Accuracy analysis of 3D object reconstruction using point cloud filtering algorithms

A N Ruchay[1,2]**, K A Dorofeev**[2] **and V V Kalschikov**[2]

[1]Federal Research Centre of Biological Systems and Agro-technologies of the Russian Academy of Sciences, 9 Yanvarya street, 29, Orenburg, Russia, 460000

[2]Chelyabinsk State University, Bratiev Kashirinykh street 129, Chelyabinsk, Russia, 454001

e-mail: ran@csu.ru, kostuan1989@mail.ru, vkalschikov@gmail.com

Abstract. In this paper, we first analyze the accuracy of 3D object reconstruction using point cloud filtering applied on data from a RGB-D sensor. Point cloud filtering algorithms carry out upsampling for defective point cloud. Various methods of point cloud filtering are tested and compared with respect to the reconstruction accuracy using real data. In order to improve the accuracy of 3D object reconstruction, an efficient method of point cloud filtering is designed. The presented results show an improvement in the accuracy of 3D object reconstruction using the proposed point cloud filtering algorithm.

1. Introduction

The 3D object reconstruction is a popular task in the field of medicine, agriculture, architecture, games, and film industry [1, 2, 3]. Accurate 3D object reconstruction is an important aspect for object recognition, object retrieval, scene understanding, object tracking, virtual maintenance and visualization [4, 5, 6, 7].

RGB-D low-cost sensors such as the Kinect can provide a high-resolution RGB color image with a depth map of the environment [8, 9, 10]. The depth map discontinuity, and a small error around object boundary may lead to significant ringing artifacts in rendered views. Also, the depth map provided by a RGB-D camera is often noisy due to imperfections associated with infrared light reflections, and missing pixels without any depth value appearing as black holes in the maps. Therefore, the point cloud obtained from the depth map inevitably suffers from noise contamination and contains outliers.

The noise and holes can greatly affect to the accurate of 3D reconstruction [11, 12], therefore, noise-reduction and hole-filling enhancement algorithms are intended to serve as a pre-processing step for 3D reconstruction systems using Kinect cameras [13, 14, 15, 16]. To reduce impulsive noise and to fill small holes, the filters [17, 18, 19, 20, 21] are used.

In this paper, we are interested in the design of a filtering algorithm of a poin cloud to improve the quality of the 3D reconstruction. In recent years, a large number of methods contributing to 3D point cloud filtering have been proposed: Normal-based Bilateral Filter [22], Moving Least Square [22], Iterative guidance normal filter [23], Bilateral Filter [24], Density-based Denoising [25], Rolling normal filter [26], Statistical Outlier Removal filter [27], Radius Outlier Removal filter [27], Voxel Grid filter [22], 3D Bilateral filter [28].

In a common approach of noise reduction it is supposed that the raw point cloud contains the ground truth, distorted by artificial noise such as additive [29, 30]. Although this common approach could be used for quantitative comparison (e.g., PSNR, MSE, etc), common methods reduce only artificial noise but not original noise contained in the raw point cloud. In this paper, we consider denoising point cloud algorithms for 3D object reconstruction. We propose a denoising method using a point cloud as the input. We also evaluate the performance of denoising methods on the base of the accuracy of 3D object reconstruction. Actually, RSME errors by ICP algorithm and hausdorf distance [31] between input cloud and filtered cloud are calculated.

General denoising methods are not designed to clean coarse noise contained in the input point cloud. Therefore, our main goal is to evaluate denoising methods in terms of reconstruction accuracy which depends on the quality of the input point cloud.

The paper is organized as follows. Section 2 discusses related denoising point cloud methods. Computer simulation results are provided in Section 3. Finally, Section 4 summarizes our conclusions.

2. Point cloud denoising filters

This section contains information about point cloud denoising filters.

The paper [22] presents a good survey of filtering approaches for 3D point cloud. We compare the following point cloud denoising algorithms in terms of accuracy of 3D object reconstruction: Statistical Outlier Removal filter (SOR) [27], Radius Outlier Removal filter (ROR) [27], Voxel Grid filter (VG) [22], 3D Bilateral filter (3DBF) [28].

