VisIO

SFU
SIMON FRASER UNIVERSITY
THINKING OF THE WORLD

gruvi graphics + usability + visualization

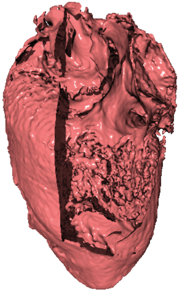
Edge Aware Anisotropic Diffusion for 3D Scalar Data

Zahid Hossain   Torsten Möller

Simon Fraser University
Burnaby, BC
Canada.

NSERC
CRSNG

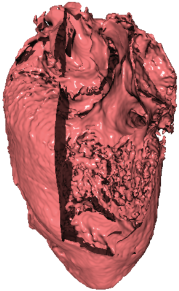
Motivation



Original Sheep's heart data

Motivation

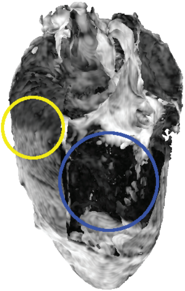
Existing gradient based anisotropic diffusion



Original Sheep's heart data

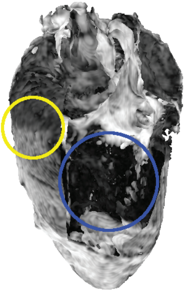
Motivation

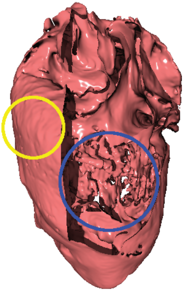
Existing gradient based anisotropic diffusion



Gradient magnitude map

Motivation



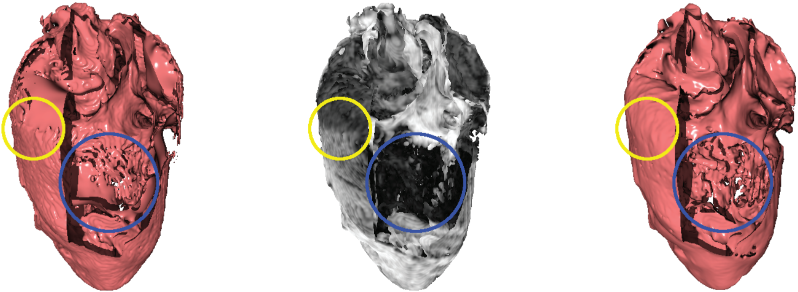


Existing gradient based anisotropic diffusion

Gradient magnitude map

Proposed method

Motivation



Existing gradient based anisotropic diffusion

Gradient magnitude map

Proposed method

Note

Our method yields a consistent smoothing over an iso-surface without 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

- Perona and Malik (**PM**) Model (TPAMI, 1990):
  $\frac{\partial f}{\partial t} = \mathbf{div}\,(h(\|\nabla f\|)\nabla f)$
- Carmona et al.(TIP, 1998) generalized the PM model in 2D:

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

SF = Stopping Function

where $\mathbf{v}$ is the orthonormal direction of the normal $\mathbf{n}$.

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) generalized the PM model 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 $\mathbf{v}$ is the orthonormal direction of the normal $\mathbf{n}$.
- Extension in 3D by Gerig et al.(TMI, 1992): remains isotropic on the tangent plane of the gradient.

Previous Work: Trouble in 3D

Original

Previous Work: Trouble in 3D

PM model in 3D

Original

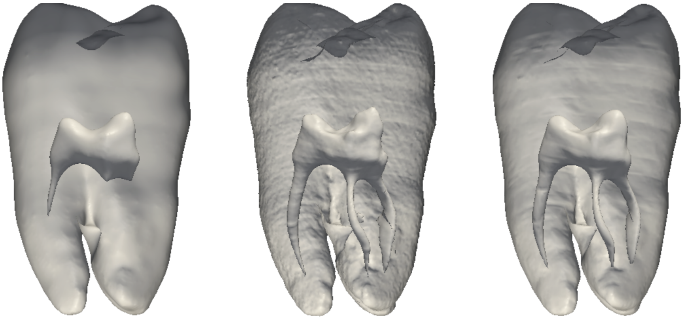
Previous Work: Trouble in 3D

PM model in 3D

Original

Curvatures taken into account

Previous Work: Trouble in 3D



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

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - The stopping function had gradient magnitude based parameter.

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - 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.

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - 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>.
  - Our definition is based on the second derivative measured <u>across the level sets</u>.

