Edge Aware Anisotropic Diffusion for 3D Scalar Data

Zahid Hossain and Torsten Möller, *Member, IEEE*

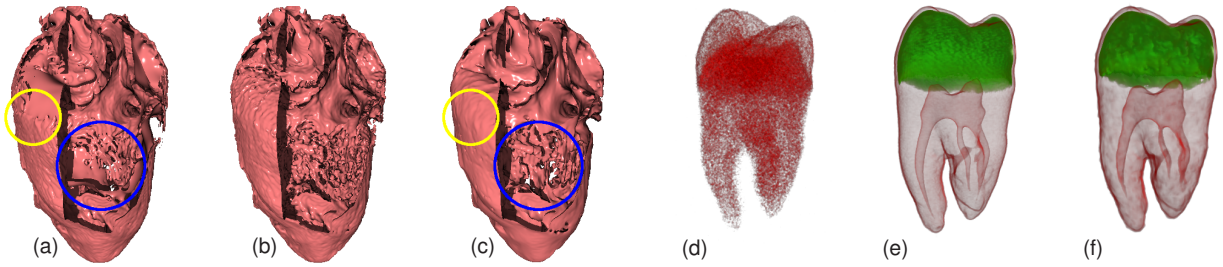


Fig. 1: The left half of the figure demonstrates the consistency in smoothing of our method compared to the existing method. The right half of the figure demonstrates the de-noising capabilities of our method. All the images from (a-c) were obtained by rendering an iso-surface of 153. (a) Diffused with an existing diffusion model proposed by Krissian et al. [20] with $k = 40$, and 100 iterations (b) The original Sheep's heart data. (c) Diffused with our method with $\sigma = 1$ and the same number of iterations. The yellow circle indicates a region where the iso-surface has both high and medium range gradient magnitude, and the blue circle marks a region where the gradient magnitude is much lower. Note the inconsistent smoothing in (a) inside the yellow circle. (d) The tooth data contaminated with Poisson noise (SNR=12.8) (e) The original tooth data (f) Diffused with our method (SNR=25.76) with $\sigma = 1$ and 25 iterations. We used the exact same transfer function to render all the images in (d-f).

Abstract—In this paper we present a novel anisotropic diffusion model targeted for 3D scalar field data. Our model preserves material boundaries as well as fine tubular structures while noise is smoothed out. One of the major novelties is the use of the directional second derivative to define material boundaries instead of the gradient magnitude for thresholding. This results in a diffusion model that has much lower sensitivity to the diffusion parameter and smoothes material boundaries consistently compared to gradient magnitude based techniques. We empirically analyze the stability and convergence of the proposed diffusion and demonstrate its de-noising capabilities for both analytic and real data. We also discuss applications in the context of volume rendering.

Index Terms—Anisotropic diffusion, PDE, De-noising, Scale-Space, Principal Curvatures.

1 Introduction

The core of volume visualization and volume rendering has been the extraction and presentation of the salient features in the volume. Often times, the data at hand has been corrupted by noise (e.g. Ultrasound [39], MRI or data range scanners [32]) or the salient features of interest are fine structures, like the tubular vessel structures [20] or the cell walls in microscopy [24]. Usually, these types of data can not be properly processed by a volume rendering pipeline, since both transfer function based approaches [33] as well as topological approaches [8] will break down and not yield a comprehensible view of the data. In all of these cases, a pre-processing step is needed in order to prepare the data for visualization and analysis purposes. A very powerful framework for this purpose is diffusion.

A simple Gaussian blur usually destroys a lot of features together with noise artifacts in an isotropic / indiscriminate way. Hence, the concept of anisotropic diffusion has been introduced by Perona and Malik in 1990 [26] (summarized in Section 2) and has become one of the most popular techniques in image and volume processing. Many different variants of anisotropic diffusion have since been introduced. One of the core drawbacks, however, of any diffusion model has been the non-intuitive setting of some parameters attached with it. This is typically rooted in the fact that the diffusion is controlled by the gradient magnitude of the underlying function. In most practical cases the distribution of gradient strength of salient boundaries is not known a priori. In fact a single gradient magnitude threshold rarely defines all the salient features within the data. Hence, the diffusion algorithm often needs to be re-fined for each new data set or each new application.

In this paper we attempt to address these problems of the existing non-linear diffusion techniques. Our main contributions are:

- We derive an anisotropic diffusion equation with the following features (see Section 3.1):
  - No diffusion is performed along the gradient direction.
  - Diffusion is stopped around the edge locations.
  - Diffusion is performed anisotropically along the direction of the minimum curvature.
  - Isotropic diffusion on the tangent plane of the normal in regions where the local iso-surface is isotropic in shape.
- We create a stopping function, that is based on the second derivative in the gradient direction, which allows us to create a robust diffusion algorithm, insensitive to parameter tuning (see Section 3.1).

Section 4 introduces an efficient way to compute our diffusion equation using the principal curvatures and therefore reducing the computational burden inherit in the scheme. In Section 5 we will discuss some properties of our proposed diffusion model along with aptly demonstrating its advantage over the gradient based method [20]. We will

- *The authors are with the Graphics, Usability, and Visualization (GrUVi) Lab, School of Computing Science, Simon Fraser University, Burnaby BC V5A 1S6, Canada, E-mail: {zha13,torsten}@cs.sfu.ca.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

follow this by demonstrating the de-noising property of our model in Section 6. In Section 6.1, we will perform an empirical analysis of convergence and stability and then critically compare the de-noising performance of our method with a very recent anisotropic diffusion based de-noising filter [19]. Finally, in Section 7 we will discuss some potential applications in volume visualization.

2 Previous Work

Throughout the paper we will use the notation f and t to denote a real valued scalar function and the time dimension respectively.

2.1 The Perona and Malik Model

To alleviate the problem of isotropic diffusion, which is similar to Gaussian blurring, Perona and Malik [26] proposed an anisotropic diffusion scheme, which we will refer to as the PM model for brevity, given by the following:

$$\frac{\partial f}{\partial t} = \mathbf{div}\left(h(\|\nabla f\|)\nabla f\right) \tag{1}$$

The function $h(\|\nabla f\|)$, termed *stopping function*, is usually a monotonically decreasing function with function values around one for smaller arguments. Perona and Malik [26] also proposed two such h functions and one of them is given below:

$$h(\alpha) = e^{-\frac{1}{2}\left(\frac{\alpha}{S}\right)^2}, \tag{2}$$

With such an $h(\cdot)$ function this approach does tend to preserve certain edges given the parameter S in Equation (2) is chosen carefully which is often not trivial and data dependent.

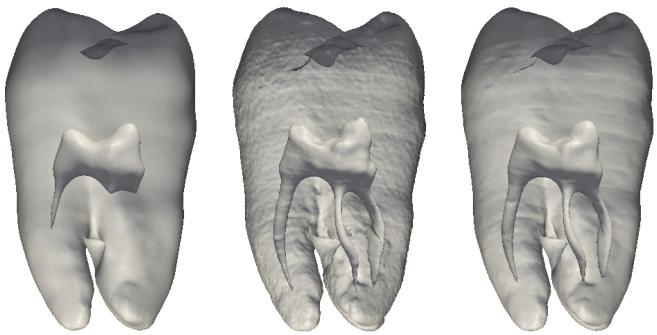
2.2 Generalizations in 2D

Carmona et al. [7] generalized the classical PM [26] model (in 2D) in a more intuitive way - performing diffusions along the gradient and the orthogonal direction to the gradient as given by the following:

$$\frac{\partial f}{\partial t} = c(\cdot)\left(a(\cdot)f_{\mathbf{nn}} + b(\cdot)f_{\mathbf{vv}}\right) \tag{3}$$

Here a and b are some scalar functions modulating diffusion along the gradient direction $\mathbf{n} = \nabla f/\|\nabla f\|$ and the orthogonal direction to the gradient, $\mathbf{v} = \mathbf{n}^{\perp}$ respectively. Here, c is a scalar function, usually called the *stopping function*, that modulates the overall diffusion. The notation $f_{\mathbf{nn}}$ and $f_{\mathbf{vv}}$ denotes the directional second derivative along the gradient direction $\mathbf{n}$ and the orthogonal direction $\mathbf{v}$ respectively.

