gruvi graphics + usability + visualization

SFU SIMON FRASER UNIVERSITY THINKING OF THE WORLD

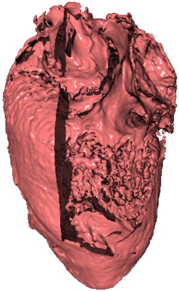
Edge Aware Anisotropic Diffusion for 3D Scalar Data on Regular Lattices

Zahid Hossain
MSc. Thesis Defense

Simon Fraser University
Burnaby, BC
Canada.

NSERC CRSNG

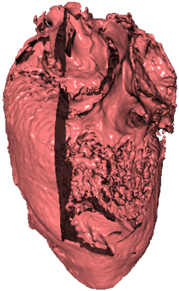
Motivation



Original Sheep's heart data

Motivation

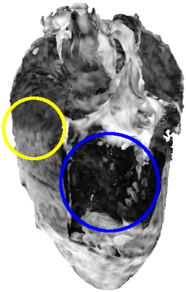
Existing gradient based anisotropic diffusion



Original Sheep's heart data

Motivation

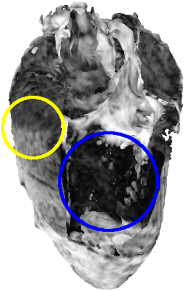
Existing gradient based anisotropic diffusion



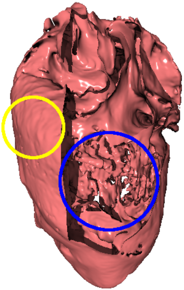
Gradient magnitude map

Motivation

Existing gradient based anisotropic diffusion



Gradient magnitude map



Proposed method

Our method is consistent and free of undesirable artifacts

Previous Work

- Perona and Malik (**PM**) Model (TPAMI, 1990):

$$\frac{\partial f}{\partial t} = \mathbf{div}\left(h(\|\nabla f\|)\nabla f\right)$$

Previous Work

SFU

- Perona and Malik (**PM**) Model (TPAMI, 1990):

$$\frac{\partial f}{\partial t} = \mathbf{div}\left(h(\|\nabla f\|)\nabla f\right)$$

- Carmona et al.(TIP, 1998) generalization in 2D:

$$\frac{\partial f}{\partial t} = \overbrace{h}^{\textbf{SF}}\left(af_{\mathbf{nn}} + bf_{\mathbf{vv}}\right), \quad \textbf{SF = Stopping Function}$$

where $\mathbf{v} = \mathbf{n}^{\perp}$.

Previous Work

- Perona and Malik (**PM**) Model (TPAMI, 1990):

$$\frac{\partial f}{\partial t} = \mathbf{div}\left(h(\|\nabla f\|)\nabla f\right)$$

- Carmona et al.(TIP, 1998) generalization in 2D:

$$\frac{\partial f}{\partial t} = \overbrace{h}^{\textbf{SF}}\left(af_{\mathbf{nn}} + bf_{\mathbf{vv}}\right), \quad \textbf{SF = Stopping Function}$$

where $\mathbf{v} = \mathbf{n}^{\perp}$.

- 3D extension by Gerig et al.(TMI, 1992): Isotropic on the tangent plane of the gradient.

Previous Work: Trouble in 3D

Original

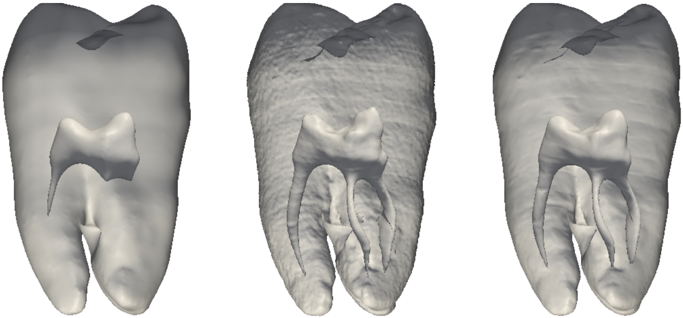
Previous Work: Trouble in 3D

PM model in 3D

Original

Previous Work: Trouble in 3D

SFU



PM model in 3D | Original | Curvatures taken into account

Therefore Weickert classified PM and higher order diffusion processes as **non-linear** rather than **anisotropic**.

Previous Work: In 3D

SFU

- Whitaker (VBC, 1994) and Krissian et al.(SCALE-SPACE, 1997) (**KM model**), took curvatures into account
  - The stopping function had gradient magnitude based parameter.