2.1. Statistical Outlier Removal filter (SOR)

SOR uses point neighborhood statistics to filter outlier data [27]. Sensor scans typically generate point cloud datasets of varying point densities. Additionally, measurement errors lead to sparse outliers which corrupt the results even more. This complicates the estimation of local point cloud characteristics such as surface normals or curvature changes, leading to erroneous values, which in turn might cause point cloud registration failures. Some of these irregularities can be solved by performing a statistical analysis on each points neighborhood, and trimming those which do not meet a certain criteria. This sparse outlier removal is based on the computation of the distribution of point to neighbors distances in the input dataset. For each point, we compute the mean distance from it to all its neighbors. By assuming that the resulted distribution is Gaussian with a mean and a standard deviation, all points whose mean distances are outside an interval defined by the global distances mean and standard deviation can be considered as outliers and trimmed from the dataset.

2.2. Radius Outlier Removal filter (ROR)

ROR removes outliers if the number of neighbors in a certain search radius is smaller than a given K [27]. We can specifie a number of neighbors which every index must have within a specified radius to remain in the point cloud.

2.3. Voxel Grid filter (VG)

VG filtering method first defines a 3D voxel grid (3D boxes in 3D space) on a point cloud. Then, in each voxel, a point is chosen to approximate all the points that lie on that voxel. Normally, the centroid of these points or the center of this voxel is used as the approximation. The former is slower than the later, while its representative of underlying surface is more accurate. The VG method usually leads to geometric information loss.

2.4. 3D Bilateral filter (3DBF)

3DBF filter denoises a point with respect to its neighbors by considering not only the distance from the neighbors to the point but also the distance along a normal direction [28].

Let us first consider a point cloud M with known normals n_v at each vertex position v. Let $N(v)$ be the 1-ring neighborhood of vertex v (i.e. the set of vertices sharing an edge with v). Then, the filtered position of v is

$$v + \delta v \cdot n_v,$$

where

$$\delta v = \frac{\sum_{p \in N(v)} w_d(\|p - v\|) w_n(|\langle n_v, p - v\rangle|) \langle\, n_v, p - v\rangle}{\sum_{p \in N(v)} w_d(\|p - v\|) w_n(|\langle n_v, p - v\rangle|)},$$

and w_d and w_n are two decreasing functions. Here vertex v is shifted along its normal toward a weighted average of points that are both close to v in the ambient space and close to the plane passing through v with normal n_v.

3. Experimental results

In this section, we evaluate the performance of the tested denoising methods in terms of reconstruction accuracy which depends on the quality of the input point cloud.

We compare the following point cloud denoising algorithms in terms of the accuracy of 3D object reconstruction and speed: Statistical Outlier Removal filter (SOR) [27], Radius Outlier Removal filter (ROR) [27], Voxel Grid filter (VG) [22], 3D Bilateral filter (3DBF) [28]. SOR, ROR, and VG are available in the Point Cloud Library (PCL). 3DBF was implemented in C++. The experiments are carried out on a PC with Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz and 16 GB memory.

In our experiments we use the point clouds of a lion and an apollo from dataset [32] and a chair from database [33]. Fig. 1 shows RGB images and depth maps of a lion, an apollo, and a chair.

