1$ to $N_{scale}$ do
            $d_{\varphi}^{k} = |\varphi_{M_k}^{i} - \varphi_{S}^{j}|$
        end for
        $idx_1 \leftarrow$ SelectIdxMin($d_\varphi$)
        $idx_2 \leftarrow$ SelectIdxSecondMin($d_\varphi$)
        $H_{Dist_1}^{i} \leftarrow$ HistogramDist($H_{M_{idx_1}}^{i},\ H_S^j$)
        $H_{Dist_2}^{i} \leftarrow$ HistogramDist($H_{M_{idx_2}}^{i},\ H_S^j$)
    end for
    $\{minVal_1,\ minIdx_1\} \leftarrow$ min($H_{Dist_1}$)
    $\{minVal_2,\ minIdx_2\} \leftarrow$ min($H_{Dist_2}$)
    if $minVal_1 < minVal_2$ then
        $C_{\mathcal{S}\rightarrow\mathcal{M}}^{j} \leftarrow minIdx_1$
    else
        $C_{\mathcal{S}\rightarrow\mathcal{M}}^{j} \leftarrow minIdx_2$
    end if
end for
    ▷ $P$ is a point cloud, $K$ is position of keypoints, $H$ is a feature
    histogram (PFHRGB).
```

In this paper, we propose multi-scale features around the keypoints in the model point cloud as described in the initialization step of Algorithm 1. First, we detect a set of keypoints $K_M \in \mathbb{R}^3$ on the model point cloud P_M using 3D SIFT keypoints [9, 10]. Second, using the voxel-grid downsamping method, we down-sample the model point cloud P_M at coarse levels of scale N_{scale}, where N_{scale} is the number of the scales. And then, we describe a PFHRGB descriptor H_M^i at different scale levels of each keypoint $i \in \{1,\ \cdots,\ N_M\}$, where N_M is the number of keypoints. During this description phase with PFHRGB, the point density on each keypoint is calculated by the number of the points existing inside a fixed sphere as $\varphi_M^i \in \{\varphi_{M_1}^i,\ \varphi_{M_2}^i,\ \cdots,\ \varphi_{M_{N_{scale}}}^i\}$.

3.2. Scale Selection

Before finding the correspondences between two point clouds, we propose a pre-processing step of feature matching as described in line 3-7 of Algorithm 1. It is about measuring the similarity of the density between each point on the model point cloud and the scene point cloud. This process gives high accuracy of feature matching. And it is computationally efficient as compared to considering all scales at once.

Let φ^{i}_{M} and φ^{j}_{S} be the point density of each point on the model point cloud and the scene point cloud, where $i \in \{1, \cdots, N_M\}$ and $j \in \{1, \cdots, N_S\}$. N_S is the number of keypoints in the scene point cloud. We measure the similarity of the density by loss function $d^{k}_{\varphi} = |\varphi_{M_k} - \varphi_{S}|$, where $k \in \{1, \cdots, N_{scale}\}$. Then, we choose two scale indexes idx_1 and idx_2 which have the smallest and the second smallest among a set of the losses d_{φ}. The reason we choose two scales is that even though two points have the most similar density, their feature histograms may not be the most similar. After two scale indexes are selected, we compare the similarity of the PFHRGB descriptor as described in line 8-9 of Algorithm 1. And then, a correspondence $C^{j}_{\mathcal{S}\to\mathcal{M}}$ is chosen as a point having the closest distance as described in line 11-17 of Algorithm 1.

3.3. 3D Object Detection and Pose Estimation

A set of the correspondences $C_{\mathcal{M}\to\mathcal{S}}$ is found by comparing the PFHRGB descriptors between the model point cloud and the scene point cloud as well. Then, we get initial correspondences $\mathcal{C} = \{C_{\mathcal{M}\to\mathcal{S}},\ C_{\mathcal{S}\to\mathcal{M}}\}$. But the correspondences $\mathcal{C}$ have uncertain correspondences which negatively affect on performance of 3D object detection and pose estimation. Therefore, we sort out the correspondences that are probably false. Given the initial correspondences $\mathcal{C}$, we only select the correspondences which have same index by checking the bi-directional consistency.

