Robust Feature-Preserving Denoising of 3D Point Clouds

Sk. Mohammadul Haque, Venu Madhav Govindu

Department of Electrical Engineering, Indian Institute of Science, Bengaluru, INDIA

Introduction and Problem Definition

3D Noise and Outlier

- 3D point clouds obtained from real world are invariable corrupted with significant amounts of **noise**.
- Accurately estimating an underlying surface becomes difficult due to the presence of **outliers**.
- Need to **identify and remove** outliers before further processing of the point cloud data.
- Need to estimate surface **robustly** to preserve sharp and fine-scale 3D features.

Problem Definition

Given an initial noisy point cloud, $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{N}$, possibly with outliers where $\mathbf{v}_i$ is the i^{th} noisy point position and N is the total number of points, estimate the unknown true point cloud as $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_i\}_{i=1}^{N}$.

Our Contribution

An approach to robustly denoise point clouds while preserving fine-scale features that:

- at first aggregates comparisons of individual points in a neighbourhood to **identify and remove outliers**, and
- then uses a robust denoising of the 3D points on the surface encouraging the careful **delineation and preservation** of sharp and fine-scale 3D features surface.

Proposed Method

Three steps:

1. **Robust outlier detection and removal** - Outliers are **detected** and **removed** based on an initial estimate of the **point normals** and the ℓ_2 **distances** between 3D points.
2. **Bilateral normal mollification** - The initial estimates of the **point normals** are **mollified**.
3. **Point set reposition** - The point set is **robust repositioned** using the mollified point normals.

① **Robust outlier detection and removal:**

- ***s*-neighbourhood function**:
  $\mathcal{N}(i) = \{\mathbf{v}_j \in \mathbf{V} \mid \|\mathbf{v}_j - \mathbf{v}_i\| \leq \|\mathbf{v}_k - \mathbf{v}_i\|, \forall k \notin \mathcal{N}(i) \text{ and } |\mathcal{N}(i)| = s\}$.
- **Normal Computation**:

$$\text{Normal at vertex } \mathbf{v}_i,\ \boldsymbol{n}_i = \underset{\mathbf{n},\mathbf{n}^T\mathbf{n}=1}{\operatorname{argmin}} \sum_{j \in \mathcal{N}(i)} w_{ij}\mathbf{n}^T\left((\mathbf{v}_j - \boldsymbol{\mu}_i)(\mathbf{v}_j - \boldsymbol{\mu}_i)^T\right)\mathbf{n}$$

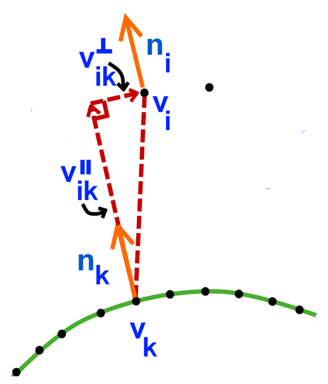
where $\boldsymbol{\mu}_i$ is the co-ordinate-wise median of $\{\mathbf{v}_j\}_{j\in\mathcal{N}(i)}$, $w_{ij} = \|\mathbf{v}_j - \mathbf{v}_i\|_2^{-1}$.

- Two **criteria** for detecting outliers:
  ✓ **Normal-based outlier detection**:

$$\text{Dissimilarity, } DS(\mathbf{v}_k, \mathbf{v}_i) = (\mathbf{n}_k^T\mathbf{n}_i)\frac{\left\|\mathbf{v}_{ik}^{\parallel}\right\|}{\left\|\mathbf{v}_{ik}^{\perp}\right\| + \epsilon}$$

$$\text{Effective Dissimilarity, } EDS(\mathbf{v}_i) = \frac{\sum_{k\in\mathcal{N}(i)} DS(\mathbf{v}_k, \mathbf{v}_i)}{|\mathcal{N}(i)|}.$$

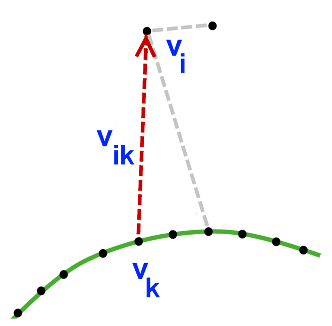
$\mathbf{v}_i$ is outlier if $EDS(\mathbf{v}_i)$ is above a threshold η_n.