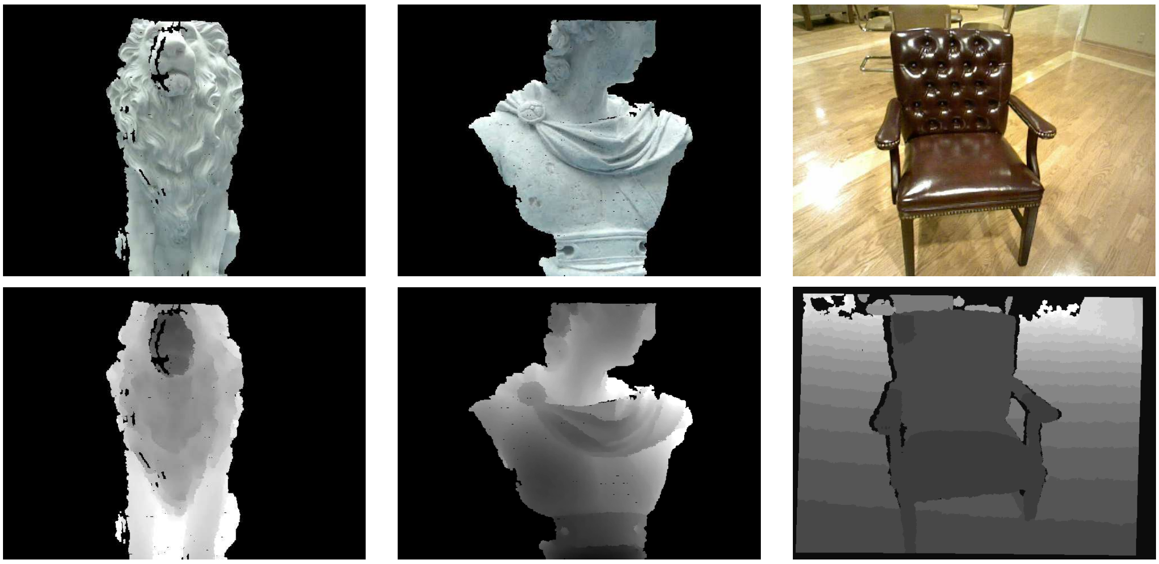

Figure 1. RGB images and depth maps of a lion, an apollo, and a chair are scanned by Kinect sensor.

We construct couples of point clouds for each model using the following steps:

(i) Registration RGB and depth data (Fig. 1).

(ii) Making point clouds (Fig. 2).

(iii) Computing point cloud statistics, such as points count, minimum, maximum and median distance between points in point cloud (Table. 1). Point clouds statistics is required for further calculation of optimum parameters of 3D filters.

(iv) Metric calculation algorithms between couples of frames of point clouds. We calculate transorfmation matrix with standart ICP algorithm and euclidian fitness score (ICP error). Since filtered point cloud can contain different points count of rather initial cloud, we also calculate the hausdorf distance between initial and filtered clouds for estimation of the quality of 3D filtration.

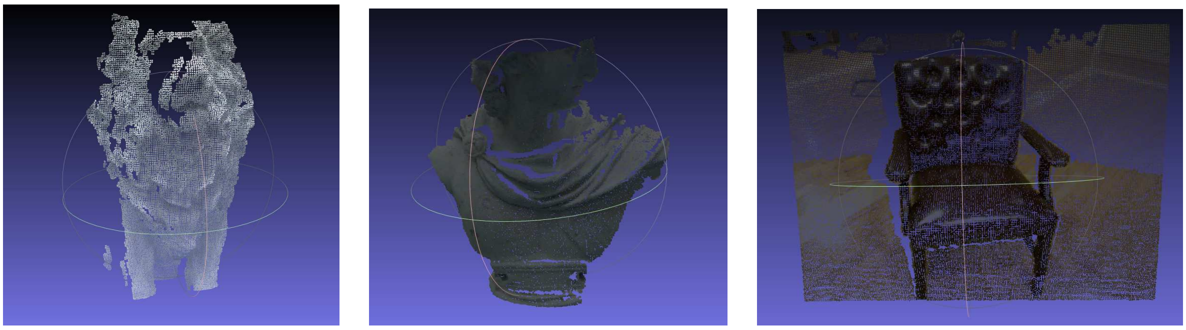

Figure 2. Obtained point clouds of a lion, an apollo, and a chair.

Table 1. Point clouds statistics.

Model	Points count	Min	Max	Median
chair	98593	0.001788	5.962374	0.013808
lion	22603	0.000837	0.670678	0.032839
apolo	30038	0.001030	0.579652	0.010399

Result of calculation and visualization of the hausdorf metric between two frames of chair model shown in Fig. 3.

The metrics between two frames of each model without filtering are presented in Table 2.

Table 2. Calculated metrics between two point clouds of each model without filtering.

Model	ICP error	Hausdorf distance
chair	3.30E-04	0.593978345
lion	6.50E-05	0.078128666
apollo	4.90E-05	0.069139026

The corresponding ICP error and hausdorf distance calculated for chair model with SOR, ROR, VG, 3DBF point cloud denoising algorithms are shown in Table 3. The ROR filter yields the best result in terms of ICP error and hausdorf distance evaluation among all point cloud denoising algorithms. Fig. 4 shows the point clouds of a chair after denoising ROR and SOR filtering.

