Adaptive edge-preserving color image regularization framework by partial differential equations

Zhu, Lin

2011

Zhu, L. (2011). Adaptive edge-preserving color image regularization framework by partial differential equations. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46451>

<https://doi.org/10.32657/10356/46451>

An Adaptive Edge-Preserving Color Image Regularization Framework by Partial Differential Equations

Zhu Lin

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2011

Acknowledgements

I would like to express my sincere thanks to my parents, who, throughout all these years, have always kept great confidence in me, and supported me unconditionally in every way. Without their love and encouragement, this thesis could never be possible.

I would also like to express my special thanks to my supervisor, Associate Professor Andrzej Stefan Sluzek, for his precious time, invaluable instructions and great support. It is him who always gave me strong support and consistent encouragement through the tough times of my research period.

Last but not least; I would like to thank all my colleagues in Center of Computational Intelligence, School of Computer Engineering, Nanyang Technological University; and all my colleagues in Hewlett Packard, Singapore during my part-time PhD study. Thanks for all the help and support you have given to me; it is my great pleasure to work with so many wonderful people.

Table of Contents

Acknowledgements	ii
List of Figures	vi
List of Tables.....	ix
Summary	x
Chapter 1. Introduction	1
1.1. Edge-preserving image regularization.....	1
1.2. Mathematical notations for images	3
1.3. Organization of this thesis	5
Chapter 2. Summary of the State of the Art of Image Regularization	
Methods.....	8
2.1. Introduction	8
2.2. Grayscale image regularization overview	8
2.2.1. Variation-based regularization methods	8
2.2.1.1. Isotropic regularization	9
2.2.1.2. Perona-Malik regularization	11
2.2.1.3. Total Variation regularization	14
2.2.1.4. Summary of variational regularization	16
2.2.2. Gradient direction oriented diffusions	16
2.2.3. Divergence-based regularization methods	19
2.3. Color image regularization overview	22
2.3.1. Vector geometry.....	22
2.3.2. Vector Φ -functional regularization.....	26
2.3.3. Vector gradient oriented and trace-based formulation.....	27
2.3.4. Vector divergence-based regularization.....	29
2.4. Data fidelity term overview.....	34
2.4.1. L^2 -norm based data fidelity term.....	35
2.4.2. L^1 -norm based data fidelity term.....	38
2.4.3. Other fidelity norms	39
Chapter 3. Locally Adaptive Edge-Preserving Color Image Regularization	
Framework	41

3.1. Adaptive divergence-based regularization term.....	41
3.1.1. Comparing divergence-based and trace-based formulations	42
3.1.2. Edge indicator function.....	47
3.1.3. Design of the edge-preserving diffusion tensor	49
3.1.4. Comparisons of different regularization terms	50
3.2. Adaptive data fidelity term.....	53
3.2.1. Adaptive edge-preserving fidelity weight.....	56
3.2.1.1. Mean-velocity based edge-preserving fidelity weight	57
3.2.1.2. Channel-wise adaptive edge-preserving fidelity weight	58
3.3. Final framework: adaptive regularization term with adaptive fidelity term.....	60
3.4. Experimental results	63
3.5. Conclusion.....	78
Chapter 4. Two-Phase Extension of the Proposed Regularization	
Framework for Color Impulse and Mixed Noise Removal	79
4.1. Impulse noise removal by the proposed framework with L^1 -norm based fidelity term.....	80
4.2. Two-phase extension of the proposed framework for color impulse noise removal.....	81
4.2.1. Color impulse noise detection.....	82
4.2.1.1. Color salt-and-pepper noise detection by color AMF	82
4.2.1.2. Color ROAD-based random-valued impulse noise detection.....	83
4.2.2. Reconstruct detected impulse noise corrupted pixels	85
4.3. Two-phase regularization framework for mixed impulse and Gaussian noises removal.....	87
4.4. Experimental results	90
4.5. Conclusion.....	102
Chapter 5. Applications and Possible Extensions of the Proposed	
Regularization Framework.....	103
5.1. Zernike moments-based color image regularization	104
5.1.1. Property of Zernike moments	104
5.1.2. Zernike moments-based color edge detection.....	105
5.1.3. Zernike moments-based color edge indicator function and the corresponding experimental results	107
5.2. Possible nonlocal extension of our proposed framework.....	111
5.3. Possible applications of our proposed regularization framework	113
Chapter 6. Conclusions and Future Work.....	116

6.1. Conclusions	116
6.2. Future research directions.....	117
Bibliography.....	120

List of Figures

Figure 2-1: Image contour and its pointwise defined gradient and tangent direction.....	12
Figure 3-1: Regularization results of a synthetic color image corrupted with additive zero-mean Gaussian white noise ($\sigma=80$) using regularization terms only. (a) Original image I ; (b) Noisy image I_0 ($\sigma=80$, PSNR=10.07); (c) TD's trace-based regularization term (PSNR=29.26); (d) The residual image ($I_0 - I + 100$) of (c); (e) Our proposed divergence-based regularization term (PSNR=34.31); (f) The residual image ($I_0 - I + 100$) of (e).	52
Figure 3-2: Regularization results of the 256x256 <i>House</i> image corrupted by additive Gaussian noise ($\sigma=40$). (a) Original image. (b) Noisy image ($\sigma=40$, PSNR=16.10dB); (c) Vector TV (PSNR=28.30dB); (d) Beltrami Flow (PSNR=28.20dB); (e) TD's trace-based method (PSNR=28.69dB); (f) Our proposed method (PSNR=29.54dB).	66
Figure 3-3: Regularization results of the 512x512 <i>Lena</i> image corrupted by additive Gaussian noise ($\sigma=20$). (a) Original image; (b) Noisy image ($\sigma=20$, PSNR=22.12dB); (c) Vector TV (PSNR=31.10dB); (d) Beltrami Flow: (PSNR=31.45dB); (e) TD's trace-based method (PSNR=31.28dB); (f) Our proposed algorithm (PSNR=31.89dB).	72
Figure 3-4: Regularization results of the 512x512 <i>Lena</i> image corrupted by additive Gaussian noise ($\sigma=40$). (a) Noisy image ($\sigma=40$, PSNR=16.10dB); (b) Vector TV (PSNR=28.70dB); (c) Beltrami Flow (PSNR=28.59dB); (d) TD's trace-based method (PSNR=28.42dB); (e) Our proposed algorithm (PSNR=29.47dB).	73
Figure 3-5: Regularization results of the 512x768 <i>Lighthouse</i> image corrupted by additive Gaussian noise ($\sigma=40$). (a) Original image; (b) Noisy image ($\sigma=40$, PSNR=16.10); (c) Vector TV (PSNR=25.62dB); (d) Beltrami Flow (PSNR=26.53dB); (e) TD's trace-based method (PSNR=26.59dB); (f) Our proposed method (PSNR=27.39dB).	75
Figure 3-6: Regularization results of the 512x512 <i>Peppers</i> image corrupted by additive Gaussian noise ($\sigma=80$). (a) Original image; (b) Noisy image ($\sigma=80$, PSNR=10.08); (c) Vector TV (PSNR=25.59); (d) Beltrami flow (PSNR=24.97); (e) TD's method (PSNR=24.92); (f) Our proposed method (PSNR=26.58).	76

Figure 3-7: Regularization results of a real noisy image (taken by DC at ISO3200). (a) Noisy image (ISO 3200); (b) Vector TV; (c) Beltrami flow; (d) TD's trace-based method; (e) Our proposed method.....	77
Figure 4-1: Regularization results for the 256x256 <i>Lena</i> image corrupted by salt-and-pepper noise. (a) <i>Lena</i> image corrupted by salt-and-pepper noise $s=20\%$ (PSNR=12.27dB); (b) Color AMF (PSNR=30.97dB); (c) Vector TV + L1 fidelity term (PSNR=27.32dB); (d) Our proposed method (PSNR=33.83dB).	96
Figure 4-2: Regularization results for the 256x256 <i>Lena</i> image corrupted by salt-and-pepper noise. (a) <i>Lena</i> image corrupted by salt-and-pepper noise $s=50\%$ (PSNR=8.26dB); (b) Color AMF (PSNR=24.31dB); (c) Vector TV + L1 fidelity term (PSNR=24.47dB); (d) Our proposed method (PSNR=31.21dB).	97
Figure 4-3: Regularization results for the 256x256 <i>Lena</i> image corrupted by random-valued impulse noise. (a) <i>Lena</i> image corrupted by random-valued impulse noise $r=20\%$ (PSNR=15.60dB); (b) Color ROAD median filter (PSNR=28.58dB); (c) Vector TV + L1 fidelity term (PSNR=27.21dB); (d) Our proposed method (PSNR=30.44dB).....	98
Figure 4-4: Regularization results for the 256x256 <i>Lena</i> image corrupted by random-valued impulse noise. (a) <i>Lena</i> image corrupted by random-valued impulse noise $r=40\%$ (PSNR=12.63dB); (b) Color ROAD median filter (PSNR=24.95dB); (c) Vector TV + L1 fidelity term (PSNR=24.47dB); (d) Our proposed method (PSNR=27.04dB).....	99
Figure 4-5: Regularization results for the 256x256 <i>Lena</i> image corrupted by mixed Gaussian and salt-and-pepper noise. (a) <i>Lena</i> image corrupted by both additive Gaussian noise $\sigma=20$ and salt-and-pepper noise $s=20\%$ (PSNR=11.93dB); (b) "Impulse removed" image after Phase-1 of the proposed method (PSNR=23.16); (c) Final result of our proposed method (PSNR=28.36dB); (d) Vector TV + L1 fidelity term (PSNR=25.52dB).....	100
Figure 4-6: Regularization results for the 256x256 <i>Lena</i> image corrupted by mixed Gaussian and random-valued impulse noise. (a) <i>Lena</i> image corrupted by both additive Gaussian noise $\sigma=20$ and random-valued impulse noise $r=20\%$ (PSNR=14.19dB); (b) "Impulse removed" image after Phase-1 of our proposed method (PSNR=23.63); (c) Final result of our proposed method (PSNR=27.37dB); (d) Vector TV + L1 fidelity term (PSNR=25.20dB).	101
Figure 5-1: 2D step edge model with sub-pixel accuracy.....	106

Figure 5-2: Comparisons of color edge responses of local color gradient norm and Zernike moment-based color gradient norm: (a) Original 256x256 <i>House</i> image; (b) <i>House</i> image corrupted by additive Gaussian noise $\sigma=80$; (c) Local color gradient norm of (a); (d) Zernike moment-based color gradient norm of (a); (e) Local color gradient norm of (b); (f) Zernike moment-based color gradient norm of (b).....	108
Figure 5-3: Comparisons of regularization results of the 256x256 <i>House</i> image corrupted by Gaussian noise using different edge indicator functions. (a) <i>House</i> image corrupted by additive Gaussian noise $\sigma=80$ (PSNR=10.07dB); (b) Regularization results of our proposed method using the original local gradient-based edge indicator function (PSNR=26.39); (c) Regularization results of our proposed method using the Zernike moment-based edge indicator function (PSNR=26.72dB); (d) Final edge map after regularization.....	110
Figure 5-4: A quick example showing the potential of the proposed image regularization framework in regularizing a heavily compressed jpeg image. (a) Original image; (b) Regularized image.	114

List of Tables

Table 2-1: Summary of advantages and disadvantages of three main kinds of image regularization framework	33
Table 3-1: Comparison of CPU time in seconds for 4 methods for image of different sizes and different noise level.....	70
Table 4-1: Comparisons of CPU time in seconds for different level of salt-and-pepper and random impulse noise	94
Table 4-2: Comparisons of CPU time in seconds for different mixed noise	95

Summary

In this thesis, we have studied the problem of color image regularization, which is a low-level process and is often used as a key pre-processing step in many image processing applications. Most of these applications require that the regularization stage can preserve as much important image features (edges and corners etc.) as possible, while still being able to effectively remove noise and unwanted details. Although there are many existing regularization methods, few of them can produce both efficient noise removal and good edge preservation. To achieve better edge-preserving regularization performance, we have proposed a locally adaptive edge-preserving regularization framework for color images. The basic idea of our proposed framework is to treat edge regions and homogenous regions adaptively by applying different regularization process to them. We proposed a locally adaptive regularization term in Chapter 3, which is better adapted to local edge geometry. Besides that, an automatically calculated adaptive data fidelity term was also proposed to help better preserve edges. Experimental results are presented to show that our proposed framework achieved a good balance between noise removal and edge preservation comparing with other methods.

In Chapter 4, we further extended our regularization framework to handle impulse noise by extending a grayscale *impulse noise detection* method to color images and used together with our proposed regularization framework. We also considered the case of mixed impulse and Gaussian noise by proposing an innovative two-phase framework inspired from *color image inpainting*. Finally, we proposed to use a semi-local Zernike moments in our regularization framework to get more robust performance for highly-noisy images. Possible extension of our proposed framework to the nonlocal version was also discussed and suggested as future research directions.

Chapter 1. Introduction

1.1. Edge-preserving image regularization

Most image processing and computer vision applications need to extract useful information from captured images; however, real images we deal with are often noisy, distorted or blurred due to poor lighting condition, capturing device noise, transmission errors, etc. This creates a great difficulty for those applications since images are not “regular” enough and contain a lot of unwanted noise or distortions.

A basic problem in image processing is, given a n -dimensional vector-valued noisy (irregular) image $\mathbf{I}_{noisy} : \Omega \rightarrow \mathbb{R}^n$ defined on a 2D spatial domain $\Omega \subset \mathbb{R}^2$, to obtain a regularized (e.g. noise-free, preserving only important features such as edges and corners etc.) image $\mathbf{I}_{regular}$, from the original noisy or corrupted image $\mathbf{I}_{noisy}$, where

$$\mathbf{I}_{regular} + \eta = \mathbf{I}_{noisy}, \quad (1.1)$$

and η are noise or other unwanted details or degradations in the original image $\mathbf{I}_{noisy}$.

The process of finding regularized image is normally defined as image regularization, which has attracted a lot of research interests in image processing and computer vision community during the past over 20 years. It is used either to directly restore degraded images, or more indirectly, as a pre-processing step that eases further analysis of the original images.

Image regularization is a key pre-processing stage for higher-level image processing and computer vision applications such as image segmentation, edge detection, corner/junction detection, image registration, object recognition and identification, automatic target tracking, etc. Most of these applications require that the regularization

1.1 Edge-preserving image regularization

stage can preserve as much edge information as possible; since edges not only contain essential information of the objects themselves; they also define locations of the objects.

With the rapid development in the quality and resolution of image capturing devices, image noise has been greatly reduced, some are expecting almost noiseless camera in the future. This has also raised the question that whether image regularization is still needed in the future as discussed recently in [23]. I think the answer should be yes. First of all, no matter how good the image capturing device could be, it always depend on the light condition, in some cases when lighting condition is not good such as remote sensing etc., the captured image will still more or less be noisy. Secondly, image regularization is a bit different from image denoising, even the complete noise free images still can be regularized because those images may contain some small scale feature such as hair, texture etc., which are not of interests; image regularization can help remove those unwanted small details and make the subsequent step such as feature extraction and object recognition easier. So even with the rapid improvement of image capturing quality, I think image regularization will still be a useful preprocessing step for most image processing applications.

Although many image regularization algorithms have been proposed in the literature, their regularization results, especially the edge-preserving performance for images of high complexity (e.g. highly noisy) are still not very satisfactory. In this thesis, we will tackle this challenge and propose an adaptive edge-preserving image regularization approach to preserve important image edge information as much as possible during the regularization process.

1.2. Mathematical notations for images

Since this thesis deals with regularization of color images, we need to define some mathematical notations which will be used throughout this thesis. Although nowadays images are mostly stored in a discrete format rather than in continuous formats, it is generally assumed that the discretization is fine enough to allow approximating these discrete signals by continuous mathematical functions.

In this thesis, we will mainly consider 2D images rather than volumes, so we will define our images on $\Omega \subset \mathbb{R}^2$, which is a 2D closed spatial domain. Images will be defined as a function $\mathbf{I}(x, y)$ from Ω to $\mathbb{R}^n$:

$$\mathbf{I}: \begin{cases} \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n \\ (x, y) \rightarrow \mathbf{I}(x, y) = (I_1(x, y), I_2(x, y), \dots, I_n(x, y))^T \end{cases} \quad (1.2)$$

Grayscale images correspond to $n=1$ and color images correspond to $n=3$, with vector values in (R, G, B) . We use **bold** letters to denote multi-valued variables such as vector-valued images and matrix. Throughout this thesis, we use $|\mathbf{X}|$ to denote the L^2 -norm $\|\mathbf{X}\|_{L^2} = \sqrt{X_1^2 + X_2^2 + \dots + X_n^2}$.

A derivative of the scalar image I with the respective variable x is written as

$$I_x = \frac{\partial I}{\partial x}.$$

For a vector-valued image $\mathbf{I}$, we define $\mathbf{I}_x \in \mathbb{R}^n$ as

$$\mathbf{I}_x = \left(\frac{\partial I_1}{\partial x}, \frac{\partial I_2}{\partial x}, \dots, \frac{\partial I_n}{\partial x} \right)^T.$$

1.2 Mathematical notations for images

The derivation of a scalar image I with respect to its spatial coordinates is normally called image gradient and represented by ∇I :

$$\begin{cases} \nabla I = (I_x, I_y)^T \\ |\nabla I| = \sqrt{I_x^2 + I_y^2} \end{cases}, \quad (1.3)$$

where $|\nabla I|$ is the L^2 -norm of the gradient, which gives a scalar and point-wise defined measure of image variations. Similarly, for a multi-valued image $\mathbf{I}$, we have ∇I_i and $|\nabla I_i|$ for each image channel I_i . However, for the gradient norm there is no natural extension, in this thesis we will use Di Zenzo's vector gradient norm $|\nabla \mathbf{I}|$ [27]:

$$\begin{cases} \nabla \mathbf{I} = (\mathbf{I}_x, \mathbf{I}_y)^T \\ |\nabla \mathbf{I}| = \sqrt{\mathbf{I}_x \cdot \mathbf{I}_x + \mathbf{I}_y \cdot \mathbf{I}_y} = \sqrt{\sum_{i=1}^n |\nabla I_i|^2} \end{cases}. \quad (1.4)$$

Similar to scalar images, $|\nabla \mathbf{I}|$ is also a useful point-wise scalar measure of local vector variations (both in terms of vector norms and orientations) of image $\mathbf{I}$.

For each image point (x, y) , we define a structure tensor $\mathbf{G}$:

$$\mathbf{G} = (\nabla I \nabla I^T) = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}. \quad (1.5)$$

We can define directional derivatives in any given direction $\mathbf{u} = (u, v)^T$ as below:

$$\begin{cases} I_u = \nabla I \cdot \mathbf{u} = u I_x + v I_y \\ \mathbf{u} = (u, v)^T \end{cases}. \quad (1.6)$$

Similarly, we can define the second order derivative of scalar image I with respect to x then y as:

1.3 Organization of this thesis

$$I_{xy} = \frac{\partial^2 I}{\partial x \partial y} . \quad (1.7)$$

Subsequently, we define the Hessian of I as the matrix $\mathbf{H}$ of the second order derivatives with respect to the spatial coordinates:

$$\mathbf{H} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix} . \quad (1.8)$$

We assume our images are regular enough so that $I_{xy} = I_{yx}$ and $\mathbf{H}$ is a symmetric matrix. We will also use the Laplacian Operator Δ defined as

$$\Delta I = I_{xx} + I_{yy} . \quad (1.9)$$

Similarly, for the second order directional-derivatives in a direction $\mathbf{u} = (u, v) \in \mathbb{R}^2$, we have:

$$I_{\mathbf{uu}} = \frac{\partial^2 I}{\partial \mathbf{u}^2} = \nabla (\nabla I \cdot \mathbf{u}) \cdot \mathbf{u} = \mathbf{u}^T \mathbf{H} \mathbf{u} = u^2 I_{xx} + 2uv I_{xy} + v^2 I_{yy} . \quad (1.10)$$

Besides the Laplacian Operator, we will also use the linear divergence operator div in this thesis:

$$\text{div}(\mathbf{F}) = \nabla \cdot \mathbf{F} = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \dots + \frac{\partial F_n}{\partial x_n} , \quad (1.11)$$

where $\mathbf{F} = (F_1, F_2, \dots, F_n)^T$ is a vector defined in a Euclidean coordinate system while

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T .$$

1.3. Organization of this thesis

The thesis is organized as follows:

1.3 Organization of this thesis

In **Chapter 2**, the state of the art of both the grayscale and color image regularization methods will be reviewed. We will try to group them into different categories based on their characteristics and compare the advantages and drawbacks of different categories.

In **Chapter 3**, we will present the proposed adaptive edge-preserving color image regularization framework. Details about how to design the edge-preserving regularization term; and how to compute the locally adaptive data fidelity term are explained. How to reliably estimate image noise variance and calculate the adaptive edge-preserving function based on that is also introduced. We will also present the experimental results comparing the proposed method with existing methods to show the improvement of the proposed framework.

In **Chapter 4**, we will discuss the problem of removing impulse noise and mixed Gaussian and impulse noise based on the proposed regularization framework. We will extend two impulse noise detection schemes to color images, and use a modified version of the proposed regularization framework to reconstruct detected impulse noise corrupted pixels. Finally, a two-phase regularization framework is proposed to remove mixed Gaussian and impulse noise. We will also present experimental results to show that after extension, the proposed framework is capable of handling both impulse noise and mixed Gaussian and impulse noises.

In **Chapter 5**, we will analyze the difficulty of the proposed framework when the noise level is very high and discuss how to solve this issue by using Zernike moments to construct more robust edge indicator function for our algorithms. The possible extension to the nonlocal framework of our proposed method will be discussed as future research directions. Some special applications of the proposed framework, including color edge detection, color image inpainting, image compression artifact regularization etc. are discussed.

1.3 Organization of this thesis

In **Chapter 6**, the conclusions are presented and future research directions related to this work are suggested.

Chapter 2. Summary of the State of the Art of Image Regularization Methods

2.1. Introduction

In this Chapter, we will review the state of the art of image regularization. We will first review those classical methods for scalar image regularization and try to summarize them into different categories depending on their characteristics. Then we will see how these methods can be successfully extended to color (vector-valued) image regularization.

2.2. Grayscale image regularization overview

2.2.1. Variation-based regularization methods

Most of the early image regularization algorithms can be generalized as variational methods. Regularizing images is often achieved by minimizing a particular energy functional which measures the overall image variations. The general idea is to preserve only high image variations such as edges while suppressing low image variations which are mainly due to image noises.