Previous Work: In 3D

- Whitaker (VBC, 1994) and Krissian et al.(SCALE-SPACE, 1997) (**KM model**), took curvatures into account
  - The stopping function had gradient magnitude based parameter.
- **Level Set:** The definition of **edge** is based on the curvature that is measured <u>on the surface of the level set</u>.

Previous Work: In 3D

SFU

- Whitaker (VBC, 1994) and Krissian et al.(SCALE-SPACE, 1997) (**KM model**), took curvatures into account
  - The stopping function had gradient magnitude based parameter.
- **Level Set:** The definition of **edge** is based on the curvature that is measured on the surface of the level set.
- **De-noising:**
  - **Bilateral filtering**, **mean-shift** filtering and **non-linear** diffusion are equivalent: Barash et al., (IVC, 2004).
  - A recent methods: **SRNRAD** and **ORNRAD** (TIP, 2009).

Contributions

1. Removal of the gradient magnitude based parameter

Contributions

1. Removal of the gradient magnitude based parameter
2. Dynamic adaptation of anisotropy

Contributions

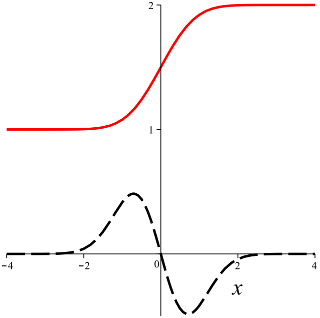
1. Removal of the gradient magnitude based parameter
2. Dynamic adaptation of anisotropy
3. Application in de-noising

Contributions

1. Removal of the gradient magnitude based parameter
2. Dynamic adaptation of anisotropy
3. Application in de-noising
4. Implementation recipe in arbitrary regular lattices

The Proposed Method

Proposed Method: Edge



The red solid curve is the **error function** while the dashed black curve is the second derivative of it.

A common technique in 2D image processing.

Four Objectives of Our Diffusion

SFU

Objectives

1. No diffusion along the gradient direction.

This prevents blurring across an edge.

Four Objectives of Our Diffusion

SFU

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.

Check for $f_{\mathbf{nn}} = 0$. No problem in constant homogeneous regions.

Four Objectives of Our Diffusion

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.

Similar to Krissian et al.(1997), i.e the KM model.

Four Objectives of Our Diffusion

SFU

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.
4. Diffuse on the tangent plane where the local iso-surface has similar principal curvatures.

For example, on the surface of a sphere or a slab.

Modelling the PDE

- Consider the 1D heat, also known as **diffusion**, equation:

$$\frac{\partial f}{\partial t} = c f_{\mathbf{xx}}$$

Modelling the PDE

- Consider the 1D heat, also known as **diffusion**, equation:

$$\frac{\partial f}{\partial t} = c f_{\mathbf{xx}}$$

- A 3D extension of the above:

$$\frac{\partial f}{\partial t} = h f_{\mathbf{r}_1 \mathbf{r}_1} + g f_{\mathbf{r}_2 \mathbf{r}_2} + w f_{\mathbf{nn}}$$

  Where the orthonormal bases $[\mathbf{r}_1, \mathbf{r}_2, \mathbf{n}]$ are **min curvature**, **max curvature** and **normal** directions respectively.

Modelling the PDE

- Consider the 1D heat, also known as **diffusion**, equation:

$$\frac{\partial f}{\partial t} = c f_{\mathbf{xx}}$$

- A 3D extension of the above:

$$\frac{\partial f}{\partial t} = h f_{\mathbf{r}_1\mathbf{r}_1} + g f_{\mathbf{r}_2\mathbf{r}_2} + w f_{\mathbf{nn}}$$

  Where the orthonormal bases $[\mathbf{r}_1, \mathbf{r}_2, \mathbf{n}]$ are **min curvature**, **max curvature** and **normal** directions respectively.