2.3 Extensions to 3D



(a) Without curvatures (b) Original (c) With curvatures

Fig. 2: Iso-surface (iso-value=100) rendering of the tooth data showing effects of taking principal curvatures into account during diffusion. (a) Diffused isotropically on the tangent plane of gradients without taking principal curvatures into account. (b) Without any diffusion. (c) Diffused anisotropically taking principal curvatures into account.

A straightforward extension of the PM model to 3D, as it was generalized by Gerig et al. [11], remains isotropic on the tangent plane of the gradient. This isotropic behavior on the tangent plane may destroy fine tubular structures which are vital, for example, in 3D medical imaging. Figure 2 clearly shows how tubular structures get destroyed when diffusion is performed isotropically on the tangent plane without taking principal curvatures[1] into account. Therefore Weickert [37] classifies the PM model and higher order diffusion processes as *non-linear* rather than *anisotropic*. Thus far, the majority of previous work on these higher order PDEs are based on 2D solutions. On the other hand, a true anisotropic diffusion in arbitrary dimension is usually derived from the *diffusion tensor* notation [38] and has the following form:

$$\frac{\partial f}{\partial t} = \mathbf{div}(\mathbf{D}\nabla f) \tag{4}$$

were $\mathbf{D}$ is a matrix known as *diffusion tensor*.

To address this, Krissian et al. [20] proposed a true anisotropic diffusion model (referred to as KM model in the remainder of the paper) whereby diffusion would be performed primarily along the direction of the minimum curvature. However, their underlying formulation was based on that of the gradient based PM model. They have a gradient threshold parameter k and the edges in a volume get implicitly defined by locations where $\|\nabla f\| > k$. However, the tuning of this parameter is difficult and very much data and application dependent. Different values for this parameter can lead to drastically different smoothing effects as we will demonstrate later. Therefore the KM model inherits similar problems related to this threshold parameter as the classical PM model. Nonetheless, the KM model based on principal curvatures is truly anisotropic in nature. In a later work, Krissian [18] proposed a flux-based anisotropic diffusion which is based on a directional first derivative, i.e. the gradient measured along a direction vector, while the author himself acknowledged the difficulty of choosing a correct threshold parameter for this directional first derivative.

Recently Mosaliganti et al. [24] reaffirm the problems of the gradient threshold based *stopping function* as found in PM or KM models and proposed a new anisotropic diffusion in 3D that is able to automatically detect and enhance specifically plate like structures in a 3D microscopy image of cell membranes. Beside being specific to a particular problem domain, i.e. detecting cell membranes which are largely planar, their method has at least four different user tunable parameters which makes it hard to apply in a practical setting.

Therefore, the problem of developing a robust general purpose anisotropic diffusion that respects edges in a 3D volume in a meaningful way remains open.

On the other hand, several interesting works have been done on anisotropic diffusion using level sets, for example Nemitz et al. [25] evolved a separate level set function to capture the tubular structures of 3D angiography data. This work is different in the sense that the level set function attempts to restore tubular structure using shape priors even when they may be broken. Other interesting level set methods were proposed by Preusser et al. [28] and Tasdizen et al. [32] where diffusion is performed on a level set and the definition of *edge* is based on *curvatures* that is measured on the surface of the level set. This is different from our method where *edge* is defined by the directional second derivative along gradient and this is measured across level sets.

2.4 Denoising

A variant of anisotropic diffusion, also known as SRAD, has been developed to specifically de-noise speckle noise in 2D by Yu and Action [40], which was then extended to 3D by Krissian et al. [21] and Sun et al. [31]. Both SRAD and 3D SRAD use a local statistical measure to define the *stopping function*. Very recently Krissian and Aja-Fernández [19] proposed a noise-driven anisotropic diffusion that is able to remove *Rician* noise from a 3D MRI volume, and this method too uses statistical measures similar to that of the 3D SRAD [21]. Both of these methods require the user to specify a *region of interest* for the estimation of noise.

State-of-the art image de-noising techniques are often based on techniques such as *bilateral filtering* [34], *mean-shift* filtering [10], or *non-local means* [6]. These techniques are often related to diffusion processes. Barash et al. [4] showed that bilateral filtering, mean-shift,

[1] Principal curvatures, measured at a point, are the minimum and the maximum curvatures of the level set surface passing through that point.

and *non-linear diffusion* are indeed equivalent and use gradient magnitude to decide on the amount of diffusion/smoothing.

3 Proposed Anisotropic Diffusion

Kindlmann and Durkin [14] used the directional second derivative along the normal direction as a measure for edge locations. They pointed out that, for a simple 1D case, an edge could be modeled using the *error function* [16] as plotted in Figure 3. This is a fair assumption as all measuring devices are band-limited and so sharp edges get convolved with a Gaussian type point spread function upon sampling. Therefore an edge location can be defined by the zero-crossing of the second derivative, a technique commonly used in computer vision [22]. The same idea can be applied in 3D by measuring the directional second derivative along the normal direction and checking for zero-crossing to define the edge/boundary locations.

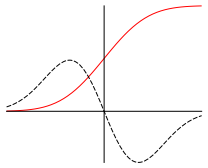


Fig. 3: The red solid curve is the *error function* while the dashed black curve is the second derivative of it. Note the second derivative crosses zero at the edge of the *error function*.

For the rest of the paper we will restrict our attention to a 3D scalar function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$. We will use the notation $f_{\mathbf{v}}$ to denote a directional first derivative along a unit vector $\mathbf{v}$ which is simply given by $f_{\mathbf{v}} = \langle \nabla f, \mathbf{v} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Similarly, we will use the notation $f_{\mathbf{vv}}$ to denote a directional second derivative measured along a unit vector $\mathbf{v}$ and this is given by $f_{\mathbf{vv}} = \mathbf{v}^{\mathsf{T}}\mathbf{H}\mathbf{v}$, where $\mathbf{H}$ is the 3D Hessian (see Section A of the Appendix in the supplementary material for details). Therefore, using the notation $\mathbf{n} = \nabla f / \|\nabla f\|$ as the normal vector we will denote the directional second derivative along the normal with $f_{\mathbf{nn}}$.

In the following subsection we will derive a PDE with the following objectives in mind:

- **O-1 No diffusion will be performed along the gradient direction.** This is one of the major differences our proposed diffusion model has with that of the classical ones [20, 26]. An edge in a 3D volume will be a surface which is perpendicular to the normal $\mathbf{n}$. Not diffusing along $\mathbf{n}$ prevents blurring across an edge.
- **O-2 Diffusion will be stopped around the edge locations.** Diffusion can be stopped in locations where $f_{\mathbf{vv}} = 0$, a condition which will be satisfied in both constant homogeneous regions and edge locations. However, stopping diffusion in constant homogeneous regions creates no problem as diffusion in such regions would not have any effect.
- **O-3 Diffusion will be performed anisotropically along the direction of the minimum curvature.** In accordance to the work of Krissian et al. [20], this motivation was derived from the fact that fine tubular structures, e.g. blood vessels in a CT scan, get preserved.
- **O-4 Diffuse isotropically on the tangent plane of the normal n in regions where the local iso-surface has similar principal curvatures.** On the surface of a sphere, for example, where both the principal curvatures are fairly close to each other, it makes more sense to diffuse isotropically on the tangent plane of $\mathbf{n}$, rather than choosing one direction, which might lead to undesirable artifacts, as is the case with Krissian et al. [20].

3.1 Our PDE Model

Consider a simple 1D heat equation as follows:

$$\frac{\partial f}{\partial t} = c f_{\mathbf{xx}} \tag{5}$$

The solution of the above Equation (5) can be approximated very well by convolving the function locally with a 1D Gaussian kernel. We will use this insight to design our new anisotropic diffusion in 3D with the goals described in the previous subsection in mind.