Consider a scalar image $I : \Omega \rightarrow \mathbb{R}$ defined on a 2D spatial domain $\Omega \subset \mathbb{R}^2$. A general variational framework is to find I which minimizes the following ϕ -functional:

$$E_{\phi}(I) = \int_{\Omega} \phi(|\nabla I|) dx dy, \quad (2.1)$$

where $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is a monotonously increasing function, which directs the regularization behavior and penalizes high gradients. Equation (2.1) has the

2.2 Grayscale image regularization overview

corresponding Euler-Lagrange equation giving the solution of I when $E_\phi(I)$ reaches its minimum:

$$\frac{\partial \phi}{\partial I} - \frac{d}{dx} \frac{\partial \phi}{\partial I_x} - \frac{d}{dy} \frac{\partial \phi}{\partial I_y} = 0 \quad (2.2)$$

Assuming Neuman boundary condition, the solution can be found by gradient descent method:

$$\begin{cases} I_{t=0} = I_0 \\ \frac{\partial I}{\partial t} = -\frac{\partial \phi}{\partial I} = \text{div} \left(\phi'(|\nabla I|) \frac{\nabla I}{|\nabla I|} \right) \end{cases} \quad (2.3)$$

Note that this Partial Differential Equation (PDE) has an (artificial) time parameter t . It describes the continuous progression of the image I until it minimizes $E_\phi(I)$.

2.2.1.1. Isotropic regularization

One of the earliest variational functional was proposed by Tikhonov in [83] by minimizing the energy functional which measures the square of image gradient norms:

$$E_{Tikhonov}(I) = \int_{\Omega} |\nabla I|^2 dx dy. \quad (2.4)$$

$E_{Tikhonov}$ is a special case of (2.1) when $\phi(s) = s^2$. The original Tikhonov functional also contains a data fidelity term $|I_{noisy} - AI|$ used to restore images filtered by the linear operator A . In this section, since we mainly focus on analyzing different regularization terms' behavior, we will temporarily ignore data fidelity terms and will discuss the effects of data fidelity terms in future sections. The Euler-Lagrange equation gives the following PDE which minimizes $E_{Tikhonov}(I)$:

2.2 Grayscale image regularization overview

$$\begin{cases} I_{t=0} = I_{noisy} \\ \frac{\partial I}{\partial t} = \Delta I \end{cases} \quad (2.5)$$

The PDE (2.5) is actually the famous *heat diffusion equation*, widely used in physics to describe heat flow through solids. This kind of PDE is also called *isotropic diffusion* because it smoothes the image with the same amount in all spatial directions indiscriminately. We can see that the solution for (2.5) without data fidelity term is actually a constant image which has no gradient variation at all.

Koenderink noticed in [55] that the solution at a particular time t is equivalent to convoluting of the original image I_{noisy} with a normalized 2D Gaussian kernel G_σ of variance $\sigma = \sqrt{2t}$:

$$I_t = I_{noisy} * G_\sigma = I_{noisy} * \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.6)$$

From (2.6), we can see that Tikhonov regularization behaves like a low-pass Gaussian filter suppressing high-frequency signal in images. As diffusion time t increases, we will have gradually regularized image I_t with less high-frequency signal. This is the same as the popular *Gaussian scale-space* which creates a multi-scale image representation by convoluting the image with Gaussian kernels of increasing scale σ , more detailed explanation of the linear *scale-space* theory can be found in [57].

The presence of the regularization term $|\nabla I|^2$ in Tikhonov regularization often leads to over smoothing and blurring of the edges. This is because the Dirichlet functional $\int |\nabla I|^2$ penalizes all steep edges while preferring smoother gradients. However, most images contain steep edges, which provide very important perceptual clues, and one would like to retain these edges during the regularization process.

2.2 Grayscale image regularization overview

2.2.1.2. Perona-Malik regularization

To overcome the limitations of linear methods leading to isotropic smoothing, Perona and Malik [74] proposed a nonlinear extension of the heat diffusion equation (2.5). They first reformulate equation (2.5) to the divergence form (1.11), and then inhibit smoothing in edge regions by proposing to add a conductance function $g(|\nabla I|)$ to the diffusion equation:

$$\frac{\partial I}{\partial t} = \operatorname{div} \left(g(|\nabla I|) \nabla I \right), \quad (2.7)$$

where $g: \mathbb{R} \rightarrow \mathbb{R}$ is a monotonically decreasing function which reaches almost 1 in homogeneous regions (low gradients) to allow isotropic-like diffusion; while decreasing to almost 0 on edges (high gradients) to inhibit diffusion. One of the conductance functions they proposed is:

$$g(|\nabla I|) = \exp \left(-\frac{|\nabla I|^2}{k^2} \right), \quad (2.8)$$

where $k \in \mathbb{R}$ is a constant gradient threshold that differentiates homogeneous regions from edge regions. The Perona-Malik regularization can be considered as a special case of the ϕ -functional formulation (2.3) when $\phi(s) = 1 - \exp(-s^2/k^2)$.

To understand the exact local diffusion geometry of the Perona-Malik regularization, a specific decomposition of this equation (2.7) has been proposed in [22, 56]. The authors first defined unit vectors η and ξ to denote local gradient and tangent direction, respectively:

$$\eta = \frac{\nabla I}{|\nabla I|} \quad \text{and} \quad \xi = \frac{\nabla I^\perp}{|\nabla I|}, \quad (2.9)$$

2.2 Grayscale image regularization overview

where unit vector η is the gradient direction with highest grayscale value fluctuation; and ξ is the tangent direction, which is everywhere tangent to the image isophote lines (i.e. lines of constant grayscale value) and points along the local image contour direction as shown in Figure 2-1 below.

Figure 2-1: Image contour and its pointwise defined gradient and tangent direction

The set of orthonormal coordinates basis (ξ, η) gives the local geometry orientation based on the first-order gradient direction. Based on the defined local gradient directions, the authors then derived $I_{\xi\xi}$ and $I_{\eta\eta}$ to denote the second order derivatives of I in orthogonal directions ξ and η , respectively:

$$\begin{cases} I_{\xi\xi} = \frac{\partial^2 I}{\partial \xi^2} = \xi^T \mathbf{H} \xi = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{I_x^2 + I_y^2} \\ I_{\eta\eta} = \frac{\partial^2 I}{\partial \eta^2} = \eta^T \mathbf{H} \eta = \frac{I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}}{I_x^2 + I_y^2} \end{cases}, \quad (2.10)$$

where $\mathbf{H}$ is the Hessian of I as defined in (1.8).

The Perona-Malik regularization (2.7) can be re-decomposed using the newly-defined local coordinate basis (ξ, η) as shown in (2.11).

2.2 Grayscale image regularization overview

$$\begin{aligned}
\frac{\partial I}{\partial t} &= \operatorname{div} \left(g(|\nabla I|) \nabla I \right) = g(|\nabla I|) \operatorname{div}(\nabla I) + \operatorname{div} \left(g(|\nabla I|) \right) \nabla I = \\
&g(|\nabla I|) \Delta I + g'(|\nabla I|) \left(\frac{\partial \left(\sqrt{I_x^2 + I_y^2} \right)}{\partial x}, \frac{\partial \left(\sqrt{I_x^2 + I_y^2} \right)}{\partial y} \right) \begin{pmatrix} I_x \\ I_y \end{pmatrix} = \\
&g(|\nabla I|) (I_{\xi\xi} + I_{\eta\eta}) + g'(|\nabla I|) \frac{I_x^2 I_{xx} + 2I_x I_y I_{xy} + I_y^2 I_{yy}}{\sqrt{I_x^2 + I_y^2}} = \\
&g(|\nabla I|) (I_{\xi\xi} + I_{\eta\eta}) + g'(|\nabla I|) |\nabla I| I_{\eta\eta} = g_\xi I_{\xi\xi} + g_\eta I_{\eta\eta},
\end{aligned} \tag{2.11}$$

where $g_\xi = g(|\nabla I|)$ and $g_\eta = g'(|\nabla I|)|\nabla I| + g(|\nabla I|)$. In the case that the conductance function is defined as (2.8), we have

$$g_\xi = \exp \left(-\frac{|\nabla I|^2}{k^2} \right) \text{ and } g_\eta = \left(1 - 2\frac{|\nabla I|^2}{k^2} \right) \left[\exp \left(-\frac{|\nabla I|^2}{k^2} \right) \right]. \tag{2.12}$$

With (2.11) we can better understand the exact diffusion behavior of the Peona-Malik regularization from the local geometry point of view. From (2.12), it is easy to see that $g_\xi \geq g_\eta$, and image diffusion is mainly directed along the image edge direction ξ , not across the edge. While in homogeneous regions where $|\nabla I| \ll k$, we can see that $g_\xi \simeq g_\eta$, this leads to isotropic-like diffusion which can better remove noise. The actual regularization results of (2.7) were very good: edges were preserved over a very long diffusion time. The authors also proved that edge detection based on this process clearly outperforms the famous linear Canny edge detector [16], even before *non-maximum suppression* and *hysteresis thresholding*.

However, from (2.12) we can see that g_η can be negative when $|\nabla I| > k/\sqrt{2}$. This will introduce inverse diffusion on some image points, possibly high contrast edges or impulse noise. Inverse diffusion is an unstable process which will enhance image

2.2 Grayscale image regularization overview

features such as edges but noise as well. In this sense, the Perona-Malik formulation is ill-posed; however, their experimental results are good and visually pleasant due to sometimes inverse diffusion will enhance image edges like the well-known *shock filter* formalism [2, 42, 70]. A lot of study [31, 50, 75, 97] has been done to analyze the ill-posedness and instability of the Perona-Malik regularization, and the results show that the numerical schemes used provide implicit regularization which stabilize the process; however, the effects that sometimes noise are also preserved and even enhanced still exist.

2.2.1.3. Total Variation regularization

Another famous regularization was the Total Variation (TV) regularization [77] proposed by Rudin et al. to recover noisy blocky images. This algorithm seeks the regularized image $I_{regular}$ by minimizing a proposed energy functional comprised of the TV norm $\int_{\Omega} |\nabla I|$ of the image I and the fidelity of this image to the original noisy image I_0 :

$$E_{TV} = \int_{\Omega} \left(|\nabla I| + \lambda (I - I_0)^2 \right) dx dy \quad (2.13)$$

TV regularization term can also be considered as a special case of the ϕ -functional formulation (2.3) when $\phi(s) = s$. Again, if we omit the data fidelity term and use the same local geometry decomposition method as shown in the previous sub-section, we can rewrite the TV regularization equation (2.13) as:

$$\frac{\partial I}{\partial t} = \frac{1}{|\nabla I|} I_{\xi\xi} \quad (2.14)$$

From this formulation we can see that TV regularization diffuses only along the isophote line direction; and not across edges at all. The amount of diffusion is inversely

2.2 Grayscale image regularization overview

proportional to the gradient norm $|\nabla I|$: in edge regions the diffusion weight is small enough to preserve edges; while in homogeneous regions the weight is high to remove noise and to smooth images.

Total Variation regularization was widely used and extensively studied during the last twenty years, both theoretically and practically. For instance, its well-posedness is proven in [17].

Total Variation regularization allows discontinuities in the image function, which means better edge preserving ability than the Tikhonov regularization term; however, it also has some drawbacks:

- First, the integrand $|\nabla I|$ is not differentiable. Though this can be replaced by $\sqrt{|\nabla I|^2 + \varepsilon}$, where $\varepsilon > 0$ is a small parameter, the resulting Euler-Lagrange equation is still nonlinear and requires sophisticated numerical methods.
- Secondly, although allowing discontinuities in the image function, Total Variation still penalizes each discontinuity proportionally to the height of each jump. Ideal image regularization functional should not punish large jumps (usually edges), at least should not punish them more than small jumps (usually noises).
- Finally, the TV regularized images often have very strong “staircasing” effect on noisy images, not as the designed piecewise constant image model. To reduce this effect, one can adaptively use Total Variation regularization near edges and isotropic smoothing in homogeneous regions; some methods based on this idea were presented in [11].

2.2 Grayscale image regularization overview

2.2.1.4. Summary of variational regularization

From the analysis above, we can see that most variational regularization methods can be generalized to the ϕ -functional formulation (2.3). However, from the ϕ -functional alone, it is difficult to directly understand the exact diffusion behavior and to analyze the edge preserving ability. Like what we did in Section 2.2.1.2 for the Perona-Malik formulation, based on the local orthogonal coordinates (ξ, η) defined in (2.10), we can rewrite the generalized ϕ -functional formulation (2.3) as:

$$\begin{cases} I_{t=0} = I_0 \\ \frac{\partial I}{\partial t} = \operatorname{div} \left(\phi'(|\nabla I|) \frac{\nabla I}{|\nabla I|} \right) = c_\xi I_{\xi\xi} + c_\eta I_{\eta\eta} \end{cases} \quad \text{where} \quad \begin{cases} c_\xi = \frac{\phi'(|\nabla I|)}{|\nabla I|} \\ c_\eta = \phi''(|\nabla I|) \end{cases} \quad (2.15)$$

Although we can select any suitable ϕ -functional to achieve different diffusion behavior, from (2.15) we can see, however, that the two diffusion coefficients c_ξ and c_η are not independent but correlated through the ϕ -function. So at least one degree of freedom is lost here and some specific diffusion behavior is not possible due to this limitation. In next section, we will discuss the efforts to overcome this limitation which makes more sophisticated diffusion behaviors possible.

2.2.2. Gradient direction oriented diffusions

To overcome the limitation of the ϕ -functional formulation, some authors [56] proposed to design a more generic diffusion equation directly based on the local gradient orientation:

$$\frac{\partial I}{\partial t} = c_1 I_{uu} + c_2 I_{vv}, \quad (2.16)$$

2.2 Grayscale image regularization overview

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, $c_1, c_2 > 0$ and $\mathbf{u} \perp \mathbf{v}$ are the local orthogonal coordinate basis. $I_{\mathbf{uu}}$ and $I_{\mathbf{vv}}$ denotes the second order derivatives of I in directions $\mathbf{u}$ and $\mathbf{v}$ respectively and can be expressed as:

$$I_{\mathbf{uu}} = \mathbf{u}^T \mathbf{H} \mathbf{u} \text{ and } I_{\mathbf{vv}} = \mathbf{v}^T \mathbf{H} \mathbf{v} \text{ where } \mathbf{H} \text{ is the Hessian of } I.$$

The regularization process (2.16) can be seen as two orthogonal and weighted 1D oriented heat flows, directed by vectors $\mathbf{u}$ and $\mathbf{v}$. Although technically the diffusion direction $\mathbf{u}$ and $\mathbf{v}$ can be chosen arbitrarily as long as they are orthogonal, practically most researchers still chose the local gradient direction η and local tangent direction ξ to make this regularization process be able to preserve edges. So with $\mathbf{u} = \xi$ and $\mathbf{v} = \eta$, the generic equation (2.16) can be written as:

$$\frac{\partial I}{\partial t} = c_\xi I_{\xi\xi} + c_\eta I_{\eta\eta} \quad (2.17)$$

The biggest difference between (2.17) and the previously mentioned formulation (2.11) is that, unlike g_ξ and g_η in (2.11) which are linked through the common function ϕ , now we can choose two independent c_ξ and c_η . This increases one degree of freedom in designing more specific regularization behavior though the global meaning of energy minimization of the ϕ -functional is lost.

A typical application of this formulation is the mean curvature flow [56, 62], obtained when selecting $c_\eta = 0$ and $c_\xi = 1$:

$$\frac{\partial I}{\partial t} = I_{\xi\xi} \quad (2.18)$$

2.2 Grayscale image regularization overview

The mean curvature flow only smooths along the local edge direction and the isotropic diffusion is never performed, so though edges are preserved the overall denoising performance is not very good.

The authors of [56] also proposed to use a similar diffusivity function as the Perona-Malik regularization [74] to enable isotropic smoothing in low gradients regions; while anisotropic diffusion is used along the edges direction ξ :

$$\frac{\partial I}{\partial t} = I_{\xi\xi} + g(|\nabla I|) I_{\eta\eta} \quad \text{with} \quad \begin{cases} \lim_{|\nabla I| \rightarrow 0} g(|\nabla I|) = 1 \\ \lim_{|\nabla I| \rightarrow +\infty} g(|\nabla I|) = 0 \end{cases} \quad (2.19)$$

Any function $g(|\nabla I|)$ satisfying the requirement in (2.19) can be used to achieve edge-preserving regularization.

Originally to improve the ill-posed Perona-Malik methods (2.7) to a well-posed regularization formulation, Alvarez et al. [1] proposed to use a function $g(|\nabla I * G_\sigma|)$ based on the Gaussian smoothed gradient norm $|\nabla I * G_\sigma|$ instead of the original gradient norm $|\nabla I|$:

$$\frac{\partial I}{\partial t} = \text{div} \left(g(|\nabla I * G_\sigma|) \nabla I \right) \quad (2.20)$$

where $G_\sigma = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$ is a normalized Gaussian kernel of variance σ .

It also allows the possibility of including a larger neighborhood to compute local image geometry which better drives the smoothing process.

2.2.3. Divergence-based regularization methods

In order to better identify image features such as corners and junctions, Weickert [91-92, 94] proposed to include more local geometry information for each investigated point to better direct diffusions. He considered image pixel as chemical concentrations diffusion with respect to *Fick Law* and proposed a generic divergence-based formulation:

$$\frac{\partial I}{\partial t} = \text{div}(\mathbf{D} \nabla I) \quad (2.21)$$

where $\mathbf{D}$ is symmetric and positive semi-definite 2×2 matrix defined for every image pixel (x, y) . It defines a gradient flux and controls the local diffusion behavior of (2.21). The biggest difference from the ϕ -functional formulation (2.3) is that instead of the scalar diffusivity function, now a matrix-valued diffusion tensor $\mathbf{D}$ is used to direct the diffusion behavior. The ϕ -functional formulation (2.3) can be seen as a special case of the divergence-based formulation (2.21) when

$$\mathbf{D} = \frac{\phi'(|\nabla I|)}{|\nabla I|} \mathbf{Id}$$

Weickert [94] then proposed to design the diffusion tensors $\mathbf{D}$ for each image point (x, y) , by selecting its two eigenvectors θ_+ , θ_- and eigenvalues λ_+ , λ_- as functions of the spectral elements of the structure tensor $\mathbf{G}$. The corresponding $\mathbf{D}$ is then computed at each image point as:

$$\mathbf{D} = \lambda_1 \mathbf{u} \mathbf{u}^T + \lambda_2 \mathbf{v} \mathbf{v}^T \quad (2.22)$$

The original structure tensor $\mathbf{G}$ is called the second-order moments tensor and defined as:

2.2 Grayscale image regularization overview

$$\mathbf{G} = (\nabla I \nabla I^T) = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}, \quad (2.23)$$

which has been widely used in corner and junction detection [69]. It is not hard to prove that the gradient direction η and the local tangent direction ξ are the eigenvectors of the structure tensor $\mathbf{G}$. Denoting the eigenvalues of $\mathbf{G}$ as μ_1 and μ_2 , we can also show that

$$\begin{cases} \xi = \frac{\nabla I^\perp}{|\nabla I|} \\ \eta = \frac{\nabla I}{|\nabla I|} \end{cases} \text{ and } \begin{cases} \mu_1 = 0 \\ \mu_2 = |\nabla I|^2 \end{cases}. \quad (2.24)$$

To ensure the well-posedness and to include a slightly larger neighborhood to compute the local image geometry, Weickert [94] proposed not to directly use the structure tensor $\mathbf{G}$ but a Gaussian smoothed structure tensor $\mathbf{G}_\sigma$ instead:

$$\mathbf{G}_\sigma = (\nabla I \nabla I^T) * G_\sigma. \quad (2.25)$$

Again let us consider the eigenvectors ξ^* and η^* , and the eigenvalues μ_1^* and μ_2^* of $\mathbf{G}_\sigma$. We can see that:

$$\begin{cases} \lim_{\sigma \rightarrow 0} \xi^* = \xi = \frac{\nabla I^\perp}{|\nabla I|} \\ \lim_{\sigma \rightarrow 0} \eta^* = \eta = \frac{\nabla I}{|\nabla I|} \end{cases} \text{ and } \begin{cases} \lim_{\sigma \rightarrow 0} \mu_1^* = \mu_1 \\ \lim_{\sigma \rightarrow 0} \mu_2^* = \mu_2 \end{cases}. \quad (2.26)$$

From (2.26) we can see that if we only use a Gaussian kernel of a small variance σ to smooth the structure tensor $\mathbf{G}$, the geometric meaning of its corresponding eigenvectors and eigenvalues are still maintained. Note that alternative method using non-smoothed structure tensor $\mathbf{G}$ can also be found in [41]. So for simplicity, in the

2.2 Grayscale image regularization overview

rest of this thesis, we will commonly use (ξ, η) to denote the eigenvectors of both the smoothed and non-smoothed structure tensor as long as σ is small enough.

μ_1 and μ_2 can be used as local structure descriptors since they contain more local geometry information than the gradient norm $|\nabla I|$ alone :

- For almost constant regions, we should have $\mu_1 \approx \mu_2 \approx 0$
- On image edges, we have $\mu_1 \gg \mu_2 \gg 0$
- On corners and junctions, we should have $\mu_1 \geq \mu_2 \gg 0$

Based on the local geometry information given by the structure tensor $\mathbf{G}_\sigma$, Weickert chose the spectral elements of the diffusion tensor $\mathbf{D}$ as below:

$$\begin{cases} \mathbf{u} = \eta \\ \mathbf{v} = \xi \end{cases} \quad \text{and} \quad \begin{cases} \lambda_1 = \alpha \\ \lambda_2 = \begin{cases} \alpha & (\text{if } \mu_1 = \mu_2) \\ \alpha + (1 - \alpha) \exp\left(\frac{-C}{(\mu_1 - \mu_2)^2}\right) & \text{else} \end{cases} \end{cases} \quad (2.27)$$

where $C > 0$ and $\alpha \in [0, 1]$ are fixed thresholds.