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - 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.
  - Our definition is based on the second derivative measured across the level sets.
- **De-noising:**
  - **Bilateral filtering**, **mean-shift** filtering and **non-linear** diffusion are equivalent: Barash et al., (IVC, 2004).

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - 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.
  - Our definition is based on the second derivative measured across the level sets.
- **De-noising:**
  - **Bilateral filtering**, **mean-shift** filtering and **non-linear** diffusion are equivalent: Barash et al., (IVC, 2004).
  - A recent methods namely **SRNRAD** and **ORNRAD** (TIP, 2009).

Previous Work: In 3D

- Krissian et al.(SCALE-SPACE, 1997) developed a true anistropic diffusion, (**KM model**), taking curvatures into account as proposed by Whitaker (VBC, 1994).
  - 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.
  - Our definition is based on the second derivative measured across the level sets.
- **De-noising:**
  - **Bilateral filtering**, **mean-shift** filtering and **non-linear** diffusion are equivalent: Barash et al., (IVC, 2004).
  - A recent methods namely **SRNRAD** and **ORNRAD** (TIP, 2009).
    - We will compare our method with these two.

Contributions

1. Removal of the gradient magnitude based parameter.

Contributions

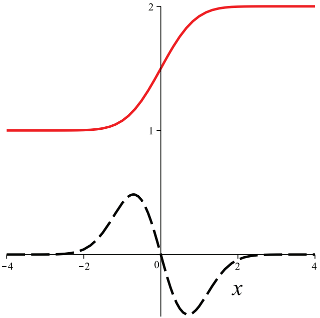
1. Removal of the gradient magnitude based parameter.
2. Dynamic adaptation of anisotropy based on local curvatures.

Contributions

1. Removal of the gradient magnitude based parameter.
2. Dynamic adaptation of anisotropy based on local curvatures.
3. Application in de-noising and visualization.

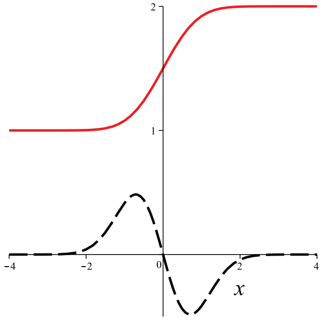
The Proposed Method

Proposed Method: Edge



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

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 to detect **edges** in 2D image processing.
- Basis of 2D transfer functions, Kindlmann and Durkin (VIS 1998).

Four Objectives of Our Diffusion

Objectives

1. No diffusion will be performed along the gradient direction.

Note

This one is different from the classical methods. This prevents blurring across an edge.

Four Objectives of Our Diffusion

Objectives

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

Note

Check for $f_{\mathbf{nn}} = 0$. Note it does not create any problem in constant homogeneous regions.

Four Objectives of Our Diffusion

Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.

Note

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

Four Objectives of Our Diffusion

Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.
4. Diffuse isotropically on the tangent plane of the normal $\mathbf{n}$ in regions where the local iso-surface has similar principal curvatures.

Note

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

Modeling the PDE

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

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

Modeling the PDE

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

$$\frac{\partial f}{\partial t} = cf_{\mathbf{xx}}$$

- A 3D extension of the above:

$$\frac{\partial f}{\partial t} = hf_{\mathbf{r}_1\mathbf{r}_1} + gf_{\mathbf{r}_2\mathbf{r}_2} + wf_{\mathbf{nn}}$$

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

Modeling the PDE

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

$$\frac{\partial f}{\partial t} = cf_{\mathbf{xx}}$$

- A 3D extension of the above:

$$\frac{\partial f}{\partial t} = hf_{\mathbf{r}_1\mathbf{r}_1} + gf_{\mathbf{r}_2\mathbf{r}_2} + wf_{\mathbf{nn}}$$

  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

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

Modeling the PDE: Objective 1

Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.
4. Diffuse isotropically on the tangent plane of the normal $\mathbf{n}$ in regions where the local iso-surface has similar principal curvatures.

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

Modeling the PDE: Objective 1

Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.
4. Diffuse isotropically on the tangent plane of the normal $\mathbf{n}$ in regions 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 $\eta = 0$

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

Modeling the PDE: Objective 2

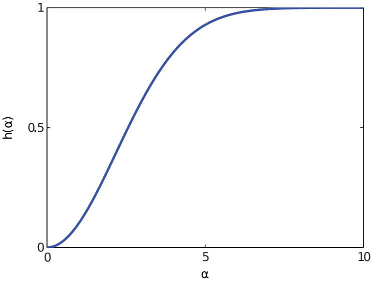
Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.
4. Diffuse isotropically on the tangent plane of the normal $\mathbf{n}$ in regions where the local iso-surface has similar principal curvatures.

