Variational methods for geometry processing

Wu, Xiaoqun

2014

Wu, X. (2014). Variational methods for geometry processing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61854>

<https://doi.org/10.32657/10356/61854>

NANYANG TECHNOLOGICAL UNIVERSITY

VARIATIONAL METHODS FOR GEOMETRY PROCESSING

BY

WU XIAOQUN

A THESIS SUBMITTED TO THE NANYANG TECHNOLOGICAL UNIVERSITY IN

FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTER ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

2014

Abstract

Inspired by the success of variational techniques in various applications such as image processing and recent advance in sparse norm minimization, this research explores a class of variational methods with sparse norm for digital geometry processing which has to deal with noisy data and non-smooth features. The variational model mainly contains two terms: one is to keep the intrinsic properties of the function and the other is to make the function satisfy some prior knowledge. Three types of geometry processing problems are studied under the similar variational framework. The research focuses on the specific formulation of the variational model and efficient numerical solvers. Its objective is to develop novel geometry processing algorithms that can effectively and robustly process geometric models characterized by the co-existence of non-smooth features and noise/outliers, which is usually difficult in prior art.

The first problem addressed in the thesis is decomposing color attributes of 3D triangular mesh models into a piecewise smooth structure component and a fine scale-detailed texture component. We formulate this as image decomposition on manifolds. The manifold is represented as a triangular mesh equipped with inner products and norms. A TV-based variational model is proposed for decomposition on manifolds. Unlike planar image decomposition, the intrinsic geometry of the manifold is taken into account in the proposed model. The TV regularization effectively extracts the structural image, and one of the L^2 , L^1 and G norms is used in the fidelity term to adapt to different properties of textures. Efficient numerical algorithms, based on augmented Lagrangian methods, are developed. The effectiveness of the proposed methods are demonstrated by many examples and two applications.

The second problem considered in the thesis is how to denoise 3D triangular meshes that may contain small scale geometry details, sharp/semi-sharp features, noise and outliers. We formulate this as a problem of geometric data decomposition on manifolds. A specially designed edge differential operator is employed and L^1 norm is also used for the

regularization in the variational model. Different from previous variational model, here the function domain is coupled with the target mesh, changing in the denoising process, which makes the problem challenging. Therefore we propose an iterative procedure to solve the variational problem, which gives a robust, feature-preserving triangular mesh denoising algorithm.

The third problem is curve and surface reconstruction. We begin with B-spline contour reconstruction from unorganized points. We define a new regularization term based on the total Laplacian (TL) for feature preservation. A $TL-L^1$ variational model is proposed to constrain both deBoor points and knots. Geometric features and noise/outliers are well decoupled without preprocessing. Next we extend the approach to B-spline surface fitting where the underlying surface is assumed to be topologically equivalent to a rectangle. The proposed total variation (TV) and L^1 norm based minimization method finds the optimal knots and control points simultaneously. Finally, to efficiently reconstruct surfaces with arbitrary topology, we consider subdivision surface reconstruction. In order to faithfully model surfaces with sharp or semi-sharp features, a new subdivision scheme with continuous sharpness control is proposed. A key idea here is to introduce a real number for each edge to continuously control the sharp/semi-sharp features, which plays the similar role as knots in B-splines. A $TV-L^1$ variational model is built to find the best sharpness and positions of the control points automatically. One important feature that distinguishes our three reconstruction algorithm from prior art is that we treat the knots or sharpness as unknowns together with the control points in the minimization problem. As a result, the optimal knots and sharpness are automatically obtained, which respect the characteristics of geometric features. Experiments demonstrate that the proposed reconstruction methods can well preserve the sharp and semi-sharp features and are robust to various noise and outliers.

Acknowledgements

Foremost, I would like to thank my supervisor Prof. Zheng Jianmin, for his patient guidance, encouragement and constant support on my Ph.D. study and research. He always provides insightful discussions and suggestions about my research and guides me step forward. I am also very grateful to him for spending so much time on the thesis. His great knowledge and serious attitude will benefit me a lot in my future career life. Besides, I would like to thank Prof. Cai Yiyu and Prof. Fu Chi-Wing for their valuable advices and financial support.

My sincere thanks also goes to Dr. Wu Chunlin for his instructions and discussions, and helping me clarify the theoretical part of this thesis.

I would like to thank my colleagues and friends, especially Dr. Ma Yuewen, Dr. Wang Kai, Li Yusha, Dr. Zhou Hailing, Zhang Wenjing, Zhang Yuzhe, Zhang Juzheng, Dr. Luo Guomin, Zhao Ming, Dr. Yang Xunnian, Yan Xiaoqi, Dr. Song Peng, Lai William, Dr. Meng Min, Zhang Minqi, Dr. Chen Renjie, and many others, for their help during my past four-year Ph.D. life.

Last but not the least, I would like to extend deepest appreciations to my family for their supporting and encouragement. Special thanks to my husband Dr. Zhang Lei for all his love.

Contents

Abstract	i
Acknowledgements	iii
Table of Contents	v
List of Figures	ix
List of Tables	xv
List of Abbreviations	xvi
List of Symbols	xvii
1 Introduction	1
1.1 Background	1
1.2 Research Problem and Objectives	3
1.3 Research Contributions	4
1.4 Thesis Organization	7
2 Literature Review	8
2.1 Variational Methods	8
2.1.1 Regularization term	10
2.1.2 Fidelity term	13
2.1.3 Numerical solvers	15
2.2 3D Mesh Denoising	17
2.3 B-spline Curve and Surface Reconstruction	19
2.4 Subdivision Surface Reconstruction	21

3	Structure-Texture Image Decomposition on Manifolds	24
3.1	Motivation and Our Work	24
3.2	Image Decomposition Models on Manifolds	26
3.2.1	Three variational models	27
3.2.2	Discretization	30
3.3	Numerical Methods	33
3.3.1	TV- L^2	33
3.3.2	TV- L^1	36
3.3.3	TV-G	38
3.4	Experiments	41
3.5	Applications	43
3.5.1	Detail exaggeration	44
3.5.2	L^1 image denoising on manifolds	45
4	Mesh Denoising Using the ROF Model	48
4.1	Motivation and Our Work	48
4.2	Variational Mesh Denoising	52
4.3	Algorithm	55
4.4	Experiments	56
5	Unsupervised Construction of B-spline Contours from Unorganized Points	62
5.1	Motivation and Our work	62
5.2	Unsupervised Contour Construction	64
5.2.1	Basic ideas	64
5.2.2	Algorithm	67
5.3	$TL - L^1$ B-spline Fitting	69
5.3.1	$\mathbf{P}$ -sub problem	70
5.3.2	$\mathcal{T}$ -sub problem	72
5.3.3	$\mathbf{T}$ -sub problem	74

5.3.4	S-sub problem	76
5.3.5	Update Lagrangian multipliers	76
5.4	Experiments	77
5.4.1	$TL - L^1$ methods	77
5.4.2	Adaptive $TL - L^1$ B-spline fitting	78
5.4.3	Error comparison	79
6	TV-L^1 Optimization for B-spline Surface Reconstruction with Sharp Features	82
6.1	Motivation and Our work	82
6.2	Surface Reconstruction Problem	84
6.3	TV- L^1 Optimization Model for B-spline Surface Reconstruction	85
6.4	Numerical algorithm	87
6.5	Experiments	91
6.5.1	Three examples	91
6.5.2	Robustness against noise and sharp feature preservation	94
6.5.3	Error comparison and performance	95
7	Feature-aware Subdivision Surface Reconstruction	98
7.1	Motivation and Our work	98
7.2	Subdivision with Continuous Sharpness Control	100
7.3	Variational Subdivision Surface Reconstruction	103
7.3.1	Variational model	104
7.3.2	Numerical solver	106
7.4	Experiments	109
8	Conclusions and Future Work	113
8.1	Summary	113
8.2	Future Work	115
	References	117

List of Figures

2.1	Denoising results by (2.4) and (2.2).	11
2.2	Comparison of TV- L^2 and TV- L^1 for image denoising.	13
2.3	Decomposition of a gray image with oscillating pattern. (b) and (d) are the u components; (c) and (e) are the v components.	15
3.1	Local parameterization of a triangle.	32
3.2	The top row is conventional L^2 and L^1 image decomposition on 2D plane. The bottom row is our proposed decomposition results on planar manifold. (b) and (d) are the structures. (c) and (e) are the textures.	43
3.3	Proposed image decomposition on a triangular surface. (a) is the input textured bunny. (b) and (d) are the extracted structures extracted using L^2 and L^1 norms, respectively. (c) and (e) are the extracted textural parts using L^2 and L^1 norms, respectively.	44
3.4	Decomposition of an image with oscillatory pattern by the proposed method. (a) is the input image with oscillatory pattern. (b), (c) and (d) are the extracted structural components using L^2 , L^1 and G norms, respectively. (e), (f) and (g) are the extracted textural components using L^2 , L^1 and G norms, respectively.	44

3.5	Decomposition of an image with sparse textures by the proposed method. (a) is the input image with salt and pepper noise. (b) and (c) are the structural image and noise extracted from (a) using L^2 norm. (d) and (e) are the structural image and noise extracted from (a) using L^1 norm. (f) and (g) are the structural image and noise extracted from (a) using G norm. . . .	45
3.6	Detail exaggeration. (a) is the textured cup, (b) and (d) are the extracted structural components using L^2 and L^1 norms. (c) and (e) are the exaggerated detail results by simply scaling the texture components. L^2 norm: $\lambda = 300, r_p = 0.012, t = 10.659s, N = 15$; L^1 norm: $\lambda = 100, r_p =$ $0.02, r_z = 200, t = 11.225s, N = 16$	46
3.7	Comparison of the proposed method and the L^2 method of [148] for denoising bunny 2. The models in the first row are gray and color textured manifolds added with 10%, 20% salt and pepper noise, respectively. The models in the second row are the denoising results using the method of [148]. The models in the third row are the denoising results using the proposed TV- L^1 method.	47
4.1	(a) is the noisy fish smoothed by L^0 norm, (b) is the noisy fish smoothed L^1 norm, (c) top is the piecewise constant shape smoothed by L^0 norm and (c) bottom is smoothed by L^1 norm.	51
4.2	One ring of a triangle (left) and edge (right).	53
4.3	From left to right, top to down: the original surface, the input surface corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.4l_e$, bilateral filtering [49], prescribed mean curvature flow [62], mean filtering [154], bilateral normal filtering [163], L^0 norm [61] and our method.	57

4.4	From left to right: surface corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.3l_e$, bilateral filtering [49], prescribed mean curvature flow [62], mean filtering [154], bilateral normal filtering [163], L^0 norm [61] and our method.	58
4.5	From left to right, top to down: the original Venus head model, the noisy input mesh corrupted by impulsive noise with variation $0.4l_e$ and Gaussian noise in random directions with standard deviation $\sigma = 0.15l_e$, the smoothed model by L^0 and our smoothed model.	59
4.6	Skull model. From left to right: the second row shows the noisy input, L^0 norm smoothing results by increasing smoothing term . The third row shows the original mesh, our smoothing results with fidelity tuning parameter parameter $6\lambda_0, 3\lambda_0, \lambda_0$ where λ_0 is the default fidelity parameter. The first and fourth rows are the local close-up views.	60
4.7	Bunny model. From left to right: the first row: the original model, the noisy input corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.2l_e$ and bilateral filtering [49]; The second row: the close-up views; The third row: bilateral normal filtering [163], L^0 norm [61] and our method; The fourth row: the close-up views.	61
5.1	(a) is the fitting result using E_2^s . (b) is the fitting result using our total Laplacian E_1^s	66
5.2	Comparison of different metrics, (a) is by MMSE. (b) is by $TL - L^2$ with $\lambda = 0.1$. (c) is by $TL - L^2$ with $\lambda = 0.01$. (d) is by MME. (e) is by $TL - L^1$ with $\lambda = 0.1$	66

5.3	$ATL - L^1$ algorithm. (a) is the quad-tree construction. The dark-red cell are the cells chosen for initial shape construction. (b) is the initial shape by connecting the centers of the dark-red cells in (a). (c) is the B-spline curve by our $ATL - L^1$ from the initial shape with 35 control points. (d) is the reconstructed B-spline curve.	67
5.4	Comparison of $TL - L^1, \lambda = 5$ without (a, c) and with (b, d) knot optimization.	77
5.5	Comparison among MMSE, MME and our $TL - L^1$ with the same knot vector $\mathcal{T}$	78
5.6	Comparison between $ATL - L^1$ method and MMSE method. (a) is the result by $ATL - L^1$ from the initial shape. (c) is the result by MMSE criterion with given parameter values and uniform knot vector.	79
5.7	Comparison of the $ATL - L^1$ method, MMSE, MME. (a), (b) and (c), (d) are the results by $ATL - L^1$ from their bounding boxes. (e), (f) and (g), (h) are the results by MMSE criterion with given parameter values and uniform knot vector. (i), (j) and (k), (l) are the results by MME criterion with given parameter values and uniform knot vector.	80
5.8	Adaptive B-spline fitting with different initial shapes.	81
6.1	(a) is the ground truth, and we add Gaussian noise and outliers to (a). (b) is the curve by (6.2) with $n = 7$. (c) is by (6.2) with $n = 35$. (d) is the B-spline curve in (e). (e) is the curve by our variational method with $n = 7$. (f) is the curve by our method with the same knot vector placement with (c).	86
6.2	Reconstruction of a piecewise linear surface with $\lambda = 100$	92
6.3	Reconstruction of Fandisk surface patch with NCP 10×20 and $\lambda = 40$. . .	93
6.4	Reconstruction of a crescent surface with 20×20 control points and $\lambda = 100$. . .	93

6.5	(a) is the input ground truth point data, and we add Gaussian noise and outliers to (a) getting (c). (b) is the result by (6.2), and (d) is the result by our approach, which keeps the overall smoothness and preserves the sharp features as well.	94
6.6	(a) is the noisy input. (b) is the reconstruction surface of [17] with NCP 5×7 . (c) is our proposed method with NCP 5×7 . (d) is the knot location mapping to 2D of (b) and (c). Different colors represent different multiplicity.	95
6.7	(a) is the noisy input with Gaussian Noise and outlier. (b) is the reconstructed surface of [17] with NCP 10×20 . (c) is our proposed method with NCP 10×20 . (d) is the knot location mapping to 2D of (b) and (c). Different colors represent different multiplicity.	96
6.8	(a) is the reconstructed surface of [17] with NCP 10×20 . (b) is our proposed approach with NCP 10×20 . (d) is the knot location mapping to 2D of (a) and (b). Different colors represent different multiplicity.	96
6.9	(a) is the reconstructed surface of [17] with NCP 25×20 . (b) is our proposed approach with NCP 25×20 . (d) is the knot location mapping to 2D of (a) and (b). Different colors represent different multiplicity.	97
7.1	Loop subdivision masks. The vertex mask (left) and the edge mask (right). .	100
7.2	Sharp feature masks. The blue line denotes the sharp edges. The red rectangle denotes the new inserted vertex.	101
7.3	Examples of various edge sharpness. The control mesh is drawn in wireframe. From left to right: $e.s = 0$ (gray), 0.6 (yellow), 0.9 (magenta), 1 (red).	103
7.4	Comparison between smoothing term using L^2 norm (left) and using total Laplacian (right). Close-up views are shown on the bottom.	105

7.5	The left is the input initial control mesh and the subdivision surface generated by smooth rule. The middle is the objective dense mesh. The right is the subdivision surface by optimizing the sharpness of the control mesh. The root mean error is $RME = 0.115403$ (left) and $RME=0.00768$ (right). The edge with non-zero sharpness are marked in yellow.	109
7.6	RockerArm model. (a) is the noisy dense mesh. (b) is the mesh after denoising. (c) is the initial mesh obtained by simplification. (d) is the reconstructed subdivision surface using classic Loop subdivision. (e) is the subdivision surface using the rules in [63]. (f) is the reconstructed subdivision surface using our method. The third row shows the close-up views of the subdivision surfaces in Row 2.	111
7.7	Vase model. From left to right: original model, subdivision surface reconstructed without sharpness, subdivision surface reconstructed using the rules of [63], and subdivision surface reconstructed using our method. .	112
7.8	Blade model. From left to right: original model, subdivision surface reconstructed without sharpness, subdivision surface reconstructed using the rules of [63], and subdivision surface reconstructed using our method. .	112

List of Tables

3.1	Statistics of the 3D meshes used in the experiments.	42
3.2	Statistics for Fig. 3.2 and Fig. 3.3, where N is the iteration number and t is the running time. The stop condition is $\epsilon_0 = 1.e^{-6}$	42
3.3	Statistics for Fig. 3.7 where N is the iteration number and t is the running time.	46
4.1	Statistic for Fig.4.2 and Fig. 4.3.	58
5.1	Statistics of contour-point number N , control point number n , $\bar{e}$, e_H in Fig. 5.4, Fig. 5.5, Fig. 5.6, Fig. 5.7, Fig. 5.8	81
6.1	Knot vector in Fig. 6.2	92
6.2	Error Comparison. NCP is the number of control points. The first row is the error of [17] and the second row is our error.	95
6.3	Performance of our approach.	95
7.1	Error comparison. Statistics of number of vertices, number of control point (NCP), the max error (ME), the root mean error (RME).	111
7.2	Performance of our approach.	112

List of Abbreviations

ALM	Augmented Lagrangian Method
LM	Levenberg-Marquardt
MMSE	Minimum Mean-Square Error
MME	Minimum Mean Error
ME	Mean Error
MAX	Max Error
NCP	Number of Control Points
PSO	Particle Swarm Optimization
ROF	Rudin, Osher, Fatemi
RME	Root Mean Error
SNR	Signal-to-Noise Ratios
TV	Total Variation
TL	Total Laplacian

List of Symbols

$\mathbb{R}$	Real number set
$diag(A)$	Diagonal element of matrix A
A^T	Transpose of matrix A
$\mathbf{r}'(u)$	First derivative of function $\mathbf{r}$
$\mathbf{r}''(u)$	Second derivative of function $\mathbf{r}$
∇u	Gradient of u
$\nabla_M u$	Intrinsic gradient of u on M
Δu	Laplacian of u
$ u $	Absolute value of u
$\ u\ _{L^p}$	L^p norm of u
$\ u\ _{L^\infty}$	L^∞ norm of u
$\ u\ _G$	G norm of u
$\langle u, v \rangle$	Inner product of u and v
$\ u\ _{1,M}$	L^1 norm of u on M
$\ u\ _{2,M}$	L^2 norm of u on M
$\ u\ _{G,M}$	G norm of u on M
$\langle u, v \rangle_M$	Inner product of u and v on M
$div(u)$	Divergence of u
$div_M(u)$	Divergence of u on M

Chapter 1

Introduction

1.1 Background

With the development of data acquisition techniques such as 3D scanner and Kinect, digital geometry has become a new type of multimedia data. 3D objects have gained wide applications in many applications such as product design, digital entertainment, biology, archeology, medical diagnostic and so on. Therefore, there is a strong need for the development of geometry processing techniques.

How to represent the underlying shapes from more detailed and complex data is a fundamental task in computer graphics and computer aided design. It is concerned with mathematical description and object modification. A suitable representation should be efficient for further geometry processing, such as rendering, editing, analyzing and manipulating. The properties like piecewise planar/constant, piecewise smooth and overall smooth with important geometry features are important to describe the underlying structure for the representation of shapes. Modeling surfaces for the underlying shapes with such properties will enhance various geometric algorithms.

To decompose objects into simple components with important underlying structure is also a fundamental problem in other research areas. In image processing, especially in computational photography, to find the underlying structure of an image can be used in the

fields of HDR tone mapping, flash/no-flash image fusion, image editing, image inpainting, matching, segmentation, and so on. In signal processing, once the structure is separated from details, they can be manipulated separately and then recombined.

Variational methods, in calculus of variation, are to deal with “functional”, that is function depends on the other function. Functionals are often expressed as definite integrals involving functions and their derivatives. For given function $f : \Omega \rightarrow \mathbb{R}$, mapping from $2D$ domain Ω to color value in $\mathbb{R}$, the variational approach to image processing constitutes the computation of a reconstructed image u based on the observed image (or more general data) f as a minimiser of a functional. Unlike traditional local approaches which find “better” solutions, variational minimization approaches attempt to find the “best” solutions globally. Variational techniques, in particular, total variation, have made great success recently, especially in image restoration, segmentation, decomposition, inpainting and reconstruction. Total variation norm is designed to capture the piecewise smooth/constant property of the function. Due to this property of the total variation, it is widely used in the problems which is needed to preserve the underlying structure of image.

Intensive research of variational methods has been conducted in last decades in image and signal processing. Especially recent years with the development of compressed sensing, a variety of variational approaches incorporated with L^p norm, usually $p < 2$, are proposed. However, there is still little work to extend this variational techniques with general L^p norm to geometry processing. The reasons are mainly three-fold:

- **Function:** It is not straightforward to define the function of 3D geometry as image and signals. The parametric surface representation is one kind of functions. However, how to represent 3D models by parametric surface is also a sophisticated problem since the parameterization is also a challenging problem.
- **Domain:** It is not easy to define the function on general domain like triangular mesh, quad mesh. Even for parametric surfaces, like NURBS, the structure of function domain also has considerable effect to the definition of function. Moreover, different

from the image case, that the domains of all the functions are the 2D fixed planar grids, the domain for 3D models may change simultaneity during the optimization processing.

- Differential operator: How to choose the differential operator of the function that can well capture the intrinsic properties of geometry is difficult. In image processing, the gradient of the function well captures the sharp edge. The problem becomes more complicated when in 3D geometry processing which may involve second order derivative, curvature information.

Inspired by the success of variational methods in image processing and the underlying structure requirement for applications in geometry processing, such as mesh smoothing, surface modeling and reconstruction, we are interested in variational methods and their applications in geometry processing.

Moreover, most previous work in geometry processing adopts L^2 norm as measure for various approximation problems. The measure of the approximation error term has considerable effect on the final results. Suitable norm types are required to be carefully designed according to different properties of data. Meanwhile, non-differential norms will bring numerical difficulties for solving, which needs new solvers to efficiently solve these problems.

1.2 Research Problem and Objectives

This research is to investigate how variational techniques can be efficiently used for geometry processing to enhance modeling of the underlying structures of 3D models by analyzing the properties of different data types and incorporating the L^p norm techniques. It aims to gain deep understanding of variational methods and digital geometry processing, and to develop efficient geometry processing algorithms such as decomposition, denoising, modeling and reconstruction. In particular, the research objectives are:

- To explore a class of variational methods with sparse norm for digital geometry processing. This variational model contains two main terms: one is to keep the intrinsic properties of the function and the other is to make the function satisfy some prior knowledge.
- To develop robust, feature preserving and effective digital geometry algorithms including decomposition of color attributes of triangular mesh, triangular mesh denoising, and curve and surface reconstruction.
- To develop efficient numerical solvers for nonlinear and non-differentiable variational model.

1.3 Research Contributions

To accomplish these objectives, we have introduced a general variational framework for geometry processing, which is to model various problems by minimizing functional. Our variational method mainly contains two terms: the regularization term and the fidelity term, with different types of norms to measure the data according to the properties of the functions. The regularization term is to maintain the underlying structure of the function, while the fidelity term is to make the function approximate the giving conditions.

The thesis mainly consists of five technical components: variational image decomposition on manifolds, 3D triangular mesh denoising, B-spline curve and surface reconstruction and feature-aware subdivision surface reconstruction.

- First, we consider the problem of decomposing the color attributes of a triangular mesh into structural and textural components as an example for developing a general variational framework for multi-channel data defined on a manifold represented by a triangular mesh. We formulate this as a variational problem, in which the total variation energy is used for extracting the structural part and based on the properties of texture one of three norms, L^2 , L^1 and G , is used in the fidelity

term for the textural part. While L^2 and G norms are used for texture of no *a-prior* knowledge or oscillating pattern, L^1 norm is used for structural or sparse texture. We develop efficient numerical methods to solve the proposed variational problems using augmented Lagrangian methods (ALM) when the manifold is represented by a triangular mesh. The contributions are two-fold: (1) We adapt the variational structure-texture image decomposition to manifolds, which takes the intrinsic property of manifolds into account. The non-quadratic fidelity terms with L^1 and G norms are extended to 3D triangular meshes for the first time. (2) We show how to efficiently tackle the variational problems with non-linearity/non-differentiability terms by iteratively solving some sub-problems that either have closed form solutions or are to solve linear equations. We demonstrate the effectiveness of the proposed methods with examples and applications in detail enhancement and impulsive noise removal.

- Second, we also adapt the variational ROF model for mesh denoising. A previous method based on L^0 norm produces high quality denoising results for piecewise constant/planar surfaces. In the other side, it also flattens the small scale geometry features because of the extreme penalty mechanism for non-zero values. We present a novel mesh denoising approach based on L^1 norm for preserving both sharp edge and small scale geometry.
- Third, we adapt the variational methods together with total variation and L^1 norm for parametric curves. A variational optimization method to reconstruct a B-spline curve from unorganized noisy contour points is presented. We define a new regularization term based on total Laplacian (TL) for feature preservation. A new metric based total Laplacian and L^1 norm is adapted for shape representation which can decouple feature from both Gaussian noise and outliers well without preprocessing. The main idea is to adaptively generate B-spline curve from initial shape under the new metric. Our initial shape is established by quad-tree construction. The choice of knot vector

and deBoor points has considerable effect on the shape of a curve. Different from most previous methods, our method treats the B-spline knot vector and deBoor points as variables simultaneously. An efficient numerical solver based on the augmented Lagrangian method is developed.

- Fourth, we present a variational approach to construct B-spline surfaces with sharp features from a set of data points. The approach finds the optimal placement of knots and control points simultaneously as opposed to most previous methods that determine the knots heuristically or in a separate step. Moreover, different from most previous methods using least squares metric, our approach adapts L^1 -norm with total variation (TV) as regularization in the fitting procedure, which enables the approach to handle both Gaussian noise and outliers in the same manner and is able to automatically optimize the placement of knot vector to faithfully reconstruct the sharp features. Experimental results demonstrate the effectiveness and efficiency of the proposed variational B-spline surface reconstruction.
- Finally, compared to B-spline surfaces that have topological restriction for surface representation, subdivision surfaces can handle shapes of arbitrary topology. We present a variational model for subdivision surface reconstruction for noisy dense mesh. In order to better represent the subdivision features, such as semi-sharp crease, crease, corner and so on, we design a new feature-aware subdivision scheme with continuous control of sharpness for each edge. According to the new subdivision rules, a variational model aimed to find the best control mesh and the edge sharpness is proposed. The variational model based on the L^1 norm can deal with noisy mesh with different types of noise. We also develop an iterative solver based on augmented Lagrangian methods to solve this nonlinear optimization problem.

1.4 Thesis Organization

The remainder of the thesis is organized as follows:

- Chapter 2 reviews the related works, which include variational methods, 3D mesh denoising, B-spline curve and surface reconstruction, and subdivision surface reconstruction.
- Chapter 3 introduce a general variational framework on manifolds and presents a variational structure-texture image decomposition on manifolds.
- Chapter 4 gives a mesh denoising method using the ROF variational model.
- Chapter 5 introduces a variational optimization methods for compact B-spline curve reconstruction from noisy point data.
- Chapter 6 presents a variational method for B-spline surface reconstruction with sharp features from noisy point cloud.
- Chapter 7 introduces a new feature-aware subdivision surface scheme and a variational subdivision surface reconstruction method based on the new scheme.
- Chapter 8 summarizes the thesis and points out some topics for future work.

Chapter 2

Literature Review

In this chapter, we review related work. First we introduce the variational methods and their applications in image processing. Then we review 3D mesh denoising, B-spline curve and surface reconstruction, and subdivision surface reconstruction.

2.1 Variational Methods

A variational method is one way of finding solutions to a problem by minimizing functionals. It formulates the objective problem as a minimiser of a functional. It has been well defined and studied in engineering. Here we take image processing as examples to review the variational methods, which shares a lot of similarity with our research work. The definition of a function involves the function domain, codomain and the mapping $u : \Omega \rightarrow \mathbb{R}^m$. In image processing, the definition for function u is simple. That is, $\forall (x, y) \in \Omega$ where $\Omega \subset \mathbb{R}^2$, $u(x, y)$ is the corresponding gray value or color value in $\mathbb{R}^m$, $m = 1$ or $m = 3$. Most of the problems like image restoration, decomposition, segmentation, reconstruction can be formulated as the minimization of functionals. Basically the formulation contains two terms:

$$\min_u F(u(x, y)) = E_r(u) + \lambda E_f(u) \quad (2.1)$$

The first term $E_r(u)$ is to keep intrinsic properties of the function u , such as piecewise smooth, piecewise constant, and sparse feature. It is usually called regularization term. The second term $E_f(u)$ is called the fidelity term, which is to make the function u satisfy some prior knowledge. For instance, for image restoration, the underlying image u should be close to the observed image u^0 .

Variational methods are increasingly being used in image processing to construct formal methods for accomplishing tasks that were performed heuristically in the past. Most of the image processing tasks like image segmentation, image denoising, image decomposition, can be formulated as minimization of functionals. People formulate the problems using mathematical frameworks that allow one to determine the “best” approach, rather than settling for “better” methods and algorithms.

In 1989 Mumford and Shah proposed a variational model (known as Mumford-Shah model) to perform classification [96]. The Mumford-Shah model is a classical model to optimally approximate a general function u^0 by piecewise smooth functions u . It has been extended to image segmentation [136] and multiclass labeling [109, 77] through its special multi-phase piece-wise constant form. The key merit of the Mumford-Shah model lies in the fact that it incorporates the similarity within one segment and the association between segments into one variational model.

Traditionally, denoising of signals has been carried out using linear/non linear filters, bilateral filters and weighted least square filters. The linear filters are simple, but are known to blur the sharp edges. The effect may be improved by using non-linear edge-preserving smoothing filters, examples of which are anisotropic diffusion [105], bilateral filters [133, 84] and weighed least square filters [47]. Filters are actually approximate solutions to some variational problems.

The variational approach for image denoising is first introduced in [112]. It is known as

the ROF model that minimizes the objective functional

$$F(u(x, y)) = \int_{\Omega} |\nabla u| dx dy + \frac{\lambda}{2} \int_{\Omega} (u - u^0)^2 dx dy \quad (2.2)$$

The first term is called the total variation (TV) which can preserve edges. The TV approach becomes a popular method for solving a wide variety of inverse problems in image processing due to the good edge-preserving property [21, 7, 101, 80, 159, 25].