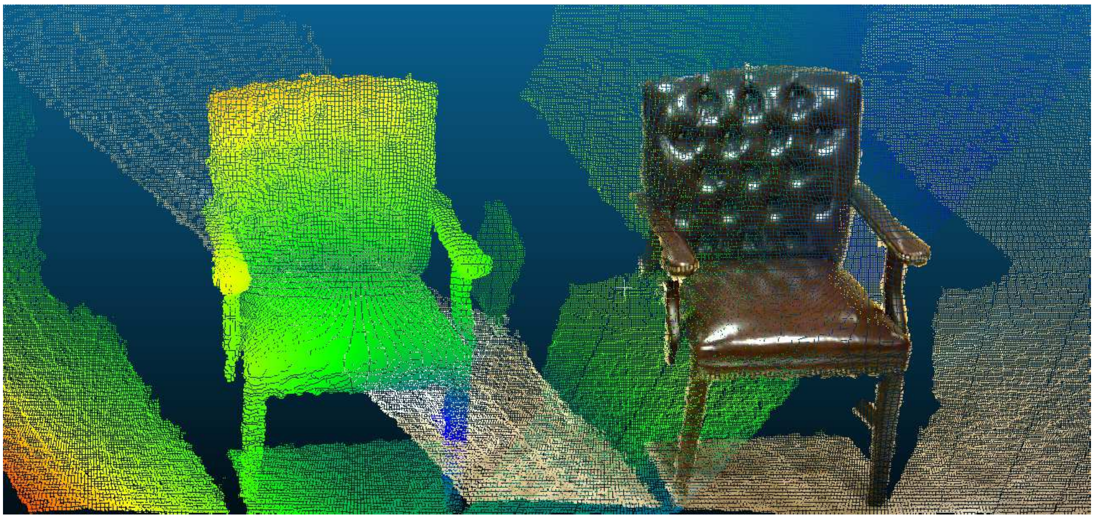

Figure 3. Visualization of hausdorf metric between two point clouds of chair model.

Table 3. Results of measurements using a common ICP algorithm and the hausdorf metric after 3D filtering for chair model.

Filter	Points count	Param1	Param2	ICP error	hausdorf distance
SOR	64405	3	0.11	3.5E-04	0.572
ROR	65448	3	0.007	4.8E-04	0.593
VG	62246	0.009		4.8E-04	0.591
3DBF	87090	0.05	0.001	3.6E-04	0.596

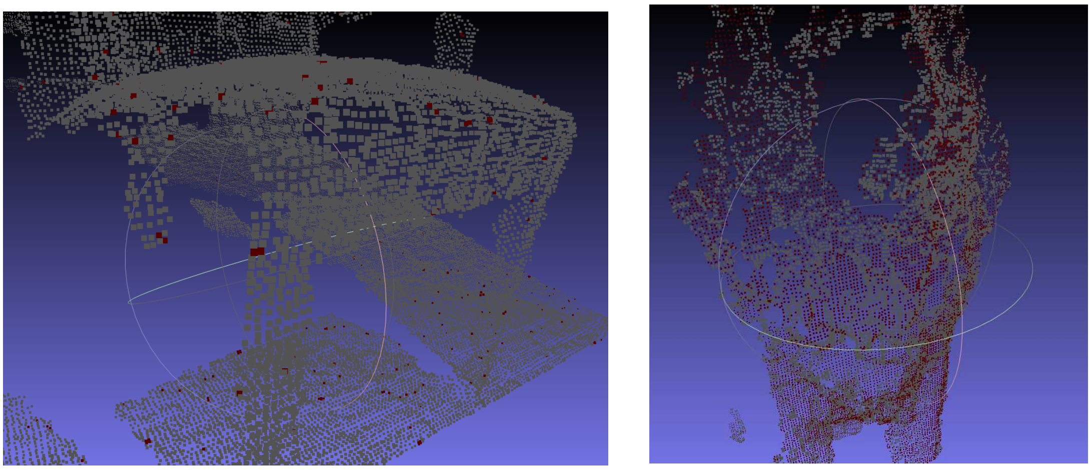

Figure 4. Results of filtering by ROR and SOR filters between two point clouds of chair model.

4. Conclusion

In this paper, we compared various point cloud algorithms in terms of accuracy of 3D object reconstruction using real data from a RGB-D sensor. The experiment has shown that the ROR filter yields the best result in terms of ICP error and hausdorf distance evaluation among all point cloud denoising algorithms.