- Setting $g = \tau h$ and $w = \eta h$ we can simplify the above

$$\frac{\partial f}{\partial t} = h \cdot \left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2} + \eta f_{\mathbf{nn}} \right), \quad \tau, \eta \in [0, 1]$$

Modelling the PDE: Objective 1

SFU

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.
4. Diffuse on the tangent plane where the local iso-surface has similar principal curvatures.

$$\frac{\partial f}{\partial t} = h \cdot \left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2} + \eta f_{\mathbf{nn}} \right), \quad \tau, \eta \in [0, 1]$$

Modelling the PDE: Objective 1

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.
4. Diffuse on the tangent plane where the local iso-surface has similar principal curvatures.

$$\frac{\partial f}{\partial t} = h \cdot \left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2} + \eta f_{\mathbf{nn}} \right), \quad \tau, \eta \in [0, 1]$$

Set $\eta = 0$

$$\frac{\partial f}{\partial t} = h \cdot \left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2} \right), \quad \tau \in [0, 1]$$

Modelling the PDE: Objective 2

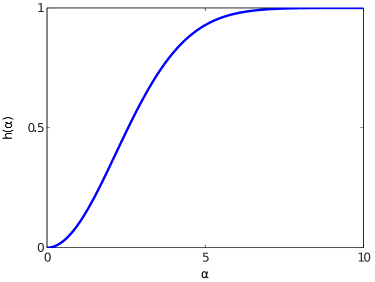
Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.
4. Diffuse on the tangent plane where the local iso-surface has similar principal curvatures.

Modelling the PDE: Objective 2

Define h

$$h(\alpha) = 1 - (0.9)^{\left(\frac{\alpha}{\sigma}\right)^2}, \quad \sigma \in \mathbb{R}$$



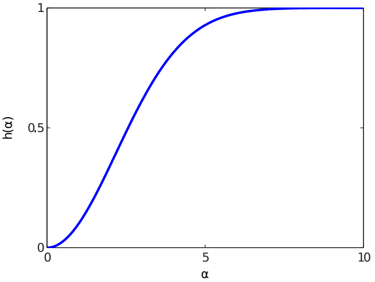
Plot of $h(\alpha)$

Modelling the PDE: Objective 2

Define h

$$h(\alpha) = 1 - (0.9)^{\left(\frac{\alpha}{\sigma}\right)^2}, \quad \sigma \in \mathbb{R}$$

Takes the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$ as argument.



Plot of $h(\alpha)$

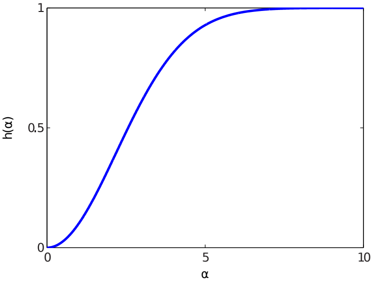
Modelling the PDE: Objective 2

Define h

$$h(\alpha) = 1 - (0.9)^{\left(\frac{\alpha}{\sigma}\right)^2}, \quad \sigma \in \mathbb{R}$$

Takes the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$ as argument.

So far: $\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2}\right)$



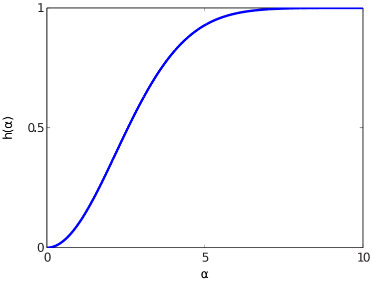
Plot of $h(\alpha)$

Modelling the PDE: Objective 2

Define h

$$h(\alpha) = 1 - (0.9)^{\left(\frac{\alpha}{\sigma}\right)^2}, \quad \sigma \in \mathbb{R}$$

Takes the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$ as argument.



Plot of $h(\alpha)$

So far: $\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2}\right)$

- Diffusion **stops** when $f_{\mathbf{nn}} \approx 0$, i.e. around the edge locations.
- Homogeneous regions remains unaffected.

Modelling the PDE: Objectives 3 and 4

SFU

Objectives

1. No diffusion along the gradient direction.
2. Diffusion stopped around the edge locations.
3. Diffuse along the direction of the minimum curvature.
4. Diffuse on the tangent plane where the local iso-surface has similar principal curvatures.

Modelling the PDE: Objectives 3 and 4

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2}\right)$$

Modelling the PDE: Objectives 3 and 4

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2}\right)$$

$$\tau = \begin{cases} \left(\frac{\kappa_1}{\kappa_2}\right)^{2\lambda} & \text{where } |\kappa_2| > 0, \lambda \in \mathbb{Z} \\ 1 & \kappa_2 = 0 \end{cases}$$

where,

κ_1 : minimum curvature

κ_2 : maximum curvature

Modelling the PDE: Objectives 3 and 4

SFU

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2}\right)$$

$$\tau = \begin{cases} \left(\frac{\kappa_1}{\kappa_2}\right)^{2\lambda} & \text{where } |\kappa_2| > 0, \lambda \in \mathbb{Z} \\ 1 & \kappa_2 = 0 \end{cases}$$

where,

κ_1 : minimum curvature

κ_2 : maximum curvature

- Usually $\lambda = 2$ works well
- Curvatures are measured from a blurred version of the data f_ρ
- τ is replaced by τ_ρ

