Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-22 (2018) 348–353

Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments

Haris Balta * **Jasmin Velagic** ** **Walter Bosschaerts** *
Geert De Cubber * **Bruno Siciliano** ***

* *Royal Military Academy, Department of Mechanics, Brussels, Belgium, (e-mail: haris.balta@rma.ac.be).*
** *Department of Automatic Control and Electronics, Faculty of Electrical Engineering, Sarajevo, Bosnia and Herzegovina, (e-mail: jasmin.velagic@etf.unsa.ba)*
*** *Department of Electrical Engineering and Information Technology, University of Naples Federico II, Napoli, Italy, (e-mail: siciliano@unina.it)*

Abstract: This paper proposes a very effective method for data handling and preparation of the input 3D scans acquired from laser scanner mounted on the Unmanned Ground Vehicle (UGV). The main objectives are to improve and speed up the process of outliers removal for large-scale outdoor environments. This process is necessary in order to filter out the noise and to downsample the input data which will spare computational and memory resources for further processing steps, such as 3D mapping of rough terrain and unstructured environments. It includes the Voxel-subsampling and Fast Cluster Statistical Outlier Removal (FCSOR) subprocesses. The introduced FCSOR represents an extension on the Statistical Outliers Removal (SOR) method which is effective for both homogeneous and heterogeneous point clouds. This method is evaluated on real data obtained in outdoor environment.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Unmanned ground vehicle, outlier removal, 3D point cloud, large-scale environment, FCSOR.

1. INTRODUCTION

Recently, there has been the growing interest in employing 3D laser range scanners in mobile robotics. The laser scanners provide a simple and direct capturing of a large amount of 3D information and can produce the precise range images (Zermas et al. (2017), Sock et al. (2016)). However, the appearance of a noise and outliers in point clouds causes serious problems in reconstruction of 3D environmental model, especially for the large-scale environments with rough terrain. In general, an outlier is an observation which is markedly distant from other observations in a dataset. The number of generated points in lidar point cloud is in order of tens million points. There will be several points which don't respect the homogeneity of the dense surrounding neighbors. Such points are likely to be outliers. Because of that it is important to ensure a fast and computational efficient method for outliers analysis and removal in 3D environmental model. Therefore, the raw measurement data of 3D laser scanner are too redundant to be directly employed in a modelling of 3D outdoor environments. Furthermore, in order to provide a precise and effective 3D model a preprocessing step is necessary to detect and remove outliers.

The presence of outliers represents a significant problem for automated 3D laser scanning system (Nurunnabi et al. (2015)). The 3D model obtained from the point clouds with outliers typically misses many details and/or reconstructs wrong geometry. The finding and addressing outliers are almost inevitable problems in large datasets. The main sources of outliers in our work are: a measurement noise caused by constrained laser range (80 meters), a movement of the mobile robot over rough terrains and marginal points which introduce an error in data processing. The laser sensor scans an environment in real-time during the robot motion and the rough terrain imposes vibrations to the robot platform. In general, laser scans generate 3D point clouds of varying point densities. Marginal points are mainly introduced by objects occlusion and surface reflectance. All outlier sources can be treated as a system uncertainty. It is very important to detect outliers in 3D input scans priori to the filtering of points in the point cloud due to negative effects which have outliers on the output of filtering algorithms (Baligh et al. (2011)). The pre-processing of noisy data, i.e. detection and reduction of outliers, is important step to make further processing in the 3D model reconstruction lighter computationally and more accurately. Extensive research activities have be devoted to detection and removal of outliers in the laser 3D point clouds. There are lots of methods for the treatment of outliers, which can be classified into the following groups: traditional, wavelets-based and artificial (AI)-based approaches (Cateni et al. (2008)). The traditional-based approaches can be either distribution-

