

A Volume Filtering and Rendering System for an Improved
Visual Balance of Feature Preservation and Noise
Suppression in Medical Imaging

Paul Ross

Submitted for the degree of Doctor of Engineering (EngD)

Heriot-Watt University

EPS/MACS

October 2017

Abstract

Preserving or enhancing salient features whilst effectively suppressing noise-derived artifacts and extraneous detail have been two consistent yet competing objectives in volumetric medical image processing. Illustrative techniques (and methods inspired by them) can help to enhance and, if desired, isolate the depiction of specific regions of interest whilst retaining overall context. However, highlighting or enhancing specific features can have the undesirable side-effect of highlighting noise. Second-derivative based methods can be employed effectively in both the rendering and volume filtering stages of a visualisation pipeline to enhance the depiction of feature detail whilst minimising noise-based artifacts. We develop a new 3D anisotropic-diffusion PDE for an improved balance of feature-retention and noise reduction; furthermore, we present a feature-enhancing visualisation pipeline that can be applied to multiple modalities and has been shown to be particularly effective in the context of 3D ultrasound.

For my mum and dad

Acknowledgements

Special thanks to Tim Day and Dr. Robert Davey, formerly at Toshiba Medical Visualisation Systems Europe Ltd, and to Professor Mike Chantler, Dr. Alex Belyaev, Professor Derryck Reid and all at the EngD office at Heriot Watt University, Edinburgh.

ACADEMIC REGISTRY

Research Thesis Submission

Name:			
School:			
Version: <i>(i.e. First, Resubmission, Final)</i>		Degree Sought:	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

- 1) the thesis embodies the results of my own work and has been composed by myself
- 2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
- 3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
- 4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
- 5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
- 6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	
-------------------------	--	-------	--

Submission

Submitted By <i>(name in capitals):</i>	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by <i>(name in capitals):</i>		
Method of Submission <i>(Handed in to SSC; posted through internal/external mail):</i>		
E-thesis Submitted <i>(mandatory for final theses)</i>		
Signature:		Date:

Contents

Table of Symbols	xv
-------------------------	-----------

Table of Acronyms	xvii
--------------------------	-------------

1 Introduction	1
1.1 Background	2
1.2 Project aims and scope	4
1.2.1 Requirements	4
1.2.2 Scope	4
1.2.3 Implementation	5
1.3 Project contributions	6
1.4 Possible applications	8
2 Literature review	9
2.1 Introduction	9
2.2 Volume-rendering overview	10
2.2.1 Raycasting	10
2.2.2 Transfer Functions	11
2.3 Illumination models	12
2.4 Illustrative techniques	13
2.4.1 Lines for emphasis	14
2.4.2 Illustrative shading to simulate pencil and ink methods	14
2.5 Illustrative techniques: volumetric data	15
2.5.1 Contours	15
2.5.2 Illustrative shading	16
2.6 Style Transfer Functions	18
2.6.1 The Lit Sphere	18
2.6.2 Transparency	18
2.6.3 Possible improvements	18
2.7 Feature enhancement: more on contours, lines and shading based on surface differential properties	20
2.8 Hybrid rendering methods, context and applications	21

2.9	Noise reduction (filtering)	22
2.9.1	Neighbourhood filters	22
2.9.2	PDE-based filters	24
2.9.3	Other methods for feature-preserving smoothing	26
2.9.4	Application to 3D ultrasound	27
2.10	Conclusion	28
2.10.1	Base renderer	29
2.10.2	Supplementary shading for feature enhancement	29
2.10.3	Improved volumetric noise-reduction	30
2.10.4	Applications and datasets	30
2.10.5	Evaluation methods	31
2.10.6	Summary	32
3	A base illustrative rendering system	33
3.1	Style Transfer Functions	34
3.1.1	Spherical lighting maps	35
3.1.2	Transfer functions	36
3.2	Selective clipping	37
3.3	Other shading methods	38
3.3.1	Curvature-based transfer functions	38
3.3.2	Combined transfer function	39
3.4	Additional shading	39
3.5	Implementation	41
3.5.1	User interface	41
3.5.2	Renderer	42
3.6	Results and examples	43
3.6.1	Blinn-Phong and other procedural local lighting models	43
3.6.2	Stippling	43
3.6.3	Layering	46
3.6.4	Highlighting salient structures	47
3.7	Application to medical imaging	47
3.7.1	CT imaging	48
3.7.2	Ultrasound keepsake imaging	52
3.7.3	4D Ultrasound	55
3.8	Performance	57
3.9	Conclusion	57
3.9.1	Summary	58
4	Improving detail: additional curvature-based shading	60
4.1	Introduction	60
4.2	Motivation: traditional illustrative methods for enhanced shape depiction	61

4.3	Curvature-based shading methods	62
4.4	Curvature	63
4.4.1	Principal curvatures: an informal introduction	64
4.4.2	Ridges and Valleys	65
4.5	Curvature-based shading	67
4.5.1	Minimum principal curvature	68
4.5.2	Strict valleys	68
4.5.3	Maximum principal curvature	68
4.5.4	Strict ridges	69
4.5.5	Summary	69
4.6	Application to medical imaging	69
4.6.1	CT imaging	70
4.6.2	Ultrasound keepsake imaging	73
4.6.3	4D ultrasound	75
4.6.4	Limitations	77
4.7	Deferred curvature shading	77
4.7.1	Method	78
4.7.2	Results	78
4.7.3	Summary	81
4.8	Performance	81
4.9	Conclusion	82
5	Further reduction of noise: effective volumetric filtering	84
5.1	The Gaussian filter and inherent limitations	85
5.1.1	Limitations	86
5.2	Feature-preserving smoothing	87
5.2.1	Nonlinear and anisotropic diffusion	87
5.2.2	Nonlinear diffusion: extension to 3D	92
5.2.3	Bilateral filtering (neighbourhood filter)	95
5.3	Curvature-based PDE filters	102
5.3.1	Mean curvature motion	102
5.3.2	Weighted curvature motion	103
5.3.3	Other curvature flows	104
5.4	The Hossain-Möller anisotropic diffusion filter	105
5.4.1	Effect of stopping function	110
5.4.2	Effect of anisotropy function	112
5.4.3	Hossain-Möller Anisotropic Diffusion: Conclusion	115
5.5	Proposed solution: anisotropic diffusion with isotropy-independent selec- tive mean curvature motion.	116
5.5.1	Thresholding the anisotropy function	117

5.6	Summary of related methods and contribution	123
5.6.1	Final formulation	124
5.7	Application to medical imaging	125
5.7.1	CT imaging	125
5.7.2	Ultrasound keepsake imaging	126
5.8	Performance	126
5.9	Conclusion	128
6	Complete System	130
6.1	Pipeline	130
6.2	Applications	131
6.2.1	Ultrasound babyface (keepsake images)	131
6.2.2	4D ultrasound	132
6.2.3	CT imaging	133
6.3	Comparison with other methods	134
6.4	Conclusion	137
7	Conclusions	139
7.1	Overview of work	140
7.1.1	Objectives	140
7.1.2	Illustrative rendering	140
7.1.3	Curvature Shading	141
7.1.4	Deferred and smoothed curvature shading	141
7.1.5	Developing a new curvature-based smoothing method	142
7.1.6	Overall system and applications	143
7.1.7	Performance	143
7.2	Novel contributions	143
7.3	Future work	144
7.3.1	Applications	144
7.3.2	Rendering	144
7.3.3	Filtering	145
7.4	Concluding remarks	146
	Appendices	147
	Appendix A	148
A.1	Curvature: background	148
A.1.1	The Gauss Map	148
A.1.2	Shape operator	148
A.1.3	Gaussian and mean curvatures	150
A.2	Curvature computation: methods to determine principal curvature values.	152

A.2.1	Kindlmann method	152
A.2.2	Gaussian and Mean curvature method	153
A.3	Curvature computation: Cuda Kernels	154
A.3.1	Kindlmann method	154
A.3.2	Gaussian and mean curvature method	155
A.4	Modified Hossain-Möller smoothing filter	156
A.4.1	Core function	156
A.4.2	Standard forward-Euler	156
A.5	Some useful vector calculus expressions	157
A.5.1	Expansion of $\nabla(\ \nabla u\)$	157
A.5.2	Expansion of $\nabla\left(\frac{1}{\ \nabla u\ }\right)$	158
A.5.3	Expansion of $\operatorname{div}\left(\frac{\nabla u}{\ \nabla u\ }\right)$	158
A.5.4	Mean curvature	158

List of Tables

3.1	Per-frame rendering time at 512x512 output resolution: comparison of local-lighting models.	57
4.1	Per-frame rendering time at 512x512 output resolution: comparison of shading methods (STF: style transfer function, normal-based spheremap shading; CS: procedural curvature shading).	82
5.1	Per-iteration timings for different smoothing methods; note that the effective ‘amount’ of smoothing performed per iteration is not necessarily equivalent across methods.	127
5.2	Forward Euler timings (per iteration) for the new PDE for various combination of 2nd derivative sampling methods.	127
5.3	RK4 timings (per iteration) for the new PDE for various combination of 2nd derivative sampling methods.	127

List of Figures

2.1	Renderings of nut, showing the effect of contour and crease lines on overall shape-depiction. Clockwise from top left: no contour or crease lines; with the addition of contour lines; with the addition of contour and crease lines; contour and crease lines only. Source: Saito (1990) [1].	14
2.2	Hat and cane modelled with B-spline surfaces and shaded with stippling and hatching respectively. Source: [2].	14
2.3	Comparison of illustrative contour-rendering based on dot product of view direction and surface normal (\mathbf{v} and \mathbf{n}), unregulated (a) and regulated (b). Source: Kindlmann et al. (2003) [3].	15
2.4	Two styles of simplified illustrative shading.	16
2.5	Illustrative hatching and stipple shading of volumetric data.	17
2.6	Increased detail due to exaggerated shading in skull model. Source: Rusinkiewicz et al. (2006) [4].	17
2.7	Top: pencil illustration of beetle. Middle and Bottom: illustrative render via two different style-transfer functions. The bottom one has two lit-spheres associated with it. Source: Bruckner et al. (2007) [5].	19
2.8	Illustrative volume rendering of segmented head dataset via style transfer functions. Source: Bruckner et al. (2007) [5].	19
2.9	Feature-enhancing shading methods	20
3.1	Shading of volumetric models using lit spheres	35
3.2	Shading process using spheremaps and parallel translation of unit surface normals	36
3.3	Style transfer function control interface showing spheremaps at control points, with resulting rendered output	37
3.4	Volumetric rendering of human head and engine datasets (both CT) using style transfer functions with selective clipping planes for the depiction of salient features in context. Source: Paul Ross (EngD-project renderer).	38
3.5	Hybrid rendering the ‘head’ dataset using combined transfer functions: spheremap-based style transfer functions in conjunction with MPR	39
3.6	Supplementary shading in conjunction with style transfer function rendering of the foot. L-R: style transfer functions alone; alone; with additional curvature shading (see §4); with curvature shading and normal-based contours for context. Source: Paul Ross . . .	40

3.7	Stipple shading effect applied to human skull via Bruckner’s style transfer function method, with some resulting artifacts	45
3.8	Two styles of shading (stipple and glossy) applied to ‘sphere’ dataset using style transfer functions for varying levels of dataset smoothness. Note the artifacts in the stipple case; glossy shading is provided to indicate surface smoothness	45
3.9	Layering effect using style transfer functions: a thin, coloured low-opacity layer around an opaque surface can increase the perception of glossiness of the underlying surface . .	46
3.10	Illustrative cutaways and salient-feature highlighting using the ‘engine’ dataset using selective clipping, partial opacity and style transfer function rendering. Source: Paul Ross.	47
3.11	Style transfer function rendering of the ‘foot’ dataset: normal-based silhouette rendering to provide added context can be seen in the leftmost and rightmost images. The addition of a highly specular spheremap for flesh rendering has an additional effect on the shading of the bones due to the former’s translucency: their overall rendering style has an appearance similar to that of oil paint in traditional artwork. Source: Paul Ross.	48
3.12	Style transfer function rendering of the ‘foot’ dataset, with additional normal-based contours for illustrative applications	49
3.13	Style transfer function rendering of the ‘head’ dataset with partial opacity, cutaways and additional contours	50
3.14	Applying contours to individual structures via spheremaps with an integral contour ring	50
3.15	Highly contrasting shading styles, selective clipping and structure-specific contours to help isolate areas of key importance	50
3.16	Style transfer function renders of the ‘head’ dataset with various combinations of spheremaps with selective clipping	51
3.17	Style transfer function renderings of the ‘head’ dataset with selective clipping (fully transparent and partial opacity) for context preservation	51
3.18	Hybrid rendering using a combined transfer function (with a curvature transfer function and style transfer function) to depict the skull and spine within the context of the whole head	52
3.19	Improving facial feature depiction for style transfer function rendering of the 3D ultrasound ‘babyface’ dataset using additional normal-based shading and contours	54
3.20	Style transfer function rendering of the 3D ultrasound ‘hand’ dataset, with additional normal-based shading and contours	54
3.21	Low feature detail in ‘babyface’ 3D ultrasound rendering using style transfer functions alone	55
3.22	Style transfer rendering of heart-valve 4D ultrasound dataset with various shading maps and additional contour lines	56
3.23	Style transfer function rendering of heart-valve 4D ultrasound dataset, showing illustrative effect of additional contour lines for improved (but limited) shape depiction. Source: Paul Ross.	56

4.1	Contributions of contours and creases to overall shape depiction of illustrated hand. Left: contour lines only; right: contours and crease-lines. Original illustration source: Tom Richmond [6].	62
4.2	Effect of curvature shading on ultrasound foetal image showing enhanced feature detail. Source: Paul Ross.	63
4.3	Change in normal direction on a curved surface; the blue lines indicate the loci of points traced out by the principal curvature directions. Source: Kindlmann.	64
4.4	Volumetric renders of synthetic test cuboid dataset, using principal curvatures as shading metrics. Areas of negative and positive curvature are shaded purple and green respectively. Source: Paul Ross, after Kindlmann et al. [3]	65
4.5	Simple valley example showing selected surface normals and principal curvature direction vectors.	66
4.6	Simplified example of ridges and valley regions, showing surface normals; note that in this particular example, the Gaussian curvature is zero everywhere	66
4.7	Cross section of sinusoidal ridges and valley as depicted in 4.6, with associated principal curvature values κ_1, κ_2 . Illustration: Paul Ross.	66
4.8	Style transfer function rendering of the ‘engine’ dataset with supplementary curvature shading (various metrics)	69
4.9	Curvature shading to supplement style transfer function rendering (CT ‘foot’ dataset) for shading the joints between bones in the toes	70
4.10	Effect of curvature shading for shape depiction (specifically the joints) in illustrative rendering of CT ‘foot’ dataset	71
4.11	Subtle difference in shading using variants of minimum principal curvature	71
4.12	Enhancement of isosurface detail via curvature shading within the context of style transfer function rendering of CT ‘head’ dataset	72
4.13	Enhancing shape and surface depiction of the skull using curvature shading (minimum and maximum principal curvatures), together with normal-based contours	72
4.14	Using a small amount of curvature-based shading (both minimum and maximum principal curvatures) to provide a subtle enhancement in shape depiction for the ear (‘head’ CT dataset)	73
4.15	Style transfer function renderings with supplementary shading applied to an obstetric ultrasound dataset (‘babyface’ dataset)	74
4.16	The noise-highlighting effect of varying degrees of curvature-based shading (minimum principal curvature)	74
4.17	Curvature shading (minimum principal curvature) with undesirable noise-highlighting using style transfer functions with an alternative spheremap	75
4.18	Curvature shading to enhance shape depiction for a 4D ultrasound sub-volume (heart-valve) rendered using style transfer functions	76

4.19	Style transfer function rendering with curvature shading, using the same dataset subvolume (heart-valve) and view as in fig.(4.18) but with a higher degree of Gaussian volume smoothing for the input data	76
4.20	Style transfer function rendering with curvature shading, using the same heart-valve subvolume as in fig.(4.18) but with a different view and clip-plane orientation	76
4.21	Sequence of rendered images from successive sub-volumes in the temporal 4D series (heart-valve dataset). Source: Paul Ross.	77
4.22	Overview of deferred shading pipeline with smoothing stages for 2D curvature and colour buffers	79
4.23	Deferred and smoothed curvature shading for keepsake 3D ultrasound	79
4.24	Application of deferred and smoothed curvature shading to an ultrasound volume with a higher degree of volumetric Gaussian smoothing applied	80
4.25	The effect of 2D curvature and colour buffer-smoothing on reducing the appearance of noise	80
4.26	Application of deferred and smoothed curvature shading as a post-processing step on a <i>global illumination</i> babyface ultrasound rendering	80
5.1	Gaussian-smoothed obstetric ultrasound dataset rendered with style transfer functions and supplementary curvature shading: increasing the level of noise reduction results in a loss of feature detail. Source: Paul Ross.	87
5.2	Stopping functions $g(\ \nabla f\)$ for various values of parameter K	90
5.3	$g'(\ \nabla f\)$ for both variants of stopping function; large negative values can result in reverse diffusion and shock formation.	90
5.4	Comparison of Perona-Malik diffusion (with stopping function defined in eq.(5.8)) and standard Gaussian smoothing in the context of a 2D CT-slice image	91
5.5	Pronounced step artifacts from 10 iterations of the 3D Perona-Malik diffusion process in an obstetric ultrasound volumetric dataset	94
5.6	Pronounced step artifacts from 10 iterations of the 3D Perona-Malik diffusion process in the engine test dataset	95
5.7	Bilateral filter kernel (centre): essentially an attenuated Gaussian kernel. L-R: step function with added noise; bilateral filter kernel, filtered output. Figure from Tomasi,Manduchi [7].	96
5.8	Comparison of bilateral-filter and standard Gaussian smoothing in the context of a 2D CT-slice image	97
5.9	Effect of the attenuating range-component of the bilateral filter on feature-preservation .	100
5.10	Comparison of Perona-Malik anisotropic diffusion and the bilateral filter for similar levels of feature-preservation	101
5.11	Two iterations of the bilateral filter with $\sigma_d = 1.6$; note that stair-step artifacts are much more prominent here than with one iteration. Source: Paul Ross.	101

5.12	Comparison of Gaussian smoothing, mean-curvature motion and weighted curvature motion applied to obstetric ultrasound dataset	104
5.13	Stopping functions $h(f_{nn})$ for various values of parameter σ	108
5.14	Effect of stopping function h on overall smoothing using the Hossain-Möller PDE (15 iterations, $\lambda = 2$)	111
5.15	Effect of stopping function h on overall smoothing using the Hossain-Möller PDE (50 iterations, $\lambda = 2$) for ‘spheres’ dataset. Hybrid style transfer function/MPR rendering is used to illustrate the effect of h on both internal and isosurface-boundary noise	112
5.16	2D scalar-field plot of Hossain-Möller anisotropy function τ (see eq.(5.47)), for $\lambda = 0.5, 1, 2$ respectively. Towards-red depicts increasing local isotropy (towards $\tau = 1$). Source: Paul Ross.	112
5.17	Left: synthetic dataset with convex surface features (bumps) of varying anisotropy. Right: associated values for $\kappa_{min} + \tau\kappa_{max}$ (larger values are brighter). Source: Paul Ross.	113
5.18	Left: synthetic dataset with concave surface features (pits) of varying anisotropy. Right: associated values for $\kappa_{min} + \tau\kappa_{max}$ (larger-magnitude negative values are darker). Source: Paul Ross.	113
5.19	Synthetic dataset of bumps with varying anisotropy filtered with the Hossain Möller PDE (the stopping function h is set to 1 in this case): note that surface bumps with greater anisotropy are preserved more; the parameter λ controls the granularity of the anisotropy scale over which features are preserved or eliminated	114
5.20	Synthetic dataset of pits with varying anisotropy filtered with the Hossain Möller PDE (the stopping function h is set to 1 in this case)	114
5.21	Effect of varying the anisotropy function parameter λ for the ‘babyface’ dataset: note that $\lambda = 0$ corresponds to MCM when stopping function h is set to 1; otherwise it is MCM with an overall Perona-Malik type stopping function. Source: Paul Ross. . . .	115
5.22	Surface-noise still remaining after 10 iterations of the unmodified Hossain-Möller PDE-based smoothing process (circled) with $\sigma_h = 0$, i.e. h set to a constant value of 1	116
5.23	Left-Right: Gaussian smoothing (1 iteration, $\sigma = 1.5$), the standard Hossain-Möller PDE (10 iterations, $\lambda = 2$), the Hossain-Möller PDE modified for selective MCM via thresholding of $\ \kappa_{max}\ $ (10 iterations, $\lambda = 2$, τ -threshold = 0.16). Note that the latter two have a small amount of regularizing Gaussian pre-smoothing applied ($\sigma = 0.85$). Source: Paul Ross.	118
5.24	Surface noise removed after 10 iterations of the Hossain-Möller PDE with selective MCM via curvature-thresholding of τ	118
5.25	Effect of thresholding the anisotropy function τ of the Hossain-Möller PDE using different threshold values of $\ \kappa_{max}\ $ (15 iterations, $\lambda = 2$)	118
5.26	Effect of increasing τ -threshold for the modified (selective-MCM) Hossain-Möller anisotropic diffusion PDE. Top row denotes the corresponding τ function: red indicates that the region will be smoothed isotropically, blue anisotropically	119

5.27	Effect of increasing τ -threshold for the modified (selective-MCM) Hossain-Möller anisotropic diffusion; top row depicts the corresponding τ function	120
5.28	Results after application of 15 iterations of the modified Hossain-Möller PDE (with thresholded anisotropy function for selective MCM) for various threshold values of $\ \kappa_{max}\ $ and stopping function parameter σ_h	121
5.29	Global illumination ultrasound (GI) rendering of the obstetric hand dataset with additional curvature-based shading for feature enhancement. Left: Bilateral filter; Right: new PDE-based method. Source: Paul Ross.	122
5.30	Global illumination ultrasound (GI) rendering of the obstetric ‘babyface’ dataset with additional curvature-based shading for feature enhancement. Left: Bilateral filter; Right: new PDE-based method. Source: Paul Ross.	122
5.31	Comparison of smoothing results using the standard Hossain-Möller PDE (middle) and our modified version (right) on the pre-smoothed CT ‘foot’ dataset (left)	125
5.32	L-R: Pre-smoothed vessel dataset using one iteration of Gaussian filtering with $\sigma = 0.85$; with additional Hossain-Möller smoothing; with the modified Hossain-Möller smoothing (our method)	126
5.33	Comparison of volumetric smoothing methods applied to obstetric ultrasound dataset (all with the same degree of curvature shading for feature-emphasis)	126
6.1	Filtering and rendering pipeline of operations for the overall system	131
6.2	Enhancement of feature detail with minimal noise in obstetric ultrasound: curvature shading and filtering are used with both style transfer functions and global illumination renderers.	132
6.3	Another example of feature-detail enhancement with minimal noise in obstetric ultrasound keepsake imaging: note the finger creases but lack of noise in the palm region. Source: Paul Ross.	132
6.4	Style transfer function rendering with supplementary curvature shading to improve shape depiction in 4D ultrasound heart-valve data. See figs. (4.18) and (4.19) for comparison with style-transfer function shading alone. Source: Paul Ross.	132
6.5	Application of the system to cranial blood vessels with additional depth cueing: multiple styles are presented according to required criteria in each case. L-R: a more subtle illustration style; non-photorealistic render featuring post-processing ‘toon’ shading using Perona-Malik filtering on the colour buffer; contrasting style designed to highlight regions of high minimum principal curvature for visual aneurysm detection (using HSV-curvature shading). Source: Paul Ross.	133
6.6	Non-photorealistic rendering styles for aneurysm depiction in vessel with depth cueing: Left: warm-to-cool Gooch shading; Right: toon shading with post-processing cel-shading effect using Perona-Malik diffusion. Source: Paul Ross.	133

6.7	Non-photorealistic rendering of the CT ‘head’ dataset using curvature-based shading for facial feature enhancement and increased definition in skull texture. Curvature-based volume filtering is also applied to reduce noise on the skull surface for a slightly abstracted look. Source: Paul Ross.	134
6.8	Curvature-based shading is highly effective in highlighting joint regions in the foot; contour lines and partial opacity provide added context. Source: Paul Ross.	134
6.9	Smoothing methods compared for style transfer function (STF) rendered obstetric ultrasound with curvature shading	135
6.10	Smoothing methods compared for style transfer function obstetric ultrasound with curvature shading (‘arm’ dataset)	135
6.11	Application to global illumination ultrasound to highlight feature detail with minimal noise. Left-hand images depict global illumination only; right-hand images have the additional curvature shading step. Top row: bilateral filter; Bottom row: our method (modified Hossain-Möller anisotropic diffusion with selective mean curvature motion). Source: Paul Ross.	136
6.12	Smoothing methods compared for global illumination obstetric ultrasound with curvature shading	137
6.13	The effect of curvature shading on feature depiction for global illumination obstetric ultrasound	137
A.1	Gauss map: the traced-out line on the surface of the unit sphere (left) corresponding to the movement of the normal on a surface along circular path (right). Source: Wolfram Demonstrations Project, Demo author: Michael Rogers.	149

List of Patents and Publications

Paul Ross, Tim Day, Robert Davey, *Deferred and Smoothed Curvature Shading for Detail Enhancement in Smoothed Ultrasound Volumes*: US Patent Application no 14/661,456

Table of Symbols

f_x	First partial derivative, $\frac{\partial f}{\partial x}$.
f_{xx}	Second partial derivative, $\frac{\partial^2 f}{\partial x^2}$.
∇	gradient operator (on scalar field), $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}]^T$.
∇f	gradient of scalar field f , $[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]^T$.
div	divergence operator (on vector field), ∇^T .
$\nabla \cdot$	divergence operator (shorthand).
$\nabla \cdot \mathbf{v}$	divergence of vector field \mathbf{v} , $\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z}$.
∇^2	Laplace operator (Laplacian), $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$.
\mathbf{n}	unit normal vector.
\mathbf{H}	Hessian matrix (of second partial derivatives) of scalar field.
κ_1	maximum principal curvature (of manifold embedded in scalar field).
κ_2	minimum principal curvature (of manifold embedded in scalar field).
κ_{min}	minimum principal curvature by magnitude.
κ_{max}	maximum principal curvature by magnitude.

K mean curvature, $\frac{\kappa_1 + \kappa_2}{2} \equiv \frac{\kappa_{min} + \kappa_{max}}{2}$.

G Gaussian curvature, $\kappa_1 \kappa_2$.

Table of Acronyms

AD	Anisotropic diffusion.
HM	Hossain-Möller anisotropic diffusion.
MCM	Mean curvature motion.
WCM	Weighted curvature motion.
PDE	Partial differential equation.
PM	Perona-Malik nonlinear diffusion.
STF	Style transfer function.
CTF	Curvature transfer function.
CS	Curvature shading.
CD	Central differences.
FE	Forward-Euler.
RK4	Runge-Kutta 4th order.
GPU	Graphics processing unit.
MPR	Multiplanar reformatting.

Chapter 1

Introduction

This research project outlines the development of an illustrative rendering (and noise filtering) system for the depiction of volumetric medical imaging datasets; it is designed to preserve and enhance salient feature-detail whilst reducing the visible effects of inherent noise more effectively than the current state-of-the art methods. Key components are a novel feature-preserving PDE-based noise filtering method and supplementary feature-enhancing shading using a unified curvature-based approach. Depiction of salient feature details (and/or their enhancement) and the reduction in visible noise-based artifacts are fundamentally competing objectives: this system provides an effective means for smoothing datasets whilst preserving and enhancing features. Whilst ostensibly an illustrative rendering system, the key novel contributions can be applied to other rendering contexts such as *global illumination* to preserve and enhance feature detail with minimal visual noise artifacts. In the field of obstetric ultrasound, the novel contributions produce visual results which surpass current state-of-the art techniques.

The projects two fundamental objectives are:

1. The development of an illustrative rendering system with potential applications in obstetric ultrasound, surgery planning, patient education and shape-perception testing.
2. Within the context of this illustrative rendering system and more generally, the development of a unified curvature-based approach to shading-enhancement and volume smoothing: supplementary shading techniques and a novel data smoothing method were developed to enhance the depiction of salient features whilst suppressing visible noise artifacts and extraneous surface detail.

1.1 Background

A major discipline within the field of medical visualisation is concerned with the presentation of rendered 2D images derived from three-dimensional (volumetric) data. Datasets are acquired via hardware scanners in the context of their respective *modalities*; each modality provides its own particular insight. Magnetic Resonance Imaging (MRI) and CT (Computed Tomography) are generally employed within the contexts of soft tissue and skeletal imaging respectively, whilst ultrasound sonography has applications within obstetrics, blood-flow analysis and internal organ visualisation. Three-dimensional ultrasound is a relatively recent development and is particularly suitable for parent-oriented *keepsake* obstetric imaging.

A volumetric dataset is essentially a three-dimensional array of data-sample points known as *voxels*, where a voxel can be thought of as a 3D analogue to a 2D pixel. The volumetric dataset can be considered to be a 3D lattice with each point representing the centre of each voxel. *Volume rendering* and *explicit isosurface extraction* are two imaging techniques used to process volumetric data and produce an output image: the former is currently more prevalent and will form the basis of the forthcoming discussion.

Illustrative rendering takes inspiration from the artistic styles and methodology employed by traditional artists and medical illustrators in order to convey information effectively with minimal extraneous detail. Such techniques aim to improve the efficiency of information communication: they emphasise and isolate key regions or structures and reduce unnecessary detail in areas of lower importance whilst retaining the overall context.

Generally, illustrative rendering methods have evolved directly from those used by traditional illustrators: most fundamental are low-level abstractions such as the overall shading style (depicting lighting and surface properties) and shape-conveying crease and silhouette lines. Higher-level constructs include expanded or exploded views, ghosting and cutaways.

There are numerous applications in the field of medical imaging: surgery planning, education and training, and the presentation of scan-data to the patient or other layperson are a few possibilities. The latter facilitates dialogue between the medical professional and the patient, and can significantly help the patient's understanding of their condition or impending surgical procedure.

Illustrative shading can enhance the depiction of vessels or other salient structures by juxtaposing their depiction against a traditionally-rendered surrounding context.

In addition to the fundamental shading style, illustrative or otherwise, traditional artistic techniques for highlighting region boundaries can also be transposed to medical visualisation applications. Crease and silhouette lines help to delineate surface structures and can provide a greater sense of depth and overall shape than the basic shading or light-

ing style alone. Such lines are the simplest and most extreme abstract expression of the highlighting of *ridges and valleys*, which often constitute the boundaries of features on a given surface. A more comprehensive graduated shading of such regions extends this to provide a greater degree of shape-cueing if required; moreover, the degree and extent of such shading can be chosen to complement the overall chosen artistic style.

Three-dimensional foetal ultrasound imaging has become increasingly popular with parents as a ‘keepsake’ or souvenir, and as means to provide a visual update on their baby’s progress through pregnancy; such imaging can provide a tangible reassurance regarding their baby’s wellbeing, and a visual record of the various stages of its development. Within this context, an overall non-photorealistic rendering style may have potential as an alternative to global illumination (the current preferred state-of-the-art rendering method for this application).

Overall rendering style aside, supplementary illustrative techniques such as ridge and valley enhancement have the potential to address one pervasive issue: the problem of indistinct feature-boundaries, such as those around the eye and mouth regions. Where global illumination itself as a main rendering method can provide some improvement over conventional gradient-based shading, ridge and valley enhancement can be used to supplement shading of feature-boundaries for both gradient-based renderers (including the illustrative ones discussed in the forthcoming chapters) and global illumination renderers.

However, 3D ultrasound data suffers from another major problem: a significant noise component. Like other modalities, 3D ultrasound suffers from noise accrued through the data acquisition process. In the case of ultrasound, a large proportion of undesirable artifacts are produced as a result of interference patterns set up between the probe transducer and soft tissues; this is known as *speckle*.

The issues of poor feature delineation and noise are inextricably linked. Attempts to address the former by increasing feature-boundary definition also increases the effect of the visible noise artifacts; moreover, attempts to reduce the visual level of speckle and other noise-based artifacts directly reduces the distinction of feature-boundaries.

Ultimately, the main objective in this context is to achieve an optimal balance of feature preservation (or enhancement) and noise suppression. This research project has this as its primary motivation: a flexible rendering pipeline featuring a specially developed feature-preserving volumetric filtering stage, combined with feature-enhancing supplementary shading for maximum feature retention with minimal noise artifacts.

Whilst particularly effective for 3D ultrasound, this approach is not limited to this application: it can be employed across all modalities which produce volumetric voxel-based data. Overall, this is presented within the context of an illustrative rendering framework; this presents an alternative to traditional rendering styles and is particularly useful for vessel and structure visualisation and highlighting. Moreover, the non-photorealistic styles

are easily interchanged and can be replaced with global illumination or other traditional renderers if required.

1.2 Project aims and scope

1.2.1 Requirements

This project seeks to develop a system for the rendering and filtering of medical images of various modalities from their respective volumetric datasets with the following specific requirements:

1. The ability to highlight salient features or structures; this is achieved through either illustrative shading styles or supplementary illustrative shading based on 2nd derivative properties (curvature).
2. Effective noise suppression with a higher level of feature preservation and noise reduction than is available with current methods.
3. Fast execution time to allow for real-time application to streaming 4D ultrasound.

In addition, shading styles should be easily interchanged and even combined with other rendering methods (or replaced entirely) for a highly flexible pipeline.

1.2.2 Scope

The above takes inspiration from some traditional illustrative rendering techniques: indeed, this thesis depicts the development of a primarily illustrative-based rendering system; however, some of the artistic methods (e.g. detail-enhancing shading) can be applied in the context of other rendering methods and styles (e.g. global illumination).

The fundamental thread running through the entire project is the combination of the first two of these requirements. The specific aims are to enhance or preserve salient features whilst removing excessive noise or unwanted detail to a greater degree than is currently available.

The system has particular value in the context of 3D ultrasound as this modality is inherently noisy with poor feature delineation. Hence, a large proportion of development time will be devoted to improving facial feature detail in obstetric ‘keepsake’ images; this application will be studied throughout the duration of the thesis and its examples will document the progression of the pipeline development to meet the underlying objectives. However, the system is not limited to 3D ultrasound in application: the techniques devel-

oped within have equal potential in the context of other modalities such as CT and MR and examples will be outlined throughout.

1.2.3 Implementation

Balancing these competing objectives is achieved by advancing the state-of-the-art in feature-preserving volumetric filtering in conjunction with supplementary feature enhancing shading; indeed, the development of a novel volumetric noise reduction method to facilitate the above primary requirements is one of the major contributions of this research project; this new PDE-based method is described in §5.5.

Overall, the pipeline consists of the following:

- i. A base illustrative rendering system using Style Transfer Functions (Bruckner [5]), although this is swappable with other renderers (e.g. global illumination).
- ii. Supplementary curvature-based shading for feature enhancement; employs curvature buffer-smoothing where required for noise-artifact reduction.
- iii. Improved feature-preserving volumetric filtering to allow for feature-enhancing shading without the side-effect of noise artifact amplification, using a combination of anisotropic diffusion and mean-curvature motion.

In order to satisfy requirement 3 (from §1.2.1), all stages of the pipeline (filtering, rendering, compositing) are implemented on GPUs (graphics processing units) using GPGPU* principles and libraries. Both rendering and filtering operations are highly parallelisable (sometimes termed *trivially* or *embarrassingly parallel*: each voxel can be processed independently with no synchronisation, dependency or write-clash issues). Hence, both map well onto parallel processing architectures and are highly suitable for implementation on GPUs.

Nvidia’s CUDA was the GPU framework of choice but the open standard OpenCL would be applicable also. As the system was developed on Nvidia GPUs, and CUDA currently offers a slight performance gain over OpenCL, the former was chosen for all project development. Moreover, Toshiba Medical’s internal libraries make use of this framework, so integration of new code back into the company’s main code-base is facilitated by its adoption.

The renderer uses the open standard graphics library OpenGL for final image screen-placement; the system is set up so that OpenGL shares internal buffers with the CUDA-based renderer.

*GPGPU or general-purpose GPU-computing is the use of the highly parallel architecture of the GPU to perform general computing (non-graphics API) tasks, e.g. numerical integration, raycasting etc.

Performance is a key requirement in general but is of fundamental importance for streaming 4D ultrasound, where a succession of 3D volumes are continually generated (at a rate of 10 volumes per second[†] at the time of writing). The major potential bottleneck over current 4D implementations is the volume filtering stage. However, on current GPU hardware (Nvidia GTX680), a 4D ultrasound stream can be processed in real-time with the newly-developed filter so that each 3D volume can be denoised in around 3ms.

1.3 Project contributions

Built upon a flexible illustrative-rendering framework, the fundamental overall contribution lies in the combination of an illustrative curvature-based shading stage for detail enhancement and an improved feature-preserving non-linear volumetric smoothing method; the core achievement is the ability to highlight salient feature detail with minimal noise artifact amplification. These objectives are in some sense a ‘holy grail’ of volumetric image processing. A significant component is a newly developed method for volumetric noise reduction which can eliminate a large proportion of rendered isosurface noise with a minimal compromising of feature detail.

To effect the goals of feature detail enhancement with minimal noise-based artifacts, the following methods are employed and form the basis of the novel contributions presented in this thesis:

- Deferred curvature based shading as a post-processing step with 2D curvature-buffer smoothing.
- An improved feature-preserving 3D volumetric filtering stage.

Overall, the contributions can be described as follows:

Volumetric filtering

The major contribution is the development of a novel PDE-based second-order filtering method which provides an improved balance of feature retention and noise reduction over current state-of-the-art methods. Building on work by Hossain and Möller [8], Krissian [9] and Carmona [10], this non-linear PDE makes use of anisotropic diffusion (as described by Hossain and Möller) for feature preservation but extends the range of local surface topologies (local anisotropy) over which purely isotropic smoothing is applied (through mean curvature motion). The latter is applied according to local curvature magnitude, regardless of anisotropy.

[†]on Toshiba 4D Ultrasound ‘Artida’ hardware

Complementary to the required feature-preservation under anisotropic flow, the additional isotropic mean-curvature motion has two effects:

- removal of residual surface noise that would otherwise remain untouched by anisotropic diffusion methods.
- a high level of performance as isotropic flow gives the largest spatial stepsize with fast convergence.

Hence this method allows for the retention of salient features as in the Hossain Möller anisotropic diffusion PDE, but offers an improved level of noise-reduction and overall performance.

Deferred and smoothed curvature shading

This is a shading post-processing step which provides the means to highlight salient feature detail with a reduction in the level of unwanted noise-enhancement. It is a supplementary shading stage which can emphasise region boundaries such as ridges and valleys, using curvature as a shading metric. Combined with the novel volume-filtering method outlined above, it is effective for enhancing feature detail with minimal noise enhancement. As a deferred shading stage, it is suitable for single rendered isosurface applications such as 3D ultrasound. Curvature data is obtained on ray termination (at the isosurface) and is stored in a 2D curvature buffer. This buffer is then smoothed using 2D Perona-Malik or bilateral filtering to reduce the effects of high curvature produced at noise-related surface peaks or pits on the supplementary shading in the any given local region.

Pipeline of operations

The fundamental contributions outlined above are presented within the context of a pipeline of operations; this is designed to facilitate the implementation of the above methods in a multitude of rendering applications for a variety of modalities. The pipeline is presented as a complete illustrative rendering system but the feature enhancement and noise reduction components can work with other rendering methods (e.g. global illumination). A generic version of the overall system (with a non-specific renderer and general non-specific feature-preserving volumetric filtering stage) is currently under review for US Patent Application no 14/661,456 (*Deferred and Smoothed Curvature Shading for Detail Enhancement in Smoothed Ultrasound Volumes*).

1.4 Possible applications

As previously discussed, one major application of this new rendering pipeline is the enhancement of feature detail and region boundaries in 3D ‘keepsake’ obstetric ultrasound imaging in order to better delineate regions such as the eyes, nose and mouth; this is applicable for both fully-illustrative overall shading styles and for global illumination.

Illustrative shading styles in general can be used for surgical planning, patient education, vessel depiction and for any application where a non-photorealistic rendering style is appropriate; moreover the supplementary curvature-based shading stage can provide additional detail enhancement where required. Curvature-based shading can be useful for highlighting pathology: in the case of vessel rendering, aneurysms can be highlighted by shading the regions of high-curvature that define their boundaries (i.e. the ‘neck’ of the aneurysm).

The core feature-preserving volumetric noise-reduction method may be generally applicable to all volumetric contexts to provide a state of the art replacement for older methods such as the bilateral filter or Perona-Malik diffusion. The volumetric smoothing component is effective at preserving tubular structures, so is particularly suited to vessel rendering or noise reduction operations on such datasets.

In addition, the illustration-inspired curvature-based shading method may be used as a post-processing step to enhance detail within any rendering style.

In summary, potential applications for the overall system include:

- Illustrative rendering, e.g. 4D ultrasound heart valve depiction and animation
- Surgical planning
- Patient education
- Perceptual psychology studies on shape perception
- Ultrasound ‘keepsake’ imaging
- Vessel aneurysm depiction (illustrative)
- Any application where feature enhancement is required.

Chapter 2

Literature review

2.1 Introduction

The fundamental project objective (as discussed in the previous chapter) is to develop a high-performance volume filtering and rendering system for application to medical imaging datasets, with the capacity for salient feature highlighting with minimal noise-based artifacts. The rendering system is to take inspiration from illustrative methods for primary shading styles and supplementary feature-enhancing shading.

What follows initially is a general overview of volume rendering, introducing key concepts such as illumination models and transfer functions. Illustrative rendering techniques for general 3D computer graphics are introduced and then discussed within a volume-rendering context. Illustrative rendering has been the subject of a considerable body of research overall: most work involves image-space or polygonal 3D object space methods. However, there has been a degree of effort to transpose much of these techniques to the context of volume rendering. Volumetric visualisation itself increases the scope of possibilities: managing occlusion is a major research area, and can derive much inspiration from traditional artistic methods.

Volumetric noise reduction forms a major component within this research project and one of its main contributions. In the literature, it is an area of significant research: state-of-the-art methods will be discussed here and developed in further chapters.

In order to address the fundamental criteria discussed in the previous chapter, current literature in the following research areas will be discussed:

- general methods for rendering volumetric data
- illustrative techniques and their application in a volume-rendering context
- supplementary illustrative shading methods, e.g. ridge and valley shading; such methods could be employed for salient-feature enhancement

- volumetric noise reduction to minimise the effect of noise-highlighting incurred through feature enhancing shading

Ultimately, selected methods outlined in this literature review chapter will form the basis for further development within the scope of this research project.

2.2 Volume-rendering overview

There are several approaches to rendering volumetric data, falling generally into two distinct categories:

- Surface-reconstruction approaches, which attempt to recreate an explicit surface mesh using an algorithm such as *marching cubes* [11]. More sophisticated reconstruction techniques have since been developed [12].
- Direct volume rendering methods, where no explicit surfaces are recreated. The final image (which may include implicit isosurfaces) is rendered directly from volume data. Examples are *slice-based* 2D-texture stack methods, and *raycasting*.

Medical imaging applications are primarily concerned with direct volume rendering. Out of the two such methods listed, the latter raycasting approach requires more powerful GPU hardware, but is now generally preferred [13]. Slice-based methods were previously favoured until the introduction of more capable GPUs: these have improved rapidly in recent years. The advent of shaders, then GPGPU* libraries like Nvidia's Cuda and the open-source standard OpenCL, allow complex flow-control and program constructs, although branching must be carefully considered for performance reasons.

2.2.1 Raycasting

Considering the volumetric data-set as a 3D texture, the raycasting algorithm is highly parallelisable and thus maps easily onto GPU architectures[†]. The raycasting algorithm essentially computes output pixel values in an image plane by approximating the *volume rendering integral*[‡] along rays fired into the volume from the image plane. This involves accumulating intensity and opacity values according in a *compositing* operation at a specific sampling interval along the ray. The intensity at a particular sampling point is usually reconstructed using either trilinear or cubic-spline interpolation, since the volumetric data-set consists of a set of discrete data points.

*General Purpose Computing on GPUs

[†]GPUs have large numbers of processing units (cores) and very fast thread switching. They are actually a hybrid SIMD/MIMD model: small groups of threads known as *warps* operate as SIMD units, and can be switched quickly in the event of a stall.

[‡]see Max (1995)[14], Krueger (1990)[15] for a comprehensive description of the volume-rendering integral.

GPU hardware is a perfect match for the raycasting algorithm:

- It is highly parallelisable (so called *embarrassingly parallel*): each output pixel value can be computed entirely independently in parallel with others on a output-pixel-per-thread basis, with no synchronisation, dependence or write-clash issues between threads.
- The output image can be constructed in texture-memory, then rendered as a textured quad through an API such as OpenGL; it can thus remain in device memory, and avoid time-consuming data copying from main-memory to the GPU. Moreover, recent iterations of Nvidia GPU hardware (Fermi generation or newer [16]) allow direct writing to texture memory, without any internal GPU data copies.
- The GPU texture-unit's hardware trilinear-interpolation features can be employed. Cubic-spline based interpolation can be implemented either via a shader or a GPGPU kernel, although the latter is preferred if using Cuda or OpenCL as part of the pipeline; a performance-optimized version is presented in [17], in which the trilinear interpolation hardware is used to affect tricubic interpolation.

2.2.2 Transfer Functions

So far, only gray-scale composited values have been considered. To model effectively the emission and absorption of various tissue component structures within the volume, a *transfer function* lookup-table is required to map volume intensity values to RGBA colour and opacity. This is frequently stored as a texture (in its simplest form a 1D texture). This process is essential to effectively classify the data into different tissue types for visualisation.

Transfer function construction can be a laborious process: to effectively separate different tissue types requires extensive manual transfer function design by trial and error. Various techniques attempt to simplify or replace this process: see [18] for a comparative overview of transfer function design approaches.

Levoy (1988) [19] extends the basic transfer-function to two-dimensions, introducing voxel intensity-gradient as a second input parameter. This can be used to help delineate boundaries between adjacent tissue types, via gradient opacity-modulation. Furthermore, it can also form the basis for illustrative techniques, such as *ghosting* [20].

Kniss et al. further discuss the possibilities for multidimensional transfer-functions in [21]; Kindlmann et al. introduce curvature as an input parameter in [3]: this has particular significance in the context of illustrative rendering (see §2.5.1).

Other recent techniques involve introducing the scale of features [22] and textural properties [23] as input parameters.

2.3 Illumination models

Illumination can help provide a clearer depiction of isosurfaces within the volume, and may improve user-perception of depth and spatial relationships between structures within a volume data-set. Lighting-model approximations generally fall into two distinct categories: *global* and *local*.

Global illumination

This is a complex model, in which scattering and global reflections between surfaces are considered [24]. This type of model can generate subtle lighting effects such as high-quality translucency and volumetric shadows, and is effective at communicating subtle detail within a volumetric data-set that may be missed by simple local illumination [25][26]. Lindemann (2010) describes a method for simulating advanced light-material interaction in volume rendering contexts [27].

Local illumination

These are relatively simple approximations, where only light rays emanating directly from the source are considered at the given surface point; scattering effects are ignored. A simple example is the *Blinn-Phong* lighting model [28]: this is frequently used in lighting implicit isosurfaces, such as the skull.

A primary concern is user-perception, and thus efficacy in information-communication. Lindemann (2011) presents a comparative user-evaluation summary of various lighting techniques in [29]. For a current overview of state-of-the art global illumination for general computer graphics, see [30].

Surface-normal estimation Computation of intensity gradients also has another important application in direct volume rendering: surface-normal approximation for illumination and shading. In direct rendering methods, no explicit isosurfaces are constructed, so some other means of normal-construction must be utilised; in fact, the normalised voxel intensity-gradient is equivalent to a surface-normal [19]. Several methods exist for gradient estimation:

1. simple finite difference schemes, central differences being preferred.
2. other convolution-based methods using larger filter kernels than simple finite differences, together with some form of interpolation should the continuous derivative be required.
3. fitting methods such as splines and patches, although the former can be considered a variant of the second type.

The second category generally produce higher quality results than simple finite-difference schemes, but are more computationally expensive. See [31] for a general overview and

[32][33] for a comparison of various schemes. Recent high-quality gradient-reconstruction techniques are discussed in [34] and [35]. Belyaev discusses implicit finite-differences in [36] for fast, high quality image derivatives.

An example of the second category above is the *cubic spline*: these are often used for greater reconstruction accuracy and are often referred to as ‘‘texture sampling with *tricubic interpolation*’. Sigg and Hadwiger [17] recommend that cubic splines are used for gradient and second derivative sampling, and that simple finite differences will suffice for data value sampling. Moreover, they present a performance-optimised implementation using a combination of trilinear texture sampling operations (as the latter is implemented in hardware on most modern GPUs).

Kindlmann et al. [3] investigate a selection of convolution-based filters for derivative estimation using the Marschner-Lobb dataset [37]; they then present a visual comparison of the results.

2.4 Illustrative techniques

Conventional approaches to rendering in computer-graphics has typically involved attempting to recreate a photorealistic depiction of a scene; in effect, approximating the real-world interaction of light with the scene’s various objects and surfaces.

Non-photorealistic rendering (NPR) attempts to simulate the styles and techniques employed by artists in traditional illustration; the emphasis is on the effective communication of information, through a degree of abstraction. Gooch and Gooch [38] provide a comprehensive overview of illustrative techniques in both 2D (image-space) and polygonal 3D (object-space) contexts, although this is limited to summarising the field up and including the SIGGRAPH 2000 conference. The main illustrative approaches described in [38] involve emulating pencil and ink illustration styles such as enhanced edges and shading; cartoon or other novel lighting/shading styles are also outlined in detail.

In summary, NPR techniques involve:

- Emphasised edge lines, traditionally known as *contours* or *silhouettes* in an artistic context. Contours delineate the transition between forward and backward-facing surfaces; other edge-lines such as *creases* signify an orientation discontinuity in forward-facing surfaces.
- Shading techniques such as *hatching* and *stippling*: these can give an enhanced perception of shape and surface curvature, whilst remaining visually minimalist in style.
- Other abstract lighting and shading effects, such as *toon shading* and *Gooch shading*.

ing.

2.4.1 Lines for emphasis

Gooch and Gooch (2001) [38] describe in detail various object-space algorithms for creating contours and crease-lines. Both types of lines can help to improve shape perception: figure (2.1) illustrates the effect on overall shape-depiction of the addition of both types of lines. Note that the bottom left image (just the lines themselves) constitutes an illustrative rendering in itself: the crease and contour lines are sufficient to provide a sense of overall shape.

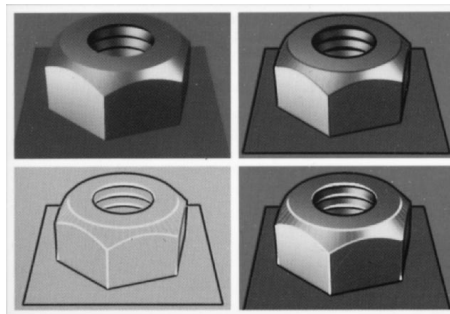


Figure 2.1: Renderings of nut, showing the effect of contour and crease lines on overall shape-depiction. Clockwise from top left: no contour or crease lines; with the addition of contour lines; with the addition of contour and crease lines; contour and crease lines only. Source: Saito (1990) [1].

2.4.2 Illustrative shading to simulate pencil and ink methods

Hatching and *stippling* are two major traditional pencil-and-ink shading techniques. These have been discussed extensively by Gooch and Gooch [38] for standard object-space 3D surfaces: see fig. 2.2; extension to a volume-rendering context is discussed in §2.5.2.

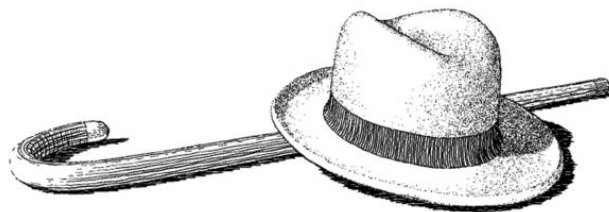


Figure 2.2: Hat and cane modelled with B-spline surfaces and shaded with stippling and hatching respectively. Source: [2].

2.5 Illustrative techniques: volumetric data

The previous section served as an introduction to general illustrative techniques; what follows is a general overview of the methods for their application in a volume-rendering context.

2.5.1 Contours

Contour lines present a boundary between front and back-facing surface geometry, and thus can be constructed as a function of the dot product of the view and surface-normal vectors. Such lines are thus based on first order differential properties of the dataset's embedded isosurfaces, i.e. their presentation is *gradient-based*. Crease-lines derived from 2nd order differential properties (curvature) will be discussed in the context of feature enhancement in §2.7.

Contours in traditional illustration generally have a consistent *stroke* or thickness. However, contour lines in a volume rendering context suffer from an inconsistency in their thickness due to variations in surface curvature along the viewing direction. Kindlmann [3] describes a method for regulating contour thickness using curvature computed along the view direction (see fig.(2.3)). Bruckner [5] develops a similar approach but with a cruder approximation for the curvature computation for improved performance.

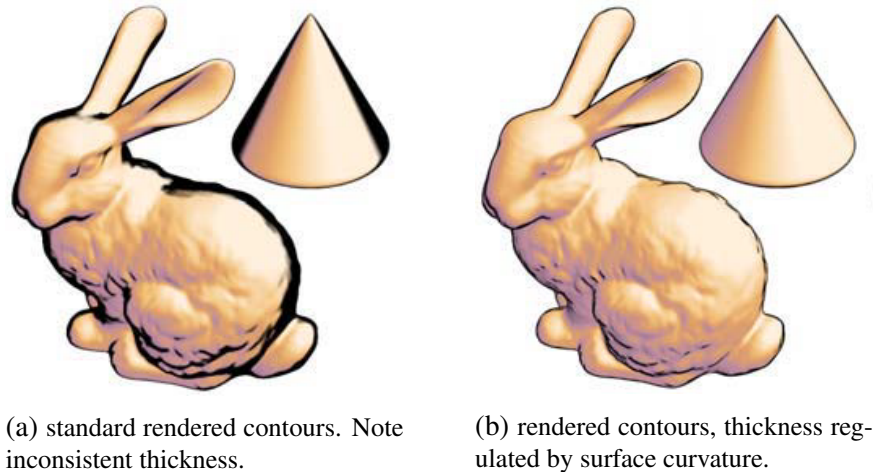


Figure 2.3: Comparison of illustrative contour-rendering based on dot product of view direction and surface normal (\mathbf{v} and \mathbf{n}), unregulated (a) and regulated (b). Source: Kindlmann et al. (2003) [3].

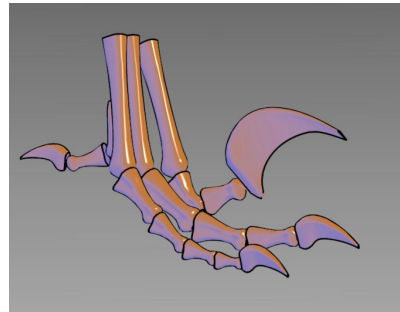
Suggestive contours, creases and general line drawing See §2.7 for a comprehensive summary of feature-enhancing shading methods.

2.5.2 Illustrative shading

Toon and Gooch Shading These stylised shading models are usually described in the context of standard polygonal-mesh rendering of surfaces, but both styles can be easily be recreated for volume-rendering algorithms in the shading of implicit isosurfaces. *Toon rendering* features a highly quantized colour-table lookup, whose shade is dependent on the relation of view angle to surface normal; it aims to replicate the illustrative style of traditional cartoon artwork with illumination depicted by bold bands of colour. *Gooch shading* or *tone shading* employs a cool-to-warm hue shift, coupled with rendered silhouettes [39]. Both styles (see fig.(2.4)) could be effective in illustrative rendering of medical data, as simplified shading can help to improve clarity by stripping away unnecessary detail.



(a) Teapot rendered with toon shading. Source: Wikimedia Commons.