In 3D, we can use the directional derivatives along the directions given by three orthonormal bases, $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{n}$ at a point and write our diffusion equation as the following:

$$\frac{\partial f}{\partial t} = h(\cdot) f_{\mathbf{r}_1\mathbf{r}_1} + g(\cdot) f_{\mathbf{r}_2\mathbf{r}_2} + w(\cdot) f_{\mathbf{nn}} \tag{6}$$

where $h(\cdot), g(\cdot)$ and $w(\cdot)$ are some scalar functions and the vector $\mathbf{n}$ is the normal direction. We emphasize that the PDE model given in Equation (6) is different from the diffusion tensor model (4). At this point we will take the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ to be the directions associated with the minimum curvature, κ_1, and maximum curvature, κ_2, respectively such that $|\kappa_1| \leq |\kappa_2|$. By definition, the vectors $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{n}$ form an orthonormal bases and thus fit our proposed diffusion model. We will also set the scalar functions such that $g(\cdot) = \tau h(\cdot)$ and $w(\cdot) = \eta h(\cdot)$ where $\tau, \eta \in [0, 1]$. With this setup, Equation (6) can be re-written:

$$\frac{\partial f}{\partial t} = h(\cdot)\left(f_{\mathbf{r}_1\mathbf{r}_1} + \tau f_{\mathbf{r}_2\mathbf{r}_2} + \eta f_{\mathbf{nn}}\right) \tag{7}$$

Without referring to the exact argument of the scalar function $h(\cdot)$ yet, Equation (7) models an anisotropic diffusion which can be intuitively thought of as the summation of the local convolutions of three different 1D Gaussian kernels oriented along the three associated vector fields (compare each term of the diffusion with Equation (5)). The amount of diffusion along the maximum curvature direction $\mathbf{r}_2$, and the normal direction $\mathbf{n}$ are modulated by τ and η respectively, while diffusion is always performed along the minimum curvature direction $\mathbf{r}_1$ and finally the overall diffusion is modulated by the scalar function $h(\cdot)$, which we will call the *stopping function*.

We will set $\eta = 0$ for the rest of the paper to achieve objective **O-1**. To fulfill objectives **O-3** and **O-4** together we will replace the notation τ with τ_ρ which is defined by the following:

$$\tau_\rho = \begin{cases} \left(\frac{\kappa_{1,\rho}}{\kappa_{2,\rho}}\right)^{2\lambda} & \text{where } |\kappa_{2,\rho}| > 0, \lambda \in \mathbb{Z} \\ 1 & \kappa_{2,\rho} = 0 \end{cases} \tag{8}$$

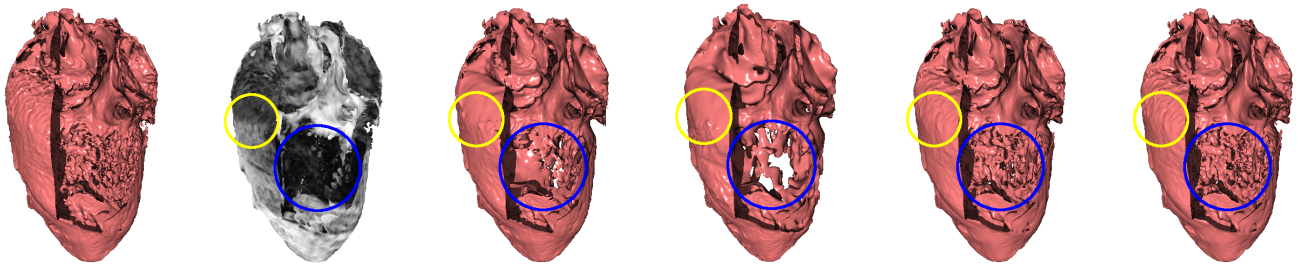
where the quantities, $\kappa_{1,\rho}$ (the minimum curvature) and $\kappa_{2,\rho}$ (the maximum curvature), are computed from a smoothed version of the data f_ρ with a Gaussian filter having a small variance ρ^2. The technique of taking measurements from a smoothed volume f_ρ is common in many PDE based methods specially under noisy conditions and we will discuss the effect of having a small ρ in section Section 6. We will use the notation $\rho = 0$ to imply that f was used to compute the curvatures instead of the smoothed version f_ρ. With the above definition, τ_ρ drops quickly to values very close to 0 whenever the maximum curvature $|\kappa_{2,\rho}|$ is higher than the minimum curvature $|\kappa_{1,\rho}|$, i.e. the surface is not isotropic. In this case, diffusion is performed primarily along the minimum curvature direction $\mathbf{r}_1$. For an isotropic surface where $|\kappa_{1,\rho}| = |\kappa_{2,\rho}|$ and a degenerate case, where $|\kappa_{2,\rho}| = 0$ (note that $|\kappa_{1,\rho}| \leq |\kappa_{2,\rho}|$), τ_ρ gets assigned to 1 which amounts to performing simple isotropic diffusion on the tangent plane of $\mathbf{n}$. The exponent of Equation (8) is always an even integer which makes sure that we are comparing only the absolute values of the curvatures.

Objective **O-2** can be addressed by computing the second derivative along the gradient direction, $f_{\mathbf{nn}}$ and *stop* diffusion whenever $f_{\mathbf{nn}} \approx 0$. To model this we can define the function $h(\cdot)$ such that it is continuous and approaches 0 for an argument close to 0 and 1 otherwise. For this we can simply modify the functions proposed by PM [26] as follows:

$$h(\alpha) = 1 - e^{-\ln\left(\frac{10}{9}\right)\left(\frac{\alpha}{\sigma}\right)^2} = 1 - (0.9)^{\left(\frac{\alpha}{\sigma}\right)^2}, \qquad \sigma \in \mathbb{R} \tag{9}$$

The scaling factor of $\ln(10/9)$ (Equation (9)) is there so that we have $h(\alpha) < 0.1$, which we considered to be very little diffusion, whenever $|\alpha| < \sigma$. This allows a more intuitive relationship between the argument α and the parameter σ. However, for a different purpose, this scaling factor could be changed or just simply be omitted. Using $f_{\mathbf{nn}}$ as the input to $h(\cdot)$ we essentially fulfill all four objectives we had set for ourselves. We finally present our anisotropic diffusion PDE by the following equation:

$$\frac{\partial f}{\partial t} = h(f_{\mathbf{nn}})(f_{\mathbf{r}_1\mathbf{r}_1} + \tau_\rho f_{\mathbf{r}_2\mathbf{r}_2} + \eta f_{\mathbf{nn}}) \tag{10}$$



(a) Original (b) $\|\nabla f\|$ map (c) KM method: $k = 40$ (d) KM method: $k = 80$ (e) Our method: $\sigma = 1$ (f) Our method: $\sigma = 10$

Fig. 4: Iso-surface (iso-value=153) rendering of the Sheep's Heart dataset. A section of the front part of the iso-surface has been culled to make the inner details visible. Except for (b) all the other images were rendered with shadows to clearly show the spatial arrangement of the iso-surface. The blue circle shows a region of low gradient magnitude, while the yellow circle shows a region where medium and high gradient magnitude meet. All diffusions were performed with 35 iterations and with the same time stepping. (a) Original dataset without any diffusion. (b) Gradient magnitude on the iso-surface coded in gray-scale; magnitude greater or equal to 130 is mapped to white while 0 is mapped to black. (c) Diffused with the KM method with $k = 40$. (d) Diffused with the KM method with $k = 80$. (e) Diffused with our method with $\sigma = 1$. (f) Diffused with our method with $\sigma = 10$.

4 Implementation

Krissian et al. [20] have shown that we could skip computing the principal curvature directions, which usually involves expensive eigenvalue decomposition of some matrix [15, 35], altogether and compute the quantities $f_{\mathbf{r}_1\mathbf{r}_1}$ and $f_{\mathbf{r}_2\mathbf{r}_2}$ directly using the following relationships:

$$f_{\mathbf{r}_1\mathbf{r}_1} = -\|\nabla f\|\kappa_1, \qquad f_{\mathbf{r}_2\mathbf{r}_2} = -\|\nabla f\|\kappa_2 \tag{11}$$

In the above, κ_1 and κ_2 can be computed in a straightforward fashion by the following

$$\kappa_i = K \pm \sqrt{K^2 - G}, \quad i \in \{1,2\} : |\kappa_1| \leq |\kappa_2| \tag{12}$$

where G and K are the Gaussian and Mean curvatures respectively. These can be computed directly from the first and second derivatives as given by Goldman [12]:

$$\begin{aligned} G &= \frac{1}{\|\nabla f\|^4} \nabla f^{\mathsf{T}} \mathbf{H}_c \nabla f \\ K &= \frac{1}{2\|\nabla f\|^3} \left(\nabla f^{\mathsf{T}} \mathbf{H} \nabla f - \|\nabla f\|^2 \operatorname{trace}(\mathbf{H}) \right) \end{aligned}$$

where $\mathbf{H}_c$ (see Section A of the Appendix in the supplementary material for details) is the co-factor matrix of the Hessian $\mathbf{H}$. Computation of both G and K can be implemented very efficiently without performing the actual matrix multiplications by expanding the equations as summations first (see Section B of the Appendix in the supplementary material).