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.11.566

based, depth-based, clustering, distance-based or density-based (Chandola et al. (2009), Hodge and Austin (2004), Pimentel et al. (2014)). The main shortcomings of traditional methods are coping with a high dimensionality of the data in large 3D point clouds and satisfying real-time requirements for the outliers removal in the large 3D point clouds (Ummenhofer and Brox (2015)). Wavelet transformations and AI based methods can be an alternatives to the traditional methods. In order to identify outliers, the wavelet-based methods transform the space and find them in the non-dense regions in transformed space (Kern et al. (2005)). In the field of AI the following methods are mostly used: neural networks (Zhang (2014)), support vector machines (Jordaan and Smits (2004)) and fuzzy logic (Cateni et al. (2007)). The main advantage of these methods is requiring poor or no a priori assumption on the considered data in datasets. The problem of real-time outlier detection and removal remains to be solved by AI methods. The one of the widely used clustering method for outliers treatment is Statistical Outlier Removal (SOR) (Rusu (2009)). It is very efficient method, nevertheless it can have its processing time limitations by directly applying it to large 3D datasets, typically consist of several million points. The motivation behind this paper is to exactly develop a time effective method for large sensor data handling and preparation with aim to simultaneously improve the accuracy and the computational tractability of the widely used SOR method.

The main contribution of our paper is an extension of the existing SOR method, named Fast Cluster Statistical Outlier Removal (FCSOR), which is time efficient for outlier analysis and removal in arbitrary large datasets, included millions of points. This method is based on a clasterization and dimensional reduction of the 3D space. It decreases a computational complexity, provides faster computation and saves memory resources for further steps in 3D environment modelling. The proposed method is applicable for both homogeneous and heterogeneous point clouds, acquired from sensors mounted on UGV and UAV platforms. The superiority of the proposed FCSOR method regarding to widely used SOR method is experimentally verified.

The paper is outlined as follows. The Section 2 describes the framework for data handling and preparation based on downsampling and filtering using the proposed FCSOR method. The obtained experimental results and comparative analysis of the FCSOR and SOR method are presented in Section 3. The Section 4 summarizes the results of the paper and gives possible research directions for the future work.

2. PROPOSED FRAMEWORK FOR DATA HANDLING AND PREPARATION

The data handling and preparation framework includes noise reduction through filtering and downsampling for input datasets (point clouds). The 3D point cloud is taken from laser scanner mounted on the UGV platform. This framework is necessary to filter out noisy measurements and to get more unified dense datasets, thereby saving computational and memory resources in further processing steps. The output of this framework is the filtered point cloud. It is also input to the postprocessing module or 3D modelling process. This preprocessing module is illustrated in Fig. 1.

In the rest of this section, the voxel-subsampling and filtering using the proposed FCSOR method will be described.

2.1 Voxel-subsampling

The purpose of the downsampling data is to obtain a highly uniform point density between different point clouds, which represent inputs to the proposed framework. This is required because the scanning technique used for the terrestrial datasets tends to gather clouds whose density changes with range measurement. The solution is based on the voxelized grid decomposition (Rusu and Cousins (2011)). The 3D space of the input point cloud is decomposed into a set of voxels. The decomposition can be done in two ways: having a constant number of voxels in each direction of the 3D space or using a constant size of the voxel. After the decomposition step, in each voxel all points present will be approximated with their centroid. The output of 3D voxel-subsampling algorithm is the uniformed point cloud with decreased number of points. This approach is a bit slower than approximating them with the closest point to the center of the voxel, but it represents the underlying 3D model more accurately. By changing the size of the voxels, different densities of the input data can be achieved.

Algorithm 1 represents the pseudo code of the voxel-subsampling procedure (Balta et al. (2017)).