5. References

[1] Echeagaray-Patron B A, Kober V I, Karnaukhov V N and Kuznetsov V V 2017 *Journal of Communications Technology and Electronics* **62** 648-652

[2] Ruchay A, Dorofeev K and Kober A 2018 *Proc. SPIE* **10752** 1075222-8

[3] Ruchay A, Dorofeev K and Kolpakov V 2018 Fusion of information from multiple Kinect sensors for 3D object reconstruction *Computer Optics* **42(5)** 898-903 DOI: 10.18287/2412-6179-2018-42-5-898-903

[4] Echeagaray-Patron B A and Kober V 2015 *Proc. SPIE* **9598** 95980V-8

[5] Echeagaray-Patron B A and Kober V 2016 *Proc. SPIE* **9971** 9971-6

[6] Ruchay A, Dorofeev K and Kober A 2018 *CEUR Workshop Proceedings* **2210** 82-88

[7] Ruchay A, Dorofeev K and Kober A 2018 *CEUR Workshop Proceedings* **2210** 300-308

[8] Tihonkih D, Makovetskii A and Kuznetsov V 2016 *Proc. SPIE* **9971** 99712D-8

[9] Nikolaev D, Tihonkih D, Makovetskii A and Voronin S 2017 *Proc. SPIE* **10396** 10396-8

[10] Gonzalez-Fraga J A, Kober V, Diaz-Ramirez V H, Gutierrez E and Alvarez-Xochihua O 2017 *Proc. SPIE* **10396** 10396-7

[11] Makovetskii A, Voronin S and Kober V 2018 *Proceedings of SPIE - The International Society for Optical Engineering* **10752** 107522V

[12] Voronin S, Makovetskii A, Voronin A and Diaz-Escobar J 2018 P*roceedings of SPIE - The International Society for Optical Engineering* **10752** 107522S

[13] Makovetskii A, Voronin S and Kober V 2017 *Analysis of Images, Social Networks and Texts* (Cham: Springer International Publishing) 326-337

[14] Tihonkih D, Makovetskii A and Voronin A 2017 *Proc. SPIE* 10396 10396-7

[15] Ruchay A, Dorofeev K, Kober A, Kolpakov V and Kalschikov V 2018 *Proc. SPIE* **10752** 1075221-10

[16] Ruchay A, Dorofeev K and Kober A 2018 *Proc. SPIE* **10752** 1075223-8

[17] Ruchay A and Kober V 2016 *Proc. SPIE* **9971** 99712Y-10

[18] Ruchay A and Kober V 2017 *Proc. SPIE* **10396** 1039626-10

[19] Ruchay A and Kober V 2017 **Proc. SPIE** *10396* 1039627-9

[20] Ruchay A and Kober V 2018 *Analysis of Images, Social Networks and Texts* (Cham: Springer International Publishing) 280-291

[21] Ruchay A, Kober A, Kolpakov V and Makovetskaya T 2018 *Proc. SPIE* **10752** 1075224-12

[22] Han X F, Jin J S, Wang M J, Jiang W, Gao L and Xiao L 2017 *Signal Processing: Image Communication* **57** 103-112

[23] Han X F, Jin J S, Wang M J and Jiang W 2018 *Multimedia Tools and Applications* **77** 16887-16902

[24] Paris S, Kornprobst P and Tumblin J 2009 *Bilateral Filtering* (Hanover, MA, USA: Now Publishers Inc.)

[25] Zaman F,Wong Y P and Ng B Y 2017 *9th International Conference on Robotic, Vision, Signal Processing and Power Applications* (Singapore: Springer Singapore) 287-295

[26] Zheng Y, Li G, Xu X, Wu S and Nie Y 2018 *Computer Aided Geometric Design* **62** 16-28

[27] Rusu R B and Cousins S 2011 *IEEE International Conference on Robotics and Automation* 1-4

[28] Digne J and de Franchis C 2017 *Image Processing On Line* **7** 278-287

[29] Boubou S, Narikiyo T and Kawanishi M 2017 3*DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)* 1-4

[30] Chen R, Liu X, Zhai D and Zhao D 2018 *Digital TV and Wireless Multimedia Communication* (Springer Singapore) 128-137

[31] Alexiou E and Ebrahimi T 2017 *Ninth International Conference on Quality of Multimedia Experience (QoMEX)* 1-3

[32] Lee K and Nguyen T Q 2016 *Mach. Vis. Appl.* **27** 377-385

[33] Choi S, Zhou Q, Miller S and Koltun V 2016 *CoRR* ArXiv: abs/1602.02481

Acknowledgments

This work was supported by the Russian Science Foundation, grant no. 17-76-20045.