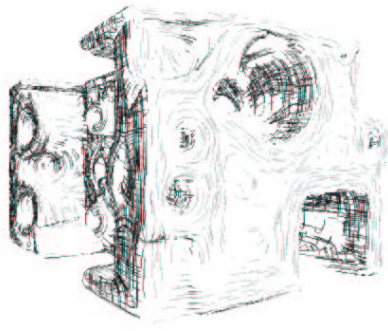
(b) Gooch or tone shading. Source: Gooch (1998) [39].

Figure 2.4: Two styles of simplified illustrative shading.

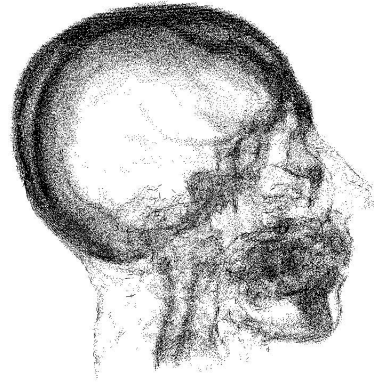
Rheingans and Ebert (2001) introduce illustrative techniques such as silhouette sketching and tone shading in a volume-rendering context in [40].

Hatching and Stippling There have been several attempts to apply fundamental illustrative shading techniques (such as hatching and stippling) to volumetric rendering contexts. Nagy et. al (2002) [41] describe a method to perform non-photorealistic hatching for volumetric data; another major illustrative shading style, stippling, is outlined in [42] for both volume and surface renderers (see fig.(2.5)).

Furthermore, Bruckner and Gröller present an example of stipple-rendering pencil illustration as one shading style possible with their *Style transfer function* rendering method [5]: this overall system is introduced briefly in §2.6 and is discussed in greater detail in §3.



(a) Illustrative hatching for volume data (engine). Source: Nagy et al. (2002) [41].



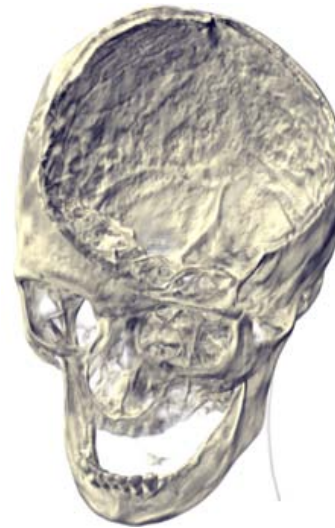
(b) Illustrative stipple rendering for volume data (skull). Source: Lu et al. (2003) [42].

Figure 2.5: Illustrative hatching and stipple shading of volumetric data.

Exaggerated shading Rusinkiewicz et al. (2006) [4] present a geometry-dependent lighting method to produce an effect similar to that seen in cartographic terrain-relief [4]. The lighting direction is modulated to effectively ‘graze’ the surface: normals are successively smoothed, and then the lighting direction is adjusted to become perpendicular to these smoothed normals. The result is an exaggerated shading style, providing greater contrast for surface features (see fig. 2.6).



(a) Skull rendered with cosine shading.



(b) Skull rendered with exaggerated shading.

Figure 2.6: Increased detail due to exaggerated shading in skull model. Source: Rusinkiewicz et al. (2006) [4].

Particle systems GPGPU methods can be used to render particle systems; these can provide an flexible extension to the other illustrative shading methods previously outlined. See Van Pelt et al. [43].

2.6 Style Transfer Functions

In order to facilitate the quick specification and swapping of different rendering styles, Bruckner and Gröller introduced the *style transfer-function* in 2007 [5]. This presents an elegant and logical extension to the standard transfer-function concept by adding normal-based shading maps at the control points. Contour colour and shading-style is determined by the transfer-function itself to retain visual consistency throughout the given style.

2.6.1 The Lit Sphere

Functionally, style-transfer functions make use of a spherical lighting-map described by Sloan, Gooch et al. as a *lit-sphere* [44]. This map was originally presented as a method for capturing shading styles from real 2D artwork, but Bruckner incorporates it into the transfer function lookup process to describe shading based on surface-normal orientation. The lit sphere (actually one hemisphere is sufficient) represents all possible surface normals and their shading via illumination: this is mapped onto a two-dimensional circular projection of RGB (and possibly opacity) values which can be referenced by a simple texture lookup. As the lit-sphere map can effectively represent an overall style of shading, it is a natural fit into the style-transfer function as a means to specify non-photorealistic shading styles.[§] Figures (2.7) and (2.8) provide some examples of the various rendering styles that can be produced using the style transfer function method.

A given style-transfer function may have more than one lit sphere associated with it, in the same manner as a regular transfer function specifies different intensity/opacity control-points; transfer-function lookup requires an additional lit-sphere lookup phase . In addition, it is also possible to interpolate between different lit-sphere maps.

2.6.2 Transparency

An additional possibility is the modulation of opacity with view-direction; this simply requires to sphere-map to include opacity in the lookup-data, i.e. RGBA instead of just merely RGB intensities. See [5] for details of this, and implementation specifics.

2.6.3 Possible improvements

Bruckner specifies that a separation of texturing and lighting operations might be preferable to the single combined spheremap model and may form the basis for future work.

[§]For reference, the lit sphere is also used in 3D modelling applications (such as *3ds Max* in their materials editor).

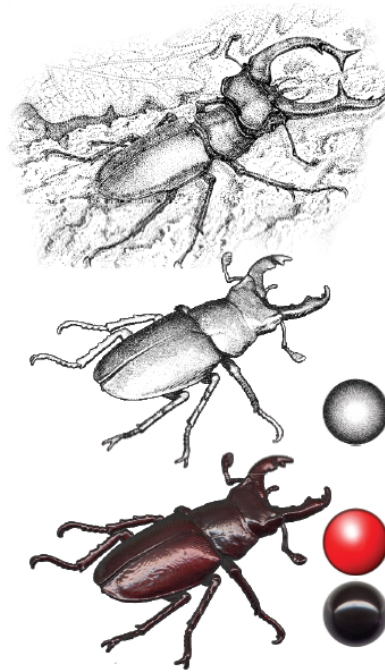


Figure 2.7: Top: pencil illustration of beetle. Middle and Bottom: illustrative render via two different style-transfer functions. The bottom one has two lit-spheres associated with it. Source: Bruckner et al. (2007) [5].

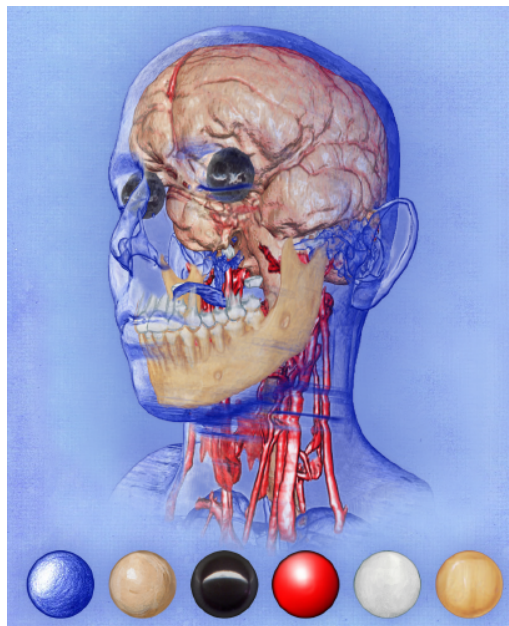


Figure 2.8: Illustrative volume rendering of segmented head dataset via style transfer functions. Source: Bruckner et al. (2007) [5].

2.7 Feature enhancement: more on contours, lines and shading based on surface differential properties



Figure 2.9: Feature-enhancing shading: L-R: contour outlines and ridges and valley shading using curvature-based transfer functions for volumetric CT dataset (source: Kindlmann et al.[3]); curvature-based deferred shading (source: Hadwiger et al.[45]); Suggestive contours and principal highlight lines on a polygonal mesh dataset (source: DeCarlo et al.[46]).

There has been a large volume of research devoted to shape depiction or enhancement, using techniques from traditional illustration such as line-drawing or supplementary shading to enhance the depiction of edge-boundaries (silhouettes) and contour lines (see fig.(2.9)). Gooch [39] observes that line drawing is a significant recurring characteristic in illustration and develops a technical-illustration system [47] where lines of both types (respectively drawn as black and white) supplement an overall shading scheme designed for improved shape depiction.

Hertzmann [48] attempts to frame the artistic process in scientific terms in order to address the fundamental questions of how artists actually create imagery and how observers respond to it. Cole et al. [49] present a study of where artists draw lines in order to examine the correlation with others; moreover they compare these results with then-current computer-graphics methods. In a later study, Cole et al. [50] further evaluate the efficacy of line drawings with regard to shape depiction.

Ridges and valley lines [¶] [51][52][53] are examples of crease lines that can delineate feature boundaries and thus enhance feature-depiction. In a 3D context, they depict isosurface ridge-peaks and valley-troughs which often form region-boundaries on a given isosurface; they can thus provide supplementary shading information to enhance the sense of overall shape. Belyaev et. al. discuss ridges and valley lines on implicit surfaces in [54]; although their work is developed within the context of polygonal models, the fundamental concepts are also applicable to isosurfaces embedded within a volumetric dataset. Eberly presents a comprehensive and highly-theoretical work on ridges in the book [55].

[¶]a precise definition is given in §4.4.2

Further developments include *apparent ridges* [56], *highlight lines* [46], *suggestive contours* [57][58] and *demarcating curves* [59]. Each of these methods presents a different set of metrics for producing their respective shape-enhancing lines and their results can be considered complementary to each other. Again, these methods are mainly described in the context of 3D polygonal meshes but can be adapted for application in volume rendering contexts. Indeed, Burns et al. [60] discuss line drawing within the context of volumetric datasets.

Essentially, these supplementary lines are indicative of some local differential property on the given isosurface: they may depict changes in normal direction or extent at which the direction changes, i.e. surface curvature. As previously noted, local lighting models implicitly depict the change in normal direction, but Gooch specifically incorporates this into the shading model described in [47]. Whilst contour lines are essentially a function of surface normal and view direction, second-order differential properties (curvature) of embedded isosurfaces form the basis for crease lines and the various other supplementary line methods discussed above.

Furthermore, Kindlmann et al. [61] make use of curvature to produce a shading of ridge and valley regions, i.e. not just the abstracted lines that form their peaks and troughs; supplementary shading is achieved through the use of *curvature transfer functions*, essentially two-dimensional lookup textures which map curvature values to colours. Hadwiger et al. [45] further discuss such curvature-based ridge and valley shading in the context of *deferred shading* for increased performance (an image-space process). In addition to their presentation of demarcating curves in [59], Kolomenkin et al. add supplementary curvature-based shading based on the metric of mean curvature.

Additionally (as previously mentioned in §2.5.1), Kindlmann et. al [61] use the isosurface curvature along the view direction to regulate the thickness of contour lines; this is an example of *view-dependent* curvature-modulated shading. Goodwin et al. [62] discuss artistic stroke thickness and its specification via *isophote distance*.

2.8 Hybrid rendering methods, context and applications

Hybrid rendering methods can combine illustrative and non-illustrative local shading or global illumination, feature-enhancing lines, volume rendering and polygonal models etc. Tietjen et al. [63] discuss a hybrid approach for surgery education and planning. A real-time illustrative rendering method for vascular structures is presented by Ritter et al. in [64]. There is considerable potential for hybrid rendering systems for the highlight and display of vessels against a background of less-salient context; the later can be rendered in a non-illustrative or contrasting style. Bruckner [65] outlines a context-preserving illustrative system for the presentation of salient features and structures whilst retaining

overall context and further discusses this general theme in [66]. Bruckner et al. further discuss hybrid methodologies for scientific illustration in [67]. Illustrative rendering systems clearly have numerous applications within medical visualisation wherever it is required to weight the presentation of salient feature detail or structures over less important background context.

2.9 Noise reduction (filtering)

The simplest noise-reduction methods or *filters* consist of a series of independent local averaging operations performed over respective neighbourhoods defined around each data point. The Gaussian filter is a linear convolution operation in which, for each pixel or voxel and its associated neighbourhood, the intensity values are weighted according to a Gaussian function of distance from the centre pixel or voxel. This is an *isotropic* operation in which smoothing is performed equally in all directions thus compromising (by blurring) the integrity of edges and feature-boundaries. It is highly effective at reducing noise, but feature-detail is also lost; *feature-preserving* methods attempt to strike an optimal balance between noise suppression and feature retention.

Two major categories of smoothing methods are the aforementioned neighbourhood or *local averaging filters* and *PDE-based diffusion models*. The simple Gaussian filter can be shown to be equivalent to the simplest case of linear isotropic diffusion as described by Laplace’s equation. Indeed, there are often links between these two methodologies (and others) as will be described in more detail in the forthcoming sub-sections.

Other methodologies exist, such as non-local means, wavelet-based, frequency-space, statistical and machine-learning-based methods. Indeed, it is possible to classify differing approaches to data-smoothing in a number of other ways: linear vs nonlinear, isotropic vs anisotropic, spectral methods vs spatial, local vs nonlocal etc.

One of the main objectives of this project is to develop a new method for the feature-preserving denoising of 3D datasets, with particular application to high-noise^{||} modalities such as 3D ultrasound. This will form a crucial component in a pipeline of operations designed to enhance feature detail in 3D ultrasound and other modalities with minimal noise and derived artifacts.

2.9.1 Neighbourhood filters

It is possible to formulate feature-preserving non-linear neighbourhood methods: the *bilateral filter* (as described by Tomasi and Manduchi [7]) is arguably the most famous

^{||}Ultrasound *speckle* is a quasi-random and noise-like signal degradation henceforth referred to as simply *noise*.

example. Building on previous work by Yaroslavsky [68], the bilateral filter is essentially a modified (from isotropic) neighbourhood averaging filter which preserves edge regions by penalising the influence of pixels (and voxels in 3D) within the smoothing kernel neighbourhood with a large intensity gradient to (or *photometric distance* from) each neighbourhood centre-pixel (or voxel); the effect is to reduce smoothing across regions of high gradient. The most common form takes the form of the standard Gaussian filter with an additional *range component* penalizer formed by a Gaussian function of photometric distances. Initially described in a 2D context, the bilateral filter in its general form can easily be extended to volumetric datasets. It was primarily intended to be a one-iteration filter, trading a very large kernel radius for low iteration count; this may be problematic for performance reasons however due to the $\Theta(N^6)$ computational complexity (where N is the length of one side in a cuboid volume) and practical implementations often truncate the kernel size.

There have been numerous attempts to improve the performance of such filters based on various approximation formulations. Since such neighbourhood filters are nonlinear, they are also fundamentally non-separable; hence a basic optimisation to reduce complexity is not possible. It should be noted however that separable approximations do exist [69], but these are not without significant artifacts and deemed by this author to be unsuitable for the purpose of the applications described within.

An early attempt at improving performance is described by Durand and Dorsey in the context of high dynamic range imaging [70] and utilises fast Fourier transforms together with subsampling. In a later work, Paris and Durand [71] make use of the representation of a 2D image $I(x, y)$ as a 3D surface $f(x, y, I(x, y))^{**}$ together with subsampling to formulate an approximation to the bilateral filter; in this approach, the inherently nonlinear bilateral filter is partially linearised with the additional nonlinear components considered separately. This is expanded on by Chen, Paris and Durand in [73].

Other fast approximations have been described by Weiss (based on methods for optimisation of the median filter) [74] and Porikli [75]; the latter presents a method independent of filter size (constant-time with respect to filter size). Building on this, Chaudhury et. al. [76] describe a constant-time (with respect to filter size) method using trigonometric approximations to the Gaussian components. Most of the above literature on the bilateral filter refers to a 2D context; extending to 3D is trivial [77] yet incurs a large performance penalty in its original form (due to being nonlinear and non-separable with the increased scaling factor of the extra dimension).

In addition to time-performance improvements, various methods have attempted to increase the quality of feature-preservation and noise suppression in neighbourhood filters, with particular application to ridges and valley regions [69] in the context of 3D datasets.

^{**}Note that surfaces embedded in \mathbb{R}^3 can also be represented as hypersurfaces in \mathbb{R}^4 [72]

The *trilateral filter* as proposed by Choudhury et al. [78] makes use of a pair of bilateral filter kernels: the first is applied to the image or dataset gradients; the second kernel, applied to the intensity values, is first tilted or skewed under an affine transformation. This filter demonstrates improved efficacy in ridge and valley regions over the standard bilateral filter in 3D contexts but has an increased computational cost [69]. The volumetric trilateral filter results in a significant reduction in *shock* or *staircase* artifacts commonly observed through use of the bilateral filter. Note that a similarly named but unrelated method was previously developed by Wong et al. [79].

Buades et al. [80] documented the tendency of the bilateral filter to form staircase artifacts by demonstrating an equivalence with a PDE-based model and performing a subsequent analysis; their solution proposed an intermediate regression correction step to reduce the severity of such artifacts.

Indeed, there has been a large body of work devoted to linking the bilateral filter (or neighbourhood filters in general) with PDE-based diffusion models. The simplest case is the equivalence of the Gaussian filter to the basic isotropic diffusion process: in this case it can be observed that the Gaussian function is the Green's function of the simple isotropic diffusion PDE. Several authors have deduced equivalences between the bilateral filter and Perona Malik diffusion process; moreover they have sought to exploit these in order to formulate an improved neighbourhood filter equivalent to the PDE in question. In addition to the work of Buades et al.[80], Elad is another such example [81] as is Barash [82] and [83].

Various parallels and equivalences between neighbourhood filters and PDEs have been studied; Buades discusses this again in [84]. The bilateral filter as a specific example has been variously described as a summation of Perona-Malik diffusion operations at different scales [69][85], analogous to robust statistical processes [70] and local mode filtering [69].

In a more recent development, the *guided filter* (He et al. [86]) attempts to improve on the bilateral filter by using an intermediate guiding image: as with the trilateral filter, it aims to reduce the occurrence of backward-diffusion based shock artifacts.

2.9.2 PDE-based filters

There have been numerous attempts to improve on Perona-Malik diffusion as a means to provide effective volumetric smoothing with a greater degree of feature preservation: to obtain positive results for both these criteria,

Weickert has performed numerous studies in the field of nonlinear diffusion; his book [87] provides a thorough grounding in its theoretical aspects. Earlier work includes a theoretical outline of the nonlinear diffusion process [88], expanded on in [87]. The then-current review of methods [89] provides an early example of the application of nonlinear

diffusion to 3D ultrasound, in an area of research mainly concerned with 2D image processing. Whilst Weickert tends to focus on diffusion tensor-based anisotropic diffusion (in its strictest sense), others have developed nonlinear diffusion models based on a simple decomposition of diffusion into orthogonal components, primarily using 2nd order differential properties such as curvature. Carmona et al. develop such a system in [10]; this provides the basis for an extension to 3D systems as described by Krissian et al. in [9] and further developed by Hossain and Möller in [8]. Curvature-based methods generally attempt to preserve feature detail and tubular structures by smoothing in the direction of minimum principal curvature (by magnitude) to smooth along feature boundaries and vessels, whilst limiting diffusion across them. Alvarez et al. [90] provide an important early framework for feature-preserving nonlinear smoothing on isosurfaces, and discuss the equivalence with the geometric motion of level sets using mean curvature.

The simplest curvature-based volumetric-smoothing PDE (and somewhat analogous to isotropic Gaussian filtering, but not equivalent) is isotropic* *mean curvature motion*. There have been numerous studies regarding the properties of the underlying PDE and its motion, both in the 2D context (where it forms the basis for *active contours* and *self-snakes* [91] with applications in segmentation in addition to filtering) and 3D, where the motion describes that of 2D manifolds embedded in 3D volumes; for a theory-intense early example of the mean curvature motion of surfaces, see [92]. Whitaker [93] provides a modification to the mean curvature motion PDE to perform smoothing in the direction of least curvature; moreover, he provides an early example of application to 3D ultrasound. This method (albeit different in formulation) provides similar results to PDE methods developed through consideration of orthogonal components of diffusion such as [8]. The latter method provides a PDE similar in form to that of mean curvature motion, but with specific anisotropy-dependent coefficients for the orthogonal curvature metrics (and hence orthogonal tangent-plane diffusion components).

Generally, the Eulerian formulation of the above PDEs [94] in the context of 3D volumes are examples of what Osher and Sethian describe as *level-set methods* [95] (for a general overview, see [96]); level sets in the context of volumetric datasets constitute embedded isosurfaces within the volume. Mean curvature motion in a 3D dataset context describes the motion of *all* level sets (isosurfaces) within the volume [97][98]: the direction of motion is in the direction of the gradient[†]. By considering how the curvature itself evolves, it can be shown that mean curvature motion is analogous to a reaction-diffusion equation with an isotropic diffusion component in the tangent plane in addition to the orthogonal movement of its level-sets (the reaction component) [99][100]; MCM is thus not volume-preserving, although variations do exist that are, such as that described by Nemitz et al. in [101]. For a comprehensive introduction and overview of geometric PDE methods, see

*isotropic in the tangent plane

[†]using the sign convention maintained throughout this document: see §4.4.

[102].

Geometric-flow based on level set motion of embedded isosurfaces and curvature-dependent tangent-plane isotropic diffusion are two possible interpretations of essentially the same given PDE [103].[‡] Moreover, the Laplace-Beltrami operator applied to an isosurface and the latter's mean curvature vector can be shown to be equal [104]. Note that the anisotropic diffusion PDE developed in [8] can be expressed as a modified version of the standard MCM formulation, where anisotropy controls the ratio of tangent-plane diffusion according to each principal curvature.

Curvature-based methods which switch the overall flow-type dependent on local differential properties may provide an improvement in noise-reduction capability over standard fixed-metric curvature flows: Sethian et al. describe a min/max curvature flow technique in [100] with application to 3D datasets.

Curvature-based methods such as MCM or min/max curvature-flow are termed *second order* geometric flow PDEs. Fourth order methods are discussed in [105]; furthermore, a volumetric equivalent of 2D Perona-Malik diffusion with its attendant edge-sharpening (due to backwards diffusion and shock formation) is proposed by Tasdizen et al [106] as a fourth order geometric evolution process. Fourth order methods are further discussed in [107], [108] and [109]. In addition, Didas and Weickert derive a curvature-based smoothing method with additional edge enhancement in [110].

There have been numerous attempts to develop anisotropic curvature-based geometric evolution processes to preserve features whilst denoising volume data, such that those described by Clarenz et al [104], whose method may also provide a degree of feature enhancement. Preusser and Rumpf derive their morphological multiscale method linking anisotropic diffusion and geometric flow in [111] by proposing an anisotropic conductance tensor. Clarenz et al. discuss the development of a variational approach to geometric surface diffusion in [112]. Nemitz et al. [101] discuss the preservation of vessel-like structures under volumetric smoothing and develop a volume-preserving anisotropic curvature motion in this regard.

2.9.3 Other methods for feature-preserving smoothing

Variational methods, where the system is framed as a minimization problem of some energy functional, were famously introduced in the context of image processing by Rudin, Osher and Fatemi in 1992 [113]. Variational methods provide alternative perspectives to the standard PDE evolution model and have been shown to be equivalent. Bertalmio et al.

[‡]The simplest PDE formulation (see eq.(5.35)) for curvature-based isotropic diffusion in the tangent-plane and the mean curvature motion PDE are the same equation. The geometric interpretation explains the normal-wise volume-reducing contraction in tangent-plane isotropic diffusion schemes.

outline a variational framework for PDE problems on implicit surfaces, whilst Chambolle [114] presents an general algorithm for total variational minimization with applications in image denoising; furthermore, the same author also discusses mean curvature motion for general hypersurfaces in a related paper [115]. For an outline of a variational approach in the context of Eulerian geometry processing, see [116]. Memoli et al. [117] present a detailed exposition of a variational framework as applicable to general target manifolds whilst Oberman et al. [118] formulate a method for anisotropic mean curvature flow using variational principals. Overall, there exists a substantial amount of literature on such methods but [119] presents a coherent introduction and overview.

The method of *non-local means* [120] attempts to exploit redundancy within the dataset or image: it considers similar subregions or *patches* throughout the image or dataset when computing average values at any given data-point (pixel or voxel), and assigns weights accordingly. The filtered value at each data-point is an average of the values of all data-points in the image or dataset within similar patches; overall this essentially constitutes a weighted average of every data-point in the image [121]. Similarity between patches is measured using a Gaussian-weighted L_2 norm. Non-local means can be accelerated for greater performance using a *blockwise* approach [122].

In a different approach, machine learning methods such as *codebook substitution* and *computational neural networks* can be used to explicitly identify specific features and are primarily used in classification or segmentation algorithms. However, segmentation of features can be used in conjunction with the smoothing process in order to limit or constrain any filtering within identifiable regions. Moreover, computational neural networks can be trained to effectively reproduce the flow inherent in PDE-based or other filtering methods, but this requires initial ground-truth obtained through that original smoothing process itself.

2.9.4 Application to 3D ultrasound

Speckle [123] is a quasi-random (noise-like) signal degradation introduced into ultrasound images at the data acquisition stage; this noise-like phenomena (henceforth referred to as simply noise) is essentially the result of interference patterns induced via the interaction of ultrasonic pulses with tissue structures. There are numerous publications devoted to the development of filtering methods for the reduction of speckle, with particular attention given to feature-preservation. Directional median filtering is outlined in [124]; statistical-based wavelet methods such as that described by Gupta et al. in [125] have also been explored.

Nonlinear anisotropic diffusion methods such as SRAD (*Speckle-Reducing Anisotropic Diffusion*) [126]. The method proposed by Abd-Elmoniem et al. (which they term *Non-*

Linear Coherent Diffusion (NCD)) [127] involves a transition of models from mean curvature motion to linear diffusion to anisotropic diffusion dependent on local surface differential properties. The formula for the mean curvature motion phase is derived via the local structure matrix as opposed to the Hessian (see eq.(5.11)) but results in the same familiar expression to specify diffusion along isophotes or isosurfaces. Krissian et al. [128] develop the anisotropic diffusion model further: OSRAD (*Oriented Speckle-Reducing Anisotropic Diffusion*) features principal-curvature direction based diffusion for improved feature-preservation.

Non-local means has also been investigated by Coupé et al. [129] as an effective approach for speckle reduction using a Bayesian formulation [130] more suitable for the structure of speckle-noise. This remains a popular and effective method in the context of ultrasound and is the basis for more recent development [131]. Other newly-developed (2016) speckle-reduction algorithms include a feature-enhancing method based on multi-scale analysis [132].

2.10 Conclusion

Recall the three fundamental criteria from §1.2: (i) the highlighting of salient structures through illustrative shading (at primary and/or supplementary shading stages); (ii) effective noise suppression to permit this with minimal amplification of noise-based artifacts; (iii) a fast overall execution time allowing realtime streaming 4D ultrasound processing. Hence the fundamental overarching objective is to design, develop and construct a flexible rendering and filtering pipeline capable of illustrative and other styles as required, with additional feature-enhancing shading, coupled with a volumetric filtering stage that can minimise unwanted noise whilst maximising feature detail retention. A state-of-the-art volume filtering stage is necessary to mitigate potential noise-highlighting from the feature-enhancing shading step.

GPU (graphics processing units) map well to the simple parallel processing tasks involved in most rendering and filtering operations, i.e. these applications have no potential inter-voxel write-clash issues, dependencies or other synchronisation considerations which can complicate implementation and reduce performance. GPU-based ray casting will be used as a basis for the overall rendering method: current GPU technology is capable of easily providing the performance required for real-time raycasting-based renderers. Raycasting is preferred over explicit surface construction and shear-warp renderers as it is simpler, easier to implement, more flexible and less prone to artifacts and distortion. Simple tissue 'classification' (i.e. without segmentation) can be achieved through the use of standard transfer functions to map intensities to colour and opacity values, which are then integrated along each ray. Note that segmentation can be added if required, but is supplemen-

tary to the objectives outlined for this project. GPUs will also be used for all per-voxel or per-pixel computations in all filtering stages (both volumetric and image-plane) for high performance.

2.10.1 Base renderer

Bruckner’s Style Transfer Functions method constitutes an ideal candidate for the base illustrative rendering system. Shading styles based on first-order differential properties (i.e. dependent on gradient or surface-normal) can be easily constructed and interchanged; moreover, appropriate styles can be applied to different tissue types via a small extension of the standard transfer function approach. The style transfer function method is thus highly configurable and can provide a simple yet comprehensive method for specifying normal-based shading styles, including non-photorealistic ones.

Other shading methods to be considered are those which employ curvature-dependent transfer functions; these depend on second-order differential properties and can be specified by a variety of curvature metrics, dependent on application: i.e. ridge and valley highlighting. Curvature-based streamline methods are not considered here for implementation but may be investigated in future work.

2.10.2 Supplementary shading for feature enhancement

Curvature-based shading (i.e. using second-order differentials properties) with a variety of metrics will be investigated as a primary candidate for a supplementary shading method for the enhancement of feature-boundaries and general shape depiction. Such shading method can perform the same role as crease lines and associated shading in traditional illustration: i.e. it can provide an increased emphasis of ridges, valleys and other regions as applicable. This technique has the potential to improve facial-feature delineation in ‘babyface’ ultrasound, both in illustrative and global-illumination contexts.

Deferred shading has some potential in the case of single isosurface rendering: originally specified for increased performance, these methods crucially allow for image-space post-processing operations. A potential expansion of this includes post-processing of auxiliary data buffers (specifically curvature data) produced by the deferred rendering process.

In addition, normal-based outline contour enhancing methods (i.e. those using first order differential properties) will also be considered as a means to improve shape-depiction.

2.10.3 Improved volumetric noise-reduction

The anisotropic diffusion PDE specified by Hossain and Möller [8] provides an ideal candidate for investigation and expansion. This PDE is simple in formulation, easy to implement with simple explicit numerical schemes and is stable for reasonably-sized time-steps with such methods (i.e. does not require a large iteration count and very small timestep). Moreover, it is a curvature-based PDE, which can make use of previously computed curvature values, allowing for improved performance or a high degree of code-reuse if required: curvature computation routines can be shared between shading and filtering operations. This PDE is formulated using a simple model: there is no explicit diffusion tensor; this can be considered in some sense an ‘expanded’ version of the more condensed tensor representation but is more suitable for implementation on GPU hardware.

Other PDE-based methods to be considered for comparison are mean curvature motion, Whitaker’s *motion by least curvature* using a normalised product of mean and Gaussian curvatures. Indeed the Hossain-Möller PDE has a specific relationship to mean-curvature motion which can form the basis for development of a new method. The simple formulation of the PDE allows for intuitive and efficient modifications (see §5.5).

Other methods for consideration are the bilateral filter and Perona-Malik diffusion, both for investigation and comparison; these methods are prevalent in current medical imaging systems. The bilateral filter has the disadvantage of larger computational complexity compared to PDE-methods (due to the much larger filter support radius) and hence is expected to perform poorly in its basic form.

2.10.4 Applications and datasets

The ‘engine’ dataset (a CT scan of a cylinder head) is often used in the literature, since it possesses numerous sub-regions which can show the effect of a given shading or filtering method on various geometric features, such as tubular components, ridges, valleys and edges, flat homogeneous regions and convex and concave structures; it will be used throughout this current document to demonstrate the application of various rendering and filtering methods and their effect on its various topological features.

In addition, development of all components in this project’s rendering and filtering pipeline is documented using an ultrasound ‘babyface’ dataset with attention being drawn to facial features and noise. This dataset is of particularly high fidelity, and thus can demonstrate the full potential of feature enhancing and delineation in 3D ultrasound; moreover it has several relatively flat regions (i.e. the forehead and cheeks) where noise can be clearly observed. The overall objective is a reduction of noise in these regions, whilst an enhancement of facial-feature delineation. Other datasets are of course used in addition,

and will feature more prevalently in §6, where the complete system is demonstrated and tested.

Vessel datasets (from a CT head-scan) will also be employed throughout to demonstrate illustrative shading, aneurysm-highlighting and vessel-preserving noise reduction; other CT datasets will also be tested.

The overall system will also be applied to 4D ultrasound heart-valve datasets in order to demonstrate the effectiveness of both the rendering and filtering stages for such data: the former to provide improved shape-depiction, and the latter to provide a feature-preserving fast volumetric filtering stage capable of the realtime processing of streaming ultrasound data.

2.10.5 Evaluation methods

For the rendering methods outlined above, relatively little user-evaluation has been performed in the original literature; it could be argued that this is not necessarily a negative trend: the literature tends to focus on producing specific effects or rendering styles rather than evaluating their effectiveness with regard to user perception. However, comparisons are usually presented with other methods. For specific user-centric research, such as *Where do people draw lines* [49], user evaluation is of course the central objective and features heavily in such papers.

In the context of the literature on novel rendering methods, it is essential to determine what the research is attempting to demonstrate: it is often a case of specifying clearly defined visual objectives and demonstrating them with comparisons; user perception can be used for additional qualitative assessment in these contexts, but a study of user *preference* is a fundamentally different question.

Noise-reduction methods can be subject to signal/noise analysis with artificially added noise of specific type and the results quantified for supposed efficacy (as is demonstrated throughout the literature), but these metrics do not always translate to what is presented visually. Moreover, it is difficult to obtain noise-free ground truth without applying noise-reduction: synthetic datasets suffer from aliasing artifacts produced by the voxelisation process[§]. Coupé et al. [133] discuss such methods in the context of non-local means; Hossain and Möller themselves perform a small quantitative analysis using signal to noise ratios in [8]. However, this project will focus on visual results obtained in conjunction with the primary and supplementary shading steps.

In the case of an overall filtering and rendering system, it may be prudent to collate images for comparison together with a clear and concise set of criteria defined as to what the

[§]For an aside on quantitative methods and their limitations in the context of this research project, see §7.3.3.

comparison is attempting to demonstrate. For example: image B attempts to demonstrate improved feature delineation with reduced noise compared to image A. Taking cues from the current literature, this would usually be presented without user evaluation (since we are not asking if the user *prefers* A or B); however, an additional preference study may be of interest, depending on research objectives.

2.10.6 Summary

The following describes an outline of the structure of the proposed rendering and filtering system, plus the candidate methods that will be investigated as potential components of the system, and potential candidates on which to base development of new methods:

- GPU implementation of all per-pixel and per-voxel rendering and filtering operations
- Raycasting-based rendering.
- Renderer: Style-Transfer Functions for illustrative and other applications, curvature-based transfer functions.
- Supplementary feature-enhancing shading candidates: normal-based, curvature-based, deferred rendering.
- Volumetric noise reduction: PDE based: Hossain-Möller anisotropic diffusion, mean curvature motion, motion by least curvature, Perona-Malik diffusion; Neighbourhood-based: bilateral filter, Gaussian filter.
- Data samples and derivative samples (for gradient and curvature) will be obtained through both central differences and cubic splines for testing, using [17] for the latter.

The system will be developed using datasets from the following sources: ultrasound ‘babyface’, CT-head-scan vessels, 4D streaming ultrasound heart-valve, the ‘engine’ dataset and various others from both the CT and ultrasound modalities. The complete system will be further tested with additional datasets.

Results will be presented for visual comparison with other methods or with various stages of the system disabled; in addition, a small user evaluation will also be employed to ascertain if consensus regarding the stated criteria is met.

Chapter 3

A base illustrative rendering system

The intrinsic essence of any visualisation system is of course the renderer itself: it defines the overall presentation of the selected data under study. The primary objective of this project is to develop a system for the depiction of medical imaging data within a primarily illustrative framework, with the potential to highlight or enhance salient structures or feature detail. The rendering components must satisfy the stated requirements (1) and (3) within the scope outlined in §1.2: i.e. in addition to these visual criteria, any rendering components must also be time-performance oriented.

GPU hardware is ideally suited for high-performance massively data-parallel tasks such as rendering. Such devices are not limited to the functionality of a traditional graphics-library (e.g. OpenGL or DirectX) and complex rendering operations can be more easily performed using a general-purpose compute library such as CUDA (for Nvidia hardware) or OpenCL. The final output from these routines can be displayed onscreen via an interaction with the desired graphics API through the use of shared buffers or textures.

A ideal base rendering system for the purposes described above should possess the following attributes:

- a raycasting-based volume renderer
- high performance GPU implementation
- flexibility: encapsulating a range of shading styles which should be easily interchangeable
- tissue-specific shading via a modified standard transfer function (but extendible to segmentation-based systems)

Most of the illustrative shading methods discussed in the previous chapter can be subdivided into two categories: *normal*-based and *curvature*-based methods. The former require significantly less computation than the latter so will generally result in lower per-

frame execution times. Complex global illumination approaches could theoretically be used for illustrative purposes but are computationally expensive.

Illustrative stippling and hatching effects are generally based on isosurface normals and curvature respectively (see Baer et al. [134]); there are several implementations of each of these styles: Lu et al. [42] and Baer et al both describe methods for applying illustrative stippling whilst Nagy et. al [41], Zander et al. [135], Gasteiger et al. [136] and Gerl and Isenberg [137] all present methods for the emulation of artistic hatching.

Bruckner’s *style transfer function* method [138] is a normal-based shading system for volume rendering applications, which employs pre-rendered lighting or shading maps in combination with a transfer-function approach for mapping colour and opacity values to dataset intensity ranges. Such maps [44] can be designed to represent any normal-based shading style including contour lines, stippling, paint strokes and simple local lighting; it is thus highly flexible and is thus preferred as a base system over more specific methods. Furthermore, it uses isosurface normals (or gradients) to reference the precomputed shading maps and thus is computationally light. Crucially, it satisfies all of the required criteria outlined above and thus will form the basis for the base rendering system.

Other methods, such as *curvature-based transfer functions* [3][45] will also be integrated into the system as alternative renderers: this will be discussed further in the context of the complete system in §6.

3.1 Style Transfer Functions

A significant development in the field of volumetric illustrative rendering is attributed to Bruckner and Gröller: the 2007 paper titled *Style Transfer Functions for Illustrative Volume Rendering* [5], together with Bruckner’s PhD thesis [138], describe a shading method highly suited for non-photorealistic applications. Shading models are specified via 2D normal-lookup maps and are easily interchangeable; the technique is essentially a local-lighting model as described in Sloan, Gooch et al *The Lit Sphere* [44] adapted for transfer-function-based volumetric applications.

3.1.1 Spherical lighting maps

Traditional illustrators often develop shading styles on a two-dimensional representation of a simple sphere before transposing to a more complex object [139]; in this way, all possible local-surface orientations (i.e. viewer-facing surface normal directions) are represented. This *lit sphere* can thus define local shading and lighting styles that can be subsequently implemented on more complex models (see fig.3.1); conversely, the shading styles on complex models can be encapsulated on a lit sphere. Sloan, Martin, Gooch and Gooch [44] describe such a process, in which illustrative shading styles can be captured from sampled two-dimensional artwork; these styles can subsequently be applied to computer graphics 3D models or structures within a volumetric rendering context.

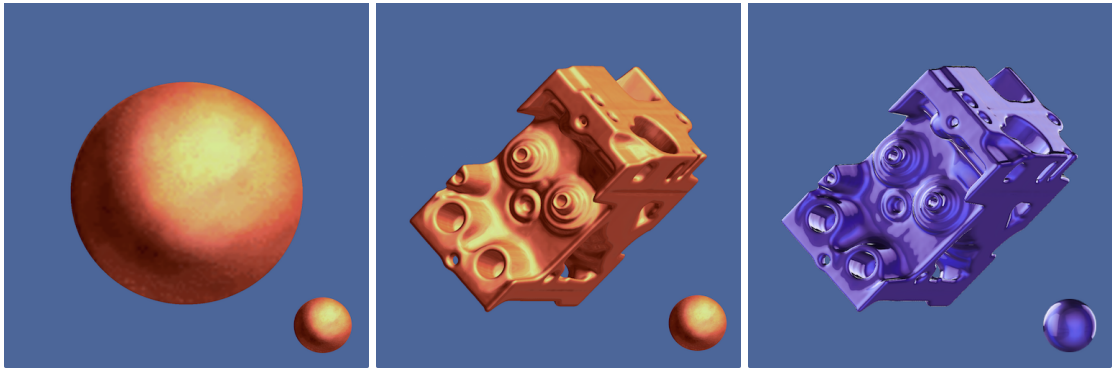


Figure 3.1: Shading of volumetric models using the specific lit spheres or *spheremaps* depicted (inset). L-R: shading applied to spherical isosurface; the same shading applied to more complex isosurface (engine volumetric dataset); substituting shading styles with a different spheremap. These particular renderings were produced using the software developed throughout this EngD project (Paul Ross) using the Style Transfer Function approach described by Bruckner and Gröller [5].

The *Style Transfer Function* method makes use of such spherical shading maps (referred to in [5] as *spheremaps*); essentially an inverse process to that described in *The Lit Sphere* is performed. Surface normals (under orthographic projection) are used as lookup coordinates of a particular spheremap texture (the specific texture is specified by the transfer function): this provides the local shading (and lighting) detail at a particular sample point. It is essentially the same process that is applied in *environment* or *reflection mapping* [140][141] to encapsulate complex ray-traced reflections and shading in a lookup texture; however, spheremap-based shading as an integral component of style transfer functions is primarily concerned with surface shading styles. The spheremap-based shading approach is an example of *image-based shading*.

Unit normals on a given isosurface can be parallel-translated to the centre of the unit sphere*; the orthographic projection of this unit sphere onto the image-plane constitutes the shading map known as a *lit sphere* or *spheremap*. The unit normals originating at

*note: parallel-translation of unit normals to the centre of the unit sphere is also used in the construction of the *Gauss Map* when considering the curvature of a surface or manifold; see §A.1.1 for details.

the centre of this sphere all correspond to points on the surface of the sphere itself, since the latter has unit radius. In order to reference the spheremap texture, the normals are themselves orthographically projected onto the image plane (by simply removing their z-coordinate); these projected normal vectors then define the texture lookup coordinates of the spheremap. Figure (3.2) depicts the various stages of this process.

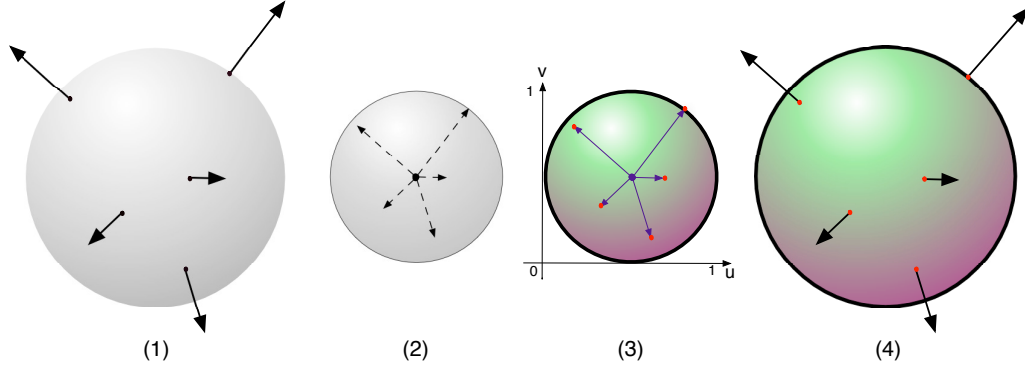


Figure 3.2: Shading process using spheremaps: (1) 3D object in eye-space with selected unit surface normals; (2) parallel translation of unit normals to centre of unit sphere; (3) orthographic projection of normals onto spheremap texture: the red points indicate the texels to be sampled for the given normals; (4) shaded 3D object. Illustration: Paul Ross.

Note that the projected sphere captures both local shading and lighting detail: these would appear to be equivalent, but shading may incorporate texture information. This is noted by Bruckner in [5] as a limitation; an extension to the method in order to address this issue is suggested as a possible future research topic. This is discussed further in §4.

3.1.2 Transfer functions

Within the context of volume rendering, the *transfer function* is a map from sampled data to colour and opacity values; it is usually a mapping from sample intensity values, but other metrics can be considered, e.g. in the case of *curvature transfer functions*. Using the standard transfer function, specific data sample ranges can be mapped to specific colour and opacity combinations, thus providing a simple means to classify and depict different structures or tissue types. The colours are specified at *control points* along a curve (often piecewise-linear) within a plane defined by opacity and data-intensity axes.

In the context of transfer functions, the spheremaps replace standard solid colours at the control points: a particular intensity region can utilise its own distinct spheremap, and thus different structures within the volume (identifiable by intensity region) can be shaded in an appropriate style. Moreover, interpolation between values derived from adjacent spheremaps in the transfer function curve is possible. Essentially, the style transfer function replaces the fixed colour at each control point with a normal-based shading process defined by referencing a pre-rendered texture. Figure (3.3) depicts the user-interface for

the style transfer function editor (with corresponding rendered output); this editor forms a core component of the EngD project volume-renderer. In this example, the spheremaps can be seen at the control points of the transfer-function; for this particular implementation, the transfer function is represented by a piecewise-linear curve, but it is not generally restricted to this form (e.g. a piecewise-cubic curve could be used).

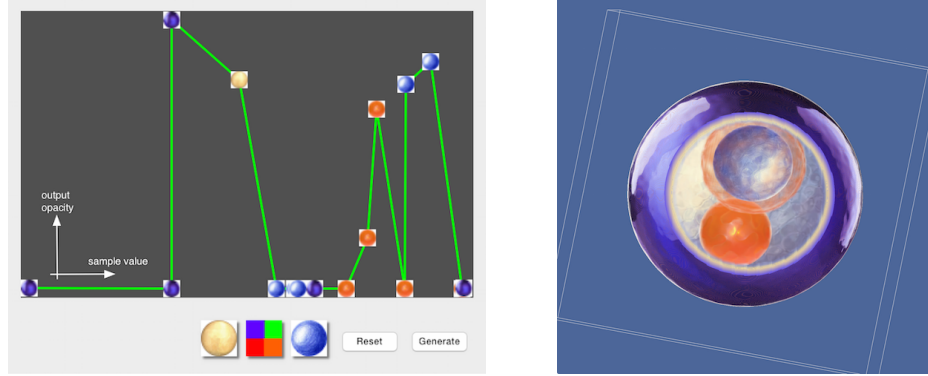


Figure 3.3: Detail of the software system developed throughout this thesis (Paul Ross): Style transfer function with associated spheremaps and corresponding rendered output of the ‘spheres’ synthetic test dataset, using some of the spheremaps from Bruckner and Gröller’s original paper [5]. Source: Paul Ross (EngD-project renderer).

3.2 Selective clipping

Cutaway views are a high level construct applied by technical illustrators to reduce or remove occluding surfaces so that salient structures beneath can be visualised in context. See [142] for a general overview of the concept (applied to polygonal models) and [143], [144] for a discussion within a volumetric rendering context. Such effect can be achieved by several means: from simple clipping planes to view-dependent opacity modulation [5] and deformable clipping membranes [145].

The base renderer for this project is augmented with a simple selective clipping plane function: clipping regions can be visually retained or clipped dependent on sample intensity value. Although clipping planes themselves are relatively crude with respect to more advanced methods such as *clipping membranes*, this selective approach allows clipping to occur within selected intensity ranges; hence the clipping can be structure or tissue-type specific. Moreover, there is a flexible control of opacity over any clipped region: it can be rendered totally transparent or with partial opacity for retained context if required. This overall clipping approach was deemed to be sufficient within the context of this research project but could be extended to incorporate more sophisticated methods for future work if required. Figure (3.4) demonstrates this renderer’s use of clipping, together with style transfer functions in order to isolate key salient features.

It should be noted that surface normals on clipped surfaces can present an issue (discussed

in [143]) with regard to artifacts. Clipped surfaces invariably form part of larger homogeneous regions and are only ‘clipped’ by setting opacity values to zero. Hence gradients on such surfaces are very small and usually do not correspond to normal vectors on the clipped surface. The clipped section is merely rendered as transparent, but data values can have the same intensity sample values as those within the non-clipped region on the other side of the clipping plane. Hence, isosurface normals on clipped surfaces are set to the clip-plane normal to avoid artifacts from near-zero gradients.

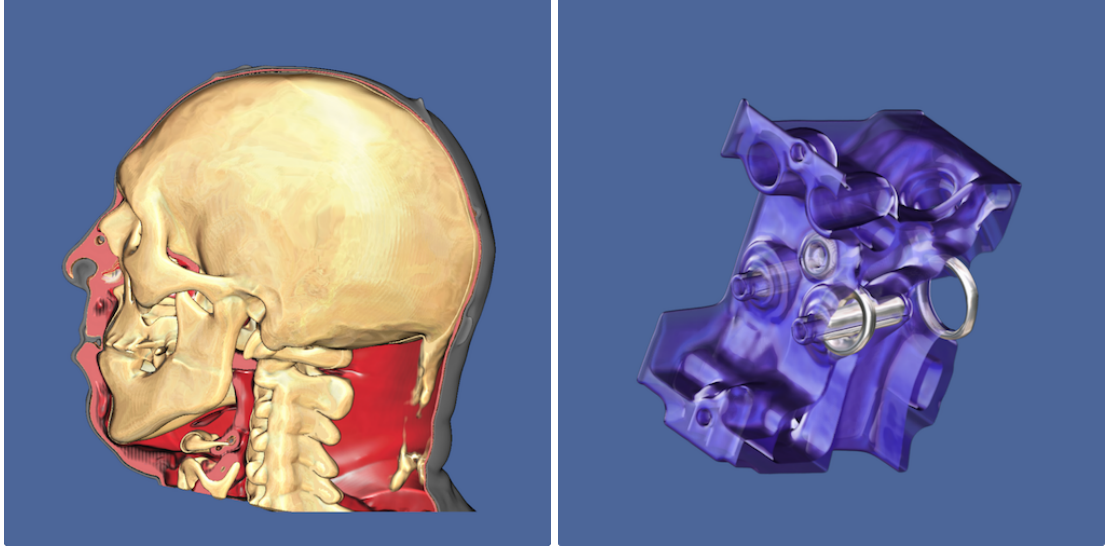


Figure 3.4: Volumetric rendering of human head and engine datasets (both CT) using style transfer functions with selective clipping planes for the depiction of salient features in context. Source: Paul Ross (EngD-project renderer).

3.3 Other shading methods

3.3.1 Curvature-based transfer functions

Curvature-based transfer functions are discussed by Kindlmann et al. [3] and Hadwiger et al. [45] as an alternative means of shading isosurfaces or structures within a volume-rendering context. Rather than mapping intensity values (and isosurface normals in the case of style transfer functions) to output colours and opacity, the rendered pixel is computed as some function of curvature by means of a (usually) two-dimensional lookup texture; note that this texture itself is termed the ‘transfer function’ by Kindlmann et al.

In addition, supplementary shading methods (e.g. deferred curvature shading) can be used to enhance the primary results produced by shading maps and style or curvature transfer functions: these will be discussed in §4.

3.3.2 Combined transfer function

Kindlmann’s concept of transfer function can be considered analogous to the shading map (spheremap) itself in Bruckner’s style transfer function method rather than the mapping between the overall range of intensities to output colour and opacity values in the latter method. However, such a curvature-based ‘transfer function’ or mapping could be substituted in place of a shading map within Bruckner’s overall transfer function. The result is a curvature-based shading map (rather than normal based), applied over a range of sampled intensities; i.e. specific tissue types or structures can be shaded according to their curvature within the context of an overall style transfer function.

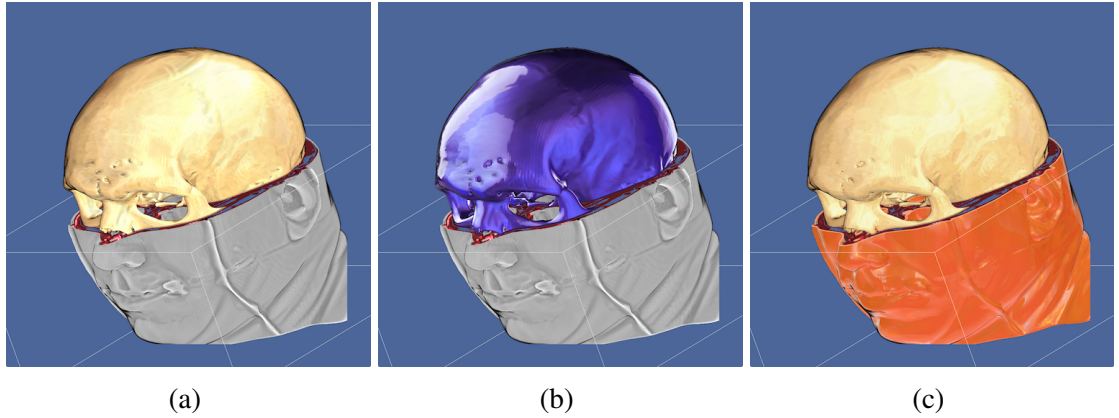


Figure 3.5: Rendering the ‘head’ dataset using combined transfer functions: (a), (b): style transfer function (spheremap) shading of skull in conjunction with MPR (note: MPR shading is enhanced with supplementary curvature shading, see §4); (c): style transfer function shading of both skull and skin. Source: Paul Ross

Such a transfer function can also specify other rendering methods such as *global illumination* for given structures defined by intensity range, and thus allows for complex hybrid rendering systems to be created. Segmentation would be ultimately preferable to transfer-function classification, but this approach may have applications in the absence of pre-segmented data.

3.4 Additional shading

Bruckner [5] uses a ring around the spheremap to provide shading for normal-based contours: in eye-space, normal vectors at the transition point from forward to back-facing surfaces are coplanar (or close to being coplanar) with the image-plane, i.e. their view-axis component or z-component is close to zero. In the context of spheremaps, the orthographically projected normals in such a scenario are at a maximum length: hence, the region at the very edge of the spheremap corresponds to such cases. Bruckner proceeds to discuss regularisation of contour-thickness: whereas Kindlmann et al. had previously introduced

a computational method to achieve this, Bruckner’s method uses the spheremaps themselves by pushing the normal vectors close to the transition point up towards the edges of the spheremap and into the surrounding ‘ring’.

Since the computational performance of GPU hardware has greatly increased since Bruckner’s paper, it is now possible to perform such additional shading in real-time within the GPU compute-kernel without having a significant negative effect on overall framerate.

Within the project renderer, contours or other supplementary normal-based shading effects are created by modulating (usually darkening) the output colour after principal shading via spheremap lookup; this occurs at each iteration within the volumetric raycaster to allow for application within partial opacity contexts: a post-processing deferred approach would limit the application to fully opaque single-isosurface renders. There is no contour thickness regularisation[†], but some manual control is achieved though the use of a logarithmic formula to control the range over which the colour modulation occurs. Each colour channel (r,g,b) is reduced in intensity by the same percentage; this has the same effect as reducing the V-channel in an HSV colour-space by the same proportion and is preferred for performance over explicit HSV conversion.

For each colour channel (generically denoted c):

$$c_{out} = c_{in} \min\{\max\{1 + a \log(b|\mathbf{n} \cdot \mathbf{v}|), 0\}, 1\}; \quad (3.1)$$

where c_{in} and c_{out} represent the channel input and output values respectively, and a and b are parameters controlling the range over which the visual shading is applied: a and b control the sharpness of the transition and the width of the contour respectively. \mathbf{n} and \mathbf{v} are the unit normal and view vectors.



Figure 3.6: Supplementary shading in conjunction with style transfer function rendering of the foot. L-R: style transfer functions alone; alone; with additional curvature shading (see §4); with curvature shading and normal-based contours for context. Source: Paul Ross

[†] contour thickness can be regulated by considering the curvature of the isosurface along the view direction as discussed in [61] and [5]

3.5 Implementation

The style transfer function renderer forms a basis for the illustrative volume-rendering system developed over the course of this project. It provides an elegant and highly configurable means with which to produce various basic illustrative styles: these can be rapidly swapped for prototyping, and thus the required style from a pre-determined selection of spheremaps can be attained. This system has been extended to include support for curvature-based shading maps in addition to Bruckner’s normal-based spheremaps; in addition, traditional MPR rendering is provided as an option. It should be noted that all volumetric data is filtered to reduce noise: for the images produced in this chapter, the filtering consisted of dual stages of Gaussian and bilateral filtering. An improved volume filtering method forms the basis of the forthcoming discussion in §5.