✓ **Distance-based outlier detection**:

$$d_{med}(\mathbf{v}_i) = \text{MEDIAN}\left(\{\|\mathbf{v}_{ik}\|_2\}_{k\in\mathcal{N}(i)}\right).$$

$\mathbf{v}_i$ is outlier $d_{med}(\mathbf{v}_i) > \eta_d$.



② **Bilateral normal mollification:**

- Points normals are mollified in an iterative manner.
- A bilateral weight is used.

$$\hat{\mathbf{n}}_i \leftarrow \left(\sum_{j\in\mathcal{N}(i)\cup i} \phi_{ij}\hat{\mathbf{n}}_j\right) \Big/ \left\|\sum_{j\in\mathcal{N}(i)\cup i} \phi_{ij}\hat{\mathbf{n}}_j\right\|_2$$

where

$$\phi_{ij} = e^{-\left(\frac{\|\hat{\mathbf{n}}_j - \hat{\mathbf{n}}_i\|^2}{\sigma_r^2} + \frac{\|\mathbf{v}_j - \mathbf{v}_i\|^2}{\sigma_s^2}\right)},$$

and σ_r and σ_s are the normal and spatial scale parameters respectively.

Proposed Method (Contd.)

③ **Point set repositioning:**

- Robust enough to preserve fine features like edges and corners.
- Enriches the fine features.

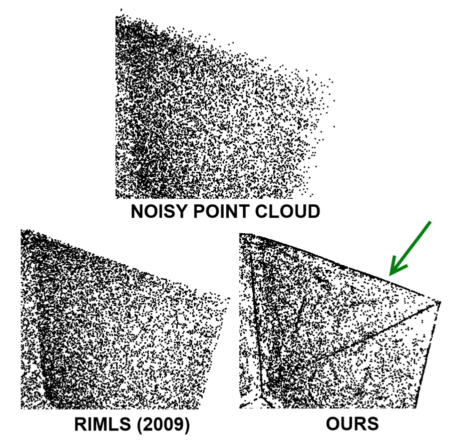
$$\min_{\{\tilde{\mathbf{v}}_i\}_{i=1}^{N}} \sum_{i=1}^{N}\sum_{j\in\mathcal{N}(i)} \gamma_{ij}\left\|\hat{\mathbf{n}}_i^T(\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j)\right\|_2^2 + \lambda\sum_{i=1}^{N}\|\tilde{\mathbf{v}}_i - \mathbf{v}_i\|_2^2$$

where

$$\gamma_{ij} = \frac{\tau_{ij}}{\sum_{j\in\mathcal{N}(i)}\tau_{ij}},\quad \tau_{ij} = \exp\left(-\frac{\|\tilde{\mathbf{v}}_j - \tilde{\mathbf{v}}_i\|^2}{\sigma_s^2}\right)$$

are the weights used to adaptively set the influence of the neighbours, $\hat{\mathbf{n}}_i$ are the mollified normals, $\mathbf{v}_i$ are the noisy point positions and λ is a small positive stabilising parameter to ensure a stable solution.

Automatic recovery of fine structures in our point set repositioning scheme as compared to the output from RIMLS (Oztireli, 2009).