Simplification

So far we have

$$\frac{\partial f}{\partial t} = \overbrace{h(f_{\mathbf{nn}})}^{\textbf{Stopping Function}}\left(f_{\mathbf{r}_1\mathbf{r}_1} + \underbrace{\tau_\rho}_{\textbf{Isotropy Function}} f_{\mathbf{r}_2\mathbf{r}_2}\right)$$

Simplification

So far we have

$$\frac{\partial f}{\partial t} = \overbrace{h(f_{\mathbf{nn}})}^{\text{Stopping Function}} \left(f_{\mathbf{r}_1\mathbf{r}_1} + \underbrace{\tau_\rho}_{\text{Isotropy Function}} f_{\mathbf{r}_2\mathbf{r}_2} \right)$$

Which can be simplified to

$$\vdots$$

$$\frac{\partial f}{\partial t} = -h(f_{\mathbf{nn}}) \|\nabla f\| (\kappa_1 + \tau_\rho \kappa_2)$$

Simplification

So far we have

$$\frac{\partial f}{\partial t} = \overbrace{h(f_{\mathbf{nn}})}^{\text{Stopping Function}} \left(f_{\mathbf{r}_1\mathbf{r}_1} + \underbrace{\tau_\rho}_{\text{Isotropy Function}} f_{\mathbf{r}_2\mathbf{r}_2} \right)$$

Which can be simplified to

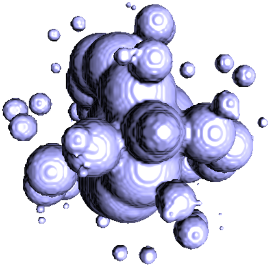
$$\vdots$$

$$\frac{\partial f}{\partial t} = -h(f_{\mathbf{nn}}) \|\nabla f\| (\kappa_1 + \tau_\rho \kappa_2)$$

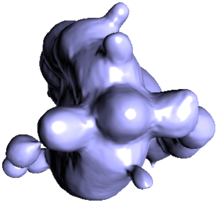
Has a similar form to **Mean Curvature Motion (MCM)**, yet essentially different.

MCM: $$\frac{\partial f}{\partial t} = -\|\nabla f\| (\kappa_1 + \kappa_2)$$

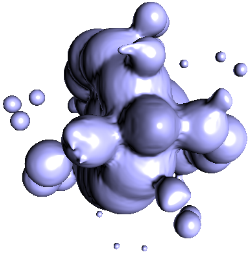
Significance of the Stopping Function



Original



MCM



Our method

Our: $\frac{\partial f}{\partial t} = -h(f_{\mathbf{nn}})\|\nabla f\|(\kappa_1 + \tau_\rho \kappa_2), \quad h(f_{\mathbf{nn}}) = 1 - (0.9)^{\left(\frac{f_{\mathbf{nn}}}{\sigma}\right)^2}, \quad \sigma \in \mathbb{R}$

MCM: $\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_1 + \kappa_2)$

Stability

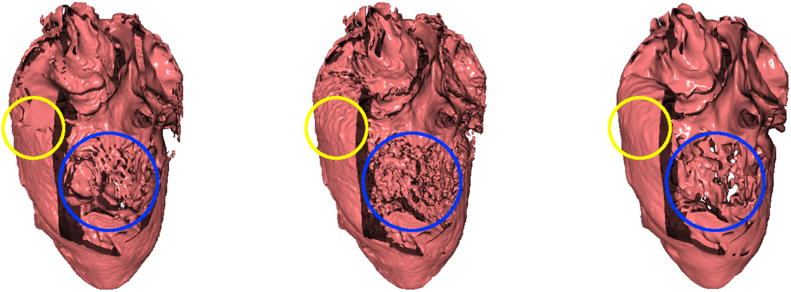
- *Courant-Friedrichs-Lewy* (CFL) revealed the following necessary condition for stability:

$$\Delta t \leq C\frac{d^2}{2}, \quad d = \min\{\Delta x, \Delta y, \Delta z\}, \quad C = \textit{Courant}\ \text{Number}$$

- $C \leq 0.8$ is stable for most practical purposes.