Image decomposition is to decompose the piecewise constant or smooth underlying structure component u from oscillating patterns or noise. It is more general than image denoising. In [7], several energy functionals in the form of TV plus a fidelity term for image decomposition are analyzed. In particular, L^2 and G norms [95, 137] are suggested for textures of no *a-prior* knowledge or oscillating pattern and L^1 norm is suggested for structural texture.

In our research, we extend the above variational framework to geometry processing. Equation (2.1) can be formulated as follows:

$$\min_u \left\{ \int_{\Omega} \|\mathbf{g}(u)\|_{L_p} d\Omega + \lambda \int_{\Omega} \|\mathbf{h}(u)\|_{L_p} d\Omega \right\} \quad (2.3)$$

where $E_r(u) = \int_{\Omega} \|\mathbf{g}(u)\|_{L_p} d\Omega$, $E_f(u) = \int_{\Omega} \|\mathbf{h}(u)\|_{L_p} d\Omega$ are the regularization term and fidelity term, respectively. In this framework, the main tasks are to investigate how to represent the function domain Ω , how to define the functional for regularization and fidelity term according to different applications, and how to numerically solve the minimization problem. Some prior work on finding better regularization term, fidelity term and numerical solvers is reviewed below.

2.1.1 Regularization term

The regularization term is to smooth the function while maintaining the important feature such as piecewise smooth and piecewise constant.

Total variation: Total variation regularization was first introduced in [112]. It has been demonstrated very successful in image restoration and extensively generalized. The famous denoising model is the ROF model (2.2), which is able to preserve the edges of the original image well while removing most of the noise. A detailed mathematical study can be found in [94]. In order to show the good edge property of TV norm, we compare it with the following variational optimization problem with L^2 norm of gradient vector of an image.

$$\min_u \int_{\Omega} \|\nabla u\|^2 d\Omega + \frac{\lambda}{2} \int_{\Omega} (u - u^0)^2 d\Omega \quad (2.4)$$

The results are shown in Fig. 2.1. It is obvious that total variation preserves the features better.

Figure 2.1: Denoising results by (2.4) and (2.2).

Total curvature: Directly using the total variation of a function does not work well for geometry processing sometimes. Elsey et. al [46] propose a new variational model as a nature analogue of total variation, which is called total absolute Gaussian curvature:

$$\int_{\Omega} |k| d\Omega$$

where k is the Gaussian curvature.

Sparse signal processing: The reason that the total variation norm is better than L^2 norm can be essentially explained by the principles of sparse signal processing. Sparse

signal processing has had a significant impact in applied science and engineering such as statistics, information theory, computer vision, and others over the last few years. In sparse signal processing, people use L^1 regularization for the purpose of sparse signal reconstruction or feature selection. Assume a signal $u(t) \in \mathbb{R}^n$ and an over-complete basis functions $\phi_i \in \mathbb{R}^n, i = 1, \dots, m$, with $m > n$. Then the signal can be represented by

$$u = \sum_{i=0}^m \alpha_i \phi_i \quad (2.5)$$

In papers [19], [40], they show that when the signal is sparse, the signal can be reconstructed from a number of samples much lower than that required by Nyquist's theorem using L^1 minimization:

$$\min \|\alpha\|_{L^1} \quad \text{subject to} \quad u = \sum_{i=0}^m \alpha_i \phi_i \quad (2.6)$$

where $\alpha = (\alpha_1, \dots, \alpha_m)$.

The underpinning of TV models aims at exploiting the sparsity of the gradient of the image. Recently [8] first applied the sparse signal processing method to geometry processing, constructed a sparse signal on surface and successfully applied the L^1 sparsity paradigm to surface reconstruction.

In general, TV is a better approximation to L^1 minimization in sparse signal reconstruction than L^2 norm. Indeed, L^2 norm tends to severely penalize outliers and propagate the residual in the objective function uniformly. Hence, the solution is typically of a very poor degree of sparsity. Least squared methods by nature handle smooth functions better than non-smooth ones. In TV based methods, most of the “energy” of the residual in the objective function is expected to be concentrated near sharp features since they are not excessively penalized. In sparse signal processing, the TV norm is more suitable for sparse signal reconstruction and preserves sharp feature better.

2.1.2 Fidelity term

Similar to the regularization term, the fidelity term is also very important, especially when we have already known some prior knowledge.

L^2 norm: Most of the previous approaches adapt L^2 norm to make the unknown u function approximate the observed function u^0 , which is $\|u - u^0\|^2$. One advantage of using L^2 norm is that it is quadratic and its minimization problem can be converted into a linear system. L^2 norm is a good choice for most cases if the residual $u - u^0$ has a smooth or uniform distribution, like Gaussian noise.

L^1 norm: It is pointed out in [98] that the total variation regularization with L^1 fidelity norm performs better for certain types of noise. Many impressive numerical results are given in [99] that clearly demonstrate the advantages of using the L^1 norm for the fidelity term. Fig. 2.2 shows the effectiveness of the L^1 norm for removing impulsive noise. The L^1 norm model is defined as follows:

$$\min_u \int_{\Omega} |\nabla u| d\Omega + \lambda \int_{\Omega} |f - u| d\Omega \quad (2.7)$$

From the sparse signal reconstruction viewpoint, it also explains why the L^1 fidelity term is more suitable for some types of functions. If the functional itself is sparse, the L^1 fidelity norm works better in reconstruction. We adapt this good property of the L^1 norm in our variational framework for geometry processing.

Figure 2.2: Comparison of TV- L^2 and TV- L^1 for image denoising.

G norm: The pioneer work of the G norm was proposed by Meyer [95] for oscillatory patterns $v = f - u$. While the v component is viewed as an error term which is not analyzed further in most cases, it is not small in general when measured with L^2 norm. In the ROF model, both u and v contain high frequency components. The oscillating pattern will be removed from v in $L^2(\Omega)$ space. Hence, Meyer proposes a different decomposition model which is believed to be more adapted:

$$\min_{(u,v)/f=u+v} \left(\int_{\Omega} |\nabla u| d\Omega + \lambda \|v\|_G \right) \quad (2.8)$$

Meyer proved that the Banach space $G(\mathbb{R}^2)$ contains signals with large oscillations. The G norm can extract the oscillatory pattern detail better than L^2 norm. The definition of $G(\mathbb{R}^2)$ is given below.

Definition 1 $G(\mathbb{R}^2)$ is the Banach space composed of the distributions $v(x, y)$ which can be written

$$v(x, y) = \partial_x g_1(x, y) + \partial_y g_2(x, y) = \text{div}(g), g_1, g_2 \in L^\infty(\mathbb{R}^2).$$

Then $\|v\|_G = \inf \{ \|g\|_{L^\infty} \mid v = \text{div}(g), |g| = \sqrt{|g_1|^2 + |g_2|^2} \}$.

Meyer justified that a function belonging to G might have large oscillations and nevertheless have a small norm. Fig. 2.3 compares G norm with L^2 norm with the same fidelity parameter λ . We can see that G norm can better extract the oscillatory pattern than L^2 norm. However, due to the complexity of G norm, it is great challenging to solve the minimization in G norm. Recently, there is still no efficient method to solve this problem directly under G norm. Vese and Osher proposed an approach to approximately solve Meyer's problem [137]. Refer to (2.9). They replaced the G norm $\|v\|_G$ by $\|\sqrt{g_1^2 + g_2^2}\|_p$ which approximates the L^∞ norm when p goes to ∞ . For numerical reasons, the authors used the value $p = 1$, and they claimed that they did not see any visual difference when they used large values for p . Later, [102] proposed another approach to this problem when

$p = 2$.

$$\inf_{\{(u, g_1, g_2) | f = u + v\}} \left\{ \int_{\Omega} |\nabla u| d\Omega + \frac{\lambda}{2} \|f - u - \operatorname{div}(g_1, g_2)\|^2 + \kappa \left(\int_{\Omega} (\sqrt{g_1^2 + g_2^2})^p \right)^{\frac{1}{p}} \right\} \quad (2.9)$$

Figure 2.3: Decomposition of a gray image with oscillating pattern. (b) and (d) are the u components; (c) and (e) are the v components.

In [5] and [4], the authors realize this decomposition by minimizing a convex functional which depends on the two variables u and v , alternately in each variable. Each minimization is based on a projection algorithm to minimize the total variation. However, they also do not solve the G norm directly. They claim that the solution of their model is the solution of Meyer's model under some constraints.

2.1.3 Numerical solvers

The challenge of above variational optimization problems comes from the non-differentiability and non-linearity of the regularization terms and the fidelity terms. A lot of efforts have been contributed to design fast solvers [27, 24, 141, 57]. Some existing numerical methods include gradient descent method [26], LAD methods [51], primal-dual method [37], the splitting-and-penalty based method [156] and the alternating direction methods [141]. We should mention that in [141], the authors treat the operators in a compact way so that penalty parameters for different auxiliary variables are the same. It is more efficient to use different parameters for different auxiliary variables [149].

Augmented Lagrangian methods (ALM) are a class of algorithms for solving con-

strained optimization problems. They are known as a good alternative to penalty methods in that they replace a constrained optimization problem by a series of unconstrained problems which have an additional term similar to a Lagrange multiplier in the unconstrained objective. The unconstrained objective in the ALM is the Lagrangian of the constrained problem with an additional penalty term called the augmentation.

For generalization, assume the variational optimization of (2.3) can be read as follows:

$$\min_u \{ \mathbf{R}(\mathbf{g}(u)) + \mathbf{F}(\mathbf{h}(u)) \} \quad (2.10)$$

where $\mathbf{R}(\mathbf{g}(u))$, $\mathbf{F}(\mathbf{h}(u))$ are regularization and fidelity functionals. $\mathbf{R}()$ and $\mathbf{F}()$ could be non-differential or non-quadratic. In particular, if it is non-differential or non-quadratic, a new variable is introduced. Suppose both terms are non-differential. Then, the problem can be reformulated to the following constrained optimization problem

$$\begin{aligned} & \min_{u, \mathbf{p}, \mathbf{z}} \mathbf{R}(\mathbf{p}) + \mathbf{F}(\mathbf{z}) \\ & \text{subject to } \mathbf{p} = \mathbf{g}(u), \mathbf{z} = \mathbf{h}(u) \end{aligned} \quad (2.11)$$

To solve (2.11), the following augmented Lagrangian functional is defined

$$\begin{aligned} \mathcal{L}(u, \mathbf{p}, \mathbf{z}; \lambda_p, \lambda_z) = & \mathbf{R}(\mathbf{p}) + \mathbf{F}(\mathbf{z}) + \langle \lambda_p, \mathbf{p} - \mathbf{g}(u) \rangle + \langle \lambda_z, \mathbf{z} - \mathbf{h}(u) \rangle \\ & + \frac{r_p}{2} \|\mathbf{p} - \mathbf{g}(u)\|^2 + \frac{r_z}{2} \|\mathbf{z} - \mathbf{h}(u)\|^2 \end{aligned} \quad (2.12)$$

with Lagrange multipliers λ_p, λ_z and positive constants r_p, r_z .

As shown in [149], this can be converted into a saddle-point problem. In the augmented Lagrangian method, an iterative process is used to solve the saddle-point problem; see Algorithm 2.1.

To decouple the variables $u, \mathbf{p}, \mathbf{z}$ in (2.13), it separates the problem to several sub-problems and applies alternative minimization.

Some convergence analysis for the augmented Lagrangian method has been done in

Algorithm 2.1 Augmented Lagrangian method for variational problem (2.10)

Initialization: $\lambda_p^0 = 0, \lambda_z^0 = 0$;

For $k = 0, 1, 2, \dots$

1. Compute $(u^k, \mathbf{p}^k, \mathbf{z}^k)$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\lambda_p^{k+1}, \lambda_z^{k+1}$;

$$(u^k, \mathbf{p}^k, \mathbf{z}^k) \approx \arg \min_{u, \mathbf{p}, \mathbf{z}} \mathcal{L}(u, \mathbf{p}, \mathbf{z}; \lambda_p^{k+1}, \lambda_z^{k+1}) \quad (2.13)$$

2. Update

$$\begin{aligned} \lambda_p^{k+1} &= \lambda_p^k + r_p(\mathbf{p}^k - \mathbf{g}(u)) \\ \lambda_z^{k+1} &= \lambda_z^k + r_z(\mathbf{z}^k - \mathbf{h}(u)) \end{aligned}$$

[149]. Let the inner iteration of sub-problems in (2.13) be N . Two cases ($N = 1$) and ($N \rightarrow \infty$) are analyzed, where rough accuracy can be reached.

2.2 3D Mesh Denoising

Mesh denoising is a common process in geometry processing. Smoothing is also related to denoising. Roughly speaking, denoising can be regarded as a special type of smoothing where the small scales are random noise. The aim of smoothing is generally to remove certain high frequency information in the mesh including the small scale details of the mesh while mesh denoising focuses more on fidelity.

A wide variety of surface mesh denoising algorithms have been proposed during the past two decades. Many mesh denoising algorithms are developed by borrowing ideas from image processing. The state-of-the-art approaches to image denoising include signal processing methods [38], nonlinear anisotropic diffusion methods [105], bilateral filtering [133], and total variation based methods [112]. These approaches can be viewed in the framework of basis pursuit [28] for image processing. Most of these well posed image denoising methods have been extended to mesh denoising.

Taubin [131] pioneers signal processing on surfaces that is based on the Laplacian operators on meshes. It is a linear and isotropic technique and successfully analogizes

the concept of signal processing of image denoising to mesh denoising. This approach is extended to irregular mesh using a geometric flow analogy in [35]. Liu et al. [85] give a volume preservation smoothing. A few global, non-iterative approaches have also been developed recently [97, 83].

While isotropic methods do not preserve sharp features, feature-preserving anisotropic diffusion methods in image denoising are extended to anisotropic geometric diffusion on surfaces [30, 36, 9, 62]. Diffusion-based methods introduce an anisotropic tensor which will preserve or sharpen geometric features during denoising. The diffusion tensor is the same as the mean curvature flow method when the diffusion tensor is locally constant [30]. Recently, Ouafdi et al. [44] introduce a probabilistic method which averages of neighboring vertices weighted by the Riemannian distance according to a well designed diffusion tensor.

Fleishman et al. [49] and Jones et al. [65] extend the bilateral filtering for image denoising to mesh denoising. The key idea is to anisotropically average the nearby vertices weighted by a monotonously decreasing function in terms of both spatial difference and vertex difference. Some improvements have also been developed in [126, 163]. A fundamental relationship between diffusion and bilateral filtering is discussed in [10].

Since the face normals are to better representation for local surface geometry than vertex positions, filtering face normals instead of directly filtering vertex positions is also explored recently. Yogou et al. [154] propose to use the mean and median filters to estimate the face normals and later use alpha-trimming filters [155]. Shen et al. [121] develop a fuzzy vector median filter for mesh denoising. Sun et al. [126] improve upon this method by ignoring neighboring normals with large differences when computing face normals and propose a new vertex updating algorithm. A random walk model to determine averaging weights is then proposed in [127]. The bilateral filter is also applied to the facet normal field defined over the mesh by [163] which achieves impressive results.

Most recently, sparsity techniques are introduced in mesh smoothing. It is inspired by the promising development of the techniques of compressed sensing in recent years [20, 39, 45]. In recent years, compressed sensing has attracted considerable attention in areas of

applied mathematics, computer science, and signal processing. The central insight is that many signals are sparse and such signals can be recovered from very few measurements (undersampled data) by a nonlinear optimization. It shows that L^1 analysis approach has the good property for identification of sparse sharp features. Weighted L^1 analysis is used to decouple the noise and sharp features in [139]. The approach mainly contains two phases. The first phase is to estimate the base mesh by global Laplacian smoothing and the second phase is to recover the sharp features from the residual between the base mesh and the input mesh by weighted L^1 norm. Most of the previous smoothing approaches can not deal with high scale noise with random directions. He et al. [61] propose a L^0 minimization method for piecewise plane models with high scale noise which extends the work from image smoothing [153]. The main contribution of this work is that it designs a new edge operator which is sparse and L^0 norm can be efficiently adapted for noise removal.

2.3 B-spline Curve and Surface Reconstruction

Shape representation is a fundamental issue in graphics, computer aided design, pattern recognition, computer vision and image processing.

In 2D shape representation, the methods can be generally classified into two groups: contour-based and region-based methods [158]. A contour-based shape representation is described by its boundary, which is usually a set of dense points acquired from sensors or extracted from image and is thus frequently hampered with aliasing, occlusion, noise and outliers. Extracting meaningful information and features from the contour data has been widely used in shape analysis, matching [11], retrieval, object recognition and detection [18], [134]. For this purpose, a number of contour representations have been proposed including Fourier Description [106], chain code [113], polygon approximation [3, 42, 132, 150, 93, 22, 104], principle curve [2], and B-spline [16]. B-spline is one of the most efficient curve representations and has very attractive properties, such as compactness, smoothness and continuity. B-spline has been widely used in industry. B-spline curve and surface

representation offers great flexibility and precision for handling both analytic and modeled shapes [16].

Curve and surface reconstruction from a set of data point is a key problem for shape representation. Many methods have been developed for B-spline curve and surface reconstruction in the last decades. From the formulation of the B-spline curves and surfaces, parameterization of the input data points, the location of knot vector, control point positions are three important issues to solve during the reconstruction process. Most of the previous methods focus on one or two of the above three issues. An intrinsic parameterization problem is addressed in [64] and later improved in [116] by iterative parameter corrections. Ma et al. also propose a parameterization method to randomly measured points [89].

Most of these methods address the problem for smooth surfaces with uniform knots [17, 72, 140], where knot vector is prescribed and self-governed in the optimization. Obviously, the use of uniform knots has difficulty in faithfully recovering the salient sharp features of the data points, and may generate a B-spline surface with redundant knots in the flat area. Some adaptively inserting or removing knot methods [56, 79, 103] are proposed for surfaces with sharp features, such that knots can be attracted to the desirable region. However, these methods are usually restricted to fairly clean data points, since they are generally based on the idea of using curvature information extracted from points as the knot placement guidance. The recent work [152] proposes an iterative optimization scheme called HOAAI, which addresses the challenging problems of automatic determination on knot placement. However, this method just heuristically adjusts knot positions iteratively according to the parameterization update within a small range. In general, the above methods may not be used to reliably reconstruct data points with sharp features, like cusps, corners, etc., where multiple knots are actually needed.

Another category of methods considers how to measure the geometric distance between the fitting curve or surface and a target shape. Different from point distance error and tangent distance error [14], Wang et al. [140] introduce a new fitting error called squared distance error term, defined by a curvature-based quadratic approximate of squared

distances from data points to target shape. This algorithm is improved by [162] recently.

Note that a B-spline surface is affected by its knot vector. Freeing the knot vector would definitely increase data fitting flexibility, but meanwhile it makes the problem complicated since it results in a nonlinear problem. Recently, some heuristic techniques that convert the nonlinear optimization into a discrete combinatorial optimization problem have been devised. Evolutionary intelligence approaches are developed in [144, 138, 115] for surface reconstruction. However, these methods do not provide sufficient accuracy and are limited to simple models. A simulated annealing based method is used for global optimization [111]. Zhao et al. [161] introduce an adaptive knot placement approach based on estimation of distribution algorithms, which is a stochastic optimization scheme. Genetic algorithms, usually converting the problem with many local optima into a discrete combinatorial optimization problem, are also adopted for surface reconstruction [54, 114, 157]. Partial swarm optimization methods are applied to compute the appropriate location of knots automatically [52, 53]. However, these methods require high computation cost. To speed up the computation, [151] converts a B-spline surface to Bézier patches for efficient gradient computation, but such conversion attenuates the shape flexibility in the reconstruction.

2.4 Subdivision Surface Reconstruction

Subdivision surfaces are currently one of the most powerful surface representation used to model smooth shapes. Compared to regular surface splines such as NURBS, subdivision surface can handle shapes of arbitrary topology in a unified framework. Also, unlike polygonal meshes, subdivision surfaces generate smooth surfaces, which is important in designing aesthetically pleasing shapes.

The first two subdivision surfaces were first introduced by Catmull and Clark [23] and Doo and Sabin [41] for arbitrary initial meshes. They both generalized tensor product B-splines of bi-degree three and two, respectively, to arbitrary topologies by extending the refinement rules to irregular parts of the control mesh. Loop's subdivision surface was

defined by Charles Loop [86] for triangular meshes. Some other subdivision rules were also proposed, such as quad/triangle subdivision scheme [125], mid-edge subdivision scheme [107, 60], and $\sqrt{3}$ subdivision scheme [69]. Stam presented a method for exact evaluation of Catmull-Clark and Loop subdivision surfaces under arbitrary parameter values [123]. Non-uniform Catmull-Clark and Doo-Sabin surfaces that generalize non-uniform tensor product B-spline surface to arbitrary topologies were proposed in [120].

Subdivision surface reconstruction has been investigated. Most methods use smooth subdivision rules for the surface fitting processing, which can not however capture the sharp feature. Lee et. al. [76] proposed a new surface representation, the displaced subdivision surface, which uses subdivision surface as a smooth domain and the displacement function is defined over the domain for detail. To best reconstruct the surface feature, the method in [128] starts from an interactively defined initial control mesh and adaptively recovers the features by inserting new vertices. A software “MeshToSS” is introduced in [66] to convert dense triangular mesh to progressive subdivision surfaces. Litke et al. [82] propose a new subdivision surface fitting algorithm with a prescribed tolerance based on the method of “quasi-interpolation”. Some methods attempt to improve foot point finding methods and distance computation methods near features. Different error terms are applied to improve the efficiency and the accuracy, such as point distance error function, tangent distance error function and squared distance error function. A subdivision surface fitting method based on parameter correction is developed in [91], and it uses the combination of point error and tangent error for better efficiency. Squared distance minimization is applied for subdivision fitting in [29], which considers the geometric error based on the curvature to better measure the distance from the subdivision surface to the target.

The other category of methods is based on the special rules for sharp features. All feature edges and vertices of the original mesh model are first identified. Fitting subdivision surfaces with sharp features was first considered in [63]. A new set of subdivision rules to accurately model objects with tangent discontinuities are proposed. In [90], a new and direct approach for fitting a subdivision surface from an irregular and dense triangle mesh of

arbitrary topological type is proposed. A new exact evaluation method for all types of sharp features is proposed in [81]. It expresses the closest point of an input mesh vertex on the limit surface as the linear combination of the control vertices through the basis functions of the subdivision surface. Following [63], DeRose et al. generalize the infinity sharp crease to semi-sharp crease by setting an integer to control the sharpness for Catmull-Clark subdivision surfaces [34]. [75] develops a semi-sharp subdivision surface fitting approach based on feature lines approximation. The semi-sharp features are achieved by relaxing according to the curvature of the target surface. However, all the above sharp feature recovery methods need to detect the sharp features first. The accuracy of feature detection will highly affect the sharp feature of the final reconstructed subdivision surface.

Chapter 3

Structure-Texture Image Decomposition on Manifolds

This chapter presents a variational framework for decomposing color attributes of 3D triangular mesh models. The problem can also be viewed as image decomposition on manifolds.

3.1 Motivation and Our Work

As one form of the task of decomposing signals into simpler signals of more coherent origin, the decomposition of an image into its structural and textural components is an important inverse problem in image processing. In general, the structural component is piecewise smooth, corresponding to the main large objects in the image, and the textural component captures the fine scale-details in the image, usually with some periodicity or oscillatory nature. When the textural component is noise, the decomposition process becomes image restoration or denoising. However, the decomposition is not just a low-pass/high-pass image denoising since both the structural and textural components may contain high and low frequencies [95, 137]. It has been shown that this decomposition is useful for HDR tone mapping [135, 47], flash/no-flash image fusion [108, 43] and image editing [100], and

fundamental for image inpainting [12], image matching, compression, segmentation and classification [6].

Manifold surfaces are common in computer graphics and vision. Especially with advance of 3D data acquisition technology, the captured data often contain both 3D geometry and color information [129]. The color information provides additional attributes to geometry and affects the appearance of the shape. In this chapter, we are interested in the problem of processing such color attributes, in particular, decomposing them into structural component and the textural component. This process is useful for further editing of color attributes on surfaces. On the other hand, this process can be considered as an extension of conventional image processing to surfaces.

In late 1990s, a general geometrical framework for low level vision was proposed in [122, 68]. This framework solves many existing problems in denoising, segmentation and scale space procedures. They treat images as manifolds in feature space and adapt a Riemannian geometry tool to solve traditional image processing problems. In our work, we are mainly interested in studying image processing defined on manifolds. There has been increasing interest in studying image processing on surfaces or manifolds in the fields of computer vision, medical imaging, urban reconstruction and 3D cartoon. Some image processes such as image segmentation and restoration have been extended to manifolds [33, 149, 148]. However, the problem of decomposing data on manifolds has not received much attention. Moreover, many image processes on surfaces are considered on implicit representations, which define a surface as a zero level set of a signed distance function on an Euclidean domain or a narrow band of the given surface. The differential operators on surfaces are approximated by combining the standard Euclidean differential operators with the projection along the normal direction [13, 70]. In practice, it is not easy to obtain the implicit representation of a surface with complicated geometry or topology structure.

Triangular meshes are a popular representation in many applications. To facilitate data processing on triangular meshes, one approach is to parameterize the mesh first and

then differential operators can be computed under the corresponding parametrization [124, 58, 88]. However, the computation of parametrization itself is a complicated task, especially for a surface with complicated structure or high genus. Recently, the image processing problems are formulated directly on meshes [145, 146, 33, 71, 148]. Fast convex optimization techniques for total variation (TV) based methods are discussed in [71], where the split Bregman iteration is used for TV denoising of gray images and Chan-Vase segmentation.

In this chapter, we adapt the multichannel variational image decomposition models to manifolds represented by triangular meshes. Different fidelity terms, L^2 , L^1 and G norms, are chosen according to the types of texture of the input image. L^2 norm is designed for extracting low-amplitude texture including Gaussian noise. L^1 norm preserves contrast and geometric structure and is thus suitable for structural or sparse textures such as impulsive noise. G norm is introduced for the texture of oscillatory pattern. The challenging problems of L^1 and G norms for multichannel images are extended to triangular meshes for the first time. We discretize the models directly on triangular meshes, which avoids parameterization but takes the intrinsic geometry of the mesh into account. The final formulations are non-linear and non-differentiable. Thus we develop three augmented Lagrangian methods based algorithms to solve the proposed variational decomposition models. The algorithms work by iterating several procedures of solving some sub-problems. Since the sub-problems either have closed form solutions or are to solve linear systems, our numerical methods are very simple and effective.

3.2 Image Decomposition Models on Manifolds

This section describes three multichannel variational models with L^2 , L^1 and G norms for image decomposition on manifolds and their discretization when the manifold is represented by a triangular mesh.

3.2.1 Three variational models

Let M be a 2-dimensional Riemannian manifold and $\{U_\alpha, x_\alpha\}_{\alpha \in \mathcal{A}}$ be its differential structure where $\mathcal{A}$ is an index set. $\{U_\alpha, x_\alpha\}_{\alpha \in \mathcal{A}}$ can be viewed as a piecewise parameterization of M . We denote by (ξ^1, ξ^2) the local coordinates U_α . For any point $x \in x_\alpha(U_\alpha) \subset M$, the tangent space $T_x M$ is spanned by $\{\frac{\partial x_\alpha}{\partial \xi^1}, \frac{\partial x_\alpha}{\partial \xi^2}\}$. The differential structure $\{U_\alpha, x_\alpha\}_{\alpha \in \mathcal{A}}$ gives a Riemannian metric tensor $\mathbb{G}$ on M :

$$\mathbb{G} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_\alpha}{\partial \xi^1} \cdot \frac{\partial x_\alpha}{\partial \xi^1} & \frac{\partial x_\alpha}{\partial \xi^1} \cdot \frac{\partial x_\alpha}{\partial \xi^2} \\ \frac{\partial x_\alpha}{\partial \xi^2} \cdot \frac{\partial x_\alpha}{\partial \xi^1} & \frac{\partial x_\alpha}{\partial \xi^2} \cdot \frac{\partial x_\alpha}{\partial \xi^2} \end{pmatrix} \quad (3.1)$$

where g_{11}, g_{12} (or g_{21}) and g_{22} are coefficients of the first fundamental form. The tensor $\mathbb{G}$ is symmetric and positive definite. It determines a bilinear form on the tangent space $T_x M$. One can calculate the inner product of any two vectors on $T_x M$ using the bilinear form.

An image defined on manifold M is a mapping $\mathbf{f} : M \rightarrow \mathbb{R}^m$. When $m = 1$, it is a gray image; and when $m = 3$, it is a color image. The structure-texture image decomposition on a manifold is to decompose $\mathbf{f}$ into $\mathbf{u}$ and $\mathbf{v}$ such that

$$\mathbf{f} = \mathbf{u} + \mathbf{v} \quad (3.2)$$

where $\mathbf{u} = (u^1, u^2, \dots, u^m)^T$ is a function defined on M , which is usually piecewise constant and conveys the geometric information; and $\mathbf{v} = (v^1, v^2, \dots, v^m)^T$ is another function defined on M , encoding textural information. One way to implement such decomposition is to find $\mathbf{u}$ and $\mathbf{v}$ satisfying

$$\inf_{\{(\mathbf{u}, \mathbf{v}) | \mathbf{f} = \mathbf{u} + \mathbf{v}\}} \int_M |\nabla_M \mathbf{u}| dM + \lambda \|\mathbf{v}\|_{fid} \quad (3.3)$$

where the energy functional is a combination of a regularization term and a fidelity term with a tuning parameter $\lambda \in \mathbb{R}$ balancing the two terms.