Taking the relations given by (11), Equation (10) can be written more compactly and in matrix form as the following:

$$\frac{\partial f}{\partial t} = -h(\mathbf{n}^{\mathsf{T}}\mathbf{H}\mathbf{n}) \left(\|\nabla f\|(\kappa_1 + \tau_\rho \kappa_2) - \eta \mathbf{n}^{\mathsf{T}}\mathbf{H}\mathbf{n} \right) \tag{13}$$

Note that κ_1 and κ_2 in the above equation are measured from f whereas τ_ρ is measured from f_ρ where ρ is usually 1 when diffusion is used for the purpose of de-noising (see Section 6) and 0 otherwise.

5 Results and Discussion

In this section we will first demonstrate the shortcomings of the classical anistropic diffusion, proposed by Krissian et al. [20] - referred to as KM model for brevity - which in turn was based on the gradient magnitude, i.e. PM model [26] and compare the results with our novel approach. Next we will show the impact of the parameter σ, in Equation (9), of our method on the final output. We will then follow the discussion showing diffusion results using higher order derivative filters and finally provide empirical analysis of stability and convergence of our PDE.

5.1 Parameter Settings

For all the diffusion experiments performed in this section we have set $\lambda = 2$ in Equation (8). Voxel spacing was assumed to be 1 in all directions and scalar values, which ranged between $[0, 255]$, in the 3D volumes were not scaled. For the time step we have chosen $\Delta t = 0.05$, and $\eta = 0$ (no diffusion along the gradient direction) for the rest of the paper and we have used f without smoothing, i.e. $\rho = 0$. For all derivative estimations we have used the standard second-order stencils unless specified otherwise.

5.2 The Impact of the Stopping Function

In this section we will use the Sheep's Heart dataset [29] and demonstrate the sensitivity of the KM method to the parameter chosen. Likewise, we will show the robustness of our novel method with respect to its parameter and yet give a compelling example of its importance. In our experiment, we chose Equation (2) and Equation (9) as the stopping functions in the KM method and $h(\alpha)$ in our new proposed method, respectively, and used 35 iterations for both diffusion models.

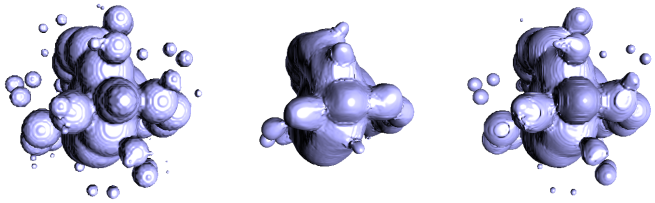
Figure 4 demonstrates the sensitivity of the KM method to its parameter k. The yellow circle marks an area where regions of small gradient magnitude merge with regions of higher gradient magnitude (see Figure 4b). Figure 4c, which was diffused with the KM model at $k = 40$ shows the selective nature of this method. Regions with a low gradient magnitude diffused much more compared to regions with higher gradient magnitude. On the other hand, regions having very low gradient magnitude (the blue circle) got diffused the most. As the value of k was increased to 80 (Figure 4d) a dramatically different output was produced. Here, regions inside the blue circle got diffused to the point of loosing structure while those with higher gradient magnitude (the lower part of the yellow circle) just started to get diffused. We also refer readers to Figure 1 (a-c) to see this selective nature of the KM method based on gradient magnitude and the resulting artifacts after 100 iterations were performed. A single iso-surface rendering may not tell the full story sometimes and for this reason we provide 2D slices of the same experiment (as performed for Figure 4) in Section D of the Appendix in the supplementary material.

On the other hand, our method has been found to be more robust with respect to its parameter σ. Unlike the KM model, smoothing of an iso-surface in our approach is performed consistently without any discrimination based on the gradient magnitudes. Since the parameter σ in our model is tied to the directional second derivative, $f_{\mathbf{nn}}$, it has a more intuitive impact on the overall diffusion process; i.e. decreasing σ amounts to smoothing a larger range of $f_{\mathbf{nn}}$. This can be thought of removing high frequencies from the 3D volume except near the boundaries between relatively homogeneous regions, where $f_{\mathbf{nn}}$ approaches zero and the stopping function $h(f_{\mathbf{nn}})$ approaches zero too making the diffusion stop (see Equation (13)). The sensitivity of parameters in both the KM and our proposed model is further illustrated in two separate animations we provide as supplementary materials (`KM_K_effect.avi` and `sigma_effect.avi` respectively). For both animations, we only changed the parameters k and σ for the KM and our model respectively and ran 35 iterations of diffusion while

all the other parameters were kept the same.

5.3 Significance of σ

In our previous examples, we have demonstrated the robustness of our new diffusion model with regards to the parameter σ. However, this robustness is observed for points away from zero. In this section, we argue with an appropriate example that the role of σ around zero is critical.



(a) Original (b) $h(\cdot) = 1$ (c) $\sigma = 40$

Fig. 5: Iso-surface rendering of a sampled phantom data with spheres. The value inside the spherical regions were 255 and 0 elsewhere. The volumes in (b) and (c) were diffused with 300 iterations. (a) The original volume (b) Diffused with $h(\cdot) = 1$ (c) Diffused with Equation (9) set as $h(\cdot)$ with $\sigma = 40$.

A quick look at Equation (13) immediately reveals that having the stopping function $h(\cdot)$ always evaluate to 1 with $\eta = 0$ makes the diffusion similar to the well known *mean curvature motion* [23] (assuming $\tau \approx 1$, i.e. on isotropic surfaces, like a sphere) except for the extra modulating term of $\|\nabla f\|$. However, on the surface of a sphere, for example, $\|\nabla f\|$ is constant and simplifying $h(\cdot)$ to 1 essentially makes Equation (13) a *mean curvature motion* diffusion in such a case. *Mean curvature motion* is a well studied diffusion scheme where spherical structures shrink until they disappear, which may not be desirable when preserving structures in a 3D volume.

To verify this we created a phantom dataset with several spherical regions of different sizes and diffused it once by making $h(\cdot)$ a constant 1 and the other by using Equation (9) with $\sigma = 40$. Figure 5 aptly demonstrates the significance of the parameter σ. When the phantom data was diffused with $\sigma = 40$ (Figure 5c) the basic structure of the original data, Figure 5a, was retained except for the very small spheres. On the other hand when the stopping function $h(\cdot)$ was set to a constant 1, Figure 5b, the overall diffusion converged to a simple *mean curvature motion* and the structure was destroyed. Our supplemental material includes animations that show the full evolution of the diffusions as given in Figure 5b (`sigma_unity.avi`), and 5c (`sigma_40.avi`).

5.4 Impact of Derivative Estimation Filters

To implement Equation (13) we need to compute the principal curvatures which require second order derivatives for the Hessian $\mathbf{H}$ in addition to the gradient components. Since we are computing all these quantities from sampled data the quality of the derivative estimation filters plays an important role.

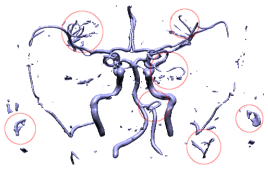
We used the Taylor Series based framework proposed by Hossain et al. [13] to construct discrete derivative estimation filters of error order 2-EF and 4-EF [2]. Usually for real data where the polynomial order of the underlying function is not known apriori, a 4-EF filter has been found to yield more accurate results than 2-EF.

For our experiment, we used an Angiography dataset [5] in which the blood vessel were the focus of the study. We used the same diffusion parameters ($\sigma = 1$) and only varied the first and second order derivative filters. Figure 6 demonstrates that the higher order filter (4-EF) preserves more details in the blood vessels while still removing some of the spurious elements.

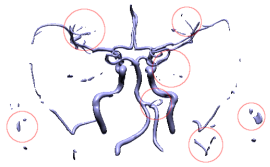
5.5 Stability and Convergence

We performed empirical tests of stability and convergence whose details can be found in Section C of the Appendix. We learned that with $\rho = 1$ (in voxel units) and all the other parameters kept the same as

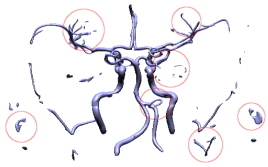
[2] A filter is called *n-EF*, where EF stands for *Error Filter*, if it estimates a given derivative with error bounded by $O(h^n)$, where h is the grid spacing.