To summarise, the base rendering system consists of:

- Raycasting-based illustrative renderer based on Bruckner’s style transfer functions
- Clipping planes with intensity-based selective clipping and adjustable opacity.
- Extension of Bruckner’s style transfer function to accommodate curvature transfer functions and MPR for hybrid rendering.
- Additional normal-based shading for contour and silhouette illustration. Curvature-based methods for crease lines will be discussed in the following chapter, §4.
- Optional post processing of output image data: edge-preserving smoothing (via Perona-Malik diffusion) can provide *toon-shaded* illustrative effects. Additional post-processing filters include special effects such as *bloom* and *gloom*[‡]. Such image-space post-processing operations won’t be considered until later chapters.

All per-pixel or per-voxel operations are implemented in CUDA for Nvidia GPU hardware: this includes the core raycasting operation and all data filtering and post-processing tasks. The entire application is constructed using C++ and the cross-platform Qt libraries, with CUDA for GPU-specific core rendering and filtering operations.

3.5.1 User interface

Early prototypes of the application were initially constructed using Objective-C and C++, together with Apple’s *Cocoa* framework on the macOS platform (Apple Macintosh hardware). This environment allows easy and rapid prototyping, particularly of UI elements. Subsequent versions were implemented using a cross-platform approach: C++ together

[‡]these effects filters are based on combining thresholded and blurred highlights with the original rendered output image

with the *Qt* toolkit, allowed development to proceed across multiple platforms (macOS, Windows). In addition, the use of Qt allows additional deployment or development on Linux or other Unix-based systems if required.

The main user interface element for controlling overall rendered appearance is the *transfer function editor* window (see fig. (3.3)). This presents a visual representation of the transfer function itself, including the control points for shading map specification. Other UI elements control noise-reduction and supplementary shading and will be discussed in §6, when the overall final system is described.

Shading maps are loaded into the system by dragging them to a ‘shelf’ within the transfer-function editor. The transfer-function curve itself (currently limited to a piecewise-linear form) is formed by clicking within the editing region to create individual control points. Each control point is automatically connected to the previous one by a straight line. Spheremaps can be allocated to control points by simply dragging them from the shelf to the required control point.

3.5.2 Renderer

The basic renderer is a *direct volume rendering* raycaster implemented as a GPGPU-based CUDA kernel. A GPU-based approach has numerous advantages in this context:

- The raycaster is highly parallelisable. Large performance gains can be achieved by making use of the multiple processing units resident on a GPU.
- Texture hardware in the GPU provides trilinear interpolation; tricubic sampling can be implemented using CUDA and a succession of trilinear samples [146]. Additionally, data-set boundaries are automatically clamped or zeroed by the texture unit if required. This again removes the requirement for explicit handling of these boundary cases.
- Easy integration and inter-operation with rendering libraries such as OpenGL.

The raycaster itself is not highly optimised: it features early ray termination for opaque isosurfaces and rudimentary single-pass empty space skipping; the latter could be improved with a multi-pass approach for future work to enhance performance.

Renderer output

The raycaster renders its colour output to an OpenGL texture, which is then displayed on screen either directly or after additional post-processing steps (post-processing will be discussed in later chapters). Additional polygon-domain rendering can be performed if required, e.g. boundary boxes, clipping planes. Furthermore, the raycaster produces

additional depth data: depth on ray termination is stored in a texture and can be used in any additional polygon-rendering stages.

Spheremap textures

In this implementation of style transfer functions, the unique shading maps (spheremaps) are stored in a *layered texture* and accessed according to layer number: the spheremap ID corresponds directly to its layer number within the texture. The use of layered textures was discussed in Bruckner’s implementation in [5] and was implemented here using the following rationale:

- texture references in early versions of CUDA are limited; using layered textures requires only one texture reference.
- grouping all spheremaps under one texture reference is self-contained and concise and intuitively translates Bruckner’s original concept to a CUDA texture context.
- having all spheremaps contained within one (small) texture avoids the possibility of individual spheremaps being repeatedly purged from and reloaded into texture cache.

3.6 Results and examples

3.6.1 Blinn-Phong and other procedural local lighting models

Procedural local lighting methods such as Blinn-Phong lighting can be easily emulated using a style transfer function with the appropriate spheremap; the spheremap itself is a 2D rendering of a 3D modelled sphere with applied Blinn-Phong lighting, and can be created using a number of possible 3D modelling tools. The spheremap encapsulates all required shading information: in this case, a red diffuse material and one specular highlight. Bruckner provides an example: this and several other examples from [5] will be used here for demonstration purposes.

3.6.2 Stippling

Illustrative stippling effects are essentially dependent on surface-normal direction (as opposed to *hatching*, which are substantially curvature-based). A stippling shading style can thus be implemented via the use of normal-based shading maps (spheremaps) as part of Bruckner’s style transfer function method. Indeed, Bruckner presents examples together with the associated spheremap in both [138] and [5].

Setting the transfer function to display only bone-structures and applying this map to the ‘head’ CT-dataset produced some mixed results (see fig.(3.7)). In certain regions, the shading effect is a convincing facsimile of a style commonly observed in traditional anatomical illustrations of the human skull; shading of the maxillofacial region is consistent and well-defined, particularly around the eye-sockets. However, significant shading artifacts can be observed on the dome of the skull. This is more pronounced at higher rendering resolutions (lower row, fig.(3.7)); furthermore, the artifacts become progressively worse with the degree of Gaussian volumetric smoothing performed on the dataset prior to the rendering stage.

Applying the same shading map to the ‘sphere’ dataset gives an insight as to why these artifacts occur at specific levels of smoothing. Figure (3.8) depicts a style-transfer-function rendering of this dataset with the stippling pattern applied; for comparison, the lower row uses a different shading map to show the changing scale of surface facets as the level of smoothing is increased. As smoothing increases, the remaining surface facets become larger in overall scale (smaller in magnitude) as higher frequencies are removed. At specific scales, the mapping of spheremap texels to the isosurface disrupts and distorts the shading pattern sufficiently to produce noticeable artifacts. Small scale bumps produce a larger degree of light-scattering, whereas larger facets that have a similar scale to the stipple pattern granularity on the spheremap result in a distorted pattern on the output image: shaded texels on the spheremap are stretched under mapping to the isosurface by the slight change in direction of normals on facet edges or ridges and valleys.

Hence, rough surfaces map well to this shading style, as do exceptionally smooth surfaces; partially smoothed isosurfaces may however have small facets or surface features that at a specific scale and rendering resolution, produce a distortion in the stipple pattern: in addition to surface smoothness (and hence scale of surface facets), this distortion is a function of both the spheremap pattern granularity and the rendering resolution.

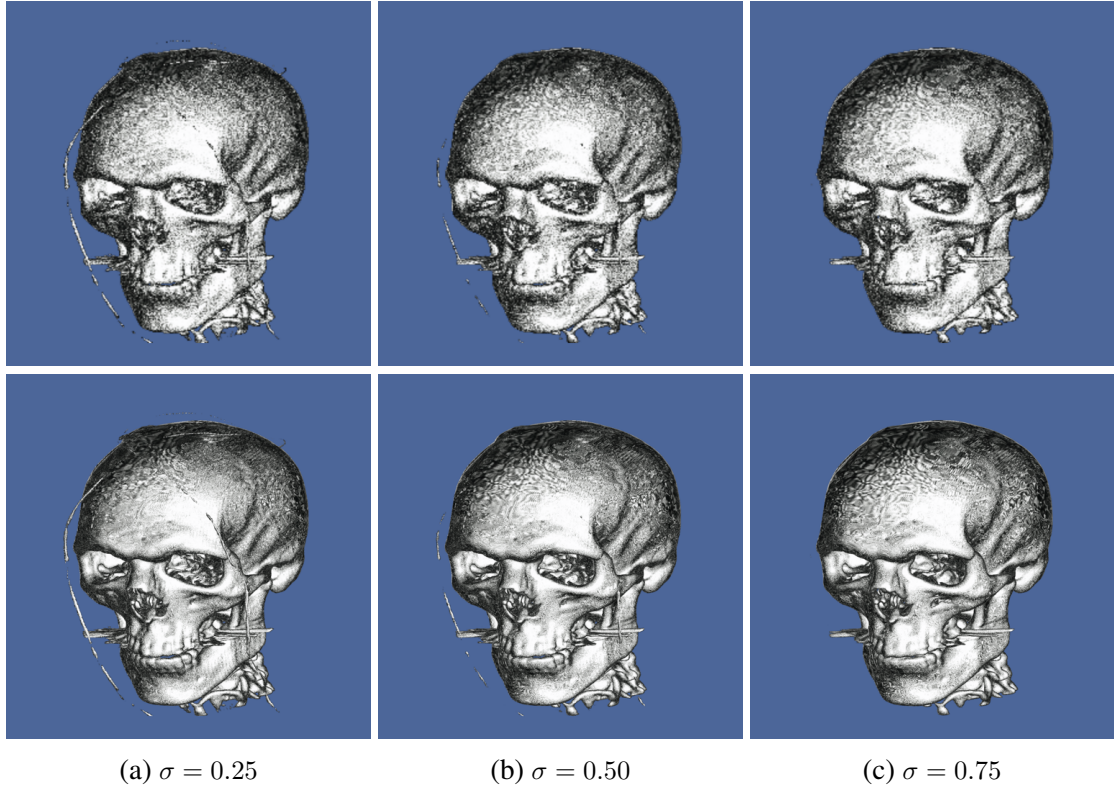


Figure 3.7: Stipple shading effect applied to human skull via Bruckner’s style transfer function method and associated spheremap, rendered at both low resolution (256x256 pixels, top row) and high resolution (1024x1024 pixels, bottom row), for the ‘head’ dataset smoothed with varying degrees of Gaussian volumetric filtering (1 iteration, varying σ , constant radius). Note the increased artifacts as the smoothing increases for high resolutions; low resolution rendering is almost unaffected due to texel-sizes of the stipple pattern relative to the smaller rendering buffer. Source: Paul Ross.

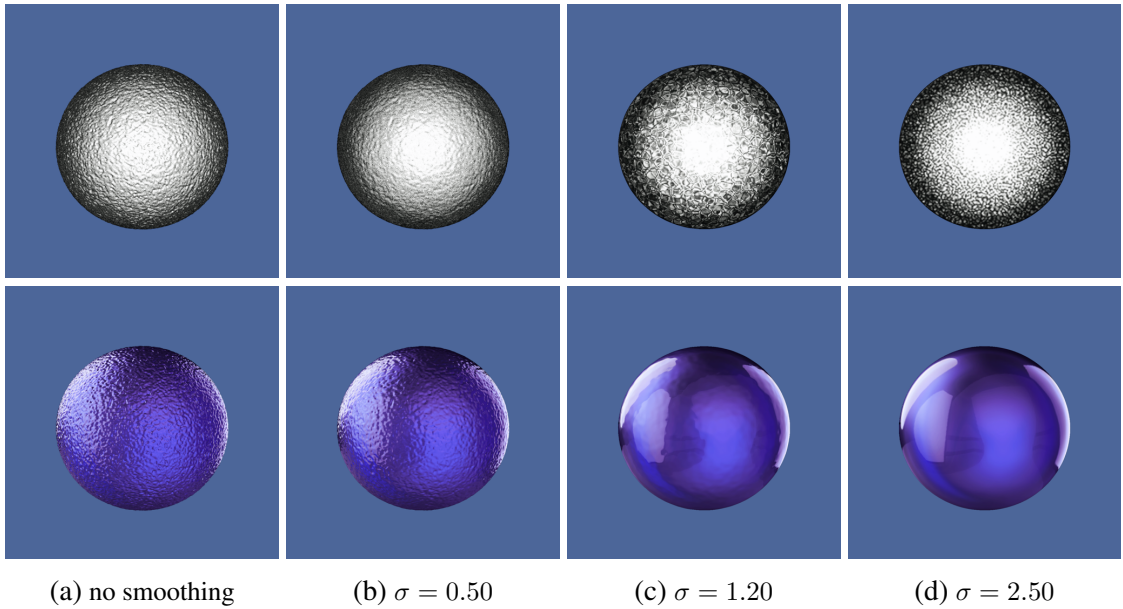


Figure 3.8: Shading applied to sphere using style transfer functions for varying levels of dataset smoothness (smoothed using one iteration of a volumetric Gaussian filter of fixed radius and specified σ). Top row: stipple shading using Bruckner’s own spheremap; bottom row: corresponding renderings using alternative lighting map to demonstrate surface roughness and faceting. All renderings are 1024x1024 resolution. Note the distortion of stippling pattern in (c). Source: Paul Ross.

3.6.3 Layering

Layered effects can be produced using combination of appropriate spheremaps and transfer function, paying particular attention to opacity of overlying isosurfaces. As real-world example, automotive spray paint is usually applied in multiple layers: a solid base-coat, and a transparent clearcoat. The former provides the solid colour base, and the latter both seals and applies a glossy finishing layer. The clearcoat layer can be tinted to achieve special effects (known as ‘candy’).

This can be replicated using style transfer functions (see fig.(3.9)): a thin low-opacity isosurface directly above the main body of the model can be shaded to approximate the clearcoat layer; tint is provided by means of the appropriate spheremap. The base-layer is specified using an opaque isosurface, together with an appropriately coloured ‘basecoat’ spheremap. Bruckner demonstrates a similar layering effect in [5], seen here in figure 2.7. Note that the example illustrated here in fig.3.9 is similar to what can be achieved using *materials property* effects featured in polygonal 3D modelling software such as *3D Studio Max*.

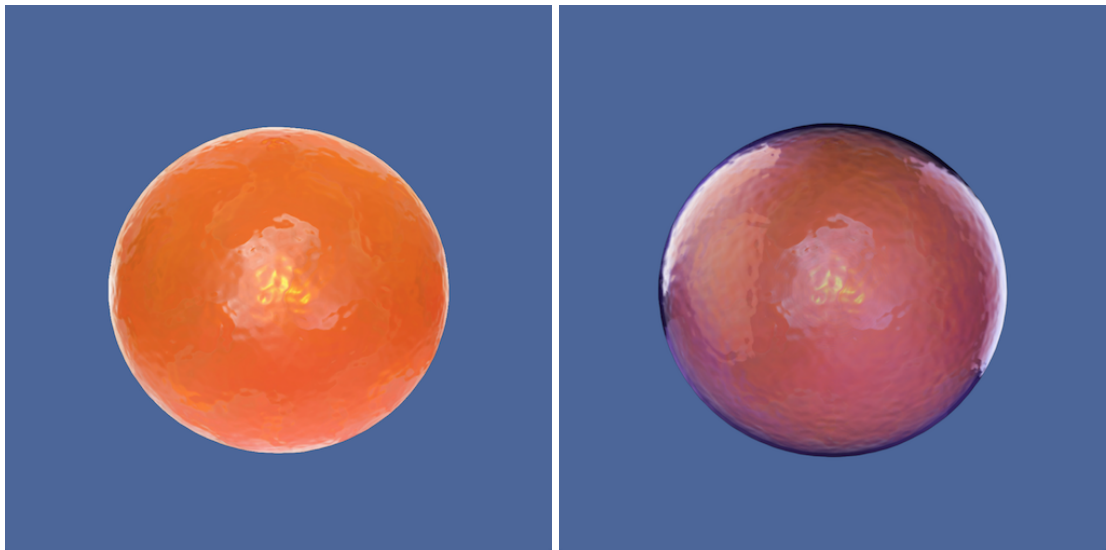


Figure 3.9: Layering effect using style transfer functions: a thin, coloured low-opacity layer around an opaque surface can increase the perception of glossiness of the underlying surface. This is essentially an analogue of the pervasive two-stage paint process used in the automotive industry: an opaque layer with an overlying application of transparent ‘clearcoat’. If the latter is coloured, it is termed a *candy* paint. Source: Paul Ross.

3.6.4 Highlighting salient structures

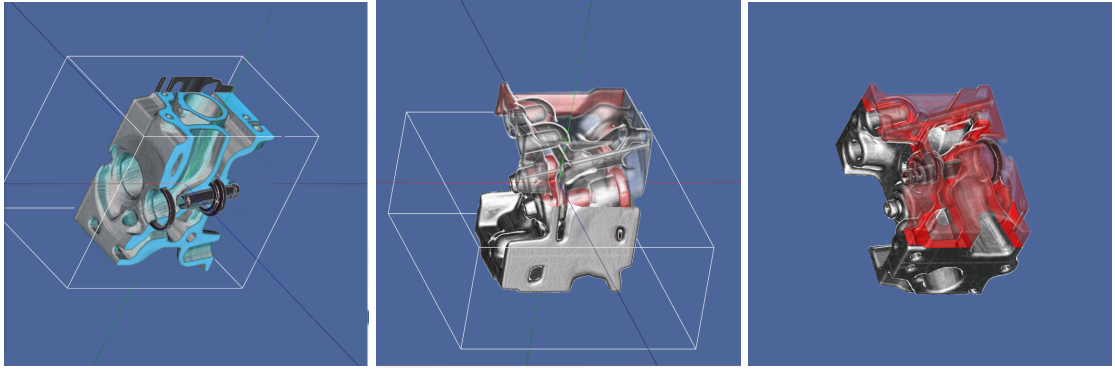


Figure 3.10: Illustrative cutaways and salient-feature highlighting using the ‘engine’ dataset using selective clipping, partial opacity and style transfer function rendering. Source: Paul Ross.

Style Transfer Functions can also be used to effectively highlight certain structures of interest. Figures 3.10 shows the engine model using selective clipping, partial opacity and region-specific spheremaps to highlight particular features in the ‘engine’ CT dataset.

3.7 Application to medical imaging

Volumetric raycasting in conjunction with style transfer functions can produce high quality rendering across a wide variety of styles: the spheremap shading method can be employed to perform both illustrative and near-photorealistic shading[§]. This technique is capable of reproducing most normal-based procedural local lighting models by essentially replacing normal-based local lighting computations with references to pre-shaded two-dimensional textures.

Bruckner discusses the concept of context-preserving illustrative rendering in [65] and demonstrates (with Gröller) the application of style transfer functions to CT datasets in [5]. Context-aware rendering with style transfer functions for CT data will be considered here also; furthermore, the application of style transfer functions will be extended to encompass additional modalities: namely 3D and streaming 4D ultrasound.

A discussion regarding the limitations of such local shading methods (and even global illumination, particularly in the context of noisy modalities such as 3D ultrasound) and how these limitations can be ultimately mitigated form the basis for the scope and fundamental motivation of this research project.

[§]as close to photorealistic as can be achieved through strictly local lighting models; subsurface-scattering and multiple reflection effects displayed by global illumination are not possible directly

3.7.1 CT imaging

Style transfer functions are particularly suitable for CT medical imaging applications: the noise level within the dataset is typically low and the imaging process penetrates through the entire body, creating the possibility for rendering with multiple layers of tissue structure of varying opacity. Context-preserving artistic techniques such as illustrative cutaways and other forms of occlusion management (e.g. view-dependent opacity), together with highly contrasting rendering styles can reduce visual complexity and emphasise salient features whilst retaining overall context.

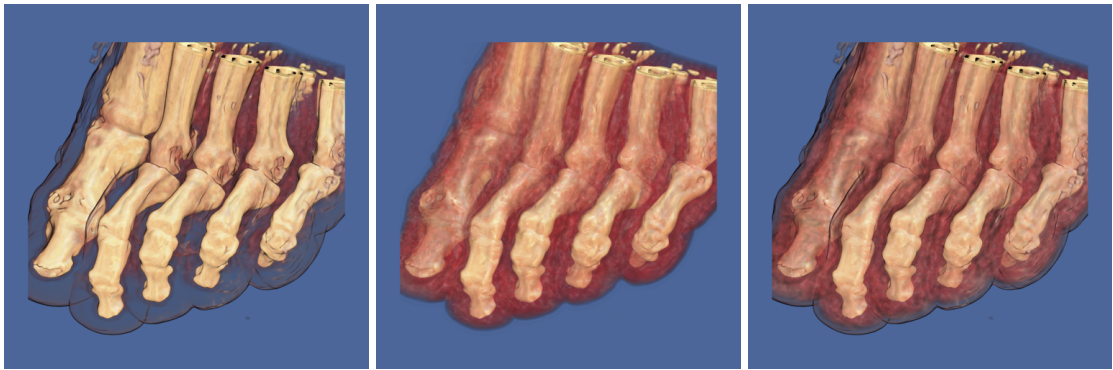


Figure 3.11: Style transfer function rendering of the ‘foot’ dataset: normal-based silhouette rendering to provide added context can be seen in the leftmost and rightmost images. The addition of a highly specular spheremap for flesh rendering has an additional effect on the shading of the bones due to the former’s translucency: their overall rendering style has an appearance similar to that of oil paint in traditional artwork. Source: Paul Ross.

Figure (3.11) depicts the ‘foot’ dataset rendered with style transfer functions (using three spheremaps: for bone, flesh and skin). Additional normal-based shading is used to provide silhouettes around the toes for context. Note that the red spheremap has a highly specular: this overlays the underlying shading of the bone (in a layering effect similar to that discussed in §3.6.3 to produce an artistic ‘oil paint’ effect for the skeletal structures. A more exaggerated illustrative style can be seen in figure (3.12); the rightmost image has additional contours which contribute largely to the non-photorealistic appearance. Enhancement via contour shading helps to isolate each distinct structure and thus improves the depiction of overall shape; however, regions of high curvature (e.g. at the joints between the bones on each toe) are lacking in definition.

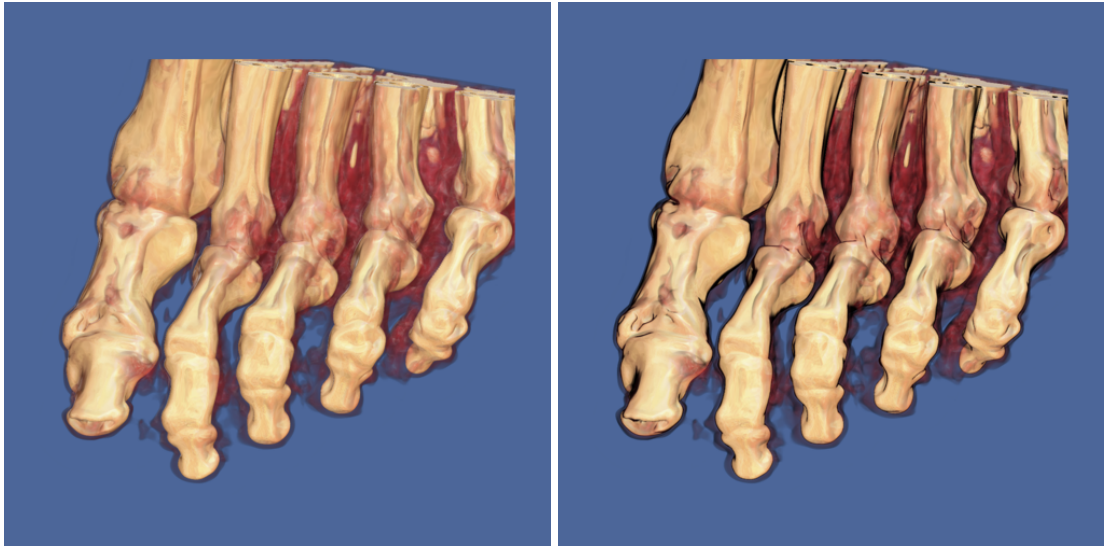


Figure 3.12: Style transfer function rendering of the ‘foot’ dataset: the rightmost image has additional normal-based shading applied to form illustrative contours, resulting in a more cartoon-like appearance. Note that there is a lack of delineation between the bones that form each toe; additional curvature-based crease lines would help in this instance. Source: Paul Ross.

Technical illustrators often make use of devices to manage occlusion, such as partial opacity and cutaway views; such techniques aim to emphasise structures of interest within context. Figure (3.13) depicts the ‘head’ dataset, rendered using style transfer functions: the intention is to visualise the skull as a feature of interest within the overall context of the head. Contour lines help to improve shape depiction of the components of the skull. Note that the additional shading is applied to all structures using the procedural method outlined in §3.4: this can be restricted to just the skull (see fig.(3.14) by using a specially constructed spheremap instead, where the contour is drawn around the edge as described in [5]; such a ‘ringed’ spheremap would be applied to the skull only in this case.

In figure (3.15), selective clipping is again applied to the same dataset in order to depict the skull in context; in this instance, contrasting shading styles are employed to maximise visual impact and contrast between structures. Also shown are the use of contours, applied to the whole head and skull only for comparison. The use of contours improves the depiction of edges but does not shade regions with high curvature on a given isosurface. The surface of the skull is relatively free of texture detail: brush-strokes from the spheremap are visible, but surface detail such as ridges and valleys are not emphasised; this may of course be desired within certain flatter rendering styles.

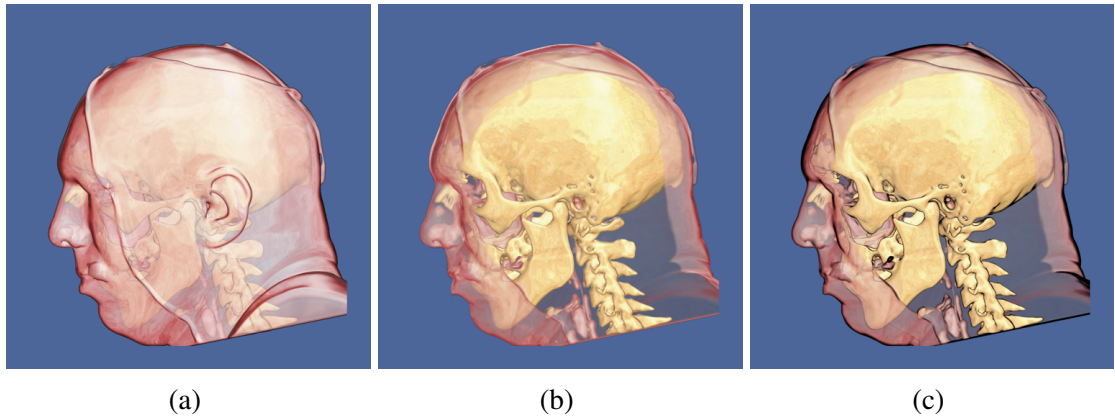


Figure 3.13: Style transfer function rendering of the ‘head’ dataset: (a) partial opacity to place the skull in context; (b) with additional cutaway for skull detail, using selective clipping; (c) with added contour lines for improved shape depiction, particularly around the edges of the skull. Source: Paul Ross.

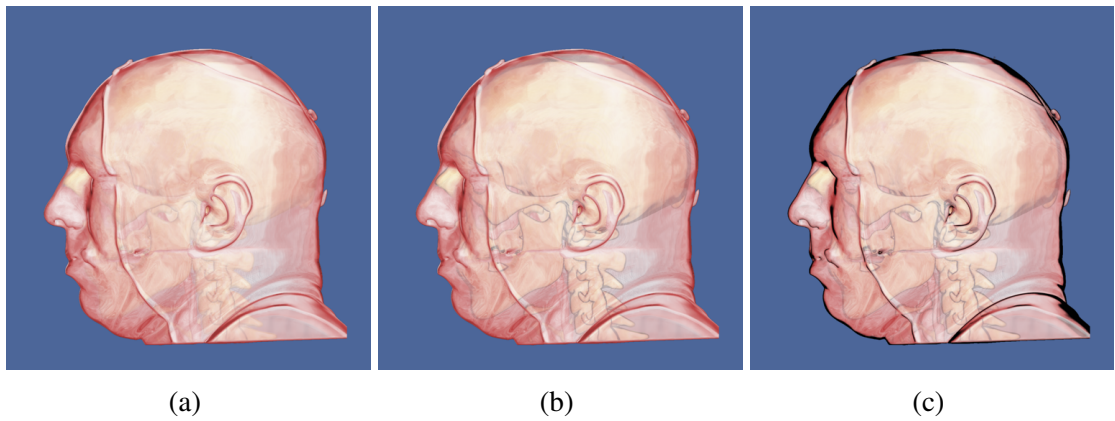


Figure 3.14: Applying contours to individual structures via spheremaps with an integral contour ring can improve the depiction of partially occluded features (in this case, the skull), without reducing the opacity of the outer layer of the skin. (a) no contour lines; (b) contour lines applied to the skull only; (c) contour lines applied universally. Source: Paul Ross.

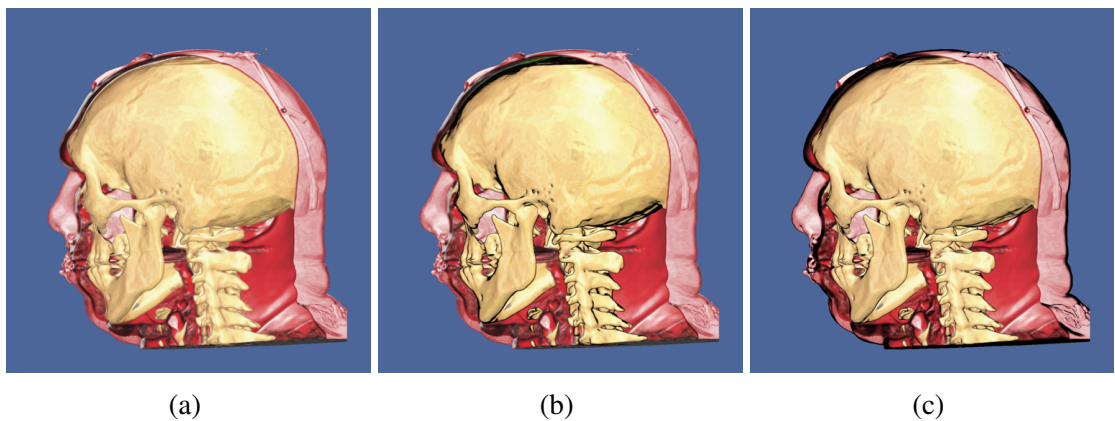


Figure 3.15: Highly contrasting shading styles, selective clipping and structure-specific contours can help isolate areas of key importance: (a) no contour lines; (b) contour lines on the skull only; (c) contours universally applied. Source: Paul Ross.

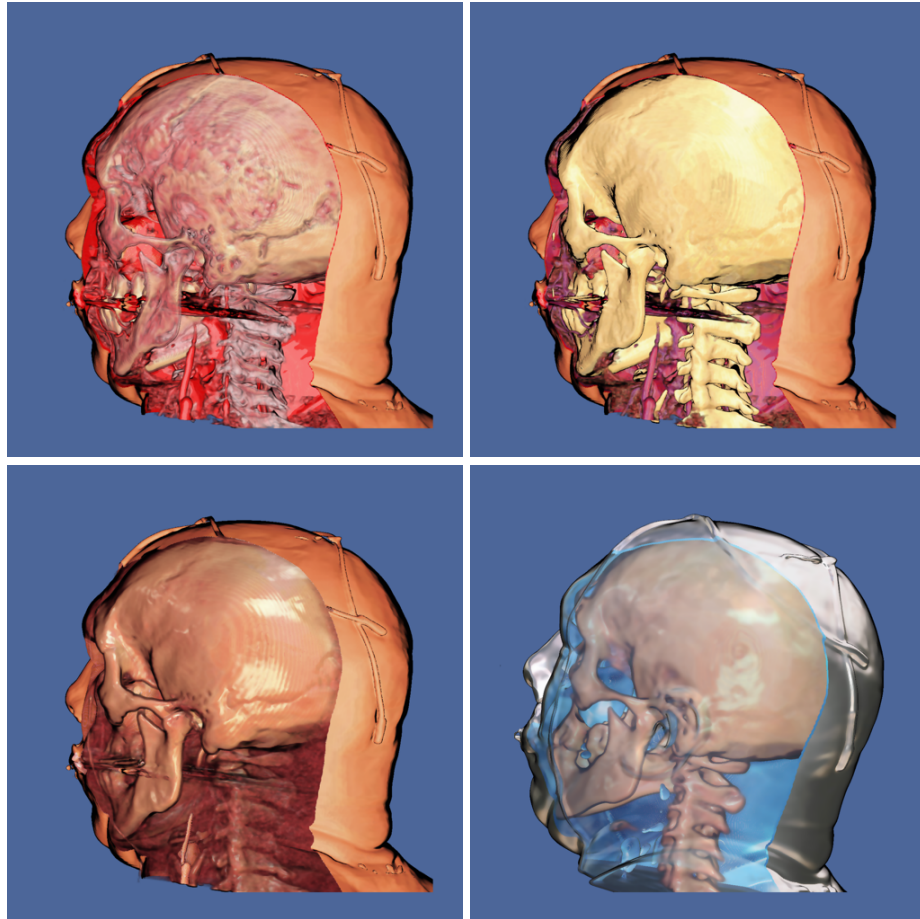


Figure 3.16: Style transfer function renders of the ‘head’ dataset with various combinations of spheremaps with selective clipping; a range of illustrative and photorealistic shading styles (or a combination thereof) can be used for enhanced contrast between structures. Note that shading of the valley regions that constitute joints between plates of the skull is entirely dependent on the choice of shading map with their intrinsic lighting direction. Source: Paul Ross.

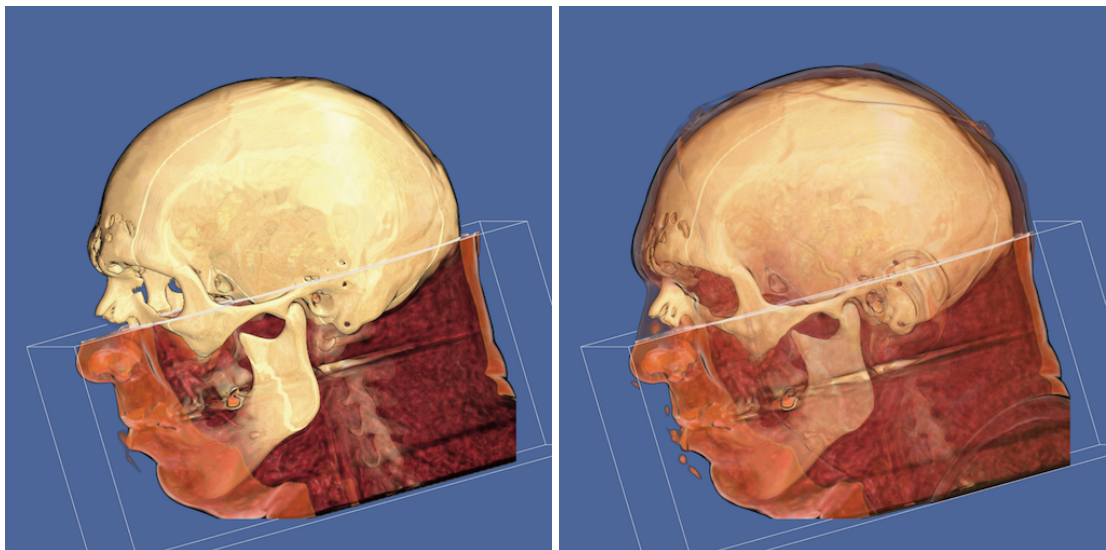


Figure 3.17: Style transfer function renderings of the ‘head’ dataset with selective clipping for context preservation: (a) fully transparent ‘traditional’ clipping (b) partial opacity clipping. Source: Paul Ross.

Figure (3.16) depicts the use of different shading styles to render the same scene for

various effects: note that depiction of surface ridges and valleys on the skull is entirely dependent on the choice of spheremap with associated lighting direction.

Lighting-independent ridge and valley enhancement through curvature shading can improve the depiction of such regions. Curvature transfer functions [3] map a specific iso-surface curvature metric or metrics to colour (or opacity) in an analogue to the normal-based spheremap. Ridges and valley regions (as regions of high curvature) can thus be highlighted using the desired colour scheme. Curvature-transfer functions have been integrated into the renderer: these can replace normal-based spheremaps for primary shading if desired. Figure (3.18) depicts a combined transfer function with both shading methods employed: the skull is rendered using the same spheremap as before, whilst a curvature transfer function is applied to the skin. Note the shading of ridges and valleys on the surface, particularly the ear region.

One caveat with curvature transfer functions is that they are an *alternative* to normal-based spheremap shading, rather than being complementary to it; they are also more difficult to design than standard spheremaps. Curvature-shading can however be employed as a secondary procedural shading method together with the primary spheremap shading style. Lighting-independent ridge and valley enhancement through supplementary curvature shading will be introduced in the next chapter to improve ridge and valley depiction in the context of a given spheremap-based shading style.

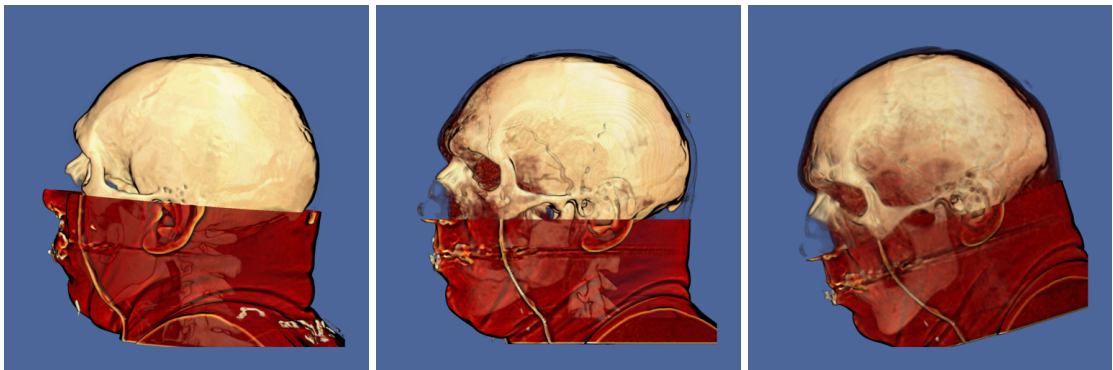


Figure 3.18: Hybrid rendering using a combined transfer function to depict the skull and spine within the context of the whole head: a curvature-transfer function is applied to the skin; the skull and flesh are rendered using two normal-based spheremaps. Overall, additional normal-based contour shading is also applied. Source: Paul Ross.

3.7.2 Ultrasound keepsake imaging

3D ultrasound presents a particular set of challenges: sound-waves cannot penetrate tissues to the extent that (X-ray based) CT-scanning can and it is currently confined to relatively low resolutions; moreover, it suffers from a high degree of noise or *speckle*, as a result of interference patterns formed by the interaction of ultrasonic pulses with body tissue. At present, the main consumers of such images are soon-to-be parents; hence, this

imaging field is often termed *keepsake imaging*.

The datasets used for the images in figures (3.19), (3.20) and (3.21) will be used throughout this project to demonstrate the limitations of specific filtering and rendering methods for the depiction of facial features with minimal noise in the context of 3D ultrasound; furthermore, they will be used to document the development of the main contributions of this research work.

Figure (3.19) depicts the ‘babyface’ dataset, rendered using a style transfer function and associated spheremap; also shown is the effect of additional normal-based shading as described in §3.4. Note that the data volume was prefiltered using one iteration of a volumetric Gaussian filter with $\sigma = 0.9$, radius 3. In the left-most image (a), there is some feature-delineation around the nose and mouth but this is largely dependent on the lighting direction as specified by the chosen spheremap. Additional normal-based shading can help to improve overall shape depiction in such regions: differing degrees of such shading can be achieved, from a subtle enhancement of feature boundaries to illustrative contour lines depicted in (b) and (c) respectively. Similarly, figure (3.20) depicts the same rendering processes applied to the ‘hand’ dataset: feature-delineation with additional normal-based shading and contour lines is slightly improved over style transfer function shading alone, but the images still lack detail in the palm and finger-joint regions.

In both cases, the additional normal-based shading steps can compensate for the limitations inherent in certain shading spheremaps (with respect to lighting direction, shading style or colour choice), but is an intrinsically *view-dependent* process. The following chapter (§4) will discuss view-independent curvature-based supplementary shading methods for improved feature delineation.

Figure 3.21 depicts style transfer function renderings of the ‘babyface’ dataset with a higher degree of Gaussian filtering. Feature detail is reduced drastically; this is exasperated by the particular lighting direction specified by the chosen spheremap. This particular case will be revisited in the following chapter.

Overall, style transfer functions offer high performance and flexibility regarding the chosen rendering style; caveats can be mitigated when coupled with an effective supplementary shading method for enhanced shape depiction and feature delineation.

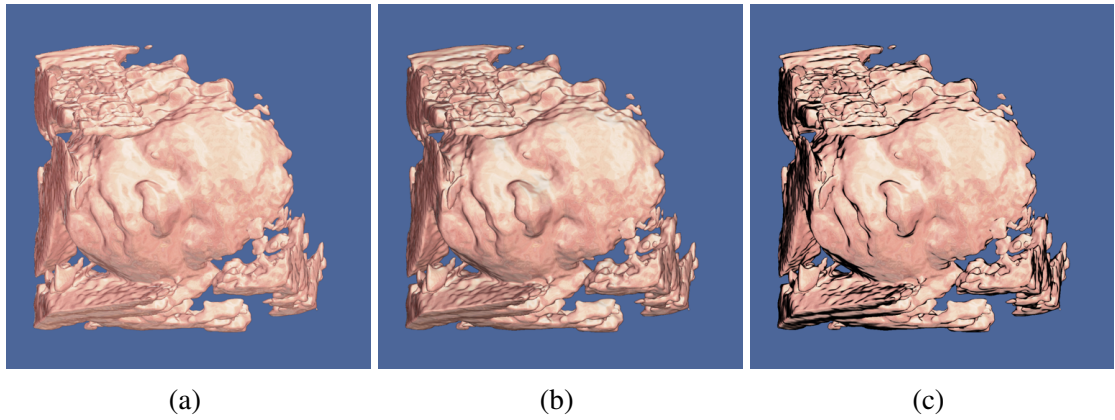


Figure 3.19: Improving facial feature depiction for style transfer function (STF) rendering of ‘babyface’ dataset: L-R: (a) STF only; (b) STF with a small amount of additional normal-based shading for improved shape depiction in the mouth and nose regions; (c) STF with normal based shading to produce contour lines. Note that the eye-regions are relatively unaffected by the additional shading and lack definition overall. Source: Paul Ross.

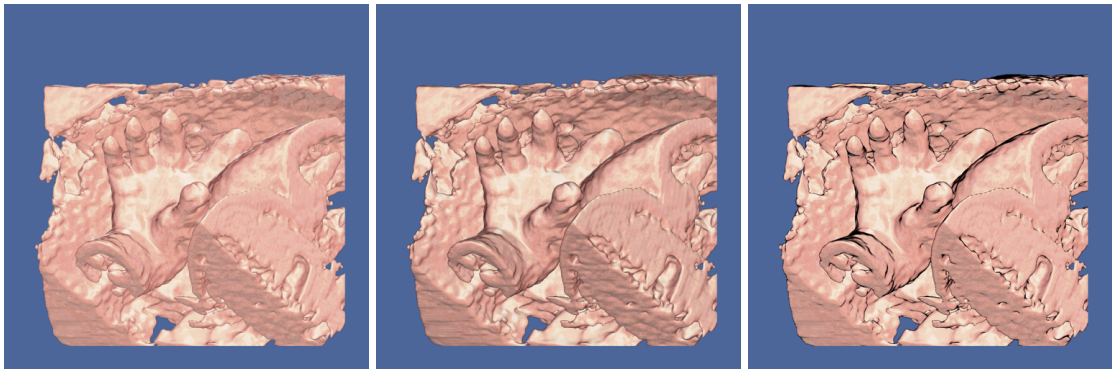


Figure 3.20: Style transfer function (STF) and additional normal-based shading applied to the ‘hand dataset’; the same approach as in fig.(3.19) was applied. L-R: STF only; STF with a small amount of additional normal-based shading for improved shape depiction; STF with normal based shading to produce contour lines. Overall, fingers and general shape are subtly highlighted through the use of the additional shading but there remains a lack of detail in the palm region and some of the finger joints. Source: Paul Ross.

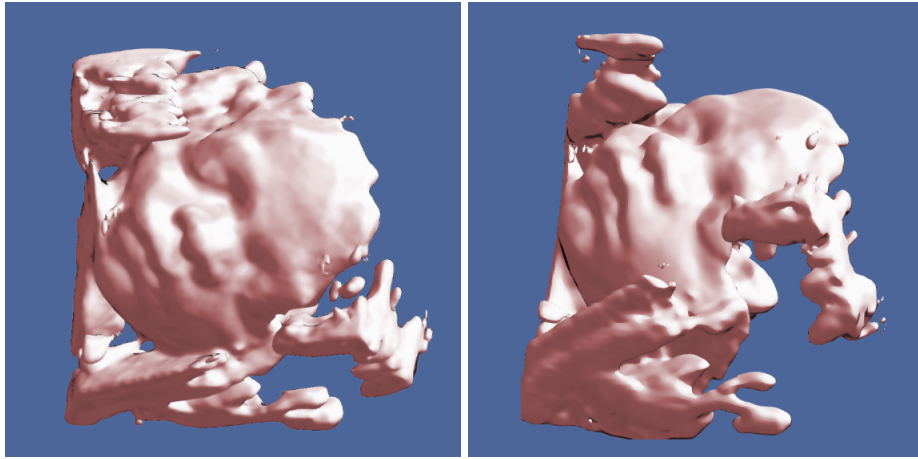


Figure 3.21: ‘Babyface’ ultrasound data rendered using the style transfer function method with associated shading/lighting spheremap. The lighting angle produced by this spheremap results in a significant loss of facial feature delineation. This dataset is smoothed using a higher degree of volumetric Gaussian filter ($\sigma = 1.6$) than used in fig. (3.19) to reduce visible isosurface noise: this results in a further loss of facial feature detail. Source: Paul Ross.

3.7.3 4D Ultrasound

3D volumetric ultrasound data can be streamed from its point of acquisition via the scanner hardware in a series of updates at a particular frequency: in this *4D ultrasound*, a temporal dimension is added as a succession of 3D data volumes are acquired, processed and rendered in real-time. Realtime streaming of volume data introduces a new set of constraints: all volumetric filtering and rendering operations must complete in limited time. Hence, at the time of writing, 4D ultrasound rendering is currently performed using monochrome MPR; volume-smoothing is commonly performed using either isotropic Gaussian or Perona-Malik feature-preserving filtering.

Application of alternative high-performance rendering methods for high quality images in this context was specified as a potential area of interest by Toshiba Medical Visualisation Systems. Initial results using style transfer functions were considered superior (by the internal research supervisors) to those produced by current MPR methods. The prototype renderer developed for this project has been developed to process pre-packed 4D datasets and produce an animated rendering as a final result in this context; it has not yet been tested running on 4D scanner hardware in realtime, but this remains an objective for future hardware iterations. Timing results can be obtained with regard to the overall filtering and rendering processes (per volume), so this can be used to evaluate suitability for given scanner hardware capabilities. This will be covered in greater detail in §6.

Style transfer functions can be applied to individual 3D subvolumes within a 4D ultrasound dataset: figure (3.22) depicts various shading maps (with a variety of lighting positions) applied to a heart-valve 3D ultrasound volume (a subvolume of a 4D series). The bottom row renderings feature an additional normal-based shading step to provide illus-

trative contour lines.

Renderings using style transfer functions alone feature the same benefits and issues as discussed in the context of keepsake ultrasound images: shading style is highly flexible but shape depiction is largely dependent on lighting direction (spheremap selection); contour lines help provide extra emphasis of internal ridge-like structures and thus provide an improvement in overall shape depiction, but this is fairly limited (see fig.(3.23)).

While contour lines can give the renderings an additional illustrative style, they are view-dependent; moreover, they are ineffective for shape depiction in valley regions of high curvature. Curvature-based methods for enhanced shape depiction will be discussed in the forthcoming chapter, with application to this particular dataset for comparison.

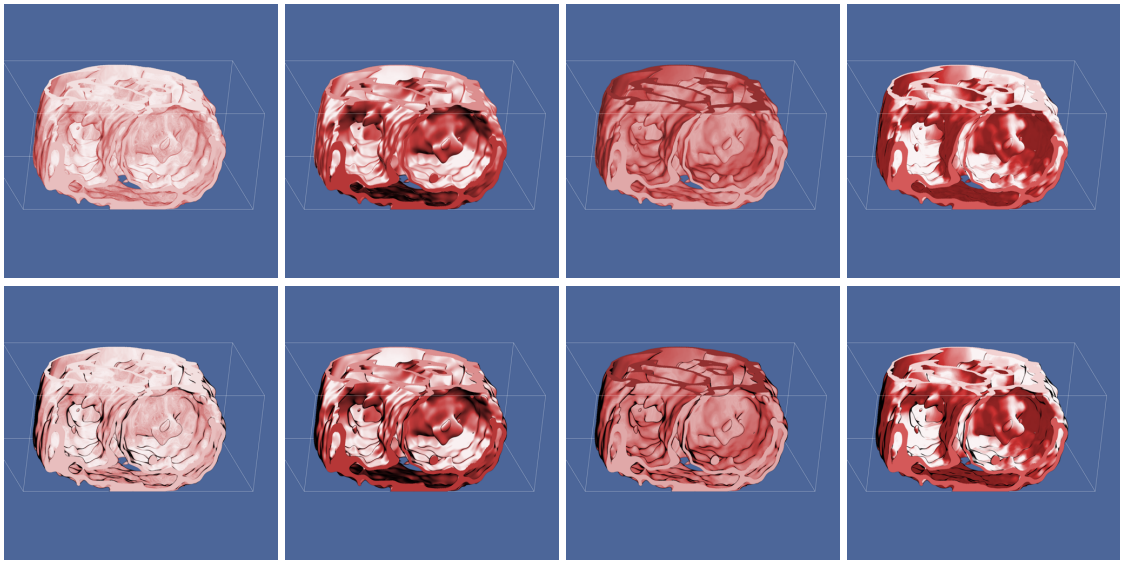


Figure 3.22: Style transfer rendering of heart-valve 4D ultrasound dataset with various shading maps (spheremaps). Bottom row: with additional contour lines. Source: Paul Ross.

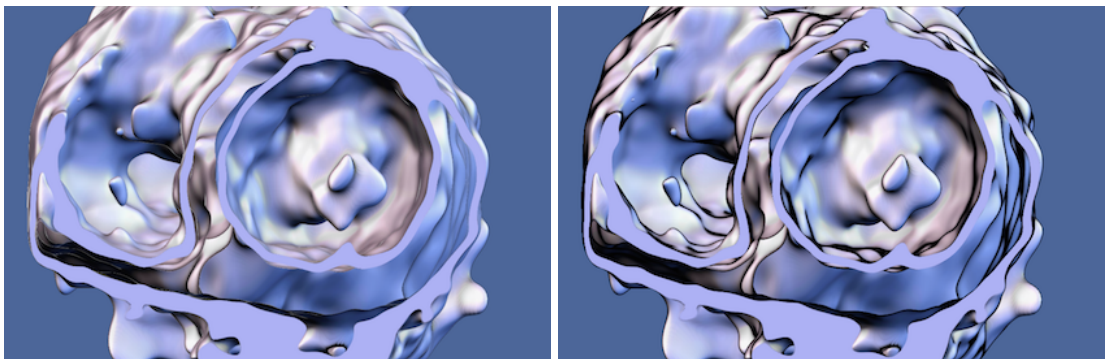


Figure 3.23: Style transfer function rendering of heart-valve 4D ultrasound dataset, showing illustrative effect of additional contour lines for improved (but limited) shape depiction. Source: Paul Ross.

3.8 Performance

Per-frame rendering timings were taken using the following hardware: Nvidia GTX680 (with 4GB memory) running on a Xeon workstation (2.4GHz processors, 2 sockets, 4 physical cores, 8 virtual ‘cores’ with hyperthreading) with 16GB RAM. The datasets were all rendered using the same transfer function with one spheremap, at identical z-distance with no partial opacity and clipping planes deactivated. As expected, style transfer functions performed comparably with Blinn-Phong lighting: a simple procedural local lighting method. Style transfer functions were slightly slower due to the texture handling overhead: modern GPUs are now very fast at simple procedural computation.

Dataset	Shading method		
	Blinn-Phong	STF	STF + contours
3D US Babyface (197x126x200, 16 bit)	13.01ms	18.00ms	18.50ms
CT Engine (256x256x256, 8 bit)	14.42ms	19.90ms	20.10ms
CT Head (128x256x256, 8 bit)	21.10ms	33.95ms	34.65ms

Table 3.1: Per-frame rendering time at 512x512 output resolution: comparison of local-lighting models.

For comparison, the overall render time for one frame using global illumination is 110ms for a 512x512 resolution using the same hardware; the style transfer function equivalent is 18ms (18.5ms with contours). Global illumination (GI) was, as expected, slower due to the multiple complex lighting computations required: STF is a local model. On limited hardware where realtime GI is not feasible, style transfer functions present a viable alternative to simple procedural lighting, and offer more stylistic choice.

3.9 Conclusion

Style Transfer Functions provide an elegant extension to the standard one-dimensional transfer function in direct volume rendering applications; this method provide a means by which local gradient/normal-based lighting and shading models can be easily incorporated into the transfer function itself. They facilitate rapid swapping of rendering styles, and thus are highly suited to both prototyping and implementing a large variety of illustrative rendering shading styles for various structures and isosurfaces within the context of different modalities. For ultrasound keepsake imaging, this method can provide an fast alternative rendering system to global illumination where performance is limited by older scanner hardware. Style transfer functions have similar performance (in terms of speed) to simple lighting models such as Blinn-Phong but offer a much greater flexibility in terms of rendering style.

The method is not without caveats however: shading in areas of high curvature (e.g. crease lines) isn't particularly pronounced for certain shading maps; it is highly-dependent on the light direction that forms an intrinsic part of the spheremap. Such high-curvature regions often form the boundaries of important structures; a lack of distinct shading in these areas can manifest as a lack of feature detail or a lack of delineation from their immediate surroundings. Emphasising these region boundaries helps to visually separate these structures from their wider context. For example: crease lines around facial features are not well represented by style transfer functions (or other local lighting methods) alone. Additional complementary shading methods, such as *curvature shading* can be employed to address this issue: these will be discussed in the following chapter.

Bruckner [138] notes that a further limitation of the single spheremap approach is that lighting and texturing are combined within the same shading map: this can result in 'moving textures' when the view-orientation changes. It is particularly prevalent with spheremaps that feature fine texture detail, such as the 'stipple' map discussed in §3.6.2. Bruckner suggests that an extended method featuring a separation of lighting and texturing may form the basis for future work.

Giving consideration to all of the above, the Style Transfer Function approach was considered on balance most suited to form the base illustrative rendering method for the duration of the project. Its various limitations will be addressed in forthcoming chapters: the full rendering pipeline will make use of this base system plus the enhancements and additions described henceforth.

3.9.1 Summary

In summary, style transfer functions offer several advantages over procedural local lighting models but are not without caveat; such properties can be summarised thus:

Advantages

- Easy specification of complex shading and lighting models.
- Allows multiple maps to be combined or swapped easily.
- Highly suitable to illustrative rendering styles.
- Allows complex shading/lighting effects not possible otherwise.
- Effective means of highlighting salient structures through contrasting shading styles.
- Has potential as a high-performance flexible alternative to global illumination for 3D keepsake ultrasound imaging (particularly when combined with supplementary

shading methods for feature enhancement; this will form the basis of discussion in subsequent chapters.)

Limitations

- Map construction is time-consuming and not adjustable in the renderer.
- Not easy to change lighting position: entire map must be changed.
- Spheremaps should ideally be designed by an artist.
- Finely detailed spheremap textures can become visually distorted under certain circumstances.
- Unified lighting and surface-texture combined in one spheremap can result in ‘moving’ texture in the rendered output; moreover, specular highlights should ideally be processed separately to allow intermediate deferred shading (see §4.7) or post-processing to be applied prior to the specular components.
- Detail is lacking in regions of high curvature (see §4); this is particularly apparent in the context of 3D keepsake ultrasound imaging.

As a fundamental component of a basic rendering system, style transfer functions fulfil the requirements for flexibility and speed within an illustrative context; together with supplementary shading methods they will form the basis of a system which will be developed over the course of the forthcoming discussion in an attempt to satisfy the fundamental criteria of enhanced feature depiction with minimal visual noise amplification.

Chapter 4

Improving detail: additional curvature-based shading

4.1 Introduction

As demonstrated in chapter 3, a major limitation of the style transfer function method is a failure to accentuate certain feature details that could significantly improve the depiction of overall shape in the final rendered image; indeed, this is an issue with local lighting models in general. The primary motivation of this chapter is therefore to develop a method which improves depiction of these salient features whilst minimising any noise-based artifacts.