The regularization term $\int_M |\nabla_M \mathbf{u}| dM$ is a total variation of $\mathbf{u}$ on M where dM is the

manifold's element measure, ∇_M is the intrinsic gradient on M , defined by

$$\nabla_M \mathbf{u} = \left(\frac{\partial x_\alpha}{\partial \xi^1}, \frac{\partial x_\alpha}{\partial \xi^2} \right) \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial \mathbf{u}}{\partial \xi^1} \\ \frac{\partial \mathbf{u}}{\partial \xi^2} \end{pmatrix} \quad (3.4)$$

with $\mathbf{u}(x) = \mathbf{u}(x_\alpha(\xi^1, \xi^2))$, and then

$$|\nabla_M \mathbf{u}| = \sqrt{\left(\frac{\partial \mathbf{u}}{\partial \xi^1}, \frac{\partial \mathbf{u}}{\partial \xi^2} \right) \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial \mathbf{u}}{\partial \xi^1} \\ \frac{\partial \mathbf{u}}{\partial \xi^2} \end{pmatrix}} \quad (3.5)$$

The total variation regularization enables good edge preservation.

The fidelity term $\|\mathbf{v}\|_{fid}$ is formulated by L^2 , L^1 and G norms on manifold M to fit three types of textures: low-amplitude texture, sparse structure texture (such as pepper and salt noise), and oscillating patterns, respectively. The details of the three norms are elaborated below.

- The L^2 -norm variational image decomposition on manifold M is to find the solution of

$$\inf_{\{(\mathbf{u}, \mathbf{v}) | \mathbf{f} = \mathbf{u} + \mathbf{v}\}} \left(\int_M |\nabla_M \mathbf{u}| dM + \lambda \|\mathbf{v}\|_{2,M}^2 \right) \quad (3.6)$$

where $\|\mathbf{v}\|_{2,M}^2 = \int_M \langle \mathbf{v}, \mathbf{v} \rangle dM$ with $\langle, \rangle$ denoting the inner product in Euclidean space, $\mathbb{R}^m$. This is an extension of the famous ROF model [112] from a plane domain to a manifold. The ROF model was proposed initially for denoising, which preserves well the edges of the input image and removing most of the noise.

- The L^1 -norm variational image decomposition on manifold M is to find the solution of

$$\inf_{\{(\mathbf{u}, \mathbf{v}) | \mathbf{f} = \mathbf{u} + \mathbf{v}\}} \left(\int_M |\nabla_M \mathbf{u}| dM + \lambda \|\mathbf{v}\|_{1,M} \right) \quad (3.7)$$

where $\|\mathbf{v}\|_{1,M} = \int_M \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} dM$. Compared to the ROF model, the L^1 -norm model does not erode geometric structure and is particularly well suited to remove salt and

pepper noise. However, L^1 -norm is highly non-linear and non-differentiable, which makes it difficult to solve the problem quickly and efficiently.

- The G -norm variational image decomposition on manifold M is to find the solution of

$$\inf_{\{(\mathbf{u}, \mathbf{v}) | \mathbf{f} = \mathbf{u} + \mathbf{v}\}} \left(\int_M |\nabla_M \mathbf{u}| dM + \lambda \|\mathbf{v}\|_{G,M} \right) \quad (3.8)$$

where $\|\mathbf{v}\|_{G,M}$ is the G -norm on manifold M . The G -norm was initially suggested by Meyer for oscillations [95]. Here we adapt it on manifolds. Let $\mathbf{v}$ be written as

$$\begin{aligned} \mathbf{v} = \text{div}_M(\mathbf{g}) = & \frac{1}{\sqrt{\det(\mathbb{G})}} \frac{\partial}{\partial \xi^1} \left(\sqrt{\det(\mathbb{G})} g^1 \right) \\ & + \frac{1}{\sqrt{\det(\mathbb{G})}} \frac{\partial}{\partial \xi^2} \left(\sqrt{\det(\mathbb{G})} g^2 \right) \end{aligned} \quad (3.9)$$

for a vector $\mathbf{g} = g^1 \frac{\partial x_\alpha}{\partial \xi^1} + g^2 \frac{\partial x_\alpha}{\partial \xi^2}$. Then

$$\|\mathbf{v}\|_{G,M} = \inf_{\mathbf{g}} \left\{ \sup_{x \in M} |\mathbf{g}| : \begin{aligned} & \mathbf{v} = \text{div}_M(\mathbf{g}), \\ & \mathbf{g} = g^1 \frac{\partial x_\alpha}{\partial \xi^1} + g^2 \frac{\partial x_\alpha}{\partial \xi^2} \end{aligned} \right\} \quad (3.10)$$

with

$$|\mathbf{g}| = \sqrt{(g^1(x), g^2(x)) \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} g^1(x) \\ g^2(x) \end{pmatrix}}.$$

Such $\mathbf{v}$ may have large oscillations with nevertheless a small norm. Thus the G norm is well-suited to capture the oscillations of a function in an energy minimization method.

However, a G -norm minimization problem is very difficult to solve. Note that there was no numerical scheme provided in [95]. Vese and Osher [137] devised a variant of the G -norm decomposition based on L^p norms and gave a numerical scheme using Euler-Lagrange equations. The variant of the G -norm decomposition on manifold M

can then be

$$\inf_{\mathbf{f}=\mathbf{u}+\operatorname{div}_M(\mathbf{g})} \left(\int_M |\nabla_M \mathbf{u}| dM + \kappa \left[\int_M |\mathbf{g}|^p dM \right]^{\frac{1}{p}} + \lambda \|\mathbf{f} - \mathbf{u} - \operatorname{div}_M(\mathbf{g})\|_{2,M}^2 \right) \quad (3.11)$$

In practice, it is often to choose $p = 1$. In this work, we only consider the situation of $p = 1$, which is an approximation of the G -norm decomposition.

3.2.2 Discretization

Now we consider a manifold represented by a triangular mesh. Triangular meshes are a common shape representation in geometric modeling and computer graphics. Without ambiguity, we also use M to denote a triangular mesh surface of arbitrary topology in R^3 with no degenerate triangles. The set of vertices and the set of face triangles of M are denoted by $\{v_i : i = 0, 1, \dots, N_v - 1\}$, $\{\tau_i : i = 0, 1, \dots, N_\tau - 1\}$, respectively, where N_v, N_τ are the numbers of vertices and triangles.

As in the finite elements literature, for each vertex v_i , we define a piecewise linear basis functions $\phi_i : M \rightarrow \mathbb{R}$ such that $\phi_i(v_j) = \delta_{ij}$, $i, j = 0, 1, \dots, N_v - 1$, where δ_{ij} is the Kronecker delta. The vector valued field $\mathbf{u} = \{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^m\}$ with $\mathbf{u}^i = (u_0^i, u_1^i, \dots, u_{N_v-1}^i)$ is defined on all vertices of the triangular mesh M . Suppose $\mathbf{u}$ reaches values $\mathbf{u}_j = (u_j^1, u_j^2, \dots, u_j^m)$ at vertex $v_j, j = 0, 1, \dots, N_v - 1$. We can use $\{\phi_i : i = 0, 1, \dots, N_v - 1\}$ to construct a piecewise linear interpolating function on M , which naturally extends $\mathbf{u}$ on M . That is, $\mathbf{u}(x) = \sum_{i=0}^{N_v-1} \mathbf{u}_i \phi_i(x)$ for any $x \in M$.

We define two Hilbert spaces $\mathbf{V}_M^m = \underbrace{V_M \times V_M \times \dots \times V_M}_m$ with $V_M = \mathbb{R}^{N_v}$ and $\mathbf{Q}_M^m = \underbrace{Q_M \times Q_M \times \dots \times Q_M}_m$ with $Q_M = \mathbb{R}^{3 \times N_\tau}$. The elements in the linear space $\mathbf{V}_M^m$ are given by values at vertices of M . For example, $\mathbf{u} = (\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^m) \in \mathbf{V}_M^m$ with $\mathbf{u}^i = (u_0^i, u_1^i, \dots, u_{N_v-1}^i) \in V_M$ means that $\mathbf{u}$ takes value $\mathbf{u}_j = (u_j^1, u_j^2, \dots, u_j^m)$ at vertex v_j for j . By the basis functions $\{\phi_i : i = 0, 1, \dots, N_v - 1\}$, $\mathbf{V}_M^m$ is isomorphic to the space of all piecewise linear vector-valued functions on M . We denote by Q_M the set of piecewise constant vector fields on M . The linear space $\mathbf{Q}_M^m$ is the product of m Q_M 's. Thus an m -

channel image $\mathbf{f} = (f^1, f^2, \dots, f^m)$ on M , where $f^i = (f_0^i, f_1^i, \dots, f_{N_v-1}^i)$, is an element of $\mathbf{V}_M^m$, and its intrinsic gradient $\nabla_M \mathbf{f} = (\nabla_M f^1, \nabla_M f^2, \dots, \nabla_M f^m)$ is an element of $\mathbf{Q}_M^m$.

We can equip the spaces $\mathbf{V}_M^m$ and $\mathbf{Q}_M^m$ with inner products and norms. For $\mathbf{u}, \mathbf{v} \in \mathbf{V}_M^m$, we define the inner product of $\mathbf{u}$ and $\mathbf{v}$ to be

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{V}_M^m} = \sum_{j=0}^{N_v-1} s_j \sum_{i=1}^m u_j^i v_j^i \quad (3.12)$$

where $s_j = \sum_{\tau \in D_1(j)} \frac{1}{3} s_\tau$ with s_τ as the area of the triangle τ and $D_1(j)$ is the set of triangles having v_j as one of their vertices. The L^2 and L^1 -norms of an element $\mathbf{u} \in \mathbf{V}_M^m$ are

$$\|\mathbf{u}\|_{2, \mathbf{V}_M^m}^2 = \langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{V}_M^m} \quad (3.13)$$

and

$$\|\mathbf{u}\|_{1, \mathbf{V}_M^m} = \sum_{j=0}^{N_v-1} s_j \sqrt{\sum_{i=1}^m (u_j^i)^2}, \quad (3.14)$$

respectively. Similarly, for $\mathbf{p}, \mathbf{q} \in \mathbf{Q}_M^m$, we define

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbf{Q}_M^m} = \sum_{\tau} s_{\tau} \sum_{i=1}^m \langle \mathbf{p}_{\tau}^i, \mathbf{q}_{\tau}^i \rangle \quad (3.15)$$

where $\langle \mathbf{p}_{\tau}^i, \mathbf{q}_{\tau}^i \rangle$ is the conventional inner product in $\mathbb{R}^3$. The L^2 and L^1 -norms of an element $\mathbf{p} \in \mathbf{Q}_M^m$ are

$$\|\mathbf{p}\|_{2, \mathbf{Q}_M^m}^2 = \langle \mathbf{p}, \mathbf{p} \rangle_{\mathbf{Q}_M^m} \quad (3.16)$$

and

$$\|\mathbf{p}\|_{1, \mathbf{Q}_M^m} = \sum_{\tau} s_{\tau} \sqrt{\sum_{i=1}^m \langle \mathbf{p}_{\tau}^i, \mathbf{p}_{\tau}^i \rangle}, \quad (3.17)$$

respectively.

The gradient operator ∇_M on M can also be viewed as a mapping from $\mathbf{V}_M$ to $\mathbf{Q}_M$ or from $\mathbf{V}_M^m$ to $\mathbf{Q}_M^m$. For instance, given $\mathbf{u} \in \mathbf{V}_M^m$, corresponding to a piecewise linear

function on M , its gradient is

$$\nabla_M \mathbf{u} = \sum_{i=0}^{N_v-1} \mathbf{u}_i \nabla_M \phi_i \in \mathbf{Q}_M^m \quad (3.18)$$

and $\nabla_M \phi_i$ is a piecewise constant vector field on M . Referring to Fig. 3.1, we introduce the local parametrization $\xi_{j,\tau} = (\xi_\tau^1, \xi_\tau^2) \in \mathbb{R}^2$ for each triangle τ on manifold M with respect to vertex v_j such that for any point $v \in \tau$, we have $v = v_i + \xi_\tau^1(v_k - v_i) + \xi_\tau^2(v_j - O)$ where O is the project of v_j on edge $v_i v_k$. Thus the tensor $\mathbb{G}$ at point v is

$$\mathbb{G} = \begin{pmatrix} (v_k - v_i) \cdot (v_k - v_i) & 0 \\ 0 & (v_j - O) \cdot (v_j - O) \end{pmatrix}$$

and then

$$\nabla_M \phi_j = \frac{v_j - O}{\|v_j - O\|^2}. \quad (3.19)$$

Figure 3.1: Local parameterization of a triangle.

The divergence operator $\text{div}_M : \mathbf{Q}_M^m \rightarrow \mathbf{V}_M^m$, for $\mathbf{p} \in \mathbf{Q}_M^m$, is given by

$$(\text{div}_M \mathbf{p})_{v_i} = -\frac{1}{s_i} \sum_{\tau, v_i \prec \tau} s_\tau \langle \mathbf{p}_\tau, (\nabla_M \phi)_\tau \rangle$$

where $v_i \prec \tau$ means that v_i is a vertex of triangle τ .

With all the above formulae, we can write the discrete versions of our L^2 , L^1 and G -

norm variational image decomposition models (3.6,3.7,3.11) as follows:

$$\min_{\mathbf{f}=\mathbf{u}+\mathbf{v}} \left(\|\nabla_M \mathbf{u}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{v}\|_{2, \mathbf{V}_M^m}^2 \right), \quad (3.20)$$

$$\min_{\mathbf{f}=\mathbf{u}+\mathbf{v}} \left(\|\nabla_M \mathbf{u}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{v}\|_{1, \mathbf{V}_M^m} \right), \quad (3.21)$$

$$\begin{aligned} \min_{\mathbf{f}=\mathbf{u}+\text{div}_M(\mathbf{g})} & \left(\|\nabla_M \mathbf{u}\|_{1, \mathbf{Q}_M^m} + \kappa \|\mathbf{g}\|_{1, \mathbf{Q}_M^m} \right. \\ & \left. + \lambda \|\mathbf{f} - \mathbf{u} - \text{div}_M(\mathbf{g})\|_{2, \mathbf{V}_M^m}^2 \right). \end{aligned} \quad (3.22)$$

3.3 Numerical Methods

The challenge of our variational optimization problems comes from the non-differentiability and non-linearity of the total variation term and the L^1 and G fidelity terms. These models are often solved by gradient descent methods such as isotropic and anisotropic diffusion approaches, which are usually quite slow. Designing fast solvers is an active research topic in numerical computation. Recently, various convex optimization techniques have been proposed to efficiently solve TV based optimization problems. In this section we adapt augmented Lagrangian methods [130, 149] to solve variational image decomposition on a triangulated surface.

3.3.1 TV- L^2

We introduce a new variable $\mathbf{p} \in \mathbf{Q}_M^m$ and reformulate the TV- L^2 problem of (3.20) to the following constrained minimization problem

$$\begin{aligned} \min_{\mathbf{u} \in \mathbf{V}_M^m, \mathbf{p} \in \mathbf{Q}_M^m} & \left\{ \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{f} - \mathbf{u}\|_{2, \mathbf{V}_M^m}^2 \right\} \\ \text{s.t.} \quad & \mathbf{p} = \nabla_M \mathbf{u} \end{aligned} \quad (3.23)$$

To solve this problem, we define the augmented Lagrangian functional

$$\begin{aligned} L(\mathbf{u}, \mathbf{p}; \mu) &= \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{f} - \mathbf{u}\|_{2, \mathbf{V}_M^m}^2 \\ &\quad + \langle \mu, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \end{aligned} \quad (3.24)$$

where $\mu \in \mathbf{Q}_M^m$ is the Lagrangian multiplier, r is a positive constant and $\frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, M}^2$ is the augmentation, and then consider the following saddle-point problem: Find $(\mathbf{u}^*, \mathbf{p}^*; \mu^*) \in \mathbf{V}_M^m \times \mathbf{Q}_M^m \times \mathbf{Q}_M^m$ such that

$$L(\mathbf{u}^*, \mathbf{p}^*; \mu) \leq L(\mathbf{u}^*, \mathbf{p}^*; \mu^*) \leq L(\mathbf{u}, \mathbf{p}; \mu^*)$$

for all $(\mathbf{u}, \mathbf{p}; \mu) \in \mathbf{V}_M^m \times \mathbf{Q}_M^m \times \mathbf{Q}_M^m$.

Similar to [148] and references therein, it can be shown that this saddle-point problem has at least one solution and all the saddle-points $(\mathbf{u}^*, \mathbf{p}^*; \mu^*)$'s have the same $\mathbf{u}^*$, which is the unique solution to the original problem (3.20). Based on the augmented Lagrangian method, we present an iterative algorithm to solve the saddle-point problem, which is described in Algorithm 3.1.

In Algorithm 3.1, we compute an approximate solution to the minimization problem (3.25) by iteratively solving two sub-problems: the $\mathbf{u}$ -sub problem and the $\mathbf{p}$ -sub problem. The number of inner iterations increases the accuracy of (3.25) and ensures the convergence of overall algorithm. When in image case, it is fully accuracy ($K \rightarrow \infty$) and rough accuracy ($K = 1$). Some convergence analysis is given in [147]. As observed in [148] and also our experiments, a small value suffices. In this work, we choose $K = 1$. In addition, the two sub-problems can be easily solved.

- The $\mathbf{u}$ -sub problem: Given $\mathbf{p}$, solve

$$\min_{\mathbf{u}} \{ \lambda \|\mathbf{f} - \mathbf{u}\|_{2, \mathbf{V}_M^m}^2 + \langle \mu^{(k)}, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \} \quad (3.26)$$

Algorithm 3.1 ALM algorithm for TV- L^2 decomposition

Initialization: $\mathbf{u}^{(0)} = \mathbf{p}^{(0)} = \mu^{(0)} = \epsilon = 0$.

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multiplier $\mu^{(k)}$:

$$(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}) \approx \arg \min_{\mathbf{u}, \mathbf{p}} L(\mathbf{u}, \mathbf{p}, \mu^{(k)}) \quad (3.25)$$

 which is done by the following iterations:

 Let $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}$.

for $l = 1, 2, \dots, K$ **do**

 compute $\mathbf{u}^{(k+1)} = \arg \min_{\mathbf{u}} L(\mathbf{u}, \mathbf{p}^{(k+1)}; \mu^{(k)});$

 compute $\mathbf{p}^{(k+1)} = \arg \min_{\mathbf{p}} L(\mathbf{u}^{(k+1)}, \mathbf{p}; \mu^{(k)});$

end for

 and then update

$$\mu^{(k+1)} = \mu^{(k)} + r(\mathbf{p}^{(k+1)} - \nabla_M \mathbf{u}^{(k+1)})$$

$\epsilon = \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{2, V_M^m}, k++.$

end while

This is a quadratic programming problem, whose optimality condition is

$$2\lambda(\mathbf{f} - \mathbf{u}) + \text{div}_M(\mu^{(k)}) + r \text{div}_M(\mathbf{p} - \nabla_M \mathbf{u}) = 0,$$

which turns to be a sparse linear problem and can be solved by various well-developed numerical packages.

- The $\mathbf{p}$ -sub problem: Given $\mathbf{u}$, solve

$$\min_{\mathbf{p}} \{ \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \langle \mu^{(k)}, \mathbf{p} \rangle_{\mathbf{Q}_M^m} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \}$$

This problem is decomposable and thus can be solved triangle-by-triangle. That is, for each triangle τ , we solve

$$\min_{\mathbf{p}_\tau} \left\{ \sqrt{\sum_{i=1}^m \langle \mathbf{p}_\tau^i, \mathbf{p}_\tau^i \rangle} + \sum_{i=1}^m \langle \mu_\tau^i, \mathbf{p}_\tau^i \rangle_{\mathbf{Q}_M^m} + \frac{r}{2} \sum_{i=1}^m \langle \mathbf{p}_\tau^i - (\nabla_M \mathbf{u}^i)_\tau, \mathbf{p}_\tau^i - (\nabla_M \mathbf{u}^i)_\tau \rangle \right\}$$

By a simple geometric analysis, it can be found that the above problem has a closed form solution

$$\mathbf{p}_\tau = \max(0, 1 - \frac{1}{r|\mathbf{w}_\tau|})\mathbf{w}_\tau \quad (3.27)$$

$$\text{where } \mathbf{w}_\tau = (\nabla_M \mathbf{u})_\tau - \frac{\mu_\tau^{(k)}}{r}.$$

3.3.2 TV- L^1

When the fidelity term is in L^1 , we introduce two new variables $\mathbf{p} \in \mathbf{Q}_M^m$, $\mathbf{z} \in \mathbf{V}_M^m$, and reformulate the TV- L^1 problem to the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{p}, \mathbf{z}} \{ & \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{z}\|_{1, \mathbf{V}_M^m} \} \\ \text{s.t. } & \mathbf{p} = \nabla_M \mathbf{u}, \mathbf{z} = \mathbf{u} - \mathbf{f} \end{aligned} \quad (3.28)$$

We then define the following augmented Lagrangian functional

$$\begin{aligned} L(\mathbf{u}, \mathbf{p}, \mathbf{z}; \mu_p, \mu_z) = & \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{z}\|_{1, \mathbf{V}_M^m} + \langle \mu_p, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \langle \mu_z, \mathbf{z} - (\mathbf{u} - \mathbf{f}) \rangle_{\mathbf{V}_M^m} \\ & + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 + \frac{r_z}{2} \|\mathbf{z} - (\mathbf{u} - \mathbf{f})\|_{2, \mathbf{V}_M^m}^2 \end{aligned}$$

where $\mu_p \in \mathbf{Q}_M^m$, $\mu_z \in \mathbf{V}_M^m$ are the Lagrangian multipliers, and r_p and r_z are positive numbers, and $\frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2$ and $\frac{r_z}{2} \|\mathbf{z} - (\mathbf{u} - \mathbf{f})\|_{2, \mathbf{V}_M^m}^2$ are the augmentations.

As in Section 3.3.1, we solve (3.28) by adapting the augmented Lagrangian method to find a solution to the saddle-point problem:

$$\min_{\mathbf{u}, \mathbf{p}, \mathbf{z}} \max_{\mu_p, \mu_z} L(\mathbf{u}, \mathbf{p}, \mathbf{z}; \mu_p, \mu_z)$$

which iteratively solves a series of unconstrained minimization problems. The details are given in Algorithm 3.2.

It can be seen that Algorithm 3.2 involves three sub problems: the $\mathbf{u}$ -sub problem, the $\mathbf{p}$ -sub problem, and the $\mathbf{z}$ -sub problem. These sub-problems actually can be solved easily.

Algorithm 3.2 ALM algorithm for TV- L^1 decomposition

Initialization: $\mathbf{u}^{(0)} = \mathbf{z}^{(0)} = \mu_z^{(0)} = \mathbf{p}^{(0)} = \mu_p^{(0)} = \epsilon = 0$.

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{z}^{(k+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\mu_p^{(k)}$ and $\mu_z^{(k)}$:

$$(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{z}^{(k+1)}) \approx \arg \min_{\mathbf{u}, \mathbf{p}, \mathbf{z}} L(\mathbf{u}, \mathbf{p}, \mathbf{z}; \mu_p^{(k)}, \mu_z^{(k)})$$

 which is done by the following iterations:

 Let $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}, \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)}$.

for $l = 1, 2, \dots, K$ **do**

 compute

$$\mathbf{u}^{(k+1)} = \arg \min_{\mathbf{u}} L(\mathbf{u}, \mathbf{p}^{(k+1)}, \mathbf{z}^{(k+1)}; \mu_p^{(k)}, \mu_z^{(k)});$$

$$\mathbf{p}^{(k+1)} = \arg \min_{\mathbf{p}} L(\mathbf{u}^{(k+1)}, \mathbf{p}, \mathbf{z}^{(k+1)}; \mu_p^{(k)}, \mu_z^{(k)});$$

$$\mathbf{z}^{(k+1)} = \arg \min_{\mathbf{z}} L(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{z}; \mu_p^{(k)}, \mu_z^{(k)});$$

end for

 and then update

$$\begin{aligned} \mu_p^{(k+1)} &= \mu_p^{(k)} + r_p(\mathbf{p}^{(k+1)} - \nabla_M \mathbf{u}^{(k+1)}) \\ \mu_z^{(k+1)} &= \mu_z^{(k)} + r_z(\mathbf{z}^{(k+1)} - (\mathbf{u}^{(k+1)} - \mathbf{f})) \end{aligned}$$

$$\epsilon = \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{2, V_M^m}, k++.$$

end while

- The $\mathbf{u}$ -sub problem: Given $\mathbf{p}$ and $\mathbf{z}$, solve

$$\begin{aligned} \min_{\mathbf{u}} \{ & \langle \mu_p^{(k)}, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \langle \mu_z^{(k)}, \mathbf{z} - (\mathbf{u} - \mathbf{f}) \rangle_{\mathbf{V}_M^m} \\ & + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 + \frac{r_z}{2} \|\mathbf{z} - (\mathbf{u} - \mathbf{f})\|_{2, \mathbf{V}_M^m}^2 \} \end{aligned}$$

whose optimality condition is

$$\text{div}_M(\mu_p^{(k)}) - \mu_z^{(k)} + r_p \text{div}_M(\mathbf{p} - \nabla_M \mathbf{u}) + r_z(\mathbf{u} - (\mathbf{z} + \mathbf{f})) = 0.$$

This is a system of sparse linear equations.

- The $\mathbf{p}$ -sub problem: Given $\mathbf{u}$ and $\mathbf{z}$, solve

$$\min_{\mathbf{p}} \{ \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \langle \mu_p^{(k)}, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \} \quad (3.29)$$

Similar to the $\mathbf{p}$ -sub problem in the TV- L^2 model, this problem is decomposable and has a closed form solution for each triangle τ :

$$\mathbf{p}_\tau = \max(0, 1 - \frac{1}{r_p |\mathbf{w}_\tau|}) \mathbf{w}_\tau \quad (3.30)$$

$$\text{where } \mathbf{w}_\tau = (\nabla_M \mathbf{u})_\tau - \frac{(\mu_p)^{(k)}_\tau}{r_p}.$$

- The $\mathbf{z}$ -sub problem: Given $\mathbf{u}$ and $\mathbf{p}$, solve

$$\min_{\mathbf{z}} \{ \lambda \|\mathbf{z}\|_{1, \mathbf{V}_M^m} + \langle \mu_z^{(k)}, \mathbf{z} - (\mathbf{u} - \mathbf{f}) \rangle_{\mathbf{V}_M^m} + \frac{r_z}{2} \|\mathbf{z} - (\mathbf{u} - \mathbf{f})\|_{2, \mathbf{V}_M^m}^2 \}. \quad (3.31)$$

Fortunately, this problem has a closed form solution:

$$\mathbf{z} = \max(0, 1 - \frac{\lambda}{r_z |\mathbf{t}|}) \mathbf{t}$$

$$\text{where } \mathbf{t} = \mathbf{u} - \mathbf{f} - \frac{\mu_z^{(k)}}{r_z}.$$

3.3.3 TV-G

To solve the minimization problem (3.22), we introduce two variables $\mathbf{p} \in \mathbf{Q}_M^m$, $\mathbf{h} \in \mathbf{Q}_M^m$ and convert the problem to

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{g}, \mathbf{h}} \{ & \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \kappa \|\mathbf{h}\|_{1, \mathbf{Q}_M^m} + \lambda \|\mathbf{f} - \mathbf{u} - \text{div}_M(\mathbf{g})\|_{2, \mathbf{V}_M^m}^2 \} \\ \text{s.t.} \quad & \mathbf{p} = \nabla_M \mathbf{u}, \quad \mathbf{h} = \mathbf{g} \end{aligned}$$

Similar to the approaches for TV- L^2 and TV- L^1 , we solve this problem by searching for

a solution of the saddle-point problem:

$$\min_{\mathbf{u}, \mathbf{p}, \mathbf{g}, \mathbf{h}} \max_{\mu_p, \mu_h} L(\mathbf{u}, \mathbf{p}, \mathbf{g}; \mu_p, \mu_h)$$

with the augmented Lagrangian functional $L(\mathbf{u}, \mathbf{p}, \mathbf{g}; \mu_p, \mu_h)$ given by

$$\begin{aligned} L(\mathbf{u}, \mathbf{p}, \mathbf{g}; \mu_p, \mu_h) &= \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \kappa \|\mathbf{h}\|_{1, \mathbf{Q}_M^m} \\ &+ \lambda \|\mathbf{f} - \mathbf{u} - \text{div}_M(\mathbf{g})\|_{2, \mathbf{V}_M^m}^2 + \langle \mu_p, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} \\ &+ \langle \mu_h, \mathbf{h} - \mathbf{g} \rangle_{\mathbf{Q}_M^m} + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 + \frac{r_h}{2} \|\mathbf{h} - \mathbf{g}\|_{2, \mathbf{Q}_M^m}^2 \end{aligned}$$

where $\mu_p \in \mathbf{Q}_M^m$, $\mu_h \in \mathbf{Q}_M^m$ are the Lagrangian multipliers, r_p and r_h are positive constants, and $\frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2$ and $\frac{r_h}{2} \|\mathbf{h} - \mathbf{g}\|_{2, \mathbf{Q}_M^m}^2$ are the augmentations. The main steps are given in Algorithm 3.3.

Algorithm 3.3 contains four sub-problems: the $\mathbf{u}$ -sub problem, the $\mathbf{p}$ -sub problem, the $\mathbf{g}$ -sub problem, and the $\mathbf{h}$ -sub problem. They can be solved in the following ways.

- The $\mathbf{u}$ -sub problem: Given $\mathbf{p}$, $\mathbf{g}$ and $\mathbf{h}$, solve

$$\min_{\mathbf{u}} \{ \lambda \|\mathbf{f} - \mathbf{u} - \text{div}_M(\mathbf{g})\|_{2, \mathbf{V}_M^m}^2 + \langle \mu_p^{(k)}, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \}$$

whose solution is

$$(2\lambda - r_p \triangle) \mathbf{u} = 2\lambda \mathbf{f} - \text{div}_M(2\lambda \mathbf{g} + \mu_p^{(k)} + r_p \mathbf{p})$$

where $\triangle = \text{div}_M \circ \nabla_M$.

- The $\mathbf{p}$ -sub problem: Given $\mathbf{u}$, $\mathbf{g}$ and $\mathbf{h}$, solve

$$\min_{\mathbf{p}} \{ \|\mathbf{p}\|_{1, \mathbf{Q}_M^m} + \langle \mu_p^{(k)}, \mathbf{p} - \nabla_M \mathbf{u} \rangle_{\mathbf{Q}_M^m} + \frac{r_p}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{2, \mathbf{Q}_M^m}^2 \}$$

Algorithm 3.3 ALM algorithm for TV-G decomposition

Initialization: $\mathbf{u}^{(0)} = \mathbf{p}^{(0)} = \mathbf{h}^{(0)} = \mathbf{g}^{(0)} = \mu_p^{(0)} = \mu_h^{(0)} = \epsilon = 0$.