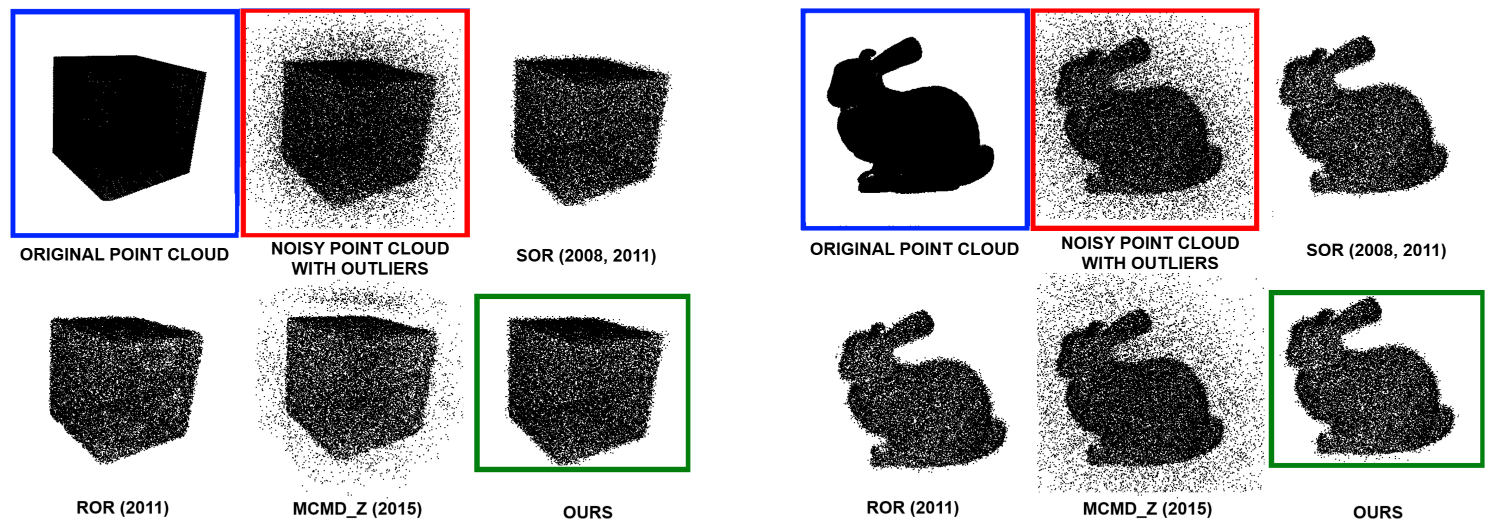


Results: Outlier Detection and Removal (Synthetic Data)

Comparative performances of SOR (Rusu et al. 2008, 2011), ROR (Rusu et al. 2011), MCMD_Z (Nurunnabi et al. 2015) and our method.

Input Model	Outliers		Accuracy			
	Density (%)	Std. deviation (%)	SOR	ROR	MCMD_Z	Ours
Cube	20	10	**0.939**	0.927	0.927	**0.939**
$N = 49154$	40	20	0.880	0.905	0.703	**0.926**
Sphere	20	10	0.949	0.921	0.937	**0.952**
$N = 40962$	40	20	0.902	0.934	0.636	**0.951**
Bunny	20	10	0.941	0.928	0.890	**0.959**
$N = 40245$	40	20	0.949	0.933	0.670	**0.969**

Visual comparison of outlier removal on a cube and the Bunny for outlier density of 40% with standard deviation of 20% of the point cloud dimensions in presence of Gaussian noise of std. dev. of avg. edge length of original meshes.

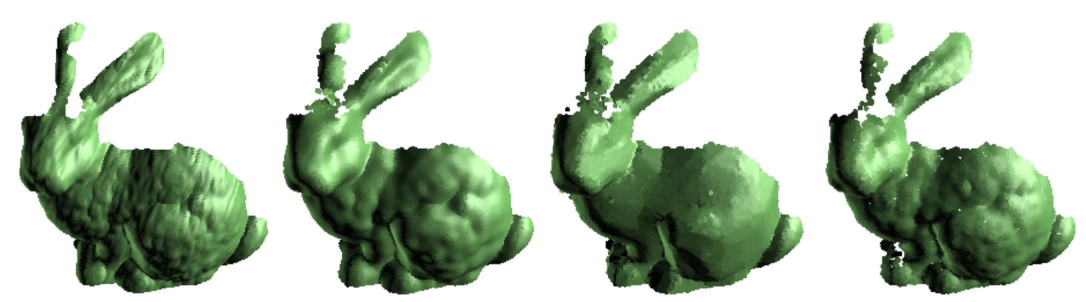


Results: Denoising (Synthetic Data)

Comparison of denoising performance of our method with RIMLS (Oztireli et al. 2009) and ℓ_0-method (Sun et al. 2015).

Input Model	Noise Std. Dev.	Mean cloud-to-mesh ℓ_2 distance		
	(Avg. Edge Length)	RIMLS	ℓ_0	Ours
Cube, $N = 49154$	100%	0.0016	0.0041	**0.0005**
Sphere, $N = 40962$	100%	**0.0046**	0.0156	0.0049
Bunny, $N = 40245$	100%	0.0023	0.0054	**0.0021**