Equation (2.27) was called *coherence-enhancing diffusion filtering* [94] by Weickert. From the local geometry analysis of the structure tensor eigenvalues, we can better understand the reason behind the diffusion tensor design:

- On almost homogeneous regions, we have $\mu_1 \approx \mu_2$, so that $\lambda_1 \approx \lambda_2 \approx \alpha$, the diffusion tensor $\mathbf{D} \approx \alpha \mathbf{Id}$ is almost an isotropic smoothing in these regions.
- On image corners and junctions, we have $\mu_1 \geq \mu_2 \gg 0$; however $(\mu_1 - \mu_2)$ is relatively small so that we still have $\lambda_2 \approx \lambda_1 \approx \alpha$, an isotropic-like smoothing

2.3 Color image regularization overview

will still be performed on corners and junctions so these features are not preserved.

- Only when along the image edges we have $\mu_1 \gg \mu_2$ and $\mu_1 - \mu_2$ is large enough, so that $\lambda_2 > \lambda_1 > 0$. The diffusion tensor $\mathbf{D}$ is anisotropic and mainly directed along the image isophote direction ξ .

Note here λ_1 and λ_2 are actually functions of μ_1 and μ_2 , not the functions of the gradient norm $|\nabla I|$ alone any more. This also increases the degree of freedom. From the analysis above, we can see that with this particular design of diffusion tensor $\mathbf{D}$, only fiber-like features in images are preserved, others non-fiber-like features such as corners and junctions and also noise are smoothed out fast. That is also the reason why this method can enhance the coherence inside images.

2.3. Color image regularization overview

2.3.1. Vector geometry

Before extending the scalar regularization framework to color (vector-valued) images, we first need to define the vector geometry since there is no natural extension for vector gradient for vector-valued images.

Historically, there are several approaches in the regularization process of color and other vector-valued images. The simplest method is to apply a regularization process for each image channel separately. This kind of approach completely ignores the correlation between different channels. Since edges in different channels are not necessarily aligned, an isotropic channel-by-channel process may blur regions where

2.3 Color image regularization overview

only one channel has an edge. In case that several strong edges exist in all channels but with a small offset, artificial colors may appear.

A slightly improved channel-by-channel color image regularization method is the *Color Total Variation (CTV)* proposed by Blomgren and Chan in [11]. The color Total Variation has been defined as the minimization problem below:

$$\min_{\mathbf{I}:\Omega\rightarrow\mathbb{R}^2} \text{CTV}(\mathbf{I}) = \sqrt{\sum_{i=1}^n \left(\int_{\Omega} |\nabla I_i| dx dy \right)^2} \quad (2.28)$$

Minimizing energy functional $\text{CTV}(\mathbf{I})$ using its corresponding Euler-Lagrange equations leads to the following diffusion equations:

$$\frac{\partial I_i}{\partial t} = \frac{\int_{\Omega} |\nabla I_i|}{\sqrt{\sum_{i=1}^n \left(\int_{\Omega} |\nabla I_i| \right)^2}} \text{div} \left(\frac{\nabla I_i}{|\nabla I_i|} \right) \quad (2.29)$$

In (2.28) we can see that the variation in each image channel is computed separately and then combined together by their individual variation contributions with respect to the sum of variations in all channels. This weighted coupling scheme is the only improvement compared with channel by channel approaches; however, for vector-valued images, it would be ideal to calculate variations at each pixel with respect to all image channels and then sum up the variation over all pixels. Furthermore, in (2.29) we can observe that diffusion is performed along the isophote direction of each image channel I_i and these directions can be quite different in different channels. For vector-valued images, it is preferred to have a common isophote direction for vector edges and to perform diffusions in all image channels along this common direction to better preserve the vector geometry.

2.3 Color image regularization overview

In order to overcome the limitations of channel-by-channel approaches and include the correlations between image channels into consideration, Di Zenzo [27] proposed to use a variation matrix $\mathbf{G}$ to measure the vector variations among all image channels. In his original work considering color images $\mathbf{I} = (R, G, B)$, $\mathbf{G}$ is defined as:

$$\mathbf{G} = \begin{pmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{pmatrix}. \quad (2.30)$$

If we further extend Di Zenzo's variation matrix for RGB color images to a more generic N -dimensional vector-valued images, we can have a general definition of $\mathbf{G}$:

$$\mathbf{G} = \sum_{i=1}^n \nabla I_i \nabla I_i^T = \begin{pmatrix} \sum_{i=1}^n I_{ix}^2 & \sum_{i=1}^n I_{ix} I_{iy} \\ \sum_{i=1}^n I_{ix} I_{iy} & \sum_{i=1}^n I_{iy}^2 \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{pmatrix}. \quad (2.31)$$

Note that (2.31) is actually a natural extension of the previously defined structure tensor $\mathbf{G}$ (2.23) for grayscale images when $n = 1$. From (2.31) it is not hard to see that the Di Zenzo structure tensor $\mathbf{G}$ for vector-valued images is also symmetric and semi-positive with its eigenvalues and eigenvectors given by:

$$\begin{cases} \lambda_{+/-} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2} \\ \theta_{+/-} \parallel \begin{pmatrix} 2g_{12} \\ g_{22} - g_{11} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2} \end{pmatrix} \end{cases}. \quad (2.32)$$

Di Zenzo [27] suggested that the eigenvectors $\theta_{\pm}$ of $\mathbf{G}$ give the direction of maximal and minimal changes in a given image point, and the eigenvalues $\lambda_{\pm}$ are the corresponding rates of change. The direction of maximal changes θ_{+} is also called

2.3 Color image regularization overview

vector gradient direction which indicates the direction of the largest vector variation, and θ_- is the direction perpendicular to the vector edges.

Similarly to what we have discussed about the local geometry descriptor of the smoothed structure tensor for grayscale images, eigenvalues $\lambda_{\pm}$ of the structure tensor $\mathbf{G}$ for vector-valued images also can be used to describe local vector geometry:

- When $\lambda_+ \approx \lambda_- \approx 0$, there should be very few vector variations around the given image point and can be considered as almost constant regions without important image features such as edges, corners or junctions.
- When $\lambda_+ \gg \lambda_- \geq 0$, there should be a lot of vector variation at one direction and few on the orthogonal direction; the given image point should be at image edges.
- When $\lambda_+ \geq \lambda_- \gg 0$, there are vector variations in both eigenvector directions, the given image point is located on a saddle point of the vector surface which is probably on vector corners and junctions.

Since there is not direct extension for vector gradient norm, in the literature quite a few vector gradient norms have been proposed for different applications. Based on the above vector geometry analysis, they can generally be classified into three different categories:

- $|\nabla \mathbf{I}| = \sqrt{\lambda_+}$, proposed in [11] to measure maximum variations as an extension of scalar gradient norm, which has high responses for both edges and corners.
- $|\nabla \mathbf{I}| = \sqrt{\lambda_+ - \lambda_-}$, proposed to mainly measure flow-like features by Weickert [94] as a coherence norm, which only has high responses at edge regions but low responses for corners similar to homogeneous regions.

2.3 Color image regularization overview

- $|\nabla \mathbf{I}| = \sqrt{\lambda_+ + \lambda_-}$, proposed in [11, 86] as another kind of extension of scalar gradient norm, which also has high responses for edges and even higher response for certain corner regions.

For the purpose of edge-preserving regularization, both important image edges and corners are of interests and should be preserved, so we are mainly interested in $\sqrt{\lambda_+}$ and $\sqrt{\lambda_+ + \lambda_-}$. Both of them can be used to measure vector edges and corners well, and $\sqrt{\lambda_+ + \lambda_-}$ has even higher responses for corners, which is a desirable property because most regularization processes have a tendency to smooth vector corners first. Besides, $\sqrt{\lambda_+ + \lambda_-}$ is also very computationally efficient as it can be rewritten as:

$$|\nabla \mathbf{I}| = \sqrt{\lambda_+ + \lambda_-} = \sqrt{\mathbf{I}_x \cdot \mathbf{I}_x + \mathbf{I}_y \cdot \mathbf{I}_y} = \sqrt{\sum_{i=1}^n |\nabla I_i|^2} \quad (2.33)$$

Unlike $\sqrt{\lambda_+}$, it does not need the eigenvalue decomposition of $\mathbf{G}$. Based on the above analysis, in this thesis we will normally choose $\sqrt{\lambda_+ + \lambda_-}$ as the vector gradient norm and use $|\nabla \mathbf{I}|$ to denote it.

2.3.2. Vector Φ -functional regularization

Based on different choices of vector gradient norms, it is quite natural to apply the same variation principle for vector-valued images and regularize them by minimizing a functional $\phi(s)$ measuring the overall vector-valued image variations:

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \phi(|\nabla \mathbf{I}|) dx dy \quad (2.34)$$

A solution similar to (2.3) can be found for scalar images regularization using Euler-Lagrange equations:

2.3 Color image regularization overview

$$\frac{\partial I_i}{\partial t} = \text{div} \left(\frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \nabla I_i \right). \quad (2.35)$$

Note now the diffusion is performed in each image channel I_i with exactly the same diffusivity function for all image channels. Unlike the channel-by-channel regularization, the nature of vector geometry is better preserved.

However, one may notice that most of these vector gradient norms can be considered as a function $f(\lambda_+, \lambda_-)$ of the eigenvalues of the structure tensor $\mathbf{G}$. The vector ϕ -functional formulation can also be generalized to a functional $\phi(\lambda_+, \lambda_-)$. We will discuss the difference between the two functionals $\phi(|\nabla \mathbf{I}|)$ and $\phi(\lambda_+, \lambda_-)$ in next section.

2.3.3. Vector gradient oriented and trace-based formulation

Inspired by the gradient directional diffusion for scalar image, its vector version was proposed using the local vector geometry coordinates (θ_+, θ_-) :

$$\frac{\partial I_i}{\partial t} = f_+ (|\nabla \mathbf{I}|) I_{i\theta_+\theta_+} + f_- (|\nabla \mathbf{I}|) I_{i\theta_-\theta_-}, \quad (2.36)$$

where $\theta_{\pm}$ are the eigenvectors of the structure tensor $\mathbf{G}$, $I_{i\theta_+\theta_+} = \theta_+^T \mathbf{H}_i \theta_+$, $I_{i\theta_-\theta_-} = \theta_-^T \mathbf{H}_i \theta_-$, and $\mathbf{H}_i$ is the Hessian of I_i .

For instance, Ringach and Sapiro [79] proposed to extend the grayscale mean curvature flow to vector-valued images:

$$\frac{\partial I_i}{\partial t} = g(\lambda_+ - \lambda_-) I_{i\theta_-\theta_-}, \quad (2.37)$$

2.3 Color image regularization overview

where $g : \mathbb{R} \rightarrow \mathbb{R}$ is a positive decreasing function to avoid smoothing high gradients regions implicitly based on the vector gradient norm $|\nabla \mathbf{I}| = \sqrt{\lambda_+ - \lambda_-}$ which differentiates vector edges from constant regions.

Tschumperle and Deriche further rewrote (2.36) into a generic trace-based formulation and proposed in [88] that its degree of freedom can be increased by constructing the 2×2 diffusion tensor $\mathbf{T}$ independently:

$$\begin{cases} \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i) = f_+(\lambda_+, \lambda_-) I_{i\theta_+ \theta_+} + f_-(\lambda_+, \lambda_-) I_{i\theta_- \theta_-} \\ \mathbf{T} = f_+(\lambda_+, \lambda_-) \theta_+ \theta_+^T + f_-(\lambda_+, \lambda_-) \theta_- \theta_-^T \end{cases}, \quad (2.38)$$

where f_+ and f_- are two independent functions based on two variables λ_+ and λ_- , not the single variable $|\nabla \mathbf{I}|$. Actually, the eigenvalues and eigenvectors of the diffusion tensor $\mathbf{T}$ can also be chosen arbitrarily, but in order to adapt to vector edges direction and preserve vector edges they are normally constructed from the structure tensor $\mathbf{G}$.

Although Tschumperle and Deriche [88] pointed that the degree of freedom can be increased, they still proposed a trace-based regularization formulation which can be reduced to depend only on the vector gradient norm $|\nabla \mathbf{I}|$ based on a smoothed structure tensor $\mathbf{G}_\sigma$:

$$\frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T} \mathbf{H}_i) = \text{trace} \left(\left(\frac{1}{\sqrt{1 + \lambda_+ + \lambda_-}} \theta_+ \theta_+^T + \frac{1}{1 + \lambda_+ + \lambda_-} \theta_- \theta_-^T \right) \mathbf{H}_i \right). \quad (2.39)$$

We can see that given the vector gradient norm $|\nabla \mathbf{I}| = \sqrt{\lambda_+ + \lambda_-}$, the two eigenvalues of the diffusion tensor $\mathbf{T}$ can be rewritten into:

2.3 Color image regularization overview

$$\begin{cases} f_+(|\nabla \mathbf{I}|) = \frac{1}{1+|\nabla \mathbf{I}|^2} \\ f_-(|\nabla \mathbf{I}|) = \frac{1}{\sqrt{1+|\nabla \mathbf{I}|^2}} \end{cases} \quad (2.40)$$

Trace-based formulation is directly originated from the scalar directional diffusion formulation using the vector geometry, but not developed from the vector variation problem. Due to the variational principle is not obeyed, the trace-based formulation has some shortcomings for the vector-valued image, which we will discuss in Chapter 3; however, it also has the advantage that one can precisely control the local diffusion behavior along the vector gradient orientation with this kind of formulation.

2.3.4. Vector divergence-based regularization

Deriving a vector-version of divergence-based regularization from the scalar-version (2.22) is not difficult given the definition of the vector structure tensor $\mathbf{G}$ (2.31). With the eigenvectors $\theta_{\pm}$ and eigenvalues $\lambda_{\pm}$ of $\mathbf{G}$ (or a slightly Gaussian smoothed tensor $\mathbf{G}_{\sigma}$), we can construct a common diffusion tensor $\mathbf{D}$ for all image channels:

$$\mathbf{D} = f_{\mathbf{u}} \mathbf{u} \mathbf{u}^T + f_{\mathbf{v}} \mathbf{v} \mathbf{v}^T = f_+(\lambda_+, \lambda_-) \theta_+ \theta_+^T + f_-(\lambda_+, \lambda_-) \theta_- \theta_-^T, \quad (2.41)$$

where f_+ and f_- are two independently defined functions. Actually the eigenvectors $\mathbf{u}$, $\mathbf{v}$ and eigenvalues $f_{\mathbf{u}}$, $f_{\mathbf{v}}$ can be chosen arbitrarily, but in practice most researchers proposed to design the diffusion tensor $\mathbf{D}$ based on the spectral elements of the vector structure tensor $\mathbf{G}$.

Having defined the common diffusion tensor $\mathbf{D}$, the vector-version of divergence-based regularization is then formulated as:

2.3 Color image regularization overview

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) \quad (2.42)$$

Instead of using a particular vector gradient norm definition, we could further extend the above functional $\phi(|\nabla \mathbf{I}|)$ to a more generic variation functional $\varphi(\lambda_+, \lambda_-)$ which is determined by two variables λ_+ and λ_- rather than a single variable $|\nabla \mathbf{I}|$. Thus, the variational problem for vector-valued images becomes a minimization of a more generic functional $\varphi(\lambda_+, \lambda_-)$:

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \varphi(\lambda_+, \lambda_-) dx dy \quad (2.43)$$

We can then solve it by gradient descend method, derive the PDE from its Euler-Lagrange equation and further develop the PDE into divergence based formulation (detailed proof can be found in [9]):

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) \quad \text{where } \mathbf{D} = 2 \frac{\partial \varphi}{\partial \lambda_+} \theta_+ \theta_+^T + 2 \frac{\partial \varphi}{\partial \lambda_-} \theta_- \theta_-^T \quad (2.44)$$

From the above derivation we can see that the original reduced functional in (2.34) can be rewritten as $\phi(|\nabla \mathbf{I}|) = \phi(\sqrt{\lambda_+ + \lambda_-}) = \varphi(\lambda_+, \lambda_-)$, and we have

$$\frac{\partial \varphi}{\partial \lambda_{\pm}} = \frac{\partial \phi(\sqrt{\lambda_+ + \lambda_-})}{\partial \lambda_{\pm}} = \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{2\sqrt{\lambda_+ + \lambda_-}} \quad (2.45)$$

Combining (2.44) and (2.45) we can get the corresponding diffusion tensor $\mathbf{D}$ below,

$$\mathbf{D} = \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{\sqrt{\lambda_+ + \lambda_-}} \theta_+ \theta_+^T + \frac{\phi'(\sqrt{\lambda_+ + \lambda_-})}{\sqrt{\lambda_+ + \lambda_-}} \theta_- \theta_-^T = \frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \theta_+ \theta_+^T + \frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \theta_- \theta_-^T \quad (2.46)$$

2.3 Color image regularization overview

which is exactly the same as the diffusion tensor $\mathbf{D}$ in the vector ϕ -functional formulation (2.35). Note that in this case $\mathbf{D}$ is an isotropic diffusion tensor (with equal eigenvalues), while for the more generic vector variation functional $\varphi(\lambda_+, \lambda_-)$, the diffusion tensor $\mathbf{D}$ we got is normally anisotropic (with unequal eigenvalues). So this verifies that the divergence-based formulation is more generic and directly related to the global energy minimization principle, vector ϕ -functional formulation is just a special case of this more generic framework.

A practical example of the advantages of the extended functional $\varphi(\lambda_+, \lambda_-)$ is the Beltrami framework [51, 81], which treats the image as a manifold and enhances it by minimizing its area. The energy functional proposed by the authors can actually be simplified and rewritten in the equivalent form defined on the $2D$ domain:

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \sqrt{\det(\mathbf{Id} + \mathbf{G})} dx dy = \int_{\Omega} \sqrt{(1 + \lambda_+)(1 + \lambda_-)} dx dy, \quad (2.47)$$

where $\mathbf{G}$ is the structure tensor defined in (2.31) and $\mathbf{Id}$ is the identity matrix. In the Beltrami framework the functional $\varphi(\lambda_+, \lambda_-) = \sqrt{(1 + \lambda_+)(1 + \lambda_-)}$ is defined independently on the two eigenvalues $\lambda_{\pm}$ of $\mathbf{G}$, not explicitly based on any vector gradient norm $|\nabla \mathbf{I}|$. So the previous vector ϕ -functional formulation cannot be applied here, but the more generic φ -functional formulation (2.44) can be used to derive the corresponding Beltrami flow:

$$\frac{\partial I_i}{\partial t} = \frac{1}{\sqrt{(1 + \lambda_+)(1 + \lambda_-)}} \operatorname{div}(\mathbf{D} \nabla I_i) \text{ where } \mathbf{D} = \sqrt{\frac{1 + \lambda_-}{1 + \lambda_+}} \theta_+ \theta_+^T + \sqrt{\frac{1 + \lambda_+}{1 + \lambda_-}} \theta_- \theta_-^T. \quad (2.48)$$

2.3 Color image regularization overview

Note that the appearance of the diffusion weight $1/\sqrt{(1+\lambda_+)(1+\lambda_-)}$ is mainly due to the gradient descent is computed with respect to the image surface rather than the Euclidean metric; however, the diffusion tensor $\mathbf{D}$ is still the same.

Although introducing a more generic functional $\varphi(\lambda_+, \lambda_-)$ gives us more freedom when designing regularization flow, we find that the two eigenvalues of the diffusion tensor $\mathbf{D}$ are not independent, as they are still linked by the φ functional. Therefore, similar to what was shown in the trace-based formulation, we can go a further step by removing the limitation of the φ functional, and directly design two independent eigenvalues for the diffusion tensor $\mathbf{D}$.

A typical example of this kind of divergence-based formulation can be found in Weickert's coherence enhancing diffusion [94] for vector-valued images, which is not originally developed from the variational principle, but inspired by the field of fluid physics and viewed as diffusion of chemical concentrations:

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) \text{ where } \mathbf{D} = \alpha \theta_+ \theta_+^T + \left(\alpha + (1-\alpha) e^{\frac{-C}{(\lambda_+ - \lambda_-)^2}} \right) \theta_- \theta_-^T \quad (2.49)$$

It is not hard to prove that we cannot find a functional $\varphi(\lambda_+, \lambda_-)$ which can give the above two eigenvalues of $\mathbf{D}$ in the above equation. This example shows that a direct design of the diffusion tensor $\mathbf{D}$ for the divergence-based regularization gives us the possibility of having different regularization behaviors which cannot be realized from the traditional variational framework. By independently selecting two eigenvalues for the diffusion tensor $\mathbf{D}$, we can improve the flexibility of the traditional divergence-based diffusion, so that a better edge-preserving behavior can be expected.

2.3 Color image regularization overview

Finally, we summarize the advantages and disadvantages of in three typical kinds of image regularization methods in Table 2-1 to conclude our review for image regularization terms in subsection 2.2 and subsection 2.3. In the next subsection, we will discuss about the data fidelity terms which is another important component of the overall image regularization framework.

Regularization Methods	Advantages	Disadvantages
Variation-based	Directly derived from the variational principle	Diffusion coefficients not independent, at least one degree of freedom is lost
Gradient direction oriented	Independently designed coefficients, better control of diffusion direction For grayscale image, it is equivalent to the Divergence-based method	For vector-valued images, some vector coupling terms are discarded, which does not obey real vector nature
Divergence-based	Independently designed diffusion tensor, better control of diffusion Variation-based is a special case of the Divergence-based methods	For vector-valued images, cannot control diffusion direction precisely

Table 2-1: Summary of advantages and disadvantages of three main kinds of image regularization framework

2.4. Data fidelity term overview

In the previous two sections we mainly focused on different forms of image regularization terms. One common problem for frameworks with regularization terms only is that most of them will converge to a constant steady-state solution (a constant image without any variations); of course such kinds of solutions are trivial and not of our interests. So these regularization frameworks all require specifying a diffusion stopping time T if one wants to get nontrivial results. In the literature, to avoid getting trivial regularization results sometimes a data fidelity term is added which keeps the steady-state solution closer to the original image.

Data fidelity term is very important for variational image regularization frameworks. Most of the PDE-based image regularization methods can be unified in the variational framework and generally be classified into two major categories according to whether they include image fidelity term in their variational formulations. The first class, e.g. by Perona and Malik [74], and Tschumperle and Deriche [88], only emphasizes on different kinds of edge-preserving regularization terms but does not include the fidelity terms. They have good de-noising ability but their regularization results may deviate too much from the original image especially when the noise level is high (or after a long diffusion time). Furthermore, determining the optimal diffusion stop time is also a difficult problem, some discussion can be found in [38]. The second class, e.g. the classical TV regularization [77] and the color TV [11], has a data fidelity term with constant scalar fidelity weight to balance the regularization term. The existence of a fidelity term can reduce the degenerative effects of regularization; however, how to select a suitable weight for the fidelity term becomes a problem.