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{h}^{(k+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\mu_p^{(k)}$ and $\mu_h^{(k)}$:

$$(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{g}^{(k+1)}) \approx \arg \min_{\mathbf{u}, \mathbf{p}, \mathbf{h}, \mathbf{g}} L(\mathbf{u}, \mathbf{p}, \mathbf{h}, \mathbf{g}; \mu_p^{(k)}, \mu_h^{(k)})$$

 which is done by the following iterations:

 Let $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}, \mathbf{g}^{(k+1)} = \mathbf{g}^{(k)}, \mathbf{h}^{(k+1)} = \mathbf{h}^{(k)}$.

for $l = 1, 2, \dots, K$ **do**

 compute

$$\mathbf{u}^{(k+1)} = \arg \min_{\mathbf{u}} L(\mathbf{u}, \mathbf{p}^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{h}^{(k+1)}; \mu_p^{(k)}, \mu_h^{(k)});$$

$$\mathbf{p}^{(k+1)} = \arg \min_{\mathbf{p}} L(\mathbf{u}^{(k+1)}, \mathbf{p}, \mathbf{g}^{(k+1)}, \mathbf{h}^{(k+1)}; \mu_p^{(k)}, \mu_h^{(k)});$$

$$\mathbf{g}^{(k+1)} = \arg \min_{\mathbf{g}} L(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{g}, \mathbf{h}^{(k+1)}; \mu_p^{(k)}, \mu_h^{(k)});$$

$$\mathbf{h}^{(k+1)} = \arg \min_{\mathbf{h}} L(\mathbf{u}^{(k+1)}, \mathbf{p}^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{h}; \mu_p^{(k)}, \mu_h^{(k)});$$

end for

 and then update

$$\mu_p^{(k+1)} = \mu_p^{(k)} + r_p(\mathbf{p}^{(k+1)} - \nabla_M \mathbf{u}^{(k+1)})$$

$$\mu_h^{(k+1)} = \mu_h^{(k)} + r_h(\mathbf{h}^{(k+1)} - \mathbf{g}^{(k+1)})$$

$$\epsilon = \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{2, V_M^m}, k++.$$

end while

which has a closed form solution for each triangle τ :

$$\mathbf{p}_\tau = \max(0, 1 - \frac{1}{r_p |\mathbf{w}_\tau|}) \mathbf{w}_\tau$$

where $\mathbf{w}_\tau = (\nabla_M \mathbf{u})_\tau - \frac{(\mu_p)_\tau^{(k)}}{r_p}$.

- The $\mathbf{g}$ -sub problem: Given $\mathbf{p}$, $\mathbf{u}$ and $\mathbf{g}$, solve

$$\min_{\mathbf{u}} \{ \lambda \|\mathbf{f} - \mathbf{u} - \text{div}_M(\mathbf{g})\|_{2, \mathbf{V}_M^m}^2 + \langle \mu_h^{(k)}, \mathbf{h} - \mathbf{g} \rangle_{\mathbf{Q}_M^m} + \frac{r_h}{2} \|\mathbf{h} - \mathbf{g}\|_{2, \mathbf{Q}_M^m}^2 \}$$

whose solution is

$$(r_h - 2\lambda(\nabla \circ \text{div})_M) \mathbf{g} = r_h \mathbf{h} - 2\lambda(\nabla_M(\mathbf{f} - \mathbf{u})) + \mu_h^{(k)}.$$

- The $\mathbf{h}$ -sub problem: Given $\mathbf{u}$, $\mathbf{p}$ and $\mathbf{g}$, solve

$$\min_{\mathbf{u}} \{ \kappa \|\mathbf{h}\|_{1, \mathbf{Q}_M^m} + \langle \mu_h^{(k)}, \mathbf{h} - \mathbf{g} \rangle_{\mathbf{Q}_M^m} + \frac{r_h}{2} \|\mathbf{h} - \mathbf{g}\|_{2, \mathbf{Q}_M^m}^2 \}$$

which has a closed form solution for each triangle τ :

$$\mathbf{h} = \max\{0, 1 - \frac{\kappa}{|\mathbf{l}| r_h}\} \mathbf{l}$$

where $\mathbf{l} = \mathbf{g} - \frac{\mu_h^{(k)}}{r_h}$.

3.4 Experiments

The proposed algorithms have been implemented in C++ with CGAL data structure. We carry out experiments under Windows 7 running on a PC with Inter Xeon Core 2.27GHz CPU and 4GB RAM to test the algorithms. Table 3.1 reports the numbers of vertices and

Table 3.1: Statistics of the 3D meshes used in the experiments.

Mesh surface	No. of vertices	No. of triangles
Lena (Fig. 3.5)	65536	130050
Fingerprint (Fig. 3.4)	65536	130050
Barbara (Fig. 3.2)	262144	522242
Cup (Fig. 3.6)	131584	262146
Bunny 1 (Fig. 3.3)	34834	69451
Bunny 2 (Fig. 3.7)	34817	69630

Table 3.2: Statistics for Fig. 3.2 and Fig. 3.3, where N is the iteration number and t is the running time. The stop condition is $\epsilon_0 = 1.e^{-6}$.

Models	L^2 norm				L^1 norm				
	λ	r_p	t	N	λ	r_p	r_z	t	N
Barbara (plane)	5.00	15	12.144s	12	0.30	15	5	17.775s	17
Barbara (manifold)	400	0.015	14.865s	16	40	0.015	200	15.354s	17
Bunny 1	200	0.02	4.279s	22	60	0.02	200	8.298s	43

triangles of the mesh models used in the experiments. The selection of the parameters and the running time for these examples are listed in Table 3.2. The algorithms take only a few seconds to complete the decomposition.

First, we compare our method with conventional image decomposition for planar images. Our proposed method decomposes images on manifolds. Planes are a special case of manifolds. For an image defined on a plane domain, it can be decomposed by our algorithms if we represent the plane domain as a triangular mesh. Fig. 3.2 is an example that shows the L^2 and L^1 decomposition results generated by the conventional planar image decomposition algorithms [159, 130] and our proposed method. The running time is reported in Table 3.2. It can be found from the experimental results that both decomposition methods perform similarly for planar images.

Second, we apply our decomposition algorithm to a textured manifold surface. The manifold surface is a 3D triangular mesh (see Fig. 3.3). Our decomposition algorithm separates the texture from the structure image. It can be seen that with L^2 norm, the structural component is smoother overall while the main structures are still preserved due

Figure 3.2: The top row is conventional L^2 and L^1 image decomposition on 2D plane. The bottom row is our proposed decomposition results on planar manifold. (b) and (d) are the structures. (c) and (e) are the textures.

to the TV term, and with L^1 norm, finer structures can also be captured in the structural component.

Third, we demonstrate the effectiveness of G norm for oscillatory pattern texture and L^1 for sparse signal details. Fig. 3.4 shows an image with oscillatory pattern. We apply TV- L^2 , TV- L^1 and TV- G decomposition models to the input image. Here, for L^2 norm, $\lambda = 250, r_p = 0.04$; for L^1 norm, $\lambda = 250, r_p = 0.04, r_z = 200$; for G norm, $\lambda = 250, r_p = 0.04, r_h = 200, \kappa = 0.01$. The results show that the fidelity with G norm extracts the oscillation best. Fig. 3.5 is an example where the input image contains sparse signal details. The decomposition results show that L^1 norm is more adapted for sparse details. Here, for L^2 norm, $\lambda = 200, r_p = 0.015$; for L^1 norm, $\lambda = 100, r_p = 0.015, r_z = 150$; and for G norm, $\lambda = 200, r_p = 0.015, r_h = 300, \kappa = 0.01$.

3.5 Applications

This section presents two applications of the proposed image decomposition algorithm.

Figure 3.3: Proposed image decomposition on a triangular surface. (a) is the input textured bunny. (b) and (d) are the extracted structures extracted using L^2 and L^1 norms, respectively. (c) and (e) are the extracted textural parts using L^2 and L^1 norms, respectively.

Figure 3.4: Decomposition of an image with oscillatory pattern by the proposed method. (a) is the input image with oscillatory pattern. (b), (c) and (d) are the extracted structural components using L^2 , L^1 and G norms, respectively. (e), (f) and (g) are the extracted textural components using L^2 , L^1 and G norms, respectively.

3.5.1 Detail exaggeration

One application of the structure-texture image decomposition is in image editing on manifolds. Once the structural and textural components of an image are separated, one can edit both components separately. For example, one can simply scale the the textural component to generate exaggerated details. Fig. 3.6(a) is a textured cup. Its structural components extracted by the TV- L^2 and TV- L^1 decomposition methods are shown in Fig. 3.6(b) and (d). The structural components are composed with their corresponding

Figure 3.5: Decomposition of an image with sparse textures by the proposed method. (a) is the input image with salt and pepper noise. (b) and (c) are the structural image and noise extracted from (a) using L^2 norm. (d) and (e) are the structural image and noise extracted from (a) using L^1 norm. (f) and (g) are the structural image and noise extracted from (a) using G norm.

scaled textural components, yielding the effects of detail exaggeration, which are shown in Fig. 3.6(c) and (e).

3.5.2 L^1 image denoising on manifolds

Another application of our proposed method is image denoising on manifolds. As mentioned earlier, image decomposition can be used as a method for image denoising or restoration if the texture is noise. Paper [148] considers the TV- L^2 model for image restoration on manifolds with Gaussian noise. However, if the noise is impulsive, the TV- L^2 model may not work well. Our proposed TV- L^1 model provides a better solution, which is confirmed by the experiments. Figs. 3.7 shows the comparison of image denoising by our TV- L^1 method and the TV- L^2 model of [148]. The parameters and experimental statistics of these examples are given in Table 3.3. The SNRs (signal-to-noise ratios) used in the examples were computed by

$$SNR = 10 \log_{10} \frac{\|\mathbf{u} - E(\mathbf{u})\|_{V_M}^2}{\|\mathbf{u} - \mathbf{g}\|_{V_M}^2}$$

where $\mathbf{u}$ is the noise-free (clean) image, $\mathbf{g}$ is the observed or recovered image, and $E(\mathbf{u})$ is the average intensity of the image $\mathbf{u}$ on M . Both the visual effects and the quantitative results show that our TV- L^1 method outperforms the method of [148] for removing impulsive noise.

Figure 3.6: Detail exaggeration. (a) is the textured cup, (b) and (d) are the extracted structural components using L^2 and L^1 norms. (c) and (e) are the exaggerated detail results by simply scaling the texture components. L^2 norm: $\lambda = 300, r_p = 0.012, t = 10.659s, N = 15$; L^1 norm: $\lambda = 100, r_p = 0.02, r_z = 200, t = 11.225s, N = 16$.

Table 3.3: Statistics for Fig. 3.7 where N is the iteration number and t is the running time.

Models		L^2					
Parameters	SNR (dB)	λ	r_p	N	t	SNR(dB)	
Bunny (10% gray)	-5.740	300	0.015	22	1.934s	6.523	
Bunny (20% gray)	-8.560	280	0.015	24	1.948s	1.806	
Bunny (10% color)	-2.789	300	0.015	26	4.320s	9.038	
Bunny (20% color)	-5.637	250	0.015	28	4.439s	4.639	
Models		L^1					
Parameters	SNR (dB)	λ	r_p	r_z	N	t	SNR(dB)
Bunny (10% gray)	-5.740	120	0.015	200	19	1.717s	13.304
Bunny (20% gray)	-8.560	120	0.015	200	22	1.988s	10.847
Bunny (10% color)	-2.789	120	0.015	200	21	4.115s	15.921
Bunny (20% color)	-5.637	120	0.015	200	23	4.224s	12.075

Figure 3.7: Comparison of the proposed method and the L^2 method of [148] for denoising bunny 2. The models in the first row are gray and color textured manifolds added with 10%, 20% salt and pepper noise, respectively. The models in the second row are the denoising results using the method of [148]. The models in the third row are the denoising results using the proposed TV- L^1 method.

Chapter 4

Mesh Denoising Using the ROF Model

This chapter considers the problem of mesh denoising and presents variational ROF model for robust, feature-preserving triangular mesh denoising. In this variational model, the function domain is the triangular mesh itself, changing during the denoising process.

4.1 Motivation and Our Work

Mesh denoising is a common process in geometry processing to improve the imperfect meshes obtained from various sources. Although a lot of mesh denoising methods already exist, research on decoupling features from noise still remains active due to its challenging nature. The reasons for this are three-fold. First, it is difficult to distinguish features from noises. Features and noise are ambiguous since there are no precise mathematical definitions for them. Local geometric features which could be of either low or high frequency should be preserved during the denoising processing. This problem is especially problematic to distinguish sharp edges and corners which are of high frequency from the noises that themselves are also often of high frequency. It would be even harder when the noises levels are high and the directions are random. Second, how to find an appropriate differential quantity or operator that can well reflect geometry properties is not clear. Traditional approaches adopt the vertex normal, face normal, mean curvature

flow to describe the geometry features. Different choices will affect the final denoising results. Isotropic operators ignore the sharp features and work well for overall smooth surfaces, while anisotropic operators attempt to preserve the sharp features for piecewise-wise smooth surface but may cause over-sharpen problem. Third, the computation accuracy of above differential descriptors for geometry features and parameters, are not reliable in the presence of noise, especially for non-iteration and global methods.

A wide variety of mesh denoising algorithms already exist. Most early surface smoothing methods are isotropic, which means the filters are independent of surface geometry directions. They usually average the neighborhood to remove the noise but also smooth sharp features and shrink the surfaces. Laplacian smoothing is an example.

Given that isotropic methods do not preserve sharp features, many recent techniques have focused on anisotropic approaches, such as anisotropic diffusion approaches, bilateral filters, which are extended from image processing. Bilateral filters in [49] and [65] anisotropically average the nearby vertices weighted by monotonously decreasing function in term of both spatial difference and vertex difference. Directly applying the bilateral filter to vertex positions fails to preserve features, since this would mean using the same set of vertex positions to measure both signal difference and spatial difference, reducing the bilateral filter to a unilateral filter. Several researchers have also explored the idea of filtering face normals defined over the mesh. The normal field is first smoothed and the denoised surface was reconstructed from the filtered normal fields by updating vertex positions. While updating vertices is trivial, filtering the normal field is important in the quality of the final surface. This filter performs well for sharp features preservation in low noise situations while it fails when noise is in high level and in random directions. It will cause severe fold-over problems.

Most recently, He et al. [61] introduce a mesh denoising scheme using L^0 normal minimization and achieve impressive results for sharp feature preservation. A new discrete differential operator for arbitrary triangular mesh is built and L^0 norm is applied to measure the sparsity of the sharp features. The new differential operator generalizes the construction

of vertex-based operator to edge which better measures the sharpness of the surface. The advantage of L^0 norm minimization is that it can handle large amounts of noise and produces high quality smoothing results. This norm is a good choice if we have the prior knowledge that the underlying surface is piecewise planar or smooth. However, it has some disadvantages as well. First, it is “too hard” for smooth surfaces with small scale geometry features which will be flattened especially when co-existing with high scale noises. This is because there are only two extreme cases of L^0 norm: $|x|_0 = 0$ only if $x = 0$, otherwise $|x|_0 = 1$, which will over-penalize the non-zero values. Another issue for L^0 norm minimization is the challenge of solving the final optimization problem. It is difficult to optimize due to the combinatorial nature. Although some algorithms have been proposed to approximately solve the problem, it is still an open problem to prove the existence and uniqueness of the solution theoretically.

Our work

In this research, we extend the famous variational ROF image denoising model [112] for triangular mesh denoising. Candes et al. [19] have shown that the L^1 norm provides a good measure of sparsity. The final optimization problem is convex which makes it possible to solve. We discover that the L^1 norm can not only remove noise for piecewise planar/smooth surfaces but also avoid the “too flat” and “over-sharpen” problems for small scale geometry features especially with the presence of different scales of noises. Fig. 4.1 illustrates the difference of L^0 and L^1 norm for fairing in the curve case. Different scales of noises are added. The differential operator for the curve is Laplacian, i.e., $L(x_i) = (x_{i+1} - x_i) - (x_i - x_{i-1})$. The example shows that the L^0 norm uses line segments to approximate the curve and will over-flatten the curve feature in order to remove the large scale noise while the L^1 norm will not.

The main challenges of extending the ROF model to triangular mesh include: a) how to define the function over triangular mesh; b) how to define an appropriate differential operator for meshes to capture the features as gradient in images; c) how to deal with the

Figure 4.1: (a) is the noisy fish smoothed by L^0 norm, (b) is the noisy fish smoothed L^1 norm, (c) top is the piecewise constant shape smoothed by L^0 norm and (c) bottom is smoothed by L^1 norm.

function domain that is the triangular mesh itself which also changes during the denoising process. A lot of efforts have done before to extend this model from image to triangular mesh. Wu et al. [145] introduce diffusion equations over a triangular mesh for filtering which is an approximation of the total variation on the triangular mesh. However, the triangular mesh is fixed in his algorithm. Recently, Elsey et al. [46] analogue the total variation to geometry processing by total curvature which is isotropic and fails to preserve the sharp features.

We present a variational ROF model to overcome these difficulties. The main benefit of our variational model is that it is more comprehensive which can handle both overall smooth and piecewise planar/smooth surfaces with large amount of different types of noise and produces higher quality results. Moreover, the final optimization problem is convex and can be solved by the augmented Lagrangian method whose convergence analysis has been proved.

4.2 Variational Mesh Denoising

Given a mesh M with vertex set $\mathbf{v}^0$ containing noise, mesh denoising is to find a clean vertex set $\mathbf{v}$ to replace $\mathbf{v}^0$. We formulate this as a decomposition problem

$$\mathbf{v}^0 = \mathbf{v} + \Delta$$

where $\mathbf{v}$ preserves the structure and shape of $\mathbf{v}^0$ and Δ stands for the noise. It is worth pointing out that in this formulation $\mathbf{v}$ is a function coupled with the function domain M which is also related to the positions of vertices, which is different from previous work and also makes our problem challenging.

The variational ROF model for color function $\mathbf{v}$ defined on manifolds M has already been introduced in (3.6), Section 3.2, Chapter 3, which is

$$\inf_{\mathbf{v}} \left(\int_M |\nabla_M \mathbf{v}| dM + \frac{\lambda}{2} \|\mathbf{v} - \mathbf{v}^0\|_{2,M}^2 \right) \quad (4.1)$$

The differential structure, gradient and divergence operators of the manifolds M have also been defined there. Hence in this section we just directly introduce our mesh denoising model when the manifold is approximated by triangular mesh which we also denote by M . The set of vertices, the set of edges and the set of triangle faces of M are denoted by $\{v_i = (v_i^x, v_i^y, v_i^z) : i = 1, 2, \dots, N_v\}$, $\{e_i : i = 1, 2, \dots, N_e\}$, $\{\tau_i : i = 1, 2, \dots, N_\tau\}$, respectively, where N_v, N_e, N_τ are the numbers of vertices, edges and triangles. We simply replace the color function $\mathbf{v}$ with the vertex positions. That is, the vector valued field $\mathbf{v} = \{\mathbf{v}^x, \mathbf{v}^y, \mathbf{v}^z\}$ with $\mathbf{v}^x = (v_1^x, v_2^x, \dots, v_{N_v}^x)$, $\mathbf{v}^y = (v_1^y, v_2^y, \dots, v_{N_v}^y)$, $\mathbf{v}^z = (v_1^z, v_2^z, \dots, v_{N_v}^z)$. $\mathbf{v}$ reaches values $\mathbf{v}_j = (v_j^x, v_j^y, v_j^z)$ at vertex $v_j, j = 1, 2, \dots, N_v$. Two Hilbert spaces are defined as $\mathbf{V}_M = V_M \times V_M \times V_M$ with $V_M = \mathbb{R}^{N_v}$ and $\mathbf{Q}_M = Q_M \times Q_M \times Q_M$ with $Q_M = \mathbb{R}^{3 \times N_\tau}$. The linear products and norms for $\mathbf{V}_M$ and $\mathbf{Q}_M$ are defined as (3.12), (3.13), (3.15) and (3.16).

As the gradient operator ∇_M can not capture the surface feature well which is isotropic,

Figure 4.2: One ring of a triangle (left) and edge (right).

we must design a new differential operator which not only removes the noise but also keeps the sharp feature. We introduce a new discrete differential face operator to replace ∇_M in (4.1). The idea is similar to the discrete gradient defined in (3.19) which defines as the height vector. Since the height vector defined in the tangent plane of the surface can not capture the sharp features well, we need some form of second order information rather than the first order information provided by gradient. He et al. [61] design a new differential edge operator that acts directly on an edge which is independent of translations and rotations. The operator extends the barycentric construction in [50] to a differential edge operator. Through the divergence theorem described in [117], the integral of an outward facing normal over a closed shape is zero. The same construction applied to edge builds the differential edge operator as shown in Fig. 4.2 (right).

$$D(e) = \begin{bmatrix} -\cot(\theta_{2,3,1}) - \cot(\theta_{1,3,4}) \\ \cot(\theta_{2,3,1}) + \cot(\theta_{3,1,2}) \\ -\cot(\theta_{3,1,2}) - \cot(\theta_{4,1,3}) \\ \cot(\theta_{1,3,4}) + \cot(\theta_{4,1,3}) \end{bmatrix}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

The geometry meaning of $D(e) = 2 \sin(\frac{\gamma}{2}) |v_3 - v_1|$ where γ is the dihedral angle between the two triangles is a measure of mean curvature. To prevent the surface from fold-over

triangles, the improved edge operator can be represented by the area of triangles,

$$D(e) = \begin{bmatrix} -\Delta_{2,3,4} \\ \Delta_{1,3,4} \\ -\Delta_{1,2,4} \\ \Delta_{1,2,3} \end{bmatrix}^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

where $\Delta_{j,k,l}$ refers to the area of triangle with vertices v_i, v_j, v_k . When the one ring of edge is planar, the area-based edge operator $D(e) = 0$, which well describes the local information of a triangular mesh.

Assume the new face differential operator $\mathbf{D}_\tau(\mathbf{v}) \in \mathbf{Q}_M$ for function $\mathbf{v}$ with $\mathbf{D}_\tau(\mathbf{v}) = (\mathbf{D}_\tau^x(\mathbf{v}), \mathbf{D}_\tau^y(\mathbf{v}), \mathbf{D}_\tau^z(\mathbf{v}))$. $\mathbf{D}_{\tau_j}(\mathbf{v})$ denotes the value at triangle τ_j . For each triangle τ_j which is shown as Fig. 4.2 (left), the three edges of the triangle τ_j is e_1, e_2, e_3 . As we have introduced the edge operator for edges, we can define $\mathbf{D}_{\tau_j}(\mathbf{v})$ as follows:

$$\mathbf{D}_{\tau_j}(\mathbf{v}) = (\mathbf{D}_{\tau_j}^x(\mathbf{v}), \mathbf{D}_{\tau_j}^y(\mathbf{v}), \mathbf{D}_{\tau_j}^z(\mathbf{v}))$$

with $\mathbf{D}_{\tau_j}^x(\mathbf{v}) = (D_{\tau_j}^x(e_1), D_{\tau_j}^x(e_2), D_{\tau_j}^x(e_3))$, $\mathbf{D}_{\tau_j}^y(\mathbf{v}) = (D_{\tau_j}^y(e_1), D_{\tau_j}^y(e_2), D_{\tau_j}^y(e_3))$, $\mathbf{D}_{\tau_j}^z(\mathbf{v}) = (D_{\tau_j}^z(e_1), D_{\tau_j}^z(e_2), D_{\tau_j}^z(e_3))$.

Besides the two terms of the ROF model, for each face τ_j , an additional term $R_{\tau_j}(\mathbf{v}) = (R_{\tau_j}(e_1), R_{\tau_j}(e_2), R_{\tau_j}(e_3))$ is also introduced to improve the quality of the triangulation of the final mesh, where $R_{\tau_j}(e) = (v_1 - v_2 + v_3 - v_4)$ for each edge e shown in Fig. 4.2 (right). This gives a new functional $\mathbf{R}_\tau(\mathbf{v}) \in \mathbf{Q}_M$. Hence, our final discrete optimization model can be formulated as

$$\min_{\mathbf{v}} \{ \|\mathbf{D}_\tau(\mathbf{v})\|_{1, \mathbf{Q}_M} + \frac{\lambda}{2} \|\mathbf{v} - \mathbf{v}^0\|_{2, \mathbf{V}_M}^2 + \frac{\alpha}{2} \|\mathbf{R}_\tau(\mathbf{v})\|_{2, \mathbf{Q}_M}^2 \} \quad (4.2)$$

where $\|\cdot\|_{1, \mathbf{Q}_M}$, $\|\cdot\|_{2, \mathbf{V}_M}$, $\|\cdot\|_{2, \mathbf{Q}_M}$ are defined in Chapter 3 and α is the tuning parameter for $\mathbf{R}_\tau(\mathbf{v})$. For given function domain M , both vectors $\mathbf{D}_\tau(\mathbf{v})$ and $\mathbf{R}_\tau(\mathbf{v})$ can be represented

as matrix times vector form, that is $\mathbf{D}_\tau(\mathbf{v}) = \mathbf{K}_1\mathbf{v}$, $\mathbf{R}_\tau(\mathbf{v}) = \mathbf{K}_2\mathbf{v}$, where $\mathbf{K}_1, \mathbf{K}_2$ are $N_\tau \times N_v$ matrices. We notice that when the function $\mathbf{v}$ changes, the function domain M also changes. An iterative algorithm is developed to find the best $\mathbf{v}$. Both of $\mathbf{v}$ optimization and M update alternate until convergence.

4.3 Algorithm

We again perform the augmented Lagrangian method to solve (4.2). A new variable $\mathbf{p} = \mathbf{K}_1\mathbf{v}$ is introduced and the problem becomes

$$\begin{aligned} \min_{\mathbf{v}, \mathbf{p}} \{ & \|\mathbf{p}\|_{1, \mathbf{Q}_M} + \frac{\lambda}{2} \|\mathbf{v} - \mathbf{v}^0\|_{2, \mathbf{V}_M}^2 + \frac{\alpha}{2} \|\mathbf{K}_2\mathbf{v}\|_{2, \mathbf{Q}_M}^2 \} \\ \text{subject to } & \mathbf{p} = \mathbf{K}_1\mathbf{v} \end{aligned}$$

We minimize the following augmented Lagrangian functional

$$\begin{aligned} \mathbf{L}(\mathbf{v}, \mathbf{p}; \lambda) = & \|\mathbf{p}\|_{1, \mathbf{Q}_M} + \frac{\lambda}{2} \|\mathbf{v} - \mathbf{v}^0\|_{2, \mathbf{V}_M}^2 + \frac{\alpha}{2} \|\mathbf{K}_2\mathbf{v}\|_{2, \mathbf{Q}_M}^2 \\ & + \langle \lambda_p, \mathbf{p} - \mathbf{K}_1\mathbf{v} \rangle_{\mathbf{Q}_M} + \frac{r_p}{2} \|\mathbf{p} - \mathbf{K}_1\mathbf{v}\|_{2, \mathbf{Q}_M}^2 \end{aligned} \quad (4.3)$$

where the Lagrangian multiplier $\lambda_p \in \mathbf{Q}_M$, $r_p > 0$, $\langle \cdot, \cdot \rangle_{\mathbf{Q}_M}$ is the inner product in $\mathbf{Q}_M$. The entire procedure for solving the problem is summarized in Algorithm 4.1.

In Algorithm 4.1, the solution is obtained by iteratively solving two sub-problems: the $\mathbf{v}$ -sub problem and the $\mathbf{p}$ -sub problem.

- $\mathbf{v}$ -sub problem: we fix $\mathbf{p}$ and solve

$$\frac{\lambda}{2} \|\mathbf{v} - \mathbf{v}^0\|_{2, \mathbf{V}_M}^2 + \frac{\alpha}{2} \|\mathbf{K}_2\mathbf{v}\|_{2, \mathbf{Q}_M}^2 + \langle \lambda_p, \mathbf{p} - \mathbf{K}_1\mathbf{v} \rangle_{\mathbf{Q}_M} + \frac{r_p}{2} \|\mathbf{p} - \mathbf{K}_1\mathbf{v}\|_{2, \mathbf{Q}_M}^2$$

- $\mathbf{p}$ -sub problem: we fix the $\mathbf{v}$ and solve

$$\|\mathbf{p}\|_{1, \mathbf{Q}} + \langle \lambda_p, \mathbf{p} - \mathbf{K}_1\mathbf{v} \rangle_{\mathbf{Q}_M} + \frac{r_p}{2} \|\mathbf{p} - \mathbf{K}_1\mathbf{v}\|_{2, \mathbf{Q}_M}^2$$

Algorithm 4.1 ALM algorithm for Equation 4.3

Initialization: $\mathbf{v}^{(0)} = \mathbf{p}^{(0)} = \lambda_p = \epsilon = 0$.

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{v}^{(k+1)}, \mathbf{p}^{(k+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multiplier $\lambda_p^{(k)}$:

$$(\mathbf{v}^{(k+1)}, \mathbf{p}^{(k+1)}) \approx \arg \min_{\mathbf{v}, \mathbf{p}} \mathbf{L}(\mathbf{v}, \mathbf{p}; \lambda_p^{(k)})$$

 which is done by the following iterations:

 Let $\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)}$.

for $l = 1, 2, \dots, K$ **do**

 compute $\mathbf{v}^{(k+1)} = \arg \min_{\mathbf{v}} L(\mathbf{v}, \mathbf{p}^{(k+1)}; \lambda_p^{(k)});$

 compute $\mathbf{p}^{(k+1)} = \arg \min_{\mathbf{p}} L(\mathbf{v}^{(k+1)}, \mathbf{p}; \lambda_p^{(k)});$

end for

 and then update

$$\lambda_p^{(k+1)} = \lambda_p^{(k)} + r_p(\mathbf{p} - \mathbf{K}_1 \mathbf{v})$$

$M \leftarrow \mathbf{v};$

$\epsilon = \|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|_{2, \mathbf{V}_M}^2, k++.$

end while

Both the $\mathbf{v}$ -sub problem and the $\mathbf{p}$ -sub problem have closed form solutions. According to [149], we set the inner iteration $K = 1$. We update the function domain M in each iteration and recompute the new differential operator $\mathbf{D}_\tau(\mathbf{v})$ for $\mathbf{v}$.