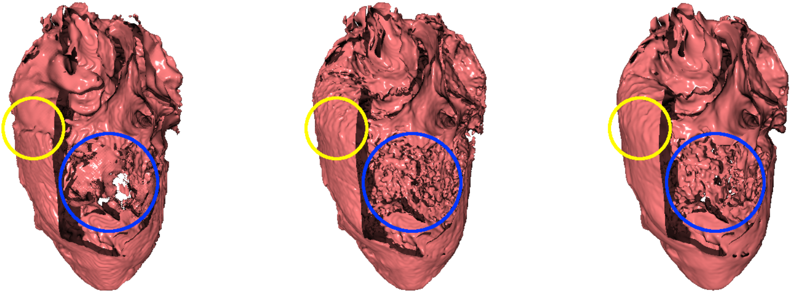
Results

Results: Consistent Smoothing



KM method: $k = 40$ Original Our method: $\sigma = 1$

Results: Consistent Smoothing



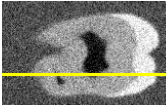
KM method: $k = 60$ | Original | Our method: $\sigma = 10$

De-noising Properties

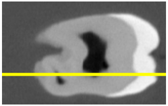
Note

We **used the same parameter settings** for all our de-noising experiments across all datasets and noise types.

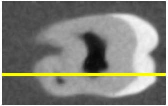
Results: De-noising (Gaussian)



Noisy, SNR=12.89

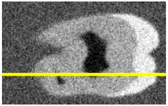


Original

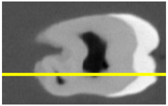


Diffused, SNR=26.12

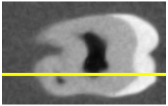
Results: De-noising (Gaussian)



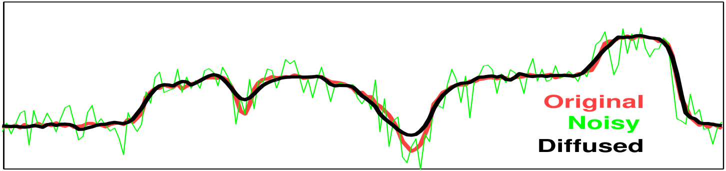
Noisy, SNR=12.89



Original

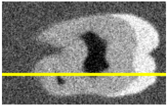


Diffused, SNR=26.12

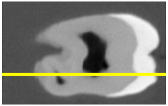


Profile

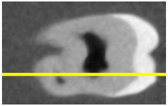
Results: De-noising (Gaussian)



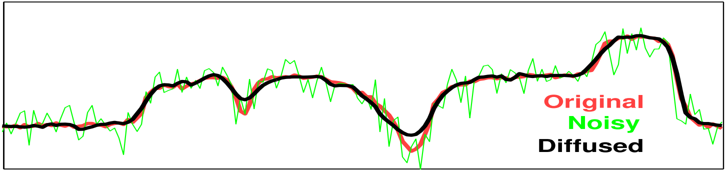
Noisy, SNR=12.89



Original

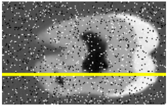


Diffused, SNR=26.12

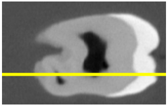


Profile

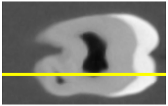
Results: De-noising (Salt and Pepper)



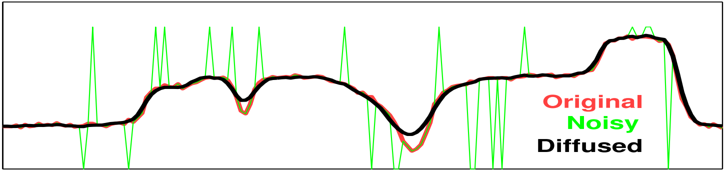
Noisy, SNR=12.89



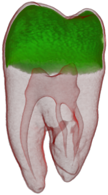
Original



Diffused, SNR=26.12



Profile



SFU

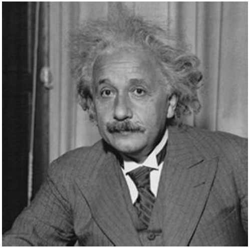
De-noising Comparison

Note

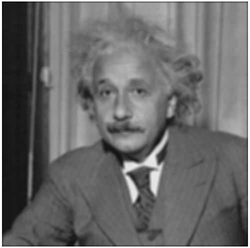
- Comparison with **SRNRAD** and **ORNRAD** (Krissian and Aja-Fernández, **2009**).
- We **did not change** any parameter settings.