Modeling the PDE: Objective 2

Define h

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



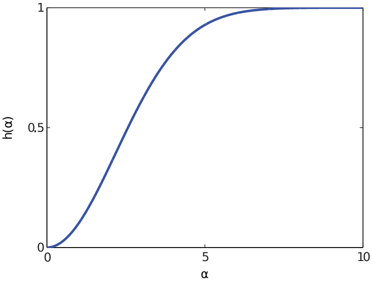
Plot of $h(\alpha)$

Modeling the PDE: Objective 2

Define h

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

h can take, as argument, the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$.



Plot of $h(\alpha)$

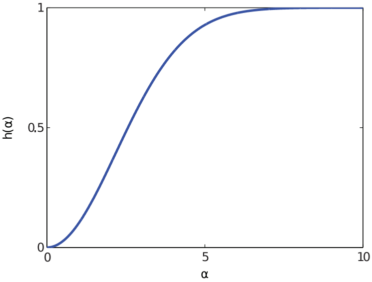
Modeling the PDE: Objective 2

Define h

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

h can take, as argument, the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$.

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



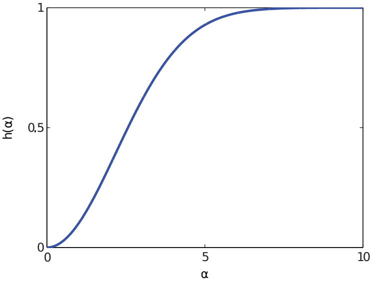
Plot of $h(\alpha)$

Modeling the PDE: Objective 2

Define h

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

h can take, as argument, the directional second derivative $f_{\mathbf{nn}}$ along the normal $\mathbf{n}$.



Plot of $h(\alpha)$

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

Note

- Diffusion **stops** when $f_{\mathbf{nn}} \approx 0$, i.e. around the edge locations.
- Constant homogeneous regions, where $f_{\mathbf{nn}} \approx 0$, would not be affected by diffusion anyway.

Modeling the PDE: Objectives 3 and 4

Objectives

1. No diffusion will be performed along the gradient direction.
2. Diffusion will be stopped around the edge locations.
3. Diffusion will be performed anisotropically along the direction of the minimum curvature.
4. Diffuse isotropically on the tangent plane of the normal $\mathbf{n}$ in regions where the local iso-surface has similar principal curvatures.

Modeling the PDE: Objectives 3 and 4

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

Modeling 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

such that, $|\kappa_1| \leq |\kappa_2|$.

Modeling 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

such that, $|\kappa_1| \leq |\kappa_2|$.

Note

- Both the curvatures are measured from a smoothed version of the data f_ρ with a Gaussian filter having a small variance ρ^2.
- So we will use the notation τ_ρ for the rest of the presentation to depict this and call it **isotropy function**.

Simplification

So far we have

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\Big(f_{\mathbf{r}_1\mathbf{r}_1} + \overbrace{\tau_\rho}^{\textbf{IF}} f_{\mathbf{r}_2\mathbf{r}_2}\Big), \quad \textbf{IF=Isotropy Function}$$

Simplification

So far we have

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \overbrace{\tau_\rho}^{\textbf{IF}} f_{\mathbf{r}_2\mathbf{r}_2}\right),$$

IF=Isotropy Function

Which can be simplied to

$$\vdots$$

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

Simplification

So far we have

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})\left(f_{\mathbf{r}_1\mathbf{r}_1} + \overbrace{\tau_\rho}^{\textbf{IF}} f_{\mathbf{r}_2\mathbf{r}_2}\right),$$

IF=Isotropy Function

Which can be simplied to

$$\vdots$$

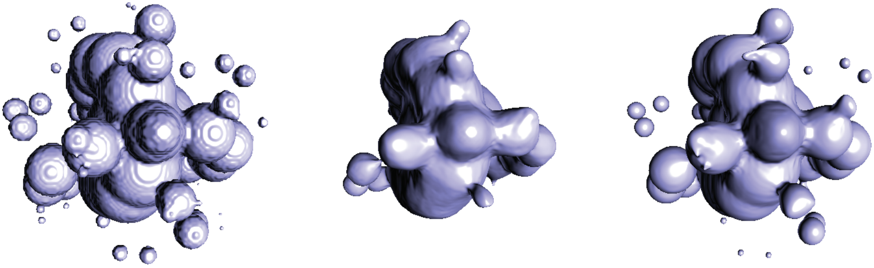
$$\frac{\partial f}{\partial t} = -h(f_{\mathbf{nn}})\|\nabla f\|(\kappa_1 + \tau_\rho \kappa_2)$$