Given a set of the consistent correspondences $\widehat{C}$, we estimate 6 DoF initial pose of each object instance in the scene point cloud. We can express the relationship of the correspondences by $\mathbf{P}^{\widehat{C}}_{\mathcal{S}} = \mathbf{R}\mathbf{P}^{\widehat{C}}_{\mathcal{M}} + \mathbf{t}$, where $\mathbf{P}^{\widehat{C}}_{\mathcal{M}} \in \mathbb{R}^3$ and $\mathbf{P}^{\widehat{C}}_{\mathcal{S}} \in \mathbb{R}^3$ are the 3D homogeneous coordinates of the correspondences $\widehat{C}$, $\mathbf{R} \in SO(3)$ is a rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. To find the optimized $\mathbf{R}$ and $\mathbf{t}$ about all pairs of the correspondences, we use Least Squares Estimation method [11].

The initial pose is possibly misaligned because it is calculated by using only some points which are the correspondences in the point cloud. Therefore, we refine the misaligned pose by Iterative Closest Point (ICP) algorithm [12] using all points in the point cloud. First, we align the model point cloud with the scene point cloud using the initial pose and iteratively revise the transformation to minimize the residual error until the pre-defined iteration number is ended.

4. EXPERIMENTS

In this section, we present comparative experiments between Rusu *et al.* [4] and our approach. The experiments are designed for the evaluation of performance in four different experiments including feature matching, pose estimation with respect to different distances from the sensor and viewpoint variation, and object detection in general scenes.

4.1. Dataset

3D object detection and 6 DoF pose estimation are being actively studied, accordingly there are publicly available datasets for evaluating this and related problem [13]. Unfortunately, the datasets with the point density variation do not exist. In the case of SHOT dataset [14], it is well-suited to evaluate 3D descriptor matching and object pose estimation. However, since the models and the scenes were captured in very similar distance from the sensor, this dataset is not suitable for our purpose. Therefore, we construct a new dataset with the point density variation. More details about the dataset will be described in a project page. (`https://sites.google.com/site/suakimpf/icip17`)

4.2. Evaluation Criteria

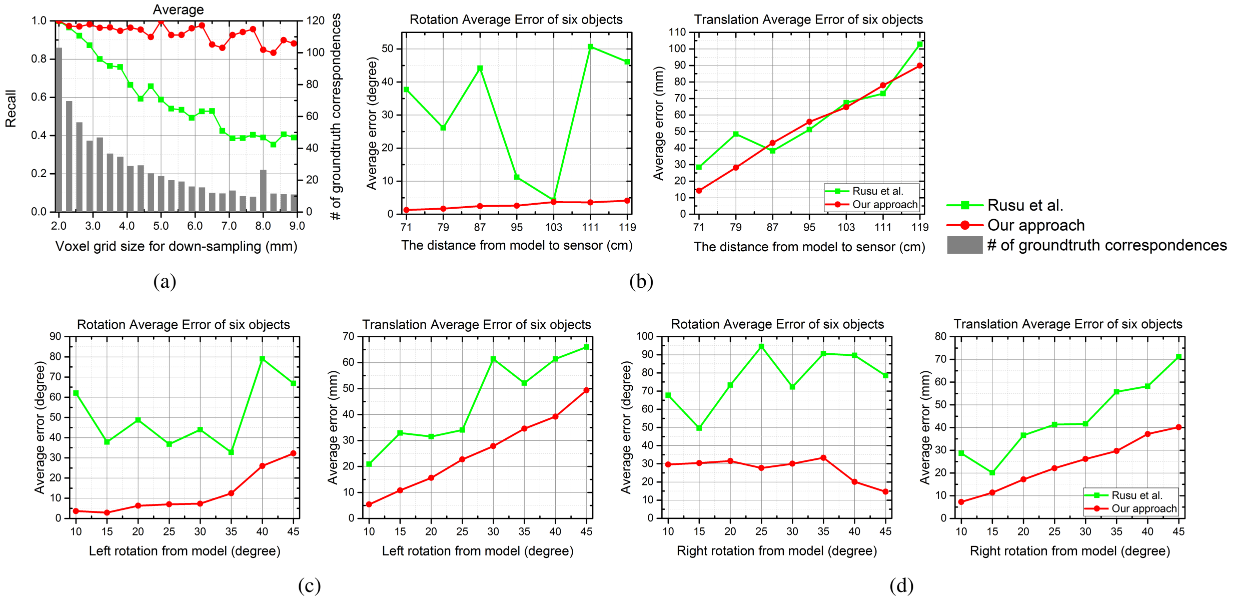
We process quantitative evaluation except the detection experiment which is evaluated by qualitative analysis. First, feature matching experiment has to be evaluated by matching accuracy which means how well the correspondences are found. We use the recall metric as represented by Eq. (1), where correct correspondences are the estimated correspondences as well as satisfied with the ground-truth correspondences.