4.4 Experiments

We have tested our denoising scheme on a variety of models and this section presents some experimental results. For the default parameters in our method, we set $\epsilon_0 = 0.0005l_e, r_p = 10, \alpha = 0.1\bar{\gamma}, \lambda = \frac{1}{l_e\bar{\gamma}}, l_e$ is the mean edge length, $\bar{\gamma}$ is the average dihedral angle of the initial surface. Our implementation uses Intel MKL sparse solver [1] to solve our sparse linear system and uses Intel(R) Xeon(R) CPU E5-1650 to perform the test. The final iteration number in our solver until stop is about 40-50.

We first present two comparisons to illustrate that our proposed variational method can preserve the sharp features for both piecewise constant and piecewise smooth surfaces. We

Figure 4.3: From left to right, top to down: the original surface, the input surface corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.4l_e$, bilateral filtering [49], prescribed mean curvature flow [62], mean filtering [154], bilateral normal filtering [163], L^0 norm [61] and our method.

have compared our method against several popular previous methods. Fig. 4.3 shows a comparison for piecewise constant surfaces. The first seven results are from paper [61] and the last one is our smoothed model. The input noisy mesh is corrupted by Gaussian noise with standard deviation $\sigma = 0.4l_e$ in random directions. The previous fourth methods fail in presence of such high level noise in random directions. It shows that both L^0 norm minimization method and our method still perform pretty well. Fig. 4.4 shows another comparison for a piecewise smooth surface. The first six results are from paper [61]. Table 4.1 lists the errors and computation time for Fig. 4.2 and Fig. 4.3. The columns e_v and e_H list the vertex-based error and Hausdorff distance between the denoised meshes and corresponding ground-truth models, respectively. Even though L^0 norm is designed particularly for piecewise constant surface denoising, two comparisons show that our denoising method produces high quality results for surfaces with sharp features and handles well for large amount of noise.

Both L^0 and our variational method work well for sharp feature preservation. However, the extreme penalty property of L^0 makes the small scale geometry features over-penalized.

Figure 4.4: From left to right: surface corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.3l_e$, bilateral filtering [49], prescribed mean curvature flow [62], mean filtering [154], bilateral normal filtering [163], L^0 norm [61] and our method.

Table 4.1: Statistic for Fig.4.2 and Fig. 4.3.

Models	Methods	e_v	e_H	Time (s)
Dodeca ($l_e = 0.024754$ $N_v = 7682$ $N_e = 23040$ $N_\tau = 15360$ Fig. 4.3)	bilateral filtering [49]	0.013395	0.027507	0.641
	prescribed mean curvature [62]	0.013180	0.026657	10.679
	mean filtering [49]	0.013020	0.025447	1.252
	bilateral normal filtering [163]	0.013541	0.026068	1.019
	L^0 norm [61]	0.010740	0.024168	26.713
	our method	0.010448	0.020460	25.941
Fandisk ($l_e = 0.028459$ $N_v = 6475$ $N_e = 19419$ $N_\tau = 12946$ Fig. 4.4)	bilateral filtering	0.015621	0.030114	0.530
	prescribed mean curvature [62]	0.014207	0.028160	8.691
	mean filtering [49]	0.014314	0.027203	0.975
	bilateral normal filtering [163]	0.013325	0.026203	0.912
	L^0 norm [61]	0.010749	0.023440	19.208
	our method	0.009031	0.019146	19.877

Figure 4.5: From left to right, top to down: the original Venus head model, the noisy input mesh corrupted by impulsive noise with variation $0.4l_e$ and Gaussian noise in random directions with standard deviation $\sigma = 0.15l_e$, the smoothed model by L^0 and our smoothed model.

Figs. 4.5 and 4.6 show the comparison between L^0 norm and our method for model with small geometry features such as the hair area in Venus model and the teeth in the Skull model. In Fig. 4.5, two types of noise are added with variation $0.4l_e$ and $0.15l_e$, respectively. Not only the noise but also the geometry feature like the hair is removed by L^0 norm. Fig. 4.6 shows the denoising processing of L^0 norm and our method by increasing the smooth term gradually for both methods. In order to remove the high scale noise near the teeth area, L^0 norm flattens the small geometry feature as shown in the close-up views near teeth area while our method not only removes the noise but also preserves the small geometry features well.

Figure 4.6: Skull model. From left to right: the second row shows the noisy input, L^0 norm smoothing results by increasing smoothing term . The third row shows the original mesh, our smoothing results with fidelity tuning parameter parameter $6\lambda_0$, $3\lambda_0$, λ_0 where λ_0 is the default fidelity parameter. The first and fourth rows are the local close-up views.

Figure 4.7: Bunny model. From left to right: the first row: the original model, the noisy input corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.2l_e$ and bilateral filtering [49]; The second row: the close-up views; The third row: bilateral normal filtering [163], L^0 norm [61] and our method; The fourth row: the close-up views.

Fig. 4.7 gives another comparison of our method with previous methods for denoising surfaces whose original surface is overall smooth as the first model. It contains some small scale features as shown in the close-up view. The model is corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.2l_e$. Both bilateral filters fail to clean the surface. L^0 normal method fails to preserve the small scale geometry. Our method performs well for both noise removing and small geometry feature preservation.

Chapter 5

Unsupervised Construction of B-spline Contours from Unorganized Points

This chapter presents a variational algorithm for constructing a B-spline curve for point data sampled from an unknown, piecewise smooth contour.

5.1 Motivation and Our work

A contour-based shape representation describes the shape by its boundary. Extracting meaningful information and features from the contour data has been widely used in shape analysis, matching, retrieval and object recognition. Usually the contour data are a set of dense points acquired from sensors or extracted from image, which are frequently hampered with aliasing, occlusion, noise and outliers. This chapter presents an unsupervised algorithm for constructing a B-spline curve to fit the unorganized contour points.

The B-spline curve is one of the most efficient curve representations and has been widely used in industry. Many methods have been developed based on B-spline curve representation in shape matching [31, 143], retrieval [73], object identification [59]. For example, in paper [48], an unsupervised B-spline curve representation is developed in a statistical framework for image contours. However, most existing methods do not work

well if different types of noise such as Gaussian noise and outliers co-exist in the point cloud. They assume that the points on the contour have been ordered, but ordering points itself is also a difficult problem in contour extraction. They often require an appropriate shape as the initial guess.

The challenges of our problem lie in several aspects: (1) The noise in the source data could be of different types. Different types of noise such as Gaussian noise and outliers have different behaviors. Usually different strategies are used to handle different types of noise. How to design an algorithm that can handle these different types of noise simultaneously is not trivial. (2) Noise often co-exists with sharp features of the shape. It is difficult to intelligently distinguish noise and features. (3) Different from previous parametric approximation approaches where the input point set is usually ordered, our input is an unorganized point set. How to assign intrinsic parameter values for unorganized point cloud for the fitting process is also an issue. (4) The specification of a B-spline curve involves degree, knot vector, and control points. Many previous methods specify knot vector heuristically in advance and then the problem becomes a problem that only finds the optimal control points. Therefore the solution is obtained from a sub-optimal problem. It is difficult to obtain a solution from a global optimization that involves knot vector and control points simultaneously.

Unlike previous work using least squares metric, our approach is to find a new variational energy function to better measure for shape fitting. A new term named total Laplacian is defined for sparse feature as the total variation. We formulate the problem of fitting a B-spline curve to a set of given data points as a variational L^1 minimization problem, which solves for the knot vector and control points simultaneously. The new metric based TL and L^1 -norm enables us to decouple feature from various types of noise effectively and makes the algorithm insensitive to noise. Our contributions include a new variational formulation for different types of noises and an efficient numerical algorithm for solving the minimization problem based on augmented Lagrangian method (ALM) which is realized by solving a few subproblems. These subproblems can have explicit

solutions, be solved by solving a system of linear equations, or be solved by the LM-algorithm. Moreover, it is adaptive and there is no need to set the initial shape manually for unorganized point cloud. The proposed algorithms outperform previous methods both visually and quantitatively, as confirmed in experiments.

5.2 Unsupervised Contour Construction

This section describes B-spline curve reconstruction for unorganized noisy contour points. The input is a set of unorganized noisy points. The output is a compact cubic B-spline curve that fits the points within a pre-specified tolerance. Though we restrict our discussion to cubic B-splines, the extension to general degree is straightforward. The presented reconstruction is formulated into a variational optimization problem. It is unsupervised and adaptive.

5.2.1 Basic ideas

To define a cubic B-spline curve, we need to determine deBoor points and knots. It is well-known that a good distribution of knots improves B-spline fitting dramatically. While most previous methods use some heuristics to determine the knots in a preprocess step and then focus on the optimization of deBoor points, our idea is to treat both knots and deBoor points as variables in a variational optimization framework. In this way, the knots are automatically placed to fit the characteristic of the target shape. In general, more knots will be placed near the high curvature area. The side effect of this approach is that the resulting variational model will be nonlinear and complicated. Therefore we need well-designed numerical solvers.

Next we need an appropriate variational model to constrain the deBoor points and knots. Basically we employ the ROF model which has two terms: the fidelity term and the regularization term. However, we need to adapt them for our problem.

The fidelity term measures how close the reconstructed curve is to the input data. To

facilitate the description, we assume that input data are points $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_N\}$, they have associated parameters $\mathbf{u} = \{u_1, \dots, u_N\}$ and the reconstructed curve is denoted by $\mathbf{r}(u)$. Most previous work uses the least squares metric which is defined by

$$\min \sum_{i=1}^n \|Q_i - \mathbf{r}(u_i)\|^2 \quad (5.1)$$

This is the minimum mean-square error (MMSE) defined in previous methods for B-spline shape representation [32, 59, 31]. However, when the given points are noisy and contain outliers, the MMSE can not produce a desirable curve especially near the sharp feature parts although a cubic B-spline curve itself is C^2 -continuous. Therefore previous methods often require a preprocess step to remove the noise. Notice that the L^1 norm has a nice behavior for removing a small set of outliers from a set of data points, as shown in paper [110]. Hence, we define the L^1 norm metric in (5.2) as minimum mean error (MME) criterion.

$$\min \sum_{i=1}^n \|Q_i - r(u_i)\| \quad (5.2)$$

The regularization term is to maintain the shape smoothness without losing the important geometric features of the shape. Inspired by the success of total variation in image processing [112], we will design a new operator which well captures the feature of a curve. Note that curvature is an important descriptor for a curve, which implies that we need some form of second order information rather than the first order information for the curve case. We define the total variation of gradient, named the total Laplacian (TL) and denoted by E_1^s , for a B-spline curve. For comparison, we also introduce E_2^s in terms of the L^2 norm.

$$E_1^s = \int |\mathbf{r}''(u)| du \quad E_2^s = \int |\mathbf{r}''(u)|^2 du$$

Fig. 5.1 shows the advantage of using our total Laplacian for regularization term.

Thus by adding the total Laplacian term into the L^1 fidelity term, we have our adapted

Figure 5.1: (a) is the fitting result using E_2^s . (b) is the fitting result using our total Laplacian E_1^s .

Figure 5.2: Comparison of different metrics, (a) is by MMSE. (b) is by $TL - L^2$ with $\lambda = 0.1$. (c) is by $TL - L^2$ with $\lambda = 0.01$. (d) is by MME. (e) is by $TL - L^1$ with $\lambda = 0.1$.

ROF model

$$\min_{\mathbf{r}} \int |\mathbf{r}''(u)| du + \lambda \sum_{i=1}^n \|Q_i - r(u_i)\| \quad (5.3)$$

which aims to maintain smoothness of the shape meanwhile preserving the sharp feature. As shown in Fig. 5.2(a), if only the MMSE is used in fitting, the overall smoothness of the curve will be affected by the outliers. The result can be improved by adding the total Laplacian smoothing term. However, if we want to avoid the bumps produced by outliers, it is at the expense of smoothing the sharp corner, as showed in Fig. 5.2(b) and (c). L^1 norm is more suitable for removing the outliers than L^2 norm, but only the MME still can not maintain the smoothness of shape, as shown in Fig. 5.2(d). With the adapted ROF model, we can remove the outliers while preserving the importance features of shape and maintaining the smoothness of the shape, as shown in Fig. 5.2(e).

Figure 5.3: $ATL - L^1$ algorithm. (a) is the quad-tree construction. The dark-red cell are the cells chosen for initial shape construction. (b) is the initial shape by connecting the centers of the dark-red cells in (a). (c) is the B-spline curve by our $ATL - L^1$ from the initial shape with 35 control points. (d) is the reconstructed B-spline curve.

5.2.2 Algorithm

With the adapted ROF model, we are now ready to present our adaptive $TL - L^1$ B-spline fitting ($ATL - L^1$) algorithm. Algorithm 5.1 outlines the main steps.

Algorithm 5.1 Adaptive $TL - L^1$ B-spline fitting

{Initialization}

Construct initial closed polygon shape, assign the
parameter value and get the initial B-spline shape

{Main loop}

while (not termination condition) **do**

Reparameterization

Knot insertion

$TL - L^1$ B-spline fitting

end while

Step 1. Initialization: Unlike most previous work which generates the initial shape manually for unorganized points, we propose a quad-tree based decomposition to find the initial closed polygon, as shown in Fig. 5.3. We first construct the quad-tree by finding the cells which contain points in point clouds. According to the neighborhood relationship established during the quad-tree construction, we find the cells which satisfy:

- (a) The cells contain points but have neighborhood containing no point;

- (b) Any adjacent three cells are not collinear;
- (c) Any adjacent two cells are not in 8-neighborhood.

The cells are marked in dark-red, as shown in Fig. 5.3(a). The initial closed polygon is constructed by connecting the centers of the dark-red cells.

Then, we assign a parameter value u_i for each point $Q_i \in \mathbf{Q}$ by projecting Q_i to the closed polygon, calculating the distance from the starting point to the projective point. The initial B-spline curve is simply generated by using the initial polygon shape as the control polygon. An alternative way is just to use the boundary of the bounding box as the initial polygon. Since our proposed method is not very sensitive to the initial polygon, this works fine in many situations.

Step 2. Termination condition: The distance between $\mathbf{Q}$ and the reconstructed B-spline curve $\mathbf{r}(u)$ can be defined in various ways. Here, we use the mean distance $\bar{e}$ and Hausdorff distance e_H to evaluate our approach.

$$\bar{e} = \frac{\sum_{i=1}^N \|Q_i - \mathbf{r}(u)\|}{N}$$

$$e_H = \max\{d(\mathbf{Q}, \mathbf{r}(u)), d(\mathbf{r}(u), \mathbf{Q})\}$$

$\|Q_i - \mathbf{r}(u)\|$ is the minimal distance from Q_i to the B-spline curve $\mathbf{r}(u)$. $d(\mathbf{Q}, \mathbf{r}(u))$ and $d(\mathbf{r}(u), \mathbf{Q})$ are the distance from $\mathbf{Q}$ to $\mathbf{r}(u)$ and from $\mathbf{r}(u)$ to $\mathbf{Q}$, respectively. The termination conditions of $ATL - L^1$ algorithm are $\bar{e} < \epsilon_1$, where ϵ_1 is a pre-specified error threshold or the maximal number of knots is greater than the given number.

Step 3. Reparameterization: The parameterization also has dramatic effect for B-spline representation. Hence, we update the parameter value u_i by projecting Q_i to a refined closed polygon which is adaptively sampled from the current B-spline curve.

Step 4. Knot insertion: A new knot $\frac{t_j + t_{j+1}}{2}$ is inserted adaptively to the segment with the biggest mean error, which bounded by two knots t_j and t_{j+1} .

Step 5. Perform $TL - L^1$ B-spline fitting described in Section 5.3.

5.3 $TL - L^1$ B-spline Fitting

$TV - L^1$ B-spline fitting is the heart of the construction algorithm. Given a set of unorganized noisy contour points $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_N\}$ that may contain noise or outliers and a set of parameter value $\mathbf{u} = \{u_1, \dots, u_N\}$ with u_i corresponding to Q_i in $\mathbf{Q}$, we want to find a cubic B-spline curve:

$$\mathbf{r}(u) = \sum_{i=1}^n B_i(u) P_i$$

such that $\mathbf{r}(u)$ best fits $\{Q_i\}_{i=1}^N$ under the $TL - L^1$ criterion where $\{B_i(u)\}_{i=1}^n$ are the B-spline basis defined on the knot vector $\{t_j\}, j = 1, \dots, n+4$. That is to find the best deBoor points $\mathbf{P} = \{P_1, P_2, \dots, P_n\}$ and the knot vector $\mathcal{T} = \{t_1, t_2, \dots, t_{n+4}\}$ minimizing

$$\int |\mathbf{r}''(u)| du + \lambda \|\mathbf{r}(u) - \mathbf{Q}\|$$

where λ is a tradeoff factor. The first term of the objective function is the total Laplacian of the B-spline curve measuring the smoothness and the second term is the fidelity term in L^1 norm measuring the approximate error. The problem can be discretized as follows:

$$\min_{\mathbf{P}, \mathcal{T}} \sum_{l=1}^m \|\Delta \mathbf{r}(s_l)\| + \lambda \sum_{k=1}^N \|\mathbf{r}(u_k) - Q_k\|$$

where $\mathbf{s} = \{s_1, \dots, s_m\}$ are the sample parameters on the B-spline curve $\mathbf{r}(u)$ and $\Delta \mathbf{r}(s_l) = \mathbf{r}(s_{l+1}) - 2\mathbf{r}(s_l) + \mathbf{r}(s_{l-1})$ stands for Laplacian of curve in parameter s_l .

Due to the TL term and L^1 -norm, the minimization problem is non-quadratic and non-differentiable. We adapt the augmented Lagrangian method to solve the $TL - L^1$ minimization. We introduce two new variables $\mathbf{T} = (T_1, \dots, T_m) = \Delta \mathbf{r}(\mathbf{s})$, $\mathbf{S} = (S_1, \dots, S_N) =$

$\mathbf{r}(\mathbf{u}) - \mathbf{Q}$ and consider the constrained problem

$$\begin{aligned} \min_{\{\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}\}} & \sum_{l=1}^m \|T_l\| + \lambda \sum_{k=1}^N \|S_k\| \\ \text{s.t. } & T_l = \Delta \mathbf{r}(s_l), \quad l = 1, 2, \dots, m \\ & S_k = \mathbf{r}(u_k) - Q_k, \quad k = 1, 2, \dots, N \end{aligned}$$

We define the following augmented Lagrangian functional

$$\begin{aligned} L(\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}; \alpha, \beta) &= \|\mathbf{T}\| + \lambda \|\mathbf{S}\| + \langle \beta, \mathbf{S} - (\mathbf{r}(\mathbf{u}) - \mathbf{Q}) \rangle \\ &+ \frac{\rho}{2} \|\mathbf{T} - \Delta \mathbf{r}(\mathbf{s})\|^2 + \langle \alpha, \mathbf{T} - \Delta \mathbf{r}(\mathbf{s}) \rangle \\ &+ \frac{\mu}{2} \|\mathbf{S} - (\mathbf{r}(\mathbf{u}) - \mathbf{Q})\|^2 \end{aligned}$$

where $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ represent the norm and inner product for vector space, α and β are the Lagrangian multipliers. ρ and μ are positive constants, and the fourth and sixth terms on the right hand side are the augmentations.

The augmented Lagrangian method does not solve the constrained problem directly. Instead, it finds a solution to the saddle-point problem:

$$\min_{\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}} \max_{\alpha, \beta} L(\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}; \alpha, \beta)$$

This can be done by solving a series of subproblems which are elaborated below. The whole algorithm is given in Algorithm 5.2

5.3.1 P-sub problem

Let

$$\mathbf{M}_{\{\mathcal{T}, 1\}} = \begin{bmatrix} B_1(u_1), & \cdots, & B_n(u_1) \\ B_1(u_2), & \cdots, & B_n(u_2) \\ \cdots & \cdots, & \cdots \\ B_1(u_N), & \cdots, & B_n(u_N) \end{bmatrix}_{\mathcal{T}}$$

Algorithm 5.2 ALM algorithm for $TL - L^1$ B-spline fitting

Initialization: $\mathbf{P}^{(0)} = \mathbf{S}^{(0)} = \mathbf{T}^{(0)} = 0$; $\alpha^{(0)} = \beta^{(0)} = \epsilon = 0$, the uniform knot vector $\mathcal{T}^{(0)}$ and the specified error threshold ϵ_0 .

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{P}^{(l+1)}, \mathcal{T}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{S}^{(l+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\alpha^{(l)}, \beta^{(l)}$

$$(\mathbf{P}^{(l+1)}, \mathcal{T}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{S}^{(l+1)}) \approx \arg \min_{\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}} L(\mathbf{P}, \mathcal{T}, \mathbf{T}, \mathbf{S}; \alpha^{(l)}, \beta^{(l)})$$

 which is done by the following iterations:

 Let $\mathbf{P}^{(l+1)} = \mathbf{P}^{(l)}, \mathcal{T}^{(l+1)} = \mathcal{T}^{(l)}, \mathbf{T}^{(l+1)} = \mathbf{T}^{(l)}, \mathbf{S}^{(l+1)} = \mathbf{S}^{(l)}$

for $ll=1,2,\dots,K$ **do**

 compute

$$\mathbf{P}^{(l+1)} = \arg \min_{\mathbf{P}} L(\mathbf{P}, \mathbf{T}^{(l+1)}, \mathbf{S}^{(l+1)}, \mathcal{T}^{(l+1)}; \alpha^{(l)}, \beta^{(l)})$$

$$\mathcal{T}^{(l+1)} = \arg \min_{\mathcal{T}} L(\mathbf{P}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{S}^{(l+1)}, \mathcal{T}; \alpha^{(l)}, \beta^{(l)})$$

$$\mathbf{T}^{(l+1)} = \arg \min_{\mathbf{T}} L(\mathbf{P}^{(l+1)}, \mathbf{T}, \mathbf{S}^{(l+1)}, \mathcal{T}^{(l+1)}; \alpha^{(l)}, \beta^{(l)})$$

$$\mathbf{S}^{(l+1)} = \arg \min_{\mathbf{S}} L(\mathbf{P}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{S}, \mathcal{T}^{(l+1)}; \alpha^{(l)}, \beta^{(l)})$$

end for

 and then update the Lagrangian multipliers

$$\alpha^{(l+1)} = \alpha^{(l)} + \rho(\mathbf{T} - \Delta \mathbf{r}(\mathbf{u}))$$

$$\beta^{(l+1)} = \beta^{(l)} + \mu(\mathbf{S} - (\mathbf{r}(\mathbf{u}) - \mathbf{Q}))$$

end while

$$\mathbf{M}_{\{\mathcal{T},2\}} = \begin{bmatrix} B_1(s_1), & \cdots, & B_n(s_1) \\ B_1(s_2), & \cdots, & B_n(s_2) \\ \cdots & \cdots, & \cdots \\ B_1(s_m), & \cdots, & B_n(s_m) \end{bmatrix}_{\mathcal{T}}$$

Then $\mathbf{r}(\mathbf{u}) = \mathbf{M}_{\{\mathcal{T},1\}}\mathbf{P}$; $\mathbf{r}(\mathbf{s}) = \mathbf{M}_{\{\mathcal{T},2\}}\mathbf{P}$.

Hence, given knot vector $\mathcal{T}$, $\mathbf{T}$, $\mathbf{S}$, the $\mathbf{P}$ -sub problem is to solve for $\mathbf{P}$ from

$$\begin{aligned} \min_{\mathbf{P}} \{ & \langle \alpha, \mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T},2\}}\mathbf{P} \rangle + \frac{\rho}{2} \|\mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T},2\}}\mathbf{P}\|^2 \\ & + \langle \beta, \mathbf{S} - \mathbf{M}_{\{\mathcal{T},1\}}\mathbf{P} + \mathbf{Q} \rangle + \frac{\mu}{2} \|\mathbf{S} - \mathbf{M}_{\{\mathcal{T},1\}}\mathbf{P} + \mathbf{Q}\|^2 \} \end{aligned}$$

which is a quadratic minimization problem and can be converted into a sparse linear system.

5.3.2 $\mathcal{T}$ -sub problem

The choice of knot vector has considerable effect on the shape of the curve. This subproblem is to find the best knot vector positions $\mathcal{T}$ for given deBoor points $\mathbf{P}$ and B-spline parameters $\mathbf{u}$.

The B-spline basis functions $\{B_i\}_{i=1}^n$ are the rational functions of the knot vector $\mathbf{u}$. In particular, $B_i(u)$ is the cubic B-spline function associated with the knot quintuples $[t_i]_4 =$

$[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$ and defined explicitly by

$$B[t_i]_4(u) = \begin{cases} \frac{(u-t_{i0})^3}{(t_{i1}-t_{i0})(t_{i2}-t_{i0})(t_{i3}-t_{i0})} & t_{i0} < u \leq t_{i1} \\ \frac{(u-t_{i0})^2(t_{i2}-u)}{(t_{i2}-t_{i1})(t_{i3}-t_{i0})(t_{i2}-t_{i0})} \\ + \frac{(t_{i3}-u)(u-t_{i0})(u-t_{i1})}{(t_{i2}-t_{i1})(t_{i3}-t_{i1})(t_{i3}-t_{i0})} \\ + \frac{(t_{i4}-u)(u-t_{i1})^2}{(t_{i2}-t_{i1})(t_{i4}-t_{i1})(t_{i3}-t_{i1})} & t_{i1} < u \leq t_{i2} \\ \frac{(u-t_{i0})(t_{i3}-u)^2}{(t_{i3}-t_{i2})(t_{i3}-t_{i1})(t_{i3}-t_{i0})} \\ + \frac{(t_{i4}-u)(t_{i3}-u)(u-t_{i1})}{(t_{i3}-t_{i2})(t_{i4}-t_{i1})(t_{i3}-t_{i1})} \\ + \frac{(t_{i4}-u)^2(u-t_{i2})}{(t_{i3}-t_{i2})(t_{i4}-t_{i2})(t_{i4}-t_{i1})} & t_{i2} < u \leq t_{i3} \\ \frac{(t_{i4}-u)^3}{(t_{i4}-t_{i3})(t_{i4}-t_{i2})(t_{i4}-t_{i1})} & t_{i3} < u \leq t_{i4} \\ 0 & u \leq t_{i0} \text{ or } u > t_{i4} \end{cases}$$

Given $\mathbf{P}$, $\mathbf{T}$, $\mathbf{S}$, the $\mathcal{T}$ -sub problem becomes solving

$$\begin{aligned} & \min_{\mathcal{T}} \{ \langle \alpha, \mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P} \rangle + \frac{\rho}{2} \|\mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P}\|^2 \\ & + \langle \beta, \mathbf{S} - \mathbf{M}_{\{\mathcal{T}, 1\}} \mathbf{P} + \mathbf{Q} \rangle + \frac{\mu}{2} \|\mathbf{S} - \mathbf{M}_{\{\mathcal{T}, 1\}} \mathbf{P} + \mathbf{Q}\|^2 \} \end{aligned}$$

Note that here only matrices $\mathbf{M}_{\{\mathcal{T}, 1\}}$ and $\mathbf{M}_{\{\mathcal{T}, 2\}}$ are related to the knot vector $\mathbf{u}$.

After removing the constant, the formulation is simplified into

$$\min_{\mathcal{T}} \left\{ \frac{\rho}{2} \|\Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P} - (\mathbf{T} + \frac{\alpha}{\rho})\|^2 + \frac{\mu}{2} \|\mathbf{M}_{\{\mathcal{T}, 1\}} \mathbf{P} - (\mathbf{Q} + \mathbf{S} + \frac{\beta}{\mu})\|^2 \right\}$$

Let

$$\begin{aligned} f_k &= \sqrt{\frac{\rho}{2}} ((\Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P})_k - (T_k + \frac{\alpha_k}{\rho})) \\ g_k &= \sqrt{\frac{\mu}{2}} ((\mathbf{M}_{\{\mathcal{T}, 1\}} \mathbf{P})_k - (Q_k + S_k + \frac{\beta_k}{\mu})) \end{aligned}$$

and $\mathbf{F}(\mathcal{T}) = (F_1, \dots, F_{m+N})$ where

$$F_k = \begin{cases} f_k & 1 \leq k \leq m \\ g_{k-m} & m+1 \leq k \leq N+m \end{cases}$$

Then we have the following sum of squares:

$$\min_{\mathcal{T}} \|\mathbf{F}\|^2 = \sum_{k=1}^{m+N} \|F_k\|^2 \quad s.t. \quad t_1 \leq t_2 \leq \dots \leq t_{n+4}$$

This is a nonlinear least squares problem. We adapt the Levenberg-Marquardt (LM) algorithm [78], [92] to find the solution. The LM algorithm is an iterative procedure. Starting from an initial guess for $\mathcal{T}$, each iteration step replaces the current $\mathcal{T}$ by a new estimate $\mathcal{T} + \Delta \mathbf{t}$, $\Delta \mathbf{t} = (\Delta t_1, \dots, \Delta t_{n+4})$, where $\Delta \mathbf{t}$ are the solution to a linear system

$$\mathbf{J}^T \mathbf{J} + \gamma \text{diag}(\mathbf{J}^T \mathbf{J}) \Delta \mathbf{t} = -\mathbf{J}^T \mathbf{F}$$

with the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{F}_1}{\partial t_1}, & \dots, & \frac{\partial \mathbf{F}_1}{\partial t_{n+4}} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{F}_{2N}}{\partial t_1}, & \dots, & \frac{\partial \mathbf{F}_{2N}}{\partial t_{n+4}} \end{bmatrix}$$

the diagonal matrix $\text{diag}(\mathbf{J}^T \mathbf{J})$ consisting of the diagonal elements of $\mathbf{J}^T \mathbf{J}$, and the damping factor γ . The damping factor is adjusted at each iteration. If the reduction of the objective function is rapid, a smaller value is used for γ ; Otherwise, if an iteration gives insufficient reduction, γ is increased. Here we simply set the initial estimation $\mathcal{T}$ to be a uniform knot vector. The pseduocode of this optimization is given in Algorithm 5.3, which is adapted from [87].