2.2 Fast Cluster Statistical Outlier Removal

Due to the massive amount of data generated by the 3D acquisition devices, there will be several points which do not respect the homogeneity of the dense surrounding neighbors. Such points do not provide a good representation of the underlying sampled environment. Thus, points which are rather isolated are likely to be outliers. Removing these noisy measurements, e.g. outliers, from a point cloud dataset leads to an overall faster computation, due to the reduced error. Therefore, in this paper we propose an extension of the statistical outliers removal (SOR) method presented in Rusu (2009). The motivation for extending this method is naturally related with the heterogeneity

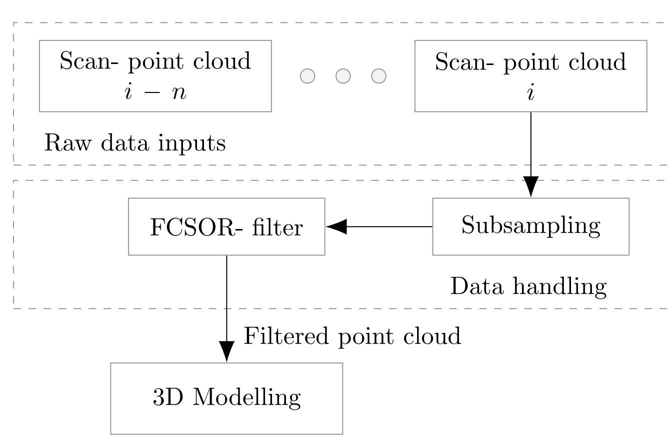


Fig. 1. Data handling and preparation framework.

Algorithm 1 3D Voxel-subsampling algorithm

INPUT: Point cloud $\boldsymbol{M} = \{\boldsymbol{m}_i\}$, $i = 1, \ldots, M_p$, $\boldsymbol{m}_i = (x_i, y_i, z_i)$

OUTPUT: Point cloud $\boldsymbol{N} = \{\boldsymbol{n}_j\}$, $j = 1, \ldots, N_p$ decreased, uniformed

```
for all point m_i in parallel do
    find bucket_m
    update table_of_found_buckets
end
in parallel sort table_of_found_buckets {radix sort}
in parallel count points in each bucket
in parallel compute centroids for all buckets
for all buckets do
    for all point in current bucket do
        find distance from centroid
    end
    mark point with minimal distance to the centroid
end
copy marked points as a result
```

of large datasets (several million points) we are dealing with. The datasets coming from airborne and terrestrial point clouds, acquired by different type of sensor systems, imposes uneven point density and measurement errors. Another motivation for the extension of the classical SOR removal method is the high computational time necessary to process such type of datasets with the traditional SOR method.

Based on the previous mentioned limitations, our proposed solution relies on a clusterization and dimensional reduction method of the 3D space that lowers the computational complexity. The pseudocode of the proposed FCSOR method is given in Algorithm 2.

The proposed FCSOR method allows to perform computations partially in parallel and also in that way we reduce the computational time. The 3D space of the input point cloud is divided into equal number of clusters $\boldsymbol{C}$. For each point $\boldsymbol{m}_i \in \boldsymbol{M}$ where $\boldsymbol{M}$ is the input dataset, the average squared Euclidean distance d_i to its k-nearest neighbors is first computed and each $\boldsymbol{m}_i$ is added to the appropriated cluster $\boldsymbol{C}_L \in \boldsymbol{C}$. The dimensional reduction of the 3D space is done by firstly computing the mean number of points μ_u from all clusters $\boldsymbol{C}_L$. If the number of points in a particular cluster is higher than the average μ_u that cluster is rejected from the further computation. By reducing the number of clusters we can greatly speed up the computation as we have less points to search. The resulting filtered point cloud $\boldsymbol{O}$ is estimated from the remaining clusters as in the SOR method.

The experimental validation of the proposed method and its comparison with SOR method will be performed the next section.

