Global Mesh Denoising with Fairness

Sk. Mohammadul Haque, Venu Madhav Govindu

Indian Institute of Science, Bengaluru, India

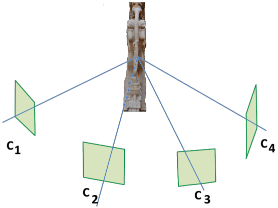
3DV 2015, Lyon, France

Presentation Outline

Motivation

3D data and noise

- 3D data is **easy to acquire** in recent years:
  - Dense multiview stereo using RGB images.
  - Depth cameras.
- **Significant amount of noise**
- **Denoising step** is necessary.



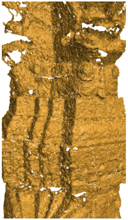
Multiview stereo

Depth cameras

Motivation

3D data and noise

- 3D data is **easy to acquire** in recent years.
  - Dense multiview stereo using RGB images.
  - Depth cameras.
- **Significant amount of noise**
- **Denoising step** is necessary.



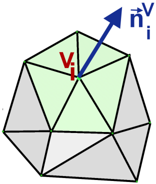
Multiview stereo

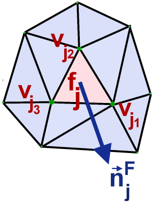
Depth cameras

3D Mesh Denoising

Notations:

- $\mathbf{M} = (\mathbf{V}, \mathbf{E}_0, \mathbf{F}_0)$ - noisy mesh
  - $\mathbf{V}$ - set of vertex positions (isotropic Gaussian noise added)
  - $\mathbf{E}_0$ - set of edges
  - $\mathbf{F}_0$ - set of faces
- $\mathbf{n}_i^V$ - normal at vertex $\mathbf{v}_i$
- $\mathbf{n}_j^F$ - normal on face $\mathbf{f}_j$
- $\mathcal{N}_V(\cdot)$ - vertex neighbourhood
- $\mathcal{N}_F(\cdot)$ - face neighbourhood
- $\widehat{\cdot}$ - an estimate of true value





Problem: Given $\mathbf{M}$, find $\widehat{\mathbf{V}}$.

Classes of mesh denoising methods

Existing methods:

- **Local methods** - Correction is applied locally and iteratively.

  Examples - Field, 1988; Taubin, 2001; Fleishman *et al.*, 2003; Sun *et al.*, 2007; Sun *et al.*, 2008; Zheng *et al.*, 2011.

- **Global methods** - Global cost function is minimised.

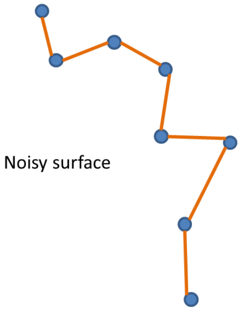
  Examples - Hoppe *et al.*, 1993; Desbrun *et al.*, 1999; Ohtake *et al.*, 2002; Ji *et al.*, 2005; Nealen *et al.*, 2006; Liu *et al.*, 2007; Zheng *et al.*, 2011; He and Schaefer, 2013; Cheng *et al.*, 2014; Wang *et al.*, 2014; Zhang *et al.*, 2015.

Limitations of existing methods

Denoising restricted to the direction of the surface normal

- **Ignore** the true distribution of noise.
- **Move** a mesh vertex along the **normal direction**.
- Residual noise in the tangent planes

> $\Rightarrow$ **Severe distortion of faces**
>
> $\Rightarrow$ Often **face flipping**



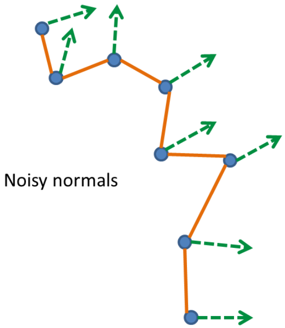
A typical example

Limitations of existing methods

Denoising restricted to the direction of the surface normal

- **Ignore** the true distribution of noise.
- **Move** a mesh vertex along the **normal direction**.
- Residual noise in the tangent planes

