

Real-Time Removing of Outliers and Noise in 3D Point Clouds Applied in Robotic Applications

Gerasimos Arvanitis^(✉), Aris S. Lalos, Konstantinos Moustakas,
and Nikos Fakotakis

Electrical and Computer Engineering Department, University of Patras,
Rio, Patras, Greece
{arvanitis, aris.lalos, moustakas, fakotaki}@upatras.gr

Abstract. Nowadays, robots are able to carry out a complex series of actions, to take decisions, to interact with their environment and generally to perform plausible reactions. Robots' visual ability plays an important role to their behavior, helping them to efficiently manage the received information. In this paper, we present a real time method for removing outliers and noise of 3D point clouds which are captured by the optical system of robots having depth camera at their disposal. Using our method, the final result of the created 3D object is smoothed providing an ideal form for using it in further processing techniques; namely navigation, object recognition and segmentation. In our experiments, we investigate real scenarios where the robot moves while it acquires the point cloud in natural light environment, so that unpleasant noise and outliers become apparent.

Keywords: Robotic vision · 3D point clouds denoising · Outliers removing

1 Introduction

The sense of sight plays an important role to human's life. It allows us to connect with our surroundings, keep us safe and help us to learn more about the world. In the same way, vision is also important for robots, affecting their behavior especially when they try to interact with their environment. Robots have become smarter and their behavior is more realistic approaching humans, in many ways. One basic reason leading to this success is the learning process which is an off-line task, though. In real time applications, robots must discover their world using only their own sensors and tools without an a priori knowledge. Undoubtedly, the type and the precision of sensors affect robot's perception and behavior. In recent years, a lot of research has been carried out into the field of robots vision, having presented excellent results applied to applications like segmentation, recognitions and others. Although there has been significant work in this way, some limitations still remain. Image analysis of the conventional RGB cameras, used mostly at the previous years, does not provide all the necessary information existing in

the real world, like the 3D perception. On the other hand, depth cameras have become popular as an accessible tool for capturing information. Nevertheless, depth cameras are very sensitive to motion and light conditions and this may cause abnormalities to the final results, especially when robot moves while it captures the point cloud.

2 Related Works

Depth cameras have started to be used in a lot of robotic applications providing extra useful information (geometric and topographic) which is not able to be provided by the conventional cameras. Robots need the 3D visual perception in order to understand better their environment and demonstrate a more satisfied behavior. A detailed vision capability is necessary in many challenging applications especially when further processing tasks are required; like segmentation, object recognition and navigation. For example, robots need to intelligibly specify and identify their environment (borders, existing objects, obstacles) before they start to navigate inside it.

In [1] a method for registration of 3D point clouds is presented. In order to identify a complete point cloud presentation of the robot gripper it is rotated in front of a stereovision camera and its geometry is captured from different angles. However this approach does not use filters to remove noise from the data. The paper in [2] also presents point cloud registration algorithms which are used in mobile robotics. In this work [3] authors use a hand-held RGB-D camera to create a 3D model. The research in [4] shows the capability of using a mobile robot for performing real-time vision tasks in a cloud computing environment, however without any pre-processing step taking place. In [5], an object recognition engine needs to extract discriminative features from data representing an object and accurately classify the object to be of practical use in robotics. Furthermore, the classification of the object must be rapidly performed in the presence of a voluminous stream of data. The article in [6] investigates the problem of acquiring 3D object maps of indoor household environments.

The analysis of the above state-of-the-art works makes it clear that a different approach should be used for real time applications. If a robot moves, while it receives points, then its line-of-sight focus might become unstable creating outliers to the acquired 3D point cloud. Additionally, different light conditions can affect the final results. The proposed approach manages to handle these challenging situations creating smoothed 3D objects by removing outliers and noise.

The rest of this paper is organized as follows: Sect. 2 presents an overview of our method. Section 3 presents our experimental results. Section 4 draws the conclusions.

3 Overview of Our Method

In this section we describe the main steps followed by the proposed method. We also introduce the used mathematical background and all the necessary assumptions. Firstly, we assume that the robot captures visual information by collecting

points using its depth camera. However, in the general case the precision of cameras is limited or a relative motion between robot and target exists, and as a result the acquired 3D point cloud suffers from noise, outliers and incomplete surfaces. These problems must be solved before robot uses this inappropriate information. In the next paragraphs we describe how our method manages to overcome these limitations.