5.3.3 T-sub problem

Given knot vector $\mathcal{T}$, $\mathbf{P}$, $\mathbf{S}$, the T-sub problem is to solve

$$\min_{\mathbf{T}} \{ \|\mathbf{T}\| + \langle \alpha, \mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P} \rangle + \frac{\rho}{2} \|\mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T}, 2\}} \mathbf{P}\|^2 \}$$

Algorithm 5.3 LM-solver

Require: A vector function $\mathbf{F}(\mathcal{T}) = (F_1, \dots, F_{2N})$ and an initial parameter estimation $\mathcal{T}_0$.

Ensure: Optimal $\mathcal{T}$.

Algorithm:

$v \leftarrow 2; \gamma \leftarrow 10^{-2}; k_{\max} \leftarrow 100;$

$\varepsilon_1 \leftarrow 10^{-15}; \varepsilon_2 \leftarrow 10^{-20};$

$k \leftarrow 0; \mathcal{T} \leftarrow \mathcal{T}_0,$

$\mathbf{A} \leftarrow \mathbf{J}\mathbf{J}^T; \mathbf{G} \leftarrow -\mathbf{J}^T\mathbf{F}$

$stop \leftarrow (\|\mathbf{G}\|_{\infty} < \varepsilon_1)$

while (stop == false) and ($k < k_{\max}$) **do**

$k \leftarrow k + 1$

repeat

 Solve the equation: $(\mathbf{A} + \gamma \text{diag}(\mathbf{J}^T\mathbf{J}))\Delta\mathbf{t}^T = \mathbf{G}$

if ($\|\Delta\mathbf{t}\|^2 < \varepsilon_1 \|\mathcal{T}\|^2$) **then**

$stop \leftarrow true$

else

 Find a positive h such that $\{\mathcal{T} + h(\Delta\mathbf{t})\}$ is monotonic.

$\mathcal{T}_{new} \leftarrow \mathcal{T} + \min\{1, h\}\Delta\mathbf{t},$

$\mathbf{F}_{new} \leftarrow \mathbf{F}(\mathcal{T}_{new})$

$\mathbf{J}_{new} \leftarrow \mathbf{J}(\mathcal{T}_{new})$

$d = \frac{\|\mathbf{F}\|^2 - \|\mathbf{F}_{new}\|^2}{(\Delta\mathbf{t})(\gamma \text{diag}(\mathbf{J}^T\mathbf{J}))\Delta\mathbf{t}^T + \mathbf{G}}$

if $\|\mathbf{F}_{new}\|^2 < \|\mathbf{F}\|^2$ **then**

$\mathcal{T} \leftarrow \mathcal{T}_{new};$

$\mathbf{F} \leftarrow \mathbf{F}_{new}$

$\mathbf{J} \leftarrow \mathbf{J}_{new}$

$\mathbf{A} \leftarrow \mathbf{J}^T\mathbf{J}$

$\mathbf{G} \leftarrow -\mathbf{J}^T\mathbf{F}$

$stop \leftarrow (\|\mathbf{G}\|_{\infty} < \varepsilon_1) \text{ or } (\|\mathbf{F}\|^2 < \varepsilon_2)$

$\gamma \leftarrow \gamma \max\{\frac{1}{3}, 1 - (2d - 1)^3\}$

$v \leftarrow 2$

else

$\gamma \leftarrow \gamma v$

$v \leftarrow 2v$

end if

end if

until ($d > 0$) or ($stop$)

end while

After reformulation, it can be written

$$\min_{\mathbf{T}} \left\{ \sum_{l=1}^m \|T_l\| + \frac{\rho}{2} \sum_{l=1}^m \|T_l - (\Delta \mathbf{M}_{\{\mathcal{T},2\}} \mathbf{P} - \frac{\alpha}{\rho})_l\|^2 + constant \right\}$$

The constant can be removed. The problem is decomposable and it has the following closed form solution:

$$\mathbf{T} = \max\{0, 1 - \frac{1}{\rho \|\mathbf{w}\|}\} \mathbf{w}$$

where $\mathbf{w} = \Delta \mathbf{M}_{\{\mathcal{T},2\}} \mathbf{P} - \frac{\alpha}{\rho}$.

5.3.4 S-sub problem

Given knot vector $\mathcal{T}$, $\mathbf{P}$, $\mathbf{T}$, the S-sub problem is to solve

$$\min_{\mathbf{S}} \{ \|\mathbf{S}\| + \langle \beta, \mathbf{S} - \mathbf{M}_{\{\mathcal{T},1\}} \mathbf{P} + \mathbf{Q} \rangle + \frac{\mu}{2} \|\mathbf{S} - \mathbf{M}_{\{\mathcal{T},1\}} \mathbf{P} + \mathbf{Q}\|^2 \}$$

Similar to the $\mathbf{T}$ -sub problem, it has a closed form solution:

$$\mathbf{S} = \max\{0, 1 - \frac{\lambda}{\mu \|\mathbf{z}\|}\} \mathbf{z}$$

where $\mathbf{z} = \mathbf{M}_{\{\mathcal{T},1\}} \mathbf{P} - \mathbf{Q} - \frac{\beta}{\mu}$.

5.3.5 Update Lagrangian multipliers

$$\alpha = \alpha + \rho(\mathbf{T} - \Delta \mathbf{M}_{\{\mathcal{T},2\}} \mathbf{P})$$

$$\beta = \beta + \mu(\mathbf{S} - (\mathbf{M}_{\{\mathcal{T},1\}} \mathbf{P} - \mathbf{Q}))$$

Figure 5.4: Comparison of $TL - L^1$, $\lambda = 5$ without (a, c) and with (b, d) knot optimization.

5.4 Experiments

This section presents experimental results to demonstrate the performance of our proposed algorithm.

5.4.1 $TL - L^1$ methods

Knot position location

We test the visual and quantitative effect of our knot vector optimization. Fig. 5.4 gives some comparison of our approach with and without knot vector optimization. Our proposed method makes the knot vector locate near the high curvature area to preserve the important features and produces lower fitting error shown in Table 5.1 for both clean (Fig. 5.4(a) and (b)) and noisy (Fig. 5.4 (c) and (d)) contours where we have added outliers (variance = 5% of diagonal of bounding box) and Gaussian noise (variance = 0.5% of diagonal of bounding box).

Smoothness and robustness to outliers

As described above, when the contour contains Gaussian noise and outliers, L^2 norm can not distinguish the outliers and features, especially when many knots are located near the area. Fig. 5.5 shows the comparison of MMSE, MME and our proposed $TL - L^1$ method with the same knot vector. To make the comparison fair, MMSE and MME use the same optimized knot location from the $TL - L^1$ method. MMSE produces the worst

Figure 5.5: Comparison among MMSE, MME and our $TL - L^1$ with the same knot vector $\mathcal{T}$.

result. Although MME can remove most of the outliers nicely, it also cannot guarantee the smoothness of the shape and may cause some wiggles due to the outliers. Our proposed method not only cleans the noise and keeps the overall smoothness but also preserves the features well.

5.4.2 Adaptive $TL - L^1$ B-spline fitting

The adaptive B-spline fitting experimental results are shown in Fig. 5.6, Fig. 5.7, Fig. 5.8. Outliers (variance=5% of the diagonal of the bounding box) and Gaussian noise (variance=0.5% of the diagonal of the bounding box) are added to the extracted contours to validate the robustness of our methods. For easy comparison, the input for MMSE and MME are ordered and parameterized by chord-length methods. Our input shapes are unorganized point sets with the same point positions as MMSE and MME. We can see from the experimental results showed in Fig. 5.6, Fig. 5.7 that the B-spline curves reconstructed by our approach preserve the features better visually, and have lower error quantitatively, see Table 5.1.

For complex shapes as in Fig. 5.6, the initial shape is generated by a quad-tree composition. As we mentioned earlier, our approach is not very sensitive with the initial shape. For simple shapes, the boundary of a bounding box can be used as the initial shape, as showed in Fig. 5.7. We can see from Fig. 5.8 that the different initial shapes (a) and (b)

Figure 5.6: Comparison between $ATL - L^1$ method and MMSE method. (a) is the result by $ATL - L^1$ from the initial shape. (c) is the result by MMSE criterion with given parameter values and uniform knot vector.

produce almost the same B-spline curve.

5.4.3 Error comparison

To illustrate that our proposed method produces smaller error quantitatively, we compute the mean distance $\bar{e}$ and the Hausdorff distance e_H listed in the Table 5.1 from our reconstructed B-spline curve to the groundtruth contour points extracted from image. As we can see, our method outperforms the previous methods.

Figure 5.7: Comparison of the $ATL - L^1$ method, MMSE, MME. (a), (b) and (c), (d) are the results by $ATL - L^1$ from their bounding boxes. (e), (f) and (g), (h) are the results by MMSE criterion with given parameter values and uniform knot vector. (i), (j) and (k), (l) are the results by MME criterion with given parameter values and uniform knot vector.

Figure 5.8: Adaptive B-spline fitting with different initial shapes.

Table 5.1: Statistics of contour-point number N , control point number n , $\bar{e}$, e_H in Fig. 5.4, Fig. 5.5, Fig. 5.6, Fig. 5.7, Fig. 5.8 .

Contour	N	n	$\bar{e}$	e_H
Fig. 5.4(a)	1206	38	2.207	14.313
Fig. 5.4(b)	1206	38	1.307	5.922
Fig. 5.4(c)	2166	40	2.136	17.581
Fig. 5.4(d)	2166	40	1.420	13.417
Fig. 5.5(a)	1093	80	1.771	16.203
Fig. 5.5(b)	1093	80	1.104	12.094
Fig. 5.5(c)	1093	80	0.954	5.921
Fig. 5.7(a)	1714	40	1.314	7.283
Fig. 5.7(e)	1714	40	2.335	12.756
Fig. 5.7(i)	1714	40	2.096	11.909
Fig. 5.7(c)	849	28	0.833	6.208
Fig. 5.7(g)	849	28	2.059	14.574
Fig. 5.7(k)	849	28	1.883	17.831
Fig. 5.8(a)	869	35	1.022	6.342
Fig. 5.8(b)	868	35	1.003	6.276

Chapter 6

TV- L^1 Optimization for B-spline Surface Reconstruction with Sharp Features

This chapter considers the problem of constructing a B-spline surface from point data sampled from an unknown, piecewise smooth surface that is topologically equivalent to a rectangle.

6.1 Motivation and Our work

In computer-aided design and digital geometry processing, surface reconstruction is a fundamental problem and broadly used in a wide range of applications, such as modeling, shape analysis and animation. Generally, the task of surface reconstruction is to construct a mathematical representation that fits a set of data points sampled from a surface. These points may contain various types of noise like Gaussian noise and outliers due to the limited accuracy of data acquisition equipments. There are two lines of research on surface reconstruction, often complementary to each other. One is dedicated to parametric functional representation determined by some explicit parameters and the other is dedicated to implicit functional representation defined by locus of scalar values, both fitting data points to meet some prescribed metric. Among the parametric functional representation,

B-spline surfaces are extensively used due to their flexibility and powerful mathematical properties such as local modification, affine invariance and strong convex hull. In particular, non-uniform rational B-splines (NURBS) are the industry standard for freeform shape representation in CAD. In this chapter, we study B-spline surface reconstruction where the reconstructed B-spline surface may contain sharp or semi-sharp features and the input data points may contain noise and outliers.

Although intensive research on B-spline surface reconstruction has been conducted in last decades, most of the methods address the problem by using prescribed knot placement and finding the optimal control points [17, 72, 140]. They set knots near the sharp features heuristically. The choice of knots has a considerable effect on the shape of the resulting B-spline surface and the reconstruction accuracy, especially in the vicinity of sharp features. Hence, an ideal solution should place the knots according to the geometry variance of data points automatically. That is, knots should be treated as variables [74] and determined by solving some optimization problem, which is known as knot placement. However, the knot placement is a multi-variate and multi-modal nonlinear optimization problem, for which reaching an optimal solution is difficult. Hence, various heuristic approaches are proposed to achieve adaptive knots placement. Examples include curvature based sampling [79, 103], evolutionary intelligence [144, 138, 115], simulated annealing [111], distribution estimation algorithm [161], genetic algorithm [54, 114, 157], and partial swarm optimization [52, 53]. These techniques are often time-consuming and use greedy search strategies, which results in a suboptimal solution that corresponds to local minima. In addition, most of these approaches assume that the underlying B-spline surfaces are smooth. They rarely consider the surfaces containing (semi-)sharp features. Noise often co-exists with sharp features of the shape, which makes the problem even more challenging since it is difficult to automatically distinguish between noise and feature.

Notice that total variation (TV) techniques have been widely used in various image problems [112, 15, 109] and L^1 -norm has the capability of well preserving the sharp feature. We formulate our problem of fitting a B-spline surface to a set of given data points as an L^1 -

norm minimization problem with TV as a smooth term. The formulation treats knots and control points as variables and thus optimizes both simultaneously. The total variation and L^1 -norm enable our method to handle various types of noise and sharp features effectively. The main contributions include a new total variational formulation for B-spline surface reconstruction and an efficient numerical algorithm for solving the minimization problem, which is based on the augmented Lagrangian method that converts the complicate non-linear problem into a set of relatively simple subproblems. These subproblems admit explicit solutions, are converted into a system of linear equations, or are solved by the Levenberg-Marquardt (LM) algorithm.

6.2 Surface Reconstruction Problem

Given a set of data points $\mathbf{Q} = \{Q_{ij} \in \mathbb{R}^3, i = 0, \dots, N; j = 0, \dots, M\}$, we assume that a parameter pair (u_i, v_j) is associated to each point Q_{ij} . Our problem is to find a B-spline surface $\mathbf{r}(u, v)$ of degrees $k \times l$ such that $\mathbf{r}(u_i, v_j)$ is as close to Q_{ij} as possible for all i and j . The B-spline surface is defined by $(m + 1) \times (n + 1)$ control points $\{P_{ij} \in \mathbb{R}^3, i = 0, \dots, m; j = 0, \dots, n\}$ and knot vectors $\mathcal{S} \times \mathcal{T} = \{s_0, s_1, \dots, s_m, \dots, s_{m+k+1}\} \times \{t_0, t_1, \dots, t_n, \dots, t_{n+l+1}\}$. It can be represented by

$$\mathbf{r}(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{i,k}(u) B_{j,l}(v) P_{ij} \quad (6.1)$$

where $B_{i,k}(u)$ and $B_{j,l}(v)$ are the B-spline basis functions defined over knot vectors $\mathcal{S}$ and $\mathcal{T}$, respectively.

Many previous methods use the least squares approach to find the B-spline surface. They usually set up both knot vectors $\mathcal{S}$ and $\mathcal{T}$ in advance, and then the surface can be found by solving

$$\min \sum_{i=0}^M \sum_{j=0}^N \|Q_{ij} - \mathbf{r}(u_i, v_j)\|^2 \quad (6.2)$$

in which only the control points are the unknowns. The advantage of this approach is that the problem is converted to a linear system. However, there are at least two issues with this approach. First, the knots need to be set in advance and they are not determined in the minimization process. Second, the input point clouds usually contain various types of noise. It has been known that L^2 -norm can not work well when the data points possess outliers and sharp features.

In this chapter, we will adopt a more suitable metric for B-spline surface reconstruction so that it can better handle noisy data or outliers and preserve geometric features.

6.3 TV- L^1 Optimization Model for B-spline Surface Reconstruction

Our basic idea is to incorporate knots as unknowns into the optimization model and also add a TV term for smoothing surfaces. During the optimization process, knots are gradually replaced to distribute more near the area with high curvature.

To simplify explanation, we use a B-spline curve to illustrate the idea in Fig. 6.1. We add Gaussian noise and outliers to the ground truth points, which are sampled from a curve containing both flat and corner regions in Fig. 6.1(a). Fig. 6.1(b) shows the reconstructed B-spline curve with control point number $n = 7$ by (6.2) with a uniform knot vector. As mentioned above, it can not produce the corner with C^0 continuity, where a knot with multiplicity 3 is actually needed. In order to recover the corner, we have to increase the number of control points to $n = 35$, which is accompanied by changing the overall smoothness as shown in Fig. 6.1(c). Fig. 6.1(e) is the result by our method with $n = 7$ and $\mathcal{T} = \{0, 0, 0, 0, 2.072, 2.072, 2.072, 4, 4, 4, 4\}$. As a comparison, using our method proposed in this chapter, a knot with multiplicity 3 is automatically generated for the corner. Fig. 6.1(f) is the curve by our method with the same knot vector with Fig. 6.1 (c). It shows that even more knots are placed near the co-existence noise, our TV- L^1

Figure 6.1: (a) is the ground truth, and we add Gaussian noise and outliers to (a). (b) is the curve by (6.2) with $n = 7$. (c) is by (6.2) with $n = 35$. (d) is the B-spline curve in (e). (e) is the curve by our variational method with $n = 7$. (f) is the curve by our method with the same knot vector placement with (c).

approach can also keep the overall smoothness and the sharp features as well.

Now let us describe our variational model and reconstruction algorithm. We use a bicubic B-spline surface ($k = l = 3$) to fit the data points. The method can be easily extended to B-spline surfaces with arbitrary order. Our approach involves two basic steps:

- (1) Parameterize input points $\mathbf{Q}$.
- (2) Solve a TV- L^1 problem for control points $\mathbf{P}$ and knot vector $\mathcal{S} \times \mathcal{T}$.

Any good mesh or point parameterization methods can be used for Step (1). We employ the method proposed in [160]. Thus in the rest of this chapter we focus on Step (2), which is to find $\mathbf{P}$ and $\mathcal{S} \times \mathcal{T}$ by minimizing

$$\min_{\{\mathbf{P}, \mathcal{S} \times \mathcal{T}\}} \int \|\nabla \mathbf{r}(u, v)\| du dv + \lambda \|\mathbf{r}(u, v) - \mathbf{Q}\|$$

In this variational model, the first term of the objective function is the total variation of the B-spline surface measuring the smoothness, and the second term is the fidelity term in L^1 -norm measuring the fitting error. $\nabla \mathbf{r}(u, v)$ is the gradient of B-spline surface defined as $(\frac{\partial \mathbf{r}}{\partial u}, \frac{\partial \mathbf{r}}{\partial v})$. Symbols $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ represent the Euclidean norm and inner product for a vector space.

Discretizing the variational model gives

$$\min_{\{\mathbf{P}, \mathcal{S} \times \mathcal{T}\}} \sum_{i=0}^{m^s} \sum_{j=0}^{n^s} \|\nabla \mathbf{r}(s_i^s, t_j^s)\| + \lambda \sum_{i=0}^M \sum_{j=0}^N \|\mathbf{r}_{ij} - \mathbf{Q}_{ij}\|$$

Here $\{s_0^s, \dots, s_{m^s}^s\} \times \{t_0^s, \dots, t_{n^s}^s\}$ are the sample parameters on the B-spline surface. $\nabla \mathbf{r}(s_i^s, t_j^s)$ stands for the gradient at parameter (s_i^s, t_j^s) . $m^s + 1$ and $n^s + 1$ are the sample numbers in s and t directions. We can further write the discrete TV- L^1 minimization in matrix form

$$\min_{\{\mathbf{P}, \mathcal{S} \times \mathcal{T}\}} \|\nabla \mathbf{M}_{\{2, \mathcal{S} \times \mathcal{T}\}} \mathbf{P}\| + \lambda \|\mathbf{M}_{\{1, \mathcal{S} \times \mathcal{T}\}} \mathbf{P} - \mathbf{Q}\|$$

where $\mathbf{M}_{\{2, \mathcal{S} \times \mathcal{T}\}}$ is the matrix for given sample $\{s_0^s, \dots, s_{m^s}^s\} \times \{t_0^s, \dots, t_{n^s}^s\}$, and $\mathbf{M}_{\{2, \mathcal{S} \times \mathcal{T}\}}$ and $\mathbf{M}_{\{1, \mathcal{S} \times \mathcal{T}\}}$ are the matrices concerned with knot vector $\mathcal{S} \times \mathcal{T}$. For simplicity, we denote them by $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively.

6.4 Numerical algorithm

Similar to previous chapters, we adapt the augmented Lagrangian method to solve the TV- L^1 minimization. For this purpose, we introduce two new variables:

$$\mathbf{T} = \{T_{00}, \dots, T_{0n^s}, \dots, T_{i0}, \dots, T_{ij}, \dots, T_{in^s}, \dots, T_{m^s0}, \dots, T_{m^sn^s}\},$$

$$\mathbf{Z} = \{S_{00}, \dots, S_{0N}, \dots, S_{i0}, \dots, S_{ij}, \dots, S_{iN}, \dots, S_{M0}, \dots, S_{MN}\}$$

where $T_{ij} = \nabla \mathbf{r}_{\{2, ij\}}$ and $S_{ij} = \mathbf{r}_{\{1, ij\}} - \mathbf{Q}_{ij}$, and consider the constrained problem

$$\min_{\{\mathbf{P}, \mathcal{S} \times \mathcal{T}\}} \|\mathbf{T}\| + \lambda \|\mathbf{Z}\|$$

$$\text{subject to } \mathbf{T} = \nabla \mathbf{r}_2$$

$$\mathbf{Z} = \mathbf{r}_1 - \mathbf{Q}$$

We define the following augmented Lagrangian functional

$$L(\mathbf{P}, \mathcal{S} \times \mathcal{T}, \mathbf{T}, \mathbf{Z}) = \|\mathbf{T}\| + \lambda \|\mathbf{Z}\| + \langle \lambda_T, (\mathbf{T} - \nabla \mathbf{r}_2) \rangle + \frac{a_T}{2} \|\mathbf{T} - \nabla \mathbf{r}_2\|^2 \\ + \langle \lambda_Z, \mathbf{Z} - (\mathbf{r}_1 - \mathbf{Q}) \rangle + \frac{a_Z}{2} \|\mathbf{Z} - (\mathbf{r}_1 - \mathbf{Q})\|^2$$

where λ_T and λ_Z are the Lagrangian multipliers. The fourth and sixth terms on the right hand side are augmentations with $a_T > 0$ and $a_Z > 0$.

Then our problem is converted to finding a solution to the saddle-point problem:

$$\min_{\{\mathbf{P}, \mathcal{S} \times \mathcal{T}, \mathbf{T}, \mathbf{Z}\}} \max_{\lambda_T, \lambda_Z} L(\mathbf{P}, \mathcal{S} \times \mathcal{T}, \mathbf{T}, \mathbf{Z}; \lambda_T, \lambda_Z)$$

which can be solved in Algorithm 6.1. Algorithm 6.1 involves a few sub-problems, which are discussed in detail below.

P-sub problem

Given $\mathcal{S} \times \mathcal{T}, \mathbf{T}, \mathbf{Z}$, we want to find $\mathbf{P}$. The problem is simplified into the following quadric problem:

$$\min_{\mathbf{P}} \{ \langle \lambda_T, (\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}) \rangle + \frac{a_T}{2} \|\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}\|^2 \\ + \langle \lambda_Z, \mathbf{Z} - (\mathbf{M}_1 \mathbf{P} - \mathbf{Q}) \rangle + \frac{a_Z}{2} \|\mathbf{Z} - (\mathbf{M}_1 \mathbf{P} - \mathbf{Q})\|^2 \}$$

Then $\mathbf{P}$ can be obtained by solving a linear system.

$\mathcal{S} \times \mathcal{T}$ -sub problem

Given $\mathbf{P}, \mathbf{Z}, \mathbf{T}$, we want to find $\mathcal{S}$ and $\mathcal{T}$ that are the solution to the following problem

$$\min_{\mathcal{S} \times \mathcal{T}} \{ \langle \lambda_T, (\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}) \rangle + \frac{a_T}{2} \|\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}\|^2 \\ + \langle \lambda_Z, \mathbf{Z} - (\mathbf{M}_1 \mathbf{P} - \mathbf{Q}) \rangle + \frac{a_Z}{2} \|\mathbf{Z} - (\mathbf{M}_1 \mathbf{P} - \mathbf{Q})\|^2 \}$$

Algorithm 6.1 ALM algorithm for TV- L^1 B-spline surface fitting

Initialization: $\mathbf{P}^{(0)} = \mathbf{Z}^{(0)} = \mathbf{T}^{(0)} = 0$; $\lambda_T^{(0)} = \lambda_Z^{(0)} = \epsilon = 0$, the uniform knot vector $\mathcal{S}^{(0)} \times \mathcal{T}^{(0)}$ and the specified error threshold ϵ_0 .

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{P}^{(l+1)}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{Z}^{(l+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\lambda_T^{(l)}, \lambda_Z^{(l)}$

$$(\mathbf{P}^{(l+1)}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{Z}^{(l+1)}) \approx \arg \min_{\mathbf{P}, \mathcal{S}, \mathcal{T}, \mathbf{T}, \mathbf{Z}} L(\mathbf{P}, \mathcal{S} \times \mathcal{T}, \mathbf{T}, \mathbf{Z}; \lambda_T^{(l)}, \lambda_Z^{(l)})$$

which is done by the following iterations:

Let $\mathbf{P}^{(l+1)} = \mathbf{P}^{(l)}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)} = \mathcal{S}^{(l)} \times \mathcal{T}^{(l)}, \mathbf{T}^{(l+1)} = \mathbf{T}^{(l)}, \mathbf{Z}^{(l+1)} = \mathbf{Z}^{(l)}$

for $ll=1,2,\dots,K$ **do**

 compute

$$\mathbf{P}^{(l+1)} = \arg \min_{\mathbf{P}} L(\mathbf{P}, \mathbf{T}^{(l+1)}, \mathbf{Z}^{(l+1)}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)}; \lambda_T^{(l)}, \lambda_Z^{(l)})$$

$$\mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)} = \arg \min_{\mathcal{S} \times \mathcal{T}} L(\mathbf{P}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{Z}^{(l+1)}, \mathcal{S} \times \mathcal{T}; \lambda_T^{(l)}, \lambda_Z^{(l)})$$

$$\mathbf{T}^{(l+1)} = \arg \min_{\mathbf{T}} L(\mathbf{P}^{(l+1)}, \mathbf{T}, \mathbf{Z}^{(l+1)}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)}; \lambda_T^{(l)}, \lambda_Z^{(l)})$$

$$\mathbf{Z}^{(l+1)} = \arg \min_{\mathbf{Z}} L(\mathbf{P}^{(l+1)}, \mathbf{T}^{(l+1)}, \mathbf{Z}, \mathcal{S}^{(l+1)} \times \mathcal{T}^{(l+1)}; \lambda_T^{(l)}, \lambda_Z^{(l)})$$

end for

and then update the Lagrangian multipliers

$$\lambda_T^{(l+1)} = \lambda_T^{(l)} + a_T(\mathbf{T} - \nabla \mathbf{r}(u, v))$$

$$\lambda_Z^{(l+1)} = \lambda_Z^{(l)} + a_Z(\mathbf{Z} - (\mathbf{r}(u, v) - \mathbf{Q}))$$

end while

which can be simplified to

$$\min_{\mathcal{S} \times \mathcal{T}} \left\{ \frac{a_T}{2} \left\| \nabla \mathbf{M}_2 \mathbf{P} - \left(\mathbf{T} + \frac{\lambda_T}{a_T} \right) \right\|^2 + \frac{a_Z}{2} \left\| \mathbf{M}_1 \mathbf{P} - \left(\mathbf{Q} + \mathbf{Z} + \frac{\lambda_Z}{a_Z} \right) \right\|^2 \right\}$$

after removing some constant. We further introduce constraints $s_i \leq s_{i+1}$, $t_i \leq t_{i+1}$. This is a typical constrained nonlinear least squares minimization problem, for which the Levenberg-Marquardt (LM) algorithm is very efficient [78], [92]. The LM algorithm uses an effective damping strategy that makes it feasible to converge quickly from a wider range of initial guesses. In particular, the B-spline basis functions are rational functions of the knots. We can compute the partial derivatives of the objective function with respect to the knots, for example, using the approach in [142].

T-sub problem

Given knot vectors $\mathcal{S} \times \mathcal{T}$, $\mathbf{P}$, $\mathbf{Z}$, we now find $\mathbf{T}$ by solving

$$\min_{\mathbf{T}} \left\{ \|\mathbf{T}\| + \langle \lambda_T, (\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}) \rangle + \frac{a_T}{2} \|\mathbf{T} - \nabla \mathbf{M}_2 \mathbf{P}\|^2 \right\}$$

After reformulation, we have

$$\min_{\mathbf{T}} \left\{ \sum_{i=0}^{m^s} \sum_{j=0}^{n^s} \left(\|T_{ij}\| + \frac{a_T}{2} \|T_{ij} - (\nabla \mathbf{M}_2 \mathbf{P} - \frac{\lambda_T}{a_T})_{ij}\|^2 + c \right) \right\}$$

The above problem is decomposable and it has the following closed form solution

$$\mathbf{T} = \max \left\{ 0, 1 - \frac{1}{a_T \|\mathbf{w}_T\|} \right\} \mathbf{w}_T$$

where $\mathbf{w}_T = \nabla \mathbf{M}_2 \mathbf{P} - \frac{\lambda_T}{a_T}$.

Z-sub problem

Given knot vectors $\mathcal{S} \times \mathcal{T}$, $\mathbf{P}$, $\mathbf{T}$, we find $\mathbf{Z}$ satisfying

$$\min_{\mathbf{Z}} \{ \lambda \|\mathbf{Z}\| + \langle \lambda_{\mathbf{Z}}, \mathbf{Z} - (\mathbf{r} - \mathbf{Q}) \rangle + \frac{a_{\mathbf{Z}}}{2} \|\mathbf{Z} - (\mathbf{r} - \mathbf{Q})\|^2 \}$$

Similar to the $\mathbf{T}$ -sub problem, it has a closed form solution:

$$\mathbf{Z} = \max\left\{0, 1 - \frac{\lambda}{a_{\mathbf{Z}} \|\mathbf{w}_{\mathbf{Z}}\|}\right\} \mathbf{w}_{\mathbf{Z}}$$

where $\mathbf{w}_{\mathbf{Z}} = \mathbf{M}_1 \mathbf{P} - \mathbf{Q} - \frac{\lambda_{\mathbf{Z}}}{a_{\mathbf{Z}}}$.