(a) Original



(b) 2-EF



(c) 4-EF

Fig. 6: Direct volume rendering of an angiography dataset of a human head. The red circles mark the regions where the images had noticeable differences. (a) The original dataset. (b) Derivatives estimated using 2-EF filters. (c) Derivatives estimated using 4-EF filters.

Section 5.1, including $\sigma = 1$, the time stepping of $\Delta t \leq 0.4$ is stable for most practical purposes. The choice of $\rho = 1$, which is common in many diffusion methods, was made only to have a better smoothing behavior in a relatively homogeneous region under noise, and this has little effect on the stability. This parameter ρ is useful only for the purpose of de-noising and the value of 1 yields best results for most noise types as we will demonstrate in the next section. Therefore for the rest of the paper we keep all the parameters the same as Section 5.1 except $\rho = 1$ and $\Delta t = 0.40$.

6 De-noising Properties of Our Proposed Model

Although our proposed PDE model is a general purpose smoothing technique to evolve a 3D volume over time removing small high frequency details while preserving edges it exhibits de-noising properties too. On the other hand, general purpose de-noising techniques have been formulated previously, e.g. simple *bi-lateral filtering*, which has been shown to be a variant of gradient magnitude based *non-linear diffusion* by Barash and Comaniciu [4]. Because of the drawbacks of the gradient magnitude based models, the equivalent de-noising operators inherit similar problems. Further, anistropic diffusion in 3D based on the KM model is very sensitive to the parameter choice. Therefore, none of these existing techniques would be well suited for de-noising in 3D without facing difficulties.

In this subsection we will discuss, both qualitatively and quantitatively, the de-noising properties our proposed PDE model has in the context of the four common noise types: additive *Gaussian* noise, additive *Poisson* noise, multiplicative *Speckle* noise and *Salt and Pepper* noise. We chose the Tooth dataset [29] which is relatively noise-free and diffused it using our proposed method after adding a particular type of noise. For all the experiments we have set the parameters as described in Section 5.5.

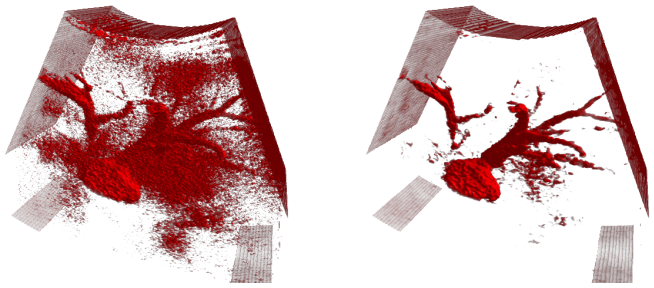
Figure 11 summarizes the de-noising properties of our proposed anistropic diffusion model and it shows that our proposed method could de-noise the tooth data quite well in all four cases. Our method works best with *Salt and Pepper* noise which is of no surprise because *Salt and Pepper* noise introduces random and very local blob type artifacts in the volume that get removed immediately and remarkably well. For *Gaussian* noise and *Poisson* noise, our method performed similarly well on both occasions. *Speckle* noise turned out to be the hardest to tackle of all, which is not surprising, and yet our method performed well achieving an SNR of over 24.5 dB (see Figure 11). We also provide animation sequences to show the de-noising process for each noise type listed in Figure 11 as supplementary materials: `gaussian.avi`, `poisson.avi`, `speckle.avi`, and `salt-pepper.avi`.

In Figure 7 we present a qualitative result of our diffusion model applied to a 3D Ultrasound data of human liver [27] with a size of $247 \times 208 \times 86$, which has been sub-sampled from the original data by a factor of two in each dimension. Ultrasound data are usually

Table 1: Performance evaluation of different diffusion methods: our method, SRNRAD and ORNRAD, on different types of naturally occurring noise. The best performing result for each metric is highlighted with a **bold number** while the second best is underlined. For the Rician noise we used a standard deviation of 20 similar to [19]. Note that there can be ties.

	Gaussian			Rician			Poisson			Salt & Pepper			Speckle		
	MSE	SSIM	QILV	MSE	SSIM	QILV	MSE	SSIM	QILV	MSE	SSIM	QILV	MSE	SSIM	QILV
Noise	558.750	0.540	0.503	450.558	0.561	0.712	482.905	0.678	0.492	2202.452	0.476	0.022	493.013	0.714	0.455
SRNRAD	162.017	0.816	**0.886**	232.459	0.792	**0.913**	91.766	0.917	**0.914**	1262.760	0.625	0.033	84.301	**0.929**	**0.906**
ORNRAD	153.873	0.819	0.859	226.405	**0.795**	0.889	78.731	0.924	0.899	1114.028	0.642	0.101	**78.308**	**0.929**	0.863
Our	**75.878**	**0.900**	0.860	**173.851**	**0.795**	0.820	**70.671**	**0.930**	0.857	**33.210**	**0.968**	**0.924**	85.129	0.922	0.838

contaminated with *speckle* noise and it is noteworthy how the noise was lessened keeping all the vital structures intact even for a relatively low resolution volume. Note how the tubular structures, which were barely discernible in Figure 7a, stick out clearly in Figure 7b. When we attempted to apply the KM model on this same dataset we faced real challenges to pick a right value for the parameter k as we had no prior knowledge about the gradient magnitude distribution around the tubular structures and this itself speaks in favor of our method where we could pick within a wide range of values for σ and still got some decent and consistent results.



(a) Original (b) Diffused

Fig. 7: A 3D Ultrasound data diffused with our method using $\sigma = 1$ and 5 iterations. (a) The original volume (b) The diffused volume using the same transfer function.

6.1 Comparison with other de-noising methods

In this section we will compare our method with two very recent PDE based de-noising filters, namely *Scalar Rician Noise-Reducing Anisotropic Diffusion* SRNRAD and *Oriented Rician Noise-Reducing Anisotropic Diffusion* ORNRAD, proposed by Krissian and Aja-Fernández [19]. These filters were designed to de-noise specifically Rician noise in 3D MRI data. At a higher SNR, Rician distribution converges to Gaussian distribution. According to Aja-Fernández et al. [3], the noise estimator used by [19] is based on the variance estimation of additive Gaussian noise. This is, however, not an unfair mismatch because the Rician noise converges to additive Gaussian noise after few iterations. Therefore, we will also apply their filters on additive Gaussian noise. It is worth mentioning that both SRNRAD and ORNRAD require users to manually specify a sub-volume for the noise estimator. Further, to demonstrate the versatility of our proposed method, we will apply all three methods on other types of noise and show that our method can be applied equally well in most types of naturally occurring noise types using exactly the same set of parameters values.

For comparison we have chosen the same dataset as [19]: the simulated structural MR data [9], and used the same quantitative metrics - namely *Mean Squared Error* MSE, *Structural Similarity Index* SSIM [36], and *Quality Index Based on Local Variance* QILV [2] - for the quality assessment. In accordance to [19], we also discarded the background, where the original noise-free volume is zero, from any assessments. It is noteworthy to mention that we have used the exact same Matlab script to compute these metrics as the authors of [19] and for SRNRAD and ORNRAD we have used their own C/C++ implementations in the AMILab software [17].

Table 1 summarizes the performance of the three diffusion techniques and we provide the corresponding images in Figure 12. In terms of MSE, our method performed significantly better than both ORNRAD and SRNRAD for all noise types except *Speckle* and *Poisson*, where the differences are close for the MSE metric. For the SSIM metric, we find that our method again performed better than both ORNRAD and SRNRAD for all noise types except *Speckle* where the numerical difference is only in the third decimal place. It is rather intriguing to find that our method tied with ORNRAD and actually performed better than SRNRAD for the SSIM metric in the case of *Rician* noise, for which those two filters were specifically designed for. Using the QILV metric, our method performed only marginally different from the ORNRAD for all the noise types except *Salt and Pepper* noise for which case our method clearly outperformed both ORNRAD and SRNRAD by a large margin for all the three metrics. However, we acknowledge that ORNRAD and SRNRAD were designed for *Rician* noise (and should also work well for *Gaussian*) but this experiment reveals that our proposed method, unlike many de-noising methods, can be applied generally for all common noise types and still produce decent results if not better in some cases and that too without re-tuning parameters.

Figure 12 corroborates our numerical results although we see the results yielded by ORNRAD and SRNRAD are visually smoother in homogeneous regions. On the other hand we argue that the fine fiber like structures as seen in the bottom left region of the original volume were preserved better by our method.