3.1 Framework Overview

In Fig. 1 the basic steps of our approach are briefly presented. Firstly, the robotic optical sensor starts capturing points of objects that exist in robot's line of vision. The acquired 3D point clouds are usually noisy and they also have misaligned outliers. We use Robust PCA for removing the outliers and consequently we create a triangulated model based on k nearest neighbors (k -NN) algorithm. The triangulation process helps us to specify the neighbors of each point so that the bilateral filtering method can efficiently be used as denoising technique. At the end, a smoothed 3D mesh is created which has a suitable form for being used by other applications or processing tasks.

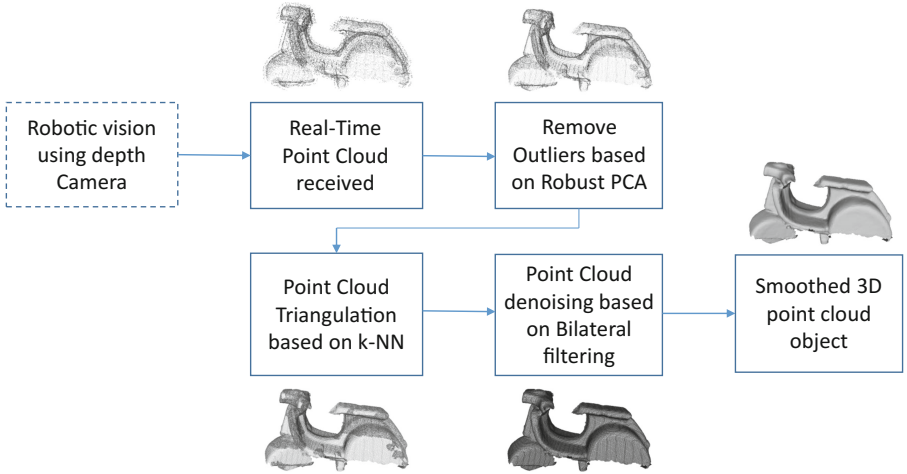


Fig. 1. Framework of our proposed method

3.2 Removing 3D Point Cloud Outliers

Let us assume that the captured 3D point cloud \mathbf{M} consists of m points represented as a vector $\mathbf{v} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]$ in a 3D coordinate space $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Re^{m \times 1}$ and $\mathbf{v} \in \Re^{m \times 3}$. Some of these points are considered as outliers and they must be removed. The method that we use for the removing of outliers is the robust

PCA [7]. We start by assuming that the point cloud, represented as a matrix of data \mathbf{M} , may be decomposed as:

$$\mathbf{M} = \mathbf{L} + \mathbf{S} \quad (1)$$

where \mathbf{L} is a low-rank matrix representing the space of real data while \mathbf{S} is a sparse matrix representing the space where outliers lie. The dimensions of \mathbf{L} and \mathbf{S} matrices as well as the location of their non-zero entries are considered as unknown. The classical Principal Component Analysis (PCA) estimates the \mathbf{L} by solving:

$$\begin{aligned} & \text{minimize} \quad \|\mathbf{M} - \mathbf{L}\| \\ & \text{subject to} \quad \text{rank}(\mathbf{L}) \leq m \end{aligned} \quad (2)$$

Assuming that there are few outliers and also that the entries of \mathbf{S} have an independent and identically Gaussian distribution then the above equation can be efficiently solved using the singular value decomposition (SVD). Nonetheless, this approach is not appropriate for this case study. The problem of outliers in the point cloud can be considered as an idealized version of Robust PCA. Our purpose is to recover the low-rank matrix \mathbf{L} from the measurements $\mathbf{M} = \mathbf{L} + \mathbf{S}$ which are highly corrupted by the matrix \mathbf{S} with randomly large magnitude of entries. The approach that we use [8] estimates the Principal Component Pursuit (PCP) by solving:

$$\begin{aligned} & \text{minimize} \quad \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ & \text{subject to} \quad \mathbf{L} + \mathbf{S} = \mathbf{M} \end{aligned} \quad (3)$$

where $\|\mathbf{L}\|_* := \sum_i \sigma_i(\mathbf{L})$ denotes the nuclear norm of the matrix which is the sum of the singular values of \mathbf{L} . This convex PCP problem can be solved using an augmented Lagrange multiplier (ALM) algorithm, as described in [9], which works with stability across a wide range of situations without the necessity of parameters configuration. The ALM method operates on the augmented Lagrangian:

$$l(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \langle \mathbf{Y}, \mathbf{M} - \mathbf{L} - \mathbf{S} \rangle + \frac{\mu}{2} \|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F^2 \quad (4)$$