> ⇒ **Severe distortion of faces**
>
> ⇒ Often **face flipping**



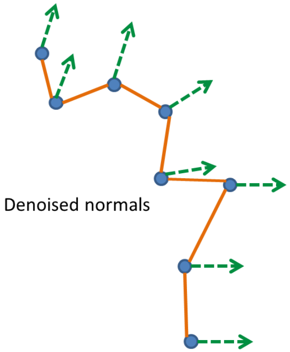
A typical example

Limitations of existing methods

Denoising restricted to the direction of the surface normal

- **Ignore** the true distribution of noise.
- **Move** a mesh vertex along the **normal direction**.
- Residual noise in the tangent planes

> ⇒ **Severe distortion of faces**
>
> ⇒ Often **face flipping**



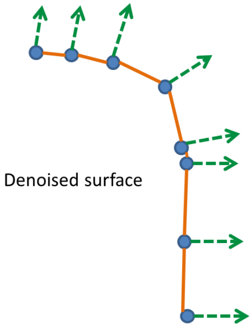
A typical example

Limitations of existing methods

Denoising restricted to the direction of the surface normal

- **Ignore** the true distribution of noise.
- **Move** a mesh vertex along the **normal direction**.
- Residual noise in the tangent planes

> ⇒ **Severe distortion of faces**
>
> ⇒ Often **face flipping**

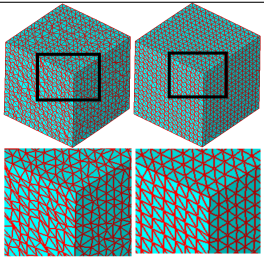


A typical example

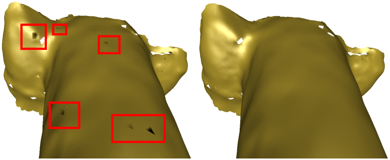
Need Face Fairness!

Face Fairness

- Regularity of face shapes
- Absence of flipped faces



Irregular faces of denoised mesh of a cube. Left image: Sun *et al.*, 2007. Right image: Desired result.



Flipped faces of a denoised mesh. Left image: Sun *et al.*, 2007. Right image: Desired result.

Limitations of existing methods

Local methods - How many times to apply?

- Usually **no convergence** to desired solution.
- Difficulty in defining the **number of times the filter** to be applied for optimal denoising.
- **Need to look** at the actual denoised mesh to decide to stop.

Limitations of existing methods

Other issues

- Significant implicit **volume shrinkage**.
  - **Explicit volume restoration** required.
- Simpler methods : **Smoothing over surface features** such as edges and corners.

Presentation Outline

1. Motivation
2. Our Method
3. Results
4. Conclusion

Our contribution

A **global 3D mesh denoising** method

Properties:

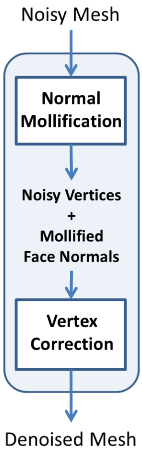
- **Global** formulation.
  - Minimises **sparse, quadratic** cost functions.
  - Yields efficient solutions.
- Allows for vertex correction **in all directions**.
- Enforces a **novel face fairness penalty** that preserves face shapes.
- Good **implicit volume preservation** property.
- **Sharp feature preserving** property.

Our proposed method

Two steps:

1. **Normal Mollification**
   - **Robust global anisotropic** surface normal denoising.

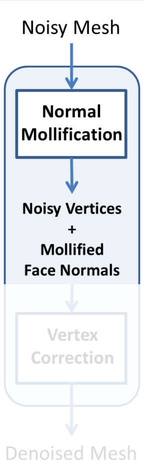
2. **Vertex Correction**
   - **Robust global anisotropic** vertex correction with **face fairness**.