Our method was implemented in Matlab where the values of $\tau_\rho, \kappa_1, \kappa_2$ and $\mathbf{n}^\mathsf{T}\mathbf{H}\mathbf{n}$ in Equation (13) were computed using MEX files (C/C++ extension for Matlab) using only a single thread. Other computations including all first and second derivative estimations and convolutions were performed using Matlab scripts. On the other hand SRNRAD and ORNRAD were implemented as multi-threaded C/C++ codes in AMILab as mentioned in [19]. Table 2 reports average time (in seconds) each method took per iteration when we ran them on an Intel Core$^{\text{TM}}$2 Duo (2.4 GHz on each core) based system using the MRI dataset. Table 2 shows that our method performs ≈ 14.4 times faster than ORNRAD despite our Matlab implementation. On the other hand, though SRNRAD runs faster, a clear winner is not yet obvious as most computations in our method were performed in Matlab.

Table 2: Average time (in seconds) for each method per iteration while diffusing the MRI dataset ($181 \times 217 \times 181$).

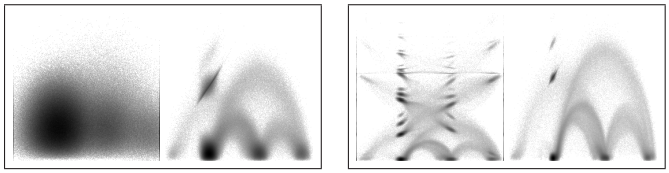
	Our	SRNRAD	ORNRAD
Time/Iteration (seconds)	7.52	3.75	108.48

The proposed method is not only superior in run-time performance, but is superior or comparable in both qualitative (Figure 12) and quantitative (Table 1) measures. In addition, we would like to re-emphasize the ease of parameter choice in our method as we achieved all of these performances using exactly the same parameter settings without re-tuning.

7 Impact on Visualization: 2D Transfer Function

Kindlmann and Durkin [14] proposed a 2D transfer function which proved to be very powerful at classifying homogeneous regions and

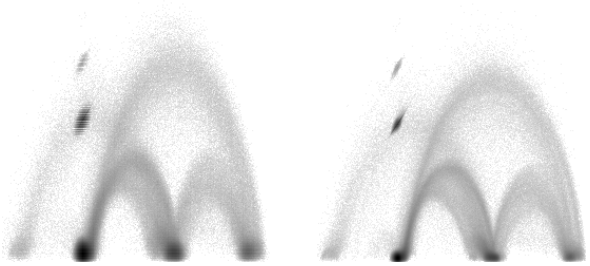
boundaries in a 3D volume. In their method a 2D histogram would be generated where the horizontal axis would be the function value and the vertical axis would be the directional first derivative along the gradient f_n, a quantity that happens to be just the gradient magnitude, i.e. $f_v = \|\nabla f\|$. For a clean 3D scalar data this histogram will have arc like patterns for every unique boundary in the volume. Because of its importance in visualization, we show the utility of our novel diffusion method in the context of this 2D transfer function. The counts in the histograms are compressed with the function $log_{10}(x+1)$ - where x is the actual count - before plotting them as pixels. Darker pixels denote higher count. For brevity, we will omit the axes labels from the 2D histogram images in this section.



(a) Gaussian (b) Salt and Pepper

Fig. 8: 2D histogram of the tooth dataset with two different noise types. Each pair of images correponds to a noise type as indicated by the caption. The histogram on the left column of each pair was computed after adding the noise and that of the right column after performing diffusion.

In Figure 8 we present the 2D histogram before and after diffusion for two different noise types while a reference histogram of the original dataset is provided in Figure 9a. It is remarkable to see how each noise type changes the 2D histogram so dramatically while our diffusion method brings the original pattern back to a recognizable form. A closer inspection of Figure 8b reveals that our diffusion not only removes the *Salt and Pepper* noise but also enhances the pattern in the 2D histogram. This indicates that running our diffusion on a clean dataset for the purpose of smoothing will also enhance its 2D histogram. To verify this, we diffused the tooth data without adding any noise and computed the 2D histogram in Figure 9 which shows that the patterns in the histogram are indeed enhanced. This makes sense because our diffusion was modeled in a way such that edges and tubular structures are well preserved while small, high frequency details are smoothed out. Figure 10 immediately demonstrates an even greater problem with the gradient magnitude based KM model where the patterns in the 2D histogram get virtually destroyed whereas the patterns get enhanced with our method instead.



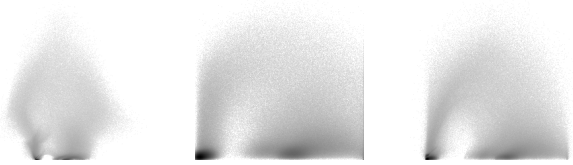
(a) Original (b) Diffused

Fig. 9: Tooth data diffused without adding any noise. (a) 2D histogram of the original tooth (b) Histogram of the tooth data after diffusion.

8 Conclusion

Often times, the salient features in 3D data are not easily detectable by volume visualization methods, especially those based on simple transfer function designs. Therefore there is often a need to preprocess data in a reliable manner so that the salient features are preserved. Such pre-processing methods are often based on a smoothing or anisotropic diffusion framework, which requires laborious parameter tuning.

In this paper we have presented a novel anisotropic diffusion model for 3D scalar data to address these issues. We have used an intuitive definition for edges based on the directional second derivative along the gradient. This led us to the design of a PDE with a *stopping function* that is much less sensitive to its parameter σ and smoothes data while preserving edges. We have shown that with this new *stopping function* diffusion is performed consistently on an iso-surface regardless of the gradient magnitude, which is in contrast to previous methods, like KM. Even more so, our proposed diffusion model remains second order and much simpler to implement unlike higher order and existing de-noising PDEs.



(a) KM model (b) Original (c) Our method

Fig. 10: 2D histogram of the Sheep's heart dataset corresponding to Figure 1 (a-c).

On the other hand, our results demonstrate some remarkable de-noising properties of the proposed diffusion model. On this end, we have compared our results with a recent PDE based de-noising technique on five different noise types.

With such consistent edge preserving smoothing and denoising properties our diffusion model has great utility in the context of visualization. We demonstrated the effect of our diffusion model on the quality of the rendered images using a variety of datasets, including both synthetic and real. Further, we showed its impact on multi-dimensional histograms, which are the basis of many volume rendering algorithms. Specifically, we are able to recover the arc patterns in the histogram even in the presence of strong noise. In the noiseless case, our diffusion has been found to still enhance these histograms without re-tuning a parameter for each new dataset. This will eventually let the practitioner use our diffusion model as a general purpose smoothing and de-noising tool without destroying salient features or tuning a parameter.

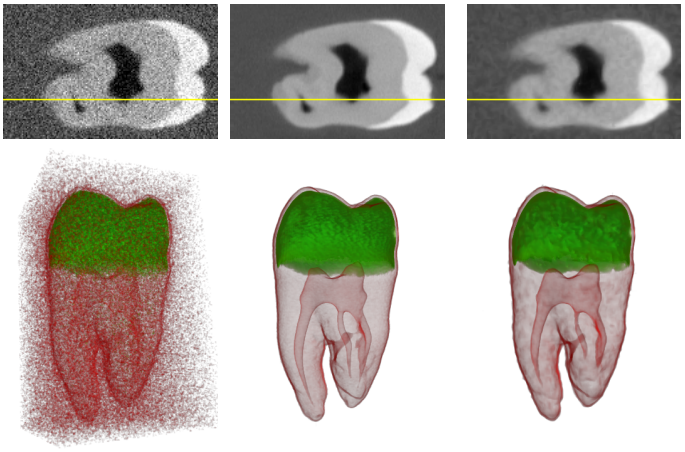
9 Future Work

Convergence, stability requirements, and continuity of the underlying function are not well understood in our method from a theoretical standpoint, which will be our next goal. Apart from the theoretical treatments, our method can also be studied more rigorously in the domain of de-noising. In addition, a comparison to *non-local means* [1,6] has been planned as a future study.