Note

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

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

Results: Significance of σ



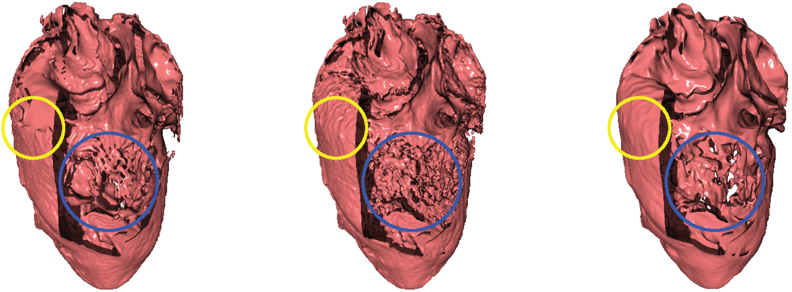
Original | $h(\cdot) = 1$ | $\sigma = 40$

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

Results

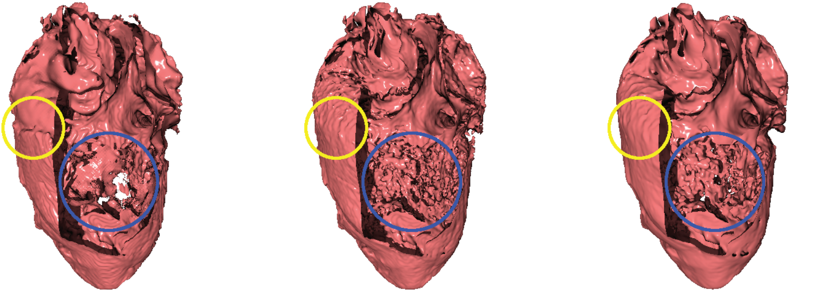
Results: Consistent Smoothing



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

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

Results: Consistent Smoothing



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

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

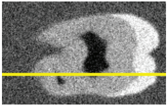
De-noising Properties

De-noising Properties

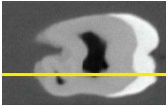
Note

- We **used the same parameter settings** for all our de-noising experiments across all datasets and noise types.
- With a unit grid spacing in all dimensions we empirically found that $\Delta t \leq 0.4$ is stable for most practical purposes.

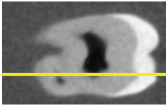
Results: De-noising (Gaussian)



Noisy, SNR=12.89

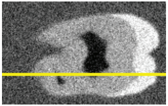


Original

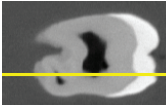


Diffused, SNR=26.12

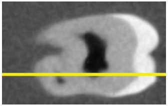
Results: De-noising (Gaussian)



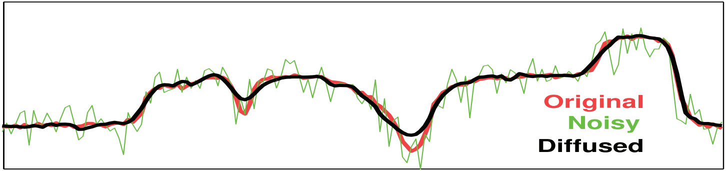
Noisy, SNR=12.89



Original

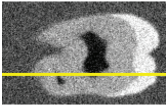


Diffused, SNR=26.12

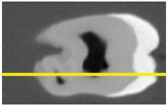


Profile

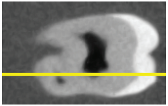
Results: De-noising (Gaussian)