Normal mollification

- **Global** formulation.
- Depends only on the **variance**
- Worl
- For e
  our c
  1.)
  2. **s penalty**

$$\sum_{j\in\mathcal{N}_F(i)} w_{ij}^2 d_s\left(\widehat{\mathbf{n}}_j^F, \widehat{\mathbf{n}}_i^F\right).$$



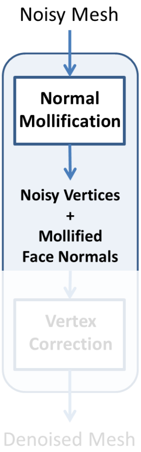
Normal mollification

- Minimise

$$\underbrace{\sum_{i=1}^{N_F} d_o\left(\widehat{\mathbf{n}}_i^F, \mathbf{n}_i^F\right)}_{\text{Data}} + \lambda_N \underbrace{\sum_{i=1}^{N_F} \sum_{j \in \mathcal{N}_F(i)} w_{ij}^2 d_s\left(\widehat{\mathbf{n}}_j^F, \widehat{\mathbf{n}}_i^F\right)}_{\text{Smoothness}}$$

such that $||\widehat{\mathbf{n}}_i^F||_2^2 = 1,\ i = 1, 2, \cdots, N_F.$

- **Gradient Projection Method**:
    - Converges within **10-15** iterations.

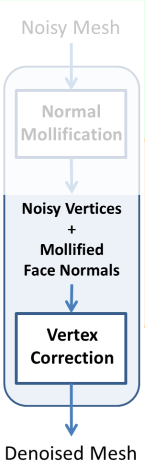


Vertex correction

- **Global** formulation.
- Depends only on the **varian**
- Our cost function has three

1. **Quadratic** data penalty d_f
2. **Quadratic anisotropic La**
    - Bilateral weighting $\Rightarrow$
3. Novel **face fairness term** $d_f^V(\widehat{\mathbf{V}}, \mathbf{V})$:
    - Denoising in the **tangent plane** about a vertex.
    - Carefully considers **edges** and **boundaries**.

$\Rightarrow$ $\|\mathbf{K}(\mathbf{V} - \mathbf{V}_c)\|_2^2$.

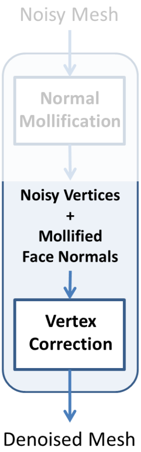


Vertex correction

- Minimise

$$\widehat{\mathbf{V}} = \left(\mathbf{I} + \lambda \mathbf{L}^T\mathbf{L} + \eta \mathbf{K}^T\mathbf{K}\right)^{-1} \left(\mathbf{V} + \eta \mathbf{K}^T\mathbf{K}\mathbf{V}_c\right)$$

- Solved using **efficient sparse linear solvers**.



Presentation Outline

1. Motivation
2. Our Method
3. Results
4. Conclusion

Result: Synthetic Data

Comparison of denoising performance

Object	Error metric	Noisy	Fleishman	Jones	Sun	Zheng	Ours
	Mean NE (°)	17.84	15.84	8.10	0.64	1.00	**0.46**
	Mean NE (°)	25.47	9.57	6.78	5.62	4.55	**3.05**
	Mean NE (°)	17.41	14.98	16.91	2.87	**2.17**	4.80
	Mean NE (°)	29.82	13.66	14.03	13.65	13.32	**8.03**
	Mean NE (°)	24.48	9.05	**8.02**	8.28	8.27	9.09

Normal angle error (NE) and vertex position Euclidean error (VPE)
'NA' denotes 'Not Available'.

Result: Synthetic Data

Comparison of denoising performance

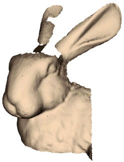
Object	Error metric	Noisy	Fleishman	Jones	Sun	Zheng	Ours
							
	Mean VPE	0.034	0.060	0.036	0.026	NA	**0.013**
							
	Mean VPE	0.040	0.079	0.038	0.032	NA	**0.017**
							
	Mean VPE	0.010	0.016	0.038	0.009	NA	**0.009**
							
	Mean VPE	0.034	0.030	0.030	0.029	NA	**0.025**
							
	MeanVPE	0.167	0.154	0.144	0.141	NA	**0.124**

Normal angle error (NE) and vertex position Euclidean error (VPE)

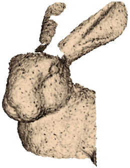
'NA' denotes 'Not Available'.