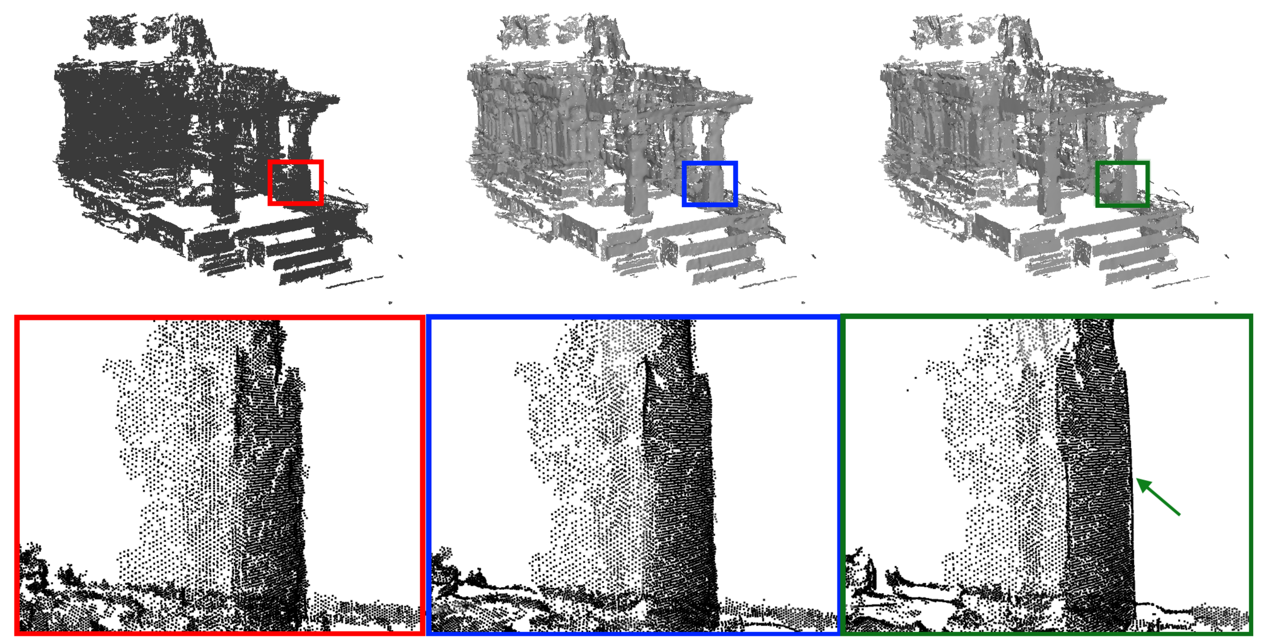
Visual comparison on a noisy Bunny ($N = 40245$).



Original Point Cloud　RIMLS (2009)　L-0 (2015)　Ours

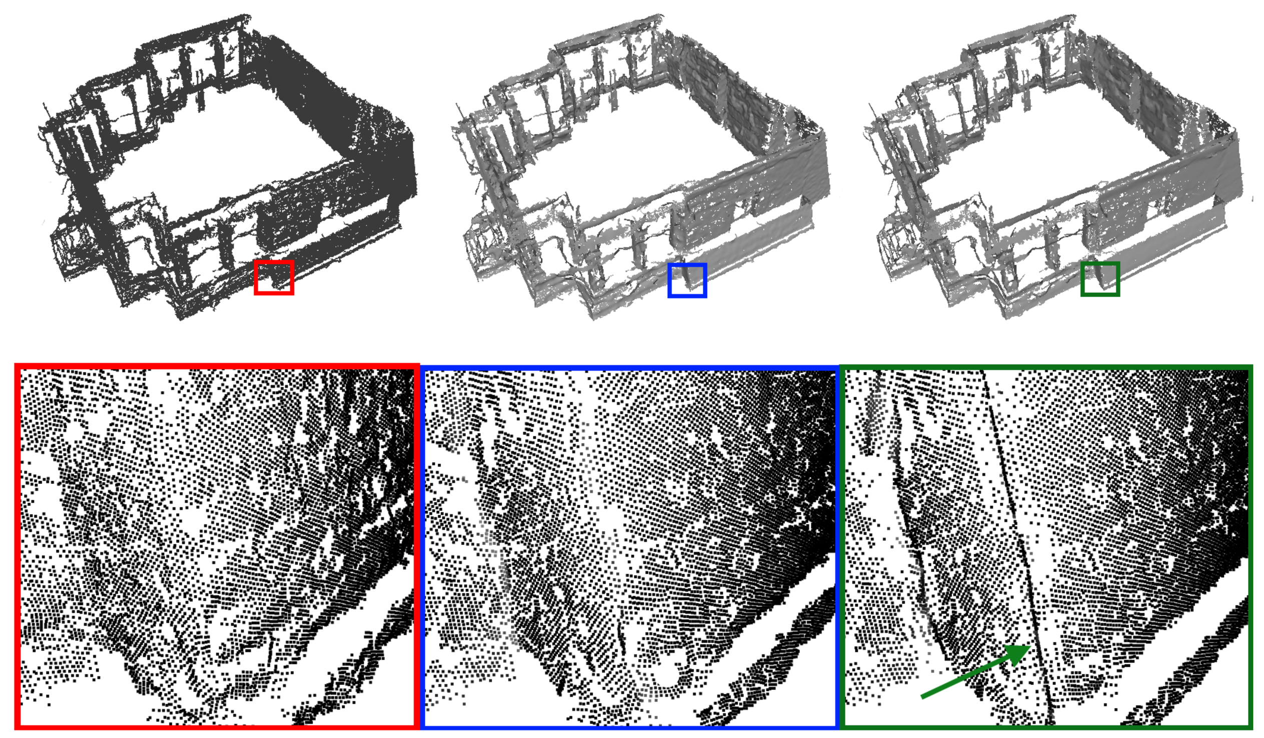
Results: Denoising (Real Data)