3. EXPERIMENTAL RESULTS

In order to verify the performance of the proposed framework we conducted a large scale mapping campaign in an urban search and rescue context. The experimental setup presented here was carried out at the Camp Roi Albert

Algorithm 2 3D data filtering - FCSOR method

INPUT: Point cloud $\boldsymbol{M} = \{\boldsymbol{m}_i\}$, $i = 1, \ldots, M_p$, $\boldsymbol{m}_i = (x_i, y_i, z_i)$

OUTPUT: Filtered point cloud $\boldsymbol{M} = \{\boldsymbol{o}_f\}$, $f = 1, \ldots, O_f$, $\boldsymbol{o}_i = (x_f, y_f, z_f)$

for *all points* $\boldsymbol{m}_i \in \boldsymbol{M}$ *in* ***parallel*** **do**

  find $maxX = max\{x_i\}$, $maxY = max\{y_i\}$, $maxZ = max\{z_i\}$

  and $minX = min\{x_i\}$, $minY = min\{y_i\}$, $minZ = min\{z_i\}$

end

Defining cluster size

 $ClusterLength = maxX - minX/NumberClusters(user\_defined)$

 $ClusterWidth = maxY - minY/NumberClusters(user\_defined)$

 $ClusterHeight = maxZ - minZ/NumberClusters(user\_defined)$ Subdividing point cloud space into clusters $\boldsymbol{C}$ **for** *all points* $\boldsymbol{m}_i \in \boldsymbol{M}$ *in* ***parallel*** **do**

  add $\boldsymbol{m}_i$ appropriate Cluster $\boldsymbol{C}_L \in \boldsymbol{C}$ $\overline{d_i} = \dfrac{\sum_i^k k\_nearestNeighbourDistance(\boldsymbol{m}_i)}{k}$

end

for *all points* $\boldsymbol{m}_i \in \boldsymbol{M}$ **do**

  $\mu = \sum_i^{M_p} \dfrac{d_i}{M_p}$

  $\xi = \sqrt{\dfrac{1}{M_p} \sum_i^{M_p} (d_i - \mu)^2}$

end

define $\boldsymbol{C}_u \in \boldsymbol{C}$ where $\boldsymbol{C}_u$, are used clusters, μ_u is the number of points in all used clusters, and U the number of used clusters $\boldsymbol{C}_u$

$\mu_u = \sum_0^{M_p} \dfrac{NumberPoints\ \boldsymbol{C}_u}{U}$ **for** *all* $\boldsymbol{C}_h \in \boldsymbol{C}_u$ **do**

  if $NumberPoints\ \boldsymbol{C}_h < \mu_u$ **then**

   $\boldsymbol{O} = \{\boldsymbol{m}_i \in \boldsymbol{M} \mid (\mu - \alpha\xi) \le \overline{d_i} \le (\mu + \alpha\xi)\}$

  end

end

(Fig. 2), one of the largest military bases of the Belgian Defence (located near the city of MarcheenFamenne, Belgium). In order to acquired 3D raw input data, the 3D mobile mapping robotic system RMA tEODor UGV with lidar scanner is employed(Fig. 3).

During the mapping process, the RMA tEODor UGV usually traverses 4-5 m between each scan. The distance between scans is one of the key parameters in the mapping framework. Because this allows a good overlap between the scans. For each scan of the environment, the laser scanner needs to perform a full revolution which usually takes around 10 sec. It means that the traverse speed of the tEODor UGV is about 0.5 m/s in our experiment. During the experiments, all the 3D environmental data was gathered online on an embedded PC, directly integrated on the RMA tEODor UGV system (Intel i7-2650 4 Core @2.4 GHz CPU with 16 GB of RAM). The software framework was developed using the C++ programming language and it was based on the Robot Operating System (ROS) Quigley et al. (2009) and Point Cloud Library (PCL) Rusu and Cousins (2011). In additional all the data are backed up via the ROS-bag mechanism. The size of the mapped area we worked on during our experiments is approximately 600 m $\times$ 200 m.

The raw data, or initial 3D point cloud, acquired by 3D robot mapping system is given in Fig. 4a. The resulting point cloud obtained by subsampling input data set is shown in Fig. 4b. The number of points in the resulting point cloud is decreased approximately five times with respect to the initial 3D point cloud. The subsampling resolution (voxel size) for this dataset was set on 0.5 m for each direction of the 3D space (x, y, z).