The PCP would be solved by a generic Lagrange multiplier algorithm setting $(\mathbf{L}_i, \mathbf{S}_i) = \arg \min_{\mathbf{L}, \mathbf{S}} l(\mathbf{L}, \mathbf{S}, \mathbf{Y}_i)$ repeatedly, and then updating the Lagrange multiplier matrix via $\mathbf{Y}_{i+1} = \mathbf{Y}_i + \mu(\mathbf{M} - \mathbf{L}_i - \mathbf{S}_i)$. This process is applied in m overlapped areas, where the i -th area a_i consists of vertex \mathbf{v}_i and its 30 nearest neighbors vertices. In our case study, where a low-rank and sparse decomposition problem exists, the $\min_{\mathbf{L}} l(\mathbf{L}, \mathbf{S}, \mathbf{Y})$ and $\min_{\mathbf{S}} l(\mathbf{L}, \mathbf{S}, \mathbf{Y})$ both have very simple and efficient solutions. We introduce two ancillary operators \mathcal{Q}_τ and \mathcal{D}_τ . $\mathcal{Q}_\tau : \mathbb{R} \rightarrow \mathbb{R}$ denotes the shrinkage operator $\mathcal{Q}_\tau[\cdot] = \text{sgn}(\cdot) \max(|\cdot| - \tau, 0)$ and extends it to matrices by applying it to each element while the $\mathcal{D}_\tau(\cdot)$ denotes the singular value thresholding operator given by $\mathcal{D}_\tau(\cdot) = U \mathcal{Q}_\tau(\Sigma) V^T$. The estimation occurs according to:

$$\arg \min_{\mathbf{S}} l(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \mathcal{Q}_{\lambda\mu^{-1}}(\mathbf{M} - \mathbf{L} + \mu^{-1}\mathbf{Y}) \quad (5)$$

$$\arg \min_{\mathbf{L}} l(\mathbf{L}, \mathbf{S}, \mathbf{Y}) = \mathcal{D}_{\mu^{-1}}(\mathbf{M} - \mathbf{S} + \mu^{-1}\mathbf{Y}) \quad (6)$$

Firstly, the l is minimized with respect to \mathbf{L} (fixing \mathbf{S}) and then the l is minimized with respect to \mathbf{S} (fixing \mathbf{L}). Finally, the Lagrange multiplier matrix \mathbf{Y} is updated based on the residual $\mathbf{M} - \mathbf{L} - \mathbf{S}$. The process is repeated for every area $a_i \forall i = 1, m$.

$$\begin{aligned} \mathbf{S} &= \mathbf{Y} = 0; \mu > 0 \\ \mathbf{L}_{i+1} &= \mathcal{D}_{\mu^{-1}}(\mathbf{M} - \mathbf{S}_i + \mu^{-1}\mathbf{Y}_i); \\ \mathbf{S}_{i+1} &= \mathcal{Q}_{\lambda\mu^{-1}}(\mathbf{M} - \mathbf{L}_{i+1} + \mu^{-1}\mathbf{Y}_i); \\ \mathbf{Y}_{i+1} &= \mathbf{Y}_i + \mu(\mathbf{M} - \mathbf{L}_{i+1} + \mathbf{S}_{i+1}); \end{aligned} \quad (7)$$

The cost of each iteration is estimating \mathbf{L}_{i+1} using singular value thresholding. This makes necessary the computation of those singular vectors of $\mathbf{M} - \mathbf{S}_i + \mu^{-1}\mathbf{Y}_i$ whose corresponding singular values exceed the threshold μ^{-1} . There are two important implementation that need to be chosen; namely, the choice of μ and the stopping criterion. The value of μ is chosen as $\mu = m^2/4\|\mathbf{M}\|_1$ [10], while the iterative process terminates when $\|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F \leq \delta\|\mathbf{M}\|_F$, with $\delta = 10^{-7}$. After this step the vertices decrease due to the removing of the outliers, so the number of the remaining vertices is m_r where $m_r < m$.