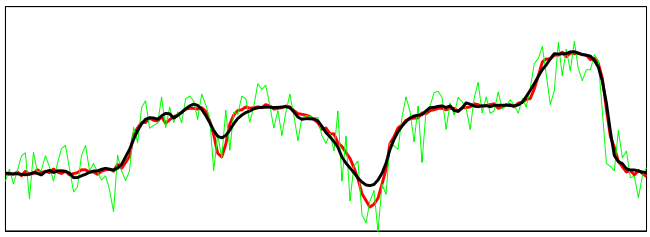
Another interesting area of research would be to extend our method for point based models and geometric meshes. For example, the classical PM model has been used within a level set framework [32], photon mapping [30], and we speculate similar applications of our method in such areas.

Acknowledgments

We would like to thank Dr. Karl Krissian and Dr. Santiago Aja-Fernández for providing Matlab codes to compute quality metrics on 3D volumes and also for the general guideline on using the AMILab software [17]. We would also like to thank Dr. Philippe Thévenaz for providing codes for generating the Sphere phantom data. This work has been funded in part by the Natural Science and Engineering Research Council of Canada (NSERC).

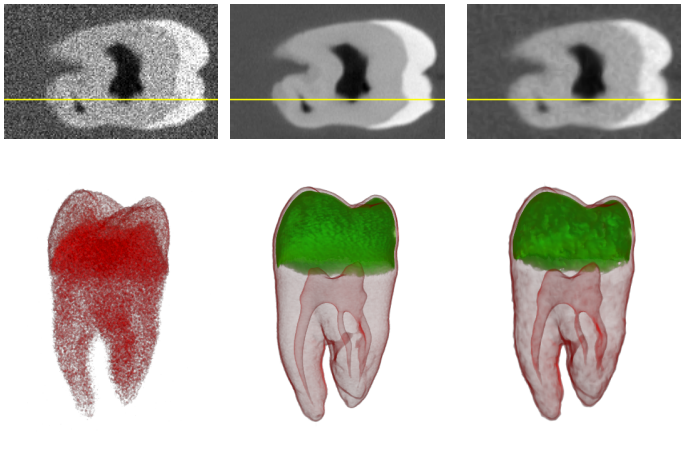


Noisy, SNR=12.89 Original Diffused, SNR=26.12

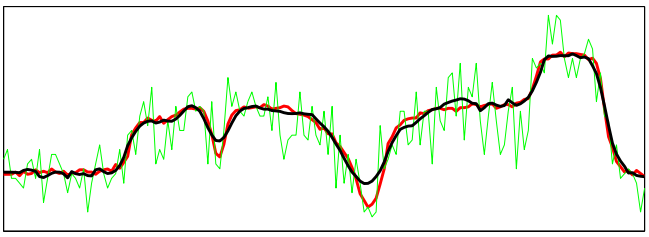


Profile

(a) Gaussian

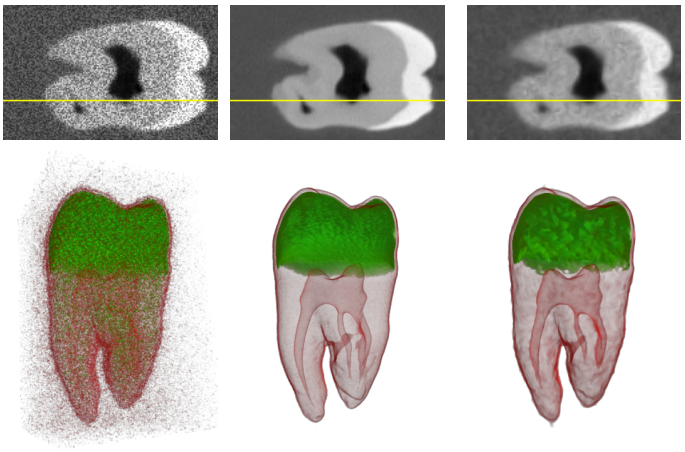


Noisy, SNR=12.80 Original Diffused, SNR=25.76

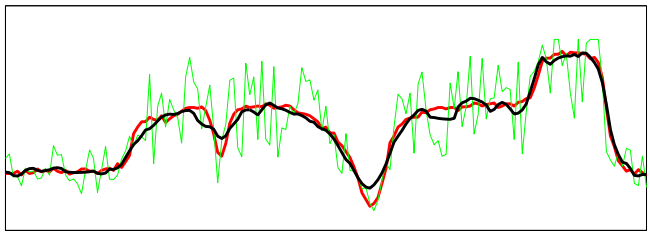


Profile

(b) Poisson

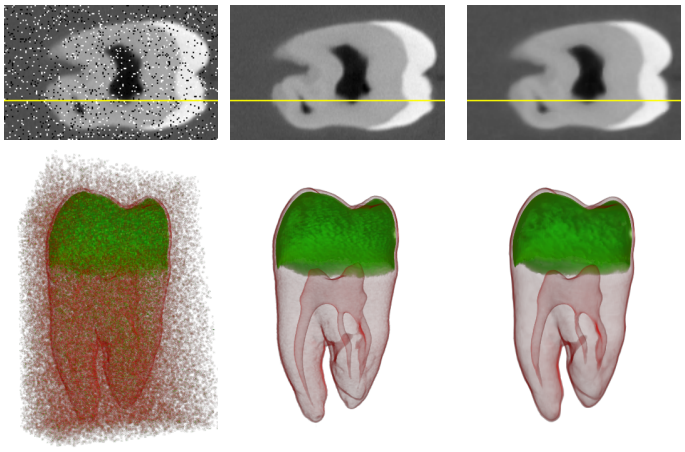


Noisy, SNR=12.99 Original Diffused, SNR=24.68

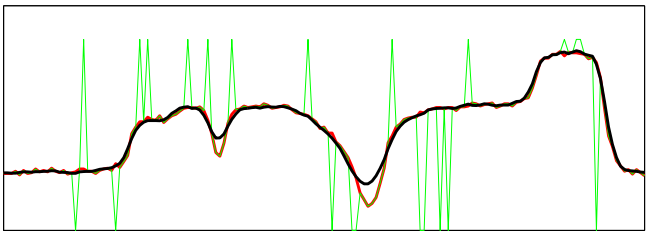


Profile

(c) Speckle



Noisy, SNR=8.17 Original Diffused, SNR=30.06



Profile

(d) Salt and Pepper

Fig. 11: Each of the Figures (11a, 11b, 11c, 11d), was derived from an experiment performed with one particular type of noise, as indicated by the corresponding captions. All the diffusions were performed with identical settings and for 25 iterations. Identical transfer functions were used to render all the 3D images above. The profile plot in each figure is a 1-D plot of the scalar values taken from a slice as indicated by the yellow line in the 2D images. The red curve is a plot of the original scalar values, the green curve is that of the noisy scalar values while the black curve is the plot of the de-noised scalar values. Note how the black curve approximates the red curve for different noise types. SNR (in dB) of the noisy and the diffused data are also provided for each experiment.