The general formulation of such energy functional with both regularization and fidelity terms is defined as:

2.4 Data fidelity term overview

$$\min_{I: \Omega \rightarrow \mathbb{R}} E(I) = \phi(|\nabla I|) + \lambda \psi(I, I_0) \quad (2.50)$$

where $\phi(|\nabla I|)$ is a regularization functional and $\psi(I, I_0)$ is a general data fidelity term with a constant fidelity weight λ . In a statistical framework, the fidelity term $\psi(I, I_0)$ accounts for both noises and distortions between the regularized image I and the original noisy image I_0 .

2.4.1. L^2 -norm based data fidelity term

One of the most widely used data fidelity terms is based on the L^2 -norm, typically the square of the L^2 -norm when $\psi(I, I_0) = |I - I_0|^2 = (I - I_0)^2$ has been used in early works such as Tikhonov regularization [83] and Total Variation restoration [77] etc. to achieve fidelity to the original image. Such data fidelity terms are widely used in denoising, image restoration, deblurring and many inverse problems. Consider a more general ϕ energy functional with a data fidelity term which requires image I to be close to the noisy input image I_0 :

$$E_\phi(I) = \int_{\Omega} \left(\phi(|\nabla I|) + \frac{1}{2} \lambda (I - I_0)^2 \right) dx dy \quad (2.51)$$

Its Euler-Lagrange equation is

$$-F \equiv \operatorname{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) + \lambda (I_0 - I) = 0 \quad (2.52)$$

where $\lambda \in \mathbb{R}$ is a scalar weight which controls the balance between the regularization term and the data fidelity term. Assuming Neumann boundary condition, the solution can be found by a gradient descent method, similarly to the ϕ -functional methods in section 2.2.1:

2.4 Data fidelity term overview

$$\frac{\partial I}{\partial t} = \operatorname{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) + \lambda(I - I_0) \quad (2.53)$$

Note that in practice, adding such a fidelity term shifts the problem of specifying diffusion stopping time T to the problem of determining the suitable fidelity weight λ . Normally the fidelity weight λ is unknown and many authors selected the constant scalar experimentally by trial-and-error; however, the constant λ is not always performing well under different noise conditions and often needs to be adjusted manually to get good results.

The most famous approach to calculate the fidelity weight λ was proposed in the classical Total Variation methods [77] based on the assumption of known image noise variance. When image noise is assumed to be an additive white process of standard deviation σ , the problem can be formulated as finding

$$\min_{I: \Omega \rightarrow \mathbb{R}^2} E(I) = \int_{\Omega} \phi(|\nabla I|) dx dy \quad \text{subject to} \quad \frac{1}{|\Omega|} \int_{\Omega} (I - I_0)^2 dx dy = \sigma^2 \quad (2.54)$$

Note that when the image noise is of the impulse type, this kind of assumption is not suitable anymore. To find the solution I which minimizes $E(I)$ while still satisfying the noise constraint, we need to solve this optimization problem using a Lagrange multiplier λ as shown in (2.51). The Euler-Lagrange (E-L) equation for the variation with respect to I is also shown in (2.52). We can do some further transformation by multiplying the E-L equation (2.52) by $(I - I_0)$ and integrating over the image domain Ω to get

$$\int_{\Omega} \operatorname{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) (I - I_0) - \lambda \int_{\Omega} (I - I_0)^2 = 0 \quad (2.55)$$

2.4 Data fidelity term overview

Together with (2.54), the constant Lagrange multiplier λ for the noise constrained problem is then given by:

$$\lambda = \frac{1}{\sigma^2 |\Omega|} \int_{\Omega} \operatorname{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) (I - I_0) dx dy \quad (2.56)$$

After the above derivation, the noise constrained regularization problem (2.54) is transformed back to the familiar ϕ functional variational formulation with an L^2 fidelity term controlled by the noise variance dependent parameter λ .

Another interesting type of formulation also using L^2 -norm fidelity term is suggested by Mumford and Shah [65] from the image segmentation perspective:

$$E(I, K) = \int_{\Omega \setminus K} (|\nabla I|^2 + \alpha |I - I_0|^2) + \beta \operatorname{length}(K) \quad (2.57)$$

where $I \in C^1(\Omega \setminus K)$, and K is the union of edges in the image, and α and β are constant weights. This choice is suggested by modeling images as piecewise smooth functions with edge sets K . Image variations inside different regions are assumed to be slow and small; while across the boundaries of regions the variations could be very large. This idea is reasonable from the segmentation perspective but a real minimization of the Mumford-Shah functional is difficult both theoretically and practically. This is because it contains both area and length terms and has to be minimized with respect to two different variables I and K . To overcome this difficulty, a weak formulation which approximates the edge set K by an edge strength function v is proposed by Ambrosio and Tortorelli [3]:

$$E(I, v) = \int_{\Omega} \left(v^2 |\nabla I|^2 + \alpha (I - I_0)^2 + \beta \left(\varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) \right) dx dy \quad (2.58)$$

2.4 Data fidelity term overview

The edge strength function v is close to 0 when $|\nabla I|$ is large and 1 otherwise. The corresponding diffusion equations are written as the following coupled system:

$$\begin{cases} \frac{\partial I}{\partial t} = v^2 \Delta I + 2v \langle \nabla v, \nabla I \rangle - \alpha (I - I_0) \\ \frac{\partial v}{\partial t} = 2v |\nabla I|^2 + \beta \left(\frac{v-1}{2\varepsilon} - 2\varepsilon \Delta v \right) \end{cases} \quad (2.59)$$

2.4.2. L^1 -norm based data fidelity term

Another type of data fidelity term is the L^1 -norm with $\psi(I, I_0) = \|I - I_0\|_{L^1}$. This kind of norm is non-smooth, but it is specially effective for removing impulse noises and allowing this kind of outliers to be detected and selectively smoothed as shown in [19, 28, 67-68, 96]. The difference between L^2 -norm and L^1 -norm based fidelity terms, and more generally smooth and non-smooth data fidelity terms has been deeply studied by Nikolova in [66].

An early example of this kind of fidelity term was proposed in [80]. The authors modified the Ambrosio-Tortorelli formulation (2.58) by using a TV-based regularization term $\phi(|\nabla I|) = |\nabla I|$ and an L^1 -norm based fidelity term:

$$E_s(I, v) = \int_{\Omega} \left(\alpha(1-v)^2 |\nabla I| + \beta \|I - I_0\|_{L^1} + \frac{\rho}{2} |\nabla v|^2 + \frac{(v-1)^2}{2\rho} \right) dx dy \quad (2.60)$$

The corresponding evolution equation for (2.60) is

$$\begin{cases} \frac{\partial I}{\partial t} = -2\nabla I \cdot \nabla v + (1-v) |\nabla I| \operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right) - \frac{\beta}{\alpha(1-v)} |\nabla I| \frac{I - I_0}{|I - I_0|} \\ \frac{\partial v}{\partial t} = \Delta v - \frac{v}{\rho^2} + \frac{2\alpha}{\rho} (1-v) |\nabla I| \end{cases} \quad (2.61)$$

2.4 Data fidelity term overview

with $v_{t=0} = 1 - 1/(1 + 2\alpha\rho|\nabla I|)$ is the initial guess for v .

From these examples, we can see that the general formulation of a variational framework with L^1 -norm based fidelity term can be defined as

$$\begin{cases} E_{\phi, L^1}(I) = \int_{\Omega} \left(\phi(|\nabla I|) + \frac{1}{2} \lambda \|I - I_0\|_{L^1} \right) dx dy \\ \frac{\partial I}{\partial t} = \text{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) + \lambda \frac{I - I_0}{|I - I_0|} \end{cases} \quad (2.62)$$

The problem with this new energy functional E_{ϕ, L^1} is that sometimes it is not strictly convex and also lacks unique minimizers, so some authors proposed to use a regularized version fidelity $\sqrt{(I - I_0)^2 + \delta}$ of the L^1 fidelity norm instead; for any given $\delta > 0$, the approximated energy functional is strictly convex and its minimizers enjoy uniqueness. The most important feature of the L^1 fidelity term is that the priority of features it preserves is only determined by the geometry (i.e. size and scale) of the features, and not by the contrast of those features. That also explains why this kind of fidelity term can better remove impulse-types of noise with small scale but very high contrast. The traditional L^2 fidelity term feature preserving priority is decided by both the feature contrast and the feature scale, so when edge-preserving applications are considered, L^2 fidelity term would be a better choice since we want to preserve those high-contrast but small-scale edges as well.

2.4.3. Other fidelity norms

Besides the most commonly used L^2 -norm and L^1 -norm based fidelity terms, there are also other fidelity terms based on different kinds of norms proposed in the literature mainly for piecewise constant “cartoon” image and texture decomposition. Meyer and

2.4 Data fidelity term overview

Haddad [45, 63] proposed to use the G -norm for cartoon-texture decomposition. Vese and Osher [90] approximated Meyer's G -norm by the $\operatorname{div}(L^p)$ -norm. Inspired by these ideas, Osher et al. [71] proposed to use the H^{-1} -norm; Garnett et al. [33] further proposed to use a more general H^{-s} -norm for cartoon-texture decomposition. Most of these norms have the property that high frequency signals like textures, edges and noises are usually much smaller when computed in these norms, so good cartoon-texture image decomposition can be achieved. The norms are not strictly based on the data fidelity paradigm. Furthermore, for our edge-preserving regularization, we want to remove image noises only and trying to preserve even small scale edges/textures as much as possible, so these norms are not considered here.

There are too many previously published papers and books working in this field to be covered in this Chapter, readers are encouraged to refer diffusion related books [57, 82, 93] and other papers [7, 24, 30, 32, 49, 59-60] for more details.

Chapter 3. Locally Adaptive Edge-Preserving Color Image Regularization Framework

Following our previous reviews on both grayscale and color image regularization methods, in this Chapter we will propose a new locally adaptive edge-preserving regularization framework for color images, (or more generically, vector-valued images). The proposed regularization framework is composed of both the adaptive regularization term and the adaptive data fidelity term. We will explain the designs of these two terms in details and compare the regularization performances of the proposed framework with existing approaches.

3.1. Adaptive divergence-based regularization term

In Chapter 2, we have shown that for both scalar and vector-valued image regularization, their regularization terms can be categorized into two major types: the divergence-based formulation or the trace-based formulation.

For the scalar image case, the most commonly used ϕ -functional based variational formulation can be considered as both the special case of the divergence-based (2.21) and trace-based formulation (2.17) when

$$\mathbf{D} = \frac{\phi'(|\nabla I|)}{|\nabla I|} \mathbf{Id} \quad \text{and} \quad \begin{cases} c_{\xi} = \frac{\phi'(|\nabla I|)}{|\nabla I|} \\ c_{\eta} = \phi''(|\nabla I|) \end{cases}.$$

However, for the case of vector-valued image regularization, the vector ϕ -functional based formulation can only be considered as a special case of divergence-based formulation. It can no longer be categorized under the special case of the vector trace-

3.1 Adaptive divergence-based regularization term

based formulation. The vector trace-based formulations are actually the direct extension of the scalar trace-based formulations to the vector version mainly from the local vector geometry point of view, not from the vector variational point of view. In this section, we will compare and analyze the difference between these two formulations.

3.1.1. Comparing divergence-based and trace-based formulations

From the previous reviews, we can see that the designs of diffusion tensors $\mathbf{D}$ and $\mathbf{T}$ are quite similar, they are both based on the eigenvectors $\theta_{\pm}$ of the structure tensor $\mathbf{G}$ or $\mathbf{G}_{\sigma}$ and two independently chosen eigenvalues $f_{\pm}$. However, their regularization behavior is different. In this sub-section, we will compare the difference of the divergence-based and the trace-based formulation in detail and show why the divergence-based formulation is preferred in our proposed regularization framework.

Let us consider a general case and denote the diffusion tensor $\mathbf{D}$ by

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix},$$

where a, b, c are functions $\Omega \rightarrow \mathbb{R}$. Then we can decompose the diffusion behavior of the divergence-based regularization:

$$\begin{aligned} \frac{\partial I_i}{\partial t} &= \text{div}(\mathbf{D} \nabla I_i) = \text{div} \left(\begin{pmatrix} a & b \\ b & c \end{pmatrix} \times \begin{pmatrix} I_{ix} \\ I_{iy} \end{pmatrix} \right) = \text{div} \begin{pmatrix} aI_{ix} + bI_{iy} \\ bI_{ix} + cI_{iy} \end{pmatrix} \\ &= aI_{ixx} + 2bI_{ixy} + cI_{iyy} + I_{ix} \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \right) + I_{iy} \left(\frac{\partial b}{\partial x} + \frac{\partial c}{\partial y} \right) \\ &= \text{trace}(\mathbf{D} \mathbf{H}_i) + I_{ix} \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \right) + I_{iy} \left(\frac{\partial b}{\partial x} + \frac{\partial c}{\partial y} \right) \end{aligned} \quad (3.1)$$

After this simple derivation, we can see that the divergence-based regularization term incorporates a few more diffusion terms than the trace-based one if they use the same

3.1 Adaptive divergence-based regularization term

diffusion tensors $\mathbf{D} = \mathbf{T}$. So for the same diffusion tensor, their diffusion behavior are different unless the requirement below

$$\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} = \frac{\partial c}{\partial y} + \frac{\partial b}{\partial x} = 0 \quad (3.2)$$

is satisfied, (for instance, when $\mathbf{D}$ is a constant diffusion tensor). With this generalized decomposition of the divergence-based formulation, first let us see whether the equation showing the equivalent condition between divergence-based and trace-based formulation (2.15) for scalar image is still valid or not for vector-valued images, consider the below vector ϕ -functional case where

$$\mathbf{D} = \frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \mathbf{Id} \quad \text{and} \quad \begin{cases} a = c = \phi'(|\nabla \mathbf{I}|)/|\nabla \mathbf{I}| \\ b = 0 \end{cases},$$

together with (3.1) we have

$$\begin{aligned} \frac{\partial I_i}{\partial t} &= \text{div}(\mathbf{D} \nabla I_i) = a I_{ixx} + 2b I_{ixy} + c I_{iyy} + I_{ix} \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \right) + I_{iy} \left(\frac{\partial c}{\partial y} + \frac{\partial b}{\partial x} \right) \\ &= \frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} I_{ixx} + \frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} I_{iyy} + \frac{\partial}{\partial x} \left(\frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \right) I_{ix} + \frac{\partial}{\partial y} \left(\frac{\phi'(|\nabla \mathbf{I}|)}{|\nabla \mathbf{I}|} \right) I_{iy} \end{aligned} \quad (3.3)$$

Unlike the scalar image case where $I_{xx} + I_{yy} = I_{\xi\xi} + I_{\eta\eta}$ (2.10), for vector-valued images

$$I_{ixx} + I_{iyy} \neq I_{i\theta_+\theta_+} + I_{i\theta_-\theta_-}, \quad (3.4)$$

because $\theta_{\pm}$ are not the single image channel gradient orientation but the vector gradient orientation of all image channels as defined in (2.32). Note that $I_{i\theta_+\theta_+} = \theta_+^T \mathbf{H}_i \theta_+$ and $I_{i\theta_-\theta_-} = \theta_-^T \mathbf{H}_i \theta_-$ and typically for a unit vector $\mathbf{u} = (u, v)$, we have

$$I_{\mathbf{u}\mathbf{u}} = u^2 I_{xx} + 2uv I_{xy} + v^2 I_{yy} \quad (3.5)$$

3.1 Adaptive divergence-based regularization term

So for the vector case, we are no longer able to rewrite equation (3.3) to the simple trace-based formulation due to the introduction of vector gradient orientation $\theta_{\pm}$. The equivalence equation (2.15) is not valid any more. Trace-based formulation is completely different from the divergence-based formulation, even the most typical vector ϕ -functional formulation cannot be represented by the trace-based formulation either. From equation (3.3), we can see that generally for vector-valued images, the divergence-based and the trace-based formulations will not lead to the same diffusion behavior unless condition (3.2) is met or diffusion tensor $\mathbf{D}$ is constant, which are both very rarely seen. Furthermore, considering the general case of the divergence-based formulation when $\mathbf{D}$ is an anisotropic ($a \neq c$), or is a non-diagonal matrix ($b \neq 0$), it is almost impossible to link the divergence-based formulation to the trace-based formulation using a 2×2 diffusion tensor $\mathbf{T}$ only.

Tschumperle and Deriche [85, 88] proposed to use a complicated hyper-matrix version of trace-based formulation to also generalize the divergence-based regularization:

$$\text{div}(\mathbf{D} \nabla I_i) = \sum_{j=1}^n \text{trace}((\delta_{ij} \mathbf{D} + \mathbf{Q}^{ij}) \mathbf{H}_j) = \text{trace}(\mathbf{D} \mathbf{H}_i) + \sum_{j=1}^n \text{trace}(\mathbf{Q}^{ij} \mathbf{H}_j) \quad (3.6)$$

where $\sum_{j=1}^n \text{trace}(\mathbf{Q}^{ij} \mathbf{H}_j)$ is actually the additional terms $I_{ix} \left(\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \right) + I_{iy} \left(\frac{\partial c}{\partial y} + \frac{\partial b}{\partial x} \right)$ in

(3.1) after complicated transformations to the traced-based formation, and δ_{ij} is the Kronecker's symbol

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}.$$

However, this kind of generalization is very complicated (explicitly across all image channels) and lacks practical meaning to directly design this kind of diffusion tensor

3.1 Adaptive divergence-based regularization term

based on equation (3.6). So basically these two formulations are different in nature and one should choose based on different applications.

These additional terms in (3.1) are the couplings between different vector channels. Tschumperle et al. [88] also noticed the difference between the above two formulations and argued that these vector couplings mix the diffusion contributions from various image channels. They argued that the couplings between vector components I_i should only appear in the diffusion PDEs for the computation of the structure tensor $\mathbf{G}$, so they did not include these terms in their trace-based formulation. They also mentioned that the diffusion is directed not only by the eigenvector of $\mathbf{D}$, so it would be difficult to precisely control the exact diffusion direction by simply designing the diffusion tensor $\mathbf{D}$.

First of all, we believe that the divergence-based formulation is closer to the nature of the variational principle since it can be directly developed from the generic energy functional $\varphi(\lambda_+, \lambda_-)$ minimization as shown in section 2.3.4. Even if we assign more freedom to the eigenvalues of the diffusion tensor $\mathbf{D}$, it is still possible to trace back to the energy functional φ whose gradients give the eigenvectors of $\mathbf{D}$. Unfortunately, for the trace-based formulation, even the classical vector ϕ -functional based formulation cannot be expressed by it. The more generic functional $\varphi(\lambda_+, \lambda_-)$, especially when

$\frac{\partial \varphi}{\partial \lambda_+} \neq \frac{\partial \varphi}{\partial \lambda_-}$, (i.e. the Beltrami Flow [51]), also cannot be minimized by the trace-based

formulation because some important terms in the E-L equations are discarded. Thus, in terms of the variational principle, the divergence-based formulation is more generic than the trace-based one.

3.1 Adaptive divergence-based regularization term

From the local vector geometry perspective, trace-based formulation is diffusing each image channel I_i along the common orientation θ_+ and θ_- , the diffusivity functions f_+ and f_- are also the same for all image channels. Except $\theta_{\pm}$ and $f_{\pm}$, there are no other couplings between different image channels. This actually means that diffusion in each channel is guided by the same orientation and diffusivity; it is also constrained within individual channel only, since no other vector diffusion couplings between different image channel I_i are allowed. This property limits the ability of minimizing the overall vector variations. This is also why some energy functional's minimizer cannot be found by the trace-based method. Tschumperle and Deriche [88] argued that those complex diffusion couplings between image channel I_i in the divergence-based formulation may not be desired for the regularization purpose, however, they did not provide proof or enough experimental results to support this statement. We argue that since traditionally the regularity of the image is measured by the energy functional variation, vector couplings between different image channels which can help to minimize such variations should be allowed to achieve the “overall” most regular image.

Secondly, from the geometric perspective, since color images are considered as vectors in the 3-dimensional vector space, the diffusion process actually works like a gradual adjustment of the vector magnitude and orientation. These adjustments should be done with the coupling between all the vector components (and not only as the channel by channel adjustments). Otherwise, the vector nature of color image will be weakened. Therefore, we argue that necessary vector couplings should be preserved.

Trace-based formulation can be considered as originating from the scalar variational problem but is adapted to a local-geometry point of view. Then it is directly extended to the vector version using the vector geometry only, and not developed from the vector

3.1 Adaptive divergence-based regularization term

variation problem (because some terms are ignored.) The advantage of trace-based formulation is that the exact diffusion behavior is explicitly determined by the diffusion tensor $\mathbf{T}$, not like the divergence-based formulation where the exact diffusion behavior is implicitly decided by the diffusion tensor $\mathbf{D}$; for example, an isotropic $\mathbf{D}$ can implicitly lead to an anisotropic diffusion behavior as it happens in the vector ϕ -functional case. For applications which need diffusions in specific orientation with exact amount of diffusivity, it is easy to use the trace-based formulation to precisely control the local diffusion by choosing a specific diffusion tensor $\mathbf{T}$. However, for the objective of regularizing the image as close to the “true” image as possible, the divergence-based method is a better choice.

Based on our objective of image regularization (vector edges and other important features should be preserved), we propose to use the divergence-based formulation, in particular for variational color image regularization. Experimental results (examples presented in the following sections) also show the improvement, both in the PSNR sense and visually, of our adaptive divergence-based method results over the trace-based formulation results.

3.1.2. Edge indicator function

In order to adaptively preserve vector edges, corners and other important features in the images, the first step we need is to be able to identify such features from the given noisy images. We proposed to use the local vector geometry information based on the vector structure tensor $\mathbf{G}$. We then designed an edge indicator function which can differentiate edges and homogeneous regions. For vector-valued images, we proposed to construct this function base on the local vector geometry of the vector structure tensor $\mathbf{G}$.