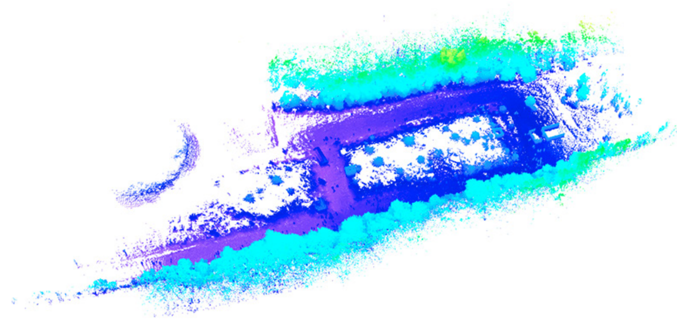
The results of employing the proposed FCSOR method are presented in Fig. 5. The blue clusters in Fig. 5a are excluded from the further computation because the number of their points are smaller than average number μ_u. Then, the resulting filtered point cloud is estimated from the remaining clusters shown in Fig. 5b. By reducing the number of clusters we can greatly speed up the computation as we have less points to search.

In order to better illustrate the proposed FCSOR method, the whole procedure of the outliers (noises) removal is presented in Fig. 6. A raw point cloud dataset with noisy measurements, e.g. outliers, is shown in Fig. 6a, where identified outliers are marked with red circles. The resulting 3D point cloud obtained by using the proposed FCSOR filter is presented in Fig. 6b, indicating the effect of removing the outliers (with number of points left: 8 891 294 out of 9 109 169, i.e. 97.60 %). For the purpose of these experiments the value of α (threshold factor) was set to 2 and the number of k - neighbors to 25. The graph presented in Fig. 6c shows the relation between the filtered and non-filtered point cloud datasets. It can be noted immediately from the figure that the mean distance to the k - nearest neighbors is drastically reduced in the filtered version.

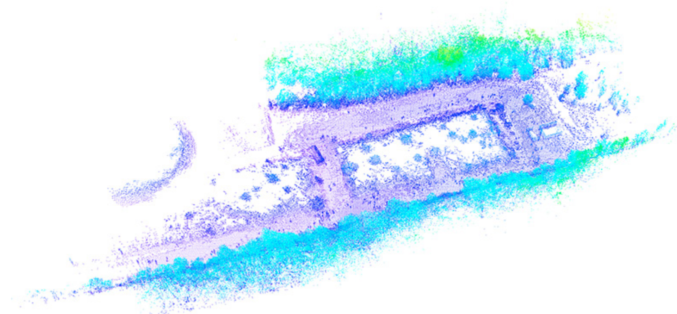
The execution time comparison between the classical SOR and our improved FCSOR method is shown in Fig. 7. The figure clearly shows that the proposed FCSOR method is about twice faster than the traditional SOR method and thereby proposes a viable alternative for the traditional method.

Fig. 2. Overview of operational environment.

Fig. 3. RMA tEODor UGV with the 3D mapping hardware.



(a) Raw data: Initial point cloud

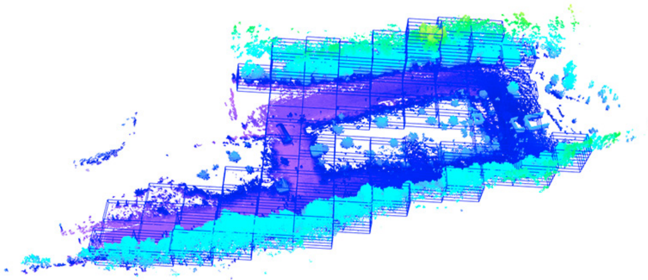


(b) Resulting down sampled point cloud

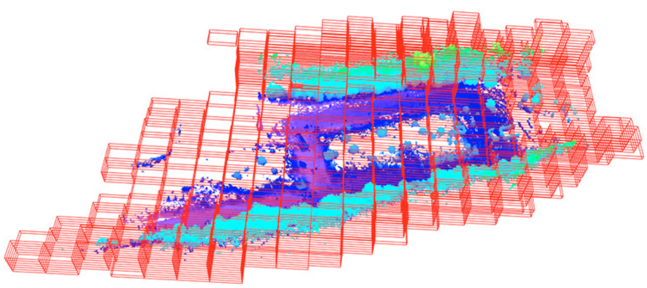
Fig. 4. Subsampling example using 9.10×10^6 points; the resulting point cloud 0.22×10^6 points (Dataset: Military-base Marche-en-Famenne, Belgium).