Traditional illustration provides some inspiration as to how this may be addressed: through the use of *contour* and *crease* lines and/or shading cues based on surface orientation [47]. Ridges and valley regions in general provide a good match for the type of features prevalent in medical images that may lack definition: e.g. facial or other anatomical features on obstetric ultrasound scans. These regions can be enhanced by the use of appropriate shading: Gooch et al. [47] adopt the convention of darkening and lightening respectively. In addition, local surrounding areas may also be shaded in accordance to the differential properties of the surface in that local region. In the context of obstetric ultrasound, valley regions generally correspond to facial feature boundaries (e.g. eyes, lips, nose): enhancement of valley regions should therefore provide increased definition and help delineate the features themselves. This provides the motivation for the methods described in this chapter.

Ridges, valleys and other surface topology can be identified by considering the curvature of the surface at a given point (essentially a measure of change in direction of the surface normal); this is easily computable in real-time for GPU renderers at the time of writing. The method developed in this chapter applies additional shading to the whole visible

surface based on these local differential properties.

Unfortunately, curvature based shading methods can also highlight any noise inherent in the dataset: possible solutions involve performing the additional shading as a post-processing step using smoothed curvature data, and improving the volumetric filtering. The optimal approach will use both: a deferred and smoothed curvature shading step will be discussed later in this chapter, whilst improvements to volumetric filtering will be discussed in chapter 4.

Curvature-based shading methods are considered in the literature mainly in the context of non-photorealistic rendering; this is a major motivation here, but curvature-based shading can also be used to enhance feature depiction in other rendering contexts (e.g. global illumination).

4.2 Motivation: traditional illustrative methods for enhanced shape depiction

Traditional illustration often places emphasis on crease and contour lines in order to enhance topologically significant features; these provide the viewer with visual cues which can induce a greater intuitive impression of the surface topology. Moreover, an illustration can be reduced to a minimal representation of the object or surface using such features alone. In a sense, they represent the essential elements by which the object's shape can be conveyed: the salient features by which the viewer can recognise the object.

Generally:

- *Contour lines* or *silhouettes* describe the points in which the surface normal direction changes from a front-facing direction to rear facing with respect to the viewer or camera; they represent the boundary between front and back-facing surface geometry [3][47].
- *Crease lines* are found in feature-boundary regions (e.g. the outline of the eyes or nose); these are areas of high local curvature magnitude with respect to the surrounding region [54].

Figure (4.1) presents two illustrations of a human hand constructed with contour lines only (left image) and additional creases (right image) to demonstrate the contribution of each to overall shape-depiction.

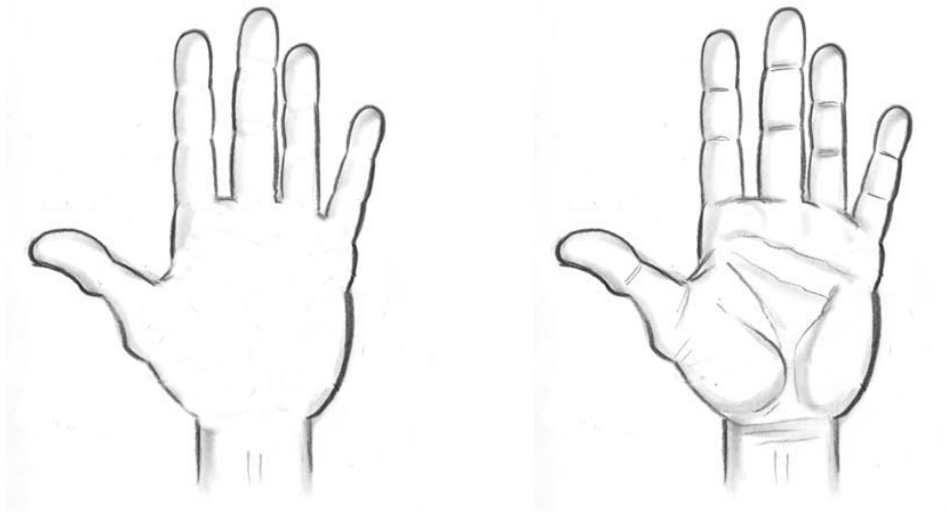
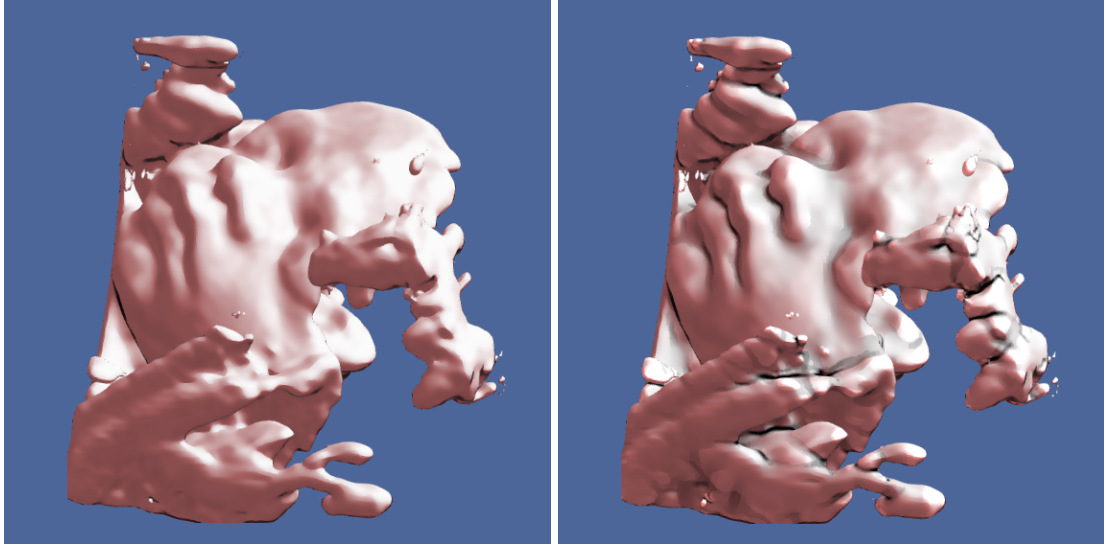


Figure 4.1: Contributions of contours and creases to overall shape depiction of illustrated hand. Left: contour lines only; right: contours and crease-lines. Original illustration source: Tom Richmond [6].

4.3 Curvature-based shading methods

One of the main limitations of local lighting models such as the *style transfer function* is lack of feature delineation: the areas that can be highlighted with contour and crease lines. In the case of contour lines, this can be mitigated by the use of additional normal-based shading (using a function of $\mathbf{v} \cdot \mathbf{n}$, where \mathbf{v} is the view-vector directed from a given point towards the viewer and \mathbf{n} is the surface normal at that point). In the case of style transfer functions, a similar effect can be achieved through the use of a spheremap with a surrounding contour ring [5]. For a more comprehensive description, see §3. However, contour lines are not sufficient to address the issue of low feature-detail. Crease lines occur in areas of high surface curvature with respect to surrounding regions; additional shading (using a curvature-based metric) in these areas can significantly enhance feature detail (see fig.(4.2)).



(a) Style-transfer function render using single spheremap. (b) Style transfer function as in (a) with additional curvature-based shading.

Figure 4.2: Effect of curvature shading on ultrasound foetal image showing enhanced feature detail. Source: Paul Ross.

4.4 Curvature

The concept of isosurface curvature in a scalar field f in \mathbb{R}^3 refers to the change in orientation of the normal of an implicit embedded isosurface (alternatively, the change in its tangent plane) in a local neighbourhood around a given point: it provides a metric with which to describe the amount the surface ‘bends’. Curvature metrics are 2nd derivative measures, and can be computed using the Hessian matrix \mathbf{H} of 2nd partial derivatives along with the gradient ∇f .

The Hessian matrix \mathbf{H} is defined as follows:

$$\mathbf{H} = \begin{bmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{bmatrix} \quad (4.1)$$

where the notation f_{xy} denotes $\frac{\partial^2 f}{\partial x \partial y}$.

Note that \mathbf{H} is symmetric, since the order of derivative computation for the mixed partials is (analytically) equivalent; different orders of computation may produce slightly different results numerically, but this will be ignored here. *

*A computationally more expensive method for numerical evaluation of the Hessian involves computing the off-diagonal elements using both possible orderings of partial derivatives and taking the average to preserve the symmetry of \mathbf{H} .

Sign conventions

Kindlmann et al. [3] use the following:

- Isosurface normal $\mathbf{n} = -\frac{\nabla f}{\|\nabla f\|}$, where f is the intensity of the scalar field at a given point. Hence, \mathbf{n} is the normalized gradient vector with its direction reversed.
- Concave surfaces are considered to possess negative curvature; positive curvature denotes convexity.

These sign conventions will be adopted here.

4.4.1 Principal curvatures: an informal introduction

At any given point on an isosurface embedded in three-dimensional space, there are an infinite number of directions in which curvature (the change in normal vector) can be considered. Given a direction tangential to the surface at the point, the change in normal in the given direction is known as the *normal curvature* associated with the given tangent vector; it is analogous to a directional derivative.

It can be shown that at a given point, there exist minimum and maximum values for normal curvature: these are known as *principal curvatures* and their associated direction vectors form an orthogonal basis for the tangent plane at that point (see fig.(4.3)).

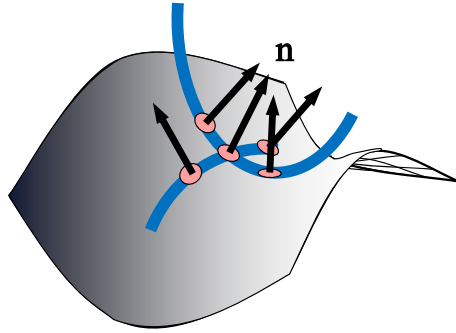
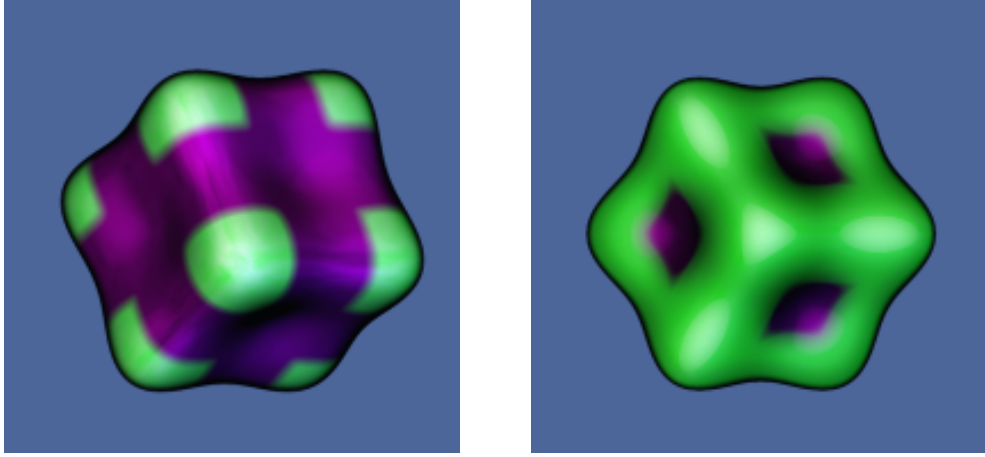


Figure 4.3: Change in normal direction on a curved surface; the blue lines indicate the loci of points traced out by the principal curvature directions. Source: Kindlmann.

The maximum principal curvature is denoted by κ_1 and the minimum by κ_2 , with $\kappa_1 > \kappa_2$ by convention[†]; note that both κ_1 and κ_2 can have positive or negative values. It is these principal curvatures that will constitute the curvature metrics used in the shading method: as can be seen from figure 4.4, distinct isosurfaces features can be highlighted effectively. This provides the motivation for possible application to medical imaging in the context of improving feature detail.

[†]Kindlmann adopts this convention, but an alternative would be to define κ_1, κ_2 such that $|\kappa_1| > |\kappa_2|$; this will not be used here.



(a) Metric: minimum principal curvature

(b) Metric: maximum principal curvature

Figure 4.4: Volumetric renders of synthetic test cuboid dataset, using principal curvatures as shading metrics. Areas of negative and positive curvature are shaded purple and green respectively. Source: Paul Ross, after Kindlmann et al. [3]

For a more in-depth coverage of the background theory and an outline of the curvature computation methods used in this research project, see §A.1

4.4.2 Ridges and Valleys

Ohtake, Belyaev and Seidel [147] provide a concise definition of surface crease lines: *the loci of points where the largest in absolute value principal curvature takes a positive maximum or negative minimum along its corresponding curvature line*. The crease line is traced by this line of curvature: this is orthogonal to the respective principal curvature direction; e.g. in the case of a valley, the valley floor cross-section curvature is essentially described by the local minimum value of the minimum principal curvature κ_2 , both in value and direction: the crease-line path of the valley floor is the corresponding line of curvature.

A valley crease line is therefore described by the loci of points (corresponding to local minima of κ_2) along the line of curvature orthogonal to the corresponding direction vector of κ_2 . Generally, valley regions (not just their minima) can be defined as regions where $\kappa_2 < -|\kappa_1|$ [147].

A ridge crease line can be similarly described as the loci of points (corresponding to local maxima of κ_1) along the line of curvature orthogonal to the corresponding direction vector of κ_1 ; generally, ridge regions (not just their maxima) can be defined as regions where $\kappa_1 > |\kappa_2|$

Figure 4.5 illustrates a simplified valley where the curvature in one principal direction is zero.

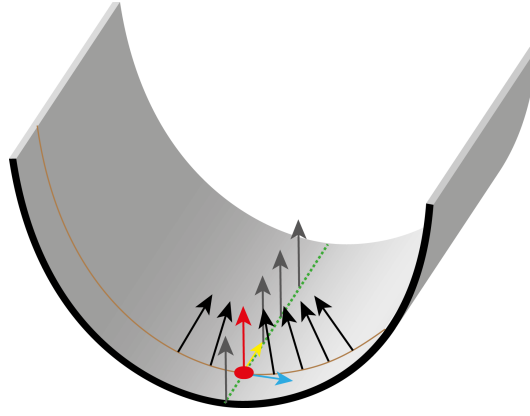


Figure 4.5: Simple valley example showing selected surface normals. In this case, $\kappa_1 = 0$, $\kappa_2 \leq 0$ at every point on the surface. At the indicated point (red), the principal curvature direction vectors are shown in yellow and blue (associated with κ_1 , κ_2 respectively). Note the loci of minima for κ_2 that trace out the valley floor (in green). The bending of surface normal vectors in both principal curvature directions clearly demonstrates negative curvature for κ_2 and zero curvature for κ_1 . Gaussian curvature $\kappa_1 \kappa_2$ is equal to zero everywhere on the surface. Illustration: Paul Ross.

A simple example of a surface with both ridges and valley regions (an extruded sinusoid) is shown in figure 4.6; figure 4.7 represents a planar cross section through this surface and illustrates how the principal curvatures change with cross-sectional position.

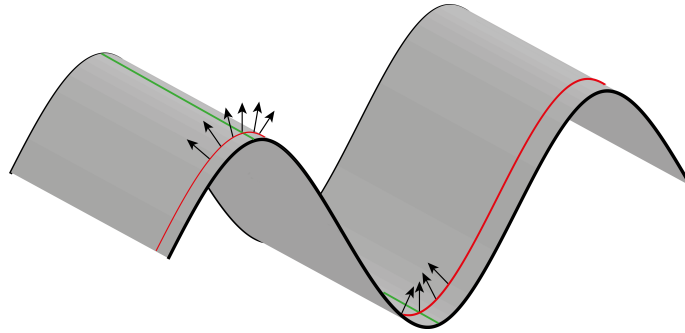
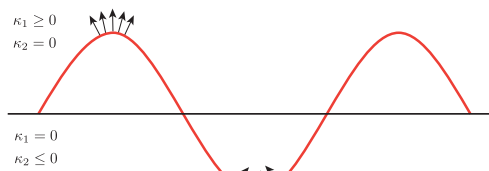
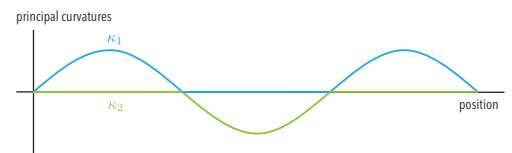


Figure 4.6: Simplified example of ridges and valley regions, showing surface normals. In this case, at each point there is always at least one principal curvature value equal to zero; hence the Gaussian curvature $\kappa_1 \kappa_2$ equals zero everywhere. The green lines mark the loci of maxima and minima of κ_1 and κ_2 respectively as described above: these trace out the ridge peaks and valley floor respectively. Illustration: Paul Ross.



(a) Cross section through plane of ridges and valley scene depicted in 4.6.



(b) Corresponding values for κ_1 and κ_2 . Note that $\kappa_1 \kappa_2 = 0$ everywhere.

Figure 4.7: Cross section of sinusoidal ridges and valley as depicted in 4.6, with associated principal curvature values κ_1 , κ_2 . Illustration: Paul Ross.

4.5 Curvature-based shading

Kindlmann [3], Hadwiger [45] and Bruckner [5] suggest using a curvature metric (or combination of metrics) together with a transfer function to provide the appropriate shading; the transfer function can be one-dimensional or multi-dimensional depending on the chosen metrics. The flexibility afforded by curvature-based transfer functions (in terms of being able to create essentially arbitrary effects with an artistically-designed transfer function) can be exploited effectively for illustrative applications. Indeed, illustrative applications applied to medical imaging data was the initial experimental remit for this project. Through experimenting with real medical datasets (acquired through various modalities), it has become repeatedly apparent that there is a fundamental requirement for a subtle increase in feature detail; as a result, the focus has shifted somewhat to providing a subtle enhancement of feature detail in an attempt to address the limitations of local illumination/shading methods such as the *style-transfer function*.

The curvature-based transfer function approach may not be required in this context as the additional shading can be achieved using a simple colour modulation. Valley regions can be highlighted by a slight darkening of colour shade with respect to the surrounding regions; similarly, ridges can be enhanced using a subtle lightening of shade. This can be easily achieved programmatically using CUDA in a final shading compute-kernel, without the need for a dedicated transfer function.

Of course, this effect can also be obtained through the use of a two-dimensional transfer function as demonstrated in [3]. Given a large increase in GPU power since the publication of the transfer-function methods, the CUDA (or OpenCL) approach offers several advantages:

- A simple modulation operation is well suited to the GPU and doesn't require the intermediate steps of texture creation.
- It is easily integrated into an already GPU-compute (CUDA or OpenCL) based renderer; no separate texture handling is required.
- It is relatively simple to modify and combine curvature metrics (simple arithmetic operations) for different shading effects; this would require the manual creation of a new transfer function texture each time.
- It may offer a performance advantage on modern hardware, since a few simple compute operations are relatively fast compared to texture handling and associated memory accesses.
- Note that that procedural supplementary shading methods are complementary to the primary shading stage; this allows for a high degree of flexibility.

As previously mentioned, the primary goal of curvature shading in this project is to pro-

vide a subtle enhancement of feature detail. An effective shading method by which to emphasize crease lines and other high-curvature regions is to simply modulate each colour channel using the desired curvature metric. The minimum-principal curvature κ_2 is mainly used in this regard, as features are delineated by valley regions. The following describes the processes involved using this metric alone and together with κ_1 , the maximum principal curvature.

4.5.1 Minimum principal curvature

To emphasize concave valley-type structures such as facial creases, the minimum principal curvature κ_2 is most appropriate as a shading control metric. Note that in this example, the same modulation operation is performed on each of the colour channels (R,G,B). In an HSV colour model, this is equivalent to modulating the V channel in the same proportion.

$$I_{out} = I_{in} + c\kappa_2 I_{in} \quad (4.2)$$

where c is a scalar multiplier controlling sensitivity and κ_2 is the minimum principal curvature; I_{out} is of course clamped to the permitted output range (colour channel values are normalised, i.e. $I_{in}, I_{out} \in [0, 1]$).

The shading produced from using κ_2 as a colour channel modulator is mainly characterised by a darkening of concave creases (since κ_2 is negative in these areas). However, κ_2 may be positive: the resulting effect is to lighten and highlight certain regions around the valley. This can give a better sense of the surface topology than just restricting κ_2 to negative values.

4.5.2 Strict valleys

To strictly restrict curvature shading to valley regions (as defined in section 4.4.2), the following operation can be performed (again on all colour channels R,G,B):

$$I_{out} = (1 + m)I_{in} \quad \text{where} \quad m = \begin{cases} c\kappa_2 & \text{if } \kappa_2 < -|\kappa_1| \\ 0 & \text{if } \kappa_2 \geq -|\kappa_1| \end{cases} \quad (4.3)$$

Colour channel values for I_{out} are again clamped to the permitted range.

4.5.3 Maximum principal curvature

In an analogous process to that used above to shade valley regions, ridge-like structures can be emphasised using κ_1 as a shading control metric: each colour channel (R,G,B) is

again modulated identically,

$$I_{out} = I_{in} + c\kappa_1 I_{in} \quad (4.4)$$

In contrast to κ_2 -based shading for concave regions, convex regions are shaded by lightening these with respect to the surrounding areas (κ_1 is positive).

There may be areas where $\kappa_1 < 0$: the resulting effect is to darken the surrounding area. Similarly to 4.5.1 above, this may help to portray surface topology more effectively than restricting κ_1 to positive values.

4.5.4 Strict ridges

Curvature shading can be restricted to strict ridge-like regions (where $\kappa_1 > |\kappa_2|$ as defined in 4.4.2):

$$I_{out} = (1 + m)I_{in} \quad \text{where} \quad m = \begin{cases} c\kappa_1 & \text{if } \kappa_1 > |\kappa_2| \\ 0 & \text{if } \kappa_1 \leq |\kappa_2| \end{cases} \quad (4.5)$$

4.5.5 Summary

Figure (4.8) depicts a style transfer function rendering of the ‘engine’ dataset with curvature shading applied using various metrics.

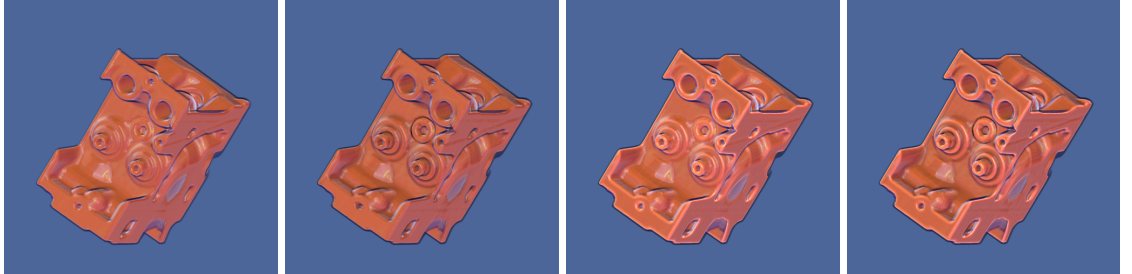


Figure 4.8: Style transfer function rendering of the ‘engine’ dataset with supplementary curvature shading (various metrics): L-R: no curvature shading; minimum principal curvature; maximum principal curvature; both minimum and maximum. Source: Paul Ross.

4.6 Application to medical imaging

As previously seen in the context of curvature transfer functions (see §3.7, fig.(3.18)), curvature based shading can enhance the depiction of facial features that are often poorly defined; this can of course be applied in general whenever detail enhancement is required. Formulating a curvature-based shading step as an additional supplementary process to the primary lighting and shading model in lieu of a purely transfer function based approach allows for the positive attributes of the primary shading method to be retained. In the case of

style transfer functions, their performance, interchangeability and flexibility with regard to multiple potential shading styles is maintained whilst allowing for some shortcomings to be addressed.

4.6.1 CT imaging

One shortcoming of such local lighting methods (and to some extent, global illumination) is a lack of feature detail within regions of high curvature, such as ridges and valleys. Curvature-based shading can address this issue by virtue of its definition: it seeks to apply a degree of local shading, whose measure is controlled by local isosurface curvature. It can highlight small undulations on an isosurface, giving added texture and depth; the effect is to generally increase the level of shape depiction within the renderer. However, it has the undesirable caveat of also highlighting small surface imperfections due to noise. CT imaging (certainly compared to ultrasound) typically has an inherently low level of noise, so this is not a major concern for curvature shading within the context of this modality.

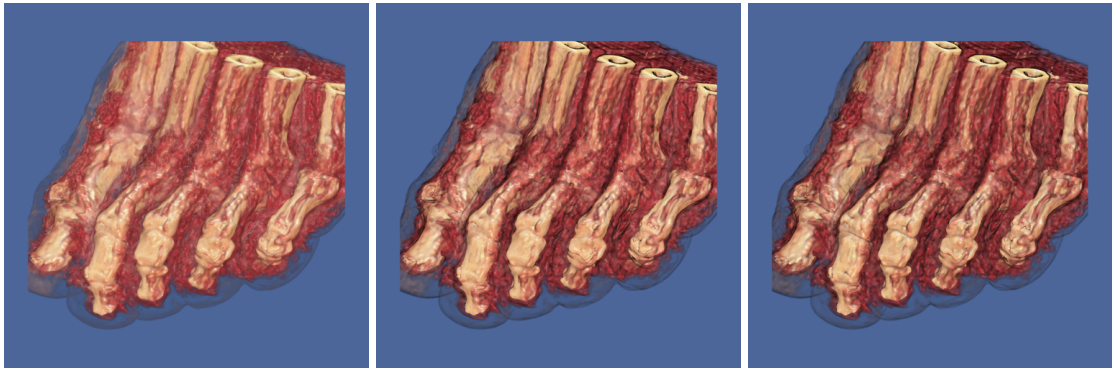


Figure 4.9: Curvature shading for the CT ‘foot’ dataset: this supplementary shading can provide an increased level of quasi-photorealism by shading the joints between bones in the toes. L-R: style transfer functions; with additional normal-based contours, with normal-based contours and curvature shading (minimum principal curvature). Source: Paul Ross.

Figure (4.9) illustrates an effective use of curvature shading to supplement the overall shading style and increase detail in specific regions. Note how the joints in the fingers have low definition for the first two images: curvature shading is applied in the third image to shade the valley regions at each joint. Such shading is also effective in highly illustrative contexts. Figure (4.10) shows a more stylised rendering of the same dataset: both the normal-based contours and curvature-based crease lines (third image) are fundamental to the overall rendering style but also provide an enhanced depiction of overall shape.



Figure 4.10: Effect of curvature shading for shape depiction (specifically the joints) in illustrative rendering of CT ‘foot’ dataset: L-R: style transfer function rendering; with additional normal-based contours; with contours and curvature shading (minimum principal curvature).

Both examples have used minimum principal curvature to emphasise valley regions; two variants of this metric are (1) its restriction to strictly negative values, and (2) strict valleys as defined in §4.5.2. Figure (4.11) depicts the somewhat subtle difference between them: note how small surface undulations are more pronounced using minimum principal curvature alone with no restrictions. In the context of CT imaging, the highlighting of small-scale surface features can be beneficial or detrimental depending on the level of realism, texture and rendering style required; in noisier modalities such as 3D ultrasound, it constitutes a significant problem as such imperfections are usually derived from speckle noise.



Figure 4.11: Subtle difference in shading using variants of minimum principal curvature: L-R: minimum principal curvature; minimum principal curvature restricted to negative values; strict valleys. Note the shading of small undulations within the flatter regions of the bones: this may or may not be desirable depending on context. Source: Paul Ross.

Curvature shading can be used to create illustrative exaggerated crease lines but can be applied in a more subtle form to enhance the overall shape or provide an increased sense of surface texture. Figure (4.12) shows the effect of applying minimum principal curvature shading to the ‘head’ CT dataset to enhance both facial features and skull surface texture. Together with normal-based contours, curvature shading (using both minimum and maximum principal curvatures) is applied in figure (4.13): note that the skull surface-detail, general shape depiction and contrast with overall context is increased.

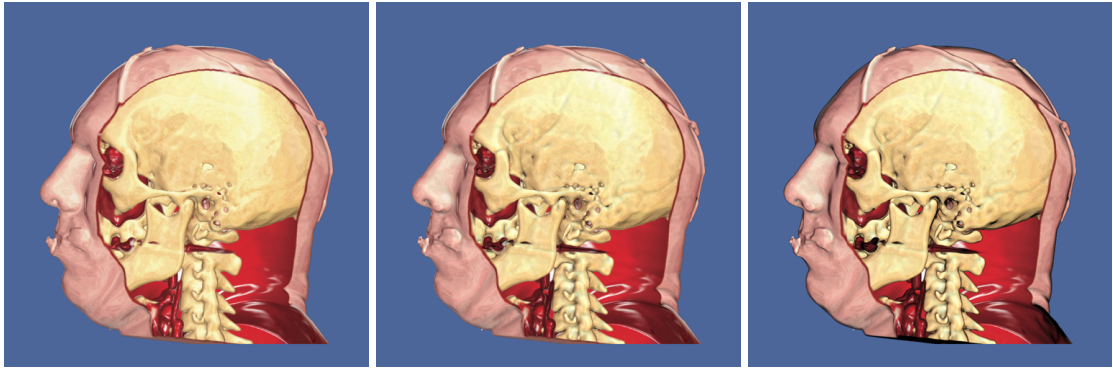


Figure 4.12: Enhancement of isosurface detail via curvature shading within the context of style transfer function rendering of CT ‘head’ dataset. L-R: style transfer function shading alone; with additional curvature shading (minimum principal curvature); with curvature shading and normal-derived contours. Source: Paul Ross.

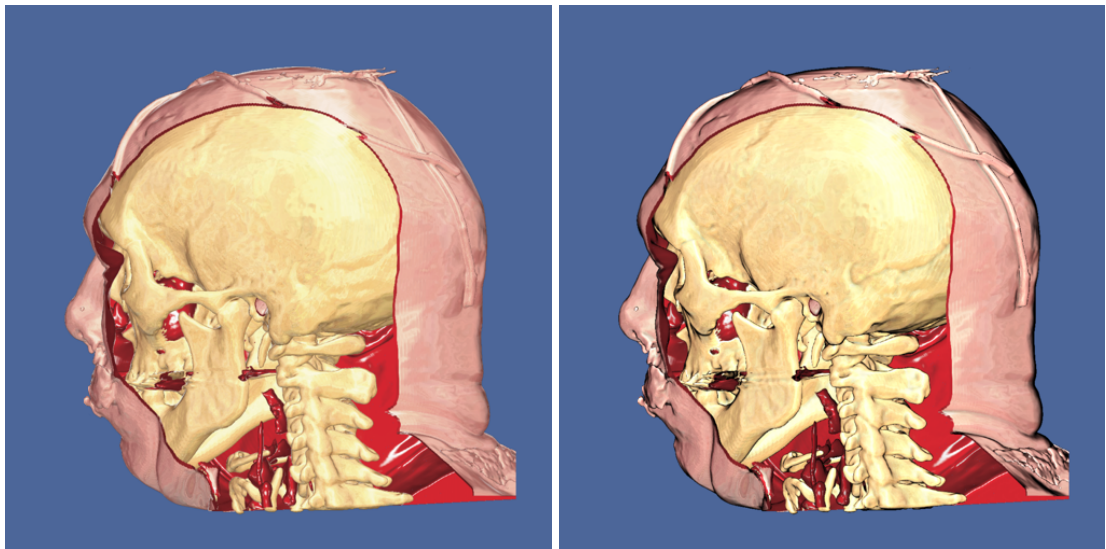


Figure 4.13: Enhancing shape and surface depiction of the skull using curvature shading (minimum and maximum principal curvatures), together with normal-based contours. Left: style transfer function (spheremap) shading only; right: with both supplementary shading steps. Source: Paul Ross.

Facial features can also be enhanced using a subtle application of curvature shading: figure (4.14) shows how subtle curvature-based shading (not just explicit crease lines) and contours can give an increased sense of realism and depth to an otherwise flat rendering and improve the depiction of overall shape.



Figure 4.14: Using a small amount of curvature-based shading (both minimum and maximum principal curvatures) to provide a subtle enhancement in shape depiction, using the ‘head’ dataset. L-R: style transfer function rendering only; with the addition of curvature shading; with the addition of curvature shading and normal-based contours. Source: Paul Ross.

4.6.2 Ultrasound keepsake imaging

As previously discussed in §3.7.2, 3D ultrasound suffers from a high degree of inherent *speckle* noise; moreover, it has comparatively low tissue penetration with respect to other modalities. In the context of volume rendering, the fidelity of output images using such data is therefore significantly less than that of rendered output obtained with CT data.

Figure 4.15 depicts the ‘babyface’ dataset rendered using style transfer functions: the first two images are produced using the same normal-based shading methods seen previously in fig.(3.19); the third image introduces curvature shading using minimum principal curvature to darken the valleys which form feature boundaries around the nose, mouth and eye regions. As can be observed, the depiction of these feature boundaries is improved; however, the major caveat is that speckle-derived surface bumps and undulations (particularly in the forehead region) are also enhanced. Increasing the level of curvature shading to highlight feature boundaries cannot be performed (using this method alone) without also similarly enhancing the visible manifestation of speckle noise.

In figure (4.17), an alternative shading spheremap is used: the lighting direction results in a further loss of feature detail in the rendered output compared with fig.(4.15). The application of curvature shading (using minimum principal curvature) significantly improves the delineation of facial features; again, however, the effect of visible-noise amplification is apparent, particularly in the forehead region.

What is required is a means by which to allow an increase of highlighting where required, whilst preventing or minimising highlighting of isosurface noise or speckle-derived artifacts. In theory, this can potentially be achieved using three approaches: (1) thresholding or restricting shading to regions of higher curvature using a power transform in the curvature-shading process; (2) performing the curvature shading step as a deferred process and smoothing the resultant curvature data using an edge-preserving method such

as a Perona-Malik filter; (3) remove as much noise as possible from the dataset volume (whilst preserving feature detail) by developing an improved volume filtering method.

Feature highlighting using this method of curvature-based shading is independent of both lighting and view direction: this can be advantageous for its consistency in depicting features outlines yet can suffer from over-darkening in regions where the primary shading method produces specular highlights. It is a consequence of applying the curvature shading after the primary shading step and is a limitation of the style transfer function approach where illumination and texturing are combined in the same shading map (as discussed by Bruckner in [138]).

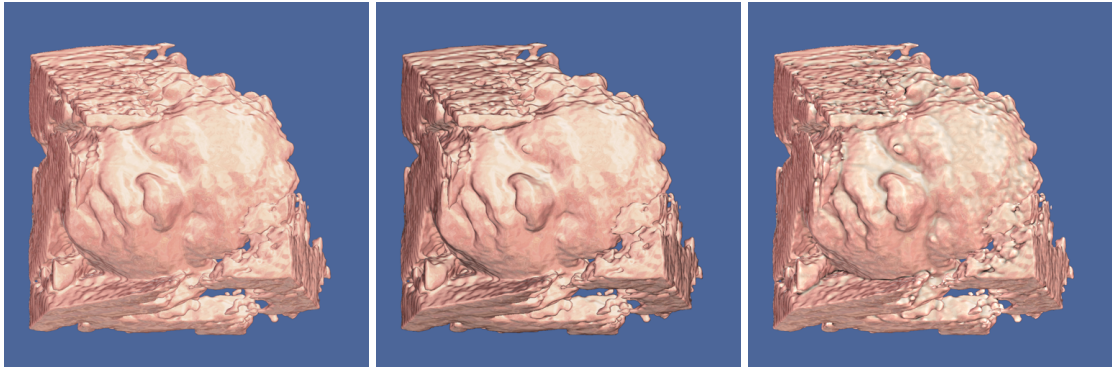


Figure 4.15: Style transfer function renderings with supplementary shading applied to an obstetric ultrasound dataset ('babyface' dataset). L-R: Style transfer function shading only; with an additional normal-based shading step for contours and enhanced shape depiction; with additional normal and curvature-based shading (minimum principal curvature). Source: Paul Ross.

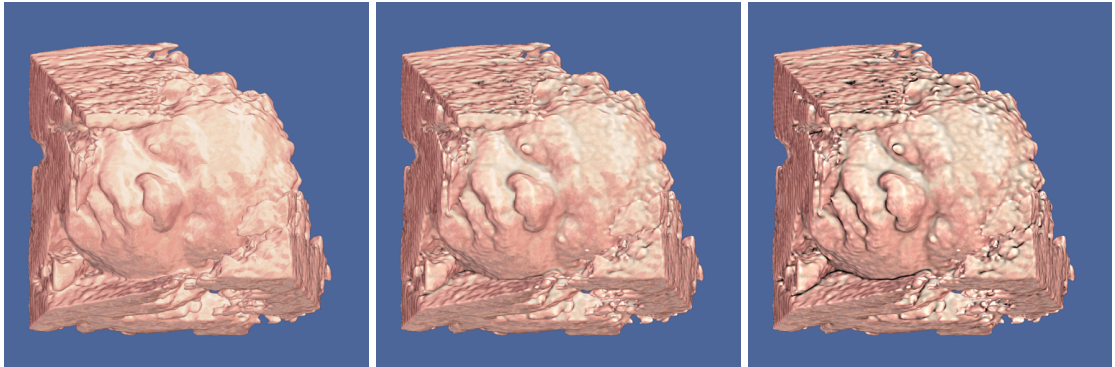


Figure 4.16: The noise-highlighting effect of varying degrees of curvature-based shading (minimum principal curvature). L-R: style transfer function shading only; curvature shading with multiplier $c = 0.33$; curvature shading with $c = 0.66$. Source: Paul Ross.

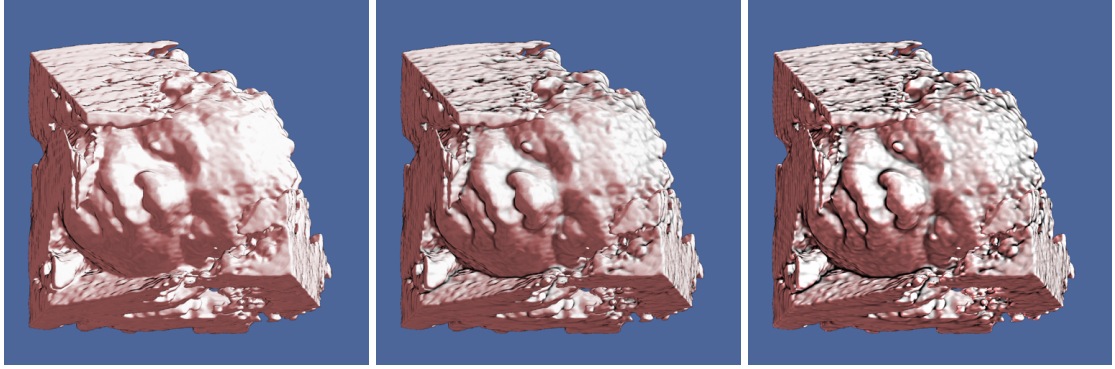


Figure 4.17: Curvature shading (minimum principal curvature) with undesirable noise-highlighting using style transfer functions with an alternative spheremap. L-R: style transfer function shading only; with additional curvature shading ($c = 0.66$); with a higher degree of curvature shading ($c = 1.00$). Source: Paul Ross.

4.6.3 4D ultrasound

4D ultrasound essentially consists of a temporal series of 3D subvolumes; such subvolumes are similar to the datasets previously discussed in the context of obstetric ultrasound in §4.6.2 above. The limitations of using style transfer functions to render output images from such subvolumes can be observed in §3.7.3, fig.(3.22) in the context of a heart-valve dataset: even with additional contours, the depiction of isosurface topology is indistinct. The use of additional curvature shading allows structures to be depicted with a much greater sense of surface geometry, as can be see in fig.(4.18); note that minimum principal curvature for was used as a shading metric (for valley enhancement) for the images within this section, as with the previous obstetric ultrasound images. Figures 4.19 and 4.20 show two more examples using the same dataset; note that in 4.19, the volume data has a greater degree of Gaussian filtering applied for smoother surfaces. However, both images illustrate the extent by which curvature shading can improve shape depiction for volumes of varying degrees of smoothness.

Figure 4.21 is a time-series depiction of rendered images from a series of 3D subvolumes: on the left of each image, the valve can be clearly seen transitioning to an open state.

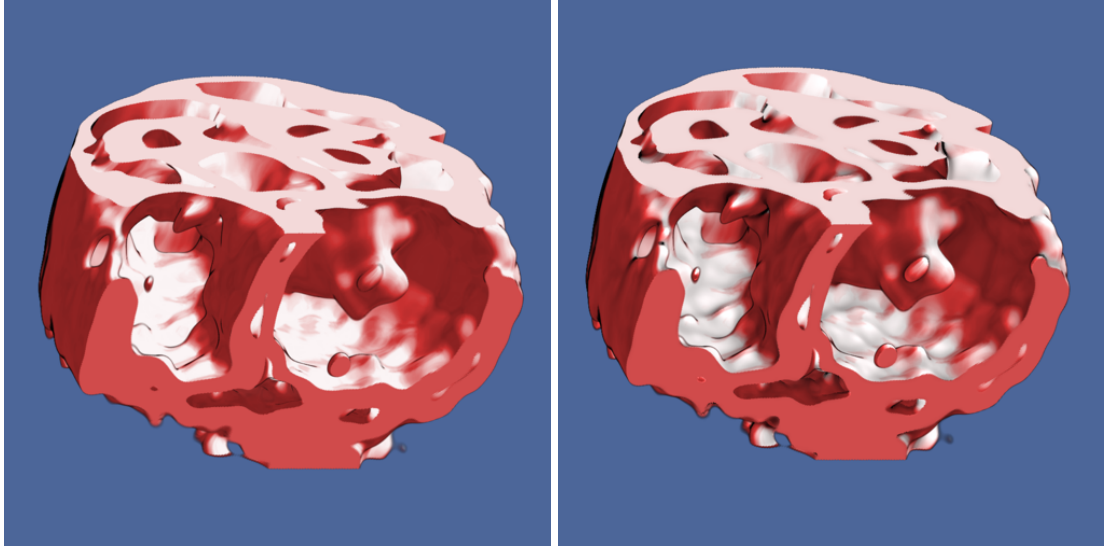


Figure 4.18: Curvature shading to enhance shape depiction for a 4D ultrasound sub-volume (heart-valve) rendered using style transfer functions. Left: style transfer function (STF) shading; Right: with additional curvature shading (minimum principal curvature). Surface detail is low for STF shading alone (partly due to the inherent lighting direction of the shading sphermmap); curvature-based shading provides an increased level of isosurface detail independent of light direction. Source: Paul Ross.

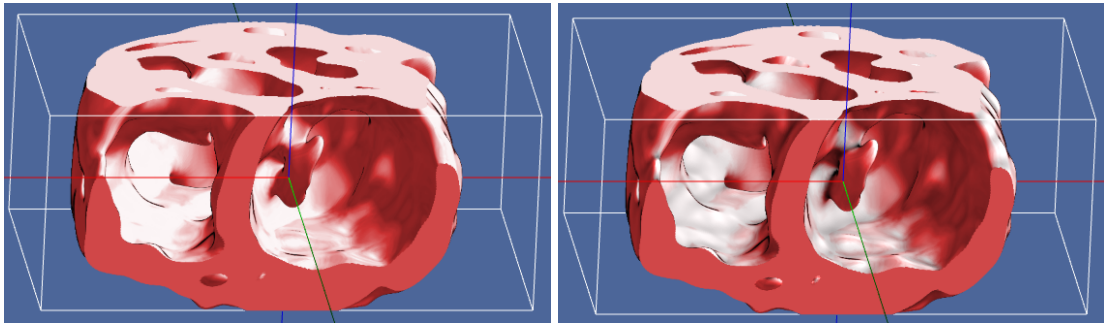


Figure 4.19: Style transfer function rendering with curvature shading, using the same dataset subvolume (heart-valve) and view as in fig.(4.18) but with a higher degree of Gaussian volume smoothing for the input data. Left: style transfer function shading only; Right: with the addition of curvature shading (minimum principal curvature). Source: Paul Ross.

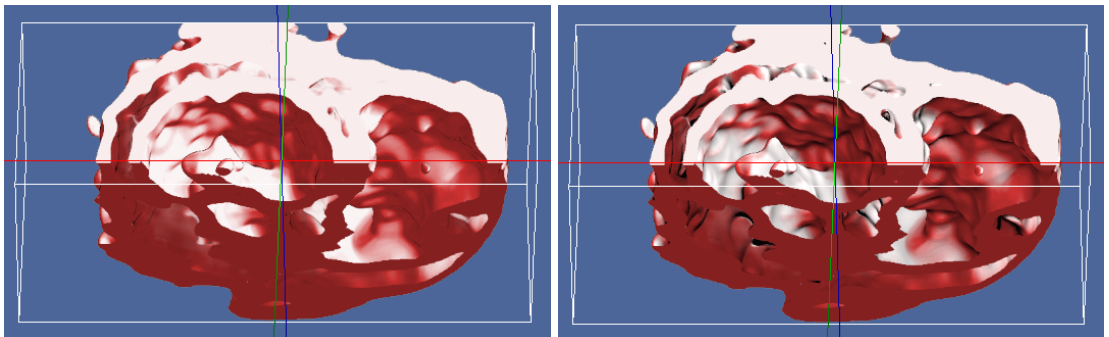


Figure 4.20: Style transfer function rendering with curvature shading, using the same heart-valve subvolume as in fig.(4.18) but with a different view and clip-plane orientation. Left: style transfer function shading only; Right: with the addition of curvature shading (using minimum principal curvature). Curvature shading in this instance can be seen to greatly increase the depiction of shape and texture within the heart. Source: Paul Ross.

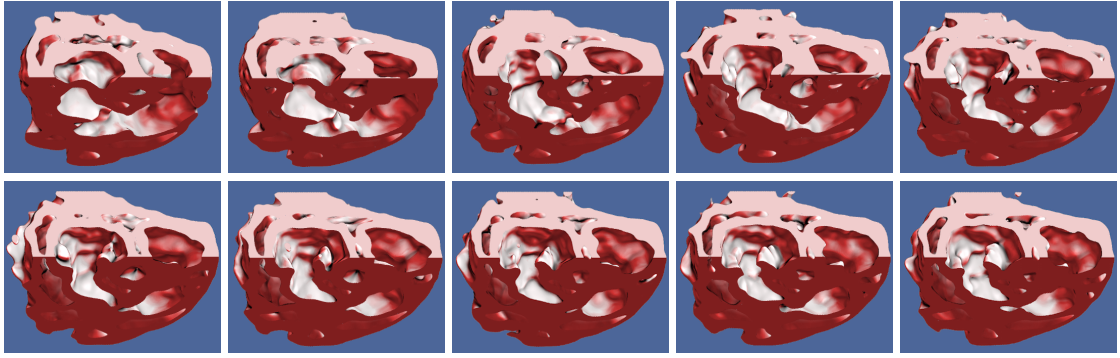


Figure 4.21: Sequence of rendered images from successive sub-volumes in the temporal 4D series (heart-valve dataset). Source: Paul Ross.

4.6.4 Limitations

Noise enhancement

Ultrasound data often possesses a large noise component: this imposes limitations on what can be achieved with curvature-based shading methods. Noise manifests itself as bumps and craters across isosurfaces: these possess high curvature and are thus shaded alongside the desired data by the curvature-shading process. Curvature-based shading essentially amplifies the appearance of any noise present in the dataset.

Solution:

- Improve volumetric filtering for more effective noise suppression, whilst maintaining feature detail.
- Perform curvature shading as a deferred post-processing step: this allows smoothing of the curvature data, helping to minimize the highlighting of noise.

This chapter will focus on the latter of the two methods.

4.7 Deferred curvature shading

Computation of curvature values at each sample point along a given ray can impede performance for simple raycasting-based renderers. One solution is to defer the curvature computation and shading and perform this as an image-space post-processing operation. This has image-space time complexity and is suitable for single-isosurface shading. Hadwiger et al. describe such a method in [45].

Crucially, performing the curvature computation and shading as a post-processing step has an additional advantage: it allows for further processing of the curvature data before applying to the final image render. This can be exploited for the objective of reducing

the effect of curvature shading on any residual noise in the dataset. Smoothing the 2D curvature data using an edge-preserving smoothing step has proved effective in this regard; this will be discussed in the following section.

In summary, the deferred curvature shading approach is characterised by the following:

- Allows for post processing of curvature data, i.e. smoothing or some other operation, prior to final rendering. This is the prime motivation in the context of this chapter.
- Limited to single opaque isosurface: this is ideal for ultrasound applications.

For single opaque isosurfaces, the result is visually identical to that obtained through volumetric curvature computation along each ray since only one sample value is used for the final colour output per ray in each case. In naive raycaster implementations (without empty-space skipping), these values are discarded and constitute wasted computational effort.

4.7.1 Method

Method (in addition to the standard volume rendering pipeline):

1. Compute the required curvature metric on termination of the ray (i.e. when the ray intersects with the opaque isosurface).
2. Store the curvature information in one or more two-dimensional floating-point data buffers (of the same dimension as the output image).
3. Smooth the curvature data. In this application, an edge-preserving filter is used: specifically a Perona-Malik PDE-based filter, but a bilateral or other nonlinear filter could be substituted. Possible future research could evaluate more modern 2D filtering methods such as the *domain transform filter* [148].
4. Composite the curvature data with the image-space volume-render image output. There are multiple methods that can be employed here: curvature transfer functions, direct colour-channel modulation are two possibilities. The latter approach is the main method used for this particular application.

Figure 4.22 depicts an overview of the deferred and smoothed curvature shading pipeline.

4.7.2 Results

In noisy datasets (with a low amount of volumetric smoothing), the smoothed curvature data offers a slight improvement only (over the standard case), as can be seen in figure (4.23).

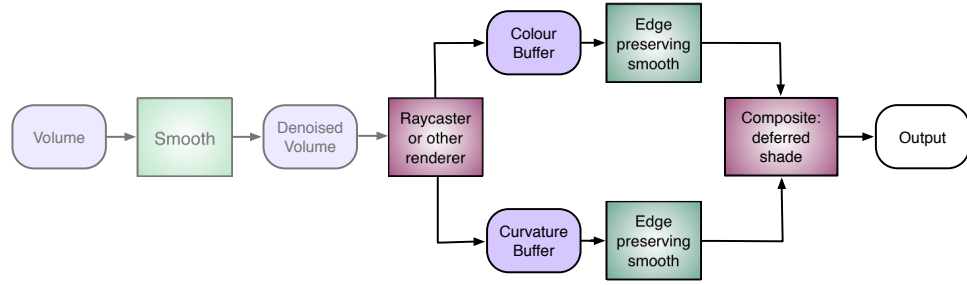


Figure 4.22: Overview of deferred shading pipeline with smoothing stages for 2D curvature and colour buffers. The initial volume smoothing stage is also shown as a pre-processing step: at this stage of development, this is restricted to a simple Gaussian 3D filter. Subsequent chapters will focus on improving this for an optimal balance of feature detail and noise reduction in the output image.

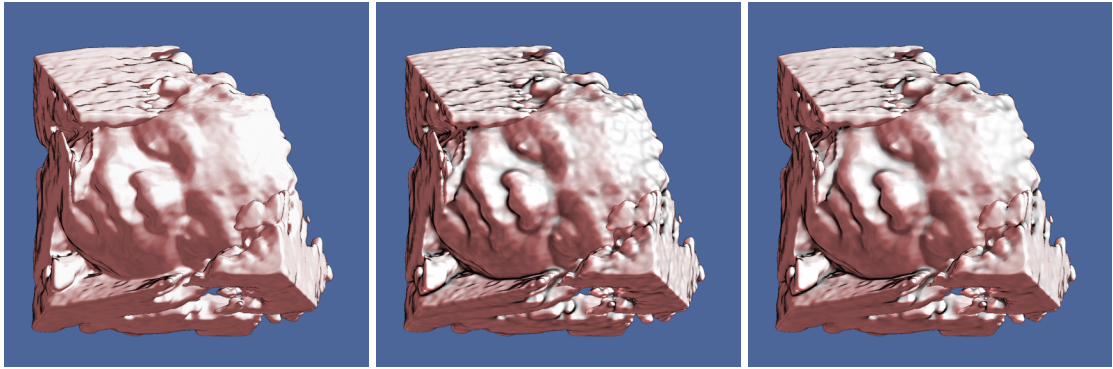


Figure 4.23: Deferred and smoothed curvature shading for keepsake 3D ultrasound: L-R: style transfer function rendering only; with the addition of curvature shading; smoothing the curvature buffer before applying to final colour image buffer. Source: Paul Ross.

Applying in conjunction with a smoother input dataset, the deferred and smoothed curvature shading step is significantly more effective in enhancing features whilst minimising visual noise amplification. Figures (4.24) and (4.25) depict much greater definition in the facial feature boundaries (particularly the mouth); the effect of noise is reduced (see fig.(4.25) for details). This shading approach can also be employed in conjunction with global illumination as shown in figure (4.26).

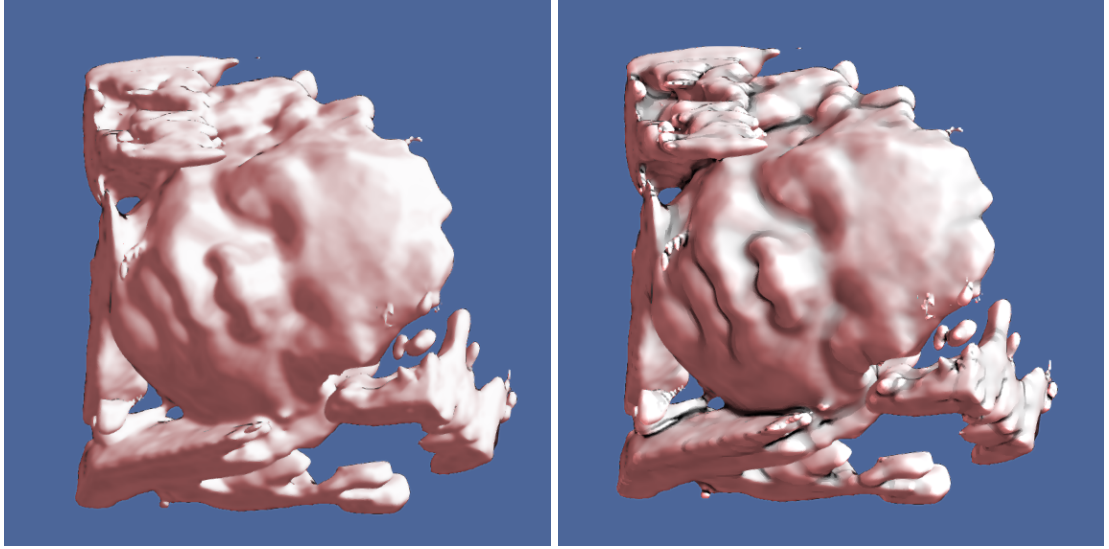


Figure 4.24: Application of deferred and smoothed curvature shading to an ultrasound volume with a higher degree of volumetric Gaussian smoothing applied. Left: style transfer function rendering alone; right: with the addition of the deferred and smoothed shading step. Source: Paul Ross.

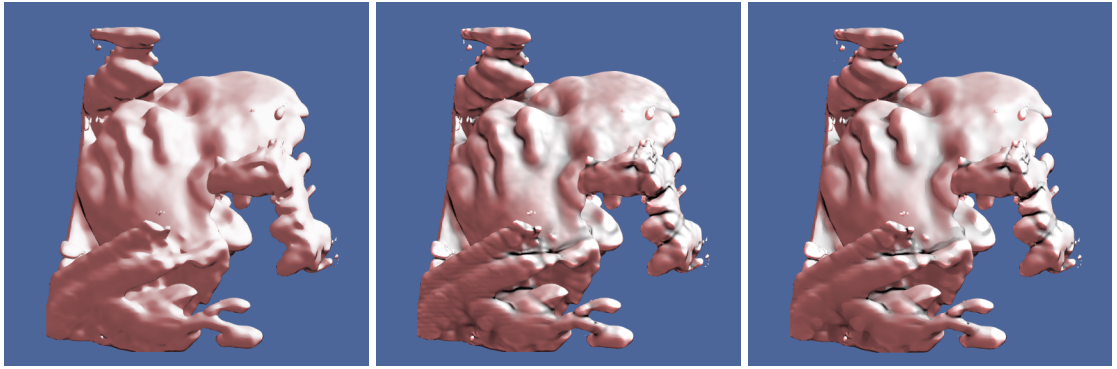


Figure 4.25: The effect of 2D curvature and colour buffer-smoothing on reducing the appearance of noise. L-R: *style transfer function* render with no additional shading; with the addition of deferred curvature shading using the κ_2 metric; with smoothing of the 2D curvature buffer. Source: Paul Ross.