$$Recall = \frac{\sharp\ of\ the\ correct\ correspondences}{\sharp\ of\ the\ groundtruth\ correspondences} \quad (1)$$

Furthermore, we design the evaluation setup for the pose estimation experiments. Let $C_{GT} \in SE(3)$ and $C_S \in SE(3)$ be the camera extrinsic matrices in the ground-truth scene and any test scene. These matrices are calculated by using the fixed check board pattern as the world coordinate. And $\mathbf{T}_{GT} \in SE(3)$ is the ground-truth object pose which is defined as the well-aligned pose by enough ICP iterations between the two point clouds captured in very similar distance or viewpoint. $\mathbf{T}_{est} \in SE(3)$ is the estimated object pose in any test scene.

$$\mathbf{T}_{err} = |\mathbf{T}^{-1}_{GT} \cdot C_{GT} \cdot C^{-1}_{S} \cdot \mathbf{T}_{est}| \quad (2)$$

The estimated object pose $\mathbf{T}_{est}$ is evaluated by the relative pose against the ground-truth object pose $\mathbf{T}_{GT}$ as represented in Eq. (2). $\mathbf{T}_{err} \in SE(3)$ includes $\mathbf{R}_{err} \in SO(3)$ and $\mathbf{t}_{err} \in \mathbb{R}^3$. A rotation matrix $\mathbf{R}_{err}$ is transformed to a rotation vector by Rodrigues formula. Conclusively, we define *a rotation error* as an angle from the rotation vector and *a translation error* as L-2 norm of the translation vector $\mathbf{t}_{err}$.



(a)

(b)

(c)

(d)

Fig. 2. (a) Recall curves for feature matching (b) Average error of the estimated pose with respect to different distances from the sensor (c, d) Average error of the estimated pose with respect to viewpoint variation (left and right rotation variation)

4.3. Experimental Results

The results of **feature matching** are presented in Fig. 2 (a). To show the results with reliability of the experiments, bar graph represents the number of the ground-truth correspondences and the curves are drawn by varying the voxel grid size for down-sampling. Our approach outperforms Rusu *et al.* [4] showing the results of accurate feature matching even if the point density variation exists.

The results of **pose estimation to different distances from the sensor** are shown in Fig. 2 (b). We can observe the average pose error of the six objects. While Rusu *et al.* shows unstable result as a whole, performance of our approach is promising. In the cases which the distances from model to sensor are 87, 95, 111 cm, the rotation errors of our approach are low, but the translation errors are relatively higher than Rusu *et al.*. Because, in the case of Rusu *et al.*, the model and scene point cloud are aligned with obviously wrong rotation and their centroids are closer.