4. CONCLUSIONS

In this paper, a framework for the data handling and preparation of datasets acquired by 3D laser scanner (lidar) is designed. This framework is composed of the two subsystems, the voxel-subsampling and data filtering. The novelty of our paper lies in a development of the fast, simple and computational very efficient method for outliers

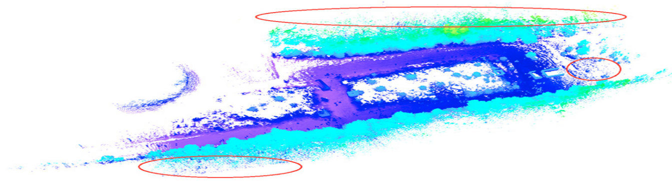


(a)

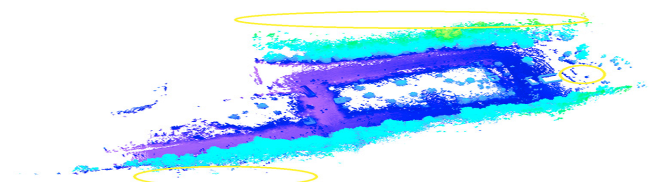


(b)

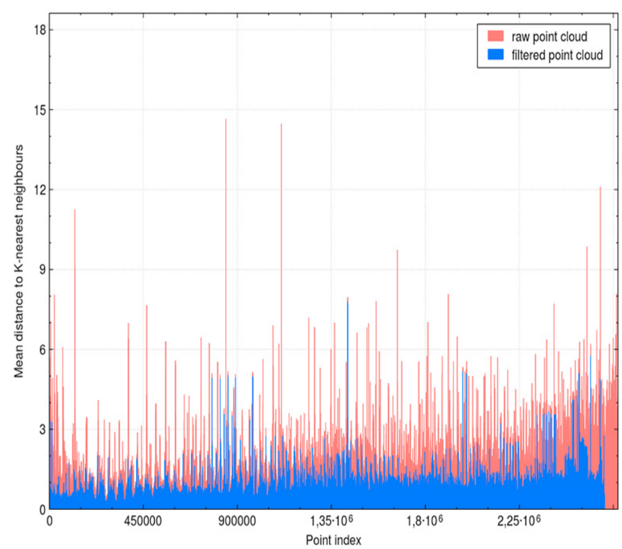
Fig. 5. a The blue clusters represent the search space which will not be taken into consideration (number of clusters: 216; number of points: 8 283 013) b Red clusters contain points of the search space (number of clusters: 1 189; number of points: 826 156). (Dataset: Military base Marche-en-Famenne, Belgium).



(a)



(b)



(c)

Fig. 6. a) Raw input point cloud P; b) FCSOR filtered point cloud P^* (around 2.40 % points rejected); c) relationship between the raw input and filtered output point cloud (Dataset: Military-base Marche-en-Famenne, Belgium).

analysis and removal in 3D point clouds. The proposed method overcomes the shortcomings of the existing statistical outlier removal (SOR) method, by improving the time complexity for processing large-scale outdoor environment with rough terrains. The effectiveness, accuracy and time complexity of the proposed method are verified through experiments and comparative analysis with SOR method.

REFERENCES

Zermas, D., Izzat, I., and Papanikolopoulos, N. (2017). Fast segmentation of 3D point clouds: a paradigm on LiDAR data for autonomous vehicle applications. IEEE International Conference on Robotics and Automation, May 29-June 3, Singapore, pages 5067–5073.