Comparative results on a point cloud ($N = 919851$) of a heritage monument in the Vitthala temple complex at Hampi, India, obtained from multi-view stereo.



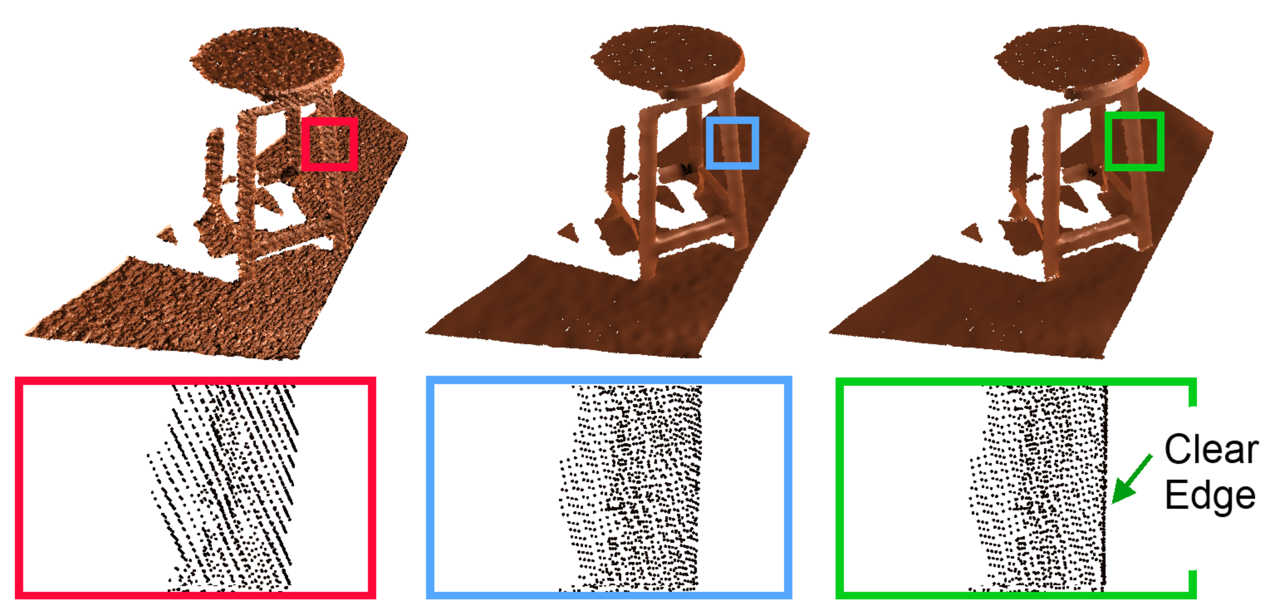
Noisy Point Cloud　RIMLS (2009)　Ours (with clear edge)

Comparative results on a point cloud ($N = 1030980$) of another heritage monument in the Vitthala temple complex at Hampi, India, obtained from multi-view stereo.



Noisy Point Cloud　RIMLS (2009)　Ours (with clear edge)

Comparative results on a point cloud ($N = 153288$) of a stool obtained using a depth scanner.



Noisy Point Cloud　RIMLS (2009)　Ours

Conclusion

- A **robust 3D point cloud denoising method** consisting of a robust outlier detection and removal, bilateral normal mollification and finally a repositioning of the 3D points that preserve the fine scale features is presented.
- Our method automatically **recovers** well-defined **edges and corners**.
- **The efficacy of our approach** over other relevant methods in the literature is established through multiple examples and experiments.

smhaque|venu@ee.iisc.ernet.in