Result: Synthetic Data

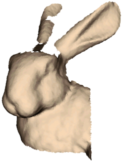
Bunny Face ($N_V = 15861, N_F = 31001$)



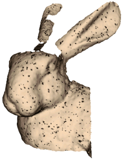
Original Mesh



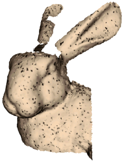
Noisy Mesh



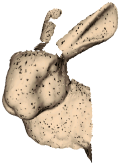
Fleishman et al. 2003



Jones et al. 2003



Sun et al. 2007

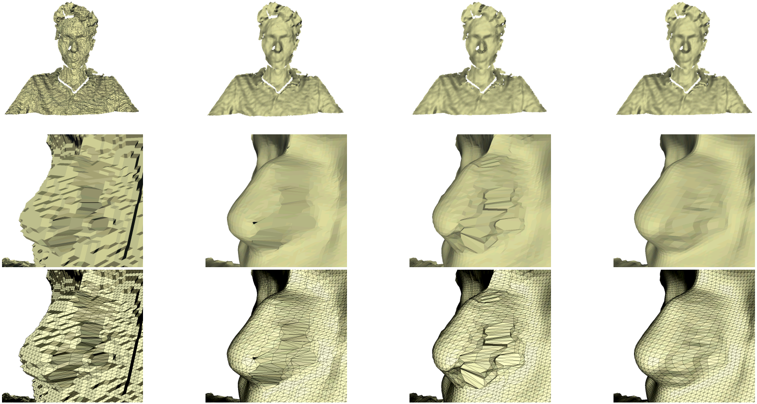


Zheng et al. 2011

Our method

Results: Real Data

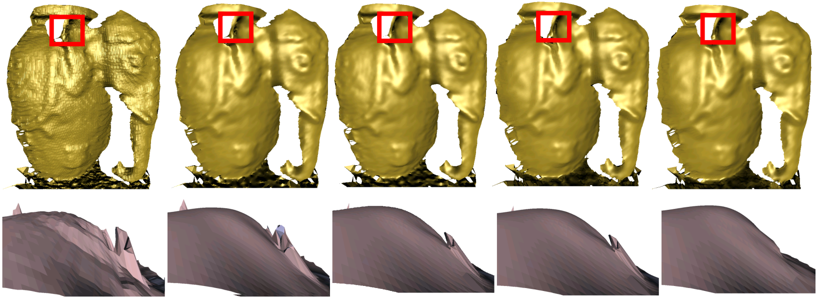
Person ($N_V = 46815$, $N_F = 91392$, Microsoft Kinect depth camera)



Noisy mesh | Fleishman et al., 2003 | Sun et al., 2007 | Our method

Results: Real Data

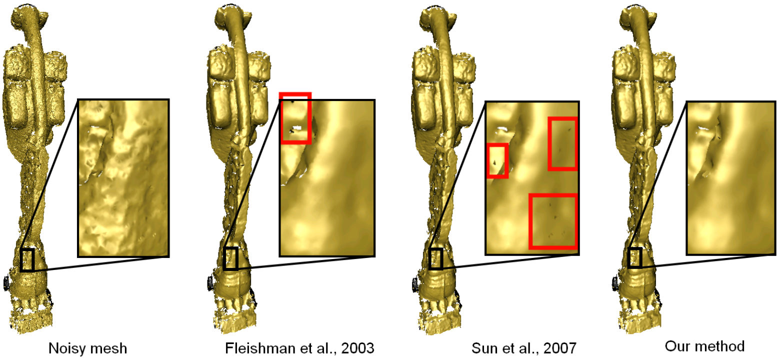
Clay Pot ($N_V = 108731$, $N_F = 216108$, Intel Realsense depth camera)



Noisy Mesh	Fleishman et al. 2003	Jones et al. 2003	Sun et al. 2007	Our method

Results: Real Data

Sculptural Pillar ($N_V = 185546$, $N_F = 360814$, multiview stereo)



Presentation Outline

Conclusion

- A **two-step denoising method** that globally solves for both normal mollification and vertex correction.
- Our vertex correction step accounts for **noise in all directions**.
- Incorporates a **novel face fairness penalty**.
- Ability to provide **good mesh denoising while preserving face fairness** is demonstrated.
- The **superiority of our approach** over other relevant methods is established.