Quantitative Measures

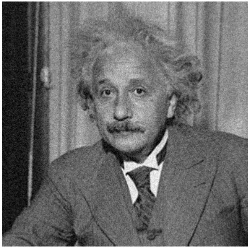
- Mean Squared Error **(MSE)**
- Structural Similarity Index **(SSIM)**
- Quality Index Based on Local Variance **(QILV)**



Original



Blurred
SSIM: **0.8235 (Good)**
QILV: 0.4793 (Bad)

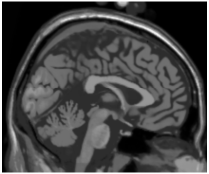


Gaussian noise
SSIM: 0.4643 (Bad)
QILV: **0.7192 (Good)**

De-noising: Comparison

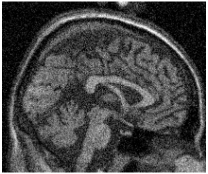
SFU

Gaussian noise

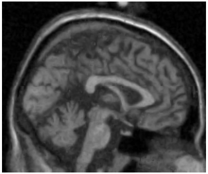


Original

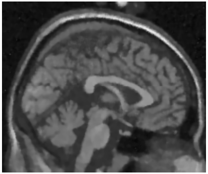
	MSE	SSIM	QILV
Noise	558.750	0.540	0.503
SRNRAD	162.017	0.816	**0.886**
ORNRAD	153.873	0.819	0.859
Our	**75.878**	**0.900**	0.860



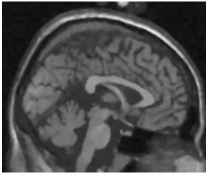
Noisy



Our



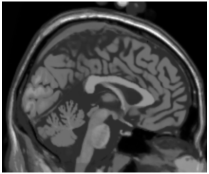
SRNRAD



ORNRAD

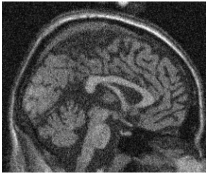
De-noising: Comparison

Rician noise

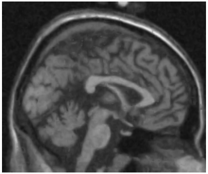


Original

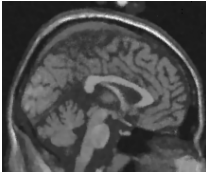
	MSE	SSIM	QILV
Noise	450.558	0.561	0.712
SRNRAD	232.459	<u>0.792</u>	**0.913**
ORNRAD	<u>226.405</u>	**0.795**	<u>0.889</u>
Our	**173.851**	**0.795**	0.820



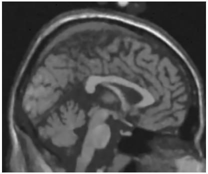
Noisy



Our



SRNRAD



ORNRAD

De-noising: Remarks

- Our method performs similarly if not better in many cases.

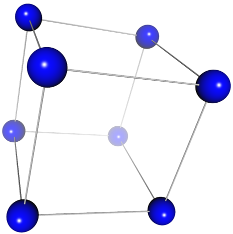
De-noising: Remarks

SFU

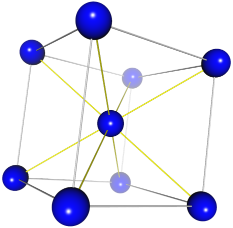
- Our method performs similarly if not better in many cases.
- Matlab implementation is $\approx \mathbf{14}\times$ **faster** than a multi-threaded C/C++ based **ORNRAD**.
  - Our formulation is simpler
  - **ORNRAD** requires, beside complex statistical measurements, eigen decomposition of structure tensor.

Extension Framework for Regular Lattices

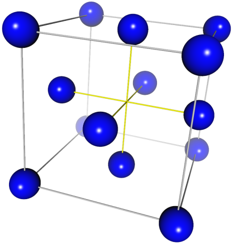
Lattices in 3D



Cartesian Cubic (CC)



Body Centric Cubic (BCC)

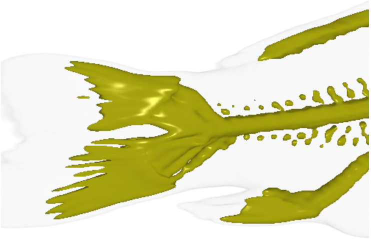


Face Centric Cubic (FCC)

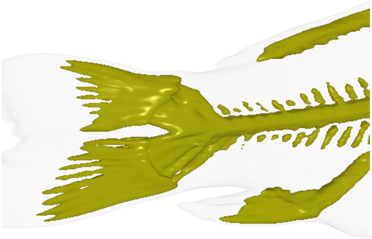
A regular lattice can be characterized by a matrix $\mathbf{L}$