Figure 4.26: Application of deferred and smoothed curvature shading as a post-processing step on a *global illumination* babyface ultrasound rendering. L-R: unsmoothed volume; Gaussian-smoothed volume; with the addition of deferred curvature shading. Note that there is much detail lost after the initial volume smoothing stage; the deferred curvature shading step (with 2D curvature buffer smoothing) attempts to restore some feature delineation in the final output render, whilst minimising the shading and apparent amplification of surface noise. Source: Paul Ross.

4.7.3 Summary

Shading limitations

As previously mentioned, this method is limited to single opaque isosurfaces. It is not applicable when there are any transparent regions; this is usually not an issue with ultrasound data, as a single opaque isosurface rendering is desirable in most cases.

In addition, a post-processing shading method may not always produce the required results by virtue of its position in the rendering pipeline. Full integration with the main lighting or shading model is not possible: curvature shading is applied after the main volume render process is complete. For full integration into the chosen lighting model, ideally curvature shading should be applied to the base or diffuse colour components, *before* any specular highlights are added. The deferred method adds further shading *after* any specular components have been incorporated into the rendering. This can produce unnatural and potentially undesirable effects: shading across specular highlights produces grey bands in valley regions (if the specular highlights are white).

In a purely illustrative context, this may be acceptable, i.e. if the objective is to highlight specific surface features with less regard for a realistic lighting model. In addition, the deferred and smoothed curvature shading method is only really effective when there is a low amount of residual noise from the initial volumetric smoothing stage.

4.8 Performance

As in the performance section of the previous chapter (§3.8), the hardware system and testing conditions consisted of the following: Nvidia GTX680 (with 4GB memory) running on a Xeon workstation (2.4GHz processors, 4 physical cores, 8 virtual ‘cores’ with hyperthreading) with 16GB RAM. The datasets were all rendered using the same transfer function with one spheremap, at identical z-distance with no partial opacity and clipping planes deactivated. All resulting timing data is on a per-frame basis.

Table (4.1) depicts timing data for various shading methods (in the context of several datasets): style transfer functions alone; with the addition of curvature shading, applied at each sample point along each ray; with the addition of curvature shading as a deferred shading operation.

Note the significant increase in performance of the deferred shading method over the volumetric equivalent; this is one clear benefit of taking the deferred approach, even if its application has somewhat limited value in the context of noise suppression in 3D ultrasound.

Dataset	Shading method		
	STF	STF + CS	STF + deferred CS
3D US Babyface (197x126x200, 16 bit)	18.50ms	56.88ms	19.37ms
CT Engine (256x256x256, 8 bit)	19.90ms	42.25ms	20.88ms
CT Head (128x256x256, 8 bit)	33.95ms	100.144ms	34.69ms

Table 4.1: Per-frame rendering time at 512x512 output resolution: comparison of shading methods (STF: style transfer function, normal-based spheremap shading; CS: procedural curvature shading).

4.9 Conclusion

Initially inspired by traditional artistic illustration techniques, curvature-shading methods can effectively simulate shape-depicting lines or artistic shading, and hence are useful in improving the depiction of overall shape; in medical imaging contexts, this can be particularly useful for enhancing the depiction of salient features as these are usually delineated by regions of high isosurface curvature.

It has been demonstrated that curvature-based shading can clearly improve feature delineation in keepsake ultrasound images, both for illustrative styles (via the Style Transfer Function renderer) and in the context of global illumination (GI), over normal/gradient-based/GI shading alone. There are of course, other medical imaging contexts in which curvature-based methods can be applied, but ultrasound presents a particularly difficult problem due to its high inherent noise component. An effective means of reducing noise whilst preserving feature detail is highly desirable in this context; hence keepsake ‘baby-face’ images will be used throughout to demonstrate the efficacy of various improvements and additions to the overall rendering pipeline.

Note that the shading methods outlined in this chapter are independent of lighting and view direction; this is primarily an advantage when used in a feature-enhancing context, as overall shape-depiction is improved regardless and is not a function of either parameter.

However, there is a disadvantage to decoupling entirely from the initial shading or lighting stage: curvature shading is applied (either volumetrically along a ray or as a deferred post-processing step) after all lighting computations. Ideally, the curvature-shading step should be performed before any specular components are added to avoid shading over them (this produces a greying effect which is often undesirable).

It should be noted that view-dependent curvature-shading also has applications: regularization of normal-based contours [3][5], and *apparent ridges* [56] are two examples.

Whilst effective at feature-demarcation, an undesirable side-effect of curvature-based shading methods is the highlighting of noise. To a certain degree the effect of noise on the final image can be mitigated by performing the curvature shading as a post-processing step: this

allows the curvature data to be collected on ray-termination, and smoothed before the final image composition. Together with colour image-buffer smoothing, this can greatly reduce the highlighted-noise component in the final image.

Deferred curvature shading clearly offers a performance improvement over a volumetric variant, although its application is limited to renderings with full opacity throughout. The ultimate project goal is to maximize feature detail whilst minimizing the shading of noise (or visible noise in the output image). Deferring the curvature shading stage and smoothing the curvature data buffer is effective in this regard, but the end-result could still be improved upon. A more effective means of volumetric filtering of the input dataset is required; currently, Gaussian filtering or feature-preserving Perona-Malik (simple anisotropic diffusion, [149][150]) or bilateral filters [69] are employed to reduce noise. The former is poor at preserving features; the latter two methods result in the retention of feature detail at the expense of effective noise-reduction. A more effective *curvature-based* volumetric filter, which provides the optimal balance between noise reduction and feature-preservation will be discussed in the following chapter; this allows for a much greater fidelity of output image.

Chapter 5

Further reduction of noise: effective volumetric filtering

Curvature-based shading methods have been shown to be effective for increasing definition around salient features, with particular effectiveness when applied in the context of ultrasound keepsake images. Unfortunately, this shading is also effective at highlighting any residual noise inherent in the dataset. As mentioned in the previous chapter, a major topic of interest for this research project is the maximisation of detail for salient features whilst minimising the effect of noise: in practice, these are competing requirements. As the research progresses, this goal has become increasingly apparent in its significance; indeed, whether considering illustrative rendering styles or not, its pursuit has become a primary objective. The use of curvature-based shading methods, involving a smoothing of curvature data prior to final shading, can provide a good solution but is far from optimal. In order to pursue this objective even further, a more effective method for *volumetric* filtering is required: this is the initial noise reduction process performed on the 3D *raw dataset** prior to any visualisation steps.

In the context of this project, volume filtering has been limited so far to simple Gaussian smoothing or the feature-preserving Perona-Malik and bilateral filters. These latter two methods employ functions of first derivatives (the bilateral filter implicitly) to control the amount of local smoothing in a given neighbourhood: this is a simple extension of their 2D application in image processing, where features are delineated by edge-lines (areas of high gradient). In volumetric contexts, features are delineated by regions of high curvature: this would suggest that an effective volumetric feature-preserving filtering approach could make use of second derivatives. Indeed, curvature in a 3D context is the natural analogue of gradient in 2D for feature demarcation.

*In practice, certain ultrasound scanners may perform a small and often undocumented noise reduction step after initial data acquisition and reconstruction: this step results in what will be considered to be the *raw dataset* for purposes of further processing or visualisation.

Mean-curvature-motion [87] is a simple curvature-based PDE filter, but isn't particularly effective with regard to feature-preservation. Extending the concept of anisotropic diffusion to volume data, more sophisticated feature-preserving PDE-based schemes include those described by Krissian in [9], Carmona in [10]. Hossain and Möller build upon these latter two methods in [8]; their approach will be discussed further in this chapter. Other methods include *guided filtering* (He et al.) [86] and *anisotropic diffusion of surface normals* (Tasdizen et al.) [106].

Volumetric smoothing methods can essentially be subdivided as such:

- Neighbourhood-based averaging methods: these include linear convolution-based filters such as the standard Gaussian, in addition to nonlinear methods such as the bilateral filter.
- PDE-based methods: these include simple gradient-based Perona-Malik implementations in addition to anisotropic geometric flows of order two and higher.
- Other methods, including nonlocal means and machine learning based techniques.

Several attempts have been made to illustrate how these first two fundamental approaches are linked [82][84][151][83][69].

This chapter will focus on the two essential criteria required for an optimal noise-reduction solution: the preservation of features, and the maximal suppression of surface noise. Time-performance is an additional concern: a fast method is a requirement for 4D real-time streaming ultrasound applications. The solution developed in this chapter will meet the two essential criteria, whilst offering very high performance.

What follows is a brief description of the basic Perona-Malik and bilateral filters with respective results (as used up to this point in this research project), expanding to curvature-based anisotropic diffusion filters, in particular that which is described by Hossain and Möller in [8]. Results from the latter PDE-based method are presented, and through discussion of its inherent limitations, a novel PDE-based method is developed which aims to optimally satisfy the fundamental criteria described above.

5.1 The Gaussian filter and inherent limitations

A naive approach to volume filtering might consider a simple Gaussian smoothing filter: this is usually implemented as a *neighbourhood filter*, where each pixel or voxel is processed by considering a weighted average of a local neighbourhood of a given size. The pixel or voxel in the same spatial position in the output dataset is effectively replaced with that weighted average from the input, in what is essentially a convolution operation with

a kernel of Gaussian-derived multiplier weights.[†] A Gaussian filter specifies weighting multipliers as a Gaussian function of distance from the input pixel or voxel: hence dataset values further from the input pixel have correspondingly less influence. It is possible to specify other functions of distance, but the Gaussian function has some special properties: it is an eigenfunction under the Fourier Transform, and a Green's function of the diffusion equation. In addition, repeated iterations of the filter with a kernel of small radius is equivalent to fewer iterations with a larger kernel.

The convolution operation can be described as:

$$p(x) = \frac{1}{W} \sum_{x_i \in \Omega} f(x_i) g(\|x_i - x\|) \quad (5.1)$$

for every data point $f(x)$ at position x in the input dataset. Ω is the local neighbourhood around each input data point $f(x)$, and is defined by the radius of the Gaussian filter kernel. $p(x)$ is the position-equivalent data point in the processed dataset. g represents the Gaussian function, and $\frac{1}{W}$ is the normalisation factor, where $W = \sum_{x_i \in \Omega} g(\|x_i - x\|)$.

It should be noted that constructing a filtered dataset using the above convolution is equivalent to numerically solving the diffusion PDE:

$$\frac{\partial f}{\partial t} = c \nabla^2 f \quad (5.2)$$

where f represents the input dataset, and c is a scalar constant. Note that $\nabla^2 f$ is rotationally invariant and the diffusion is isotropic.

The convolution method will be used throughout this section to obtain results; for 3D volume data, an $O(N^3 n^3)$ operation[‡] can be replaced an $O(N^3 n)$ one by splitting it into three separate convolutions with a one-dimensional kernel.

5.1.1 Limitations

In the context of 2D image processing, simple neighbourhood-averaging linear filters (such as the standard Gaussian filter) perform *isotropic* smoothing: at a given centre pixel, the amount of smoothing performed is equal in all directions, regardless of local differential features. Smoothing is thus performed across edges (areas of high gradient). Whilst such methods are effective at reducing noise, they have the additional consequence of reducing feature boundary detail, which may be undesirable (see fig.(5.1)).

[†]Strictly speaking, the correct term is a *correlation*; however, since the Gaussian filter kernel is symmetric, this is equivalent to a convolution.

[‡]assuming a cuboid volume of size N^3 voxels with a convolution kernel of diameter n .

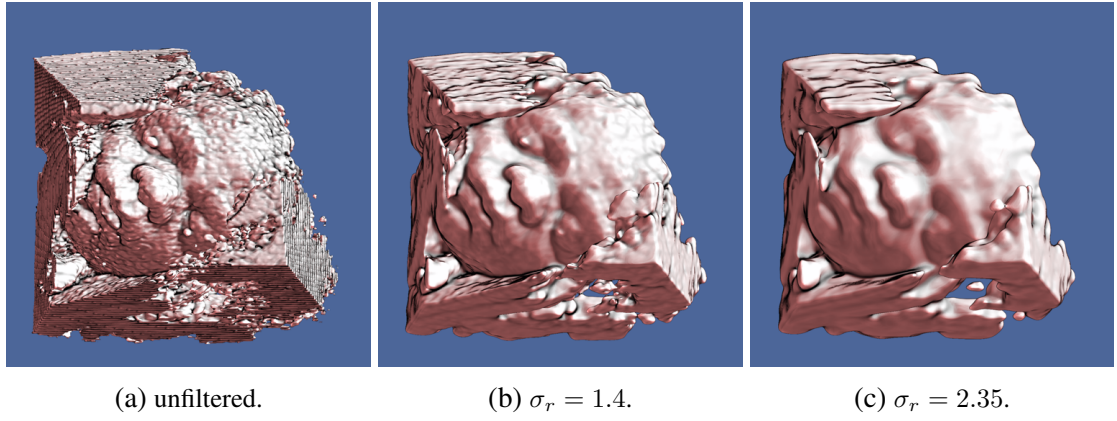


Figure 5.1: Gaussian-smoothed obstetric ultrasound dataset rendered with style transfer functions and supplementary curvature shading: increasing the level of noise reduction results in a loss of feature detail. Source: Paul Ross.

5.2 Feature-preserving smoothing

Numerous methods had been devised as a means of restricting smoothing into distinct piecewise near-homogeneous regions delineated by edges, thus preserving image feature detail. PDE-based nonlinear-diffusion filters such as *Perona-Malik* [149] attempt to limit smoothing across edges; this and more sophisticated examples will be discussed in this chapter.

Neighbourhood filters such as the *bilateral* filter attempt to perform edge-preserving smoothing using a different approach: attenuating a standard isotropic Gaussian filter with an penalizing *range component*, based on the pixel intensities themselves. This is essentially an additional multiplier weighted according to a Gaussian function of the difference in intensity from the centre pixel. Hence smoothing is constrained across edges, since these are regions of high gradient. The effect is similar to that of Perona-Malik diffusion; indeed the two seemingly fundamentally different approaches are in fact related: the bilateral filter is a special case of Perona-Malik diffusion [82]. Bilateral filtering with large support kernels can be considered to be equivalent to a sum of Perona-Malik diffusion operations at separate scales; furthermore, other links have been observed (see §2.9).

5.2.1 Nonlinear and anisotropic diffusion

Perona Malik

Recall the case of isotropic diffusion (equivalent to convolution-based Gaussian smoothing):

$$\frac{\partial f}{\partial t} = c \nabla^2 f = \text{div}(c \nabla f) \quad (5.3)$$

where c is a scalar constant.

Replacing the constant term with a function $g(x, y, f, \nabla f)$ of the gradient magnitude, $g(\|\nabla f\|)$ gives:

$$\frac{\partial f}{\partial t} = \text{div}(g(\|\nabla f\|) \nabla f) \quad (5.4)$$

expanding via the product and chain rules:

$$\begin{aligned} \frac{\partial f}{\partial t} &= g(\|\nabla f\|) \nabla^2 f + \nabla(g(\|\nabla f\|)) \cdot \nabla f \\ &= g(\|\nabla f\|) \nabla^2 f + (g'(\|\nabla f\|) \nabla(\|\nabla f\|)) \cdot \nabla f \end{aligned} \quad (5.5)$$

Since $\nabla(\|\nabla f\|) = \mathbf{H} \nabla f / \|\nabla f\|$ (see §A.5.1), further expansion of eq. (5.5) gives an expression in terms of the Hessian matrix \mathbf{H} , gradient ∇f and Laplacian $\nabla^2 f$:

$$\begin{aligned} \frac{\partial f}{\partial t} &= g(\|\nabla f\|) \nabla^2 f + \left(g'(\|\nabla f\|) \frac{\mathbf{H} \nabla f}{\|\nabla f\|} \right) \cdot \nabla f \\ &= g(\|\nabla f\|) \nabla^2 f + g'(\|\nabla f\|) \frac{\nabla f^T \mathbf{H} \nabla f}{\|\nabla f\|} \end{aligned} \quad (5.6)$$

Let η be a unit vector in the direction of the gradient ∇f . Note that $\nabla f^T \mathbf{H} \nabla f / \|\nabla f\|^2$ is the directional 2nd derivative of f in the direction of η , denoted by $f_{\eta\eta}$. Hence eq. (5.6) can be re-written as:

$$\frac{\partial f}{\partial t} = g(\|\nabla f\|) \nabla^2 f + g'(\|\nabla f\|) \|\nabla f\| f_{\eta\eta} \quad (5.7)$$

The first term can be considered the isotropic *diffusion component* and the latter the *shock component* of the flow [91]. Further decompositions are indeed possible and will be discussed in §5.2.1.

The g' term also allows for backward diffusion, where edges are effectively sharpened [110]; in the continuous case, this would result in an ill-posed system with a high sensitivity to gradient [152]. However, the commonly used finite-difference spatial-discretisation scheme (proposed in [149]) provides a degree of regularisation [102][87], resulting in a

well-posed system with stable edges within the stability constraints of the chosen temporal discretisation.

The function g (known as a *stopping function* or *diffusivity function*) takes the form of a monotonically decreasing function between 0 and 1, and is employed to control the overall amount of diffusion as a function of gradient magnitude: diffusion is attenuated in high-gradient (edge) regions. In the strict terminology of Weickert [87], this constitutes a locally-constrained *isotropic diffusion*, rather than pure anisotropic, since g is a scalar function and not a diffusion tensor. For the purposes of this document, the term *nonlinear diffusion* will be used to generally refer to Perona-Malik and other nonlinear diffusion PDE-based systems. Diffusion tensor-based and orthogonal-component 2nd order geometric flows will be termed *anisotropic diffusion*.

Details of stopping function g

The requirements for $g(\|\nabla f\|)$ are such that it should:

- Allow isotropic smoothing in near-homogeneous or intra-feature regions (low gradient).
- Constrain smoothing in edge or feature-boundary regions (high gradient).
- Consist of a smooth monotonically decreasing step-like function with $g(0) = 1$.

Perona and Malik proposed two possible forms for g :

$$g(\|\nabla f\|) = e^{-(\|\nabla f\|/K)^2} \quad (5.8)$$

and

$$g(\|\nabla f\|) = \frac{1}{1 + \left(\frac{\|\nabla f\|}{K}\right)^2} \quad (5.9)$$

where K in both cases is a parameter, with $K > 0$.

Figure (5.2) depict the two variants of g as specified by Perona and Malik; figure (5.3) shows how the derivatives of both forms have a large negative minimum for a specific input of $\|\nabla f\|$: this can result in shocks via backward diffusion (consider eq.(5.7)) and is responsible for the edge-sharpening/enhancing effects and stairstep artifacts seen in the evolution of the Perona-Malik PDE.

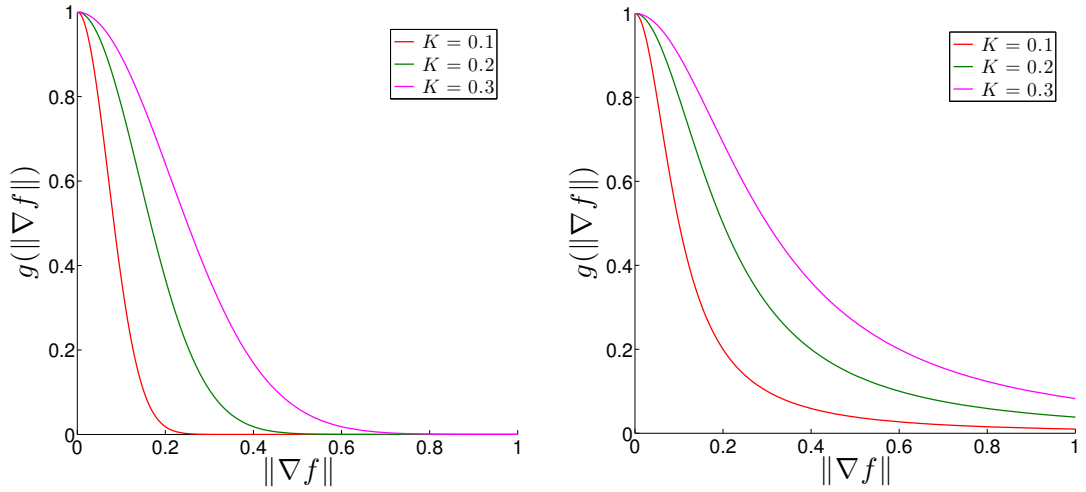


Figure 5.2: Stopping functions $g(\|\nabla f\|)$ for various values of parameter K . Left and right subfigures represent equations (5.8) and (5.9) respectively.

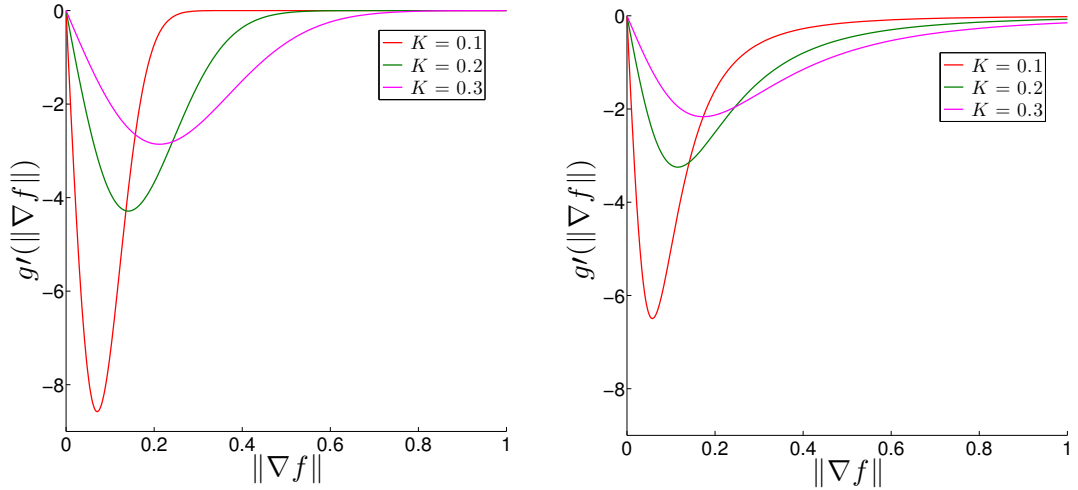


Figure 5.3: $g'(\|\nabla f\|)$ for both variants of stopping function; large negative values can result in reverse diffusion and shock formation.

A comparison of results produced by 2D Perona-Malik diffusion and isotropic Gaussian smoothing can be seen in figure (5.4): note how the edge-regions in the Perona-Malik case are not only preserved, but undergo a degree of sharpening due to the g' component.

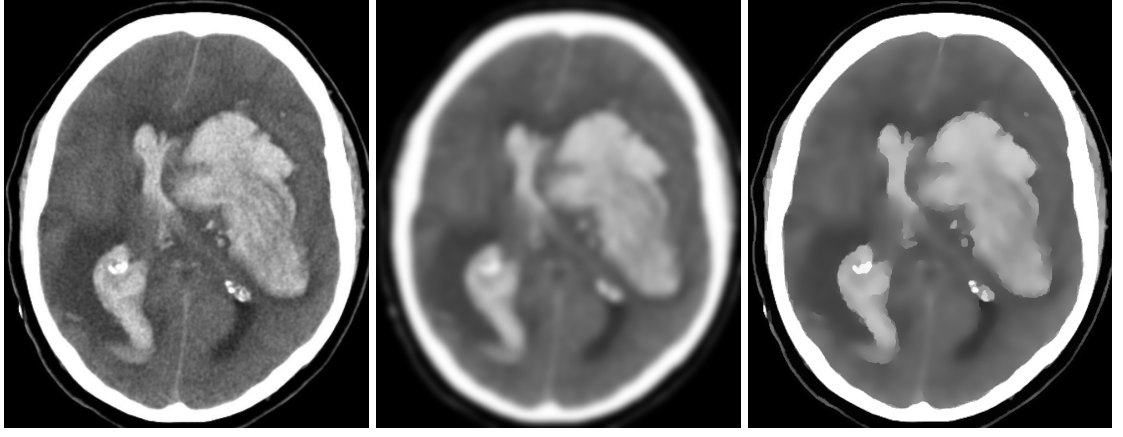


Figure 5.4: Comparison of Perona-Malik diffusion (with stopping function defined in eq.(5.8)) and standard Gaussian smoothing in the context of a 2D CT-slice image. Left to right: original, Gaussian-smoothed, Perona-Malik smoothed. Source: Paul Ross.

Decomposition of diffusion into orthogonal components

Consider eq.(5.7) again: Alvarez et al. [153] (also Carmona et al. [10] and Didas, Weickert [110]) further expand on this by decomposing the Perona-Malik diffusion into two orthogonal directions: in the direction of the gradient, and its perpendicular, by considering directional second derivatives $f_{\eta\eta}$ and $f_{\xi\xi}$ respectively:

$$f_{\eta\eta} = \frac{\nabla f^T \mathbf{H} \nabla f}{\|\nabla f\|^2} \quad (5.10)$$

$$f_{\xi\xi} = \frac{(\nabla f^\perp)^T \mathbf{H} \nabla f^\perp}{\|\nabla f\|^2} \quad (5.11)$$

Expanding eqs. (5.10) and (5.11) gives:

$$f_{\eta\eta} = \frac{f_{xx}f_y^2 + 2f_xf_yf_{xy} + f_{yy}f_x^2}{f_x^2 + f_y^2} \quad (5.12)$$

$$f_{\xi\xi} = \frac{f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2}{f_x^2 + f_y^2} \quad (5.13)$$

Adding eqs. (5.12) and (5.13):

$$\begin{aligned} f_{\xi\xi} + f_{\eta\eta} &= \frac{f_{xx}(f_x^2 + f_y^2) + f_{yy}(f_x^2 + f_y^2)}{f_x^2 + f_y^2} \\ &= f_{xx} + f_{yy} \\ &= \nabla^2 f \end{aligned} \quad (5.14)$$

i.e. the sum of these orthogonal directional second derivatives is equal to the Laplacian[§].

Hence, eq. (5.7) can be expressed as:

$$\begin{aligned}\frac{\partial f}{\partial t} &= g(\|\nabla f\|)(f_{\eta\eta} + f_{\xi\xi}) + g'(\|\nabla f\|)\|\nabla f\|f_{\eta\eta} \\ &= g(\|\nabla f\|)f_{\xi\xi} + (g(\|\nabla f\|) + g'(\|\nabla f\|)\|\nabla f\|)f_{\eta\eta}\end{aligned}\quad (5.15)$$

Generalised adaptive smoothing using orthogonal components: towards an effective 3D filter formulation

Considering eq. (5.14), isotropic diffusion (eq.(5.3)) can be expressed as:

$$\frac{\partial f}{\partial t} = c(f_{\eta\eta} + f_{\xi\xi}) \quad (5.16)$$

where c is a scalar constant.

The anisotropic extension of eq.(5.16) allows for separate control of diffusion in both gradient and isophote directions; this forms the basis for the general case of orthogonal-component diffusion for adaptive smoothing, as described by Carmona et. al [10]. This PDE (eq.(5.17)) is no longer of the Perona-Malik type and is considered a more general class of nonlinear diffusion equation; indeed, it is potentially (given suitable parameters) truly anisotropic (using Weickert's definition)[87] for $a \neq b$, albeit without explicit use of a diffusion tensor.

$$\frac{\partial f}{\partial t} = c(af_{\eta\eta} + bf_{\xi\xi}) \quad (5.17)$$

where c , a and b are scalar-valued functions.

5.2.2 Nonlinear diffusion: extension to 3D

Perona-Malik

Gerig et al. [150] demonstrated that the Perona-Malik diffusion process can be extended to 3D datasets; the practicalities of implementing such an extension are relatively trivial.

The Perona-Malik equation (5.7) is applicable to both 2D and 3D scalar fields: this equation specifies an isotropic diffusion component in terms of the Laplacian $\nabla^2 f$ and a shock

[§]More generally, it can be shown that for $f : \mathbb{R}^n \mapsto \mathbb{R}$, the sum of any n mutually-orthogonal directional second derivatives forming a basis for \mathbb{R}^n is equal to $\nabla^2 f$, i.e. the Laplacian is *rotationally invariant*.

component in terms of the second derivative $f_{\eta\eta}$ in the direction of the gradient vector [91].

For $f : \mathbb{R}^3 \mapsto \mathbb{R}$:

$$\begin{aligned}\frac{\partial f}{\partial t} &= g(\|\nabla f\|)\nabla^2 f + g'(\|\nabla f\|)\|\nabla f\|f_{\eta\eta} \\ &= g(\|\nabla f\|)(f_{xx} + f_{yy} + f_{zz}) + g'(\|\nabla f\|)\|\nabla f\|f_{\eta\eta}\end{aligned}\quad (5.18)$$

Note that the sum of the second derivatives in the direction of any set of mutually orthogonal basis vectors in \mathbb{R}^3 is equal to the Laplacian (due to its property of rotational invariance). Hence:

$$\frac{\partial f}{\partial t} = g(\|\nabla f\|)(f_{e_1 e_1} + f_{e_2 e_2} + f_{\eta\eta}) + g'(\|\nabla f\|)\|\nabla f\|f_{\eta\eta} \quad (5.19)$$

where vectors e_1 and e_2 are any two orthogonal basis vectors spanning the tangent plane, and $f_{e_1 e_1}$ and $f_{e_2 e_2}$ are the 2nd derivatives in their respective directions. Since η is orthogonal to the tangent plane, $\{e_1, e_2, \eta\}$ forms a basis for \mathbb{R}^3 .

Rearranging (similar to eq.(5.15) for the 2D case):

$$\frac{\partial f}{\partial t} = g(\|\nabla f\|)(f_{e_1 e_1} + f_{e_2 e_2}) + (g(\|\nabla f\|) + g'(\|\nabla f\|)\|\nabla f\|)f_{\eta\eta} \quad (5.20)$$

From eq.(5.20), it can be observed that smoothing is performed isotropically in the tangent plane, with an additional shock component in the direction of the gradient vector.

Limitations

Isophote-direction smoothing in the 2D case of Perona-Malik diffusion is effective at preserving features, since the features boundaries tend to be defined by isophotes. In the 3D case, the equivalent component of the Perona Malik equation was shown to perform isotropic smoothing across the tangent-plane at points on any given isosurface (see eq.(5.20)). Furthermore, as with the 2D case, the Perona-Malik has a diffusion component in the gradient direction (see eq.(5.7)); i.e. it is not strictly confined to directions orthogonal to it. Indeed, an additional potential consequence is the tendency to form stepping artifacts (see figs.(5.5), (5.6)): these are a result of the g' term in the decomposition seen in eq.(5.7). Large negative values for g' can produce a shock effect through reverse diffusion (via negative values for the normal component: see eq.(5.20) and fig.(5.3)). See also [153][84][154].

In the context of 2D images, edge regions (regions of high gradient) constitute feature boundaries. In 3D datasets however, feature delineation is often defined by regions of

high curvature within a given isosurface [106][155], rather than at high-gradient boundaries across isosurfaces (although both may be a factor in certain cases). Hence Perona-Malik diffusion may not be sufficient to effectively preserve feature detail in 3D; true feature-preserving smoothing will require diffusion along lines of curvature on any given isosurface. Indeed, Tasdizen et al. [106] describe a method which may be considered a true volumetric analogue of Perona-Malik diffusion with its inherent feature-enhancing properties: they utilise 4th order geometric flows. However, 2nd order geometric flows are sufficient to address the issue of feature preservation; this is the approach taken by Krissian et al. [9] by way of Carmona et al. [10] and is further explored by Hossain and Möller in [8].

Hence, the directional orthogonal-component formulation in [10](see eq.(5.15)) suggests one possible solution to these issues: extend this 2D diffusion PDE to the 3D case with diffusion performed along three separately controllable orthogonal directions. This is indeed the approach taken by Hossain and Möller in [8] and will be discussed further in §5.4.

Results

Two datasets were volumetrically filtered using 3D Perona-Malik diffusion for ten iterations (with $K = 0.37$): the project-reference ultrasound ‘babyface’ dataset, and the standard engine cylinder-head CT dataset. The results were rendered using Style Transfer Functions, using a shading map chosen for maximum noise and artifact highlighting, and appropriate rendering style. Moreover, note that curvature-based shading is employed throughout: this greatly amplifies the visual presence of any artifacts or noise. In both examples, stopping function $g(\|\nabla f\|) = 1/(1 + (\frac{\|\nabla f\|}{K})^2)$ was used.

The Perona-Malik results are presented in comparison to those achieved using standard isotropic Gaussian smoothing. In both Perona-Malik cases, there is less smoothing across feature boundaries, but clear stepping artifacts can be observed.

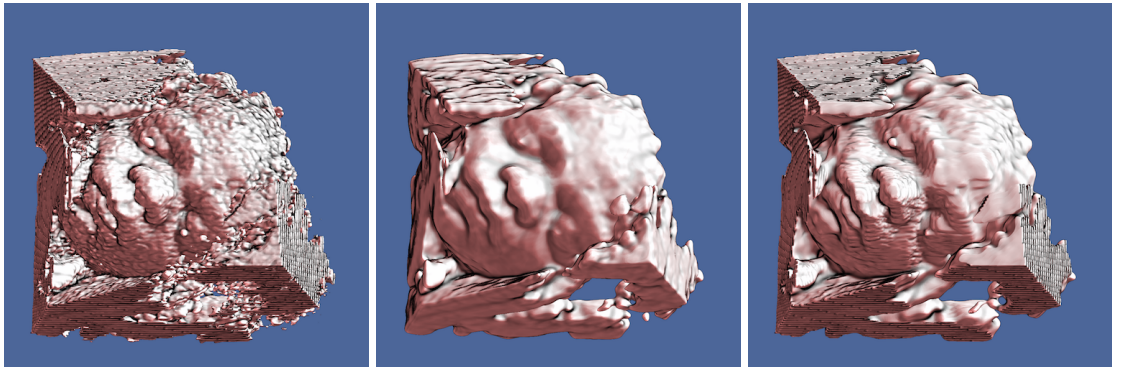


Figure 5.5: Pronounced step artifacts from 10 iterations of the 3D Perona-Malik diffusion process in an obstetric ultrasound volumetric dataset. Left-Right: original unfiltered volume; Gaussian-smoothed volume; Perona-Malik smoothed volume. Source: Paul Ross.

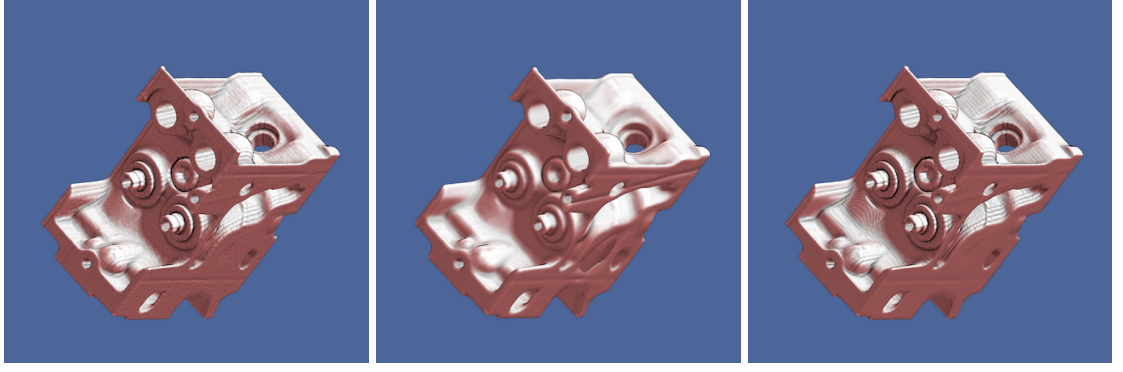


Figure 5.6: Pronounced step artifacts from 10 iterations of the 3D Perona-Malik diffusion process in the engine test dataset. Left-Right: original unfiltered volume; Gaussian-smoothed volume; Perona-Malik smoothed volume. Source: Paul Ross.

3D Anisotropic diffusion

Tasdizen et al. [106][155] stipulate a minimum requirement of 2nd order differential properties for geometric flows for effective feature preservation in a volumetric filtering context: geometric flows using curvature will be discussed in §5.3 and will form the basis for an improved filtering method developed from a 3D extension to the directional 2nd derivatives form of the general anisotropic diffusion equation (eq.(5.17)).

5.2.3 Bilateral filtering (neighbourhood filter)

An alternative non-iterative methodology for performing feature-preserving smoothing can be formulated by imposing constraints on an otherwise isotropic neighbourhood-averaging filter. Tomasi and Manduchi describe such an approach in [7], and term their method a *Bilateral filter*, due to its two components: the isotropic smoothing function, and a penalising function which attenuates the action of the isotropic filter where required.

The most common implementation consists of a standard isotropic Gaussian smoothing filter (effectively a Gaussian function of the difference in position in the local neighbourhood from the current pixel being averaged, and thus known as the *domain component*), penalised by a second Gaussian function of the difference in intensity from the current pixel (the *range component*). Thus isotropic smoothing can be limited to near-homogeneous regions; smoothing is thus not performed across edges, i.e. areas of high intensity difference (gradient).

Generally, (in n -dimensions) for each pixel position \mathbf{x} in input image (or scalar field) f and output f_{out} , with filter-kernel spatial-domain Ω :

$$f_{out}(\mathbf{x}) = \frac{1}{W} \sum_{\mathbf{x}_i \in \Omega} f(\mathbf{x}_i) g_{\sigma_d}(\|\mathbf{x}_i - \mathbf{x}\|) g_{\sigma_r}(\|f(\mathbf{x}_i) - f(\mathbf{x})\|) \quad (5.21)$$

where

$$g_{\sigma_d}(\|\mathbf{x}_i - \mathbf{x}\|) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma_d^2}} \quad (5.22)$$

$$g_{\sigma_r}(\|f(\mathbf{x}_i) - f(\mathbf{x})\|) = e^{-\frac{\|f(\mathbf{x}_i) - f(\mathbf{x})\|^2}{2\sigma_r^2}} \quad (5.23)$$

are Gaussian functions of standard-deviation σ_d and σ_r respectively. $\frac{1}{W}$ is the normalisation factor, with

$$W = \sum_{\mathbf{x}_i \in \Omega} g_{\sigma_d}(\|\mathbf{x}_i - \mathbf{x}\|) g_{\sigma_r}(\|f(\mathbf{x}_i) - f(\mathbf{x})\|) \quad (5.24)$$

$\|\mathbf{x}_i - \mathbf{x}\|$ represents the Euclidean distance (denote by d_1) between the centre sample point and those in the local neighbourhood (filter kernel); $g_{\sigma_d}(d_1)$ is the *domain component*.

$\|f(\mathbf{x}_i) - f(\mathbf{x})\|$ is the *photometric distance* (denote by d_2) between the centre sample point and those in the local neighbourhood (filter kernel), i.e. the difference in intensity. $g_{\sigma_r}(d_2)$ is the *range component* of the bilateral filter.

The bilateral filter kernel can be visualised as a truncated isotropic kernel, where attenuation occurs across a region of high gradient (an edge). Figure (5.7) illustrates the process of applying a neighbourhood weighted-averaging filtering operation using a bilateral filter-kernel on a noisy step-function defined over a 2D domain (represented as a heightmap). Note that the middle image depicts the kernel as computed and applied over the upper part of the step function, such that the original function values on the lower step have little influence on the filtered values in the upper step. When computing filtered values on the lower step, the kernel would appear flipped across the step line (with reference to the middle image). Thus the filtering operation can produce piecewise-homogeneous regions delineated by high-gradient edges (see fig.(5.8)).

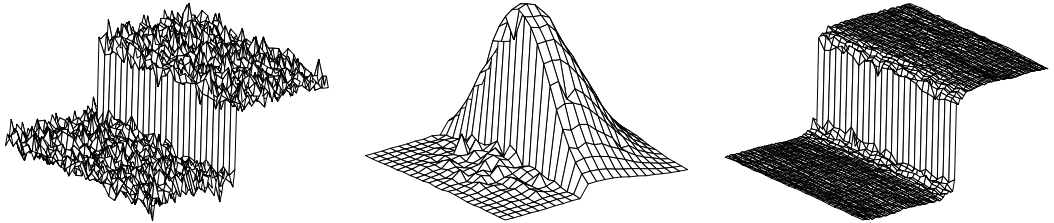


Figure 5.7: Bilateral filter kernel (centre): essentially an attenuated Gaussian kernel. L-R: step function with added noise; bilateral filter kernel, filtered output. Figure from Tomasi, Manduchi [7].

Most discussions of the bilateral filter in the current literature focus on a 2D image-processing context. A component-explicit form of (5.21) that can be translated directly to code can be expressed as:

$$f_{out} = \frac{1}{W} \sum_{i=-r}^r \sum_{j=-r}^r f(x_i, y_j) g_{\sigma_d}(d_1) g_{\sigma_r}(d_2) \quad (5.25)$$

with

$$g_{\sigma_d}(d_1) = e^{-\frac{\|x_i - x\|^2 + \|y_j - y\|^2}{2\sigma^2}} \quad (5.26)$$

$$g_{\sigma_r}(d_2) = e^{-\frac{\|f(x_i, y_j) - f(x, y)\|^2}{2\sigma^2}} \quad (5.27)$$

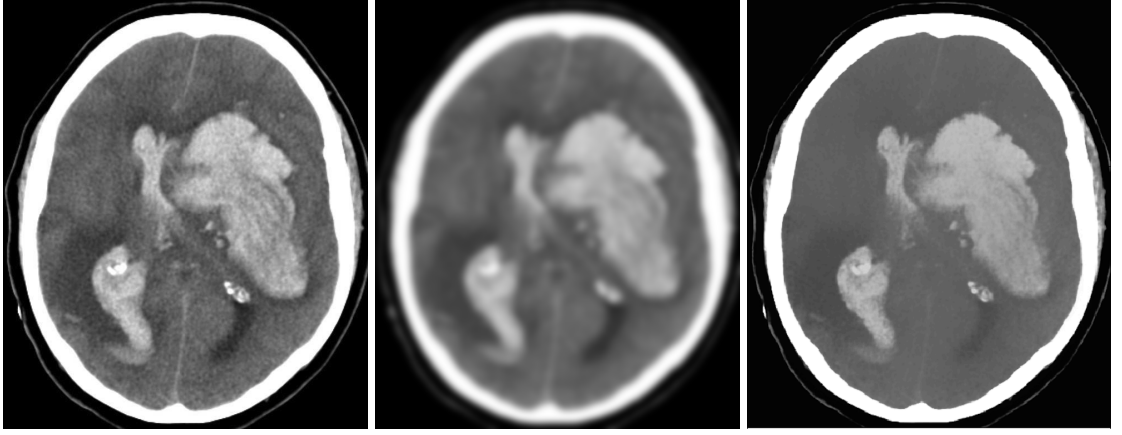


Figure 5.8: Comparison of bilateral-filter and standard Gaussian smoothing in the context of a 2D CT-slice image. Left to right: original, Gaussian-smoothed, bilateral-filtered image. Source: Paul Ross.

Possible advantages

The bilateral filter was conceived as a non-iterative method: a large spatial support replaces the requirement for multiple iterations of a PDE with a five-point stencil for 2D spatial discretisation[¶]

This may offer advantages over iterative PDE-based method in some circumstances, albeit with some caveats:

- A bilateral filter with a large kernel has a larger spatial support than a PDE stencil: PDE's cannot diffuse across edges whilst keeping the edges intact [69]. This can provide a more reliable solution for some applications (e.g. HDR lighting [70]), but may be of less consequence for noise reduction applications. Moreover, filter kernels with a large radius replace the requirement for multiple iterations at radius 1 (as in the case of PDE-based diffusion); due to the time-complexity of full-support non-separable filters ($O(N^4)$ for an N^2 image), there may be a poorer performance overall with regards to execution time unless an optimised approximation is used.

[¶]Note: a 2D nine-point stencil can be considered equivalent to a neighbourhood averaging filter of radius one with a specific set of weights.

- Bilateral filters are less prone to stairstep artifacts than Perona-Malik-based PDE methods [70]. Results in fig.(5.10) concur with this; multiple iterations however can result in the formation of such artifacts (see fig.(5.11)).
- Whilst the full-support one-iteration bilateral filter is relatively slow, there is the potential for a performance gain over iterative methods using an optimised approximation. Examples of different approaches can be found in [156],[73],[75],[76]. Note that simple truncation of the filter kernel (i.e. reduction of the radius) will significantly reduce the execution time: this will reduce the spatial support but is usually sufficient for noise-reduction purposes [69].

Disadvantages

The primary disadvantage of the bilateral filter is its poor performance with regard to speed of execution. Since the filter kernels are nonlinear and shift-invariant, it is not possible to employ the same optimisation methods that are appropriate for the Gaussian smoothing (separable kernels without resultant artifacts, frequency-domain processing). This is of particular significance when extending the bilateral filter to a 3D volumetric context ($O(N^6)$ for an N^3 volume). However, as mentioned above, it should be noted that several optimised approximations do exist.

Extension to 3D

Extending the bilateral filter to 3D is a simple process [77]. Equation (5.28) is a component-explicit depiction of eq.(5.21) which translates directly to a triple-loop construction when realised in code. For a filter kernel of radius r from the

$$f_{out} = \frac{1}{W} \sum_{i=-r}^r \sum_{j=-r}^r \sum_{k=-r}^r f(x_i, y_j, z_k) g_{\sigma_d}(d_1) g_{\sigma_r}(d_2) \quad (5.28)$$

where,

$$g_{\sigma_d}(d_1) = e^{-\frac{\|x_i - x\|^2 + \|y_j - y\|^2 + \|z_k - z\|^2}{2\sigma_d^2}} \quad (5.29)$$

$$g_{\sigma_r}(d_2) = e^{-\frac{\|f(x_i, y_j, z_k) - f(x, y, z)\|^2}{2\sigma_r^2}} \quad (5.30)$$

$$(5.31)$$

$\frac{1}{W}$ is the normalizing factor, with:

$$W = \sum_{i=-r}^r \sum_{j=-r}^r \sum_{k=-r}^r g_{\sigma_d}(d_1) g_{\sigma_r}(d_2) \quad (5.32)$$

For cubic datasets of size N^3 voxels with cubic kernels of size n^3 (or radius r , such that $n = 2r + 1$, where n is the per-dimension size), the computational complexity is $O(N^3 n^3)$.

Method

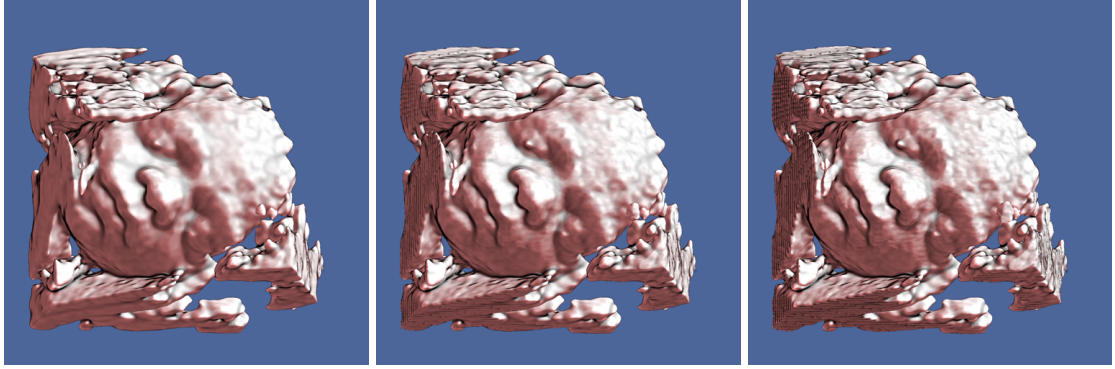
The obstetric ultrasound ‘babyface’ dataset was again used for a series of tests using the bilateral filter in order to evaluate the balance of noise reduction and feature preservation for a given set of parameters. The eye region (specifically the eyelid line) provides a good visual indicator of the feature-preserving properties of the filter. The forehead area is particularly troublesome for noise: a filter which offers greater preservation of features such as the eyelid line tends to suffer from greater noise in the forehead region; conversely, greater noise suppression in the forehead region usually results in a degradation of feature-preservation in the eyelid region. The ultimate goal of filter construction seeks to maintain features such as the eyelid line whilst maximally suppressing noise in large areas such as the forehead.

There are three distinct objectives in this section:

- To evaluate the effect of the range function on feature preservation and noise suppression. Moreover, a comparison with Perona-Malik diffusion is performed.
- To observe the filter’s efficacy and any potential artifacts when the iteration count is greater than one.
- To determine an optimum filter radius for a truncated kernel; optimised approximations of the bilateral filter are not considered at this stage.

Results

For the first objective above, one iteration of the bilateral filter (with $\sigma_d = 1.6$, radius 3) was performed for various values of σ_r in the range function; the effect of the attenuating range function can be observed in fig.(5.9).



(a) Gaussian $\sigma = 1.6$ (b) Bilateral $\sigma_d = 1.6, \sigma_r = 0.6$. (c) Bilateral $\sigma_d = 1.6, \sigma_r = 0.4$.

Figure 5.9: Effect of the attenuating range-component of the bilateral filter on feature-preservation. All images feature volumetric filters with a spatial Gaussian component of $\sigma_d = 1.6$ (the first is a standard Gaussian filter). The bilateral-filtered images (middle and right) introduce the attenuating range component with various values for σ_r . Note the increasing detail in the eyelid region at the expense of some noise reduction in the forehead area. Source: Paul Ross.

According to Durand and Dorsey [70], the one-iteration bilateral filter is less prone to step artifacts than the (multi-iteration) Perona-Malik diffusion process. This can be observed in figure (5.10), where several degrees of filtering using each method are presented together: the parameters for each were chosen to produce a similar level of feature-preservation across the two different methods. Stair-step artifacts can clearly be seen in the Perona-Malik case as the level of feature-preservation increases; the bilateral filter has much less pronounced stepping for a given feature-preservation level.

The bilateral filter was originally suggested as a one-iteration method, where a similar level of smoothing to multi-iteration diffusion PDEs is achieved through one iteration with increased filter support, rather than a large iteration count; however, the shock-derived edge-enhancement effects are visibly reduced in comparison with Perona-Malik diffusion (compare figures (5.8) and (5.4)).

In addition, the bilateral filter can be used with multiple iterations (and smaller support) as described in [69]. However, increasing the iteration count of the bilateral filter can produce severe stepping artifacts, as can be observed in fig.(5.11); in this case, two iterations were performed for a range of σ_r values. Truncation of the spatial support for kernels with smaller σ_d is a trivial optimisation step for the one-iteration case and produces results with very little difference visually from full-support kernels (for obvious reasons); moreover, restricting to one iteration minimises the potential to form stepping artifacts.

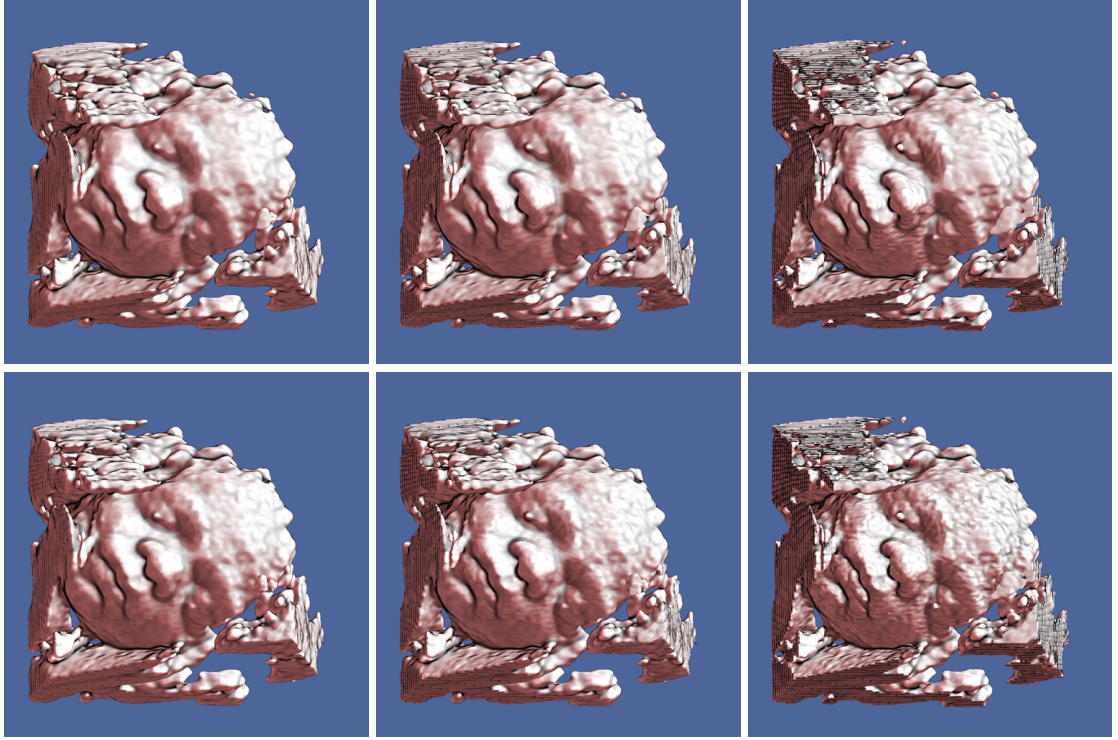
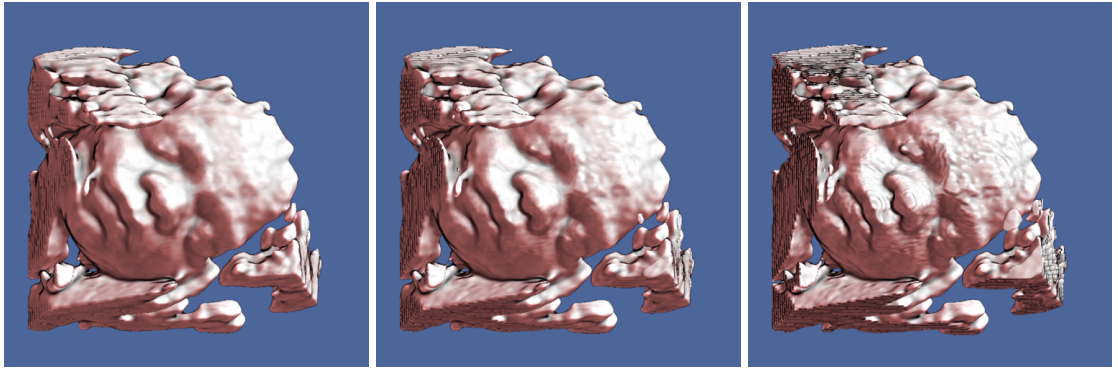


Figure 5.10: Comparison of Perona-Malik anisotropic diffusion and the bilateral filter for similar levels of feature-preservation (see eyelid line). Top row: Perona-Malik for 15 iterations, $\lambda = \frac{1}{6}$. L-R: $k = 0.7, k = 0.5, k = 0.3$. Bottom row: bilateral filter with $\sigma_d = 1.6$. L-R: $\sigma_r = 0.6, \sigma_r = 0.4, \sigma_r = 0.2$. Note the stair-step artifacts in the Perona-Malik examples. Source: Paul Ross.



(a) $\sigma_r = 0.4$.

(b) $\sigma_r = 0.3$.

(c) $\sigma_r = 0.2$.

Figure 5.11: Two iterations of the bilateral filter with $\sigma_d = 1.6$; note that stair-step artifacts are much more prominent here than with one iteration. Source: Paul Ross.

Conclusion

The one-iteration bilateral filter offers an improvement over Perona-Malik diffusion for volumetric smoothing as it is significantly less prone to stepping artifacts for a given level of feature preservation. However, it is significantly slower for large filter radii (supports), due to a time-complexity of $O(N^3n^3)$: a trivial appropriate truncation of the spatial support (e.g. for kernels with $\sigma_d = 1.5$, truncate to a radius of 3) offers significant speedup whilst maintaining a visually near-identical level of efficacy. Large kernel supports (with

large σ_d values) are not practical in a 3D context from a performance perspective unless an optimised approximation approx is used.