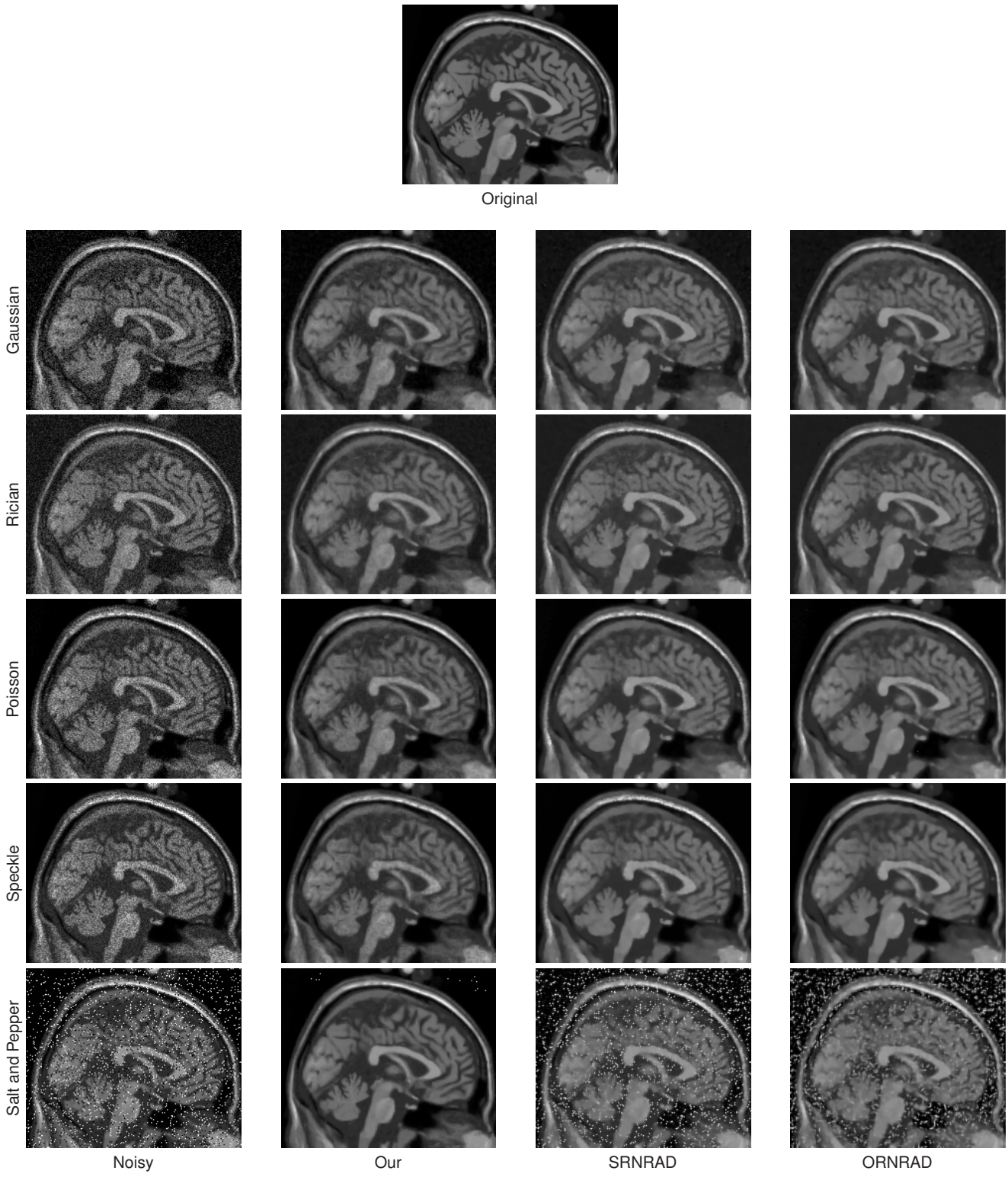


Fig. 12: Comparison of $x = 90$ slice of the synthetic MRI data for each type of noise after diffusing with three different methods. The image in the first row is taken from the original volume. After that, each row, going from top to bottom, corresponds to a particular noise type: *Gaussian*, *Rician*, *Poisson*, *Speckle* and *Salt and Pepper* respectively. Images in the first column are taken from the noisy volumes, while those in the second, third and fourth columns are taken from the volumes diffused with our, SRNRAD, and ORNRAD filters respectively.

REFERENCES

[1] A. Adams, J. Baek, and M. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. John Wiley & Sons, 2010.

[2] S. Aja-Fernandez, R. San-José-Estépar, C. Alberola-Lopez, and C.-F. Westin. Image quality assessment based on local variance. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 4815 –4818, Aug 2006.

[3] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López. Automatic noise estimation in images using local statistics. Additive and multiplicative cases. *Image and Vision Computing*, 27(6):756–770, 2009.

[4] D. Barash and D. Comaniciu. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Vision Computing*, 22(1):73–81, 2004.

[5] D. Bartz. Volvis, [online]. `http://www.volvis.org`. Last accessed: 21 July 2010.

[6] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[7] R. Carmona and S. Zhong. Adaptive smoothing respecting feature directions. *IEEE Transactions on Image Processing*, 7(3):353–358, 1998.

[8] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *15th IEEE Visualization 2004 Conference (VIS 2004), 10-15 October 2004, Austin, TX, USA*, pages 497–504. IEEE Computer Society, 2004.

[9] D. Collins, A. Zijdenbos, V. Kollokian, J. Sled, N. Kabani, C. Holmes, and A. Evans. Design and construction of a realistic digital brain phantom. *IEEE Transactions on Medical Imaging*, 17(3):463, 1998.

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002.

[11] G. Gerig, O. Kubler, R. Kikinis, and F. Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11(2):221–232, 1992.

[12] R. Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005. Geometric Modelling and Differential Geometry.

[13] Z. Hossain, U. R. Alim, and T. Möller. Toward high-quality gradient estimation on regular lattices. *IEEE Transactions on Visualization and Computer Graphics*, 99(RapidPosts), 2010.

[14] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *VVS '98: Proceedings of the 1998 IEEE Symposium on Volume Visualization*, pages 79–86, New York, NY, USA, 1998. ACM.

[15] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 513–520. IEEE Computer Society, 2003.

[16] G. Korn and T. Korn. *Mathematical Handbook for Scientists and Engineers*. McGraw-Hill, New York, 1968.

[17] K. Krissian. AMILab, [online]. `http://amilab.sourceforge.net`. Version: 2.0.4, Last accessed: 21 July 2010.

[18] K. Krissian. Flux-based anisotropic diffusion applied to enhancement of 3D angiogram. *IEEE Transactions on Medical Imaging*, 21(11):1440–1442, 2002.

[19] K. Krissian and S. Aja-Fernández. Noise-driven anisotropic diffusion filtering of MRI. *IEEE Transaction on Image Processing.*, 18(10):2265–2274, 2009.

[20] K. Krissian, G. Malandain, and N. Ayache. Directional anisotropic diffusion applied to segmentation of vessels in 3D images. In *SCALE-SPACE '97: Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, pages 345–348, London, UK, 1997. Springer-Verlag.

[21] K. Krissian, C. Westin, R. Kikinis, and K. Vosburgh. Oriented speckle reducing anisotropic diffusion. *IEEE Transactions on Image Processing*, 16(5):1412–1424, 2007.

[22] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London Series B*, 207:187–217, 1980.

[23] B. Merriman, J. Bence, and S. Osher. Diffusion generated motion by mean curvature. In *Computational Crystal Growers Workshop*, 1992.

[24] K. Mosaliganti, F. Janoos, A. Gelas, R. Noche, N. Obholzer, R. Machiraju, and S. Megason. Anisotropic plate diffusion filtering for detection of cell membranes in 3D microscopy images. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pages 588 –591, April 2010.

[25] O. Nemitz, M. Rumpf, T. Tasdizen, and R. Whitaker. Anisotropic curvature motion for structure enhancing smoothing of 3D MR angiography data. *Journal of Mathematical Imaging and Vision*, 27(3):217–229, 2007.

[26] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, 1990.

[27] R. Prager, G. Andrew, and G. Treece. Stradx, [online]. `http://mi.eng.cam.ac.uk/~rwp/stradx/`. Version: 7.4g, Last accessed: 21 July 2010.

[28] T. Preusser and M. Rumpf. A level set method for anisotropic geometric diffusion in 3D image processing. *SIAM Journal on Applied Mathematics*, 62(5):1772–1793, 2002.

[29] S. Roettger. The volume library, [online]. `http://www9.informatik.uni-erlangen.de/External/vollib/`. Last accessed: 21 July 2010.

[30] L. Schjøth, J. Sporring, and O. Olsen. Diffusion based photon mapping. In *Computer Graphics Forum*, volume 27, pages 2114–2127. John Wiley & Sons, 2008.

[31] Q. Sun, J. Hossack, J. Tang, and S. Acton. Speckle reducing anisotropic diffusion for 3D ultrasound images. *Computerized Medical Imaging and Graphics*, 28(8):461–470, 2004.

[32] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. *IEEE Conference on Visualization*, pages 125–132, 2002.

[33] S. Tenginakai, J. Lee, and R. Machiraju. Salient iso-surface detection with model-independent statistical signatures. In T. Ertl, K. I. Joy, and A. Varshney, editors, *IEEE Conference on Visualization*. IEEE Computer Society, 2001.

[34] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[35] L. Velho, L. H. de Figueiredo, and J. A. Gomes. *Implicit Objects in Computer Graphics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[36] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[37] J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G. Teubner, Stuttgart, Germany, 1998.

[38] J. Weickert. Coherence-enhancing diffusion filtering. *Int. J. Comput. Vision*, 31(2-3):111–127, 1999.

[39] R. Whitaker. Volumetric deformable models: Active blobs. In *Proceedings of SPIE*, pages 122–134, 1994.

[40] Y. Yu and S. Acton. Speckle reducing anisotropic diffusion. *IEEE Transactions on Image Processing*, 11(11):1260–1270, 2002.