Thank You!

Normal mollification

- Minimise

$$\sum_{i=1}^{N_F} d_o\left(\widehat{\mathbf{n}}_i^F, \mathbf{n}_i^F\right) + \lambda_N \sum_{i=1}^{N_F} \sum_{j \in \mathcal{N}_F(i)} w_{ij}^2 d_s\left(\widehat{\mathbf{n}}_j^F, \widehat{\mathbf{n}}_i^F\right) \text{ subject to } ||\widehat{\mathbf{n}}_i^F||^2 = 1,\ i = 1, 2, \cdots, N_F$$

where λ_N is a regularising parameter depending on the noise variance and the face neighbourhood operator $\mathcal{N}_F(i)$ is the set of faces sharing a common vertex with $\mathbf{f}_i$ and

$$w_{ij}\left(\widehat{\mathbf{n}}_j^F, \widehat{\mathbf{n}}_i^F\right) = \begin{cases} \left(\widehat{\mathbf{n}}_j^{F,T} \widehat{\mathbf{n}}_i^F - t\right) & \text{if } \widehat{\mathbf{n}}_j^{F,T} \widehat{\mathbf{n}}_i^F > t \\ 0 & \text{otherwise} \end{cases}$$

where t is a threshold.

Vertex Correction

Global Laplacian

- **Anisotropic** in nature, defined along the normal directions at the vertices.
- For each vertex $\mathbf{v}_i$,

$$\mathbf{L}_i(\mathbf{v}_i) = \sum_{j \in \mathcal{N}_V(i)} \left(\frac{a_j b_j}{(1+b_j) \sum_{j \in \mathcal{N}_V(i)} a_j} \right) \left(\mathbf{n}_j^F \mathbf{n}_j^{F,T} \right) \left(\mathbf{v}_i - \frac{(\mathbf{v}_{j_1} + \mathbf{v}_{j_2} + \mathbf{v}_{j_3})}{3} \right) \quad (1)$$

- Anisotropic bilateral weights:

$$a_j = \exp\left(-\frac{\left(\mathbf{n}_j^T \Delta \mathbf{v}_{ij}\right)^2}{2\sigma_1^2} \right), b_j = \exp\left(-\frac{\left\| \Delta \mathbf{v}_{ij} \right\|_2^2}{2\sigma_2^2} \right).$$

Vertex Correction

Face fairness penalty

- For a single denoised vertex $\hat{\mathbf{v}}_i$,

$$d_f^V(\hat{\mathbf{v}}_i) = \left\| r_i (\mathbf{I} - \mathbf{n}_i^V \mathbf{n}_i^{V,T})(\hat{\mathbf{v}}_i - \mathbf{v}_{c,i}) \right\|_2^2 \qquad (2)$$

$\mathbf{v}_{c,i}$ - centroid of the $\mathcal{N}_V(i)$ around the vertex $\mathbf{v}_i$

$$r_i = \begin{cases} 0 & \text{if } \mathbf{v}_i \in \mathbf{V}^B \\ 0 & \text{else if } \beta < 0 \\ \beta & \text{otherwise} \end{cases}$$

$\beta = \min_{p,q \in \mathcal{N}_V(i)} (\mathbf{n}_p^{F,T} \mathbf{n}_q^F - \delta)$ and δ is a small positive value.

- $$\hat{\mathbf{V}} = \left(\mathbf{I} + \lambda \mathbf{L}^T \mathbf{L} + \eta \mathbf{K}^T \mathbf{K}\right)^{-1} \left(\mathbf{V} + \eta \mathbf{K}^T \mathbf{K} \mathbf{V}_c\right).$$

  where $\mathbf{K}$ is formed from Eqn. 2 and $\mathbf{V}_c$ is the concatenated vector formed from $\mathbf{v}_{c,i}, i = 1, 2, \cdots, N_V$.