Overall the bilateral filter, whilst suffering from less artifacts than the Perona-Malik process, still produces an unsatisfactory tradeoff between noise suppression and feature-preservation. In the ‘babyface’ dataset, there remains significant noise in the forehead area for a reasonable level of feature preservation (where the eyelid line can still be seen). Hence, whilst time-performance optimised versions do exist [156][73][75][76], they won’t be considered further here; an alternative filtering approach will be developed instead.

5.3 Curvature-based PDE filters

As previously noted in §5.2.2, true feature-preserving smoothing in a volumetric context will require consideration of 2nd order differential features [155], as feature boundaries are delineated by regions of high curvature within an isosurface. One of the simplest forms of a second order geometric flow is *mean curvature flow* or *mean curvature motion (MCM)* [157][110] (see eq.(5.33)). Whilst MCM can effectively smooth isosurfaces (it is a locally tangent-plane-isotropic nonlinear diffusion process), it has the additional property of shrinking isosurfaces within the volume by minimisation of surface area. Convex shapes are reduced to spherical structures as the flow evolves; these decrease in volume in the evolving system. For such features, the effect is similar to Gaussian smoothing [104], but non-convex features may split apart [157]. Thus MCM as a stand-alone approach is not effective for feature preservation; it should be noted however that it is equivalent to the general component-wise diffusion PDE (eq.(5.38)) with a specific setting of parameters [110]. This is of some significance when attempting to formulate an improved feature-preserving PDE-based method. Such an anisotropy-dependent modified-MCM PDE forms the basis for the method introduced by Hossain and Möller [8] and will be discussed in §5.4; further constraints will add conditional isotropic mean curvature motion for a greater range of local isosurface-feature anisotropies: this will be crucial in the development of a novel PDE-based smoothing method which attempts to optimally balance feature preservation and noise suppression.

5.3.1 Mean curvature motion

Mean curvature motion is arguably the simplest case with which to introduce 2nd order geometric flows. Volumes are isotropically smoothed tangentially to their isosurfaces and contract in the direction of the gradient.

Considering mean curvature at a given point on an isosurface as a measure of the divergence of the surface normal, the motion can be described by:

$$\frac{\partial f}{\partial t} = \|\nabla f\| \text{div}(\mathbf{n}) \quad (5.33)$$

Using the sign convention detailed in §4.4:

$$\frac{\partial f}{\partial t} = -\|\nabla f\| \text{div}\left(\frac{\nabla f}{\|\nabla f\|}\right) \quad (5.34)$$

Expressed in terms of curvature (ignoring a scalar factor of 0.5):

$$\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_1 + \kappa_2) \quad (5.35)$$

This form is of particular importance when considering local-anisotropy-aware diffusion processes in §5.4.

5.3.2 Weighted curvature motion

Whitaker [93] describes a method for smoothing isosurfaces based on a weighted sum of principal curvatures (essentially the normalised product of the Gaussian and mean curvatures). The aim is to smooth in the direction of least curvature in order to preserve anisotropic features such as tubular structures (e.g. vessels) or isosurface feature-boundaries (e.g. around the eye and nose in foetal ultrasound imaging).

Replacing the mean curvature $\kappa_1 + \kappa_2$ with κ_w :

$$\frac{\partial f}{\partial t} = -\|\nabla f\| \kappa_w \quad (5.36)$$

with κ_w defined as:

$$\kappa_w = \frac{(\kappa_1 + \kappa_2)\kappa_1\kappa_2}{\kappa_1^2 + \kappa_2^2} \quad (5.37)$$

for $\kappa_1^2 + \kappa_2^2 \neq 0$.

This method is highly effective at preserving anisotropic feature detail (and thus features delineated by anisotropic boundaries) and tubular structures: see figure (5.12) for a comparison with mean curvature motion; however, some residual noise remains in the form of anisotropic bumps and creases. The results are similar to those produced by Hossain and

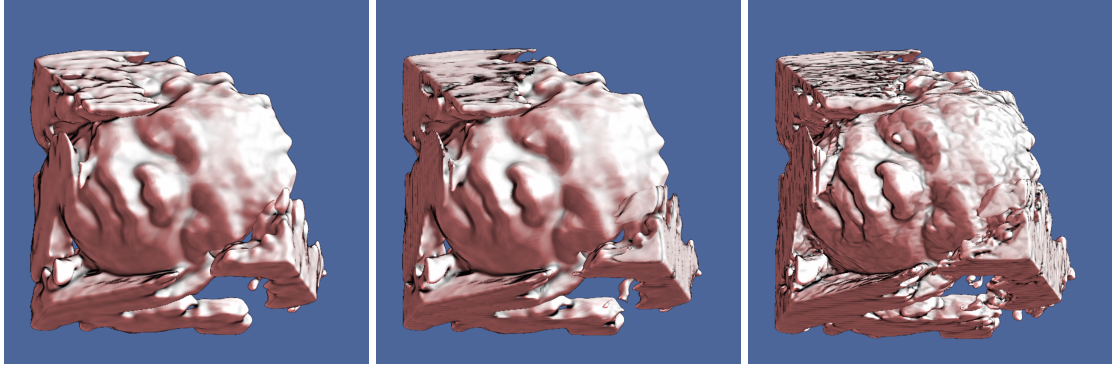


Figure 5.12: Left-Right: Gaussian smoothing, mean-curvature motion, weighted curvature motion. Note that the weighted example has a small degree of Gaussian pre-smoothing regularization. Source: Paul Ross.

Möller’s curvature-based anisotropic diffusion method [8] discussed in the forthcoming §5.4.

Since surface noise tends to be less defined and of lower curvature than the distinctive feature-boundary regions, this suggests a potential solution to the problem of residual noise: smoothing regions of lower curvature isotropically using a selective mean curvature motion based on curvature magnitude. Indeed, such a method is proposed in §5.5 as an extension to [8].

5.3.3 Other curvature flows

Malladi and Sethian [100] describe flows using other curvature metrics, such as minimum and maximum principal curvatures themselves, or a combination thereof. Their method involves selecting either a minimum or maximum curvature flow dependent on local surface properties, in order to smooth out local perturbations whilst preserving overall shape. This objective is concurrent with those here; indeed a similar thresholding approach will form a crucial component of the new method described in the forthcoming §5.5.1.

As previously mentioned in §5.2.2, Tasdizen et. al [106] describe a method which extends the feature-sharpening characteristics of the 2D Perona-Malik nonlinear diffusion process into a 3D context using 4th order geometric flows (via a system of two 2nd order PDEs); it is in effect a true generalisation of Perona-malik to 3D volumes. Clarenz et al. [104] use a diffusion tensor derived from the shape operator in their implementation of anisotropic diffusion using geometric flows.

Consideration of 2nd order geometric flows via a 3D extension to the generalised orthogonal directional-derivatives component-wise anisotropic diffusion equation (eq.(5.17)) is the approach taken by Hossain and Möller in their formulation of a 3D anisotropic diffusion PDE [8].

5.4 The Hossain-Möller anisotropic diffusion filter

Anisotropic features such as salient-feature boundaries on isosurfaces and tubular structures can be preserved by smoothing along the direction of *least curvature* (by magnitude): an early example was proposed by Whitaker [93] (see eqs.(5.36,5.37)).

For the purposes of describing the anisotropy of isosurface features, the absolute values of the principal curvatures are required, since these do not change with respect to orientation. Both convex and concave features will be equivalent in this regard: it is only their anisotropy that is of interest in the context of smoothing. Note that this differs from the shading context where minimum and maximum principal curvatures are used directly for shading modulation, and thus a distinction made for convex and concave features.

The direction of least curvature (by magnitude) is along the direction of κ_{min} , where $\kappa_{min} = \min\{\|\kappa_1\|, \|\kappa_2\|\}$, with κ_1, κ_2 the principal curvatures, defined (as before in §4.4) with $\kappa_1 \geq \kappa_2$. Similarly, $\kappa_{max} = \max\{\|\kappa_1\|, \|\kappa_2\|\}$. Note that $\|\kappa_{min}\| \leq \|\kappa_{max}\|$ always.

The generalised anisotropic diffusion equation (5.17) as described by Carmona et al. can be extended to 3D scalar fields.

$$\frac{\partial f}{\partial t} = d(a f_{\eta\eta} + b f_{e_1 e_1} + c f_{e_2 e_2}) \quad (5.38)$$

with $\{\eta, e_1, e_2\}$ forming an orthonormal basis in \mathbb{R}^3 , and $f_{\eta\eta}$, $f_{e_1 e_1}$ and $f_{e_2 e_2}$ the 2nd derivatives in the respective directions.

This allows control of diffusion in separate orthogonal directions: choosing a suitable basis is fundamental to obtaining an optimal diffusion scheme for salient feature preservation on embedded isosurfaces in \mathbb{R}^3 . Recall from §4.4 that the principal curvature direction vectors and surface normal at any given point on the isosurface are mutually orthogonal: selecting this as the basis for the general anisotropic diffusion scheme allows for control of diffusion along curvature directions. Moreover, it allows diffusion to be directed primarily along the direction of minimum (by magnitude) principal curvature; this is a key requirement for feature-preservation on isosurfaces in 3D (e.g. tubular structures, facial feature-boundaries).

Krissian et al. [9] discuss such a method. Generally, these anisotropic diffusion schemes can be described by:

$$\frac{\partial f}{\partial t} = d(a f_{nn} + b f_{r_{min} r_{min}} + c f_{r_{max} r_{max}}) \quad (5.39)$$

where a, b, c and d represent scalar-valued functions or constants. Note that $\{n, r_{min}, r_{max}\}$ forms an orthonormal basis in \mathbb{R}^3 , where n , r_{min} and r_{max} represent the unit surface

normal, minimum and maximum principal curvature (by magnitude) direction vectors respectively. Furthermore, observe that $n = -\eta$, following the sign convention outlined in §4.4.

Hossain and Möller [8] present such a diffusion scheme, in which diffusion is always performed along the direction of minimum curvature by magnitude. In addition, they propose an overall stopping function based on one of the Perona Malik diffusivity functions, but using the second derivative in the normal direction in lieu of gradient magnitude. Zero-crossings of f_{nn} denote boundaries between homogeneous regions in the volume (similar to edges in the 2D case), and thus overall diffusion can be limited around these regions.

The properties of their PDE can be summarised thus:

1. control of overall diffusion through use of a Perona-Malik stopping function of f_{nn} to constrain flow near homogeneous-region boundaries.
2. performing smoothing on embedded isosurfaces along the direction (r_{min}) of least curvature by magnitude (κ_{min}), based on local anisotropy (computed using the ratio of κ_{min} to κ_{max}); note that isotropic features (including isotropic noise-based artifacts) are thus smoothed isotropically. Smoothing is always performed along the direction of r_{min} ; smoothing is performed in the direction of r_{max} in proportion to local anisotropy.
3. limiting smoothing across isosurfaces (i.e. smoothing in the direction of the surface normal).

The 2nd and 3rd properties are of fundamental importance to feature-preservation whilst maximising the suppression of noise; however, the 1st is of less significance in the case of single-isosurface rendering such as obstetric ultrasound, since features tend to be defined and delineated by lines of curvature *on* isosurfaces (i.e. feature boundaries are often regions of high anisotropy and curvature such as ridges and valleys) rather than by inter-volume region boundaries in this context [106]. However, some surface feature detail may be lost if surfaces contract under a locally isotropic flow, so careful adjustment of overall diffusion on isosurface boundaries via property 1 is often desirable: note that too much restriction will limit any anisotropic smoothing specified by property 2.

The Hossain-Möller PDE is thus given by:

$$\frac{\partial f}{\partial t} = h(f_{nn})(\zeta f_{nn} + f r_{min} r_{min} + \tau f r_{max} r_{max}) \quad (5.40)$$

with $h(f_{nn}) \in [0, 1]$, $\tau \in [0, 1]$ and $\zeta = 0$ to fulfil properties 1, 2 and 3 above respectively.

Recall that the curvature along direction v is the normalised directional 2nd derivative in that direction (see §4.4, also [9]).

$$\kappa_v = -\frac{1}{\|\nabla f\|} f_{vv} \quad (5.41)$$

Note that the negative sign is a consequence of the convention outlined in §4.4 previously.

Hence eq.5.40 (with parameter ζ set to zero) can be expressed in terms of curvature:

$$\frac{\partial f}{\partial t} = -h(f_{nn})\|\nabla f\|(\kappa_{min} + \tau\kappa_{max}) \quad (5.42)$$

Compare with mean curvature motion:

$$\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_1 + \kappa_2) = -\|\nabla f\|(\kappa_{min} + \kappa_{max}) \quad (5.43)$$

using the definitions of κ_{min} , κ_{max} above.

The Hossain-Möller anisotropic diffusion process is essentially an local-anisotropy-dependent curvature motion with an overall Perona-Malik-style stopping function (but utilising 2nd directional derivatives).

Stopping function

Hossain and Möller specified an overall stopping function based on the Perona-Malik exponential variant (eq.5.8), modified to produce low outputs for input values of f_{nn} close to zero, i.e. zero-crossings of the 2nd derivative of f in the direction of the isosurface normal. They argue (in [8]) that low values for f_{nn} can signify isosurface boundaries irrespective of gradient magnitude, and thus this measure is more appropriate than the latter in the context of volumetric embedded-isosurfaces. It should be noted that the value of f_{nn} will also be close to zero in near-homogeneous regions within the volume. For homogeneous regions this is inconsequential, since no further diffusion is required; for near-homogeneous regions, the sensitivity parameter σ_h can be adjusted to obtain the desired degree of smoothing.

The stopping function $h(f_{nn})$ is given by:

$$h(f_{nn}) = 1 - e^{-s(f_{nn}/\sigma_h)^2} \quad (5.44)$$

where σ_h is a parameter controlling the smoothing-attenuation effect of h , such that $h(f_{nn}) \in [0, 1]$; s is a scaling factor which Hossain and Möller set to $\ln(0.9)$.

Thus eq.(5.44) can be re-written as:

$$h(f_{nn}) = 1 - 0.9(f_{nn}/\sigma_h)^2 \quad (5.45)$$

Although not specified in [8], the alternative Perona-Malik stopping-function can be substituted when similarly modified:

$$h(f_{nn}) = 1 - \frac{1}{1 + s\left(\frac{f_{nn}}{\sigma_h}\right)^2} \quad (5.46)$$

Again, the parameter σ_h controls the attenuating effect of h on overall smoothing, and s is a chosen scaling factor.

Figure (5.13) depicts stopping functions $h(f_{nn})$ for equations (5.44) and (5.46) respectively for various values of parameter σ_h .

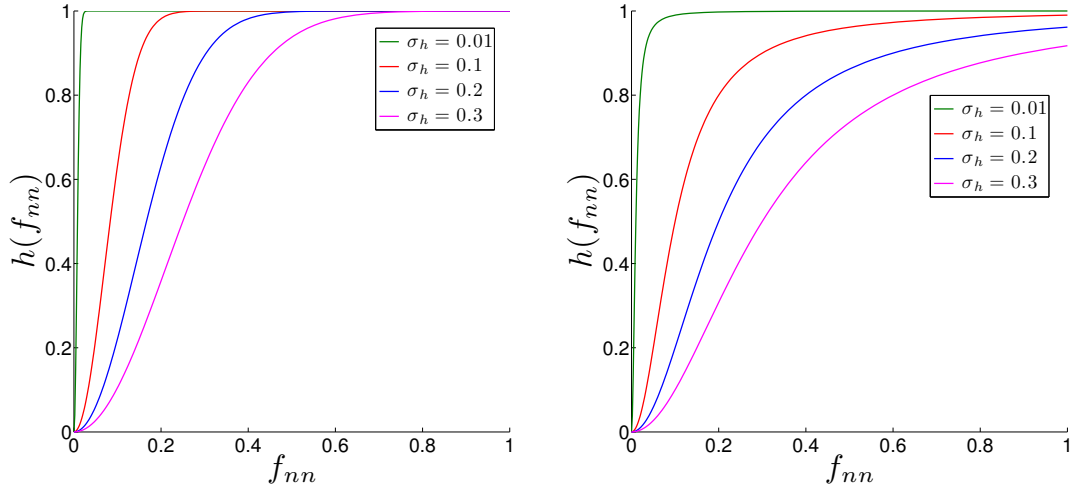


Figure 5.13: Stopping functions $h(f_{nn})$ for various values of parameter σ . Left and right subfigures represent equations (5.44) and (5.46) respectively with $s = 1$.

The original exponential variant of h (eq.(5.45) as presented in [8]) will be used exclusively for all implementations of the PDE in this document.

Anisotropy function

The *anisotropy function*, denoted by τ is fundamental to the feature-preserving characteristics of Hossain-Möller diffusion. In the absence of stopping function h , the sole difference between mean curvature motion and Hossain-Möller's PDE is the inclusion of τ as a coefficient of the κ_{max} flow component. τ is a curvature-based measure of local anisotropy (within a small neighbourhood) and is used to direct flow accordingly. Note that overall, $\tau \in [0, 1]$. Setting τ to zero corresponds to minimum-curvature flow; $\tau = 1$ corresponds to isotropic mean curvature motion.

τ is defined as follows:

$$\tau = \begin{cases} \left(\frac{\kappa_{min}}{\kappa_{max}}\right)^{2\lambda} & \text{if } \kappa_{max} \neq 0 \\ 1 & \text{if } \kappa_{max} = 0 \end{cases} \quad (5.47)$$

where λ is a non-negative integer controlling overall scaling. The factor of two is to ensure a positive value for τ . Note that $\tau \in [0, 1]$, since $\kappa_{min} \geq \kappa_{max}$ by definition.

For the purposes of this project, the restriction on λ will be relaxed, so that $\lambda \in \{\mathbb{R}^+ \cup \{0\}\}$, and τ specified as:

$$\tau = \begin{cases} \left\|\frac{\kappa_{min}}{\kappa_{max}}\right\|^{2\lambda} & \text{if } \kappa_{max} \neq 0 \\ 1 & \text{if } \kappa_{max} = 0 \end{cases} \quad (5.48)$$

This allows a much finer control of the anisotropy function, and hence the transition to full isotropic smoothing; the factor of 2 will be kept for consistency with the original literature.

Implementation

Volume data samples are obtained using central differences, whilst 1st and 2nd derivatives are computed using cubic splines (using the fast GPU implementation method described by Sigg and Hadwiger in [146]). The PDE time-solution f at iteration $n + 1$ is approximated using the standard forward-Euler numerical integration method depicted in eq.(5.49).

$$f^{n+1} = f^n + \Delta t \frac{\partial f^n}{\partial t} \quad (5.49)$$

In the context of this PDE, the numerical process becomes:

$$f^{n+1} = f^n - \Delta t \left(h(f_{\mathbf{nn}}^n) \|\nabla f^n\| (\kappa_{min}^n + \tau^n \kappa_{max}^n) \right) \quad (5.50)$$

For implementation specifics (i.e. the relevant GPU compute-kernels), see §A.4. For stability, Δt is limited to a value of 0.4 [8]; for the practical implementation here, Δt is set to a constant value of 0.3 (small enough to remain accurate and stable with all tested datasets, yet large enough to allow for good performance. Although allowing larger values for Δt , implicit numerical schemes were disregarded for performance reasons due to the additional complexity of solving a nonlinear system at each iteration.

For higher accuracy, a 4th-order Runge-Kutta (RK4) variant was also implemented. This improves the solution accuracy at each iteration step but as an explicit method won't re-

duce the stability constraints on the timestep Δt ; moreover, the RK4 method requires more samples, an increase in GPU memory consumption (to hold intermediate volume data) and a more complicated implementation from a code perspective. Hence performance is both slower and requires more GPU-memory than the standard forward-Euler approach. Visible results were near identical to those obtained by forward-Euler, so the latter method was used throughout for performance reasons.

Regularisation

In a similar approach to that taken by Catté (in that case, with respect to Perona-Malik diffusion), Hossain and Möller specify that second derivatives should be computed on a Gaussian-smoothed version of the original dataset to provide a small amount of regularisation. In this project, this regularisation process is slightly altered: prior to the PDE evolution, the volume is pre-smoothed by a small amount. Hence all samples (dataset values and derivatives) at the 1st iteration of the PDE are taken from a pre-smoothed dataset, not just the second derivatives.

This has the following advantages:

- Allows for construction of a more flexible smoothing pipeline, where an initial pre-smoothing stage can be Gaussian-based or nonlinear (Perona-Malik, bilateral filter or other), with near-identical results; moreover, the amount of pre-smoothing of the data can be controlled visually and subsequent PDE-based stages omitted or constrained depending on requirements.
- A small degree of initial Gaussian smoothing can reduce the level of inherent noise to that which the secondary smoothing stage (curvature-based) can further process to ensure smooth isosurfaces with minimal artifacts or excessive MCM-like contraction of isosurfaces. The optimum combination may be dataset dependent: i.e. highly noisy datasets may require a more aggressive Gaussian pre-smoothing stage.
- Dataset sample values are consistent with derivatives potentially providing a more effective regularisation.
- Only one volume needs to be sampled per iteration during the PDE evolution, reducing the amount of texture-reads required (or code complexity if the Hossain-Möller/Catté approach is performed on only the first iteration).

5.4.1 Effect of stopping function

As previously discussed, the stopping function h limits the overall diffusion near edge regions (isosurface boundaries): note that in the context of 3D, this does not correspond

directly with feature boundaries (as in the 2D case). In the 3D case, lines of curvature and ridge and valley-like regions of high anisotropy tend to define feature boundaries. For feature preservation, smoothing must be performed such that the overall shape of such regions is preserved; the inter-isosurface boundaries (edges) within the volume are of less significance overall. The stopping function h attenuates overall smoothing in the latter case: however, smoothing across a particular isosurface (usually across its boundary with another within the volume) is a fundamental requirement in order to reduce surface noise.

The overall attenuation in such regions should therefore be kept to a minimum; some may be required however to reduce feature-shrinkage due to mean-curvature-motion in isotropic regions. Figure (5.14) demonstrates the balance between these conflicting objectives: as the attenuating effect of h is reduced, it can be observed that surface noise is also reduced correspondingly (5.14a-c). As depicted in figure (5.14d), setting h to 1 (by setting parameter σ_h to 0) produces maximum smoothing across the isosurface, but results in some shrinkage (note the corneal regions). Also note that there is some residual noise in the forehead region for all values of σ_h .

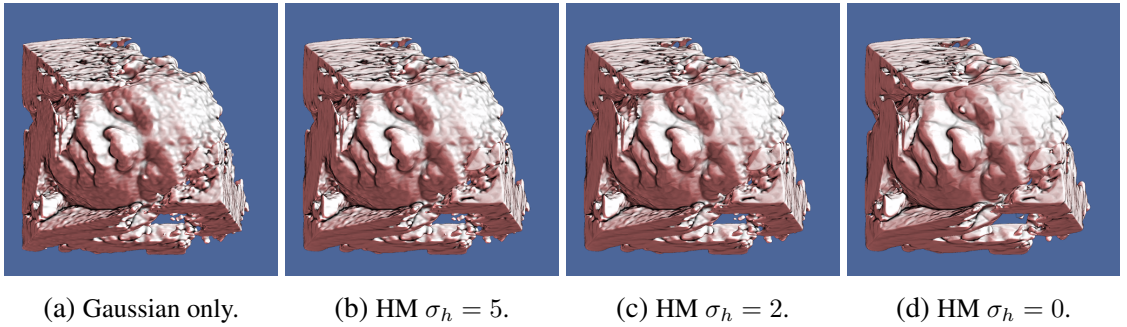


Figure 5.14: Effect of stopping function h on overall smoothing using the Hossain-Möller PDE (15 iterations, $\lambda = 2$). Note that Gaussian pre-smoothing is applied throughout (1 iteration, $\sigma = 0.85$). Source: Paul Ross.

The hybrid style-transfer-function/MPR rendering in figure (5.15) demonstrates the effect of h on the smoothing of internal noise inside the volume (visible in the MPR regions), in addition to the smoothing of the isosurface boundary (edge) of the smaller internal sphere. The degree of preservation of shape required on the latter boundary can be finely tuned by use of the σ_h parameter. Again, the appropriate balance of noise reduction and boundary preservation must be carefully considered on a per-application basis.

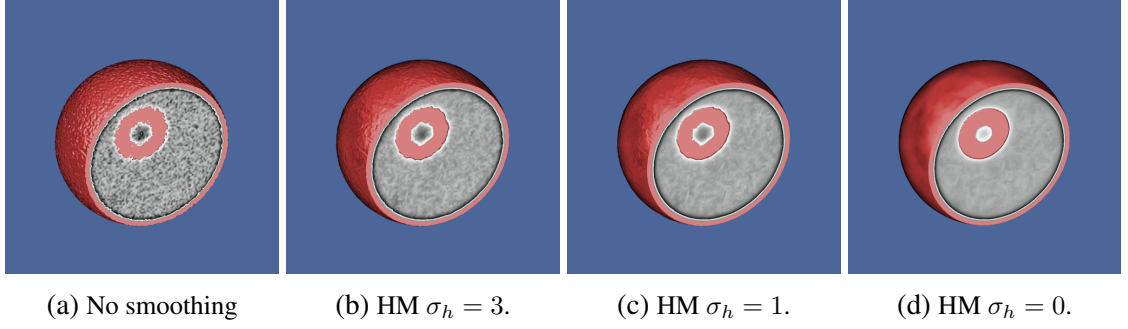


Figure 5.15: Effect of stopping function h on overall smoothing using the Hossain-Möller PDE (50 iterations, $\lambda = 2$). No Gaussian pre-smoothing is performed in this example. Rendering is a hybrid of STF and MPR to illustrate the effect of h on both internal noise and isosurface boundaries. Source: Paul Ross.

5.4.2 Effect of anisotropy function

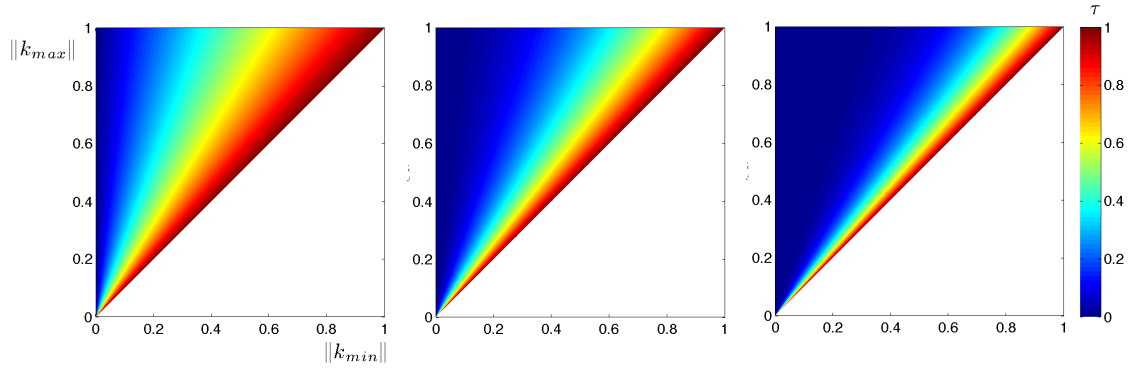


Figure 5.16: 2D scalar-field plot of Hossain-Möller anisotropy function τ (see eq.(5.47)), for $\lambda = 0.5, 1, 2$ respectively. Towards-red depicts increasing local isotropy (towards $\tau = 1$). Source: Paul Ross.

Isosurface-feature delineation tends to be defined along lines of surface curvature, particularly ridges and valley-type regions with high local anisotropy (where there is a significant difference between the values of κ_{max} and κ_{min}). It would be prudent to examine the role of the anisotropy function τ and its associated parameter λ (see fig.(5.16)) in the Hossain-Möller anisotropic diffusion process, in order to gain a greater insight to its smoothing effects on differing feature topologies.

Method

In order to demonstrate the role of the τ on the PDE evolution on regions of varying anisotropy (in particular the tendency to preserve features with high anisotropy), synthetic datasets with isosurface bumps and pits of varying anisotropy were processed using 40 iterations of the PDE. Note that there is a small amount of Gaussian pre-smoothing employed to reduce any step artifacts from the voxelisation process (i.e. incurred during the construction of the datasets). The stopping function h was set to a constant value of 1 for this section in order to concentrate on the effect of τ alone on the smoothing process.

Consider the synthetic dataset depicted in figures (5.17), (5.18), with bumps and pits of varying anisotropy respectively. Also depicted (in the rightmost images) are the values of $\kappa_{min} + \tau \kappa_{max}$, which represent the unnormalised update to the forward-Euler iteration-step at iteration $n + 1$, i.e. for $f^{n+1} = f^n + \Delta t \frac{\partial f^n}{\partial t}$, where $\frac{\partial f^n}{\partial t} = -\|\nabla f^n\|(\kappa_{min}^n + \tau \kappa_{max}^n)$. Note that these values become smaller as feature anisotropy increases: hence flow in regions of high anisotropy is reduced.

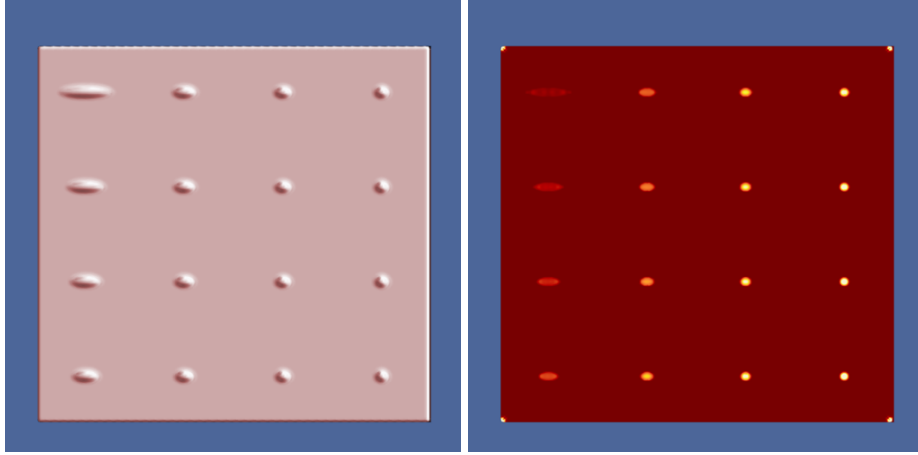


Figure 5.17: Left: synthetic dataset with convex surface features (bumps) of varying anisotropy. Right: associated values for $\kappa_{min} + \tau \kappa_{max}$ (larger values are brighter). Source: Paul Ross.

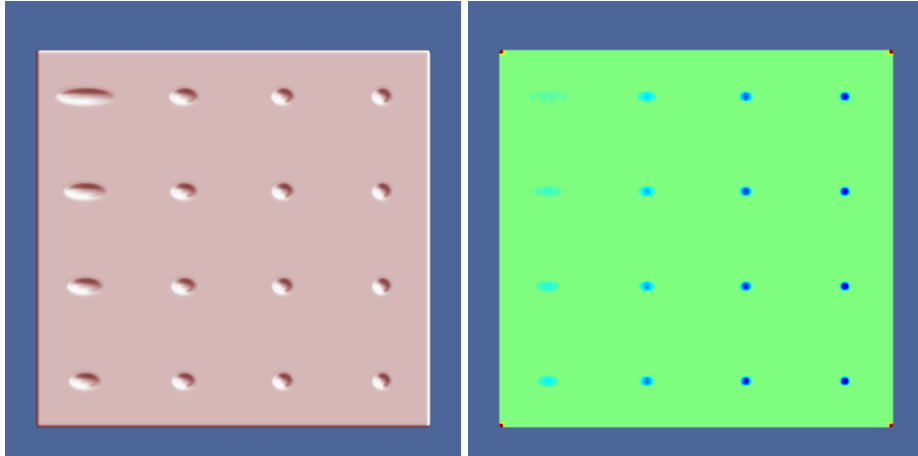


Figure 5.18: Left: synthetic dataset with concave surface features (pits) of varying anisotropy. Right: associated values for $\kappa_{min} + \tau \kappa_{max}$ (larger-magnitude negative values are darker). Source: Paul Ross.

Results

Figures (5.19) and (5.20) demonstrate the smoothing effect and associated level of feature preservation of the Hossain-Möller PDE-based filter on the synthetic datasets depicted in figures (5.17) and (5.18) respectively; note that although the surface features vary in anisotropy, $\|\kappa_{max}\|$ is constant for all features. Due to the increased flow in isotropic

regions, these features (towards the bottom right of each image depiction) undergo an increased level of smoothing overall compared with more anisotropic features; it can be observed that the most anisotropic features are preserved. Decreasing the scaling parameter λ decreases the tendency to preserve anisotropic features and ultimately results in mean-curvature-motion for $\lambda = 0$.

In the context of the ‘babyface’ dataset (see (5.21)), decreasing λ resulted in a reduction of residual surface noise at the expense of feature detail; ultimately isotropic mean-curvature-motion occurs when $\lambda = 0$. Careful selection of λ can thus be useful for attempting to procure a balance between level of surface noise and feature-detail retention, but a modified version of the PDE will be required for an optimum solution (see §5.5).

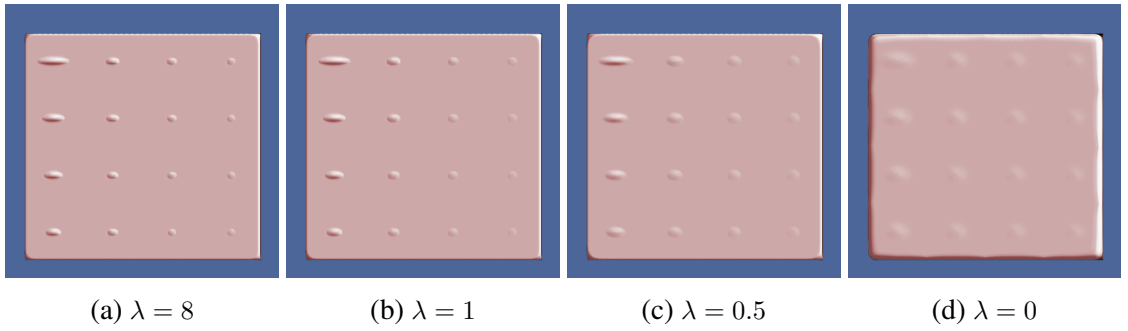


Figure 5.19: Synthetic dataset of bumps with varying anisotropy filtered with the Hossain Möller PDE (the stopping function h is set to 1 in this case): note that surface bumps with greater anisotropy are preserved more. Furthermore, the parameter λ controls the granularity of the anisotropy scale over which features are preserved or eliminated (see fig.(5.16)). The rightmost image depicts the result for $\lambda = 0$; this is fully isotropic smoothing (mean curvature motion: since $h = 1$, the PDE reduces to eq.(5.35)). Source: Paul Ross.

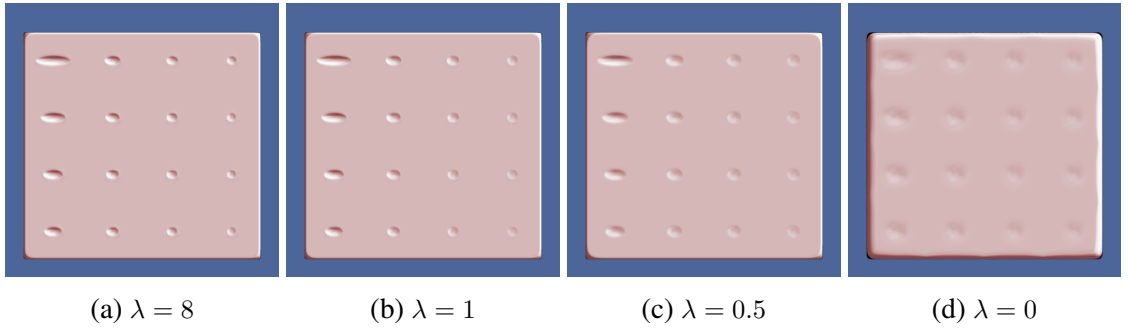


Figure 5.20: Synthetic dataset of pits with varying anisotropy filtered with the Hossain Möller PDE (the stopping function h is set to 1 in this case): note that surface pits with greater anisotropy are preserved more. Again, the parameter λ controls the granularity of the anisotropy scale over which features are preserved or eliminated (see fig.(5.16)). Source: Paul Ross.

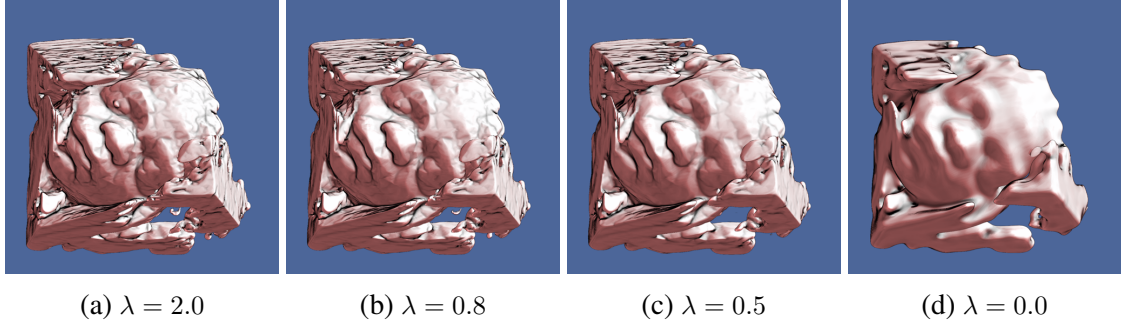


Figure 5.21: Effect of varying the anisotropy function parameter λ for the ‘babyface’ dataset: note that $\lambda = 0$ corresponds to MCM when stopping function h is set to 1; otherwise it is MCM with an overall Perona-Malik type stopping function. Source: Paul Ross.

5.4.3 Hossain-Möller Anisotropic Diffusion: Conclusion

Hossain and Möller’s PDE is clearly effective at preserving anisotropic features: much more so than the basic Perona-Malik 3D implementation or the standard bilateral filter. Moreover, there are little or no stepping artifacts. Combined with an additional regularising Gaussian pre-smoothing stage, full filtering can be performed in as little as 5-10 iterations; the performance is thus very high making this method suitable for realtime streaming 4D ultrasound applications. Current testing on an Nvidia GTX680 GPU produced a full-filter execution time of 3ms per-volume.

The visual results are very similar to that of Whitaker [93] (see §5.3.2) for parameter $\lambda \in [1, 2]$ and stopping function h set to a constant value of 1. However, its component-wise formulation presents opportunities for further refinement in order to tackle the problem of residual noise (such as that found in the forehead region in the obstetric ultrasound ‘babyface’ dataset.)

In summary:

- Very effective at preserving features with high anisotropy with little or no stepping artifacts.
- Filtering results are similar to Whitaker [93] for $\lambda \in [1, 2]$, $h = 1$, but the additional parameters offer a higher degree of control.
- For high-noise applications such as obstetric ultrasound, the method is most effective for low values of overall stopping function parameter σ_h , to facilitate maximum smoothing across (tangential to) any given isosurface; some feature detail may be lost but optimal selection of σ_h can minimise this.
- Still some residual noise, especially for highly noisy ultrasound datasets; a modified method will be required to address this issue.

5.5 Proposed solution: anisotropic diffusion with isotropy-independent selective mean curvature motion.

Whilst Hossain and Möller’s anisotropic diffusion PDE is very effective for preserving isosurface features defined by (or delineated by) regions of high anisotropy (e.g. ridges and valleys), there is often residual noise that this method cannot entirely remove. Figure (5.22) depicts residual noise in various contexts: note the forehead region in the first image. The ultimate objective is to maximally reduce noise in these regions, whilst preserving the feature boundaries and other regions of high anisotropy.

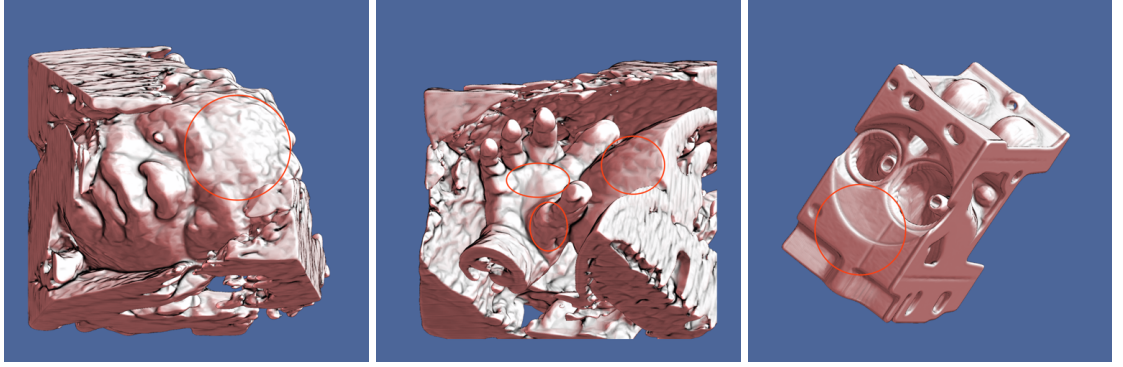


Figure 5.22: Surface-noise still remaining after 10 iterations of the unmodified Hossain-Möller PDE-based smoothing process (circled) with $\sigma_h = 0$, i.e. h set to a constant value of 1. Note that all datasets were pre-smoothed with one iteration of a volumetric Gaussian filter with $\sigma = 0.85$. Source: Paul Ross.

I propose to use a *selective mean curvature motion* based on the curvature magnitude(s); areas such as the forehead will be smoothed isotropically using MCM, whilst other regions are smoothed anisotropically using the Hossain Möller PDE via their standard τ function. The aim is to preserve shape whilst smoothing out small surface perturbations.

The desired behaviour can be achieved by two approaches:

- Thresholding the anisotropy function τ based on a chosen curvature metric so that a greater range of features are smoothed isotropically than would be suggested by their local anisotropy. A thresholding-approach based on local differential properties of the isosurface has been previously employed by Malladi and Sethian [157][100], but their method involves switching between minimum and maximum curvature flows based on a thresholded average value of the level set within a specified radius. My approach computes second order differential properties and flows either isotropically or anisotropically based on a threshold or one or more of these properties.
- Modifying the Hossain-Möller anisotropy function τ such that a greater range of surface-feature anisotropies are smoothed isotropically without the need for a threshold; This aims to provide a similar result to that of thresholding, but avoiding ar-

tifacts due to discontinuities either side of the threshold for the given curvature metric. This is under investigation for future publication.

5.5.1 Thresholding the anisotropy function

This form of selective mean curvature motion can be achieved by increasing the scope for which the anisotropy function τ returns a value of one (signifying isotropic smoothing): regions whose maximum curvature $\|\kappa_{max}\|$ is less than a given threshold will be smoothed isotropically.[‡]

This new thresholded anisotropy function τ can be constructed thus:

$$\tau = \begin{cases} \left\| \frac{\kappa_{min}}{\kappa_{max}} \right\|^{2\lambda} & \text{if } \|\kappa_{max}\| \geq \tau\text{-threshold} \\ 1 & \text{if } \|\kappa_{max}\| < \tau\text{-threshold} \end{cases} \quad (5.51)$$

for $\lambda \in \{\mathbb{R}^+ \cup \{0\}\}$.

Results

The selective MCM approach is highly efficient at reducing noise that cannot be removed by the standard Hossain-Möller (HM) PDE (see figure (5.22)). Surface noise tends to have less-pronounced curvature than feature-delineating ridges and valley regions, and is thus effectively suppressed by the action of mean curvature motion under the curvature threshold (see fig.(5.23)); other regions are filtered using the standard PDE, preserving regions of high anisotropy (via the standard HM τ function). In addition, compare the circled regions of figure (5.24) with the same regions in the volumes smoothed using the standard HM PDE in figure (5.22).

[‡]note that the threshold effect can be achieved using other curvature metrics and norms (e.g. l_2 norm with κ_{min} and κ_{max}).

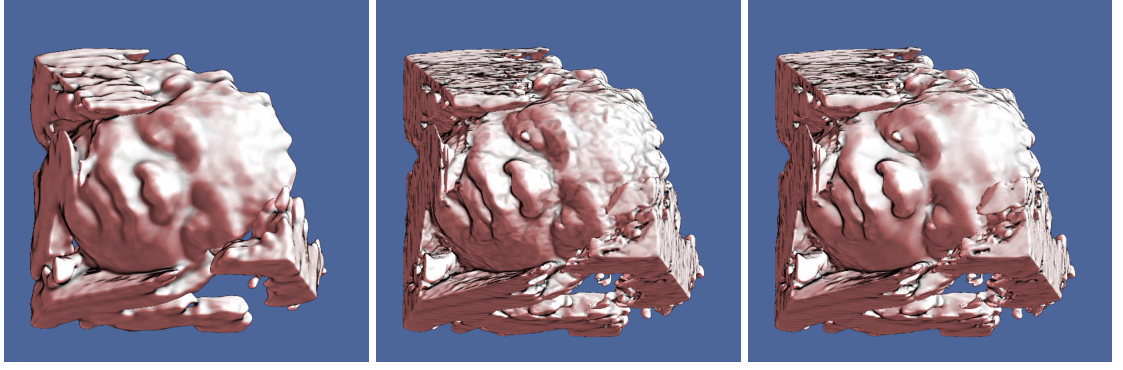


Figure 5.23: Left-Right: Gaussian smoothing (1 iteration, $\sigma = 1.5$), the standard Hossain-Möller PDE (10 iterations, $\lambda = 2$), the Hossain-Möller PDE modified for selective MCM via thresholding of $\|\kappa_{max}\|$ (10 iterations, $\lambda = 2$, τ -threshold = 0.16). Note that the latter two have a small amount of regularizing Gaussian pre-smoothing applied ($\sigma = 0.85$). Source: Paul Ross.

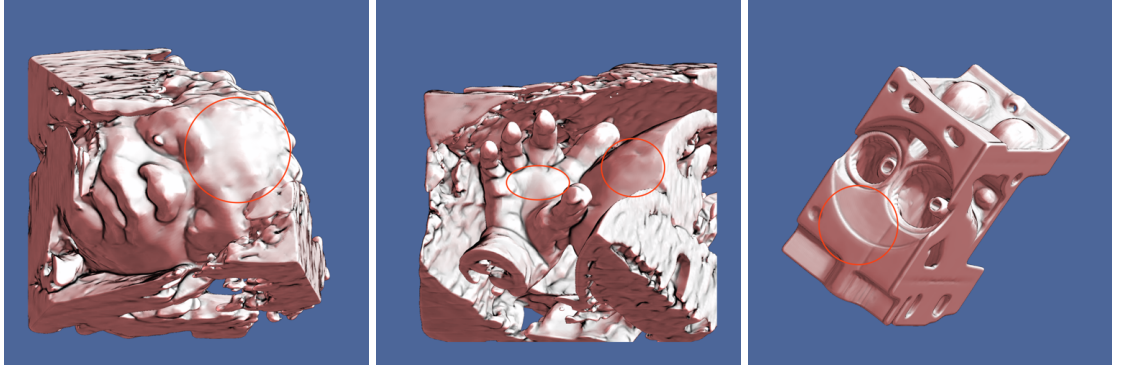
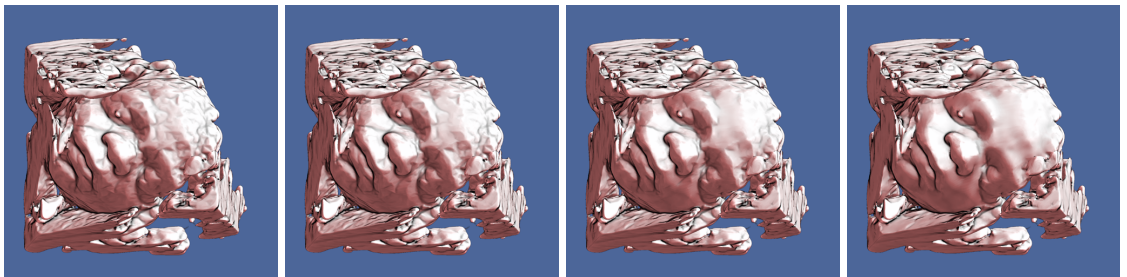


Figure 5.24: Surface noise removed after 10 iterations of the Hossain-Möller PDE with selective MCM via curvature-thresholding of τ . Note that all datasets were again pre-smoothed with one iteration of a volumetric Gaussian filter with $\sigma = 0.85$. Compare with fig.(5.22). Source: Paul Ross.

The effect of the threshold is illustrated in figure (5.25); Note how the unwanted surface noise is gradually eliminated as the threshold value for $\|\kappa_{max}\|$ is increased; i.e. as a greater range of surface is smoothed isotropically using mean curvature motion.



(a) No thresholding. (b) τ -threshold = 0.05 (c) τ -threshold = 0.10 (d) τ -threshold = 0.16

Figure 5.25: Effect of thresholding the anisotropy function τ of the Hossain-Möller PDE using different threshold values of $\|\kappa_{max}\|$ (15 iterations, $\lambda = 2$). Note that Gaussian pre-smoothing was applied throughout (1 iteration, $\sigma = 0.85$). Source: Paul Ross.

Figure (5.26) demonstrates what is actually occurring within the τ function itself (in this case, at the 6th iteration, just before the PDE evolution step). The top row colours indi-

cate the value of τ at all points on the isosurface: blue indicates the smoothing will be performed with high anisotropy, whilst red signifies isotropic smoothing (i.e. $\tau = 1$). Increasing the threshold value can be thus be seen to greatly increase the proportion of the isosurface that is to be smoothed isotropically. Note that facial features are smoothed anisotropically, as indicated by the blue regions. The bottom row depicts the rendered result after smoothing with the above threshold values, using the new selective MCM method (after the 6th iteration step). Similarly, figure (5.27) demonstrates the effect of altering the threshold value for τ (and hence fully isotropic smoothing).

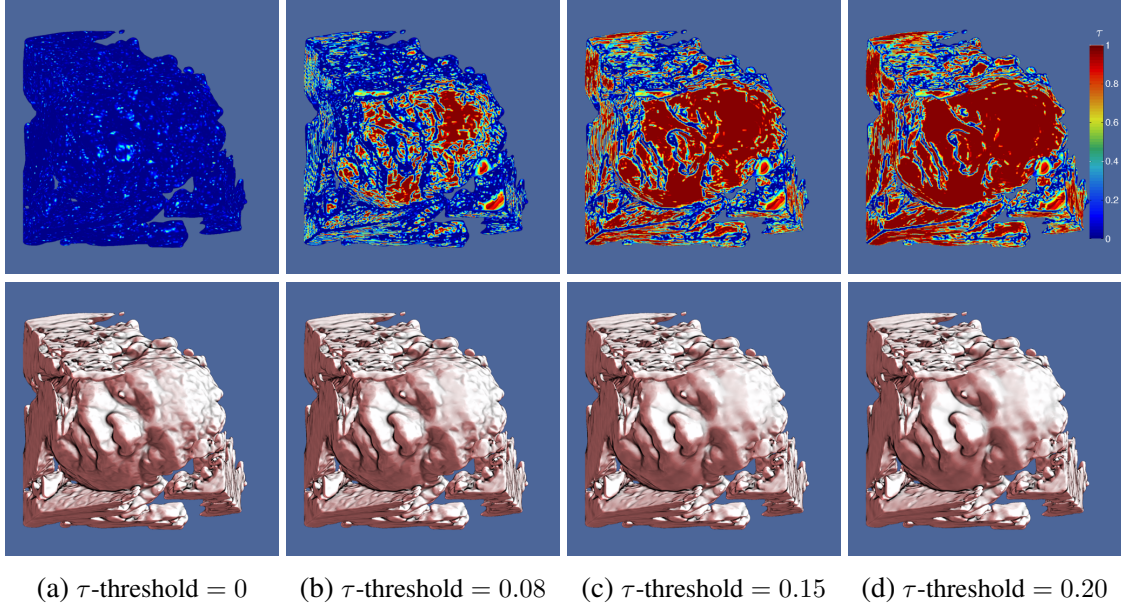
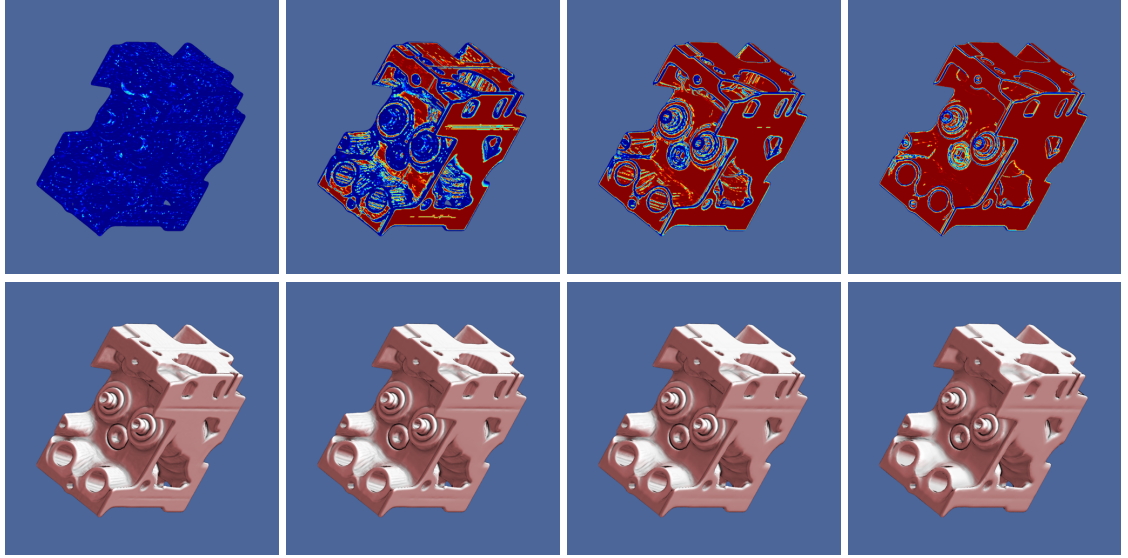


Figure 5.26: ‘Babyface’ dataset: effect of increasing τ -threshold for the modified (selective-MCM) Hossain-Möller anisotropic diffusion PDE. Top row: At iteration 6: colours indicate value for τ , and hence how isotropic the smoothing will be; red indicates isotropic smoothing, blue indicates anisotropic. Bottom row: corresponding filtered results after 6 iterations. Source: Paul Ross.



(a) τ -threshold = 0 (b) τ -threshold = 0.05 (c) τ -threshold = 0.16 (d) τ -threshold = 0.30

Figure 5.27: Engine dataset: Effect of increasing τ -threshold for the modified (selective-MCM) Hossain-Möller anisotropic diffusion PDE. Top row: At iteration 6: colours indicate value for τ , and hence how isotropic the smoothing will be; red indicates isotropic smoothing, blue indicates anisotropic. Bottom row: corresponding filtered results after 6 iterations. Source: Paul Ross.

Figure (5.28) depicts the filtered output for a range of threshold values for $\|\kappa_{max}\|$, with consideration given to the effect of overall stopping function h (controlled by parameter σ_h).