Sock, J., Kim, J., and Min, J. (2016). Probabilistic traversability map generation using 3D-LIDAR and camera. IEEE International Conference on Robotics and Automation, May 16-21, Stockholm, Sweden, pages 5631–5637.

Nurunnabi, A., West, G. and Belton, D. (2015). Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognition*, volume 48, pages 1404–1419.

Baligh, J., Valadan, M., Mohammadzadeh, M. and Sadeghian, S. (2011). A novel filtering algorithm for bare-earth extraction from airborne laser scanning data using an artificial neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, volume 4, pages 836–843.

Cateni, S., Colla, V. and Vannucci, M. (2008). Outlier detection methods for industrial applications. In book *Advances in Robotics, Automation and Control*, pages 265–282. Springer Verlag, Heidelberg.

Chandola, V, Banerjee, A. and Kumar, V. (2009). Anomaly detection: a survey. *ACM Computing*, volume 43, pages 1–72.

Hodge, V., and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, volume 22, pages 85–126.

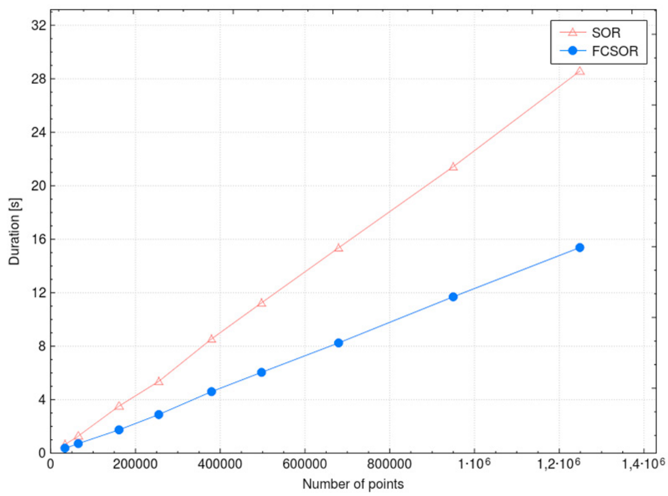


Fig. 7. Execution time comparison between FCSOR and SOR methods for different density 3D input datasets. (Dataset: Military-base Marche-en-Famenne, Belgium - Computing hardware: i7 processor with 16 GB of RAM).

Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, volume 99, pages 215–249.

Ummenhofer, B., and Brox, T. (2015). Global,dense multiscale reconstruction for a billion points. International Conference on Computer Vision, September 11-18, Santiago, Chile, pages 1341–1349.

Kern, M., Preimesberger, T., Allesch, M., Pail, R., Bouman, J., and Koop, R. (2005). Outlier detection algorithms and their performance in GOCE gravity field processing. *Journal of Geodesy*, volume 78, pages 509–519.

Zhang, X., and Zhang, Y. (2014). Outlier detection based on the neural network for tensor estimation. *Biomedical Signal Processing and Control*, volume 13, pages 148–156.

Jordaan, E., and Smits, G. (2004). Robust outlier detection using SVM regression. IEEE International Joint Conference on Neural Networks, pages 2017–2022.

Cateni, S., Colla, V., and Vannucci, M. (2007). A fuzzy logic-based method for outlier detection. IASTED International Multi-Conference, Innsbruck, Austria, pages 561–566.

Rusu, R.B. (2009). *Semantic 3D object maps for everyday manipulation in human living environments.* PhD thesis, University of Munich.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., and Wheeler, R. (2009). ROS: An open-source robot operating system. IEEE International Conference on Robotics and Automation, Kobe, Japan, pages 1–6 .

Rusu, R., and Cousins, S. (2011). 3D is here: point cloud library (PCL). IEEE International Conference on Robotics and Automation, Shanghai, China, pages 1–4.

Balta, H., Bedkowski, J., Govindaraj, S., Majek, K., Musialik, P., Serrano, D., Alexis, K., Siegwart, R. and De Cubber, G. (2017). Integrated data management for a fleet of search-and-rescue robots. *Journal of Field Robotics*, volume 34, pages 539–582.