3.3 Triangulation Based on k-NN

The acquired 3D point cloud is unorganized, meaning that the connectivity of its points is unknown thus we are unable to take advantage of the geometric relation between neighboring vertices. To overcome this limitation we estimate the connectivity between points based on k-NN algorithm using $k = 7$ as the length of each cell, as a matter of fact each point is connected with the 7 nearest neighbor points. The defined connectivity helps us to create the binary adjacency matrix $\mathbf{C} \in \mathbb{R}^{m_r \times m_r}$ consisting of the following elements:

$$c_{ij} = \begin{cases} 1 & \text{if } i, j \in \mathbf{r}_i \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where \mathbf{r}_i represents the first ring area of vertex i consisting of its k nearest neighboring vertices, so that $\mathbf{r}_i = [r_{i1}, r_{i2}, \dots, r_{ik}]$. The 1s of this sparse matrix represents edges; by connecting 3 corresponding vertices a face is created such that $f_i = [\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}] \forall i = 1, m_f$ where $m_f > m_r$. We also estimate the normals of each face based on the below formula:

$$\mathbf{n}_{f_i} = \frac{(\mathbf{v}_{i2} - \mathbf{v}_{i1}) \times (\mathbf{v}_{i3} - \mathbf{v}_{i1})}{\|(\mathbf{v}_{i2} - \mathbf{v}_{i1}) \times (\mathbf{v}_{i3} - \mathbf{v}_{i1})\|} \quad \forall i = 1, m_f \quad (9)$$

where $\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}$ represents the vertices that are related with the face f_i .

3.4 3D Mesh Denoising Using Bilateral Filtering

The final step of our approach is the mesh denoising using the bilateral filtering algorithm directly applied to the triangulated 3D point cloud. The used approach

of the bilateral algorithm, as described in [11, 12], is a fast method and it can be easily implemented in real time applications. The denoising process occurs by firstly estimating the d factor according to Eq. 10 and then updating the noisy vertices according to Eq. 11.

$$d_i = \frac{\sum_{j \in k_i} W_p W_n h}{\sum_{j \in k_i} |W_p W_n|} \quad (10)$$

where $W_p = e^{\frac{-t^2}{2\sigma_p^2}}$, $W_n = e^{\frac{-h^2}{2\sigma_n^2}}$, $t = \|\mathbf{v}_i - \mathbf{v}_j\|$, $h = \langle \mathbf{n}, \mathbf{v}_i - \mathbf{v}_j \rangle$ and $\mathbf{n} = \frac{\sum_{j \in k_i} \mathbf{n}_j}{k} \forall i = 1, m_r, \forall j \in k_i$. Once the $\mathbf{d} = [d_1 \ d_2 \ \dots \ d_{m_r}]$ is estimated then each vertex is updated according to:

$$\hat{\mathbf{v}} = \mathbf{v} + \mathbf{n} \cdot \mathbf{d}^T \quad (11)$$

where $\hat{\mathbf{v}} \in \mathbb{R}^{m_r \times 3}$ represents the vector of smoothed vertices. This method manages to preserve the features of the object without further shrinking of the already smoothed areas.

4 Experimental Analysis and Results

In this section we present the results of our approach following the steps of the proposed method as described above. For the evaluation we used a variety of data in different light and motion conditions. Here we present two corresponding examples using models of two different geometrical categories. The first model consists of flat surfaces (Moto) while the second has a lot of features and details (Cactus), as presented in Fig. 2.