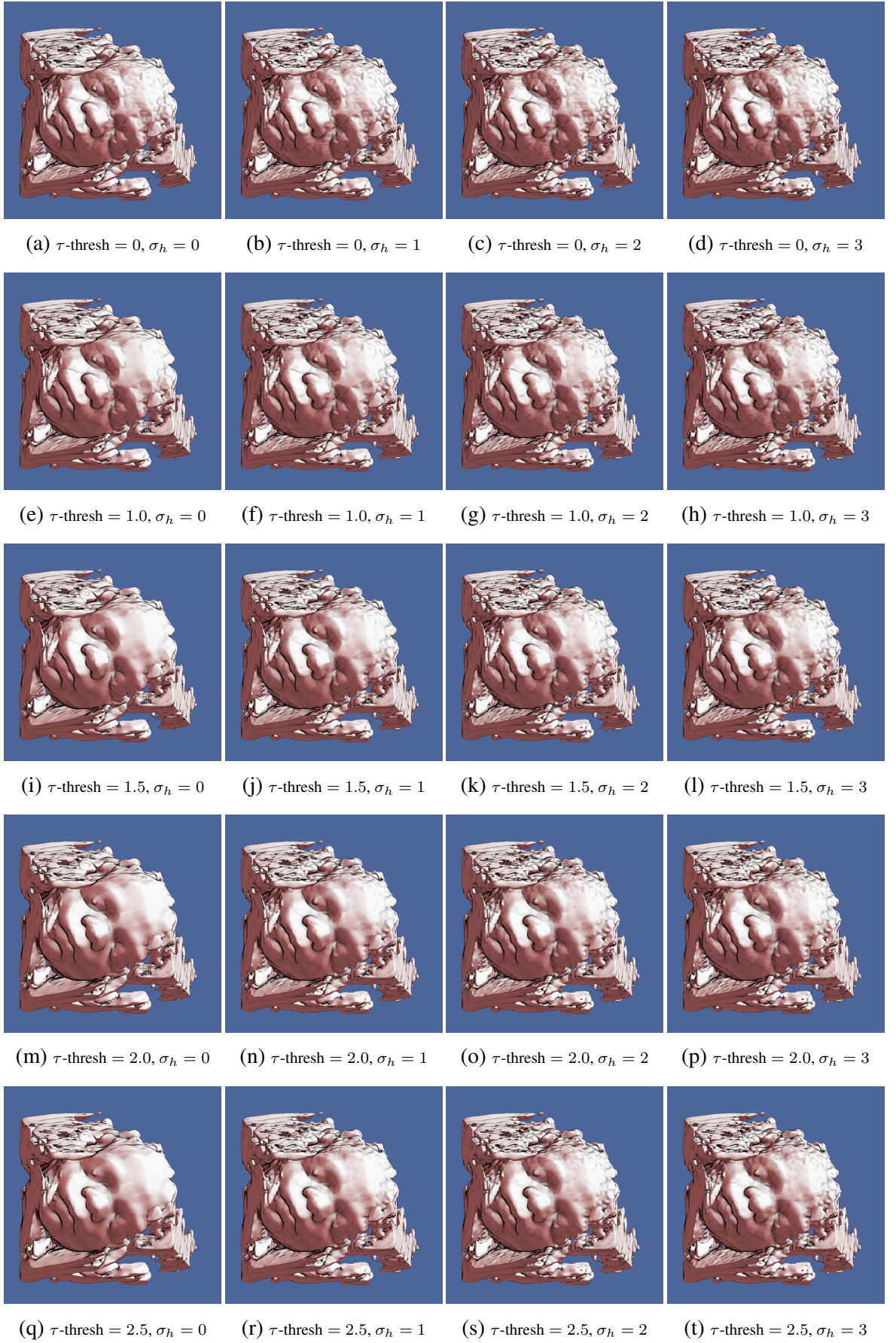


Figure 5.28: Results after application of 15 iterations of the modified Hossain-Möller PDE (with thresholded anisotropy function for selective MCM) for various threshold values of $\|\kappa_{max}\|$ and stopping function parameter σ_h . A Gaussian pre-smoothing step was applied throughout (1 iteration, $\sigma = 0.85$). Source: Paul Ross.

Comparison with the bilateral filter

Using global illumination (GI) as the rendering method illustrates the effectiveness of the combined curvature-shading and selective-MCM anisotropic diffusion filtering in preserving feature detail whilst optimally suppressing unwanted noise. Figures (5.29) and (5.30) demonstrate the highlighting of noise that occurs with curvature-based shading: the bilateral filter (left) is ineffective at suppressing this noise; our method is effective at removing most of this unwanted surface noise, whilst retaining nearly all feature detail.

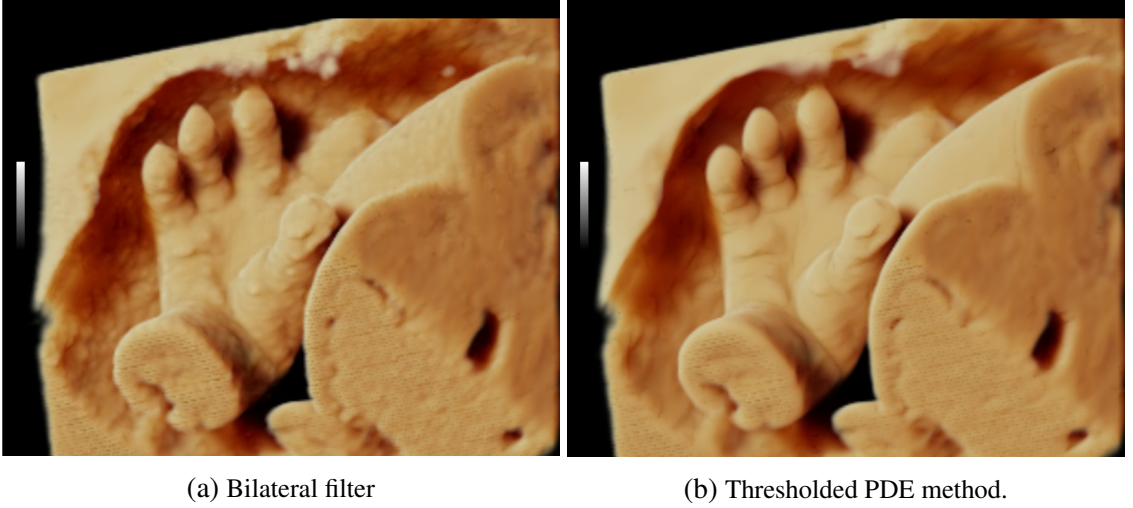


Figure 5.29: Global illumination ultrasound (GI) rendering of the obstetric hand dataset with additional curvature-based shading for feature enhancement. Left: Bilateral filter; Right: new PDE-based method. Source: Paul Ross.

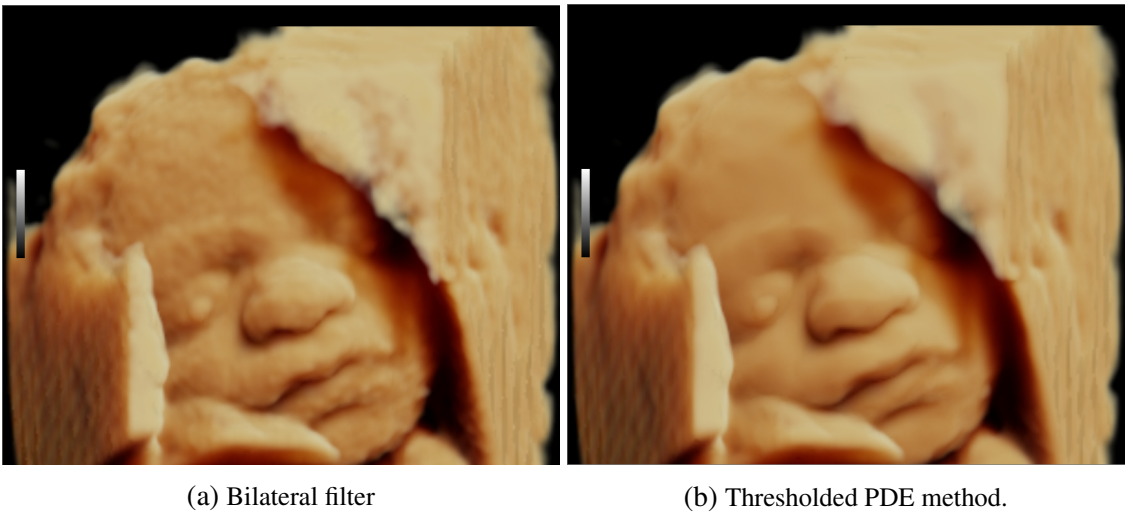


Figure 5.30: Global illumination ultrasound (GI) rendering of the obstetric 'babyface' dataset with additional curvature-based shading for feature enhancement. Left: Bilateral filter; Right: new PDE-based method. Source: Paul Ross.

5.6 Summary of related methods and contribution

At an isosurface level, noise (or other extraneous low-level detail) manifests itself as small undulations on that isosurface. Mean curvature motion is effective at smoothing such noise at the expense of feature-preservation. Conversely, Hossain and Möller’s anisotropic diffusion method is effective at preserving feature detail whilst smoothing, but leaves some residual noise. This remaining noise can be substantially reduced by combining the two approaches based on local curvature magnitude, as features are often defined by highly curved anisotropic regions, whilst noise consists of low-curvature-magnitude bumps and pits on an given isosurface.

In absence of the stopping function h , the PDE formulations of mean curvature motion and Hossain and Möller’s anisotropic diffusion PDE differ in one factor only: the coefficient τ of the κ_{max} component of curvature-based flow. This coefficient denotes the measured anisotropy in a small neighbourhood around each data sample point and is fundamental in directing flow along ridge and valley creases, hence preserving feature-detail in feature-boundary regions (features are usually delineated by ridges and valleys). In MCM, τ is equal to unity.

The novel contribution involves switching the flow from anisotropic (feature-preserving) to isotropic (non-feature preserving) dependent on local curvature magnitude; this is easily achieved by simply setting τ equal to 1 for regions where local curvature magnitude falls below a desired threshold (user controlled).

Mean Curvature Motion:

$$\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_{min} + \kappa_{max}) \quad (5.52)$$

Hossain-Möller Anisotropic Diffusion (no stopping function h):

$$\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_{min} + \tau\kappa_{max}), \quad \tau = \begin{cases} \left\| \frac{\kappa_{min}}{\kappa_{max}} \right\|^{2\lambda} & \text{if } \kappa_{max} \neq 0 \\ 1 & \text{if } \kappa_{max} = 0 \end{cases} \quad (5.53)$$

Hossain-Möller Anisotropic Diffusion with Selective MCM (no stopping function h):

$$\frac{\partial f}{\partial t} = -\|\nabla f\|(\kappa_{min} + \tau\kappa_{max}), \quad \tau = \begin{cases} \left\| \frac{\kappa_{min}}{\kappa_{max}} \right\|^{2\lambda} & \text{if } \|\kappa_{max}\| \geq \tau\text{-threshold} \\ 1 & \text{if } \|\kappa_{max}\| < \tau\text{-threshold} \end{cases} \quad (5.54)$$

for $\lambda \in \{\mathbb{R}^+ \cup \{0\}\}$.

5.6.1 Final formulation

Note that the following PDE requires that a small degree of Gaussian prefiltering regularization be performed on the initial dataset prior to application.

With the stopping function h included (although this is usually set to 1 for our purposes):

$$\frac{\partial f}{\partial t} = -h(f_{nn})\|\nabla f\|(\kappa_{min} + \tau\kappa_{max}), \quad \tau = \begin{cases} \left\|\frac{\kappa_{min}}{\kappa_{max}}\right\|^{2\lambda} & \text{if } \|\kappa_{max}\| \geq \tau\text{-threshold} \\ 1 & \text{if } \|\kappa_{max}\| < \tau\text{-threshold} \end{cases} \quad (5.55)$$

for $\lambda \in \{\mathbb{R}^+ \cup \{0\}\}$.

Note that as in Hossain and Möller’s original PDE, the stopping function h is defined as:

$$h(f_{nn}) = 1 - 0.9^{(f_{nn}/\sigma_h)^2} \quad (5.56)$$

where σ_h is a positive parameter controlling the overall influence of h : setting this to a value close to 0 reduces h to unity.

Note on parameters:

- τ -threshold: this is the main parameter controlling the threshold of curvature magnitude at which flow switches from mean curvature motion to Hossain-Möller anisotropic diffusion. Setting this to zero is equivalent to pure Hossain-Möller diffusion; increasing this applies mean curvature motion instead to regions below the curvature magnitude threshold, significantly reducing residual noise in these regions.
- λ : as in standard Hossain-Möller diffusion, this controls the scaling of the anisotropy coefficient τ and is usually set to 2. Note that setting λ to zero results in pure mean curvature motion.
- σ_h : as in standard Hossain-Möller diffusion, this controls the overall flow-attenuation effect of stopping function h at isosurface boundaries. Since our formulation is focused on reducing surface undulations on isosurfaces themselves, h is usually set to one (i.e. no attenuation effect), by setting σ_h to a value close to zero.

Halting criteria

Halting criteria for the combined HM/MCM diffusion equation is dataset-dependent and will ultimately depend on the proportion of mean curvature motion present in the final flow: MCM contracts isosurfaces within a volume towards smaller spherical structures

and ultimately contracts the volume to a point: there is no convergence on a single isosurface. Hence, there is a degree of trial and error in determining the optimal iteration count for a given dataset. For τ -threshold values of around 0.15, around 10 iterations was deemed sufficient for most datasets tested.

5.7 Application to medical imaging

5.7.1 CT imaging

One means by which to visually ascertain the level of smoothness of an isosurface is by application of a large amount of curvature shading; the intention is to deliberately highlight small surface undulations. Indeed, Kindlmann suggests this as a possible use-case for curvature shading in [61]. Applying a larger amount than would otherwise be necessary ($c = 1.4$) of curvature shading (based on minimum principal curvature) to the ‘foot’ dataset, the results of smoothing using both the standard Hossain Möller filtering method and our modified version can be compared (see fig.(5.31)). The modified version (using selective MCM) retains the same salient feature-detail as that achieved via the standard HM filtering, but the small surface undulations are removed. In general, such small undulations can be manifestations of noise or merely extraneous fine detail that may be undesirable in a simplified salient-feature-preserving rendering.



Figure 5.31: Comparison of smoothing results using the standard Hossain-Möller PDE (middle) and our modified version (right) on the pre-smoothed CT ‘foot’ dataset (left). Note that the pre-smoothing stage consisted of 1 iteration of a volumetric Gaussian filter, $\sigma = 0.8$. Exaggerated curvature shading ($c = 1.4$, minimum principal curvature) is applied in addition to the style transfer function shading to depict surface smoothness in all images. Source: Paul Ross.

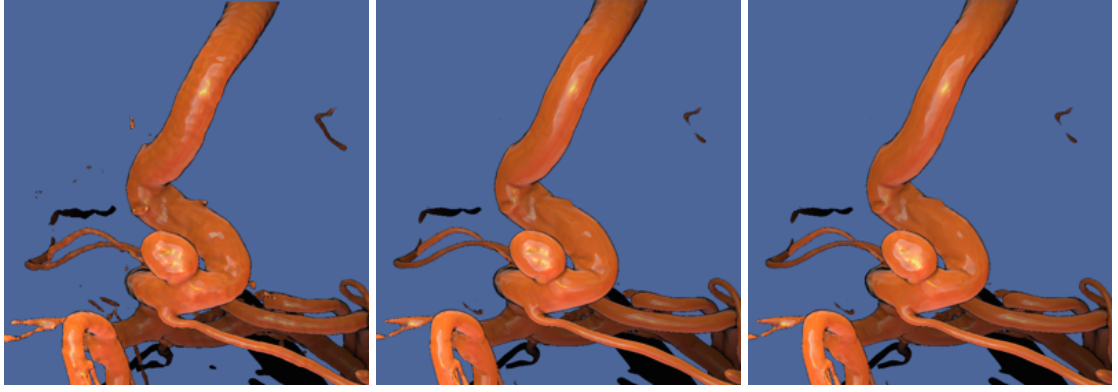


Figure 5.32: L-R: Pre-smoothed vessel dataset using one iteration of Gaussian filtering with $\sigma = 0.85$; with additional Hossain-Möller smoothing; with the modified Hossain-Möller smoothing (our method). Note that in the context of vessel smoothing, the difference between the latter two methods is small. Source: Paul Ross.

Note that the modified PDE in fig.(5.32) produces similar results to that of the standard HM PDE at this level of curvature shading; whilst the former retains smaller valley features, the new PDE reduces these to slightly darker regions through isotropic diffusion.

5.7.2 Ultrasound keepsake imaging

The new PDE is particularly effective when used in conjunction with curvature shading in the context of 3D ultrasound: see fig.(5.33).

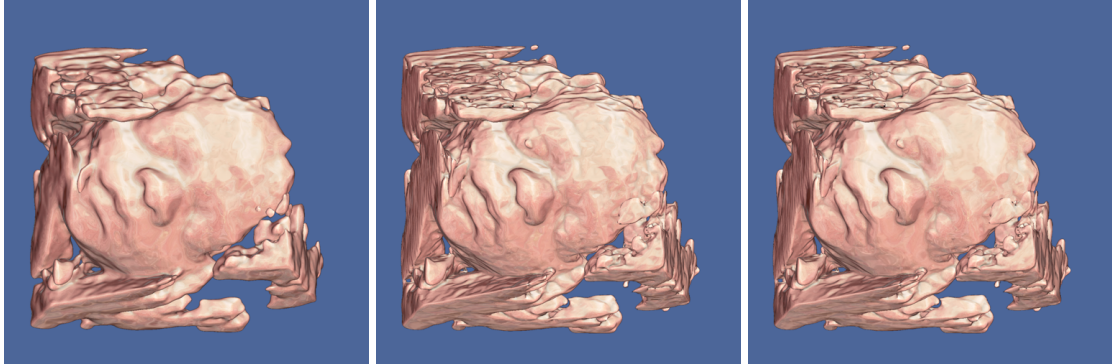


Figure 5.33: Comparison of volumetric smoothing methods applied to obstetric ultrasound dataset (all with the same degree of curvature shading for feature-emphasis): L-R: Gaussian smoothing, $\sigma = 1.65$, 1 iteration, radius = 7; standard Hossain-Möller PDE (10 iterations, $\sigma_h = 0$); our method (10 iterations, $\sigma_h = 0$, τ -thresh = 0.13). Note that the two PDE-based methods use a Gaussian prefilter (1 iteration, radius = 4, $\sigma = 0.8$) prior to the PDE stage. Source: Paul Ross.

5.8 Performance

As previously, the hardware system and testing conditions consisted of the following: Nvidia GTX680 (with 4GB memory) running on a Xeon workstation (2.4GHz proces-

sors, 4 physical cores, 8 virtual ‘cores’ with hyperthreading) with 16GB RAM. Performance per-iteration was measured for a variety of volumetric smoothing methods: see table (5.1); note that the bilateral filter (in unoptimised form) is particularly slow. All PDE-based methods are implemented using forward-Euler numerical integration; Runge-Kutta 4 produced visually near-identical results with greatly decreased performance (see table 5.3). Timings were taken for different combination of spline derivatives and central differences (see table.(5.2)); all direct data values for the system are computed using central differences. Note that all tricubic sampling was performed using the accelerated method outlined in [146].

Dataset	smoothing method						
	Gauss	BF	PM	WCM	MCM	HM	our method
3D US Babyface	2.81ms	43.15ms	0.56ms	1.44ms	1.44ms	1.435ms	1.432ms
CT Engine	9.12ms	131.0ms	1.83ms	3.61ms	3.62ms	3.315ms	3.19ms
CT Head	4.95ms	70.71ms	0.92ms	2.88ms	2.84ms	2.84ms	2.68ms

Table 5.1: Per-iteration timings for different smoothing methods; note that the effective ‘amount’ of smoothing performed per iteration is not necessarily equivalent across methods.

Dataset	Sampling method (gradient, Hessian)			
	CD, CD	CD, Spline	Spline, CD	Spline, Spline
3D US Babyface	1.36ms	4.27ms	2.89ms	11.47ms
CT Engine	3.24ms	9.88ms	8.54ms	33.57 ms
CT Head	2.84ms	9.78ms	5.09ms	21.01 ms

Table 5.2: Forward Euler timings (per iteration) for the new PDE for various combination of 2nd derivative sampling methods.

Dataset	Sampling method (gradient, Hessian)			
	CD, CD	CD, Spline	Spline, CD	Spline, Spline
3D US Babyface	6.87ms	18.97ms	13.02ms	48.36ms
CT Engine	17.40ms	48.11ms	37.99ms	145.40ms
CT Head	13.01ms	45.52ms	21.56ms	90.78ms

Table 5.3: RK4 timings (per iteration) for the new PDE for various combination of 2nd derivative sampling methods.

Note that the overall time taken to produce an equivalent visual reduction in noise when filtering the ‘babyface’ dataset is drastically lower with the new PDE than the standard HM PDE: the former requires 3 iterations (4.10ms), whilst the latter requires 40 (54ms). Together with the Gaussian pre-smooth stage, the total filter times per volume are 6.9s and 57.2s respectively. The new PDE is significantly faster as the standard Hossain-Möller methods in this context because selective mean curvature motion for the new method can

reduce small amounts of surface noise by isotropic diffusion, whereas HM attempts to preserve anisotropic features even at the scale of small surface imperfections. It should be noted that in practice, the PDE-based methods usually require a small amount of initial Gaussian smoothing as a regularization step.

5.9 Conclusion

For feature-preserving noise reduction in the context of volumetric data, it has been observed that first-order methods are not an effective approach. Perona-Malik and other gradient-based anisotropic diffusion methods (including related neighbourhood methods such as the bilateral filter) can effectively preserve edges in 2D images, but a simple extension to 3D is insufficient to retain feature detail in a volumetric context. These methods are not aware of 2nd order geometric isosurface properties and do not make sufficient distinction between noise and feature detail. Moreover, Perona-Malik suffers from stepping artifacts that are a result of a shock component in the PDE formulation [91]; to a lesser extent [70], the bilateral filter can also display such artifacts in certain circumstances [80].

As salient features tend to be delineated by lines of curvature, or surrounding boundary regions of high curvature and high anisotropy (such as ridges and valleys), a higher order geometric approach is required to effectively preserve such features when removing noise [155]. A generalised 2nd order geometric-flow anisotropic diffusion PDE [10] can be formulated in 3D by considering an orthonormal basis in \mathbb{R}^3 [9], composed of the principal curvature directions and the surface normal, and then considering 2nd directional derivatives in those respective directions [8]. Other methods explicitly make use of a diffusion tensor [87] or 4th order geometric flows [106].

In formulating a general 3D anisotropic diffusion PDE with directional derivatives, Hossain and Möller produced a PDE which is fundamentally a modified mean curvature motion (they add an anisotropy-dependent multiplier for controlling smoothing in the two independent principal curvature directions; an overall stopping function is also included). This method produces results similar to Whitaker’s weighted curvature motion [93] but is in a form that facilitates further modification. The Hossain-Möller (HM) method is very effective at preserving feature boundaries and features themselves, but some residual noise usually remains. Our proposed *anisotropic diffusion with isotropy-independent selective mean curvature motion* essentially extends the feature-preserving HM anisotropic-diffusion process by performing isotropic mean curvature motion on regions of low curvature (of which several metrics may be used). Hence areas of higher curvature are smoothed anisotropically, preserving feature detail, whilst low curvature regions (often flat but noisy regions) are effectively denoised. This new method can be applied across modalities, but is particularly applicable to 3D ultrasound imaging as this is often inherently noisy.

An initial Gaussian filter stage performs several functions: regularisation for the computation of derivatives; pre-smoothing also reduces the embedded noise component enough to allow a reduction in the number of iterations required in the PDE smoothing stage. In addition, selective MCM rapidly reduces unwanted noise in few iterations. Combined, the new smoothing method has very good performance: between 5 and 10 iterations is all that is usually required, even for noisy ultrasound data. On an Nvidia GTX680 GPU, the execution time for the new PDE filter on the ‘babyface’ dataset (192x126x200 voxels) was 4.08ms (3 iterations - see fig.(5.2)); for comparison, the Gaussian pre-filter stage completed in 2.8ms (using separable kernels of radius 3, 1 iteration). Note that the combined time for the regularization pre-smoothing step and the PDE stage should be taken into account here when considering overall performance (still under 7ms per volume).

In conclusion: combined with curvature-based shading, this new smoothing approach forms part of a smoothing/rendering pipeline that can produce images of high detail with minimal noise. The curvature shading component is highly effective for the enhancement of feature detail but is also prone to highlighting noise; the new filtering method ensures that noise is maximally suppressed whilst retaining the highlighted feature details. In addition, this method has very high performance (low execution time) and is thus suitable for real-time filtering of streaming 4D ultrasound data. Overall, our smoothing method has been demonstrated (through visual results) to be more effective at suppressing noise than the unmodified Hossain-Möller anisotropic diffusion PDE method for ultrasound imaging applications; the latter tends to leave small amounts of residual noise in areas of low curvature.

Chapter 6

Complete System

The main objective of this research was to develop a rendering and filtering system (for application to multiple modalities but particularly applicable to 3D keepsake ultrasound) which satisfied three crucial objectives: (1) the capability to enhance feature delineation, (2) have minimal noise or visible noise-based artifacts, (3) have a level of overall performance such that it is suitable for application to real-time streaming 4D ultrasound. Aside from these main criteria, the overall system was developed within the context of an illustrative approach for maximum flexibility and potential to employ contrasting rendering styles with ease. Several features are provided in addition to those designed to satisfy the fundamental requirements: depth-cueing such as *Gooch shading*, HSV-space colour modulation for curvature and depth-based effects in addition to several post-processing image-space operations for *cel-shading*.

6.1 Pipeline

The preceding chapters have outlined the specific details of the major components and their significance, but fig.(6.1) presents an overview of the entire system. All computation stages (filtering, rendering) are performed on the GPU via the CUDA runtime; final screen rendering is performed using OpenGL to display the output texture (buffer) of the CUDA rendering steps. Filtering is performed in multiple stages: stage 1 is essentially a pre-smoothing (regularization) step prior to the application of the curvature-based diffusion process. This stage can be switched between Gaussian, bilateral and Perona-Malik volumetric filters, although Gaussian smoothing is usually applied here.

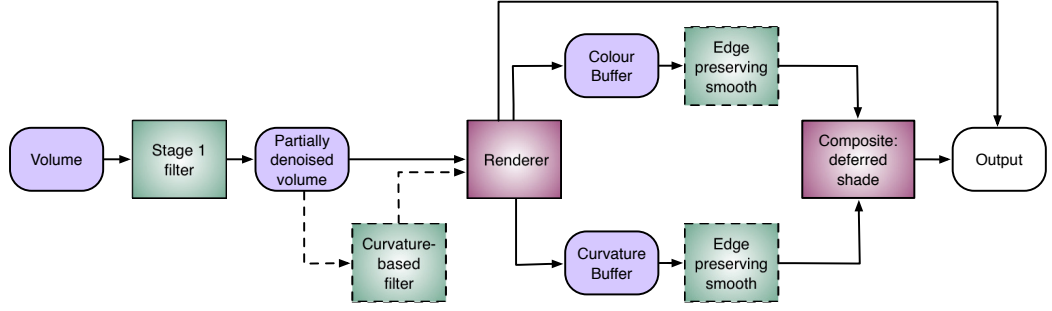


Figure 6.1: Filtering and rendering pipeline of operations for the overall system. The ‘stage 1’ filter normally consists of a volumetric Gaussian filter to pre-smooth the dataset prior to the curvature-based PDE smoothing stage; this Gaussian filter can be switched to a bilateral or a Perona-Malik variant if required. The image-plane smoothing filters for the image-space post-processing steps (deferred shading and others) are Perona-Malik based.

6.2 Applications

The system is applicable to multiple modalities, where the emphasis of specific requirements vary with modality and application. High-quality non-photorealistic rendering in multiple styles can be achieved using high-definition CT data, whereas the depiction of 3D keepsake ultrasound images (with particular attention to the feature detail) can be improved using the combination of curvature-based shading and filtering. Moreover, the techniques used in this latter application can also be employed in other rendering contexts, e.g. global illumination; the advantages they offer can be transposed to that context.

6.2.1 Ultrasound babyface (keepsake images)

The system is particularly effective at highlighting and enhancing feature detail in obstetric ultrasound applications, which suffer from a large inherent noise component. This application is therefore especially challenging: feature detail is low yet noise is high. Such a combination presents a problem in that the noise component tends to be highlighted along with the feature detail. The volumetric noise reduction stage is therefore crucial in allowing shape-enhancing shading to effectively shade feature boundaries without significant visual noise induction. Figures (6.2) and (6.3) show two datasets within different rendering contexts and how the combination of curvature shading and volume filtering can enhance both global illumination and illustrative rendering styles.



Figure 6.2: Enhancement of feature detail with minimal noise in obstetric ultrasound: curvature shading and filtering are used with both style transfer functions and global illumination renderers.



Figure 6.3: Another example of feature-detail enhancement with minimal noise in obstetric ultrasound keepsake imaging: note the finger creases but lack of noise in the palm region. Source: Paul Ross.

6.2.2 4D ultrasound

4D ultrasound is essentially a temporal stream of 3D sub-volumes, and thus suffers from the same issues as the static 3D case: feature boundaries and texture is indistinct and noise is high. Figure (6.4) depicts a heart-valve dataset where curvature shading has enhanced the sense of texture in the heart walls.

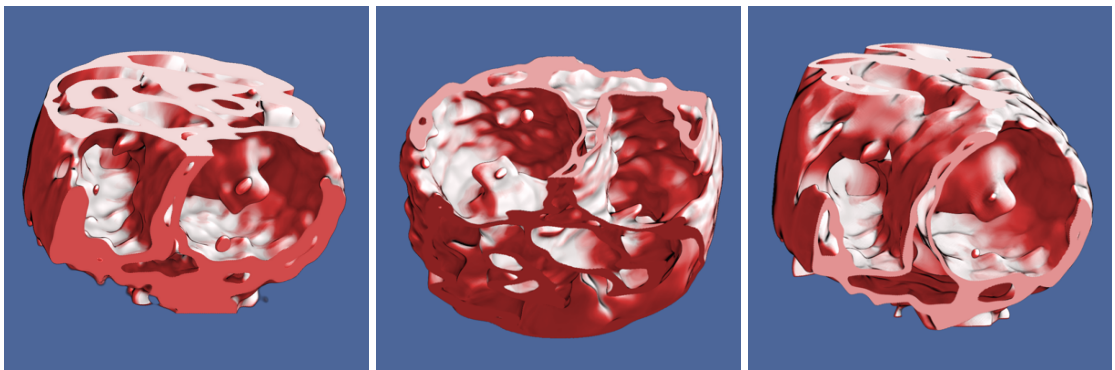


Figure 6.4: Style transfer function rendering with supplementary curvature shading to improve shape depiction in 4D ultrasound heart-valve data. See figs. (4.18) and (4.19) for comparison with style-transfer function shading alone. Source: Paul Ross.

6.2.3 CT imaging

Vessel rendering

Figures (6.5) and (6.6) depict a range of possible contexts in which different illustrative styles can be applied. Note that the boundary (neck) of the aneurysm can be highlighted using curvature shading.

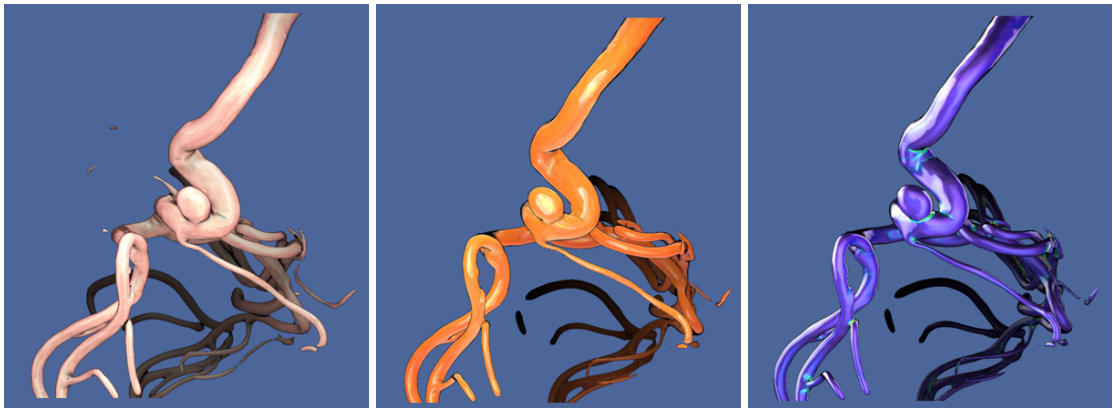


Figure 6.5: Application of the system to cranial blood vessels with additional depth cueing: multiple styles are presented according to required criteria in each case. L-R: a more subtle illustration style; non-photorealistic render featuring post-processing 'toon' shading using Perona-Malik filtering on the colour buffer; contrasting style designed to highlight regions of high minimum principal curvature for visual aneurysm detection (using HSV-curvature shading). Source: Paul Ross.

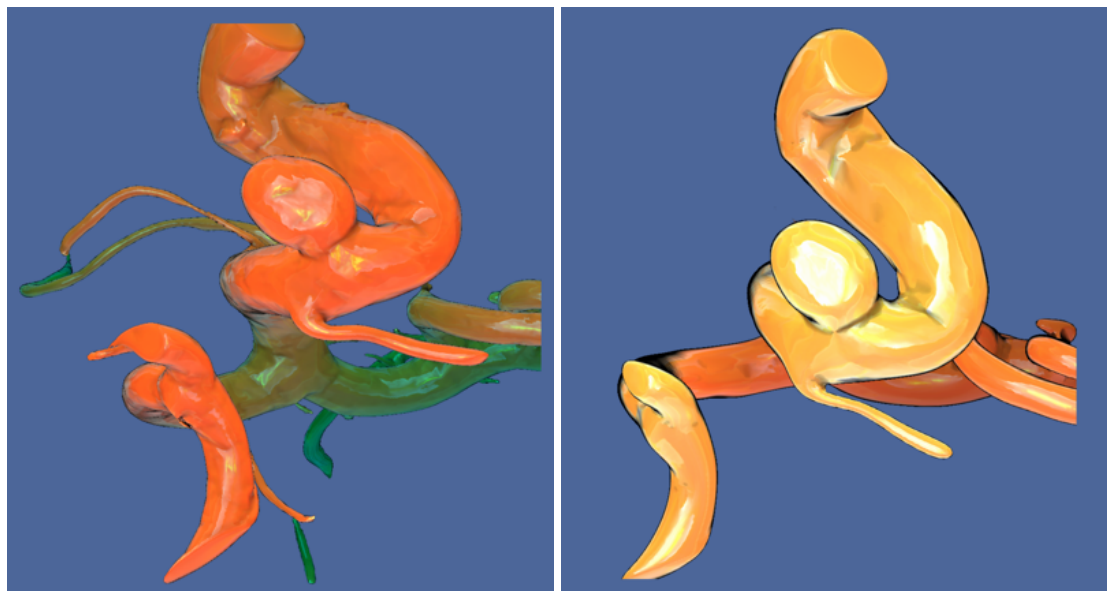


Figure 6.6: Non-photorealistic rendering styles for aneurysm depiction in vessel with depth cueing: Left: warm-to-cool Gooch shading; Right: toon shading with post-processing cel-shading effect using Perona-Malik diffusion. Source: Paul Ross.

Head and foot

Style transfer functions in conjunction with curvature shading can be used for both illustrative (fig.(6.7)) and quasi-photorealistic styles (fig.(6.8), rightmost image).

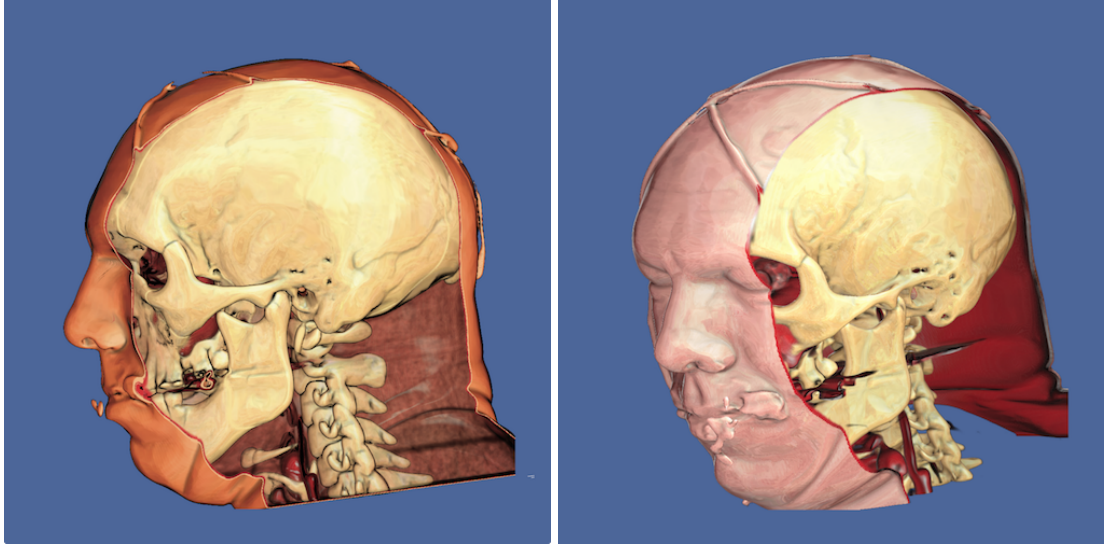


Figure 6.7: Non-photorealistic rendering of the CT ‘head’ dataset using curvature-based shading for facial feature enhancement and increased definition in skull texture. Curvature-based volume filtering is also applied to reduce noise on the skull surface for a slightly abstracted look. Source: Paul Ross.

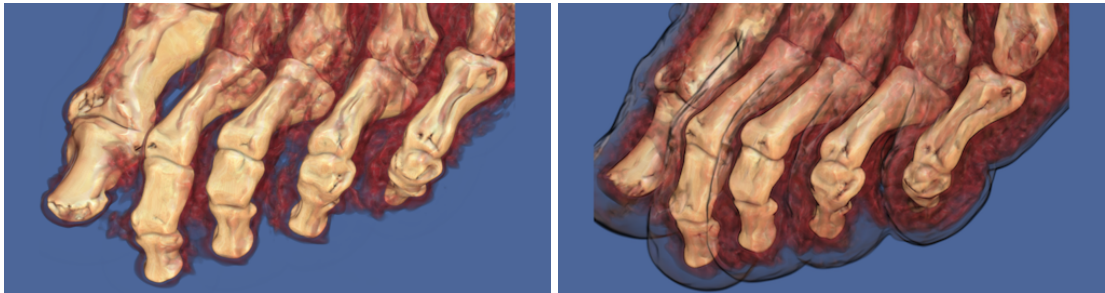


Figure 6.8: Curvature-based shading is highly effective in highlighting joint regions in the foot; contour lines and partial opacity provide added context. Source: Paul Ross.

6.3 Comparison with other methods

Achieving an improved balance of feature preservation/depiction and noise suppression is the fundamental premise behind this work, whether in the context of illustrative rendering, global illumination or other lighting and shading models. The combination of curvature-based smoothing and curvature-based shading allows a greater level of feature retention and depiction with less noise than other methods such as the bilateral filter and 3D anisotropic diffusion. This combination of curvature-based techniques is particularly effective in application to obstetric ultrasound imaging: figures (6.9) and (6.10) demonstrate the new smoothing approach applied to style transfer function rendered datasets

with curvature shading, in comparison to the bilateral filter and standard Hossain-Möller diffusion.

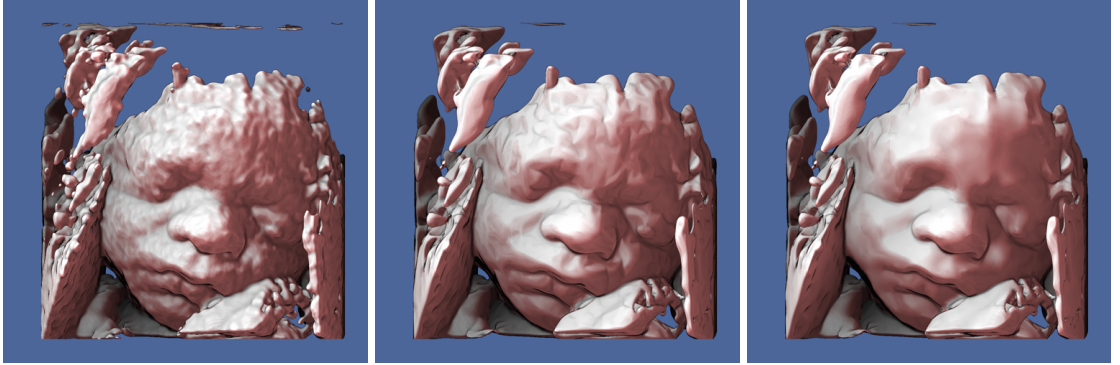


Figure 6.9: Smoothing methods compared for style transfer function (STF) rendered obstetric ultrasound with curvature shading. L-R: bilateral filter; Hossain-Möller anisotropic diffusion filter; our method. Source: Paul Ross.

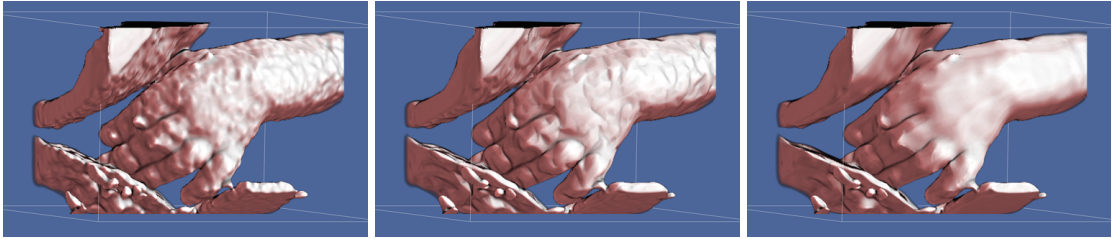


Figure 6.10: Smoothing methods compared for style transfer function obstetric ultrasound with curvature shading ('arm' dataset). L-R: bilateral filter; Hossain-Möller anisotropic diffusion filter; our method. Source: Paul Ross.

It can be observed that Hossain-Möller diffusion is effective at preserving feature-detail in comparison with the bilateral filter; however, some residual noise remains. The addition of selective mean curvature motion applied to regions of lower local curvature visibly reduces any residual noise in these regions.

The combined curvature smoothing and shading approach to improve the balance of feature depiction and noise reduction can also be applied to global illumination rendering; it was found to be highly effective at achieving these objectives in this context, producing visibly higher fidelity results than have been obtained from the application of global illumination alone. Figure (6.11) depicts the previously seen 'hand' dataset, rendered using global illumination: the effect of the additional curvature shading stage can be seen on the rightmost images. Bilateral filtering is presented as a comparison (top row); bottom row images are smoothed using the new method.



Figure 6.11: Application to global illumination ultrasound to highlight feature detail with minimal noise. Left-hand images depict global illumination only; right-hand images have the additional curvature shading step. Top row: bilateral filter; Bottom row: our method (modified Hossain-Möller anisotropic diffusion with selective mean curvature motion). Source: Paul Ross.

Figures (6.12) and (6.13) respectively compare smoothing methods and illustrate the effectiveness of curvature-shading in a global illumination context for improved feature depiction.



Figure 6.12: Smoothing methods compared for global illumination obstetric ultrasound with curvature shading. L-R: bilateral filter; Hossain-Möller anisotropic diffusion filter; our method. Source: Paul Ross.



Figure 6.13: The effect of curvature shading on feature depiction for global illumination obstetric ultrasound. L-R: bilateral filter (no curvature shading); filtering using our method (no curvature shading); our filtering method with additional curvature shading. Source: Paul Ross.

6.4 Conclusion

The system is effective across different modalities: an increase in surface texture detail is apparent in the context of CT imaging through the use of curvature shading. This can contribute to abstract illustrative rendering styles or quasi-photorealistic rendering, where it resembles the shading produced by ambient occlusion. Curvature shading can also be used to highlight specific pathology, such as vessel aneurysms; highly contrasting illustrative styles are effective in this regard. The overall system is particularly effective when applied to 3D ultrasound, as this typically has a high level of intrinsic noise; feature detail can be enhanced using supplementary curvature shading within the context of the

chosen rendering style, whilst minimising the effect of noise.

The development of a new PDE-based smoothing method allows the use of curvature shading without visually amplifying any inherent noise in the dataset. The noise-reduction and curvature-shading are thus complementary and, in combination, succeed in addressing the somewhat competing objectives of the individual components.

Chapter 7

Conclusions

This project has documented the development of a flexible volumetric filtering and rendering system for medical visualisation applications with the core requirements of enhancing feature delineation whilst minimising noise or undesired low-curvature feature detail. Curvature-based shading and curvature-based filtering can be used together to achieve good results for both these criteria; their combination is fundamental in this regard. This system has been shown to be particularly effective in enhancing facial feature detail in 3D ‘keepsake’ ultrasound imaging.

Curvature-based shading is used a supplementary shading step to augment the primary rendering method, in order to enhance feature detail and improve shape depiction by highlighting surface ridges, valleys and general second order differential properties. One particular caveat is that noise generally manifests itself (particularly speckle noise in the case of 3D ultrasound) as surface undulations, which are themselves shaded by the curvature-shading stage. In order to mitigate this, initial development incorporated a smoothed deferred curvature shading stage to filter the curvature data before application as a post-processing shading step. This works well for low-noise datasets but is insufficient in reducing the effect of the significant noise component in ultrasound data.

A PDE-based volume filtering method was thus developed using Hossain and Möller’s 3D anisotropic diffusion [8] as a solid basis; the anisotropy conditions in this are relaxed allowing pure mean curvature motion (a fast isotropic flow) for small local curvature, regardless of isotropy. This has two fundamental benefits for noisy datasets: faster evolution overall than pure anisotropic diffusion and the ability to remove small surface imperfections. Clearly defined ridges and valleys that form feature boundaries on isosurfaces are preserved, as the flow becomes anisotropic as in the case of the Hossain and Möller PDE. In the context of 3D obstetric ultrasound, this has the ability to remove noise in relatively flat regions such as the forehead, whilst preserving feature boundaries around the eyes, nose and mouth. The increase in speed as a result of an increased level of selective isotropic smoothing compared to the original Hossain and Möller PDE allows the filtering

process to complete with far fewer iterations (around 10x). This improved performance renders it highly suitable for streaming 4D ultrasound as volumes can be filtered and rendered in real time.

The combination of curvature-based filtering via the new *anisotropic diffusion with selective mean curvature motion* PDE and supplementary curvature-based shading allows salient features to be depicted in detail without the visual amplification of noise. These methods together are especially suited to noisy modalities such as 3D ultrasound and produce results that are visibly superior than other tested state of the art methods in such contexts.

7.1 Overview of work

7.1.1 Objectives

As stated in the introduction chapter, this research project’s aims were twofold:

1. The development of an illustrative rendering system with potential applications in obstetric ultrasound, surgery planning, patient education and shape-perception testing.
2. Within the context of this illustrative rendering system and more generally, the development of a unified curvature-based approach to shading-enhancement and volume smoothing: supplementary shading techniques and a novel data smoothing method were developed to enhance the depiction of salient features whilst suppressing visible noise artifacts and extraneous surface detail.

7.1.2 Illustrative rendering

A base illustrative rendering system (see §3) was constructed using Bruckner and Gröller’s style transfer function approach: in this method, shading/lighting maps constitute transfer-function control points and provide pre-rendered lookup-textures for normal-based shading for specific regions in the transfer function (which is itself a mapping of dataset intensity values to shading and opacity).

Style transfer functions proved an effective and flexible means by which to implement a wide-range of rendering styles; however, traditional illustration also makes use of contour and crease lines to accentuate or enhance shape depiction and feature detail. Style transfer functions can incorporate normal-based contour lines in their lighting maps (as a black band around the edge), but additional curvature-based crease-lines were considered

a necessary addition, as feature detail was found to be lacking using normal-based shading alone.

7.1.3 Curvature Shading

The base rendering system was extended through the addition of curvature-based shading to provide enhanced shape-depiction along crease lines such as ridges and valleys. The system can shade based on a selection of metrics: minimum and maximum principal curvature measures are used for valley and ridge shading respectively. Curvature-shading is primarily performed using a simple curvature-metric-based modulation of the normal-based rendered colour. This supplementary shading was shown to be effective for visual enhancement of feature boundaries (see §4); however, since volumetric noise (manifesting as small undulations on a given isosurface) also has inherent surface curvature, its depiction is also enhanced using curvature-based shading.

Limiting this noise amplification effect is achieved via two approaches in this system (in the final formulation detailed in the next two subsections, both constitute novel methods):

- Reducing the highlighting of noise by smoothing the curvature data used in shading modulation (see §7.1.4 below).
- Smoothing the volumetric data itself in a manner effective for our objectives: preserving salient features whilst suppressing noise or extraneous detail (see §7.1.5 below).

7.1.4 Deferred and smoothed curvature shading

The *deferred and smoothed* curvature-based shading method was our novel, yet somewhat limited first approach (US Patent Application no 14/661,456) for reducing the visible effects of noise amplification inherent in the curvature shading process. For opaque isosurfaces, noise highlighting is reduced by performing curvature shading as a deferred step: the curvature data is sampled on ray termination (intersection with the isosurface) and stored in a 2D buffer. This is then smoothed using Perona-Malik diffusion. For CT data, this method results in a significant reduction in the appearance of noise-related artifacts (see §4.7). For 3D ultrasound (a modality with a high intrinsic noise component), this technique is more limited in its efficacy: substantial noise-induced shading artifacts usually remain. To solve this issue, a new volumetric filtering method was developed.

7.1.5 Developing a new curvature-based smoothing method

The new PDE-based smoothing method constitutes this research project’s major novel contribution. Both the bilateral filter and Perona Malik gradient-based diffusion methods were tested initially but the results suffered from shock-derived stepping artifacts; these are particularly prevalent when implemented in the context of 3D volumes. Whilst they can be effective in 2D image processing, gradient-based methods such as these are not sufficient to preserve features in 3D, as these are generally defined by regions of high curvature within isosurfaces rather than high-gradient isosurface boundaries.

Curvature-based diffusion methods make use of second derivatives. The 3D anisotropic diffusion PDE developed by Hossain and Möller proved effective in preserving feature detail (where features themselves or their boundaries are delineated by highly anisotropic ridge and valley-like structures of high curvature magnitude).

In volumetric contexts, noise or other undesirable surface detail manifests itself as surface undulation with low curvature-magnitude (compared to that of feature boundaries). Removal of these unwanted components is achieved in our method by smoothing isotropically using non-feature-preserving mean curvature motion in these regions; for regions of high curvature-magnitude (such as feature boundaries), feature-preserving Hossain-Möller smoothing is performed. The novel smoothing approach performs switching between Hossain-Möller’s anisotropic diffusion and mean curvature motion, based on a threshold of the local curvature magnitude (similarly to how Malladi and Sethian [157] switch between minimum and maximum curvature flow in their diffusion process).

In summary (data smoothing):

- Gradient based methods are not sufficient for volume smoothing due to shock components and the restrictions inherent in consideration of only first-order differential properties: in 3D volumes, features (and their boundaries) within isosurfaces are defined by regions of high curvature (i.e. 2nd-order differential properties).
- Hossain-Möller anisotropic diffusion is a curvature-based method that is effective in preserving anisotropic features; however, some residual low-level noise remained in the datasets that were tested.
- Mean curvature motion (MCM) is an isotropic curvature-based smoothing method that rapidly reduces noise in 3D volumes but does not preserve features.
- The novel contribution here is a PDE that combines feature-preserving Hossain-Möller anisotropic diffusion with isotropic mean curvature motion: flow switches between them dependent on local curvature magnitude: thus, features defined by regions of high curvature are preserved by anisotropic diffusion whilst low-curvature noise or other undesirable surface detail is removed using isotropic MCM.

7.1.6 Overall system and applications

The overall system can be employed to present high quality rendered results from numerous modalities and has numerous potential applications where the visible presence of surface noise or extraneous low-level surface detail must be reduced without compromising the depiction of salient features. Moreover, supplementary shading methods can provide shape-enhancement by highlighting ridges and valley regions surrounding major features within an isosurface.

Potential applications include:

- applications where non-photorealistic illustrative rendering or exaggerated shading and enhanced shape depiction is required.
- surgical planning applications to highlight and isolate salient structures (e.g. vessels) against a traditionally rendered background (e.g. MPR or Blinn-Phong-shaded 3D rendering).
- assessment of shape perception in perceptual psychology experimentation or research contexts.
- patient education where information can be presented in a simplified form but with salient features intact.
- high fidelity 3D obstetric ultrasound for ‘keepsake’ imaging.
- applications (illustrative or other rendering contexts) where feature retention and reduction of visible noise artifacts is desired.

Note that the supplementary shading and volume filtering components can be used within any rendering context, and have been shown to be particularly effective in improving the fidelity of global illumination ultrasound imaging.

7.1.7 Performance

All per-voxel or per-pixel operations are highly parallel GPU-compute based (implemented via Nvidia’s Cuda libraries and runtime) implementations, resulting in a high level of performance overall. In particular, the volumetric filtering method is performant enough to allow it to be used for real-time filtering of subvolumes with a streaming 4D ultrasound context.

7.2 Novel contributions

In summary:

- A curvature-based PDE-based diffusion method for volumetric smoothing combining anisotropic diffusion (Hossain-Möller) for feature preservation in highly anisotropic regions with high curvature magnitude (features and their boundaries, within isosurfaces) and isotropic mean curvature motion (MCM) to reduce residual noise and extraneous detail in regions of low curvature magnitude.
- A deferred and smoothed curvature based shading method for supplementary shading of ridge and valley regions with minimal noise amplification.
- An overall illustrative rendering system incorporating the two above contributions for an improved balance of salient feature preservation and visible noise-artifact suppression over what is currently available in state-of-the-art systems. One potentially novel application is the use of the supplementary feature-enhancing shading in perceptual psychology shape-perception testing. Other applications include surgery planning, patient education and ‘keepsake’ obstetric ultrasound imaging.

Note that the first two contributions are generally applicable to all rendering methods: they were shown to be effective in non-illustrative rendering contexts such as volumetric global-illumination ultrasound.

7.3 Future work

7.3.1 Applications

In addition to those considered initially by the author, one interesting suggested possible application of this system (and briefly mentioned above) is its potential use as a framework (or intrinsic part of one) for perceptual psychology studies into human shape and surface structure perception. As such, it could form the basis for a testing apparatus and thus contribute to a substantial research area.

7.3.2 Rendering

As Bruckner mentioned in [138], a possible improvement to the style transfer function rendering system is a separation of lighting and texturing operations. This would stop the ‘moving texture’ issue and separate specular highlights from diffuse and ambient shading/lighting; a curvature shading step can be performed between these shading stages to avoid shading over highlights with the resulting ‘greying’ effect.

7.3.3 Filtering

The new PDE method achieves a switching between mean curvature motion and anisotropic diffusion based on a threshold of a chosen curvature metric. This can introduce discontinuities in the flow as it switches from one to the other; an alternative method is under investigation for future work, in which a new anisotropy function approximates the flow achieved through thresholding, without the discontinuity. This has proved effective so far but is still at the testing stage.

A comparison of our results with non-local means would also be of interest for future work. Additionally, an in-depth comparative quantitative analysis of various state-of-the-art smoothing methods (including ours) presents an opportunity for further research, shifting the focus somewhat from visual presentation. This does however present several issues:

- It is difficult to obtain noise-free ground-truth without performing some initial smoothing. Synthetic datasets can be generated, but the voxelisation process produces aliasing artifacts: these should ideally be smoothed by some explicit filtering mechanism or by supersampling and downsampling (itself equivalent to box filtering). Yet filtering is undesirable as this could introduce bias (or its perception) into any result.
- In the context of obstetric ultrasound, using our method itself to produce ground-truth would introduce a bias into the result; yet other smoothing methods cannot produce visual results of equivalent fidelity to obtain respectable noise-free feature-preserved ground-truth.
- Hossain and Möller used a CT dataset of a human tooth as a low-noise ground-truth basis for quantitatively testing their algorithm. This dataset has low-level surface imperfections and striations, which are effectively preserved by the HM filtering method. Applying our method to such a dataset reduces this low-level surface detail with respect to the ground-truth and can result in a worse signal-to noise ratio than expected. Our algorithm doesn't aim to preserve ground-truth per-se; it reduces low-level surface imperfections (from either noise or what constitutes extraneous detail) and preserves highly anisotropic, highly curved regions and their boundaries. The result is a reduction in fine-surface detail and a preservation of salient features.

Our method is thus difficult to quantify with respect to ground truth, and is more concerned with presenting an optimal visual balance of feature preservation and low-level noise or extraneous detail reduction. More work is certainly required in this area. An alternative would be a user study involving clinicians in which questions such as 'which image presents a more visually optimal balance of feature-retention and noise suppression?' can be posed; a lay-user study would necessarily be an exercise in gathering information on

user preference: an interesting research question, but a subtly different one to that which we seek to answer.