6.5 Experiments

To verify the efficiency and effectiveness of the proposed algorithm, we conduct numerical experiments for both synthetic and realistic examples. Two error functions are used to measure the reconstruction accuracy quantitatively, i.e., the mean error (ME) and the maximal error (MAE) defined as follows:

$$\mathbf{ME} = \frac{\sum_{i=0}^M \sum_{j=0}^N \|Q_{ij} - r(u_i, v_j)\|}{N^Q}$$

$$\mathbf{MAE} = \max_{i,j} \|Q_{ij} - r(u_i, v_j)\|$$

Obviously, ME gives an average evaluation on the disparity of points and reconstructed surface, while MAE judges the best performance for reconstruction.

6.5.1 Three examples

Example 1: A piecewise linear surface. This example is to demonstrate that our method can preserve the sharp feature and bring multiple knots around the sharp edge as showed in

Figure 6.2: Reconstruction of a piecewise linear surface with $\lambda = 100$.

Table 6.1 (Fig. 6.2). Due to *the re-placement of the knot vector* in data fitting, our method can faithfully recover the C^0 continuity in the vicinity near the sharp edge.

Table 6.1: Knot vector in Fig. 6.2

s	0	0	0	0	5	10	10	10	10		
t	0	0	0	0	5	5	5	10	10	10	10

Example 2: Fandisk surface patch. This example shows that our approach can preserve the sharp features (Fig. 6.3). This example, 10×20 control points are used to generate the desired surface.

Example 3: Crescent surface. This surface is defined in the parametric form by the following equations:

$$\begin{cases} x = [2 + \sin(2\pi u) \sin(2\pi v)] \sin(3\pi v) \\ y = [2 + \sin(2\pi u) \sin(2\pi v)] \cos(3\pi v) \\ z = \cos(2\pi u) \sin(2\pi v) + 4v - 2 \end{cases} \quad u, v \in [0, 1]$$

Fig. 6.4 shows the reconstruction result by our approach. It demonstrates that our approach also works well for smooth surfaces though it is designed for surfaces with sharp feature and noise.

Figure 6.3: Reconstruction of Fandisk surface patch with NCP 10×20 and $\lambda = 40$.

Figure 6.4: Reconstruction of a crescent surface with 20×20 control points and $\lambda = 100$.

Figure 6.5: (a) is the input ground truth point data, and we add Gaussian noise and outliers to (a) getting (c). (b) is the result by (6.2), and (d) is the result by our approach, which keeps the overall smoothness and preserves the sharp features as well.

6.5.2 Robustness against noise and sharp feature preservation

Our method can preserve the sharp features for clean input data points as showed in Figs. 6.2, 6.3 and 6.4. We will show the robustness of our approach, when the sharp features co-exist with different types of noise. Figs. 6.5, 6.6 and 6.7 are noisy input with Gaussian noise and outliers. The variances of Gaussian noise and outliers are 0.2% and 1.2% with respect to diagonal length of the bounding box. Fig. 6.5(a) is the ground truth. Figs. 6.5(b) and (d) are the fitting surfaces using the least squares metric [17] and our metric with the same number of control points, respectively. By L^1 norm and relocating knot vector, our approach can keep the overall smoothness and preserve the important sharp corners. The least squares metric can not preserve the smoothness of surfaces that are affected by the outliers. Moreover, it can not produce the sharp edges, since no multiple knots are produced as showed in Fig. 6.5(b). On the contrary, our approach is able to reconstruct surfaces with implied sharp feature by optimizing the knot vector.

Figs. 6.6 and 6.7 are another two reconstruction examples from noisy points with sharp features. The knot vector mapping to 2D is showed in Figs. 6.6(d) and 6.7(d). Figs. 6.8 and 6.9 also give the comparison between [17] and our approach for more complicated CAD models. We can see that multiple knots are produced for the surfaces with sharp features

Table 6.2: Error Comparison. NCP is the number of control points. The first row is the error of [17] and the second row is our error.

	Vertices	NCP	ME	MAE
Fig. 6.5	16626	15×25	2.093×10^{-3} 7.990×10^{-4}	2.324×10^{-2} 1.724×10^{-2}
Fig. 6.6	30401	5×7	1.457×10^{-3} 1.240×10^{-4}	1.257×10^{-2} 1.841×10^{-3}
Fig. 6.7	43970	10×20	3.343×10^{-3} 9.130×10^{-4}	1.875×10^{-2} 1.521×10^{-2}
Fig. 6.8	15996	10×20	5.888×10^{-3} 4.782×10^{-3}	3.387×10^{-2} 3.210×10^{-2}
Fig. 6.9	10552	25×20	4.769×10^{-3} 4.456×10^{-3}	5.765×10^{-2} 4.210×10^{-2}

Figure 6.6: (a) is the noisy input. (b) is the reconstruction surface of [17] with NCP 5×7 . (c) is our proposed method with NCP 5×7 . (d) is the knot location mapping to 2D of (b) and (c). Different colors represent different multiplicity.

and the knots can properly be allocated in the high curvature area.

6.5.3 Error comparison and performance

Our approach exhibits good visual effect as demonstrated in the examples above. To further evaluate the performance of our approach, we give comparison on the quantitative difference. Table 6.2 shows the error comparison between [17] and our approach. Obviously, our methods produce lower error than [17]. Table 6.3 shows the performance of our approach.

Table 6.3: Performance of our approach.

Model	Time (s)	Model	Time (s)	Model	Time (s)
Fig. 6.5	20.90	Fig. 6.6	31.59	Fig. 6.7	203.32
Fig. 6.8	73.36	Fig. 6.9	60.73

Figure 6.7: (a) is the noisy input with Gaussian Noise and outlier. (b) is the reconstructed surface of [17] with NCP 10×20 . (c) is our proposed method with NCP 10×20 . (d) is the knot location mapping to 2D of (b) and (c). Different colors represent different multiplicity.

Figure 6.8: (a) is the reconstructed surface of [17] with NCP 10×20 . (b) is our proposed approach with NCP 10×20 . (c) is the knot location mapping to 2D of (a) and (b). Different colors represent different multiplicity.

Figure 6.9: (a) is the reconstructed surface of [17] with NCP 25×20 . (b) is our proposed approach with NCP 25×20 . (d) is the knot location mapping to 2D of (a) and (b). Different colors represent different multiplicity.

Chapter 7

Feature-aware Subdivision Surface Reconstruction

This chapter considers arbitrary topology surface reconstruction and proposes to use feature-aware subdivision surfaces to fit data.

7.1 Motivation and Our work

In previous chapter we have described a B-spline surface reconstruction algorithm. Due to the topological limitation of B-spline surface representation, the method has difficulty in representing and handling a shape with complicated structure or topology. However, in practice we have many shapes with arbitrary topology. Therefore to efficiently reconstruct surfaces of complicated topology, we propose to use subdivision surfaces as the basic surface representation in reconstruction.

Subdivision surfaces have many nice properties such as compactness, smoothness, arbitrary topology and local control. Various subdivision schemes have been developed. Some of them have been widely used in animation and entertainment industry. For example, subdivision surfaces have been integrated to the MPEG4 standard. While most subdivision schemes are designed to create surfaces smooth everywhere, some special

(sharp) subdivision rules have to be introduced to model sharp features such as corners, darts and creases [63]. As a result, Hoppe et al.’s approach defines a piecewise smooth subdivision surface. Noticing that in the real world objects are rarely infinity sharp, DeRose et al. [34] introduce the notion of semi-sharp feature and propose to perform a few steps of geometry preserving subdivision and then switch to the classic smooth Catmull-Clark subdivision. Thus an integer is introduced to represent how many iterations geometry preserving subdivision will perform, which actually serves to control the sharpness of the shape.

Research of using subdivision surfaces for surface reconstruction has been conducted. It has been observed that using smooth subdivision surfaces to reconstruct the shape containing sharp features may not recover the features very well. Therefore Hoppe et al. [63] propose to use their piecewise smooth subdivision surfaces to fit models with sharp features and achieve very convincing reconstruction results. However, their method has two drawbacks. First, their method needs to detect the sharp features before fitting and thus the reconstruction results depend on the correctness of the pre-detected sharp features. Second, in Hoppe et al.’s subdivision scheme, the classification of edges is actually “binary”: sharp or smooth. This is not effective to model semi-sharp features. Though DeRose et al.’s approach [34] provides a simple way to model semi-sharp features, their method of controlling the iterations of performing geometry preserving subdivision is not convenient for reverse engineering where such sharpness control is to be determined. Ideally, such a sharpness control parameter should be a floating point value.

Different from the prior art, our idea is to introduce some real numbers to control the sharpness of the geometry features, treat both the sharpness parameters and the control points as unknowns and solve for them within a variational framework. Therefore in this chapter we first extend the Loop subdivision scheme and Hoppe et al.’s sharp subdivision rules to introduce real numbers as sharpness parameters to continuously control the geometric features of surfaces. Using the proposed subdivision scheme, we then present a variational algorithm for feature-aware subdivision surface reconstruction from dense

Figure 7.1: Loop subdivision masks. The vertex mask (left) and the edge mask (right).

noisy mesh models. In the variational algorithm, L^1 norm is used to handle different types of noise and outliers in a consistent way. The optimal sharpness parameters and control points are obtained by solving a minimization problem. This approach can be viewed as an extension of the variational B-spline surface reconstruction proposed in the previous chapter to arbitrary topology. The sharpness parameters here work like the knots in B-spline surfaces. They are automatically determined by the variational method. As a result, our algorithm can faithfully recover the sharp and semi-sharp features in the input model. Overall, the contribution of this work includes: (1) a new subdivision scheme with continuous sharpness control, which can conveniently model both sharp and semi-sharp features; (2) a variational, L^1 , feature-aware subdivision reconstruction algorithm which is robust to different types of noise and feature preserving.

7.2 Subdivision with Continuous Sharpness Control

Loop subdivision is a popular subdivision scheme for triangular meshes [86]. The refinement step proceeds by splitting each triangular face into four sub-triangles. The subdivision rules for generation of new edge points and update of old vertex points are shown in Fig. 7.1 where $a(n) = \frac{1}{64}(40 - (3 + 2\cos(\frac{2\pi}{n}))^2)$ for vertex valence n .

In order to introduce sharp features into the Loop subdivision surfaces, new subdivision rules were proposed in [63], in which the tangent plane continuity across sharp edges was

Figure 7.2: Sharp feature masks. The blue line denotes the sharp edges. The red rectangle denotes the new inserted vertex.

relaxed. The masks for a crease edge, a dart and a corner are shown in Fig. 7.2.

Inspired by these works, we introduce a new subdivision scheme with continuous sharpness control. In particular, each edge e in the control mesh is associated with a real number $e.s \in [0, 1]$ called the sharpness. When $e.s = 0$, the subdivision generates a smooth edge just as in the normal Loop subdivision. When $e.s = 1$, it generates a crease edge as in [63]. When $0 < e.s < 1$, it generates a blending effect between smooth and crease.

For this purpose, the new edge point e^v is computed by blending the smooth and sharp rules:

$$e^v = (1 - e.s)e_{smooth}^v + e.se_{crease}^v$$

where e_{smooth}^v and e_{sharp}^v are the edge points generated by smooth rules and crease rules proposed in [63], respectively.

For a vertex v , let $v.n$ denote the number of incident edges with $e.s > 0$. According to $v.n$, the rule for new vertex point can be classified into the following different cases:

- When $v.n < 2$, the vertex is considered to be a smooth one ($v.n = 0$) or a dart ($v.n = 1$). The new vertex point is then computed using the smooth vertex mask.
- When $v.n = 2$, the vertex is considered to be a crease vertex. If the two incident edges corresponding to nonzero sharpness are e_j and e_k , we define the sharpness of

the crease vertex to be $v.s = \frac{e_j.s + e_k.s}{2}$. Then

$$v_{crease}^{new} = (1 - v.s)v_{smooth} + v.sv_{crease}$$

where v_{smooth} and v_{crease} are the vertex points generated by the smooth and crease rules, respectively.

- When $v.n > 2$, the vertex is considered to be a corner vertex. Denote by A_s the set of incident edges corresponding to nonzero sharpness. Then the sharpness of the vertex is defined as $v.s = \frac{\sum_{e_i \in A_s} e_i.s}{v.n}$ and the new vertex point is computed by

$$v_{corner}^{new} = (1 - v.s)\bar{v}_{crease}^{new} + v.sv_{corner}$$

where $\bar{v}_{crease}^{new}$ is the weighted average of the crease vertices and v_{corner} is the corner vertex generated by the mask shown in Fig. 7.2.

After one refinement, each old edge is replaced by two new edges. The sharpness of the new edges is inherited from the old edge.

Following the idea of [34], we also introduce a global integer k besides the edge sharpness into the subdivision process. The k represents the number of iterations of performing the proposed refinement rules. After the k iterations, the subdivision rules are switched to the modified Loop subdivision of [63]. The advantage of this approach is that the nice property of the modified Loop subdivision surface is guaranteed mathematically while the visual semi-sharp features are presented.

With this proposed feature-aware subdivision scheme, we can easily create fillets with various curvature radii. Fig. 7.3 shows an example where different shapes can be generated from the shape initial control mesh but with different values for edge sharpness.

Figure 7.3: Examples of various edge sharpness. The control mesh is drawn in wireframe. From left to right: $e.s = 0$ (gray), 0.6 (yellow), 0.9 (magenta), 1 (red).

7.3 Variational Subdivision Surface Reconstruction

Subdivision surface reconstruction aims to construct a subdivision surface to recover the underlying surface including its geometric features, from which the input points are sampled. This is actually a very challenging problem. The challenge lies in several aspects. First, the problem by nature is an inverse problem. Both the geometry and topology of the underlying surface is unknown. Second, to define a subdivision surface, we have to specify at least the control points and the connectivity. The determination of the control points and connectivity involves two different types of problems. The former is a common optimization problem and the latter is a combinatorial problem. Third, the underlying surface may contain sharp or semi-sharp features. How to automatically reflect those features in the optimization is nontrivial. Fourth, the input data usually contain noise and outliers. In particular, the co-existence of noise/outliers and sharp features makes the problem even more difficult.

This section presents an engineering solution to the problem. It consists of a few steps outlined below.

1. If the input is just a set of discrete points without connectivity information, we use some existing software package to convert them into a triangular mesh. If the input data is already a triangular mesh, then this phase is not needed.

2. The input mesh or the generate dense mesh is denoted by S^0 with point data set $\mathbf{v} = \{v_1, v_2, \dots, v_n\}$. It may contain noise and outliers. This step removes the noise while keeping the sharp features. Our variational denoising method presented in Chapter 4 is then used.
3. The smoothed triangular mesh is simplified into a coarse one while preserving the sharp features. The QEM method [55] and its feature preserving variant is used to get the coarse mesh denoted by S_c^0 . Assume that S_c^0 has point set $\mathbf{p} = \{p_1, p_2, \dots, p_m\}$.
4. We treat S_c^0 as the initial control mesh for the constructed subdivision surface defined by the new subdivision rules defined in Section 7.2. Now we fix the connectivity of S_c^0 and try to optimize the position of the control points and the edge sharpness to achieve the best fitting to mesh S^0 .

In the rest of this section, we focus on Step 4, which is a key component of the surface reconstruction. A solution to it can also be incorporated into existing subdivision surface reconstruction algorithms to enhance their performance.

7.3.1 Variational model

The control points $\mathbf{p}$ and sharpness $\mathbf{h} = \{h_1, \dots, h_{n_e}\}$ for the edges, where n_e is the number of the control mesh edges, are sought so as to satisfy

$$\min_{\mathbf{p}, \mathbf{h}} E(\mathbf{p}, \mathbf{h}) = E_s(\mathbf{p}) + \lambda E_f(\mathbf{p}, \mathbf{h})$$

where the energy function includes two terms, the smoothing term and the error (fidelity) term.

Smoothing term

To prevent the control mesh from self-intersection and the influence of the noise, we use the following term for smoothing:

$$E_s(\mathbf{p}) = \sum_{i=1}^m \|L(p_i)\| = \mathbf{L}\mathbf{p}$$

where $L(p_i)$ is the discrete Laplacian. $\mathbf{L}$ is the Laplacian matrix. We call $\sum_{i=1}^m \|L(p_i)\|$ total Laplacian term. L^1 norm is used instead of L^2 to discourage the crease and corner from being smoothed out and being blended, as shown in Fig. 7.4.

Figure 7.4: Comparison between smoothing term using L^2 norm (left) and using total Laplacian (right). Close-up views are shown on the bottom.

Error term

To compute the error, we project the data points q_i onto the subdivision surface S . This is not feasible in practice as the surface is defined as the limit of a set of refined mesh sequences. Hence, we use a piecewise linear approximation $\tilde{S}$ for S instead. $\tilde{S}$ is obtained by subdividing the initial control mesh k times to produce a refined mesh S^k by our feature-aware subdivision masks and then pulling all the vertices of S^k to the limit positions using

limit position masks of [63]. Hence, $\tilde{S}$ can be written as the affine combination of the initial positions $\mathbf{p}$.

To make our fitting algorithm independent from the results of smoothing approach, we use the points of input noisy mesh S^0 instead of the smoothed clean mesh obtained from Step 2. To handle the possible existence of different types of noises, we use L^1 norm to measure the distance. For each data point $v_i \in S^0$, we denote the closest point on the approximated surface $\tilde{S}$ by w_i . w_i can also be represented by the linear combination of vertices on $\tilde{S}$. Then the error term can be represented as follows:

$$E_f(\mathbf{p}, \mathbf{h}) = \|\mathbf{v} - M(\mathbf{h})\mathbf{p}\|$$

where $M(\mathbf{h})$ is the matrix depending on $\mathbf{h}$. For simplicity, we denote $M(\mathbf{h})$ by $\mathbf{M}$. $\mathbf{Mp}$ is the foot point vector of $\mathbf{v}$ on $\tilde{S}$.

Combining both terms, we derive our variational optimization model:

$$\min_{\mathbf{p}, \mathbf{h}} \{ \|\mathbf{Lp}\| + \lambda \|\mathbf{v} - \mathbf{Mp}\| \}$$

7.3.2 Numerical solver

To solve the variational optimization model, we let $\mathbf{q} = \mathbf{Lp}$ and $\mathbf{z} = \mathbf{v} - \mathbf{Mp}$. Our variational optimization becomes a constrained problem:

$$\begin{aligned} \min_{\mathbf{p}, \mathbf{h}} \quad & \|\mathbf{q}\| + \lambda \|\mathbf{z}\| \\ \text{subject to} \quad & \mathbf{q} = \mathbf{Lp} \\ & \mathbf{z} = \mathbf{v} - \mathbf{Mp} \end{aligned}$$

Construct the following augmented Lagrangian functional

$$\begin{aligned} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q, \lambda_z) = & \|\mathbf{q}\| + \lambda \|\mathbf{z}\| + \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle \\ & + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2 + \langle \lambda_z, \mathbf{z} - (\mathbf{v} - \mathbf{Mp}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{Mp})\|^2 \end{aligned}$$

where λ_q and λ_z are the Lagrangian multipliers, $\langle, \rangle$ is the vector dot product operator, and a_q and a_z are two positive numbers. To solve the above constrained problem, we seek a solution to the saddle-point problem:

$$\min_{\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}} \max_{\lambda_q, \lambda_z} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q, \lambda_z)$$

which is achieved in Algorithm 7.1. Algorithm 7.1 involves solving a few sub-problems. The details are given below.

Algorithm 7.1 ALM algorithm for subdivision surface fitting

Initialization: Preprocessing to get $\mathbf{p}^{(0)}$; Let $\mathbf{h}^{(0)} = \mathbf{z}^{(0)} = \mathbf{q}^{(0)} = 0$; $\lambda_q^{(0)} = \lambda_z^{(0)} = \epsilon = 0$, and the specified error threshold ϵ_0 .

while $\epsilon < \epsilon_0$ **do**

 Compute $(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)})$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrangian multipliers $\lambda_q^{(l)}, \lambda_z^{(l)}$

$$(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}) \approx \arg \min_{\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}} L(\mathbf{p}, \mathbf{h}, \mathbf{q}, \mathbf{z}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

 which is done by the following iterations:

 Let $\mathbf{p}^{(l+1)} = \mathbf{p}^{(l)}, \mathbf{h}^{(l+1)} = \mathbf{h}^{(l)}, \mathbf{q}^{(l+1)} = \mathbf{q}^{(l)}, \mathbf{z}^{(l+1)} = \mathbf{z}^{(l)}$

for $ll=1,2,...,K$ **do**

 compute

$$\mathbf{p}^{(l+1)} = \arg \min_{\mathbf{p}} L(\mathbf{p}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{h}^{(l+1)} = \arg \min_{\mathbf{h}} L(\mathbf{p}^{(l+1)}, \mathbf{h}, \mathbf{q}^{(l+1)}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{q}^{(l+1)} = \arg \min_{\mathbf{q}} L(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}, \mathbf{z}^{(l+1)}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

$$\mathbf{z}^{(l+1)} = \arg \min_{\mathbf{z}} L(\mathbf{p}^{(l+1)}, \mathbf{h}^{(l+1)}, \mathbf{q}^{(l+1)}, \mathbf{z}; \lambda_q^{(l)}, \lambda_z^{(l)})$$

end for

 and then update the Lagrangian multipliers

$$\lambda_q^{(l+1)} = \lambda_q^{(l)} + a_q(\mathbf{q} - \mathbf{Lp})$$

$$\lambda_z^{(l+1)} = \lambda_z^{(l)} + a_z(\mathbf{z} - (\mathbf{v} - \mathbf{Mp}))$$

end while

1. **p-sub problem:** Fixing $\mathbf{h}, \mathbf{q}, \mathbf{z}$, we solve the following quadratic equation

$$\begin{aligned} \min_{\mathbf{p}} & \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2 \\ & + \langle \lambda_z, \mathbf{z} - (\mathbf{v} - \mathbf{Mp}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{Mp})\|^2. \end{aligned}$$

This can be converted into a linear system.

2. **h-sub problem:** Fixing $\mathbf{p}, \mathbf{q}, \mathbf{z}$, we find the sharpness $\mathbf{h}$ which only affects the subdivision rule. Hence, the **h-sub problem** is

$$\min_{\mathbf{h}} \langle \lambda_z, \mathbf{z} - (\mathbf{v} - M(\mathbf{h})\mathbf{p}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - M(\mathbf{h})\mathbf{p})\|^2$$

To solve this minimization problem, we adopt particle swarm optimization (PSO) [67]. PSO is an evolutionary optimization method starting from many random initialization seeds. At each iteration a set of candidate solutions is sought, the score of each solution is calculated, and the solutions are updated in the search space by shifting them toward the best current solution.

3. **q-sub problem:** Fixing $\mathbf{p}, \mathbf{h}, \mathbf{z}$, we find $\mathbf{q}$ by solving

$$\min_{\mathbf{q}} \{ \|\mathbf{q}\| + \langle \lambda_q, \mathbf{q} - \mathbf{Lp} \rangle + \frac{a_q}{2} \|\mathbf{q} - \mathbf{Lp}\|^2 \}$$

After reformulation, we have

$$\min_{\mathbf{q}} \left\{ \sum_{i=0}^m \left(\|q_i\| + \frac{a_q}{2} \|q_i - (\mathbf{Lp} - \frac{\lambda_q}{a_q})_i\|^2 + c \right) \right\}$$

This problem is decomposable and it has a closed form solution

$$\mathbf{q} = \max \left\{ 0, 1 - \frac{1}{a_q \|w_q\|} \right\} w_q$$

where $w_p = \mathbf{Lp} - \frac{\lambda_q}{a_q}$.

4. **z**-sub problem: Fixing **p**, **h**, **q**, we find **z** by solving

$$\min_{\mathbf{z}} \{ \lambda \|\mathbf{z}\| + \langle \lambda_z, \mathbf{z} - (\mathbf{v} - \mathbf{M}\mathbf{p}) \rangle + \frac{a_z}{2} \|\mathbf{z} - (\mathbf{v} - \mathbf{M}\mathbf{p})\|^2 \}$$

which has a closed form solution:

$$\mathbf{z} = \max\{0, 1 - \frac{\lambda}{a_z \|\mathbf{w}_z\|}\} \mathbf{w}_z$$

where $\mathbf{w}_z = \mathbf{v} - \mathbf{M}\mathbf{p} - \frac{\lambda_z}{a_z}$.

7.4 Experiments

We first show the effect of the sharpness in reconstruction. For this purpose, we fix the positions of the control mesh and optimize the sharpness. Fig. 7.5 shows such an example, where the sharpness is obtained by solving the optimization problem. The edges whose sharpness is non-zero are marked in yellow. We can see that the reconstruction with optimized sharpness can achieve better reconstruction. It shows that the sharpness provides additional degrees of freedom for reconstruction.

Figure 7.5: The left is the input initial control mesh and the subdivision surface generated by smooth rule. The middle is the objective dense mesh. The right is the subdivision surface by optimizing the sharpness of the control mesh. The root mean error is $\text{RME} = 0.115403$ (left) and $\text{RME} = 0.00768$ (right). The edge with non-zero sharpness are marked in yellow.

Fig. 7.6 is another example, where we add two types of noises: outliers with variance 0.5 of the mean edge length and Gaussian noise with variance 0.1 of the mean edge length, to generate a noisy mesh as shown in Fig. 7.6(a). Fig. 7.6(b) is the mesh after denoising using our L^1 smoothing method. (c) is an initial mesh obtained by simplification from (b). Using this initial mesh to find the optimal position and sharpness, we get the subdivision surface shown in Fig. 7.6(f). Fig. 7.6(d) is the reconstructed subdivision surface by classic Loop subdivision surfaces. The reconstructed surface is much smoother and can not recover the high curvature area well without increasing the density of the control point near the area. Fig. 7.6(e) is the subdivision surface reconstructed by the approach in [63]. In [63], the crease edges are tagged as sharp by computing the dihedral angle of the faces incident to the edge. It can recover the sharp edges but it highly depends on the process of edge detection. Even though the edges are correctly detected, the previous method only handles sharp features. Since we tune the sharpness continuously, our method can handle both sharp and semi-sharp features well.

Two comparisons are given in Figs. 7.7 and 7.8. The original models and the reconstructed surfaces generated by smooth rules, piecewise smooth rules and our methods are presented. In particular, Fig. 7.7 indicates that when the sharp edges are detected incorrectly, piecewise smooth subdivision rules of [63] may yield sharp feature edges near the original smooth area. We can see that both the smooth subdivision schemes and the sharp subdivision scheme [63] can not handle semi-sharp edges well.

Table 7.1 gives statistics of number of vertices, number of control point (NCP), the max error (ME), the root mean error (RME) of above examples. Obviously, our method produces lower errors. The performance of our algorithm is also showed in the Table 7.2, which takes several minutes. We have found that most of time is taken by solving the sharpness sub-problem. The method reconstructed without sharpness takes less than second for examples in the paper since it does not consider the feature and only need to solve one least square minimization problem. The method reconstructed by rules of [63] is not given, since it should detect the sharp edge first.

Figure 7.6: RockerArm model. (a) is the noisy dense mesh. (b) is the mesh after denoising. (c) is the initial mesh obtained by simplification. (d) is the reconstructed subdivision surface using classic Loop subdivision. (e) is the subdivision surface using the rules in [63]. (f) is the reconstructed subdivision surface using our method. The third row shows the close-up views of the subdivision surfaces in Row 2.

Table 7.1: Error comparison. Statistics of number of vertices, number of control point (NCP), the max error (ME), the root mean error (RME).

		Vertices	NCP	MAE	RME
RockerArm	(d)	70400	275	2.9030×10^{-2}	5.4469×10^{-3}
	(e)			1.9094×10^{-2}	2.1700×10^{-3}
	(f)			1.0011×10^{-2}	1.0385×10^{-3}
Vase	(b)	64000	252	5.4290×10^{-2}	6.6434×10^{-3}
	(c)			2.1082×10^{-2}	2.3517×10^{-3}
	(d)			7.5719×10^{-3}	3.7550×10^{-4}
Blade	(b)	51200	202	5.6106×10^{-2}	8.4150×10^{-3}
	(c)			1.5792×10^{-2}	2.2141×10^{-3}
	(d)			6.6826×10^{-3}	7.6794×10^{-4}

Figure 7.7: Vase model. From left to right: original model, subdivision surface reconstructed without sharpness, subdivision surface reconstructed using the rules of [63], and subdivision surface reconstructed using our method.

Figure 7.8: Blade model. From left to right: original model, subdivision surface reconstructed without sharpness, subdivision surface reconstructed using the rules of [63], and subdivision surface reconstructed using our method.

Table 7.2: Performance of our approach.

Model	Time (min)	Model	Time (min)	Model	Time (min)
RockerArm	20.3	Vase	18.6	Blade	13.8

Chapter 8

Conclusions and Future Work

8.1 Summary

The research reported in this thesis has investigated variational techniques with sparse norm for geometry processing, aiming to gain insights of variational theory and to present efficient algorithms for geometry processing. Three types of geometry processing problems have been studied. Novel algorithms have been developed, which accomplishes the goal of the research.

First, we have presented a variational formulation for structure-texture decomposition of color attributes of a triangular mesh. In this formulation, the function domain is fixed to be the triangular mesh, the regularization term is modeled as the total variation on the triangular mesh, and the fidelity term is modeled by one of L^2 , L^1 and G norms based on the properties of texture. The total variation energy is used for extracting the structural part. The fidelity term is responsible for the textural part. While L^2 and G norms are used for texture of no a prior knowledge or oscillating pattern, L^1 norm is used for structural or sparse texture. An efficient augmented Lagrangian method is developed. Two applications: detail enhancement and impulsive noise removal are also presented. The experimental results demonstrate the effectiveness of the proposed decomposition.

Second, we have presented a robust and feature preserving triangular mesh denoising

algorithm by adapting the ROF variational model. The L^1 norm is used to define the regularization term. The use of L^1 norm improves the over-penalty issue compared to L^0 norm which is a good choice for piecewise constant surface, but not necessarily for piecewise smooth surface. The experimental results show that our variational approach has advantages for preserving sharp features and small scale geometry.

Third, we have also extended L^1 variational techniques to B-spline curves and surfaces. A new metric based total Laplacian and L^1 norm is introduced for B-spline curve representation. The total Laplacian is defined for better feature preservation and the L^1 norm is adapted for different types of noise. As for tensor product B-spline surfaces, a total variation is used. The automatical optimization of knots and the use of L^1 and total variation or total Laplacian enable our reconstruction algorithms to faithfully reconstruct the sharp features as well as smooth surfaces. Moreover, an efficient augmented Lagrangian method is realized using the LM-algorithm.