3.1 Adaptive divergence-based regularization term

Since we are interested in preserving both vector edges and corners, we choose $\sqrt{\lambda_+ + \lambda_-}$ as our vector gradient norm $|\nabla \mathbf{I}|$, as discussed before. This gradient norm has high responses for vector edges and even higher responses for vector corners, which is a desirable feature for us. We want to emphasize that the selection of the vector gradient norm is not unique and there are other candidates available; for instance, $|\nabla \mathbf{I}| = \sqrt{(1 + \lambda_+)(1 + \lambda_-)}$ as proposed in the Beltrami flow [51] is also possible. After choosing the gradient norm, we then need to normalize it to (0, 1). From a variety of choices, we chose a function similar to one of the diffusivity functions proposed by Perona and Malik in [74]. Our desired property for the edge indicator function is that its value needs to be small (close to 0) at homogeneous regions; while at vector edges and corners, its value should be high (close to 1). Based on this requirement, a possible edge indicator function is introduced:

$$V(\lambda_+, \lambda_-) = 1 - \frac{1}{1 + \frac{\sqrt{\lambda_+ + \lambda_-}}{k}} = \frac{\sqrt{\lambda_+ + \lambda_-}}{\sqrt{\lambda_+ + \lambda_-} + k}, \quad (3.7)$$

where $k \in \mathbb{R}$ is a scalar color gradient threshold and can be set arbitrarily, but normally we use the estimated noise variance σ_e in this thesis to set k automatically. Then the proposed edge indicator function is actually also pointwise defined as:

$$V(x, y) = \frac{\sqrt{\lambda_+ + \lambda_-}}{\sqrt{\lambda_+ + \lambda_-} + \sigma_e}. \quad (3.8)$$

Note that the edge indicator function is not defined as the edge responses of individual channels; instead it measures the vector edges which include all channels, so that local vector geometry is better preserved than by using the channel-by-channel definition.

3.1 Adaptive divergence-based regularization term

Additionally, with the estimated noise variance given, the proposed regularization framework does not need any manually set parameter.

3.1.3. Design of the edge-preserving diffusion tensor

Having chosen the divergence-based regularization formulation which can better reflect the global variational principle, we have to design the edge-preserving eigenvalues for the diffusion tensor $\mathbf{D}$. To better remove noise while preserving edges in the regularization process, the basic idea is that in homogeneous regions, no edges need to be preserved and isotropic-like diffusion is preferred to remove noise efficiently without introducing undesired image structures. In edge regions, however, edges should be carefully preserved, so that diffusion orthogonal to the edge direction should be inhibited and diffusion along the edge direction is preferred. Basically these ideas can be translated into the diffusion coefficient criteria formulated below:

$$\begin{cases} \lim_{|\nabla \mathbf{I}| \rightarrow 0} f_+(\lambda_+, \lambda_-) = \lim_{|\nabla \mathbf{I}| \rightarrow 0} f_-(\lambda_+, \lambda_-) = 1 \\ \lim_{|\nabla \mathbf{I}| \rightarrow \infty} \frac{f_+(\lambda_+, \lambda_-)}{f_-(\lambda_+, \lambda_-)} = 0 \end{cases} \quad (3.9)$$

Practically, one can have many choices for these two functions $f_{\pm}$, as long as the requirement (3.9) is satisfied. For color image regularization, we chose the two diffusion coefficients below based on the previously defined edge indicator function (3.7):

$$\begin{cases} f_+(\lambda_+, \lambda_-) = \frac{1}{(1 + \lambda_+ + \lambda_-)^{\frac{1+V(\lambda_+ + \lambda_-)}{2}}} \\ f_-(\lambda_+, \lambda_-) = \frac{1}{\sqrt{1 + \lambda_+ + \lambda_-}} \end{cases} \quad (3.10)$$

3.1 Adaptive divergence-based regularization term

where $f_-(\lambda_+, \lambda_-)$ is similar to a regularized stable TV regularization coefficient; and we already showed that $0 < V(\lambda_+, \lambda_-) < 1$ and $\lim_{|\nabla \mathbf{I}| \rightarrow 0} V(\lambda_+, \lambda_-) = 1$ in (3.7). Note that

$$\frac{f_+(\lambda_+, \lambda_-)}{f_-(\lambda_+, \lambda_-)} = (1 + \lambda_+ + \lambda_-)^{-\frac{V(\lambda_+ + \lambda_-)}{2}} \quad (3.11)$$

In vector edge or corner regions where $\lambda_+ + \lambda_- \gg 0$ and $V(\lambda_+, \lambda_-) \approx 1$, we can easily show that the edge-preserving requirement in (3.9) is guaranteed. Based on the above two eigenvalues $f_\pm$, our proposed diffusion tensor $\mathbf{D}$ is defined as:

$$\mathbf{D} = \frac{1}{(1 + \lambda_+ + \lambda_-)^{\frac{1+V(\lambda_+ + \lambda_-)}{2}}} \theta_+ \theta_+^T + \frac{1}{\sqrt{(1 + \lambda_+ + \lambda_-)}} \theta_- \theta_-^T, \quad (3.12)$$

where θ_+ and θ_- are the eigenvectors of the structure tensor $\mathbf{G}$ giving the minimal and maximal vector variation directions correspondingly as defined in equation (2.32).

3.1.4. Comparisons of different regularization terms

To compare the regularization performances, especially the edge-preserving abilities, we compare our proposed adaptive regularization term (3.12) with the trace-based regularization method proposed by Tschumperle and Deriche in [88]. As is usually done, the quality of regularization is quantitatively measured by Peak Signal-to-Noise Ratio (PSNR):

$$\text{PSNR} = 10 \log_{10} \frac{255^2 n |\Omega|}{\sum_{i=1}^n \sum_{x,y \in \Omega} (I_i(x,y) - \tilde{I}_i(x,y))^2}, \quad (3.13)$$

where $\mathbf{I}$ and $\tilde{\mathbf{I}}$ denote the regularized and the original clean image respectively, $|\Omega|$ is the area of the spatial image domain and n is the number of color channels.

3.1 Adaptive divergence-based regularization term

Since diffusion by regularization terms only will generally lead to trivial solutions, and no optimal diffusion stopping time is available. To fairly compare these algorithms, since the *clean* image is known, for each algorithm, we let it iterate up to 1000 iterations (which always lead to over-smoothed images) and calculate the PSNR value of the regularized image after each iteration. Then, we select the regularized image with the highest PSNR value from all the 1000 iterated images to represent each method. By doing that, it is fair to all methods and we can compare their true denoising performance. Note that images with the highest PSNR value may not necessarily be visually the best since human sometimes would prefer a slightly over-smoothed image with less PSNR; however, in general they are good to reflect the regularization ability of those selected algorithms.

We look at a relatively simple 282×282 synthetic piece-wise constant color image corrupted with additive zero-mean Gaussian noise with standard deviation $\sigma=80$ in Figure 3-1. One can clearly see that the proposed divergence-based edge-reserving regularization term did preserve the object boundaries better than the trace-based method. In terms of PSNR, the proposed method is also almost 5dB higher. The edges are sharper and the color uniformity looks better as well. If we take a closer look at the residual images, we can roughly see the contour of object edges in the residual of the trace-based method; while for the proposed method, the residual image is mainly composed of noises only and object edges are barely visible. This also shows that using the proposed method, edges are better preserved in the regularized image; they are not filtered out to the residual image.

3.1 Adaptive divergence-based regularization term

(a) Original image

(b) Noisy image ($\sigma=80$)

(c) TD's trace-based method

(d) Residual ($I_0 - I + 100$) of TD

(e) Our proposed method

(f) Residual ($I_0 - I + 100$) of our proposed

Figure 3-1: Regularization results of a synthetic color image corrupted with additive zero-mean Gaussian white noise ($\sigma=80$) using regularization terms only. (a) Original image I ; (b) Noisy image I_0 ($\sigma=80$, PSNR=10.07); (c) TD's trace-based regularization term (PSNR=29.26); (d) The residual image ($I_0 - I + 100$) of (c); (e) Our proposed divergence-based regularization term (PSNR=34.31); (f) The residual image ($I_0 - I + 100$) of (e).

3.2. Adaptive data fidelity term

Having chosen the edge-preserving divergence-based regularization term, we then need to select a suitable data fidelity term to couple with the regularization term in order to get best edge-preserving results.

In section 2.4, we have briefly reviewed fidelity terms on the basis of different norms such as L^2 , L^1 and G norm etc. and their effects under different conditions. We can see that for normal images corrupted with Gaussian or uniform noises, fidelity terms based on L^2 -norm generally give the best regularization results. L^1 -norm based fidelity terms are more suitable for images corrupted with impulse noise; we can choose them when we need to deal with impulse noise. G -norm and other similar norms based data fidelity terms are more suitable for image “cartoon” and texture decomposition; however, generally the piece-wise constant “cartoon” component only is not of our interests. For our objective of edge-preserving regularization, we also want to preserve some textures and details as much as possible while removing most noises. Furthermore, we assume that in most circumstances, image noise can be approximated by Gaussian model, so in the proposed regularization framework, we mainly use L^2 -norm based data-fidelity term.

After we have decided which kind of data fidelity term is more suitable for our regularization objective; how to select a suitable weight for the fidelity term becomes the next task. It is known that the existence of a fidelity term can reduce the degenerative effects of regularization getting trivial results; however, that is based on the assumption that the fidelity terms are of similar magnitude as the regularization terms, or more precisely, the fidelity weights are of suitable magnitude. For example, a very small fidelity weight has almost ignorable effects to balance the regularization term and still possible to result an almost trivial solution. On the contrary, a very large

3.2 Adaptive data fidelity term

fidelity weight can prevent getting constant solutions, but it also limits the denoising ability of the regularization term, the regularization results will be still noisy and close to the original noisy images. Therefore, selecting suitable fidelity weights is of great importance to most of regularization frameworks.

This aspect of the problem was often disregarded, and most of the previously used fidelity weights are constant and mainly decided experimentally by trial and error; there is no satisfactory way to automatically selecting them. Selecting a suitable fidelity weight is never easy; it actually transferred the difficult problem of selecting diffusion stopping time [43, 64, 72] for PDE-based diffusions to selecting suitable fidelity weight. As we have discussed in section 2.4, so far there is no good way to automatically select diffusion stopping time; however, there are some methods to automatically compute fidelity weight λ based on the assumption of known noise variance (i.e. the method proposed in TV restoration [77]). Though we are able to estimate the fidelity weight based on knowledge of image noise variance, a globally constant λ sometimes is not performing well under different noise conditions and often needs to be adjusted manually to get good results. This is because different regions in image have different characteristics and should be treated differently; for example, homogeneous regions should be treated differently from texture regions as suggested by Gilboa et al. in [44] to adaptively select fidelity weight in the application of texture-preserving total variation denoising for grayscale images.

The author suggested that the global noise variance constraint in the traditional TV denoising are not good enough for preserving textures and small scale details, thus the author proposed to use local variance constraints for better performance. The basic idea is to assign relatively smaller fidelity weights to texture regions to inhibit smoothing while for constant regions relatively larger weights are used to ensure enough

3.2 Adaptive data fidelity term

denoising. The authors assumed that the real noise variance is already known as σ^2 , they first used a strong TV pre-filtering with a much higher noise variance constraint ($1.5\sigma^2$) to separate the piece-wise constant (*cartoon*) part of the image with its residuals, then estimated the local variance $P_R(x, y)$ of the residual to identify high variance regions as texture regions:

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \phi(|\nabla I|) dx dy \quad \text{subject to} \quad \frac{1}{|\Omega|} \int_{\Omega} (I - I_0)^2 dx dy = \frac{\sigma^4}{P_R(x, y)}, \quad (3.14)$$

where $P_R(x, y)$ is the local variance of the residual image I_R from the TV pre-filtering. By using the similar transformations as shown in section 2.4.1, the authors derived the pointwise defined adaptive fidelity weight $\lambda(x, y)$ and the corresponding diffusion equation:

$$\begin{cases} \frac{\partial I}{\partial t} = \text{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) + \lambda(x, y)(I - I_0) \\ \lambda(x, y) = \frac{P_R(x, y)}{\sigma^4} \text{div} \left(\frac{\phi'(|\nabla I|)}{|\nabla I|} \nabla I \right) (I - I_0) \end{cases} \quad (3.15)$$

Because strong TV pre-filtering is applied, textures will be filtered together with image noise and included in the residual image I_R and the corresponding local variance $P_R(x, y)$ should be larger than the noise only variance σ^2 . Thus, larger fidelity weights are used in texture regions to reduce the amount of smoothing over these regions so that textures are better preserved.

Like the texture-preserving regularization, when considering our objective of edge-preserving regularization, edge regions should also be treated differently in the data fidelity sense than homogeneous regions. In our adaptive regularization framework, we

3.2 Adaptive data fidelity term

also prefer to use an adaptive data fidelity term, or more specifically, locally adaptive fidelity weights to achieve better edge-preservation during the overall regularization process because a constant fidelity term will limit the edge-preserving performance of the proposed framework.

In the following sub-sections, we will discuss the problem of selecting suitable adaptive fidelity weights and introduce the proposed adaptive edge-preserving fidelity weights.

3.2.1. Adaptive edge-preserving fidelity weight

The objective of our variational framework is to achieve both good noise-removal and edge preservation simultaneously. Most of the previous image regularization methods work by using edge-preserving regularization terms which treat edges and homogenous regions differently. In homogenous regions, isotropic diffusion is used to better remove noise; while in edge regions, anisotropic diffusion is used to inhibit the diffusion orthogonal to the vector edge direction (to better preserve edges). Thus, the key to the success of the edge-preserving regularization terms is treating edge regions and homogenous regions differently.

Since variational framework is composed of both the regularization term and the fidelity term, it is therefore natural to also apply an adaptive edge-preserving fidelity term to better preserve edges. Different fidelity weights should be adaptively assigned to edge regions and homogenous regions. The basic idea is that in edge regions, we want a higher fidelity weight to keep the regularization results closer to original image, while in noisy homogenous regions, relatively lower fidelity weights are needed to reduce the effects of the data fidelity term and to allow a better noise removal performance.

3.2 Adaptive data fidelity term

To adaptively select suitable local values for $\lambda(x, y)$, we have to reliably distinguish between homogenous and edge regions. Actually, this can be achieved by the edge indicator function proposed by us in equation (3.7). The proposed edge indicator function is close to 1 when the probability of edge existence is high and close to 0 otherwise, so we can use it to control the amount of data fidelity applied at different regions.

3.2.1.1. Mean-velocity based edge-preserving fidelity weight

Our first approach, which was presented in [99, 101], is to extend the mean-based fidelity weight previously used in TV regularization to color images and adaptively scaled them using the proposed edge indicator function:

$$\lambda(x, y) = \frac{V(x, y)}{n\sigma_e^2|\Omega|} \sum_{i=1}^n \int_{\Omega} \operatorname{div}(\mathbf{D}\nabla I_i)(I_i - I_{0i})d\Omega \quad (3.16)$$

This scaled mean-based fidelity weight is high in edge regions and low in noisy homogenous regions thanks to the edge indicator function $V(x, y)$, and has achieved very good edge-preserving results. From (3.16), we can see that the magnitude of the fidelity weight in the above formulation is based on the mean of diffusion velocity in all image channels. Then this mean-based fidelity weight is used to scale the amount of diffusions in all channels, here some adaptivity may be lost because theoretically different image channel can have different diffusion velocity and should be scaled accordingly. Furthermore, the mean of diffusion velocity of different channels does not have specific geometry meaning, so in the next sub-section, we will overcome this limitation and propose a channel-wise defined fidelity weight which can better adapt to the local vector geometry. However, the adaptivity is only determined by the edge

3.2 Adaptive data fidelity term

indicator function $V(x, y)$, while the mean-based fidelity term is not channel-wise defined and may not indicate the true adaptive nature of the fidelity term.

3.2.1.2. Channel-wise adaptive edge-preserving fidelity weight

Based on the above consideration, we propose to use a channel-wise formulation instead of the mean-based method for the adaptive fidelity term. First, we extend the scalar noise-constrained regularization to color images as follows:

$$\min_{\mathbf{I}: \Omega \rightarrow \mathbb{R}^n} E(\mathbf{I}) = \int_{\Omega} \varphi(\lambda_+, \lambda_-) d\Omega \quad \text{subject to} \quad \frac{1}{n|\Omega|} \sum_{i=1}^n \int_{\Omega} (I_i - I_{0i})^2 d\Omega = \sigma^2 \quad (3.17)$$

From the vector Euler-Lagrange equations of our variational framework for color images, we can derive the channel-wise fidelity weight:

$$L_i = \text{div}(\mathbf{D}\nabla I_i) - \lambda_i(x, y)(I_i - I_{0i}) = 0 \quad (3.18)$$

By multiplying the Euler-Lagrange equation (3.18) by $(I_i - I_{0i})$ and integrate over the image domain, we can get the formulation of the point-wise fidelity weight:

$$\lambda_i(x, y) = \frac{\text{div}(\mathbf{D}\nabla I_i)(I_i(x, y) - I_{0i}(x, y))}{(I_i(x, y) - I_{0i}(x, y))^2} \quad (3.19)$$

Based on the noise constraint (3.17), we assume that the noise variance is generally uniform in the whole image and thus we can use the estimated noise variance σ_e^2 instead of the denominator of (3.19) as follows:

$$\lambda_i(x, y) = \frac{\text{div}(\mathbf{D}\nabla I_i)(I_i(x, y) - I_{0i}(x, y))}{\sigma_e^2} \quad (3.20)$$

This is a truly point-wise definition and we believe that it can better reveal the local nature of the fidelity term than the previous mean-based method. This kind of point-

3.2 Adaptive data fidelity term

wise fidelity weight has been used by Gilboa et al. together with TV regularization term for adaptive texture-preserving filtering for grayscale images [44]. However, with the TV regularization term, the point-wise fidelity term normally has very large variations in its value range and will cause instabilities in the diffusion process. The authors proposed to use a strong Gaussian smoothing to average the fidelity weights to make them stable, but too strong Gaussian smoothing reduces the advantages of the point-wise approach.

In the proposed regularization framework, we proposed to use the pointwise fidelity weight together with the adaptive divergence-based regularization term to further weight it using the edge indicator function to ensure better edge-preserving abilities:

$$\lambda_i(x, y) = \frac{V(x, y) \operatorname{div}(\mathbf{D} \nabla I_i)(I_i(x, y) - I_{0i}(x, y))}{\sigma_e^2} \quad (3.21)$$

When used together with our divergence-based regularization term, the adaptive fidelity term can better preserve edges; furthermore, it is very stable and needs no or little smoothing.

Note that the noise variance constraint σ^2 has a very important role in the computation of the fidelity weight, and it is normally assumed as known in previous works but in reality we seldom can have this information. So we proposed to use a simple statistical method to estimate the noise variance σ_e^2 from the given noisy input images. We subtracted a mean-filter smoothed image from the original image $\mathbf{I}_0$ to get a residual image $\mathbf{I}_R$, which is supposed to contain most image noise as defined in [35]:

$$I_{Ri}(x, y) = \frac{1}{\sqrt{20}} [4I_{0i}(x, y) - I_{0i}(x-1, y) - I_{0i}(x+1, y) - I_{0i}(x, y-1) - I_{0i}(x, y+1)] \quad (3.22)$$

3.3 Final framework: adaptive regularization term with adaptive fidelity term

Then we can calculate the global variance of $\mathbf{I}_R$ of all image channels by the least mean square method:

$$\sigma_e^2 = \frac{\sum_{i=1}^n \sum_{\mathbf{x} \in \Omega} I_{Ri}^2(\mathbf{x})}{n|\Omega|} - \left[\frac{\sum_{i=1}^n \sum_{\mathbf{x} \in \Omega} I_{Ri}(\mathbf{x})}{n|\Omega|} \right]^2, \quad (3.23)$$

where $\mathbf{x} = (x, y)$ is the spatial coordinates, and $|\Omega|$ is the area of the image domain.

Since $\mathbf{I}_R$ contains mainly the image noises, its variance σ_e^2 is a pretty good estimation of the real noise variance σ^2 for most images. Of course, there are other estimation methods available to use, like the *Median Absolute Deviation (MAD)* based on robust statistics as proposed in [10], which can also be used to get similar noise variance estimation results.

3.3. Final framework: adaptive regularization term with adaptive fidelity term

With all the terms available, we can now present our adaptive edge-preserving regularization framework:

$$\begin{cases} \frac{\partial I_i}{\partial t} = \text{div}(\mathbf{D} \nabla I_i) + \lambda_i (I_i - I_{0i}) \\ \mathbf{D} = \frac{1}{(1 + \lambda_+ + \lambda_-)^{\frac{1+\nu(\lambda_+ + \lambda_-)}{2}}} \theta_+ \theta_+^T + \frac{1}{\sqrt{(1 + \lambda_+ + \lambda_-)}} \theta_- \theta_-^T \end{cases} \quad (3.24)$$

Assume $\mathbf{I}^t$ as $\mathbf{I}$ at a specific discrete PDE iteration t of (3.24), the next iterated image $\mathbf{I}^{t+1}$ is computed by the steps listed below:

3.3 Final framework: adaptive regularization term with adaptive fidelity term

- Initialization and estimation: For the original image $\mathbf{I}_0$, first compute the pseudo residual image $\mathbf{I}_R$ using equation (3.22), then estimate the variance of $\mathbf{I}_R$ as the global noise variance σ_e^2 as shown in (3.23).