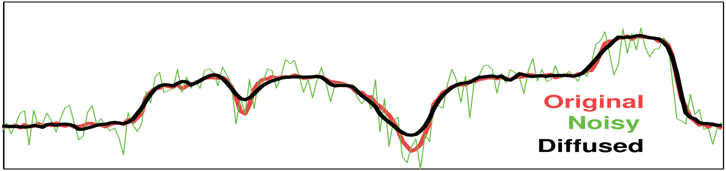
Noisy, SNR=12.89



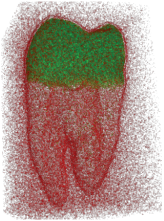
Original



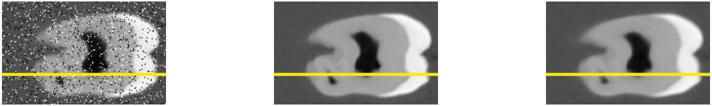
Diffused, SNR=26.12



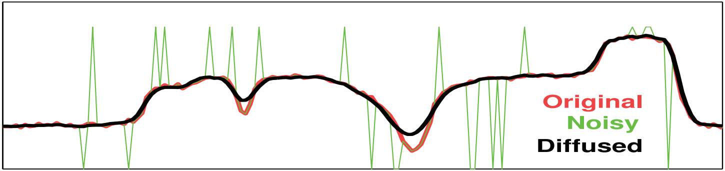
Profile



Results: De-noising (Salt and Pepper)



Noisy, SNR=12.89 | Original | Diffused, SNR=26.12



Profile

De-noising Comparison

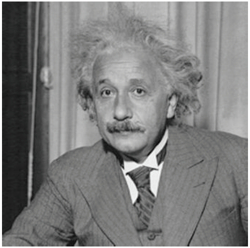
De-noising Comparison

Note

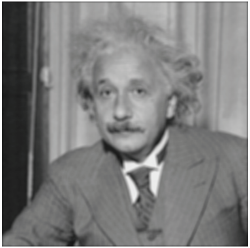
- We compared our anisotropic diffusion with two recent de-noising methods: **SRNRAD** and **ORNRAD** proposed by Krissian and Aja-Fernández **(2009)**.
- We **did not change** any parameter settings.

Quantitative Measures

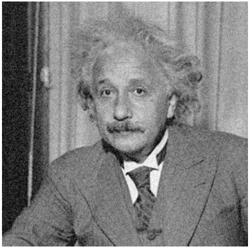
- Mean Squared Error **(MSE)**.
- Structural Similarity Index **(SSIM)**.
- Quality Index Based on Local Variance **(QILV)**.



Original



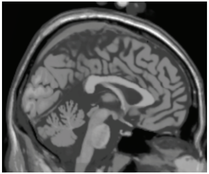
Blurred
SSIM: 0.8235 **(Good)**
QILV: 0.4793 (Bad)



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

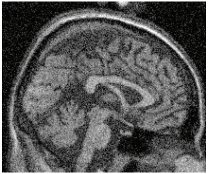
De-noising: Comparison

Gaussian noise

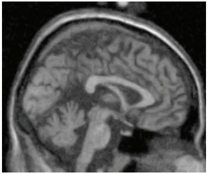


Original

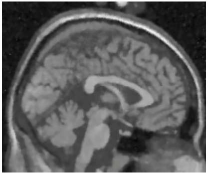
	MSE	SSIM	QILV
Noise	558.750	0.540	0.503
SRNRAD	162.017	0.816	**0.886**
ORNRAD	153.873	0.819	0.859
Our	**75.878**	**0.900**	0.860



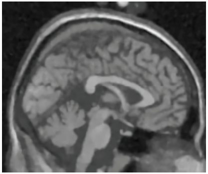
Noisy



Our



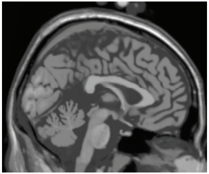
SRNRAD



ORNRAD

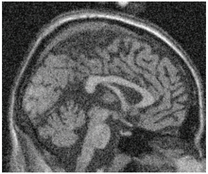
De-noising: Comparison

Rician noise

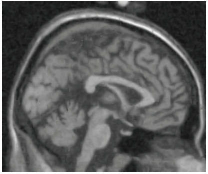


Original

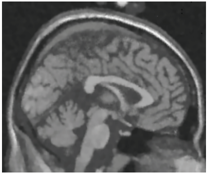
	MSE	SSIM	QILV
Noise	450.558	0.561	0.712
SRNRAD	232.459	0.792	**0.913**
ORNRAD	226.405	**0.795**	0.889
Our	**173.851**	**0.795**	0.820



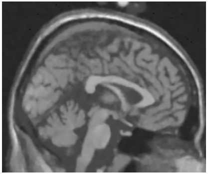
Noisy



Our



SRNRAD



ORNRAD

De-noising: Comparison

Note

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

De-noising: Comparison

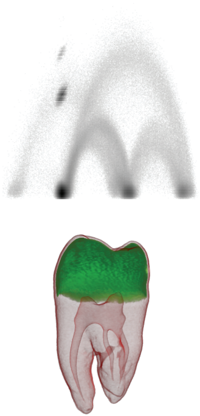
Note

- Our method performs similarly if not better in many cases.
- And our Matlab implementation is $\approx \mathbf{14}$ times faster than a multi-threaded C/C++ based implementation of **ORNRAD**.