Motivation For Other Lattices

Tai Meng et al. (2011) and Entezari et al. (2008)



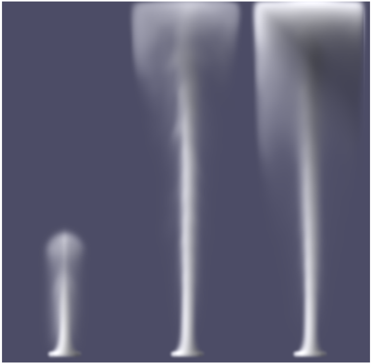
CC



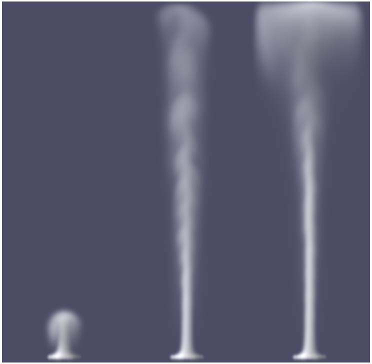
BCC

Motivation For Other Lattices (Contd.)

Alim et al. (2009)



CC



BCC

Taylor Expansion of Discrete Convolution

SFU

Discrete convolution of $f[\mathbf{k}]$ with a filter $\Delta[\mathbf{k}]$ can be written as the following:

$$f * \Delta = f_r^w[\mathbf{k}] = \sum_{\mathbf{n}} \underbrace{a_{\mathbf{n}}^{\Delta}}_{\text{Taylor Coefficient}} \cdot D^{\mathbf{n}} f[\mathbf{k}] \quad f[\mathbf{k}] = f(\mathbf{L}\mathbf{k})$$

Taylor Expansion of Discrete Convolution

SFU

Discrete convolution of $f[\mathbf{k}]$ with a filter $\Delta[\mathbf{k}]$ can be written as the following:

$$f * \Delta = f_r^w[\mathbf{k}] = \sum_{\mathbf{n}} \underbrace{a_{\mathbf{n}}^{\Delta}}_{\text{Taylor Coefficient}} \cdot D^{\mathbf{n}} f[\mathbf{k}] \quad f[\mathbf{k}] = f(\mathbf{L}\mathbf{k})$$

And $a_{\mathbf{n}}^{\Delta}$ is a **linear** function of the filter weights $\Delta[\mathbf{k}]$

Linear System

SFU

$$f * \Delta = f_r^w[\mathbf{k}] = \sum_{\mathbf{n}} a_{\mathbf{n}}^{\Delta} \cdot D^{\mathbf{n}} f[\mathbf{k}], \quad f[\mathbf{k}] = f(\mathbf{L}\mathbf{k})$$

To extract the $\mathbf{u}$-th derivative $D^{\mathbf{u}} f[\mathbf{k}]$, correct up to a polynomial order of n, the following must be satisfied

$$a_{\mathbf{n}}^{\Delta} = \begin{cases} 0, & \mathbf{n} \in \eta_d^{[0,n]} \text{ and } \mathbf{n} \neq \mathbf{u} \\ 1, & \mathbf{n} = \mathbf{u}. \end{cases}$$

Linear System (Contd.)

Forms a linear system with the unknown filter weights

$$w_i = \Delta[\mathbf{k}_i]$$

$$\begin{bmatrix} a^{\Delta}_{\mathbf{n}_1} \\ a^{\Delta}_{\mathbf{n}_2} \\ \vdots \\ a^{\Delta}_{\mathbf{n}_m} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Linear System (Contd.)

Forms a linear system with the unknown filter weights
$w_i = \Delta[\mathbf{k}_i]$

$$\begin{bmatrix} a^{\Delta}_{\mathbf{n}_1} \\ a^{\Delta}_{\mathbf{n}_2} \\ \vdots \\ a^{\Delta}_{\mathbf{n}_m} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,m} \\ \vdots & \vdots & \vdots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,m} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathbf{G} \qquad \mathbf{W} = \mathbf{C}$$

Infinite Solution Space

- The linear system is often not full-rank

Infinite Solution Space

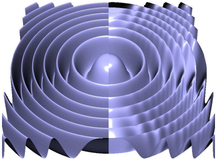
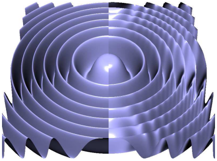
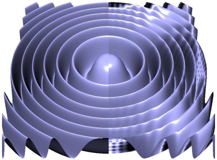
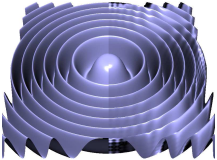
- The linear system is often not full-rank
- Finding a suitable solution by:
  - Imposing symmetry/anti-symmetry in the filter geometry
  - Minimizing error in the higher polynomial order