Compute the smoothed vector structure tensor $\mathbf{G}^*$: the structure tensor $\mathbf{G}^t$ for

- image $\mathbf{I}^t$ is computed as below:

$$\mathbf{G}^t = \begin{pmatrix} \sum_{i=1}^n I_{ix}^{t^2} & \sum_{i=1}^n I_{ix}^t I_{iy}^t \\ \sum_{i=1}^n I_{ix}^t I_{iy}^t & \sum_{i=1}^n I_{iy}^{t^2} \end{pmatrix} = \begin{pmatrix} g_{11}^t & g_{12}^t \\ g_{12}^t & g_{22}^t \end{pmatrix},$$

where the first-order spatial derivatives I_{ix}^t and I_{iy}^t are computed using the classical central difference schemes:

$$\begin{cases} I_{ix}^t = 0.5 \times (I_i^t(x+1, y) - I_i^t(x-1, y)) \\ I_{iy}^t = 0.5 \times (I_i^t(x, y+1) - I_i^t(x, y-1)) \end{cases}.$$

Following the ideas of Weickert's methods [92, 94], we also use a 2D normalized Gaussian kernel G_σ with a very small σ to smooth $\mathbf{G}$ and get the Gaussian smoothed vector structure tensor $\mathbf{G}^*$. This kind of Gaussian smoothing can give us more coherent diffusion geometry by removing some small-scale noise and also make it mathematically convex:

$$\mathbf{G}^{t*} = \mathbf{G}^t * G_\sigma.$$

- Construct the diffusion tensor $\mathbf{D}$: Compute the eigenvalues $\lambda_\pm^t$ and eigenvectors $\theta_\pm^t$ of $\mathbf{G}^{t*}$, then calculate $f_\pm^t$ based on $\lambda_\pm^t$ and σ_e as shown in (3.10) as the new eigenvalues for $\mathbf{D}^t$, and construct the diffusion tensor $\mathbf{D}^t$ using eigenvectors $\theta_\pm^t$:

$$\mathbf{D}^t = f_u^t \mathbf{u}^t \mathbf{u}^{tT} + f_v^t \mathbf{v}^t \mathbf{v}^{tT} = f_+^t(\lambda_+^t, \lambda_-^t) \theta_+^t \theta_+^{tT} + f_-^t(\lambda_+^t, \lambda_-^t) \theta_-^t \theta_-^{tT} = \begin{pmatrix} D_{11}^t & D_{12}^t \\ D_{12}^t & D_{22}^t \end{pmatrix}.$$

3.3 Final framework: adaptive regularization term with adaptive fidelity term

- Compute the PDE diffusion velocity $\mathbf{V}_R$ contributed by regularization term $\text{div}(\mathbf{D}\nabla I_i)$. Using equation (3.1), we can get

$$\text{div}(\mathbf{D}^t \nabla I_i^t) = D_{11}^t I_{ixx}^t + 2D_{12}^t I_{ixy}^t + D_{22}^t I_{iyy}^t + I_{ix}^t \left(\frac{\partial D_{11}^t}{\partial x} + \frac{\partial D_{12}^t}{\partial y} \right) + I_{iy}^t \left(\frac{\partial D_{22}^t}{\partial y} + \frac{\partial D_{12}^t}{\partial x} \right),$$

$$\text{where } \begin{cases} I_{ixx}^t = I_i^t(x+1, y) + I_i^t(x-1, y) - 2I_i^t(x, y) \\ I_{iyy}^t = I_i^t(x, y+1) + I_i^t(x, y-1) - 2I_i^t(x, y) \\ I_{ixy}^t = 0.25 \times (I_i^t(x+1, y) + I_i^t(x-1, y) + I_i^t(x, y+1) + I_i^t(x, y-1)) \end{cases}.$$

Since diffusion tensor $\mathbf{D}^t$ is also computed at every (x, y) , we can get the first spatial derivative of D_{ij}^t using the same central difference schemes:

$$\begin{cases} \frac{\partial D_{ij}^t}{\partial x} = 0.5 \times (D_{ij}^t(x+1, y) - D_{ij}^t(x-1, y)) \\ \frac{\partial D_{ij}^t}{\partial y} = 0.5 \times (D_{ij}^t(x, y+1) - D_{ij}^t(x, y-1)) \end{cases}.$$

- Compute the adaptive edge-preserving fidelity weights: first calculate the diffusion velocity introduced only by the regularization term using the divergence-based formulation; then we can get the adaptive fidelity weight $\lambda_i(x, y)$ as shown in (3.21). Since we have already computed the regularization term in previous step, we can directly use it without need to compute again. Thus, we can get the discrete implementation of the fidelity term:

$$V_{Fi}^t(x, y) = \lambda_i^t(x, y) (I_i^t(x, y) - I_{0i}(x, y)) = \frac{V^t(x, y) \text{div}(\mathbf{D}^t \nabla I_i^t) (I_i^t(x, y) - I_{0i}(x, y))^2}{\sigma_e^2}.$$

- Integrate and regularize iteratively: Finally we can combine the diffusion velocity from regularization term and fidelity term together to get the overall diffusion velocity as the discrete version of (3.24):

3.4 Experimental results

$$\mathbf{I}^{t+1} = \mathbf{I}^t + (\mathbf{V}_R^t + \mathbf{V}_F^t) dt = \mathbf{I}^t + \mathbf{V}_{R+F}^t dt \quad (3.25)$$

By following the computation loop above, the image can be regularized iteratively until the PDE diffusion velocity magnitude is small enough or the variance of the residual image has been close to the pre-estimated noise variance σ_e^2 .

3.4. Experimental results

In this section we present numerical results by applying the proposed regularization framework to color image denoising. For comparisons, we also present results from some typical previous methods. We choose the divergence-based Beltrami Flow (2.48) proposed by Kimmel et al. in [51], and the trace-based regularization framework (2.39) proposed by Tschumperle and Deriche (TD) in [88]. Note that these two approaches are proposed without data fidelity terms. We also select a vector Total Variation (Vector TV) regularization as generalized by Brook et al. in [81] from the classical grayscale TV [77] for color images with data fidelity term for our comparisons. As we have discussed previously, compared with the channel-by-channel Color TV [11], vector geometry is better preserved by Vector TV, so we prefer to use the Vector TV which indicates the true vector nature better.

To quantitatively assess the regularization performance, we still use Peak-Signal-Noise-Ratio (PSNR) defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2 n |\Omega|}{\sum_{i=1}^n \sum_{x,y \in \Omega} (I_i(x,y) - \tilde{I}_i(x,y))^2} \quad (3.26)$$

3.4 Experimental results

where $\mathbf{I}$ and $\tilde{\mathbf{I}}$ denote the regularized and the original clean image respectively, $|\Omega|$ is the area of the spatial image domain and n is the number of color channels.

For the two methods without data fidelity term, typically they will not converge as those with data fidelity terms. As stated in section 3.1.4, to be fair for all methods, we let them iterate up to 1000 iterations (which always lead to over-smoothed images) and calculate the PSNR values of the regularized image after each iteration. Then, we select the regularized image with the highest PSNR value from all the 1000 iterated images to represent each method. By doing that, it is fair to all methods and we can compare their true denoising performance.

We use the classical explicit central schemes for all the gradient and PDE computations:

$$\begin{cases} I_{ix} = 0.5 \times (I_i(x+1, y) - I_i(x-1, y)) \\ I_{iy} = 0.5 \times (I_i(x, y+1) - I_i(x, y-1)) \end{cases} \quad (3.27)$$

Actually, more complex first derivative computation schemes could be used here, for instance the methods proposed in [26, 95].

For the Vector TV method, the non-differentiability of the total variation terms in the energy needs some sort of regularization, thus in our numerical implementation, we use such regularized total variation energy defined as the following:

$$E_{TV}(\mathbf{I}) = \int_{\Omega} \sqrt{|\nabla \mathbf{I}|^2 + \varepsilon} dx dy + \lambda \int_{\Omega} (\mathbf{I} - \mathbf{I}_0)^2 dx dy \quad (3.28)$$

This type of regularization of total variation energy is very standard, and in our implementation we use $\varepsilon = 10^{-4}$, which is small enough while still can make the approximated total variation energy strictly convex so that it has a unique minimizer.

3.4 Experimental results

The constant fidelity weight λ is automatically calculated using the noise variance σ_e^2 estimated by (3.23).

Except the Vector TV method, we use a Gaussian smoothed structure tensor $\mathbf{G}_\sigma$ with a small $\sigma=0.5$ for all three other methods to ensure well-posedness as suggested by Weickert in [92]. Most of the images we presented are available online from the USC SIPI test image database [89] and the Kodak Lossless True Color Image Suite [54].

In Figure 3-2, we present the regularization results of the 256×256 *House* image with additive white Gaussian noise $\sigma = 40$. We can see that the original image has some fine textures which have been almost completely corrupted in the noisy version. The Vector TV method (c) preserves edges well; however the *staircase* effects are noticeable as we can see small color patches instead of uniform color. In (d), the Beltrami Flow also preserves edges well thanks to the weight $1/\sqrt{(1+\lambda_+)(1+\lambda_-)}$, which turns very small at edges thus quickly reduces the amount of diffusions near image edges. However, we can see that this kind of weight also makes this method susceptible to high noises as some noises are not completely removed. Relatively, TD's trace-based approach (e) does not preserve edges very well (i.e. the windows) and color uniformity is not so good either. In (f), the proposed model shows the best results both visually and in terms of PSNR: edges are well preserved and the homogeneous regions also look quite uniform.

3.4 Experimental results

(a) Original image

(b) Noisy image ($\sigma=40$)

(c) Vector TV

(d) Beltrami Flow

(e) TD's trace-based method

(f) Our proposed method

Figure 3-2: Regularization results of the 256x256 *House* image corrupted by additive Gaussian noise ($\sigma=40$). (a) Original image. (b) Noisy image ($\sigma=40$, PSNR=16.10dB); (c) Vector TV (PSNR=28.30dB); (d) Beltrami Flow (PSNR=28.20dB); (e) TD's trace-based method (PSNR=28.69dB); (f) Our proposed method (PSNR=29.54dB).

3.4 Experimental results

In Figure 3-3 and Figure 3-4, we present the regularization results of the 512×512 *Lena* image with additive white Gaussian noise $\sigma = 20$ and $\sigma = 40$ respectively. In Figure 3-3, we can see that for relatively lower noise level, most of the four methods perform very well though the Vector TV method still shows some slight *staircase* effects. The Beltrami Flow gives better results than TD's trace-based method when the noise level is low. The proposed method still yields the highest PSNR value compared with other methods. When noise level increases to $\sigma = 40$ in Figure 3-4, we can see that TD's trace-based method does not preserve edges well any more as can be seen from the eyes and hair in (e), homogeneous regions are not smooth enough as well. The Beltrami Flow preserves edges better but suffers from high noise as some high variations caused by image noise are not removed. The proposed approach overcomes the problem of the Beltrami Flow and preserves edges well while still being able to remove most of the noise. Visually edges are of higher contrast as well compared with TD's trace-based results. In terms of PSNR, the two methods with data fidelity terms get higher PSNR values, this also shows the importance of data fidelity term especially when noise level is high.

In Figure 3-5, regularization results of a real 512×768 color photograph *Lighthouse* from the Kodak image database with additive white Gaussian noise $\sigma = 40$ are presented. This image is a bit difficult because it contains both textures (grass field) and relative constant regions (houses and sky). Again our proposed method gets the highest PSNR value, visually also achieving a good balance between edge preservation and noise removal. The Vector TV method also preserves edges very well, but again visually it suffers from severe *Staircase* effects. TD's trace-based method does not preserve edges well, for example, the window frame of the lighthouse becomes rounded. Furthermore, the grass field looks a bit blended with fiber-like features.

3.4 Experimental results

In Figure 3-6, we show the regularization results of a highly-noised 512×512 *Peppers* image (PSNR=10.08) corrupted with additive white Gaussian noise $\sigma=80$. We can see that when noise level is high, the Beltrami Flow again suffers from not enough smoothing at some noisy regions. TD's trace-based formulation does not perform too well either as some edges in the top-left corner are blurred; furthermore, in homogeneous regions some color blending can be observed. In our opinion, this is due to some necessary vector coupling between image channels are not included in the trace-based formulation. Again the two divergence-based formulations with data fidelity terms give better PSNR values.

Finally in Figure 3-7, the regularization results of a real noisy image taken by a digital camera using 3200 ISO is shown. This time we cannot judge their performance by PSNR because the “ground truth” image is not available. For those methods that cannot stop automatically I have to manually stop them and select the visually best image. The noise level is not very high compared with the previous cases of using synthetic noise; all the selected methods can produce a good denosing performance for it. It is a bit difficult to compare among, but we can still see the Vector TV shows some alight *staircase* effects which can be seen from the color uniformity. The proposed method still kept a good balance between noise removal and edge preservation. This also shows that using additive Gaussian noise as an approximation for real world noise is valid at least for this case.

We also want to briefly discuss about the computational efficiency. For most PDE-based methods, due to the iterative computation of the PDEs, they generally need longer time than the traditional non-iterative based methods. Depending on image size, number of iterations, noise level and time step dt etc., their computation time also varies. The most time consuming computation is that for each iteration it has to loop

3.4 Experimental results

through each pixel. So consider a k dimension $M \times N$ multi-valued image $\mathbf{I}$, the basic scale of the computation is proportional to the total number of pixels multiplied by the vector dimension which is $O(M \times N \times k)$. And to construct the diffusion equation for each pixel, up to the second order spatial derivatives are needed, so it can be considered as a 5×5 local mask is applied throughout the image. The number of iterations needed dominates the computation time; and it is directly related to the overall diffusion velocity $\mathbf{V}_{R+F}$ and the discrete time step dt as shown in (3.25). The higher the overall changes each iterative step is allowed, the less iteration number is required. However, the overall changes cannot be unlimited high otherwise it will cause the PDE instable thus lead to trivial results. The proposed framework has both diffusion velocity contributed by the regularization terms and the fidelity terms, those two are typically opposite to each other, so the overall diffusion velocity is smaller than those use diffusion velocity by the regularization terms only, like TD's trace based method. This may look like a disadvantage; however the interactions of those two terms make the overall diffusion velocity more stable, thus we can use a larger step time without the risk of unstable and getting trivial results. Furthermore, we have shown that our framework has the best PSNR even when noise level is high. Another important parameter is the noise level, for highly noisy images typically more iterations are needed to regularize them to remove noise and make the overall image smooth and regular.

In Table 1, we compare the CPU time needed for all the four methods; they are all implemented by C++ and run on a HP nc6400 laptop with Intel T7200 CPU (2.0GHz) and 1G RAM memory. We use the standard Lena image of two different sizes and noise level for comparisons of computational time; for the 512×512 Lena image, the CPU

3.4 Experimental results

time is recorded when they reached the optimal PSNR as we shown previously in this section for denoising performance comparisons.

Lena Size	Noise Level	Vector TV	Beltrami	TD	Proposed
256×256	$\sigma = 20$	9.27s	10.04s	3.30s	4.48s
256×256	$\sigma = 40$	27.98s	40.78s	10.69s	9.67s
512×512	$\sigma = 20$	63.91s	65.19s	17.80s	17.10s
512×512	$\sigma = 40$	232.34s	231.31s	51.34s	46.28s

Table 3-1: Comparison of CPU time in seconds for 4 methods for image of different sizes and different noise level.

From Table 3-1, we can see that the proposed method is among the top of compared methods. When noise level is low, the CPU time needed is very similar to TD's trace-based method; however when noise level and image size are increased the proposed method started showing some slight advantages thanks to the adaptive framework. Though for each iteration, the proposed method will take longer to compute the edge indicator function and fidelity term etc., the overall computation time is still slightly better. The Beltrami flow is a bit slow mainly because of the overall diffusive weight $1/\sqrt{(1+\lambda_+)(1+\lambda_-)}$ as shown in (2.48) which greatly reduced the overall diffusion speed especially when noise level is high. While the Vector TV method is slower due to the non-adaptive fidelity term reduced the overall diffusion velocity especially when closer to the optimal PSNR. The experimental results also confirm that for the proposed method, the CPU time needed is proportional to image size for the same noise level. The CPU time to regularize a 512×512 Lena image is about 4 times the CPU time to

3.4 Experimental results

process a 256×256 Lena image. So for a megapixel image, when noise level is not high (below $\sigma = 20$), we can expect roughly 68 seconds to finish regularization.

Another important feature I want to mention is about the automatic stop of the PDE diffusion framework. This is also a drawback of many PDE base methods, especially those without fidelity terms like TD's trace-based method and the Beltrami flow. In real application where no "ground truth" image is available so there is no way to compute the best PSNR, one has to manually stop diffusion before it over smooths the image or save the image after each iteration and select visually the best one. Neither of these two methods is convenient and satisfactory. Some study has been done to find the optimal stopping time as shown in [43, 64, 72], they typically stop the diffusion when the diffusion velocity is small enough or compare the correlation between signal and noise. For the proposed framework, since we already estimated the noise variance σ_e^2 in our computation, we then compute the residual image's variance σ_r^2 after each iteration and stop the diffusion when $\sigma_r^2 \geq 0.9\sigma_e^2$. Using this auto stop scheme, we can reach PSNR quite close to the optimal PSNR and can avoid the necessity of human intervention in practical applications.

3.4 Experimental results

(a) Original image

(b) Noisy image ($\sigma=20$)

(c) Vector TV

(d) Beltrami Flow

(e) TD's trace-based method

(f) Our proposed method

Figure 3-3: Regularization results of the 512x512 *Lena* image corrupted by additive Gaussian noise ($\sigma=20$). (a) Original image; (b) Noisy image ($\sigma=20$, PSNR=22.12dB); (c) Vector TV (PSNR=31.10dB); (d) Beltrami Flow: (PSNR=31.45dB); (e) TD's trace-based method (PSNR=31.28dB); (f) Our proposed algorithm (PSNR=31.89dB).

3.4 Experimental results

(a) Noisy image ($\sigma=40$)

(b) Vector TV

(c) Beltrami Flow

(d) TD's trace-based method

(e) Our proposed method

Figure 3-4: Regularization results of the 512x512 *Lena* image corrupted by additive Gaussian noise ($\sigma=40$). (a) Noisy image ($\sigma=40$, PSNR=16.10dB); (b) Vector TV (PSNR=28.70dB); (c) Beltrami Flow (PSNR=28.59dB); (d) TD's trace-based method (PSNR=28.42dB); (e) Our proposed algorithm (PSNR=29.47dB).

3.4 Experimental results

(a) Original image

(b) Noisy image ($\sigma=40$)

(c) Vector TV

(d) Beltrami Flow

3.4 Experimental results

(e) TD's trace-based method

(f) Our proposed method

Figure 3-5: Regularization results of the 512x768 *Lighthouse* image corrupted by additive Gaussian noise ($\sigma=40$). (a) Original image; (b) Noisy image ($\sigma=40$, PSNR=16.10); (c) Vector TV (PSNR=25.62dB); (d) Beltrami Flow (PSNR=26.53dB); (e) TD's trace-based method (PSNR=26.59dB); (f) Our proposed method (PSNR=27.39dB).

(a) Original image

(b) Noisy image ($\sigma=80$)

3.4 Experimental results

(c) Vector TV

(d) Beltrami Flow

(e) TD's trace-based method

(f) Our proposed method

Figure 3-6: Regularization results of the 512x512 *Peppers* image corrupted by additive Gaussian noise ($\sigma=80$). (a) Original image; (b) Noisy image ($\sigma=80$, PSNR=10.08); (c) Vector TV (PSNR=25.59); (d) Beltrami flow (PSNR=24.97); (e) TD's method (PSNR=24.92); (f) Our proposed method (PSNR=26.58).

3.4 Experimental results

(a) Noisy image (ISO3200)

(b) Vector TV

(c) The Beltrami Flow

(d) TD's trace-based method

(e) Our proposed method

Figure 3-7: Regularization results of a real noisy image (taken by DC at ISO3200). (a) Noisy image (ISO 3200); (b) Vector TV; (c) Beltrami flow; (d) TD's trace-based method; (e) Our proposed method

3.5. Conclusion

In this Chapter, we have proposed an adaptive edge-preserving regularization framework for vector-valued (color) image denoising and restoration. The proposed framework is composed of both an adaptive edge-preserving regularization term using the generic divergence-based formulation together with the proposed edge indicator function; and an L^2 -norm based data fidelity term together with an adaptively computed edge-preserving fidelity weight.

We have also presented the numerical results for color image denoising of the proposed framework comparing with the classical Vector TV method, the Beltrami Flow framework and the trace-based formulation. The regularization results obtained by our proposed framework are improved, both visually and quantitatively (in terms of PSNR) over those selected methods; important image features like edges and corners are better preserved while in homogeneous regions noise are better removed and colors are kept uniform as well. In terms of computational time, the proposed method is also among the best of those selected methods.

An important thing to notice is that our framework proposed in this Chapter is mainly optimized for color images corrupted with non-impulse noises such as Gaussian and uniform noises. For impulse noises such as salt-and-pepper noises, our model is not suitable mainly because the selected L^2 -norm based fidelity term is not suitable for removing impulse noises and the proposed edge indicator function cannot distinguish impulse noise from real image features like edges and corners. In the next Chapter, we will present a modification of our proposed framework based on L^1 -norm based fidelity term and an *impulse detection* scheme which can remove impulse noise or the mixture of impulse and Gaussian noises quite well.

Chapter 4. Two-Phase Extension of the Proposed Regularization Framework for Color Impulse and Mixed Noise Removal

In Chapter 3, we have presented our adaptive edge-preserving regularization framework and showed that the framework is mainly based on the assumption that image noise is additive Gaussian noise, not impulse noise. However, in practical systems image noise cannot always be modeled as Gaussian, sometimes they are heavy-tailed and of impulse nature. For instance, *salt-and-pepper noise* is perhaps the most typical impulse noise; it can be caused by bit errors during signal transmission or malfunction of imaging system etc. *Random-valued impulse noise* is less common in practical applications but more difficult to remove than the salt-and-pepper noise. Moreover, in reality, mixture of different types of noises, for instance, mixed Gaussian and impulse noises are also observed due to noise corruption at different stages of the image capturing flow. This kind of mixed noise will cause a lot of difficulties for most regularization frameworks which normally only consider Gaussian noise. Impulse noise corrupted pixels are often misinterpreted as image features and preserved in the regularized images.

To overcome this problem, in this Chapter, we will modify and extend our previously proposed regularization framework to handle both impulse noise and mixed Gaussian and impulse noises in color images.

Before introducing the possible ways to deal with impulse noise, we will first define the color impulse noise models which will be used in this thesis. Impulse noise are commonly considered as outliers in the image, and often modeled as *salt-and-pepper noise* or *random-valued impulse noise*:

4.1 Impulse noise removal by the proposed framework with L1-norm based fidelity term

The model of *salt-and-pepper* noise used in this thesis for a color image $\mathbf{I}$ with its pixel value's dynamic range $[d_{\min}, d_{\max}]$ is given by:

$$\tilde{I}_i(x, y) = \begin{cases} d_{\min} & \text{with probability } s/2 \\ d_{\max} & \text{with probability } s/2 \\ I_i(x, y) & \text{with probability } 1-s \end{cases}, \quad (4.1)$$

where s defines the level or percentage of *salt-and-pepper* noise.

The model of *random-valued* impulse noise used in this thesis for a color image $\mathbf{I}$ is:

$$\tilde{I}_i(x, y) = \begin{cases} d_{xy} & \text{with probability } r \\ I_i(x, y) & \text{with probability } 1-r \end{cases}, \quad (4.2)$$

where d_{xy} is a uniformly distributed random value in image's dynamic range $[d_{\min}, d_{\max}]$ and r determines the level or percentage of the *random-valued* impulse noise.