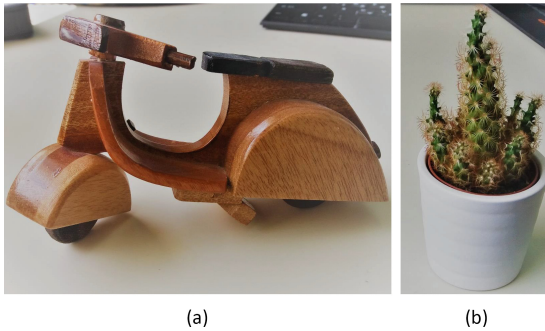


Fig. 2. Images of the original objects used as models for the evaluation process, (a) Moto (b) Cactus

In order to establish experimental conditions as close as possible to the real cases, the experiments have taken place using only natural lights, no artificial

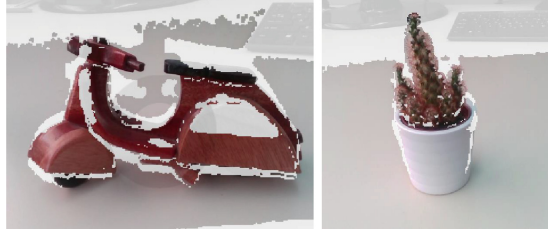


Fig. 3. The moment when the capturing point process starts

lights or other additional light sources are used. Figure 3 depicts the moment when robot's vision starts, while in Fig. 4, snapshots of the capturing process are illustrated for the two models.

Figure 4 depicts two different case studies of acquiring points. In the first case Fig. 4(a), a carefully gathering takes place while in the second case Fig. 4(b), a sudden motion of the robot causes noise and outliers. Nevertheless, observing carefully the figures, we can see that even in the first case there are imperfections that need to be covered.

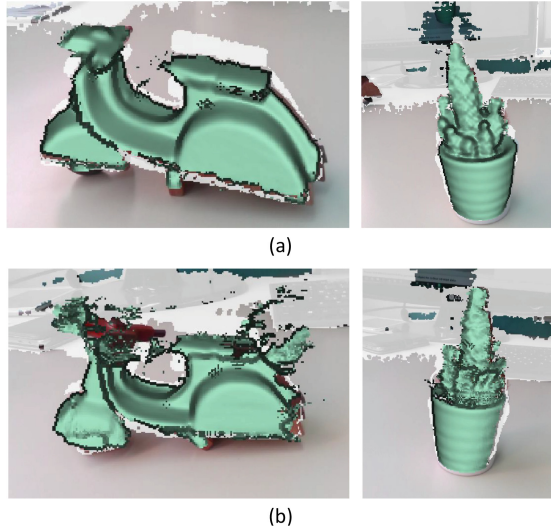


Fig. 4. (a) Stable collection of points, (b) Collection of points with a sudden robot's movement

The step-by-step extracted results of the proposed method are presented in Fig. 5. More specifically, Fig. 5(a) presents the point clouds having noise and outliers, as received by the depth camera sensor. In Fig. 5(b) the outliers of the

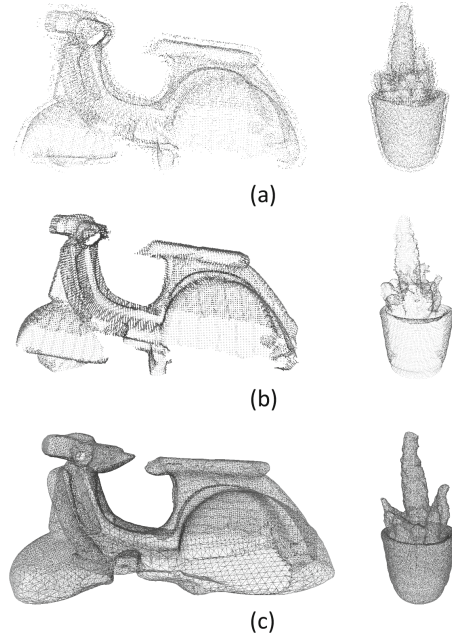


Fig. 5. (a) 3D point clouds with noise and outliers, (b) outliers have been removed but noisy parts still remain, (c) triangulated and smoothed results