Infinite Solution Space

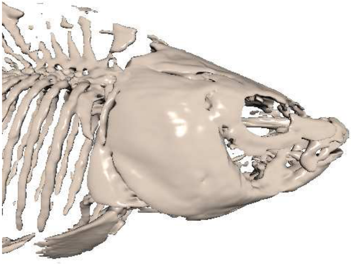
- The linear system is often not full-rank
- Finding a suitable solution by:
  - Imposing symmetry/anti-symmetry in the filter geometry
  - Minimizing error in the higher polynomial order
- Usually yields a filter that has small support

Results

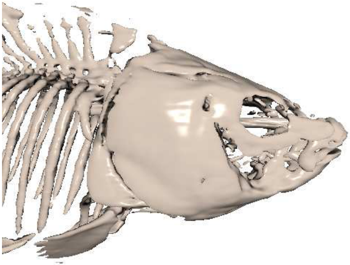
Quality of Normals: Synthetic Data

	2-EF	4-EF
CC	 (Mag=3.13, Angle=49.4°)	 (2.79, 44.1°)
BCC	 (2.88, 39.2°)	 (2.42, 28.7°)

Quality of Normals: Real Data (Iso-Surface)

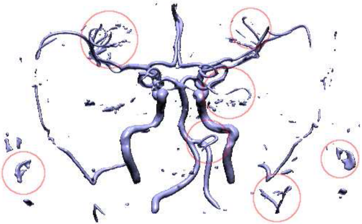


CC (4-EF)

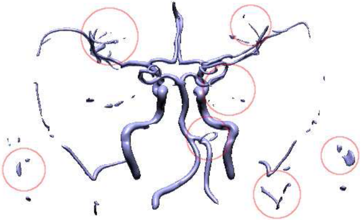


BCC (4-EF)

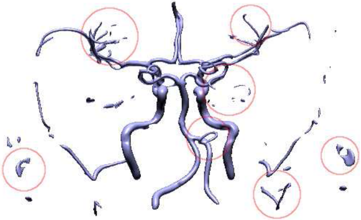
Effects of Higher Order Filters (CC)



Original



2-EF



4-EF

Conclusion

SFU

We developed . . .

- **An anisotropic diffusion:**
  - Consistent smoothing
  - Less effort to choose a parameter
  - Performs similarly, if not better, to a recent de-noising method
  - Fast and easy to implement

Conclusion

We developed . . .

- **An anisotropic diffusion:**
  - Consistent smoothing
  - Less effort to choose a parameter
  - Performs similarly, if not better, to a recent de-noising method
  - Fast and easy to implement

- **A framework to design discrete derivative filters:**
  - Works on arbitrary regular lattice
  - Any derivative could be estimated
  - Size of the filter could be specified
  - PDE implementation on any regular lattice is made possible

Future Work

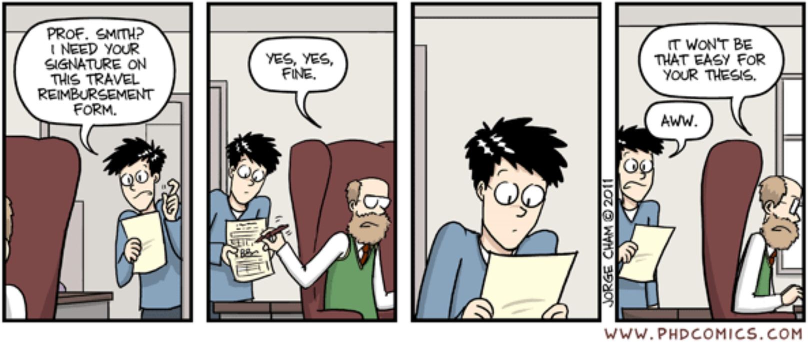
- Actual implementation in BCC and FCC
- De-noising performance in BCC and FCC
- Different functions for τ
- Alternate edge detectors

Acknowledgements

SFU

I would like to thank:

- My supervisor, co-supervisor and the examining committee
- All the GrUVi members for their relentless support
- Natural Science and Engineering Research Council of Canada (NSERC) for funding my research



Thank you for your attention :)