Having defined two types of color impulse noise which will be used in this thesis, we can then discuss how to modify previously proposed regularization framework to remove them and the mixture of them together with Gaussian noise as well.

4.1. Impulse noise removal by the proposed framework with L^1 -norm based fidelity term

As we reviewed in section 2.4.2, L^1 -norm based fidelity term is very good for removing impulse noise and quite a few of applications [18-19, 28, 66-67, 96] have been proposed to remove impulse noises using L^1 -norm based fidelity term. Another kind of framework was proposed by Bar et al. [4-6] for deblurring color images corrupted by

4.2 Two-phase extension of the proposed framework for color impulse noise removal
impulse noise, whose energy functional is composed of the Mumford-Shah regularization term and a L^1 -norm based fidelity term. These methods are all full variational frameworks with L^1 -norm based fidelity term.

An important thing to note is that unlike the noise variance constrained case of additive Gaussian noise, there is no good way to automatically compute the fidelity weight for L^1 -norm based fidelity term. Thus the fidelity weight λ can only be selected experimentally.

Although regularization with L^1 -norm based fidelity term is suitable for removing impulse noise whose noise level is not high, it cannot handle the mixed impulse and Gaussian noise well. Therefore, we are interested to develop a better regularization framework which can handle both higher-level of impulse noise as well as mixtures of impulse and Gaussian noises. We will discuss them in the following sections in details.

4.2. Two-phase extension of the proposed framework for color impulse noise removal

As discussed in section 4.1, though regularization frameworks coupled with L^1 -norm based fidelity term have desired geometry property to remove small-scale impulse noise, they still have some limitations. For instance, when removing salt-and-pepper noise, these methods did not fully utilize the property of salt-and-pepper and applied regularization indiscriminately to all pixels in the image, even noise free ones. Thus their performances are degraded especially when salt-and-pepper noise level is high. To overcome this issue, some authors proposed to utilize the property of salt-and-pepper noise and first use a median filter based impulse noise detector to detect them; then applied denoising only to those detected impulse noise candidates while kept others intact. For instance, in [19] the authors proposed to use an *Adaptive Median Filter*

4.2 Two-phase extension of the proposed framework for color impulse noise removal (*AMF*) [46] to detect salt-and-pepper noise for grayscale images first, then apply regularization only on those detected noise candidates and achieve quite good results especially when noise level is high. Similarly to the case of salt-and-pepper noise, for random-valued impulse noise, noise detectors such as the *Adaptive Centre-Weighted Median Filter (ACWMF)* [53] was used in [15, 18], and the *Ranked Ordered Absolute Differences (ROAD)* was used in [34] to help establish a two-phase regularization to better remove them.

In general, those *impulse detection* based two-phase methods gave better results than the full variational frameworks [4, 67] especially when noise level is high. So in this section, we will first show that those impulse noise detectors can be successfully extended to color images. Then we will propose a modified version of our regularization framework inspired mainly by the *image inpainting* application to reconstruct those detected impulse noise corrupted pixels.

4.2.1. Color impulse noise detection

In this sub-section, we will extend those impulse noise detectors for grayscale images to color versions.

4.2.1.1. Color salt-and-pepper noise detection by color AMF

In [19], the authors proposed to use the Adaptive Median Filter (AMF) [46] to detect impulse noise, especially salt-and-pepper noise in grayscale images and achieved very satisfactory detection results. In this thesis, we propose to extend the AMF to a vector version to detect impulse noise in color images.

The detailed steps of the Color AMF impulse noises detection algorithm are:

4.2 Two-phase extension of the proposed framework for color impulse noise removal

First initialize $w = 3$ and $w_{\max} = 13$, then we define an impulse noise indicator function $\text{Imp}(x, y, i)$ for each pixel location (x, y) at each color channel I_i and assign the default value 0 to all locations.

For each pixel location (x, y) at color channel I_i , let N_{xyi} be a $w \times w$ window centered at (x, y) in channel I_i , do

1. Compute $p_{xyi}^{\min}$, p_{xyi}^{med} and $p_{xyi}^{\max}$, which are the minimum, median and maximum of the pixels values of N_{xyi} respectively.
2. If $p_{xyi}^{\min} < p_{xyi}^{\text{med}} < p_{xyi}^{\max}$ go to step 4, else $w = w + 2$.
3. If $w \leq w_{\max}$, go to step 1, else assign $\text{Imp}(x, y, i) = 1$.
4. If $p_{xyi}^{\min} < p_{xyi} < p_{xyi}^{\max}$, then (x, y) should not be a candidate impulse noise location for color channel I_i , else assign $\text{Imp}(x, y, i) = 1$.

After the Color AMF impulse noise detection algorithm is performed, those locations with $\text{Imp}(x, y, i) = 1$ are considered as corrupted by color impulse noise. It is important to notice that during the process of “impulse detection”, we only record impulse noise candidates’ positions and the original image is left unmodified.

Note that as suggested in [19], the authors mentioned that the maximum mask size $w_{\max} = 13$ is able to detect 70% salt-and-pepper noise, which is enough for most conditions. However, if $w_{\max}$ is increased to 39, it will be able to detect 90% salt-and-pepper noise.

4.2.1.2. Color ROAD-based random-valued impulse noise detection

For the random-valued impulse noise, since its value can be any arbitrary value within image’s dynamic range, it is more difficult to detect than salt-and-pepper noise and the Color AMF is not suitable here. So we extend the *Ranked-Ordered Absolute Difference*

4.2 Two-phase extension of the proposed framework for color impulse noise removal
(ROAD) Statics proposed by Garnett et al. in [34] for grayscale images to do color random-valued impulse noise detection.

ROAD is a simple but effective statistics, and can be easily extended to color images. Let us consider a 3×3 neighborhood $\Omega_{\mathbf{x}}$ centered at $\mathbf{x} = (x_1, x_2)$ in image channel i , for each point $\mathbf{y} \in \Omega_{\mathbf{x}}$ and ($\mathbf{y} \neq \mathbf{x}$), define $d_{i,\mathbf{xy}}$ as the absolute difference in intensity between pixels $\mathbf{x}$ and $\mathbf{y}$:

$$d_{i,\mathbf{xy}} = \|I_{i\mathbf{x}} - I_{i\mathbf{y}}\|_{L^1} \quad (4.3)$$

Then sort the $d_{i,\mathbf{xy}}$ values in increasing order and define:

$$\text{ROAD}_i^m(\mathbf{x}) = \sum_{k=1}^m r_{i,k}(\mathbf{x}) \quad (4.4)$$

where $2 \leq m \leq 7$ and $r_{i,k}(\mathbf{x})$ is the k th smallest absolute difference $d_{i,\mathbf{xy}}$ for $\mathbf{y} \in \Omega_{\mathbf{x}}$ and ($\mathbf{y} \neq \mathbf{x}$) in color channel i . The authors in [34] used $m=4$ for grayscale images, for our case we find that ROAD_i^4 is also good to detect impulse noise in color images in 3×3 neighborhood.

Similarly to the Color AMF impulse detection, for each pixel location (x, y) at each color channel I_i , we compute its $\text{ROAD}_i^4(x, y)$, if it is larger than a predefined threshold value we will consider it as a random-valued impulse noise candidate and assign $\text{Imp}(x, y, i) = 1$, else we still keep $\text{Imp}(x, y, i) = 0$. Experimentally, we find that for moderate level of random-valued color impulse noise ($r \leq 25\%$), the $\text{ROAD}_i^4(x, y)$ in 3×3 neighborhood with threshold $T=80$ is enough. For even higher level of random-valued noise, the authors suggested in [34] to use ROAD^m with $m=12$ in 5×5 neighborhood to achieve more robust results.

4.2 Two-phase extension of the proposed framework for color impulse noise removal

The ROAD statistic provided a measure of how close a pixel value is to its four most similar neighbors. The assumption underlying the statistic is that unwanted impulses will vary greatly in intensity from most or all of their neighboring pixels, whereas most pixels composing the actual image should have at least half of their neighboring pixels of similar intensity, even for pixels on an edge. Note that other impulse detection schemes can also be used, for instance, the *Adaptive Centre-Weighted Median Filter* [53] was used in [18] and [15] for detecting random-valued impulse noise in grayscale image. In this thesis, we prefer the relatively simple yet efficient ROAD statistics to detect random-valued color impulse noise.

4.2.2. Reconstruct detected impulse noise corrupted pixels

Having detected possible impulse noise corrupted pixels, the next step is how to reconstruct them. From the characteristics of impulse noise, we know that if a pixel is corrupted by impulse noise (salt-and-pepper or random-valued), basically the original image signal is completely lost; unlike additive Gaussian noise which just adds some perturbations to the signal. It seems like some empty “holes” have been created in the image where those “holes” do not contain any meaningful information. The task of reconstructing pixel values in these “holes” is very similar to the application of *image inpainting* [8, 20-21, 88] which recovers pre-masked image pixel values by interpolation. Inspired by this idea, we propose to use image inpainting principle to reconstruct our impulse noise corrupted pixel values.

Image inpainting is a very useful application which can be used to remove unwanted objects, or reconstruct obstructed objects in images etc. The basic idea of image inpainting is that pixel information lost in the image like the undesired holes can be estimated by interpolating the data located at the neighborhood of the holes. PDE-based regularization algorithms like what we discussed in previous chapters, including our

4.2 Two-phase extension of the proposed framework for color impulse noise removal

proposed framework, can be used to interpolate the data in a way that image structures are coherently completed. Image inpainting is a difficult inverse problem which itself is a research topic and is beyond the scope of this thesis. In this thesis, we are mainly inspired by the idea of image inpainting and will use it to reconstruct image pixels corrupted by impulse noise.

A color image inpainting algorithm was suggested in [85, 88] by Tschumperle using the trace-based formulation

$$\forall (x, y) \in \Omega, \forall i = 1, 2, 3, \quad \begin{cases} \frac{\partial I_i}{\partial t} = \frac{1}{\sqrt{1 + \lambda_+ + \lambda_-}} I_{i\theta, \theta_-} & \text{if Mask}(x, y) = 1 \\ \frac{\partial I_i}{\partial t} = 0 & \text{if Mask}(x, y) = 0 \end{cases} \quad (4.5)$$

where $\text{Mask}(x, y)$ is a pre-defined binary mask indicating the regions in the images where data needs to be interpolated. The author does not allow isotropic smoothing here by restricting the diffusion to the single direction θ_- only to avoid the risk of structure blurring.

For the case of image inpainting, normally the data of all image channels at a particular location (x, y) where $\text{Mask}(x, y) = 1$ are considered completely missing; however, for the case of color images corrupted by impulse noise, the situation is slightly different that possibly only the pixel value of a single color channel is corrupted while pixel values of other image channels are not affected. So by doing a simple modification, we can extend the image inpainting algorithm (4.5) to impulse noise removal for color images

$$\forall (x, y) \in \Omega, \forall i = 1, 2, 3, \quad \begin{cases} \frac{\partial I_i}{\partial t} = \frac{1}{\sqrt{1 + \lambda_+ + \lambda_-}} I_{i\theta, \theta_-} & \text{if Imp}(x, y, i) = 1 \\ \frac{\partial I_i}{\partial t} = 0 & \text{if Imp}(x, y, i) = 0 \end{cases} \quad (4.6)$$

The proposed two-phase regularization framework with the L^1 -norm based fidelity term is defined as below:

$$\forall (x, y) \in \Omega, \forall i = 1, 2, 3, \quad \begin{cases} \frac{\partial I_i}{\partial t} = \operatorname{div}(\mathbf{D} \nabla I_i) & \text{if } \operatorname{Imp}(x, y, i) = 1 \\ \frac{\partial I_i}{\partial t} = 0 & \text{if } \operatorname{Imp}(x, y, i) = 0 \end{cases} \quad (4.7)$$

4.3. Two-phase regularization framework for mixed impulse and Gaussian noises removal

As we showed in previous sections, L^2 -norm based fidelity term is very good for Gaussian noise but not suitable for impulse noise; while L^1 -norm based fidelity term is very suitable for impulse noises but not very good for Gaussian noise. So for the case of mixed impulse and Gaussian noises, the most straight-forward idea is to adaptively assign these two kinds of fidelity terms to pixels corrupted with these two kinds of noises respectively. Cai et al. has proposed in [15] by using the “*impulse detection*” phase to differentiate pixels corrupted by impulse noise from pixels corrupted by Gaussian noises, and assign different types of fidelity terms to them.

However, we think that such methods may not be the best way to remove mixtures of these noises. First of all, most impulse noise, (no matter salt-and-pepper or random-valued noises) are corrupting the image signal completely. Unlike Gaussian noise which just adds some perturbations to the signal, they do not contain any useful image information. During the local regularization process, they will only affect pixels close to them even if they are assigned relatively lower weight in regularization terms as suggested in [67] by Nikolova. So we suggest to minimize this kind of negative effects,

4.3 Two-phase regularization framework for mixed impulse and Gaussian noises removal

the best way should be stopping local diffusions around impulse noise since they contain completely wrong data. We should first reconstruct those wrong data by interpolation using information in their neighborhood. After this process is done, we can then apply regularization to the whole image since now completely wrong data are removed, all the data left are more or less correlated with the original image data and contain some useful information.

A modified regularization term together with an L^1 fidelity term was proposed in [19] to recover the grayscale pixels detected as impulse noise candidates:

$$\sum_{(x,y) \in N} \|I_{xy} - I_{0xy}\|_{L^1} + \beta \left(2 \sum_{(i,j) \in V_{xy} \cap U} \varphi(\|I_{xy} - I_{0ij}\|_{L^1}) + \sum_{(i,j) \in V_{xy} \cap N} \varphi(\|I_{xy} - I_{0ij}\|_{L^1}) \right) \quad (4.8)$$

where N is the sets of detected impulse noise, and U is the sets of *clean* pixels which is the compliment of N , and V_{xy} is the 4 or 8 neighborhood of (x, y) . We can see from this formulation that the weight used to fit noisy pixels I_{xy} of N to neighboring noise-free pixels $I_{0ij} \in U$ is three times of the weight used to fit noisy pixels which are both belonging to the noisy sets N .

So instead of simultaneously removing impulse and Gaussian noises by adaptively assigning different fidelity terms to them, we propose to use a two-phase framework. First, detect impulse noise candidates and replace them by image inpainting like algorithms until they converge. Secondly, for the converged image which contains mainly Gaussian-like noise, apply the adaptive edge-preserving regularization framework with L^2 -norm based fidelity term as we suggested in Chapter 3 to get the overall best results. Note that for those “*impulse removed*” images, we can again estimate their noise variances and use the estimated noise constraints to automatically calculate corresponding fidelity weights. By this two-phase regularization process, we

4.3 Two-phase regularization framework for mixed impulse and Gaussian noises removal can minimize the negative effects introduced by impulse noise during simultaneously smoothing; furthermore, we can still fully take the advantages of established frameworks for image regularization under Gaussian-like noises.

Based on the discussions above, we proposed a two-phase adaptive edge-preservation regularization framework which can handle color images corrupted with both impulse and Gaussian noises.

I. Color impulse noise detection and interpolation

- Use color impulse detection based on color AMF (for salt-and-pepper noise) or color ROAD (for random-valued noise or unknown impulse noise) to decide the candidate impulse noise set where $\text{Imp}(x, y, i) = 1$.
- For salt-and-pepper noise, apply our proposed non-adaptive divergence-based regularization without fidelity term to do impulse noise interpolation for impulse noise candidates where $\text{Imp}(x, y, i) = 1$:

$$\forall (x, y) \in \Omega, \forall i = 1, 2, 3, \quad \begin{cases} \frac{\partial I_i}{\partial t} = \text{Imp}(x, y, i) \text{div}(\mathbf{D} \nabla I_i) \\ \mathbf{D} = \frac{1}{(1 + \lambda_+ + \lambda_-)} \theta_+ \theta_+^T + \frac{1}{\sqrt{(1 + \lambda_+ + \lambda_-)}} \theta_- \theta_-^T \end{cases} \quad (4.9)$$

- For random-valued impulse noise or unknown impulse noise, apply our proposed non-adaptive divergence-based regularization with a L^1 -norm based fidelity term to do impulse noise interpolation for impulse noise candidates where $\text{Imp}(x, y, i) = 1$:

$$\forall (x, y) \in \Omega, \forall i = 1, 2, 3, \quad \begin{cases} \frac{\partial I_i}{\partial t} = \text{Imp}(x, y, i) \left(\text{div}(\mathbf{D} \nabla I_i) + \lambda \frac{I_i - I_{0i}}{|I_i - I_{0i}|} \right) \\ \mathbf{D} = \frac{1}{(1 + \lambda_+ + \lambda_-)} \theta_+ \theta_+^T + \frac{1}{\sqrt{(1 + \lambda_+ + \lambda_-)}} \theta_- \theta_-^T \end{cases} \quad (4.10)$$

4.4 Experimental results

- After (4.9) is converged, we get the “*impulse removed*” image $\tilde{\mathbf{I}}$
- II. Adaptive edge-preserving regularization for the “*impulse removed*” image $\tilde{\mathbf{I}}$:
 - Estimate noise variance of image $\tilde{\mathbf{I}}$ using equation (3.23)
 - Apply our proposed adaptive edge-preserving regularization framework with adaptive L^2 -norm based fidelity term for image $\tilde{\mathbf{I}}$:

$$\begin{cases} \frac{\partial \tilde{I}_i}{\partial t} = \text{div}(\mathbf{D} \nabla \tilde{I}_i) + \lambda_i (\tilde{I}_i - \tilde{I}_{0i}) \\ \mathbf{D} = \frac{1}{(1 + \lambda_+ + \lambda_-)^{\frac{1+V(\lambda_+ + \lambda_-)}{2}}} \theta_+ \theta_+^T + \frac{1}{\sqrt{(1 + \lambda_+ + \lambda_-)}} \theta_- \theta_-^T \end{cases} \quad (4.11)$$

Note that we applied slightly different formulation (4.10) for random-valued impulse noise reconstruction than equation (4.9) for salt-and-pepper noise. This is mainly because that random-valued impulse noise is more difficult to precisely detect than salt-and-pepper noise, and we will inevitably have some misses or false hits especially when noise level is high. Due to these false hits, we added a L^1 -norm based fidelity term because it can help maintain some wrongly detected pixels to their original “*correct*” values. Actually for the case of salt-and-pepper noise, we can also add a L^1 -norm based fidelity term and still get satisfactory results, but normally it is not necessary because the detection accuracy for salt-and-pepper noise is generally very high. Generally, for unknown type of impulse noise it is good to keep the L^1 -norm based fidelity term to recover some wrongly detected pixels.

4.4. Experimental results

In this section, we present regularization results for color images corrupted with different types of impulse noise and mixed noises. Among all the color images we tested, here we show the results of the famous 256-by-256 24-bit color *Lena* image,

4.4 Experimental results

available on line from University of Granada [25]. Impulse noise models are shown in (4.1) and (4.2). For the case of mixed noise, Gaussian noise was added first, and then impulse noise was added.

To quantitatively assess the regularization performance, we still use Peak-Signal-Noise-Ratio (PSNR) defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2 n |\Omega|}{\sum_{i=1}^n \sum_{x,y \in \Omega} (I_i(x,y) - \tilde{I}_i(x,y))^2}, \quad (4.12)$$

where $\mathbf{I}$ and $\tilde{\mathbf{I}}$ denote the regularized and the original clean image, respectively, $|\Omega|$ is the area of the spatial image domain and n is the number of channels.

To compare the impulse noise removing ability of the proposed two-phase “impulse detection” based framework with the traditional one-phase regularization frameworks based on L^1 -norm based fidelity terms, we chose the most typical Vector TV framework [81] with an L^1 -norm based fidelity term for comparison.

$$E_{\text{TV},L^1}(\mathbf{I}) = \int_{\Omega} \left(\sqrt{|\nabla \mathbf{I}|^2} + \lambda \|\mathbf{I} - \mathbf{I}_0\|_{L^1} \right) dx dy \quad (4.13)$$

Note that the non-differentiability of the terms inside the above energy needs some sort of regularization. So in our numerical experiments, we use a regularized version instead:

$$E_{\text{TV},L^1}(\mathbf{I}) = \int_{\Omega} \left(\sqrt{|\nabla \mathbf{I}|^2 + \varepsilon} + \lambda \sqrt{|\mathbf{I} - \mathbf{I}_0|^2 + \delta} \right) dx dy, \quad (4.14)$$

where ε and δ are regularization parameter and we assign both 1×10^{-4} throughout our experiments. For the fidelity weight λ , as we discussed before that there is no good

4.4 Experimental results

way to automatically calculate it, we chose it experimentally to give the best performance to the Vector TV regularization.

First, we consider the case of salt-and-pepper noise. In Figure 4-1 and Figure 4-2, we show the regularization results for the Lena image corrupted by 20% and 50% salt-and-pepper noise respectively. Since in our proposed framework we selected color AMF as our impulse detector, we also included the result of directly replacing the color AMF detected pixels with their median to show the improvement of our interpolation formulation (4.9). The maximum neighborhood size we used in color AMF was 13 throughout the test. Obviously, the proposed algorithm generated much better results than other two methods both quantitatively and visually. We can observe that L^1 -norm based fidelity term did help removed most impulse noise; however, due to indiscriminately applying regularization on even noise-free pixels, the overall regularization performance is affected. Furthermore, because those impulse corrupted pixels contain completely wrong data, they will have negative contribution to diffusions for noise-free pixels around them as we mentioned in previous sections, especially when salt-and-pepper noise level is higher (50%). Experimental results showed the advantages of applying an impulse detection first followed by selective regularization. Finally, our proposed framework also exhibited good interpolation results, much better than median filter based methods.

Secondly, we will see the case of random-valued impulse noise. In Figure 4-3 and Figure 4-4, we showed the regularization results for the *Lena* image corrupted by 20% and 40% random-valued impulse noise respectively. Random-valued impulse noise is more difficult to detect than salt-and-pepper noise; we will inevitably have some misses or false hits. Due to these false hits, we added a L^1 -norm based fidelity term in our regularization framework because it can help maintain some false hits to their original

4.4 Experimental results

“correct” value. We also used a color ROAD median filter for comparisons. We can see that for normal level of random-valued impulse noise, our method performed quite well. However, when noise level is increased to 40%, though quantitatively our method still produced PSNR about 2dB higher than other methods, visually we can see a few noisy patches. This is mainly due to the accuracy of the impulse detection by color ROAD was reduced when noise level is high, though we already used a ROAD¹² with a 5×5 neighborhood ($T=250$). This can be improved if we apply ROAD iteratively for a few rounds similar as suggested by [18]; however, we did not apply iterations in our test because in reality the chance of getting such high level of random-valued impulse noise is low.