Finally, since B-spline surfaces have topological restriction for surface representation, we have developed a variational subdivision surface reconstruction from noisy dense mesh. Subdivision surfaces are able to model shapes of arbitrary topology. A new feature-aware subdivision scheme is designed to better represent surfaces with various geometric features such as semi-sharp crease, crease, and corners. The key idea here is introducing a real number as the sharpness control for each edge. The function domain is determined by both the control mesh and the sharpness of edges. According to the new subdivision rules, a variational model based on L^1 norm aimed to find the best control mesh and the edge sharpness is proposed. It can handle noisy mesh with different types of noises.

In summary, we have investigated a variational framework for geometry processing and adapted it to various geometric models: triangular meshes, B-spline curves, B-spline surfaces, and subdivision surfaces. Different quantities such as total variation, and total Laplacian are designed, and different norms such as L^1 norm, L^2 norm, and G norm are applied for the regularization term or the fidelity term. New algorithms including image decomposition on manifold, mesh denoising, B-spline curve and surface reconstruction,

and subdivision surface reconstruction validate the potential of the variational techniques.

8.2 Future Work

This research has also brought some topics for further studies:

- To solve the proposed L^1 variational minimization problems, we have adapted the augmented Lagrangian method which proceeds by iteratively solving a few subproblems. Some of the subproblems have closed form solutions or can be converted into solving a linear system, but there are still subproblems that need sophisticated solvers. We have proposed to use the Levenberg-Marquardt algorithm or Particle Swarm Optimization according to the characteristic of the subproblems. While these approaches are heuristic, they provide a good foundation for future work in designing optimal algorithms, which is of both theoretical and practical interest.
- We have developed a new approach for B-spline surface fitting, in which both knots and control points of a B-spline surface are optimized within the same framework. However, the tensor-product structure of a B-spline surface prevents the knots from changing locally. T-splines generalize tensor-product B-spline surfaces by allowing the existence of partial rows or columns of control points [119], [118]. Compared to the control mesh of a B-spline surface, the structure of a T-mesh is more flexible and provides more degrees of freedom. An interesting research topic is thus to extend our variational methods to T-splines for surface reconstruction.
- We have shown that the ROF variational model with sparse norm has advantages in solving the problems of image decomposition on manifolds, mesh denoising, and freeform curve and surface construction. In particular, it makes the respective algorithms well handle geometry features and be insensitive to noise or outliers. Note that there is a wide range of problems in digital geometry processing. It is

interesting to investigate the use of the proposed variational framework in other geometry processes. It is also worthwhile to study the benefits of L^1 and G norms in geometry applications.

References

- [1] [Online]. Available: <http://software.intel.com/en-us/intel-mkl>
- [2] E. Ataer-Cansizoglu, E. Bas, J. Kalpathy-Cramer, G. C. Sharp, and D. Erdogmus, “Contour-based shape representation using principal curves,” *Pattern Recognition*, vol. 46, no. 4, pp. 1140 – 1150, 2013.
- [3] F. Attneave, “Some informational aspects of visual perception,” *Psychological Review*, vol. 61, no. 3, pp. 183–193, 1954.
- [4] J.-F. Aujol, G. Aubert, L. Blanc-Fraud, and A. Chambolle, “Image decomposition into a bounded variation component and an oscillating component,” *Journal of Mathematical Imaging and Vision*, vol. 22, pp. 71–88, 2005.
- [5] J.-F. Aujol and A. Chambolle, “Dual norms and image decomposition models,” *Int. J. Comput. Vision*, vol. 63, pp. 85–104, 2005.
- [6] J.-F. Aujol and T. F. Chan, “Combining geometrical and textured information to perform image classification,” *J. Visual Communication and Image Representation*, vol. 17, no. 5, pp. 1004–1023, 2006.
- [7] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, “Structure-texture image decomposition–modeling, algorithms, and parameter selection,” *International Journal of Computer Vision*, vol. 67, pp. 111–136, 2006.

-
- [8] H. Avron, A. Sharf, C. Greif, and D. Cohen-Or, “ l^1 -sparse reconstruction of sharp point set surfaces,” *ACM Trans. Graph.*, vol. 29, pp. 135:1–135:12, 2010.
- [9] C. L. Bajaj and G. Xu, “Anisotropic diffusion of surfaces and functions on surfaces,” *ACM Trans. Graph.*, vol. 22, no. 1, pp. 4–32, 2003.
- [10] D. Barash, “A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 844–847, 2002.
- [11] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, 2002.
- [12] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *Image Processing, IEEE Transactions on*, vol. 12, pp. 882 – 889, 2003.
- [13] M. Bertalmío, L.-T. Cheng, S. Osher, and G. Sapiro, “Variational problems and partial differential equations on implicit surfaces,” *J. Comput. Phys.*, vol. 174, pp. 759–780, 2001.
- [14] A. Blake and M. Isard, *Active Contours*. Springer London, 1998.
- [15] P. Blomgren and T. Chan, “Color TV: total variation methods for restoration of vector-valued images,” *Image Processing, IEEE Transactions on*, vol. 7, pp. 304–309, 1998.
- [16] C. D. Boor, *A practical Guide to Splines*. Springer-Verlag, 1978.
- [17] C. F. Borges and T. Pastva, “Total least squares fitting of Bézier and B-spline curves to ordered data,” *Computer Aided Geometric Design*, vol. 19, no. 4, pp. 275 – 289, 2002.

-
- [18] E. F. Can and P. Duygulu, “A line-based representation for matching words in historical manuscripts,” *Pattern Recogn. Lett.*, vol. 32, pp. 1126–1138, 2011.
- [19] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489 – 509, 2006.
- [20] E. Candes and T. Tao, “Decoding by linear programming,” *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [21] E. J. Candès and F. Guo, “New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction,” *Signal Processing*, vol. 82, pp. 1519–1543, 2002.
- [22] A. Carmona-Poyato, F. Madrid-Cuevas, R. Medina-Carnicer, and R. M. noz Salinas, “Polygonal approximation of digital planar curves through break point suppression,” *Pattern Recognition*, vol. 43, no. 1, pp. 14 – 25, 2010.
- [23] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350 – 355, 1978.
- [24] A. Chambolle, “An algorithm for total variation minimization and applications,” *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004.
- [25] A. Chambolle and P.-L. Lions, “Image recovery via total variation minimization and related problems,” *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, 1997.
- [26] T. F. Chan and S. Esedoglu, “Aspects of total variation regularized L^1 function approximation,” *SIAM J. Appl. Math.*, vol. 65, pp. 1817–1837, 2005.
- [27] T. F. Chan, G. H. Golub, and P. Mulet, “A nonlinear primal-dual method for total variation-based image restoration,” *SIAM J. Sci. Comput.*, vol. 20, pp. 1964–1977, 1999.

-
- [28] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.
- [29] K. S. Cheng, W. Wang, H. Qin, K. Y. Wong, H. Yang, and Y. Liu, “Design and analysis of optimization methods for subdivision surface fitting,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 878–890, 2007.
- [30] U. Clarenz, U. Diewald, and M. Rumpf, “Anisotropic geometric diffusion in surface processing,” in *Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)*, ser. VISUALIZATION ’00. Washington, DC, USA: IEEE Computer Society, 2000.
- [31] F. Cohen, Z. Huang, and Z. Yang, “Invariant matching and identification of curves using B-splines curve representation,” *Image Processing, IEEE Transactions on*, vol. 4, no. 1, pp. 1–10, 1995.
- [32] F. Cohen and J.-Y. Wang, “Part I: Modeling image curves using invariant 3-D object curve models-a path to 3-D recognition and shape estimation from image contours,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 1, pp. 1–12, 1994.
- [33] A. Delaunoy, K. Fundana, E. Prados, and A. Heyden, “Convex multi-region segmentation on manifolds,” in *The 12th IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 662–669.
- [34] T. DeRose, M. Kass, and T. Truong, “Subdivision surfaces in character animation,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’98. ACM, 1998, pp. 85–94.
- [35] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, “Implicit fairing of irregular meshes using diffusion and curvature flow,” in *Proceedings of the 26th annual*

- conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '99, 1999, pp. 317–324.
- [36] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, “Anisotropic feature-preserving denoising of height fields and bivariate data,” in *In Graphics Interface*, 2000, pp. 145–152.
- [37] Y. Dong, M. Hintermüller, and M. Neri, “An efficient primal-dual method for L1 TV image restoration,” *SIAM J. Img. Sci.*, vol. 2, no. 4, pp. 1168–1189, 2009.
- [38] D. L. Donoho, “De-noising by soft-thresholding,” *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.
- [39] D. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April.
- [40] D. Donoho, M. Elad, and V. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 6 – 18, 2006.
- [41] D. Doo and M. Sabin, “Behaviour of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, vol. 10, no. 6, pp. 356 – 360, 1978.
- [42] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.
- [43] E. Eisemann and F. Durand, “Flash photography enhancement via intrinsic relighting,” *ACM Trans. Graph.*, vol. 23, pp. 673–678, 2004.
- [44] A. F. El Ouafdi, D. Ziou, and H. Krim, “A smart stochastic approach for manifolds smoothing,” in *Proceedings of the Symposium on Geometry Processing*, ser. SGP '08, 2008, pp. 1357–1364.

-
- [45] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge University Press, 2012.
- [46] M. Elsey and S. Esedoglu, “Analogue of the total variation denoising model in the context of geometry processing.” *Multiscale Modeling & Simulation*, vol. 7, no. 4, pp. 1549–1573, 2009.
- [47] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, “Edge-preserving decompositions for multi-scale tone and detail manipulation,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–67:10, 2008.
- [48] M. A. Figueiredo, J. M. Leitaó, and A. K. Jain, “Unsupervised contour representation and estimation using B-splines and a minimum description length criterion,” *Image Processing, IEEE Transactions on*, vol. 9, no. 6, pp. 1075–1087, 2000.
- [49] S. Fleishman, I. Drori, and D. Cohen-Or, “Bilateral mesh denoising,” *ACM Trans. Graph.*, vol. 22, pp. 950–953, 2003.
- [50] M. S. Floater, K. Hormann, and G. Ks, “A general construction of barycentric coordinates over convex polygons,” in *Advances in Computational Mathematics*, 2006, pp. 311–331.
- [51] H. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, “Efficient minimization methods of mixed L2-L1 and L1-L1 norms for image restoration,” *SIAM J. Sci. Comput.*, vol. 27, no. 6, pp. 1881–1902, 2005.
- [52] A. Gálvez and A. Iglesias, “Efficient particle swarm optimization approach for data fitting with free knot B-splines,” *Computer-Aided Design*, vol. 43, no. 12, pp. 1683 – 1692, 2011.
- [53] A. Gálvez and A. Iglesias, “Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3d data points,” *Information Sciences*, vol. 192, no. 0, pp. 174 – 192, 2012.

-
- [54] A. Gálvez, A. Iglesias, and J. Puig-Pey, “Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction,” *Information Sciences*, vol. 182, no. 1, pp. 56 – 76, 2012.
- [55] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’97, 1997, pp. 209–216.
- [56] R. Goldman and T. Lyche, *Knot insertion and deletion algorithms for B-spline curves and surfaces*, R. Goldman and T. Lyche, Eds. SIAM, 1993.
- [57] T. Goldstein and S. Osher, “The Split Bregman Method for L^1 -regularized problems,” *SIAM J. Img. Sci.*, vol. 2, pp. 323–343, 2009.
- [58] X. Gu and S.-T. Yau, “Global conformal surface parameterization,” in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ser. SGP ’03. Switzerland: Eurographics Association, 2003, pp. 127–137.
- [59] Y.-H. Gu and T. Tjahjadi, “Coarse-to-fine planar object identification using invariant curve features and B-spline modeling,” *Pattern Recognition*, vol. 33, no. 9, pp. 1411 – 1422, 2000.
- [60] A. Habib and J. Warren, “Edge and vertex insertion for a class of c1 subdivision surfaces,” *Comput. Aided Geom. Des.*, vol. 16, no. 4, pp. 223–247, 1999.
- [61] L. He and S. Schaefer, “Mesh denoising via L_0 minimization,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 64:1–64:8, 2013.
- [62] K. Hildebrandt and K. Polthier, “Anisotropic filtering of non-linear surface features,” *Computer Graphics Forum*, vol. 23, pp. 391–400, 2004.
- [63] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, “Piecewise smooth surface reconstruction,” in *Proceedings of*

- the 21st annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '94. ACM, 1994, pp. 295–302.
- [64] J. Hoschek, “Intrinsic parametrization for approximation,” *Comput. Aided Geom. Des.*, vol. 5, no. 1, pp. 27–31, 1988.
- [65] T. R. Jones, F. Durand, and M. Desbrun, “Non-iterative, feature-preserving mesh smoothing,” *ACM Trans. Graph.*, vol. 22, pp. 943–949, 2003.
- [66] T. Kanai, “MeshToSS: Converting subdivision surfaces from dense meshes,” in *Proc. Vision Modeling and Visualization Conf. (VMV)*, 2001, pp. 325–332.
- [67] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1995, pp. 1942–1948.
- [68] R. Kimmel, R. Malladi, and N. Sochen, “Images as embedded maps and minimal surfaces: Movies, color, texture, and volumetric medical images,” *Int. J. Comput. Vision*, vol. 39, no. 2, pp. 111–129, 2000.
- [69] L. Kobbelt, “ $\sqrt{3}$ subdivision,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00, New York, NY, USA, 2000, pp. 103–112.
- [70] M. Krueger, P. Delmas, and G. Gimel'Farb, “Active contour based segmentation of 3d surfaces,” in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV '08, 2008, pp. 350–363.
- [71] R. Lai and T. F. Chan, “A framework for intrinsic image processing on surfaces,” Department of Mathematics, UCLA, Los Angeles, CA 90095, USA, Tech. Rep., 2010.
- [72] Y.-K. Lai, S.-M. Hu, and H. Pottmann, “Surface fitting based on a feature sensitive parametrization,” *Computer-Aided Design*, vol. 38, no. 7, pp. 800 – 807, 2006.

-
- [73] N. Laiche and S. Larabi, “Retrieval of 2D objects and shape matching using the b-splines representation,” in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, 2011, pp. 495–500.
- [74] P. Laurent-Gengoux and M. Mekhilef, “Optimization of a NURBS representation,” *Computer-Aided Design*, vol. 25, no. 11, pp. 699–710, 1993.
- [75] G. Lavoué and F. Dupont, “Semi-sharp subdivision surface fitting based on feature lines approximation,” *Computers & Graphics*, vol. 33, pp. 151–161, 2009.
- [76] A. Lee, H. Moreton, and H. Hoppe, “Displaced subdivision surfaces,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’00, 2000, pp. 85–94.
- [77] J. Lellmann and C. Schnörr, “Continuous multiclass labeling approaches and algorithms,” *SIAM J. Img. Sci.*, vol. 4, no. 4, pp. 1049–1096, 2011.
- [78] K. Levenberg, “A method for the solution of certain nonlinear problems in least squares,” *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.
- [79] W. Li, S. Xu, G. Zhao, and L. P. Goh, “Adaptive knot placement in B-spline curve approximation,” *Computer-Aided Design*, vol. 37, pp. 791–797, 2005.
- [80] Y. Lin, N. Wohlberg, and H. Guo, “UPRE method for total variation parameter selection,” *Signal Processing*, vol. 90, pp. 2546–2551, 2010.
- [81] R. Ling, W. Wang, and D. Yan, “Fitting sharp features with loop subdivision surfaces,” in *Proceedings of the Symposium on Geometry Processing*, ser. SGP ’08, 2008, pp. 1383–1391.
- [82] N. Litke, A. Levin, and P. Schröder, “Fitting subdivision surfaces,” in *Proceedings of the Conference on Visualization ’01*, ser. VIS ’01. IEEE Computer Society, 2001, pp. 319–324.

-
- [83] L. Liu, C.-L. Tai, Z. Ji, and G. Wang, “Non-iterative approach for global mesh optimization,” *Computer-Aided Design*, vol. 39, no. 9, pp. 772–782, 2007.
- [84] Q. Liu, J. Luo, and Y. Zhu, “Adaptive image decomposition by improved bilateral filter,” *International Journal of Computer Applications*, vol. 23, no. 7, pp. 16–22, 2011.
- [85] X. Liu, H. Bao, H.-Y. Shum, and Q. Peng, “A novel volume constrained smoothing method for meshes,” *Graph. Models*, vol. 64, no. 3-4, pp. 169–182, May 2002.
- [86] C. T. Loop, “Smooth subdivision surfaces based on triangles,” Master’s thesis, University of Utah, Department of Mathematics, 1987.
- [87] M. I. A. Lourakis, *A brief description of the Levenberg-Marquardt algorithm implemented by levmar*, Institute of Computer Science Foundation for Research and Technology-Hellas, 2005, Available at <http://www.ics.forth.gr/lourakis/levmar/levmar.pdf>.
- [88] L. M. Lui, X. Gu, T. F. Chan, and S.-T. Yau, “Variational method on riemann surfaces using conformal parameterization and its applications to image processing,” *Methods and Applications of Analysis*, vol. 15, pp. 513–538, 2008.
- [89] W. Ma and J. Kruth, “Parameterization of randomly measured points for least squares fitting of b-spline curves and surfaces,” *Computer-Aided Design*, vol. 27, no. 9, pp. 663 – 675, 1995.
- [90] W. Ma, X. Ma, S.-K. Tso, and Z. Pan, “A direct approach for subdivision surface fitting from a dense triangle mesh,” *Computer-Aided Design*, vol. 36, no. 6, pp. 525 – 536, 2004.
- [91] M. Marinov and L. Kobbelt, “Optimization methods for scattered data approximation with subdivision surfaces,” *Graph. Models*, vol. 67, no. 5, pp. 452–473, 2005.

-
- [92] D. Marquardt, “An algorithm for Least-Squares estimation of nonlinear parameters,” *SIAM Journal on Applied Mathematics*, vol. 11(2), pp. 431–441, 1963.
- [93] A. Masood, “Optimized polygonal approximation by dominant point deletion,” *Pattern Recognition*, vol. 41, no. 1, pp. 227–239, 2008.
- [94] N. Mathematik and A. Chambolle, “Image recovery via total variation minimization and related problems,” *Numerische Mathematik*, vol. 76, pp. 167–188, 1997.
- [95] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations: The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*. Boston, MA, USA: American Mathematical Society, 2001.
- [96] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Comm. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, Jul. 1989.
- [97] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, “Laplacian mesh optimization,” in *GRAPHITE*, ser. GRAPHITE ’06. ACM, 2006, pp. 381–389.
- [98] M. Nikolova, “Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers,” *SIAM J. Numer. Anal.*, vol. 40, pp. 965–994, March 2002.
- [99] M. Nikolova, “A variational approach to remove outliers and impulse noise,” *J. Math. Imaging Vis.*, vol. 20, pp. 99–120, 2004.
- [100] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, “Image-based modeling and photo editing,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’01. ACM, 2001, pp. 433–442.
- [101] J. Oliveira, J. Nioucas-Dias, and M. Figueiredo, “Adaptive total variation image deblurring: A majorization-minimization approach,” *Signal Processing*, vol. 89, pp. 1683–1693, 2009.

-
- [102] S. Osher, A. Sol, and L. Vese, “Image decomposition and restoration using total variation minimization and the h^1 norm,” *Simul.*, vol. 1, pp. 349–370, 2003.
- [103] H. Park and J.-H. Lee, “B-spline curve fitting based on adaptive curve refinement using dominant points,” *Computer-Aided Design*, vol. 39, pp. 439–451, 2007.
- [104] M. T. Parvez and S. A. Mahmoud, “Polygonal approximation of digital planar curves through adaptive optimizations,” *Pattern Recogn. Lett.*, vol. 31, no. 13, pp. 1997–2005, 2010.
- [105] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 629–639, 1990.
- [106] E. Persoon and K.-S. Fu, “Shape discrimination using fourier descriptors,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 3, pp. 170–179, 1977.
- [107] J. Peters and U. Reif, “The simplest subdivision scheme for smoothing polyhedra,” *ACM Trans. Graph.*, vol. 16, no. 4, pp. 420–431, 1997.
- [108] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, “Digital photography with flash and no-flash image pairs,” *ACM Trans. Graph.*, vol. 23, pp. 664–672, 2004.
- [109] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, “An algorithm for minimizing the Mumford-Shah functional,” in *Computer Vision, IEEE 12th International Conference on*, 2009.
- [110] W. H. Richard, R. Bartels, and G. Labahn, “Fitting uncertain data with NURBS,” in *in Proc. 3rd Int. Conf. on Curves and Surfaces in Geometric Design*. Vanderbilt University Press, 1996, pp. 1–8.

-
- [111] M. Riyazuddin, "Visualization with NURBS using simulated annealing optimization technique," Master's thesis, Information & Computer Science, King Fahd University Of Petroleum & Minerals, Dhahran, Saudi Arabia, 2004.
- [112] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
- [113] J. A. Saghri and H. Freeman, "Analysis of the precision of generalized chain codes for the representation of planar curves," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-3, pp. 533–539, 1981.
- [114] M. Sarfraz and S. A. Raza, "Capturing outline of fonts using genetic algorithm and splines," in *Information Visualisation, 2001. Proceedings. Fifth International Conference on*, 2001.
- [115] M. Sarfraz, S. A. Raza, and M. H. Baig, "Computing optimized curves with NURBS using evolutionary intelligence," in *Proceedings of the 2005 international conference on Computational Science and its Applications - Volume Part I*, 2005.
- [116] E. Saux and M. Daniel, "An improved hoschek intrinsic parametrization," *Comput. Aided Geom. Des.*, vol. 20, no. 8-9, pp. 513–521, 2003.
- [117] S. Schaefer, T. Ju, and J. Warren, "A unified, integral construction for coordinates over closed curves," *Comput. Aided Geom. Des.*, vol. 24, no. 8-9, pp. 481–493, 2007.
- [118] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 276–283, 2004.
- [119] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, "T-splines and T-NURCCs," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 477–484, 2003.

-
- [120] T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin, “Non-uniform recursive subdivision surfaces,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998, pp. 387–394.
- [121] Y. Shen and K. Barner, “Fuzzy vector median-based surface smoothing,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 10, no. 3, pp. 252–265, 2004.
- [122] N. Sochen, R. Kimmel, and R. Malladi, “A general framework for low level vision,” *Image Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 310–318, 1998.
- [123] J. Stam, “Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’98. ACM, 1998, pp. 395–404.
- [124] J. Stam, “Flows on surfaces of arbitrary topology,” *ACM Trans. Graph.*, vol. 22, pp. 724–731, 2003.
- [125] J. Stam and C. Loop, “Quad/triangle subdivision,” *Computer Graphics Forum*, vol. 22(1), pp. 79–85, 2003.
- [126] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, “Fast and effective feature-preserving mesh denoising,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 925–938, 2007.
- [127] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, “Random walks for feature-preserving mesh denoising,” *Comput. Aided Geom. Des.*, vol. 25, no. 7, pp. 437–456, 2008.
- [128] H. Suzuki, S. Takeuchi, F. Kimura, and T. Kanai, “Subdivision surface fitting to a range of points,” in *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, ser. PG ’99. IEEE Computer Society, 1999, pp. 158–167.

-
- [129] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [130] X.-C. Tai and C. Wu, “Augmented Lagrangian method, dual methods and split bregman iteration for ROF model,” in *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, ser. SSVM '09, 2009, pp. 502–513.
- [131] G. Taubin, “A signal processing approach to fair surface design,” in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '95. ACM, 1995, pp. 351–358.
- [132] C.-H. Teh and R. Chin, “On the detection of dominant points on digital curves,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 8, pp. 859–872, 1989.
- [133] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of the Sixth International Conference on Computer Vision*, 1998.
- [134] A. Toshev, B. Taskar, and K. Daniilidis, “Object detection via boundary structure segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 950–957.
- [135] J. Tumblin and G. Turk, “LCIS: a boundary hierarchy for detail-preserving contrast reduction,” in *Annual Conference on Computer Graphics*, 1999, pp. 83–90.
- [136] L. A. Vese and T. F. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” *Int. J. Comput. Vision*, vol. 50, no. 3, pp. 271–293, 2002.
- [137] L. A. Vese and S. J. Osher, “Modeling textures with total variation minimization and oscillating patterns in image processing,” *Journal of Scientific Computing*, vol. 19, pp. 553–572, 2003.

-
- [138] T. Wagner, T. Michelitsch, and A. Sacharow, “On the design of optimisers for surface reconstruction,” in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007.
- [139] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, “Decoupling noises and features via weighted L1-analysis compressed sensing,” *ACM Transactions on Graphics*, 2014, to appear.
- [140] W. Wang, H. Pottmann, and Y. Liu, “Fitting B-spline curves to point clouds by curvature-based squared distance minimization,” *ACM Trans. Graph.*, vol. 25, pp. 214–238, 2006.
- [141] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction,” *SIAM J. Img. Sci.*, vol. 1, pp. 248–272, 2008.
- [142] Y. Wang and J. Zheng, “Control point removal algorithm for T-spline surfaces,” in *Geometric Modeling and Processing*. Springer Berlin Heidelberg, 2006.
- [143] Y. Wang and E. K. Teoh, “2D affine-invariant contour matching using B-spline model,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 1853–1858, 2007.
- [144] K. Weinert, T. Surmann, and J. Mehnen, “Evolutionary surface reconstruction using CSG-NURBS-hybrids,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, 2001.
- [145] C. Wu, J. Deng, and F. Chen, “Diffusion equations over arbitrary triangulated surfaces for filtering and texture applications,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, pp. 666–679, 2008.
- [146] C. Wu, J. Deng, F. Chen, and X. Tai, “Scale-space analysis of discrete filtering over arbitrary triangulated surfaces,” *SIAM J. Img. Sci.*, vol. 2, pp. 670–709, 2009.

-
- [147] C. Wu and X.-C. Tai, “Augmented Lagrangian method, dual methods, and split bregman iteration for ROF, vectorial TV, and high order models,” *SIAM J. Img. Sci.*, vol. 3, pp. 300–339, 2010.
- [148] C. Wu, J. Zhang, Y. Duan, and X.-C. Tai, “Augmented Lagrangian method for total variation based image restoration and segmentation over triangulated surfaces,” *Journal of Scientific Computing*, vol. 50, pp. 145–166, 2012.
- [149] C. Wu, J. Zhang, and X.-C. Tai, “Augmented Lagrangian method for total variation restoration with non-quadratic fidelity,” *Inverse Problems and Imaging*, vol. 5, pp. 237–261, 2011.
- [150] W.-Y. Wu, “An adaptive method for detecting dominant points,” *Pattern Recognition*, vol. 36, no. 10, pp. 2231 – 2237, 2003.
- [151] H. Xie and H. Qin, “A novel optimization approach to the effective computation of NURBS knots,” *International Journal of Shape Modeling*, vol. 07, pp. 199–227, 2001.
- [152] W.-C. Xie, X.-F. Zou, J.-D. Yang, and J.-B. Yang, “Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines,” *Computer-Aided Design*, vol. 44, no. 11, pp. 1127–1140, 2012.
- [153] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via L0 gradient minimization,” *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, 2011.
- [154] H. Yagou, Y. Ohtake, and A. Belyaev, “Mesh smoothing via mean and median filtering applied to face normals,” in *Geometric Modeling and Processing, 2002. Proceedings*, 2002, pp. 124–131.
- [155] H. Yagou, Y. Ohtake, and A. Belyaev, “Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding,” in *Computer Graphics International, 2003. Proceedings*, 2003, pp. 28–33.

-
- [156] J. Yang, Y. Zhang, and W. Yin, “An efficient tvl1 algorithm for deblurring multichannel images corrupted by impulsive noise,” *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, 2009.
- [157] F. Yoshimoto, T. Harada, and Y. Yoshimoto, “Data fitting with a spline using a real-coded genetic algorithm,” *Computer-Aided Design*, vol. 35, no. 8, pp. 751 – 760, 2003.
- [158] D. Zhang and G. Lu, “Review of shape representation and description techniques,” *Pattern Recognition*, vol. 37, no. 1, pp. 1 – 19, 2004.
- [159] J. Zhang and C. Wu, “Fast optimization for multichannel total variation minimization with non-quadratic fidelity,” *Signal Processing*, vol. 91, pp. 1933–1940, 2011.
- [160] L. Zhang, L. Liu, C. Gotsman, and H. Huang, “Mesh reconstruction by meshless denoising and parameterization,” *Comput. Graph.*, vol. 34, no. 3, pp. 198–208, 2010.
- [161] X. Zhao, C. Zhang, B. Yang, and P. Li, “Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation,” *Computer-Aided Design*, vol. 43, pp. 598–604, 2011.
- [162] W. Zheng, P. Bo, Y. Liu, and W. Wang, “Fast B-spline curve fitting by L-BFGS,” *Comput. Aided Geom. Des.*, vol. 29, no. 7, pp. 448–462, 2012.
- [163] Y. Zheng, H. Fu, O. Au, and C. Tai, “Bilateral normal filtering for mesh denoising,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, pp. 1521 – 1530, 2010.

Author's Publications

1. **Xiaoqun Wu**, Jianmin Zheng, Chunlin Wu, Yiyu Cai, “Variational structure-texture image decomposition on manifolds,” *Signal Processing*, 93(7): 1773-1784, 2012.
2. **Xiaoqun Wu**, Yiyu Cai , Jianmin Zheng, “TV- L^1 Optimization For B-spline Surface Reconstruction With Sharp Features,” *CAD/Graphics*: 44-51, 2013.
3. **Xiaoqun Wu**, Jianmin Zheng, Xunnian Yang, Yiyu Cai, “Simplification of meshes into curved PN triangles,” *Proceedings of the Second APSIPA Annual Summit and Conference*, Singapore, pp.598-604, 2010.
4. **Xiaoqun Wu**, Jianmin Zheng, Yiyu Cai, “Variational B-spline Shape Representation for Unorganized Contour points”, to be submitted to journal, 2014.
5. **Xiaoqun Wu**, Jianmin Zheng, Chi-Wing Fu, “Mesh Denoising using ROF model”, to be submitted, 2014.
6. **Xiaoqun Wu**, Jianmin Zheng, Chi-Wing Fu, Yiyu Cai, “Variational Feature-aware Subdivision Surface Reconstruction”, in preparation, 2014.