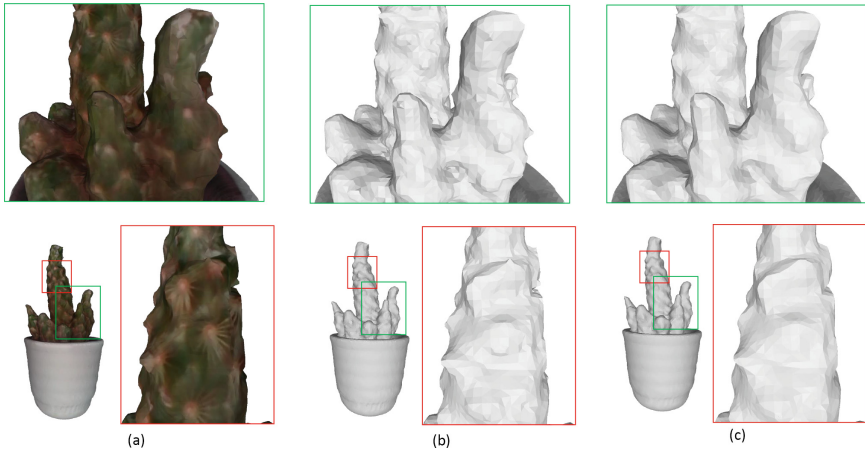


Fig. 6. (a) Cactus model is presented having its original texture, (b) noisy surfaces after outliers' removing and triangulation process, (c) the 3D denoised mesh based on bilateral filtering

point clouds have been removed, however abnormalities of noise still remain. In Fig. 5(c) the final triangulated and smoothed objects are presented.

In Fig. 6, the smoothed results using bilateral filtering for the Cactus model are presented. In addition, enlarged presentation of details are also shown for an easier comparison between the 3D objects.

5 Conclusions

In this paper we presented a method, ideally suited for real-time applications. The proposed approach can be used for removing outliers and noise that lie in 3D point clouds, captured by robot's depth cameras. Additionally, the method handles problems like the relative motion between robot and its capturing target and also the natural light conditions which can cause abnormalities in the acquired point cloud. Using the proposed approach, we manage to overcome these limitations and as a result a smoothed 3D mesh is created which has an ideal form for further processing usage.

References

1. Jerbić, B., Šuligoj, F., Švaco, M., Šekoranja, B.: Robot assisted 3D point cloud object registration. *Procedia Eng.* **100**, 847–852 (2015). ISSN 1877-7058
2. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. *Found. Trends® Robot.* **4**(1), 1–104 (2015)
3. Kidson, R., Stanimirovic, D., Pangercic, D., Beetz, M.: Elaborative evaluation of RGB-D based point cloud registration for personal robots. In: *ICRA 2012 Workshop on Semantic Perception and Mapping for Knowledge-Enabled Service Robotics*, 14–18 May 2012
4. Beksi, W.J., Papanikolopoulos, N.: Point cloud culling for robot vision tasks under communication constraint. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, pp. 3747–3752 (2014)
5. Beksi, W.J., Spruth, J., Papanikolopoulos, N.: CORE: a cloud-based object recognition engine for robotics. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4512–4517 (2015)
6. Rusu, R.R., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.* **56**(11), 927–941 (2008)
7. Aravkin, A., Becker, S., Cevher, V., Olsen, P.: A variational approach to stable principal component pursuit. In: Zhang, N., Tian, J. (eds.) *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pp. 32–41. AUAI Press, Arlington (2014)
8. Candès, E.J., Li, X., Ma Y., Wright, J.: Robust principal component analysis? *J. ACM* **58**(3), Article 11 (2011)
9. Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented Lagrange multiplier method for exact recovery of a corrupted low-rank matrices. In: *Mathematical Programming* (2009, submitted)
10. Yuan, X., Yang, J.: Sparse and Low-Rank Matrix Decomposition Via Alternating Direction Methods, November 2009. optimization-online.org
11. Zheng, Y., Fu, H., Au, O.K.C., Tai, C.L.: Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph.* **17**(10), 1521–1530 (2011). doi:[10.1109/TVCG.2010.264](https://doi.org/10.1109/TVCG.2010.264)
12. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. In: *ACM SIGGRAPH 2003 Papers (SIGGRAPH 2003)*, pp. 950–953. ACM, New York (2003)

Interactive Collaborative Robotics

Second International Conference, ICR 2017, Hatfield,

UK, September 12-16, 2017, Proceedings

Ronzhin, A.; Rigoll, G.; Meshcheryakov, R. (Eds.)

2017, IX, 288 p. 137 illus., Softcover

ISBN: 978-3-319-66470-5