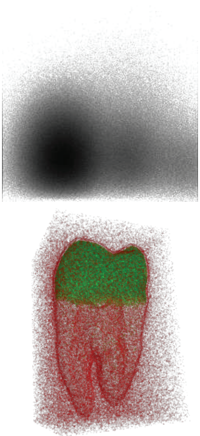
Impact on Visualization

2D Transfer Function (With Noise)

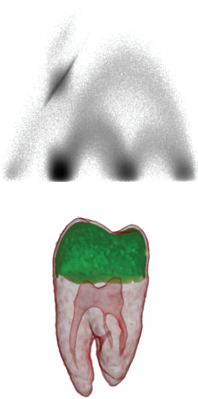
Gaussian noise



Original



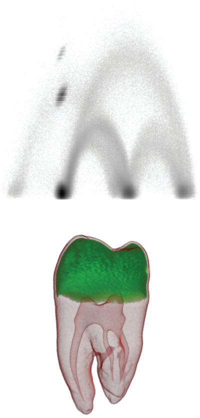
Noisy



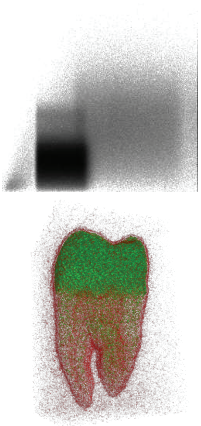
Diffused

2D Transfer Function (With Noise)

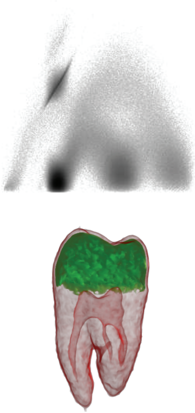
Speckle noise



Original



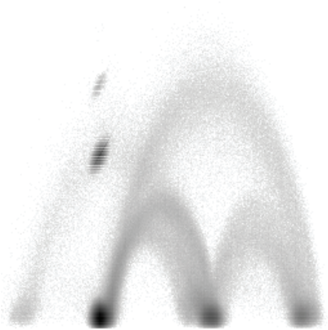
Noisy



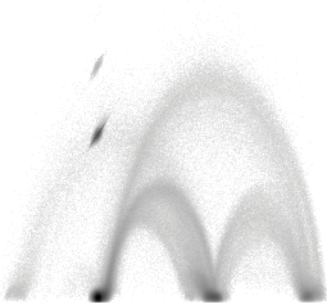
Diffused

2D Transfer Function (Without Noise)

Tooth Dataset



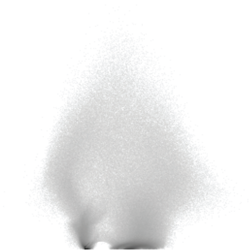
Original



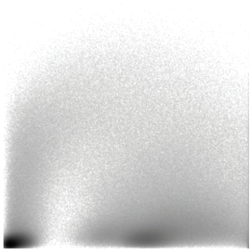
Diffused with our method

2D Transfer Function

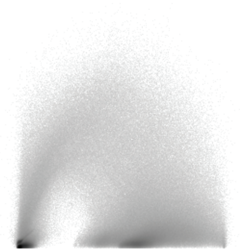
Sheep's Heart Dataset



KM method



Original



Our method

Conclusion

We developed an anistropic diffusion that:

- Smoothes consistently over an iso-surface
- Requires much less effort to choose a parameter value.
- Performs similarly, if not better, to a recent de-noising method.
- Is easy to implement and fast.
- Enhances 2D transfer functions.

Acknowledgements

We would like to thank:

- Dr. Karl Krissian and Dr. Santiago Aja-Fernández for providing supports in various occasions.
- Dr. Philippe Thévenaz for the Sphere phantom data.
- Natural Science and Engineering Research Council of Canada (NSERC) for funding this project.

Thank you for your attention :)

Contact Information

Zahid Hossain
Email: zha13@sfu.ca
http://www.cs.sfu.ca/~zha13/personal

Dr. Torsten Möller
Email: torsten@cs.sfu.ca
http://www.cs.sfu.ca/~torsten