7.4 Concluding remarks

In conclusion, this research project has developed and introduced a system for the filtering and rendering of volumetric datasets in which the primary aims are to provide an improved visual balance of feature retention and noise suppression over the current state-of-the-art. In this regard, it presents output images with comparatively minimal visible noise artifacts and a high level of salient feature-retention. The system as a whole provides a primarily illustrative rendering application, but the core novel filtering and supplementary shading techniques can be used in other rendering contexts to increase the fidelity (with respect to feature depiction and visible noise reduction) of rendered output. Of particular significance is its application to 3D obstetric ultrasound: current state-of-the art systems use global illumination as their preferred mode of rendering; application of the novel techniques developed throughout this project provide global illumination rendered output which far surpasses anything currently offered from the major medical imaging technology providers.

Appendices

Appendix A

A.1 Curvature: background

In order to formulate a means by which principal curvatures can be computed, it is first necessary to introduce a degree of formalism. The *Gauss Map* and its derivatives provide a more structured framework by which to describe and derive various curvature metrics.

A.1.1 The Gauss Map

The following can provide a succinct and intuitive means of describing the change in normal direction. Under this mapping, surface normals are normalized and parallel-translated to the centre of the unit sphere $S^2 = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}$ [158], effectively becoming position vectors for points on the surface of the sphere with respect to its centre (see figure A.1). For a surface X embedded in \mathbb{R}^3 , with orientation N , the mapping $N : X \rightarrow S^2$ is called the *Gauss Map* [159]. It can be shown that the Gauss map is differentiable: this is of significance in the attempt to quantify the amount by which the normals (and hence the tangent plane) change direction at a given point on the surface.

A.1.2 Shape operator

Curvature metrics can be computed by consideration of the differential of the Gauss Map, dN_p at a given point p on the embedded surface. Consider a mapping from the tangent space T_p to T_p :

The *Weingarten map* or *shape operator* is the differential of the Gauss Map: for a two-dimensional manifold (surface in this context) embedded in \mathbb{R}^3 , it is a 2x2 matrix whose eigenvalues and eigenvectors are the principal curvatures and associated direction vectors in the tangent plane at a given point on the surface. Informally, it essentially encapsulates the change in direction of the normal vector at a given point in an infinitesimally small local neighbourhood of the surface.

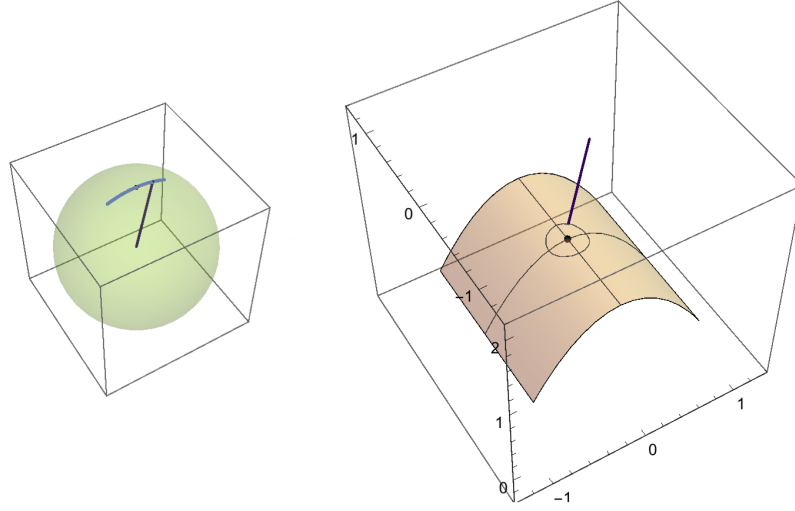


Figure A.1: Gauss map: the traced-out line on the surface of the unit sphere (left) corresponding to the movement of the normal on a surface along circular path (right). Source: Wolfram Demonstrations Project, Demo author: Michael Rogers.

The 2x2 matrix form of the shape operator L is a useful consideration in the context of explicit parameterized surfaces, but is of limited practicality in the context of this application. For implicit isosurfaces defined as level sets in a scalar field $f \in \mathbb{R}^3$, a more suitable approach involves the extension of L to a 3x3 matrix: the tangent-space basis in \mathbb{R}^2 is extended to \mathbb{R}^3 by the addition of the surface normal vector.

Extension to 3x3 Jacobian matrix

For an implicit surface embedded in \mathbb{R}^3 (with unit normal $\mathbf{n} = [n_1, n_2, n_3]^T$ at a given point), consider the 3x3 Jacobian matrix $\nabla \mathbf{n}^T$:

$$\nabla \mathbf{n}^T = \begin{bmatrix} \frac{\partial n_1}{\partial x} & \frac{\partial n_2}{\partial x} & \frac{\partial n_3}{\partial x} \\ \frac{\partial n_1}{\partial y} & \frac{\partial n_2}{\partial y} & \frac{\partial n_3}{\partial y} \\ \frac{\partial n_1}{\partial z} & \frac{\partial n_2}{\partial z} & \frac{\partial n_3}{\partial z} \end{bmatrix} \quad (\text{A.1})$$

The matrix $\nabla \mathbf{n}^T$ is the shape operator extended to a 3x3 matrix in the standard \mathbb{R}^3 Euclidean basis [111]: it has an extra eigenvector (the normal vector) and associated zero eigenvalue. The eigenvalues and eigenvectors are invariant under a change of basis, hence computing these for $\nabla \mathbf{n}^T$ will give the principal curvatures and associated directions.

It can be shown to be equivalent to the normalised tangent-plane projection of the Hessian matrix \mathbf{H} [3][160][161]:

$$\nabla \mathbf{n}^T = -\nabla \left(\frac{\nabla f^T}{\|\nabla f\|} \right) = -\frac{1}{\|\nabla f\|} (\mathbf{I} - \mathbf{n} \mathbf{n}^T) \mathbf{H} \quad (\text{A.2})$$

where \mathbf{I} is the 3x3 identity matrix and $\mathbf{I} - \mathbf{nn}^T$ projects onto the tangent plane. This can give an intuitive insight into the properties of $\nabla \mathbf{n}^T$; it is also used by Kindlmann et al in [3] in their method for the computation of principal curvatures.

A.1.3 Gaussian and mean curvatures

Two fundamental measures of curvature are the *Gaussian curvature* G and the mean curvature K . The Gaussian curvature G is defined as the product of the principal curvatures, i.e. the product of the eigenvalues (the determinant) of the shape operator. It is an *intrinsic* curvature metric: this curvature property of a manifold is independent of the space in which it is embedded (see Gauss's *Theorema Egregium* [159]). The mean curvature K for a two-dimensional manifold embedded in \mathbb{R}^3 is simply half of the sum of the principal curvatures, i.e. half of the trace of the shape operator L . It is an *extrinsic* curvature metric: it is dependent on the space in which the manifold is embedded.

Hence:

$$G = \kappa_1 \kappa_2 \quad (\text{A.3})$$

$$K = \frac{\kappa_1 + \kappa_2}{2} \quad (\text{A.4})$$

Conversely, given G and K , it is possible to compute κ_1 and κ_2 by simple application of the quadratic formula.

Gaussian curvature

An expression for the determinant $\kappa_1 \kappa_2$ of the shape operator L can be derived from its 3x3 extended form $\nabla \mathbf{n}^T$.

The directional derivative in a direction \mathbf{v} is a quadratic form $-\frac{1}{\|\nabla f\|} \mathbf{v}^T \mathbf{H} \mathbf{v}$, restricted to the tangent plane, whose maximum and minima are the principal curvatures [55].

These are equivalent to the eigenvalues in the following system:

$$(\nabla \mathbf{n}^T) \mathbf{v} = \lambda \mathbf{v} \quad (\text{A.5})$$

Factorising (as in eq. A.2) in terms of the normal vector and Hessian gives:

$$\left(-\frac{1}{\|\nabla f\|} (\mathbf{I} - \mathbf{nn}^T) \mathbf{H} \right) \mathbf{v} = \lambda \mathbf{v} \quad (\text{A.6})$$

The characteristic equation is:

$$\lambda(\lambda^2 - 2K\lambda + G) \quad (\text{A.7})$$

which has eigenvalues κ_1 , κ_2 and 0. It can be shown [55][162] that G can be expressed as follows:

$$G = \frac{1}{\|\nabla f\|^4} \nabla f^T \text{adj}(\mathbf{H}) \nabla f \quad (\text{A.8})$$

where *adj* refers to the classical adjoint.

Mean curvature

The mean curvature K can also be computed by considering the above eigensystem (eq. A.5) and its characteristic equation (A.7). K is simply half of the sum of the eigenvalues:

$$K = \frac{\kappa_1 + \kappa_2}{2} = \frac{\text{trace}(\nabla \mathbf{n}^T)}{2} = \frac{\frac{\partial n_1}{\partial x} + \frac{\partial n_2}{\partial y} + \frac{\partial n_3}{\partial z}}{2} \quad (\text{A.9})$$

It should be noted that this is equal to half the divergence of \mathbf{n} . i.e.:

$$2K = \text{div}(\mathbf{n}) = -\nabla \cdot \frac{\nabla f}{\|\nabla f\|} \quad (\text{A.10})$$

By the product rule:

$$2K = -\frac{\nabla^2 f}{\|\nabla f\|} - \nabla \left(\frac{1}{\|\nabla f\|} \right) \cdot \nabla f \quad (\text{A.11})$$

where ∇^2 denotes the *Laplacian*, with $\nabla^2 f = f_{xx} + f_{yy} + f_{zz} = \text{trace}(\mathbf{H})$. Further expansion using the chain and product rules (see §A.5.4) produces the following expression in terms of the gradient ∇f and Hessian \mathbf{H} :

$$K = \frac{\nabla f^T \mathbf{H} \nabla f - \|\nabla f\|^2 \text{trace}(\mathbf{H})}{2\|\nabla f\|^3} \quad (\text{A.12})$$

This concurs with the formula for implicit functions in [162] (disregarding the opposite sign convention).

A.2 Curvature computation: methods to determine principal curvature values.

There are numerous expressions by which curvature can be derived (see [162] for a comprehensive list of formulae). Two main methods previously described in the literature for principal curvature computation were implemented and tested here. These are:

- *Kindlmann method*: employs matrix invariants (under a change of basis) to compute principal curvatures using a further restriction of the Jacobian matrix $\nabla \mathbf{n}^T$ to its tangent-plane components only. In this approach, the restricted Jacobian is computed explicitly, and the curvature metrics derived thereof [3].
- *Gaussian and mean curvature method*: computes the Gaussian and mean curvatures directly. These constitute the product and sum of the eigenvalues of the shape-operator L ; formulae for both can be derived by considering the eigensystem $(\nabla \mathbf{n}^T)\mathbf{v} = \lambda\mathbf{v}$, since $\nabla \mathbf{n}^T$ is the 3x3 extended form of L . The principal curvatures (the eigenvalues themselves) can easily be computed from the Gaussian and mean curvatures.

The principal curvature direction vectors are not required for the curvature-shading application discussed herein.

A.2.1 Kindlmann method

Kindlmann et al [3] make direct use of the factorisation of $\nabla \mathbf{n}^T$, as depicted in (A.2) in terms of the Hessian matrix and unit normal vector. Furthermore, consider an orthonormal basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{n}\}$ where $\mathbf{v}_1, \mathbf{v}_2$ are the principal curvature direction vectors spanning the tangent plane at a given point, and \mathbf{n} is the unit surface normal at that point. With respect to this basis, it can be shown that $\nabla \mathbf{n}^T$ takes the following form:

$$\nabla \mathbf{n}^T = \begin{bmatrix} \kappa_1 & 0 & \sigma_1 \\ 0 & \kappa_2 & \sigma_2 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.13})$$

σ_1 and σ_2 represent the change in normal in the direction of the normal itself; these values are not required for computation of principal curvatures. Since this is upper triangular, its eigenvalues lie on the diagonal. In the $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{n}\}$ basis, $\nabla \mathbf{n}^T$ can be post-multiplied by the projection matrix $\mathbf{I} - \mathbf{nn}^T$ so that the σ_1, σ_2 terms are eliminated:

$$\mathbf{M} = -\frac{1}{\|\nabla f\|} \nabla \mathbf{n}^T (\mathbf{I} - \mathbf{n} \mathbf{n}^T) = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{A.14})$$

For practical purposes, $\nabla \mathbf{n}^T (\mathbf{I} - \mathbf{n} \mathbf{n}^T)$ is computed from data with respect to the standard Euclidean basis for \mathbb{R}^3 . Kindlmann makes use of matrix invariants (under change of basis: its trace and Frobenius norm) in order to obtain κ_1, κ_2 . The post-multiplication projection step allows this by removing the σ terms from the Frobenius norm whilst leaving the eigenvalues unchanged.

$$\kappa_1 = T + \sqrt{2F^2 - T^2} \quad (\text{A.15a})$$

$$\kappa_2 = T - \sqrt{2F^2 - T^2} \quad (\text{A.15b})$$

where $T = \text{trace}(\mathbf{M})$, $F = \|\mathbf{M}\|_F = \sqrt{\sum_i \sum_j |m_{ij}|^2}$, with $\kappa_2 < \kappa_1$.

Since $\nabla \mathbf{n}^T = -\frac{1}{\|\nabla f\|} (\mathbf{I} - \mathbf{n} \mathbf{n}^T) \mathbf{H}$ (see eq.A.2), the final matrix \mathbf{N} required for computation is thus:

$$\mathbf{N} = -\frac{1}{\|\nabla f\|} (\mathbf{I} - \mathbf{n} \mathbf{n}^T) \mathbf{H} (\mathbf{I} - \mathbf{n} \mathbf{n}^T) \quad (\text{A.16})$$

From this matrix, its trace and Frobenius norms and (A.15), the principal curvatures can be obtained.

Note that the Hessian matrix and normal vectors are used in direct matrix/vector computation steps: this may present a performance advantage on SIMD hardware (compared to expanding into scalar terms) if implemented in vector assembly language.

A.2.2 Gaussian and Mean curvature method

A slightly different approach can be taken using the mean and Gaussian curvatures as a starting point: the principal curvatures can be computed from these standard curvature metrics. Although the derivation makes use of a slightly different perspective conceptually than the direct method given above, they are essentially equivalent. The difference is purely computational: the above method makes use of direct matrix and vector arithmetic, whilst the following computes the principal curvatures using explicit scalar derivative components. Either method may or may not produce a performance advantage, hardware dependant.

For mean curvature K , expanding equation A.12 in terms of explicit partial derivatives, the following formula is used in the CUDA curvature computation kernel:

$$K = \frac{1}{2\|\nabla f\|^3} \left(2f_y f_z f_{yz} - f_x^2(f_{yy} + f_{zz}) + \right. \\ \left. 2f_x f_z f_{xz} - f_y^2(f_{xx} + f_{zz}) + \right. \\ \left. 2f_x f_y f_{xy} - f_z^2(f_{xx} + f_{yy}) \right) \quad (\text{A.17})$$

Similarly, expanding A.8 produces the following expression for Gaussian curvature G :

$$G = \frac{1}{\|\nabla f\|^4} \left(f_x^2(f_{yy}f_{zz} - f_{yz}^2) + 2f_y f_z(f_{xz}f_{xy} - f_{xx}f_{yz}) + \right. \\ \left. f_y^2(f_{xx}f_{zz} - f_{xz}^2) + 2f_x f_z(f_{yz}f_{xy} - f_{yy}f_{xz}) + \right. \\ \left. f_z^2(f_{xx}f_{yy} - f_{xy}^2) + 2f_x f_y(f_{xz}f_{yz} - f_{zz}f_{xy}) \right) \quad (\text{A.18})$$

These expressions can also be found in [55], [96], [8]. Principal curvatures κ_1, κ_2 can be easily obtained given K and G . Recall that $G = \kappa_1 \kappa_2$ and that $K = \kappa_1 + \kappa_2$. Hence, given both these curvature metrics and by application of the quadratic formula, the principal curvatures (maximum κ_1 and minimum κ_2) can be computed as:

$$\kappa_1 = K + \sqrt{K^2 - G} \quad (\text{A.19a})$$

$$\kappa_2 = K - \sqrt{K^2 - G} \quad (\text{A.19b})$$

with $\kappa_2 < \kappa_1$.

A.3 Curvature computation: Cuda Kernels

The following are the author's implementation of Cuda compute kernels (GPU-compute kernels using Nvidia's Cuda libraries and runtime) for the curvature computation methods outlined in §A.2.

A.3.1 Kindlmann method

```
inline __device__ float2 calcCurvature(const float3x3& H, const float3& grad)
{
    float grad_mag = length(grad);
    if(grad_mag < GRAD_THRESHOLD)
        return make_float2(0.0f, 0.0f);

    //normal, gradient magnitude
    float3 n = -1.0f*normalize(grad);
```

```

//n*transpose (nnT)
float3x3 nnT;
float nxny = n.x*n.y;
float nxnz = n.x*n.z;
float nynz = n.y*n.z;
nnT.m[0] = make_float3(n.x*n.x, nxny, nxnz);
nnT.m[1] = make_float3(nxny, n.y*n.y, nynz);
nnT.m[2] = make_float3(nxnz, nynz, n.z*n.z);

//P = I - nnT
float3x3 I;
I.m[0] = make_float3(1.0f, 0.0f, 0.0f);
I.m[1] = make_float3(0.0f, 1.0f, 0.0f);
I.m[2] = make_float3(0.0f, 0.0f, 1.0f);
float3x3 P;
P.m[0] = I.m[0] - nnT.m[0];
P.m[1] = I.m[1] - nnT.m[1];
P.m[2] = I.m[2] - nnT.m[2];

//compute G = -PHP/|g|
float one_over_grad_mag = 1.0f/grad_mag;
float3x3 temp = mul(H,P);
float3x3 temp2 = mul(P, temp);
float3x3 G;
G.m[0] = -temp2.m[0]*one_over_grad_mag;
G.m[1] = -temp2.m[1]*one_over_grad_mag;
G.m[2] = -temp2.m[2]*one_over_grad_mag;

//trace of G
float trace = G.m[0].x + G.m[1].y + G.m[2].z;

//Frobenius norm of G (squared)
float F_sq = 0.0f;
F_sq += (G.m[0].x*G.m[0].x);
F_sq += (G.m[0].y*G.m[0].y);
F_sq += (G.m[0].z*G.m[0].z);

F_sq += (G.m[1].x*G.m[1].x);
F_sq += (G.m[1].y*G.m[1].y);
F_sq += (G.m[1].z*G.m[1].z);

F_sq += (G.m[2].x*G.m[2].x);
F_sq += (G.m[2].y*G.m[2].y);
F_sq += (G.m[2].z*G.m[2].z);

//curvature
float2 curvature;
float tmp = sqrt(2.0f*F_sq-(trace*trace));
curvature.x = (trace + tmp)*0.5f; //kappa1
curvature.y = (trace - tmp)*0.5f; //kappa2

return curvature;
}

```

A.3.2 Gaussian and mean curvature method

```

inline __device__ float2 calcCurvatureAlt(const float3x3& H, const float3& grad)
{
    float grad_mag = length(grad);
    if(grad_mag < GRAD_THRESHOLD)
        return make_float2(0.0f, 0.0f);

    float G, K;
    float grad_mag_cubed = grad_mag*grad_mag*grad_mag;

    float gradx_sq = grad.x*grad.x;
    float grady_sq = grad.y*grad.y;
    float gradz_sq = grad.z*grad.z;

    float gradx_grady = grad.x*grad.y;
    float gradx_gradz = grad.x*grad.z;
    float grady_gradz = grad.y*grad.z;

    G = gradx_sq*(H.m[1].y*H.m[2].z - H.m[1].z*H.m[1].y) + 2.0f*grady_gradz*(H.m[0].z*H.m[0].y - H.m[0].x*H.m[1].z);
    G += grady_sq*(H.m[0].x*H.m[2].z - H.m[0].z*H.m[0].x) + 2.0f*gradx_gradz*(H.m[1].z*H.m[0].y - H.m[1].y*H.m[0].z);
    G += gradz_sq*(H.m[0].x*H.m[1].y - H.m[0].y*H.m[0].y) + 2.0f*gradx_grady*(H.m[0].z*H.m[1].z - H.m[2].z*H.m[0].y);
    G /= grad_mag_cubed*grad_mag;

```

```

K = 2.0f*grady_gradz*H.m[1].z - gradx_sq*(H.m[1].y + H.m[2].z);
K += 2.0f*gradx_gradz*H.m[0].z - grady_sq*(H.m[0].x + H.m[2].z);
K += 2.0f*gradx_grady*H.m[0].y - gradz_sq*(H.m[0].x + H.m[1].y);
K /= 2.0f*grad_mag_cubed;

float tmp = sqrt(K*K - G);
float k1 = K + tmp;
float k2 = K - tmp;

return make_float2(k1, k2);
}

```

A.4 Modified Hossain-Möller smoothing filter

The following is the author’s Cuda implementation for the modified HM filter (Anisotropic Diffusion with Selective MCM). Note that the system has been tested on Kepler, Maxwell and Pascal-class Nvidia GPUs.

A.4.1 Core function

```

float __inline__ __device__ Modified_HM_core_func(const float3x3& H, const float3& grad, float grad_mag, float curv_mult,
float one_over_sigma, float tau_threshold,
float two_lambda)
{
    float2 curvature = curv_mult*calcCurvatureAlt(H, grad);
    float kappa_min = curvature.y;//min
    float kappa_max = curvature.x;// max

    float3 n = -1.0f*normalize(grad);
    float f_nn = dot(n, mul(H,n));
    float h = 1.0f-pow(h_base, f_nn*one_over_sigma*f_nn*one_over_sigma);

    if (fabs(kappa_max)<fabs(kappa_min))
    {
        float temp = kappa_min;
        kappa_min=kappa_max;
        kappa_max=temp;
    }

    float tau = (fabs(kappa_max) > tau_threshold) ? pow(fabs(kappa_min)/(fabs(kappa_max)), two_lambda) : 1.0f;
    tau = clamp(tau, 0.0f, 1.0f);
    return -grad_mag*h*(kappa_min + tau*kappa_max);
}

```

A.4.2 Standard forward-Euler

```

template<typename GradSampler, typename HessSampler>
__global__ void modified_hossain_moller(int width, int height, int depth, float one_over_sigma,
float two_lambda, float timestep,
float grad_threshold, float vox_threshold,
float tau_threshold, float curv_mult)
{
    int x = blockIdx.x*blockDim.x + threadIdx.x;
    int y = blockIdx.y*blockDim.y + threadIdx.y;
    int z = blockIdx.z*blockDim.z + threadIdx.z;
    if(x>=width || y>=height || z>=depth)
        return;

    GradSampler grad_sampler;
    float3 position = make_float3(x+0.5f,y+0.5f,z+0.5f);
    float vox = tex3D(tx, x+0.5f, y+0.5f, z+0.5f));
    float3 grad = grad_sampler(tx, position);
    float grad_mag = length(grad);
    float vox_new=0;
}

```



```

if(grad_mag > grad_threshold && vox > vox_threshold)
{
    HessSampler hess_sampler;
    float3x3 H = hess_sampler(tx, position);
    vox_new = vox + timestep*Modified_HM_core_func(H, grad, grad_mag, curv_mult,
                                                    one_over_sigma, tau_threshold,
                                                    two_lambda);

    int is_nan = (int) isnan(vox_new);
    vox_new = ((float)is_nan)*vox + ((float)!is_nan)*vox_new; //avoid branch
}
else
    vox_new = vox;

clamp(vox_new, 0.0f, 255.0f);
surf3Dwrite<float>((float) vox_new, vol_out, sizeof(float)*x, y, z, cudaBoundaryModeZero);
}

```

Note that `GradSampler` and `HessSampler` are functors which specify the sampling method for the 1st and 2nd derivatives; they are of the form:

```

struct HessSamplerTricubic
{
    template<class T, enum cudaTextureReadMode mode>
    inline __device__ float3x3 operator()(texture<T, 3, mode> tex, float3 position)
    {
        return tricubic_hessian(tex, position);
    }
};

```

A.5 Some useful vector calculus expressions

The following are useful in the derivation of an expression for mean-curvature in terms of the Hessian matrix \mathbf{H} and gradient ∇u :

A.5.1 Expansion of $\nabla(\|\nabla u\|)$

2D example ($\Omega \subset \mathbb{R}^2, u : \Omega \mapsto \mathbb{R}$):

$$\begin{aligned}
 \nabla(\|\nabla u\|) &= \nabla(u_x^2 + u_y^2)^{\frac{1}{2}} \\
 &= \begin{bmatrix} \frac{\partial}{\partial x}(u_x^2 + u_y^2)^{\frac{1}{2}} \\ \frac{\partial}{\partial y}(u_x^2 + u_y^2)^{\frac{1}{2}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{1}{2}(u_x^2 + u_y^2)^{-\frac{1}{2}}(2u_{xx}u_x + 2u_{xy}u_y) \\ \frac{1}{2}(u_x^2 + u_y^2)^{-\frac{1}{2}}(2u_{yy}u_y + 2u_{xy}u_x) \end{bmatrix} \\
 &= \frac{1}{\|\nabla u\|} \begin{bmatrix} u_{xx}u_x + u_{xy}u_y \\ u_{yy}u_x + u_{xy}u_x \end{bmatrix} \\
 &= \frac{\mathbf{H}\nabla u}{\|\nabla u\|} \tag{A.20}
 \end{aligned}$$

Similarly for $\Omega \subset \mathbb{R}^3, u : \Omega \mapsto \mathbb{R}$.

A.5.2 Expansion of $\nabla\left(\frac{1}{\|\nabla u\|}\right)$

2D example ($\Omega \subset \mathbb{R}^2, u : \Omega \mapsto \mathbb{R}$):

$$\begin{aligned}
\nabla\left(\frac{1}{\|\nabla u\|}\right) &= \nabla(u_x^2 + u_y^2)^{-\frac{1}{2}} \\
&= \begin{bmatrix} \frac{\partial}{\partial x}(u_x^2 + u_y^2)^{-\frac{1}{2}} \\ \frac{\partial}{\partial y}(u_x^2 + u_y^2)^{-\frac{1}{2}} \end{bmatrix} \\
&= \begin{bmatrix} -\frac{1}{2}(u_x^2 + u_y^2)^{-\frac{3}{2}}(2u_{xx}u_x + 2u_{xy}u_y) \\ -\frac{1}{2}(u_x^2 + u_y^2)^{-\frac{3}{2}}(2u_{yy}u_y + 2u_{xy}u_x) \end{bmatrix} \\
&= -\frac{1}{\|\nabla u\|^3} \begin{bmatrix} u_{xx}u_x + u_{xy}u_y \\ u_{yy}u_y + u_{xy}u_x \end{bmatrix} \\
&= -\frac{\mathbf{H}\nabla u}{\|\nabla u\|^3} \tag{A.21}
\end{aligned}$$

A.5.3 Expansion of $\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|}\right)$

General case, $\Omega \subset \mathbb{R}^n, u : \Omega \mapsto \mathbb{R}$:

Using the product rule and eq.(A.21):

$$\begin{aligned}
\operatorname{div}\left(\frac{\nabla u}{\|\nabla u\|}\right) &= \frac{\nabla^2 u}{\|\nabla u\|} - \left(\frac{\mathbf{H}\nabla u}{\|\nabla u\|^3}\right) \cdot \nabla u \\
&= \frac{\nabla^2 u}{\|\nabla u\|} - \frac{\nabla u^T \mathbf{H} \nabla u}{\|\nabla u\|^3} \\
&= \frac{\|\nabla u\|^2 \nabla^2 u}{\|\nabla u\|^3} - \frac{\nabla u^T \mathbf{H} \nabla u}{\|\nabla u\|^3} \\
&= \frac{\|\nabla u\|^2 \operatorname{trace}(\mathbf{H}) - \nabla u^T \mathbf{H} \nabla u}{\|\nabla u\|^3} \tag{A.22}
\end{aligned}$$

A.5.4 Mean curvature

Following the convention for the direction of surface normal \mathbf{n} as used throughout, i.e.:

$$\mathbf{n} = -\frac{\nabla u}{\|\nabla u\|} \tag{A.23}$$

The mean curvature K can be defined as the divergence of the surface normal:

$$\begin{aligned}
K &= -\frac{1}{2}\text{div}\left(\frac{\nabla u}{\|\nabla u\|}\right) \\
&= \frac{\nabla u^T \mathbf{H} \nabla u - \|\nabla u\|^2 \text{trace}(\mathbf{H})}{2\|\nabla u\|^3}
\end{aligned} \tag{A.24}$$

using eq.(A.22).

References

- [1] T. Saito and T. Takahashi, “Comprehensible Rendering of 3D Shapes,” in *ACM SIGGRAPH Computer Graphics*, vol. 24, pp. 197–206, ACM, 1990.
- [2] G. Winkenbach and D. Salesin, “Rendering Parametric Surfaces in Pen and Ink,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 469–476, ACM, 1996.
- [3] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, “Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications,” in *Visualization, 2003. VIS 2003. IEEE*, pp. 513–520, IEEE, 2003.
- [4] S. Rusinkiewicz, M. Burns, and D. DeCarlo, “Exaggerated Shading for Depicting Shape and Detail,” in *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 1199–1205, ACM, 2006.
- [5] S. Bruckner and M. E. Gröller, “Style Transfer Functions for Illustrative Volume Rendering,” in *Computer Graphics Forum*, vol. 26, pp. 715–724, Wiley Online Library, 2007.
- [6] T. Richmond, *The Mad Art of Caricature!: A Serious Guide to Drawing Funny Faces*. Deadline Demon Pub., 2011.
- [7] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” in *Computer Vision, 1998. Sixth International Conference on*, pp. 839–846, IEEE, 1998.
- [8] Z. Hossain and T. Möller, “Edge Aware Anisotropic Diffusion for 3D Scalar Data,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1376–1385, 2010.
- [9] K. Krissian, G. Malandain, and N. Ayache, *Directional Anisotropic Diffusion Applied to Segmentation of Vessels in 3D Images*. Springer, 1997.
- [10] R. Carmona, S. Zhong, *et al.*, “Adaptive Smoothing Respecting Feature Directions,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 353–358, 1998.

- [11] W. Lorensen and H. Cline, “Marching Cubes: A High Resolution 3D Surface Construction Algorithm,” in *ACM Siggraph Computer Graphics*, vol. 21, pp. 163–169, ACM, 1987.
- [12] L. Kobbelt, M. Botsch, U. Schwanerke, and H. Seidel, “Feature Sensitive Surface Extraction from Volume Data,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 57–66, ACM, 2001.
- [13] M. Hadwiger, P. Ljung, C. Salama, and T. Ropinski, “Advanced Illumination Techniques for GPU-based Volume Raycasting,” in *ACM SIGGRAPH 2009 Courses*, p. 2, ACM, 2009.
- [14] N. Max, “Optical Models for Direct Volume Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 1, no. 2, pp. 99–108, 1995.
- [15] W. Krueger, “The Application of Transport Theory to Visualization of 3D Scalar Data Fields,” in *Proceedings of the 1st conference on Visualization'90*, pp. 273–280, IEEE Computer Society Press, 1990.
- [16] N. Inc., “Nvidia’s Next Generation Cuda Compute Architecture,” *White paper*, 2009.
- [17] C. Sigg and M. Hadwiger, “Fast Third-Order Texture Filtering,” in *GPU Gems 2*, pp. 313–329, Addison-Wesley, Nvidia, 2005.
- [18] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Raghu, R. Machiraju, and J. Lee, “The Transfer Function Bake-Off,” *Computer Graphics and Applications, IEEE*, vol. 21, no. 3, pp. 16–22, 2001.
- [19] M. Levoy, “Display of Surfaces from Volume Data,” *Computer Graphics and Applications, IEEE*, vol. 8, no. 3, pp. 29–37, 1988.
- [20] S. Bruckner, S. Grimm, A. Kanitsar, and M. Gröller, “Illustrative Context-Preserving Exploration of Volume Data,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 6, pp. 1559–1569, 2006.
- [21] J. Kniss, G. Kindlmann, and C. Hansen, “Multidimensional Transfer Functions for Interactive Volume Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 3, pp. 270–285, 2002.
- [22] C. Correa and K. Ma, “Size-Based Transfer Functions: A New Volume Exploration Technique,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1380–1387, 2008.
- [23] J. Caban and P. Rheingans, “Texture-Based Transfer Functions for Direct Volume Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1364–1371, 2008.

- [24] J. Arvo, “Transfer Equations in Global Illumination,” *Global Illumination, SIGGRAPH '93 Course Notes*, vol. 42, 1993.
- [25] J. Kniss, S. Premoze, C. Hansen, and D. Ebert, “Interactive Translucent Volume Rendering and Procedural Modeling,” in *Visualization, 2002. VIS 2002. IEEE*, pp. 109–116, IEEE, 2002.
- [26] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson, “A Model For Volume Lighting and Modeling,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 2, pp. 150–162, 2003.
- [27] F. Lindemann and T. Ropinski, “Advanced Light Material Interaction for Direct Volume Rendering,” *Eurographics/IEEE VGTC on Volume Graphics*, pp. 101–108, 2010.
- [28] J. Blinn, “Models of Light Reflection for Computer Synthesized Pictures,” in *ACM SIGGRAPH Computer Graphics*, vol. 11, pp. 192–198, ACM, 1977.
- [29] F. Lindemann and T. Ropinski, “About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 1922–1931, 2011.
- [30] T. Ritschel, C. Dachsbacher, T. Grosch, and J. Kautz, “The State of the Art in Interactive Global Illumination,” in *Computer Graphics Forum*, vol. 31, pp. 160–188, Wiley Online Library, 2012.
- [31] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-Time Volume Graphics*. Natick, MA, USA: A. K. Peters, Ltd., 2006.
- [32] T. Möller, R. Machiraju, K. Mueller, and R. Yagel, “A Comparison of Normal Estimation Schemes,” in *Visualization'97., Proceedings*, pp. 19–26, IEEE, 1997.
- [33] B. Csébfalvi, “An Evaluation of Prefiltered Reconstruction Schemes for Volume Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 2, pp. 289–301, 2008.
- [34] U. Alim, T. Möller, and L. Condat, “Gradient Estimation Revitalized,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1495–1504, 2010.
- [35] Z. Hossain, U. R. Alim, and T. Moller, “Toward High-Quality Gradient Estimation on Regular Lattices,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 4, pp. 426–439, 2011.
- [36] A. Belyaev, “On Implicit Image Derivatives and Their Applications,” tech. rep., Heriot-Watt University, Edinburgh, 2011.

- [37] S. R. Marschner and R. J. Lobb, “An Evaluation of Reconstruction Filters for Volume Rendering,” in *Proceedings of the Conference on Visualization’94*, pp. 100–107, IEEE Computer Society Press, 1994.
- [38] B. Gooch and A. Gooch, *Non-Photorealistic Rendering*. Natick, MA. (254 pages hardback): AK Peters Ltd, July 2001.
- [39] A. Gooch, B. Gooch, P. Shirley, and E. Cohen, “A Non-Photorealistic Lighting Model for Automatic Technical Illustration,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 447–452, ACM, 1998.
- [40] P. Rheingans and D. Ebert, “Volume Illustration: Nonphotorealistic Rendering of Volume Models,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 7, no. 3, pp. 253–264, 2001.
- [41] Z. Nagy, J. Schneider, and R. Westermann, “Interactive volume illustration,” in *Proceedings of Vision, Modeling, and Visualization*, vol. 2, 2002.
- [42] A. Lu, C. Morris, J. Taylor, D. Ebert, C. Hansen, P. Rheingans, and M. Hartner, “Illustrative Interactive Stipple Rendering,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 9, no. 2, pp. 127–138, 2003.
- [43] R. Van Pelt, A. Vilanova, and H. Van De Wetering, “Illustrative Volume Visualization Using GPU-based Particle Systems,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 4, pp. 571–582, 2010.
- [44] P.-P. J. Sloan, W. Martin, A. Gooch, and B. Gooch, “The Lit Sphere: Volume Capturing NPR Shading from Art,” in *Graphics Interface*, vol. 2001, pp. 143–150, Citeseer, 2001.
- [45] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross, “Real-Time Ray-Casting and Advanced Shading of Discrete Isosurfaces,” in *Computer Graphics Forum*, vol. 24, pp. 303–312, Wiley Online Library, 2005.
- [46] D. DeCarlo and S. Rusinkiewicz, “Highlight Lines for Conveying Shape,” in *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 63–70, ACM, 2007.
- [47] B. Gooch, P. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld, “Interactive Technical Illustration,” in *Proceedings of the 1999 Symposium on Interactive 3D Graphics*, pp. 31–38, ACM, 1999.
- [48] A. Hertzmann, “Non-Photorealistic Rendering and the Science of Art,” in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 147–157, ACM, 2010.

- [49] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz, “Where Do People Draw Lines?,” in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 88, ACM, 2008.
- [50] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh, “How Well do Line Drawings Depict Shape?,” in *ACM Transactions on Graphics (ToG)*, vol. 28, p. 28, ACM, 2009.
- [51] R. M. Haralick, “Ridges and Valleys on Digital Images,” *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 28–38, 1983.
- [52] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, “Ridges for Image Analysis,” *Journal of Mathematical Imaging and Vision*, vol. 4, no. 4, pp. 353–373, 1994.
- [53] A. López and D. Lloret, “On Ridges and Valleys,” in *Proceedings of the International Conference on Pattern Recognition-Volume 4*, p. 4059, IEEE Computer Society, 2000.
- [54] A. G. Belyaev, A. A. Pasko, and T. L. Kunii, “Ridges and Ravines on Implicit Surfaces,” in *Computer Graphics International, 1998. Proceedings*, pp. 530–535, IEEE, 1998.
- [55] D. Eberly, *Ridges in Image and Aata Analysis*, vol. 7. Springer Science & Business Media, 1996.
- [56] T. Judd, F. Durand, and E. Adelson, “Apparent Ridges for Line Drawing,” in *ACM Transactions on Graphics (TOG)*, vol. 26, p. 19, ACM, 2007.
- [57] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, “Suggestive Contours for Conveying Shape,” *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 848–855, 2003.
- [58] D. DeCarlo, A. Finkelstein, and S. Rusinkiewicz, “Interactive Rendering of Suggestive Contours with Temporal Coherence,” in *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*, pp. 15–145, ACM, 2004.
- [59] M. Kolomenkin, I. Shimshoni, and A. Tal, “Demarcating Curves for Shape Illustration,” in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 157, ACM, 2008.
- [60] M. Burns, J. Klawe, S. Rusinkiewicz, A. Finkelstein, and D. DeCarlo, “Line Drawings from Volume Data,” in *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 512–518, ACM, 2005.
- [61] G. Kindlmann, “Transfer Functions in Direct Volume Rendering: Design, Interface, Interaction,” *Course notes of ACM SIGGRAPH*, vol. 3, 2002.

- [62] T. Goodwin, I. Vollick, and A. Hertzmann, “Isophote Distance: A Shading Approach to Artistic Stroke Thickness,” in *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 53–62, ACM, 2007.
- [63] C. Tietjen, T. Isenberg, and B. Preim, “Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning,” in *EuroVis*, pp. 303–310, 2005.
- [64] F. Ritter, C. Hansen, V. Dicken, O. Konrad, B. Preim, and H.-O. Peitgen, “Real-Time Illustration of Vascular Structures,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 877–884, 2006.
- [65] S. Bruckner, S. Grimm, A. Kanitsar, and M. Gröller, “Illustrative Context-Preserving Volume Rendering,” in *Proceedings of EUROVIS*, vol. 2005, pp. 69–76, 2005.
- [66] S. Bruckner, M. E. Gröller, K. Mueller, B. Preim, and D. Silver, “Illustrative Focus+Context Approaches in Interactive Volume Visualization,” *Dagstuhl Follow-Ups*, vol. 1, 2010.
- [67] S. Bruckner, P. Rautek, I. Viola, M. Roberts, M. C. Sousa, and M. E. Gröller, “Hybrid Visibility Compositing and Masking for Illustrative Rendering,” *Computers & Graphics*, vol. 34, no. 4, pp. 361–369, 2010.
- [68] L. Yaroslavsky, “Digital Picture Processing: An Introduction,” 1985.
- [69] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, *Bilateral Filtering: Theory and Applications*. Now Publishers Inc, 2009.
- [70] F. Durand and J. Dorsey, “Fast Bilateral Filtering for the Display of High-Dynamic-Range Images,” in *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 257–266, ACM, 2002.
- [71] S. Paris and F. Durand, “A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach,” in *European Conference on Computer Vision*, pp. 568–580, Springer, 2006.
- [72] A. Yezzi, “Modified Curvature Motion for Image Smoothing and Enhancement,” 1998.
- [73] J. Chen, S. Paris, and F. Durand, “Real-Time Edge-Aware Image Processing with the Bilateral Grid,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 103, 2007.
- [74] B. Weiss, “Fast Median and Bilateral Filtering,” *Acm Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 519–526, 2006.

- [75] F. Porikli, “Constant Time $O(1)$ Bilateral Filtering,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.
- [76] K. N. Chaudhury, D. Sage, and M. Unser, “Fast Bilateral Filtering using Trigonometric Range Kernels,” *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3376–3382, 2011.
- [77] E. Bethel, “High Performance, Three-Dimensional Bilateral Filtering,” tech. rep., Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2008.
- [78] P. Choudhury and J. Tumblin, “The Trilateral Filter for High Contrast Images and Meshes,” in *ACM SIGGRAPH 2005 Courses*, p. 5, ACM, 2005.
- [79] W. C. Wong, A. C. Chung, and S. C. Yu, “Trilateral Filtering for Biomedical Images,” in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pp. 820–823, IEEE, 2004.
- [80] A. Buades, B. Coll, and J.-M. Morel, “The Staircasing Effect in Neighborhood Filters and its Solution,” *Image Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 1499–1505, 2006.
- [81] M. Elad, “On the Origin of the Bilateral Filter and Ways to Improve It,” *Image Processing, IEEE Transactions on*, vol. 11, no. 10, pp. 1141–1151, 2002.
- [82] D. Barash, *Bilateral Filtering and Anisotropic Diffusion: Towards a Unified Viewpoint*. Springer, 2001.
- [83] D. Barash, “Fundamental Relationship between Bilateral filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 844–847, 2002.
- [84] A. Buades, B. Coll, and J.-M. Morel, “Neighborhood Filters and PDEs,” *Numerische Mathematik*, vol. 105, no. 1, pp. 1–34, 2006.
- [85] D. Barash and D. Comaniciu, “A Common Framework for Nonlinear Diffusion, Adaptive Smoothing, Bilateral Filtering and Mean Shift,” *Image and Vision Computing*, vol. 22, no. 1, pp. 73–81, 2004.
- [86] K. He, J. Sun, and X. Tang, “Guided Image Filtering,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [87] J. Weickert, *Anisotropic Diffusion in Image Processing*, vol. 1. Teubner Stuttgart, 1998.
- [88] J. Weickert, “Theoretical Foundations of Anisotropic Diffusion in Image Processing,” in *Theoretical Foundations of Computer Vision*, pp. 221–236, Springer, 1996.
- [89] J. Weickert, “A Review of Nonlinear Diffusion Filtering,” in *International Conference on Scale-Space Theories in Computer Vision*, pp. 1–28, Springer, 1997.

- [90] L. Alvarez, P.-L. Lions, and J.-M. Morel, “Image Selective Smoothing and Edge Detection by Nonlinear Diffusion II,” *SIAM Journal on Numerical Analysis*, vol. 29, no. 3, pp. 845–866, 1992.
- [91] G. Sapiro, “From Active Contours to Anisotropic Diffusion: Connections between Basic PDEs in Image Processing,” in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1, pp. 477–480, IEEE, 1996.
- [92] K. A. Brakke, *The Motion of a Surface by Its Mean Curvature*. Princeton University Press, 2015.
- [93] R. T. Whitaker, “Volumetric Deformable Models: Active Blobs,” in *Visualization In Biomedical Computing 1994*, pp. 122–134, International Society for Optics and Photonics, 1994.
- [94] R. T. Whitaker and X. Xue, “Variable-Conductance, Level-Set Curvature for Image Denoising,” in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, pp. 142–145, IEEE, 2001.
- [95] S. Osher and J. A. Sethian, “Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [96] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3. Cambridge University Press, 1999.
- [97] P. Smereka, “Semi-Implicit Level Set Methods for Curvature and Surface Diffusion Motion,” *Journal of Scientific Computing*, vol. 19, no. 1, pp. 439–456, 2003.
- [98] D. Chopp and J. A. Sethian, “Motion By Intrinsic Laplacian of Curvature,” *Interfaces and Free Boundaries*, vol. 1, no. 1, pp. 107–123, 1999.
- [99] L. C. Evans, J. Spruck, *et al.*, “Motion of Level Sets by Mean Curvature I,” *J. Diff. Geom.*, vol. 33, no. 3, pp. 635–681, 1991.
- [100] R. Malladi and J. A. Sethian, “Image Processing: Flows Under Min/Max Curvature and Mean Curvature,” *Graphical Models and Image Processing*, vol. 58, no. 2, pp. 127–141, 1996.
- [101] 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*, vol. 27, no. 3, pp. 217–229, 2007.
- [102] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2006.

- [103] R. Malladi and J. A. Sethian, “Image Processing via Level Set Curvature Flow,” *Proceedings of the National Academy of Sciences*, vol. 92, no. 15, pp. 7046–7050, 1995.
- [104] U. Clarenz, U. Diewald, and M. Rumpf, “Anisotropic Geometric Diffusion in Surface Processing,” in *Proceedings of the Conference on Visualization’00*, pp. 397–405, IEEE Computer Society Press, 2000.
- [105] Y.-L. You and M. Kaveh, “Fourth-Order Partial Differential Equations for Noise Removal,” *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, 2000.
- [106] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, “Geometric Surface Smoothing via Anisotropic Diffusion of Normals,” in *Proceedings of the Conference on Visualization’02*, pp. 125–132, IEEE Computer Society, 2002.
- [107] M. R. Hajiaboli, “An Anisotropic Fourth-Order Partial Differential Equation for Noise Removal,” in *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 356–367, Springer, 2009.
- [108] P. Guidotti and K. Longo, “Well-Posedness for a Class of Fourth Order Diffusions for Image Processing,” *Nonlinear Differential Equations and Applications NoDEA*, vol. 18, no. 4, pp. 407–425, 2011.
- [109] P. Guidotti and K. Longo, “Two Enhanced Fourth Order Diffusion Models for Image Denoising,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 2, pp. 188–198, 2011.
- [110] S. Didas and J. Weickert, “Combining Curvature Motion and Edge-Preserving Denoising,” in *Scale Space and Variational Methods in Computer Vision*, pp. 568–579, Springer, 2007.
- [111] M. Rumpf and T. Preusser, “A Level Set Method for Anisotropic Geometric Diffusion in 3D Image Processing,” *SIAM Journal on Applied Mathematics*, vol. 62, no. 5, pp. 1772–1793, 2002.
- [112] U. Clarenz, F. Haußer, M. Rumpf, A. Voigt, and U. Weikard, “On Level Set Formulations for Anisotropic Mean Curvature Flow and Surface Diffusion,” in *Multiscale Modeling in Epitaxial Growth*, pp. 227–237, Springer, 2005.
- [113] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear Total Variation Based Noise Removal Algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.
- [114] A. Chambolle, “An Algorithm for Total Variation Minimization and Applications,” *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

- [115] A. Chambolle, “An Algorithm for Mean Curvature Motion,” *Interfaces and Free Boundaries*, vol. 6, no. 2, pp. 195–218, 2004.
- [116] P. Mullen, A. McKenzie, Y. Tong, and M. Desbrun, “A Variational Approach to Eulerian Geometry Processing,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 66, 2007.
- [117] F. Mémoli, G. Sapiro, and S. Osher, “Solving Variational Problems and Partial Differential Equations Mapping into General Target Manifolds,” *Journal of Computational Physics*, vol. 195, no. 1, pp. 263–292, 2004.
- [118] A. Oberman, S. Osher, R. Takei, and R. Tsai, “Numerical Methods for Anisotropic Mean Curvature Flow Based on a Discrete Time Variational Formulation,” *Commun. Math. Sci*, vol. 9, no. 3, pp. 637–662, 2011.
- [119] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, vol. 147. Springer Science & Business Media, 2006.
- [120] A. Buades, B. Coll, and J.-M. Morel, “A Non-Local Algorithm for Image Denoising,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 2, pp. 60–65, IEEE, 2005.
- [121] A. Buades, B. Coll, and J.-M. Morel, “A Review of Image Denoising Algorithms, with a New One,” *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [122] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, “An Optimized Blockwise Nonlocal Means Denoising Filter for 3D Magnetic Resonance Images,” *IEEE transactions on medical imaging*, vol. 27, no. 4, pp. 425–441, 2008.
- [123] C. B. Burckhardt, “Speckle in Ultrasound B-Mode Scans,” *IEEE Transactions on Sonics and ultrasonics*, vol. 25, no. 1, pp. 1–6, 1978.
- [124] R. N. Czerwinski, D. L. Jones, and W. D. O’Brien, “Ultrasound Speckle Reduction by Directional Median Filtering,” in *Image Processing, 1995. Proceedings, International Conference on*, vol. 1, pp. 358–361, IEEE, 1995.
- [125] S. Gupta, R. Chauhan, and S. Sexana, “Wavelet-Based Statistical Approach for Speckle Reduction in Medical Ultrasound Images,” *Medical and Biological Engineering and computing*, vol. 42, no. 2, pp. 189–192, 2004.
- [126] Y. Yu and S. T. Acton, “Speckle Reducing Anisotropic Diffusion,” *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1260–1270, 2002.
- [127] K. Z. Abd-Elmoniem, A.-B. Youssef, and Y. M. Kadah, “Real-Time Speckle Reduction and Coherence Enhancement in Ultrasound Imaging via Nonlinear

- Anisotropic Diffusion,” *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 9, pp. 997–1014, 2002.
- [128] K. Krissian, C.-F. Westin, R. Kikinis, and K. G. Vosburgh, “Oriented Speckle Reducing Anisotropic Diffusion,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1412–1424, 2007.
 - [129] P. Coupé, P. Hellier, C. Kervrann, and C. Barillot, “Nonlocal Means-Based Speckle Filtering for Ultrasound Images,” *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2221–2229, 2009.
 - [130] C. Kervrann, J. Boulanger, and P. Coupé, “Bayesian Non-Local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal,” in *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 520–532, Springer, 2007.
 - [131] G. Di Martino, A. Di Simone, A. Iodice, and D. Riccio, “Scattering-Based Non-local Means SAR Despeckling,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3574–3588, 2016.
 - [132] J. Kang, J. Y. Lee, and Y. Yoo, “A New Feature-Enhanced Speckle Reduction Method Based on Multiscale Analysis for Ultrasound B-Mode Imaging,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1178–1191, 2016.
 - [133] P. Coupé, P. Yger, and C. Barillot, “Fast Non Local Means Denoising for 3D MR Images,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 33–40, Springer, 2006.
 - [134] A. Baer, C. Tietjen, R. Bade, and B. Preim, “Hardware-Accelerated Stippling of Surfaces derived from Medical Volume Data,” in *EuroVis*, pp. 235–242, 2007.
 - [135] J. Zander, T. Isenberg, S. Schlechtweg, and T. Strothotte, “High Quality Hatching,” in *Computer Graphics Forum*, vol. 23, pp. 421–430, Wiley Online Library, 2004.
 - [136] R. Gasteiger, C. Tietjen, A. Baer, and B. Preim, “Curvature- and Model-Based Surface Hatching of Anatomical Structures Derived from Clinical Volume Datasets,” in *International Symposium on Smart Graphics*, pp. 255–262, Springer, 2008.
 - [137] M. Gerl and T. Isenberg, “Interactive Example-Based Hatching,” *Computers & Graphics*, vol. 37, no. 1, pp. 65–80, 2013.
 - [138] S. Bruckner, *Interactive Illustrative Volume Visualization*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2008.
 - [139] E. Laning, *The Act of Drawing*. McGraw-Hill Companies, 1971.
 - [140] J. F. Blinn and M. E. Newell, “Texture and reflection in computer generated images,” *Communications of the ACM*, vol. 19, no. 10, pp. 542–547, 1976.

- [141] N. Greene, “Environment mapping and other applications of world projections,” *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 21–29, 1986.
- [142] W. Li, L. Ritter, M. Agrawala, B. Curless, and D. Salesin, “Interactive Cutaway Illustrations of Complex 3D Models,” in *ACM Transactions on Graphics (TOG)*, vol. 26, p. 31, ACM, 2007.
- [143] D. Weiskopf, K. Engel, and T. Ertl, “Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 298–312, 2003.
- [144] M. Burns, M. Haidacher, W. Wein, I. Viola, and E. Groeller, “Feature Emphasis and Contextual Cutaways for Multimodal Medical Visualization,” in *EuroVis*, vol. 7, pp. 275–282, 2007.
- [145] A. Birkeland, S. Bruckner, A. Brambilla, and I. Viola, “Illustrative Membrane Clipping,” in *Computer Graphics Forum*, vol. 31, pp. 905–914, Wiley Online Library, 2012.
- [146] C. Sigg and M. Hadwiger, “Fast Third-Order Texture Filtering,” in *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, pp. 313–329, Addison-Wesley Professional, 2005.
- [147] Y. Ohtake, A. Belyaev, and H.-P. Seidel, “Ridge-Valley Lines on Meshes via Implicit Surface Fitting,” *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 609–612, 2004.
- [148] E. S. Gastal and M. M. Oliveira, “Domain Transform for Edge-Aware Image and Video Processing,” in *ACM Transactions on Graphics (TOG)*, vol. 30, p. 69, ACM, 2011.
- [149] P. Perona and J. Malik, “Scale-Space and Edge Detection Using Anisotropic Diffusion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.
- [150] G. Gerig, O. Kübler, R. Kikinis, F. Jolesz, *et al.*, “Nonlinear Anisotropic Filtering of MRI Data,” *Medical Imaging, IEEE Transactions on*, vol. 11, no. 2, pp. 221–232, 1992.
- [151] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, “Robust Anisotropic Diffusion,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, 1998.
- [152] J. Weickert and B. Benhamouda, “A Semidiscrete Nonlinear Scale-Space Theory and its Relation to the Perona-Malik Paradox,” in *Advances in Computer Vision*, pp. 1–10, Springer, 1997.

- [153] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel, “Axioms and Fundamental Equations of Image Processing,” *Archive for Rational Mechanics and Analysis*, vol. 123, no. 3, pp. 199–257, 1993.
- [154] S. Osher and L. I. Rudin, “Feature-Oriented Image Enhancement using Shock Filters,” *SIAM Journal on Numerical Analysis*, vol. 27, no. 4, pp. 919–940, 1990.
- [155] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, “Geometric Surface Processing via Normal Maps,” *ACM Transactions on Graphics (TOG)*, vol. 22, no. 4, pp. 1012–1033, 2003.
- [156] S. Paris and F. Durand, “A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach,” *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.
- [157] R. Malladi and J. A. Sethian, “Level Set Methods for Curvature Flow, Image Enhancement and Shape Recovery in Medical Images,” in *Visualization and Mathematics*, pp. 329–345, Springer, 1997.
- [158] W. Kühnel, *Differential Geometry: Curves-Surfaces-Manifolds*, vol. 16. American Mathematical Soc., 2006.
- [159] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, vol. 2. Prentice-hall Englewood Cliffs, 1976.
- [160] D. Mitchell and P. Hanrahan, “Illumination from Curved Reflectors,” *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 283–291, 1992.
- [161] G. M. Turkiyyah, D. W. Storti, M. Ganter, H. Chen, and M. Vimawala, “An Accelerated Triangulation Method for Computing the Skeletons of Free-Form Solid Models,” *Computer-Aided Design*, vol. 29, no. 1, pp. 5–19, 1997.
- [162] R. Goldman, “Curvature Formulas for Implicit Curves and Surfaces,” *Computer Aided Geometric Design*, vol. 22, no. 7, pp. 632–658, 2005.