The results of **pose estimation to viewpoint variation** are shown in Fig. 2 (c, d). The graphs present the average pose errors of the six objects with respect to viewpoint variation. Rusu *et al.* has unstable and higher errors than our approach. But the errors of right rotation from model especially show high tendency in both approaches. Because, since the right sides of *Juice* are very similar with the front side, many false positives of feature matching are made. Consequently, it causes negative effects to the average results of the right rotation experiment.

The results of **detection** are presented in Fig. 3 with the selected results among the 15 test scenes which are general scenes with the point density variation, occlusion, and clutter. If the estimated object poses are close enough to the true pose, we judge them as the detected objects. While Rusu *et al.* recognizes at most two objects per scene which are even false positives, our approach does averagely more than half of the test objects. This experiment demonstrates that our approach can recognize the object well in general scenes.

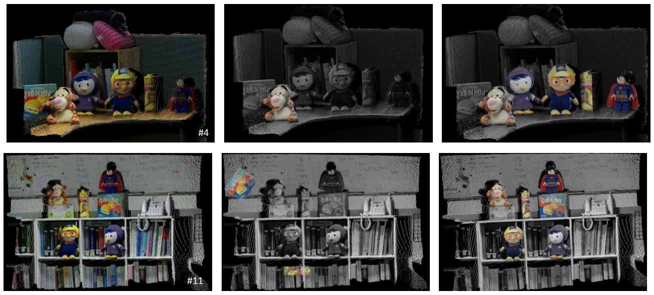


Fig. 3. Selected detection results(#4, #11) of Rusu *et al.* [4] (second column) and our approach (third column). The first column shows the images of test scenes. Only detected objects are depicted in color point clouds in the second and third columns. (Best viewed in color)

5. CONCLUSION

We posed the problem of the point density variation in 3D object detection and pose estimation. To solve this problem, we proposed multi-scale feature representation and similarity measure of the point density. The proposed method shows stable and low errors under the point density variation. Additionally, the average computation time of our approach, which is required to estimate the correspondences and the initial pose of each object, is similar to that of Rusu *et al.* (Rusu *et al.*: 2.161 sec, Our approach: 2.164 sec).

6. REFERENCES

[1] Changhyun Choi and Henrik I Christensen, "3D Pose Estimation of Daily Objects Using an RGB-D Camera," in *IROS*, 2012.

[2] Luís A Alexandre, "3D Descriptors for Object and Category Recognition: A Comparative Evaluation," in *IROS Workshops*, 2012.

[3] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic, "Model Globally, Match Locally: Efficient and Robust 3D Object Recognition," in *CVPR*, 2010.

[4] Radu Bogdan Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[5] Janusz Bedkowski, Karol Majek, and Andreas Nüchter, "General purpose computing on graphics processing units for robotic applications," *Journal of Software Engineering for Robotics (JOSER)*, vol. 4, no. 1, pp. 23–33, 2013.

[6] Jesus Martínez-Gómez, Miguel Cazorla, Ismael García-Varea, and Cristina Romero-González, "Object categorization from rgb-d local features and bag of words," in *Robot 2015: Second Iberian Robotics Conference*. Springer, 2016, pp. 635–644.

[7] Sudhanshu Mittal, "Small object discovery and recognition using actively guided robot.," in *ICPR*, 2014.

[8] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[9] Sılvio Filipe and Luıs A Alexandre, "A Comparative Evaluation of 3D Keypoint Detectors in a RGB-D Object Dataset," in *International Conference on Computer Vision Theory and Applications*, 2014.

[10] David G Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[11] Donald W Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[12] Paul J Besl and Neil D McKay, "Method for Registration of 3-D Shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992.

[13] Michael Firman, "Rgbd datasets: Past, present and future," *arXiv preprint arXiv:1604.00999*, 2016.

[14] Samuele Salti, Federico Tombari, and Luigi Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *CVIU*, vol. 125, pp. 251–264, 2014.