Finally, we showed the case of mixed Gaussian and impulse noise, which is normally quite difficult to handle for most regularization frameworks. In Figure 4-5, we showed the results of the *Lena* image corrupted by additive Gaussian noise ($\sigma = 20$) and salt-and-pepper noise ($s=20\%$). We can see from the results in (b) after the first impulse removal phase that most salt-and-pepper noise were successfully removed and those pixel values were interpolated with values similar to Gaussian corrupted ones. Although the remaining noise was not strictly Gaussian distributed, we can still apply our proposed framework in Chapter 3 on it. The final result was very good with PSNR=28.36dB and a much better edge-preservation performance than the full variational Vector TV method with L^1 -norm based fidelity term.

In Figure 4-6, we showed a more difficult case of the *Lena* image corrupted by additive Gaussian noise ($\sigma = 20$) and random-valued impulse noise ($r=20\%$). Again, after the first phase, most obvious impulse noise was removed, though the remaining noise was more different from Gaussian than the case of salt-and-pepper noise. However, after the second phase, we still got a good PSNR (27.27dB) and a visually good edge-preserving

4.4 Experimental results

result than the full variational method. Overall, these two mixed noise results proved that our proposed two-phase regularization framework is capable of handling the case of mixed Gaussian and impulse noises.

The computation time of those methods are listed in Table 4-1 and Table 4-2. The simulation is run by the same HP nc6400 laptop as mentioned in Chapter 3. We can see that compared with the non-iterative filtering method such as Color AMF and Color ROAD which will typically take less than 1 second for a 256×256 color image, obviously the proposed iterative based framework is much slower; however the PSNR gain is quite substantial especially when impulse noise level is higher than normal non-iterative method can handle. So one has to select the PSNR gain and computation loss depends on their practical application needs. Compared with similar iterative diffusion based method Vector TV with L^1 -norm, the proposed method is much faster, thanks to the use of impulse detection scheme, especially when impulse noise level is high.

Noise Level	Color AMF	Color ROAD	Vector TV-L1	Proposed
$s = 20\%$	0.43s	N.A.	47.92s	8.62s
$s = 50\%$	0.53s	N.A.	138.46s	22.78s
$r = 20\%$	N.A.	0.66s	42.37s	30.25s
$r = 40\%$	N.A.	0.71s	119.82s	63.73s

Table 4-1: Comparisons of CPU time in seconds for different level of salt-and-pepper and random impulse noise

In Table 4-2, we showed the computation time to process images with mixed Gaussian and impulse noise. Since no non-iterative method can handle mixed noise, we can only compare with Vector TV with L^1 -norm. One can see that the proposed method can run faster than the Vector TV with better PSNR, also because of the introduction of impulse detection scheme. We also list down the detail computation time for the two different

4.4 Experimental results

phases of our framework: the impulse noise detection and removal phase (Impulse Removal), and the adaptive edge-preserving regularization phase (Gaussian Removal). From Table 4-2 we can see clearly how our framework works, and typically for impulse noise removal, it takes longer than removing Gaussian noise. Similar to what we have discussed and compared in Chapter 3, the computation time is proportional to image size, so for a mega pixel image, one can expect 16 times the time listed in the table based for different noise level.

Mixed Noise	Vector TV-L1	Proposed	Impulse Removal	Gaussian Removal
$\sigma = 20, s = 20\%$	87.48s	17.57s	12.65s	4.92s
$\sigma = 20, r = 20\%$	54.48s	38.51s	33.5s	5.21s

Table 4-2: Comparisons of CPU time in seconds for different mixed noise

4.4 Experimental results

(a) Noisy image ($s=20\%$)

(b) Color AMF

(c) Vector TV + L1

(d) Our proposed method

Figure 4-1: Regularization results for the 256x256 *Lena* image corrupted by salt-and-pepper noise. (a) *Lena* image corrupted by salt-and-pepper noise $s=20\%$ (PSNR=12.27dB); (b) Color AMF (PSNR=30.97dB); (c) Vector TV + L1 fidelity term (PSNR=27.32dB); (d) Our proposed method (PSNR=33.83dB).

4.4 Experimental results

(a) Noisy image ($s=50\%$)

(b) Color AMF

(c) Vector TV + L1

(d) Our proposed method

Figure 4-2: Regularization results for the 256x256 *Lena* image corrupted by salt-and-pepper noise. (a) *Lena* mage corrupted by salt-and-pepper noise $s=50\%$ (PSNR=8.26dB); (b) Color AMF (PSNR=24.31dB); (c) Vector TV + L1 fidelity term (PSNR=24.47dB); (d) Our proposed method (PSNR=31.21dB).

4.4 Experimental results

(a) Noisy image ($r=20\%$)

(b) Color ROAD

(c) Vector TV + L1

(d) Our proposed method

Figure 4-3: Regularization results for the 256x256 *Lena* image corrupted by random-valued impulse noise. (a) *Lena* image corrupted by random-valued impulse noise $r=20\%$ (PSNR=15.60dB); (b) Color ROAD median filter (PSNR=28.58dB); (c) Vector TV + L1 fidelity term (PSNR=27.21dB); (d) Our proposed method (PSNR=30.44dB).

4.4 Experimental results

(a) Noisy image ($r=40\%$)

(b) Color ROAD

(c) Vector TV + L1

(d) Our proposed method

Figure 4-4: Regularization results for the 256x256 *Lena* image corrupted by random-valued impulse noise. (a) *Lena* image corrupted by random-valued impulse noise $r=40\%$ (PSNR=12.63dB); (b) Color ROAD median filter (PSNR=24.95dB); (c) Vector TV + L1 fidelity term (PSNR=24.47dB); (d) Our proposed method (PSNR=27.04dB).

4.4 Experimental results

(a) Mixed noisy image ($\sigma=20$, $s=20\%$)

(b) Our proposed method – Phase 1

(c) Our proposed method – Phase 2

(d) Vector TV + L1

Figure 4-5: Regularization results for the 256x256 *Lena* image corrupted by mixed Gaussian and salt-and-pepper noise. (a) *Lena* image corrupted by both additive Gaussian noise $\sigma=20$ and salt-and-pepper noise $s=20\%$ (PSNR=11.93dB); (b) “Impulse removed” image after Phase-1 of the proposed method (PSNR=23.16); (c) Final result of our proposed method (PSNR=28.36dB); (d) Vector TV + L1 fidelity term (PSNR=25.52dB).

4.4 Experimental results

(a) Mixed noisy image ($\sigma=20$, $r=20\%$)

(b) Our proposed method– Phase 1

(c) Our proposed method – Phase 2

(d) Vector TV + L1

Figure 4-6: Regularization results for the 256x256 *Lena* image corrupted by mixed Gaussian and random-valued impulse noise. (a) *Lena* image corrupted by both additive Gaussian noise $\sigma=20$ and random-valued impulse noise $r=20\%$ (PSNR=14.19dB); (b) “Impulse removed” image after Phase-1 of our proposed method (PSNR=23.63); (c) Final result of our proposed method (PSNR=27.37dB); (d) Vector TV + L1 fidelity term (PSNR=25.20dB).

4.5. Conclusion

In this Chapter, we have discussed the problem of removing impulse noise either by L^1 -norm based fidelity term or by an “impulse detection” phase. We then extended two grayscale impulse noise detection schemes to color images, and proposed to use our regularization framework to reconstruct detected impulse noise corrupted pixels. Finally, a two-phase regularization framework was proposed to remove mixed Gaussian and impulse noise. Using the two-phase framework, undesired interactions between pixel values corrupted by impulse noise and Gaussian noise can be minimized; the negative contributions from those completely non-informative impulse noise corrupted pixels are also reduced. Experimental results also showed that after the two-phase extension, the proposed framework is capable of handling both impulse noise and mixed of Gaussian and impulse noises.

Chapter 5. Applications and Possible Extensions of the Proposed Regularization Framework

In previous chapters, we have reviewed most of the classical PDE based regularization methods and presented our proposed edge-preserving regularization framework. One common characteristics shared by these methods, including our proposed method, is that most of their regularization terms are based on local derivative-based image operators. We have shown that many of these local operator based regularization methods can remove image noise and preserve edges well in most images. However, due to the limitation of local image operators, sometimes it is difficult for these methods to handle complicated images with lots of fine structures like textures. Furthermore, local derivative-based image operators are more sensitive to image noise especially when the noise level is high. Therefore, it would be helpful to include some *higher-level* information when dealing with those complicated tasks.

In this chapter, we want to explore the possibilities of extending our proposed regularization framework to include more *global* information during the regularization process. We will first propose a more robust edge-indicator function after extending the Zernike moments [98] based edge detector [37] to color images. Then we will discuss the possible future research direction of extending our proposed regularization framework using non-local operators proposed by Buades et al.[12-13].

We would also like to point out that some possible applications such as color edge detection, color image de-blurring, and color image inpainting etc., can be achieved as applications of our proposed regularization framework.

5.1. Zernike moments-based color image regularization

Moments-based techniques have been widely used in the fields of image processing, computer vision and pattern recognition with applications such as edge detection [37, 76, 100], texture segmentation [9], image compression [73] and object matching [47-48], etc. One of the most important advantages of using moment-based image processing technique is that it is less sensitive to image noise, compared with those derivative-based operators.

There are different kinds of moments in the literature, for instance, the most common and the simplest moment is the geometric moments [61], which can be used to represent image features but in many cases it is not very efficient. Ghosal and Mehorthra [36-37] proposed to use Zernike moments [98] to detect a set of image features with sub-pixel accuracy and less sensitive to image noise. In this section, we will extend the Zernike moments based edge detector to color images and show it can be used as our edge indicator function.

5.1.1. Property of Zernike moments

Zernike moments of order n and repetition m for a grayscale image $I(x, y)$ is defined in [98] as

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} I(x, y) V_{nm}^*(\rho, \theta) dx dy \quad (5.1)$$

where V_{nm}^* denotes the complex conjugate of V_{nm} , which is defined as

$$\begin{cases} V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{im\theta} \\ R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)! \rho^{n-2s}}{s! ((n+|m|)/2 - s)! ((n-|m|)/2 - s)!} \end{cases} \quad (5.2)$$

5.1 Zernike moments-based color image regularization

It is easy to derive a few lower order moments from (5.2) as in [37]:

$$\begin{cases} V_{11}(\rho, \theta) = \rho e^{j\theta} \\ V_{20}(\rho, \theta) = 2\rho^2 - 1 \end{cases} \quad (5.3)$$

In Cartesian coordinates, (5.3) can be rewritten as

$$\begin{cases} V_{11}(x, y) = x + jy \\ V_{20}(x, y) = 2x^2 + 2y^2 - 1 \end{cases} \quad (5.4)$$

One of the most important properties of Zernike moments is that they are rotational invariant. Consider the case when image $I(x, y)$ is rotated by an angle ϕ , the Zernike moments A_{nm} of the original image $I(x, y)$ and the Zernike moments A'_{nm} of the rotated image $I'(x, y)$ has the following relationship:

$$A'_{nm} = A_{nm} e^{-jm\phi} \quad (5.5)$$

5.1.2. Zernike moments-based color edge detection

In [37], the authors proposed a 2D step edge model as shown in Figure 5-1 defined on a unit circle, and calculate the four step edge parameters using Zernike moments for each edge point. As shown in Figure 5-1, k is the step edge strength, l is the perpendicular distance from the center of the unit circle to this edge also defines the angle ϕ with respect to the x -axis, and h is the background grayscale value.

5.1 Zernike moments-based color image regularization

Figure 5-1: 2D step edge model with sub-pixel accuracy

Because of its rotational invariant property, when the step edge is rotated by an angle $-\phi$, it will be parallel to the x -axis so we will have the imaginary component of A'_{11} equal to 0 as below:

$$\text{Im}[A'_{11}] = \sin(\phi)\text{Re}[A_{11}] - \cos(\phi)\text{Im}[A_{11}] = 0, \quad (5.6)$$

so we have

$$\phi = \tan^{-1}\left(\frac{\text{Im}[A_{11}]}{\text{Re}[A_{11}]}\right). \quad (5.7)$$

From (5.5) and (5.7), we can derive A'_{11} as

$$A'_{11} = \text{Re}[A'_{11}] = \text{Re}[A_{11}]\cos(\phi) + \text{Im}[A_{11}]\sin(\phi) = \sqrt{\text{Re}[A_{11}]^2 + \text{Im}[A_{11}]^2}. \quad (5.8)$$

In [37, 76], the step edge strength k with sub-pixel accuracy is given as

$$k = \frac{3A'_{11}}{2(1-l^2)^{1.5}}, \quad (5.9)$$

5.1 Zernike moments-based color image regularization

where the sub-pixel edge distance l can help get thinner edges with sub-pixel edge locations; however, for our regularization framework, we are mainly interested in each pixel's edge response, not to the sub-pixel level yet. And in real simulation l is typically very small, so we propose to use A'_{11} to approximate the step edge response k .

From (5.8) we can see that A'_{11} works similarly to the grayscale gradient norm

$|\nabla I| = \sqrt{I_x^2 + I_y^2}$ and can be used to measure edge responses of grayscale images. In our regularization framework, we are more interested to measure color edge responses.

Inspired by this similarity, we extended the Zernike moment A'_{11} for grayscale images to color images using the same formulation as our color gradient norm definition as

$$\mathbf{A}'_{11} = \sqrt{\sum_{i=1}^n \left(\text{Re}[A'_{i11}]^2 + \text{Im}[A'_{i11}]^2 \right)}, \quad (5.10)$$

where n is the number of color channels.

In the next sub-section, we will show that our extended Zernike moments-based color gradient norm can respond to color edges well and also can be used as a more robust edge indicator function for our regularization framework.

5.1.3. Zernike moments-based color edge indicator function and the corresponding experimental results

In our experiment, Zernike moments are computed within a circular window around each pixel. As the window size increases, more global information is included. To include more global information than the local image operators, we use a 7×7 window. Other parameters are kept the same.

5.1 Zernike moments-based color image regularization

(a) Original image

(b) Noisy image ($\sigma=80$)

(c) Local color gradient of (a)

(d) Zernike moments of (a)

(e) Local color gradient of (b)

(f) Zernike moments of (b)

Figure 5-2: Comparisons of color edge responses of local color gradient norm and Zernike moment-based color gradient norm: (a) Original 256x256 *House* image; (b) *House* image corrupted by additive Gaussian noise $\sigma=80$; (c) Local color gradient norm of (a); (d) Zernike moment-based color gradient norm of (a); (e) Local color gradient norm of (b); (f) Zernike moment-based color gradient norm of (b).

5.1 Zernike moments-based color image regularization

In Figure 5-2, we presented the color gradient norm responses and the Zernike moment-based color gradient norm responses for both noise-free and a highly noisy *House* image. From Figure 5-2 (b) and (c), we can see that for the noise-free image, Zernike moment-based color gradient norm $\mathbf{A}'_{11}$ did produce similar color edge responses as the local color gradient norm $|\nabla \mathbf{I}|$, but with slightly thicker edges and less details. From Figure 5-2 (d) and (e), for the highly noisy image corrupted by additive white Gaussian noise ($\sigma=80$), one can clearly see that Zernike moment-based color gradient norm is less sensitive to noise compared with the local color gradient norm $|\nabla \mathbf{I}|$. Therefore, when noise level is very high, we may get some advantages if we use the more robust Zernike moment-based $\mathbf{A}'_{11}$ for our edge indicator function.

With the proposed Zernike moment-based color gradient norm $\mathbf{A}'_{11}$, we can construct a new edge indicator function similarly as the original formulation in (3.8):

$$V(x, y) = \frac{\mathbf{A}'_{11}}{\mathbf{A}'_{11} + \sigma_e} \quad (5.11)$$

In Figure 5-3, the regularization results of our proposed regularization framework using the original local gradient-based edge indicator function (b), and using the Zernike moments-based edge indicator function (c) are presented. One can see that in terms of PSNR, (c) is slightly higher than (b), which shows the improvement brought by the more robust edge indicator function. Visually, (c) is also slightly better than (b), but no obvious difference. We also show the final color edge map after the regularization in (d), which is much improved compared with the initial edge maps shown in Figure 5-2 (e). This also shows that color edge detection can be easily achieved together with our proposed regularization framework.

5.1 Zernike moments-based color image regularization

Note that when noise level is not high, the improvement by using Zernike moment-based edge indicator function is not very obvious. When noise level is low, there is even some slight degradation in the PSNR sense. This is due to the use of larger window so that some fine details cannot be captured well. A better method is to select different edge indicator function based on different conditions.

(a) Noisy image ($\sigma=80$)

(b) Local color gradient based

(c) Zernike moment based

(d) Final edge map

Figure 5-3: Comparisons of regularization results of the 256x256 *House* image corrupted by Gaussian noise using different edge indicator functions. (a) *House* image corrupted by additive Gaussian noise $\sigma=80$ (PSNR=10.07dB); (b) Regularization results of our proposed method using the original local gradient-based edge indicator function (PSNR=26.39); (c) Regularization results of our proposed method using the Zernike moment-based edge indicator function (PSNR=26.72dB); (d) Final edge map after regularization.

5.2. Possible nonlocal extension of our proposed framework

In this section, we will briefly review the background of nonlocal framework for image denoising, and then suggest that extending our proposed regularization framework to the nonlocal version is a practical future research direction.

Recently, nonlocal methods in image processing have been proposed for the purpose of improving texture denoising as most of the traditional denoising methods use only local image information and tend to treat texture as noise, which will result in losing textures. Nonlocal methods [14, 52, 87] and the bilateral filters [84]. The basic idea of neighborhood filters is to restore a pixel by averaging the values of its neighboring pixels with similar grayscale value to it.

Buades et al. [14] generalized this idea by applying patch-based methods proposed for texture synthesis [29] to image denoising, which is the famous nonlocal-means (NL-means) neighborhood filter:

$$\begin{cases} NLI(x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{d_a(I(x), I(y))}{h^2}} I(y) dy \\ d_a(I(x), I(y)) = \int_R G_a(t) |I(x+t) - I(y+t)|^2 dt \end{cases}, \quad (5.12)$$

where d_a is the patch distance, G_a is the Gaussian kernel with standard deviation a which determines the patch size, $C(x)$ is the normalizing factor and h is the filtering parameter which corresponds to noise level. The NL-means not only compares the grayscale value at a single pixel but also the geometrical configuration in a whole neighborhood (patch). Thus, to denoise a pixel, it is better to average the nearby pixels with similar structures rather than just with similar intensities.

5.2 Possible nonlocal extension of our proposed framework

In variational framework, Kindermann et al. [52] formulated the neighborhood filters and NL-means filters as nonlocal regularizing functional which have the general form:

$$\psi(I) = \int_{\Omega \times \Omega} \phi \left(\frac{|I(x) - I(y)|^2}{h^2} \right) w(|x - y|) dx dy \quad (5.13)$$

where $w(|x - y|)$ is a positive weight function. However, these functional are generally not convex. To overcome this problem, Gilboa and Osher [39] proposed a convex functional inspired from graph theory:

$$\psi(I) = \frac{1}{2} \int_{\Omega \times \Omega} \phi(|I(x) - I(y)|) w(x, y) dx dy \quad (5.14)$$

where $\phi(x)$ is convex and positive, and the weight function $w(x, y)$ is nonnegative and symmetric. Furthermore, based on the gradient and divergence definitions on graphs in the context of machine learning, Gilboa and Osher [40] derived the nonlocal operators. Let $I: \Omega \rightarrow \mathbb{R}$ be a function, and $w: \Omega \times \Omega \rightarrow \mathbb{R}$ is a nonnegative and symmetric weight function. The authors defines the nonlocal gradient as

$$(\nabla_w I)(x, y) = (I(x) - I(y)) \sqrt{w(x, y)} \quad (5.15)$$

and the nonlocal gradient norm as

$$|\nabla_w I|(x) = \sqrt{\int_{\Omega} (I(y) - I(x))^2 w(x, y) dy} \quad (5.16)$$

The nonlocal divergence operator $div_w \vec{v}$ of the vector $\vec{v}$ is also defined as the ad joint of the nonlocal gradient

$$(div_w \vec{v})(x) = \int_{\Omega} (v(x, y) - v(y, x)) \sqrt{w(x, y)} dy \quad (5.17)$$

5.3 Possible applications of our proposed regularization framework

Based on these nonlocal operators, the authors introduced the general formulation of the nonlocal regularization functional as

$$\psi(I) = \int_{\Omega} \phi(|\nabla_w I|^2) dx \quad (5.18)$$

With the nonlocal gradient and divergence operators, it is possible that we can extend them to color images and directly apply them to our proposed adaptive edge-preserving regularization framework to extend it to the nonlocal version. The most straightforward way is to use nonlocal gradient and nonlocal divergence operators in our regularization framework and construct structure tensor G and diffusion tensor T based on them, and then our framework can give nonlocal denoising performance.

Another similar way proposed in [87] is to first map the target image to a high-dimensional patch-space given the preferred patch size. Thus in this high-dimensional space, each existing patch is now a single point. Our regularization formulation can be applied directly on the patch space, and project back to the original image domain if necessary.

5.3. Possible applications of our proposed regularization framework

In this section, we will briefly discuss the possible applications of our proposed regularization framework besides image denoising.

As we have mentioned, color image regularization is often used as a pre-processing step for many image processing applications such as color edge detection, object matching etc. Since our proposed regularization framework is an edge-preserving, basically color edge detection can be considered as a by-product of our framework. As shown in section 5.1, the final edge map (color gradient norm responses) in Figure 5-3 (d) is quite

5.3 Possible applications of our proposed regularization framework

good as an edge detection results. Some additional thresholding, thinning and edge linking can be applied accordingly to get better results.

Another application is color image inpainting [8], which has been discussed and used to reconstruct impulse noise corrupted pixels in Chapter 4. The performance of color image inpainting based on our proposed framework can be seen from the impulse noise removing results shown in section 4.4.

Another important application is to regularize image noises and distortions due to image compression. With the advancement of information technology, the overall quantity and size of images have been increasing significantly for a few decades. To store and communicate those images more efficiently, image compression, especially lossy data compression has become very popular. For lossy image compression, such as JPEG compression, a lot of compression artifacts will become the byproduct of the more aggressive compression, such as Contouring, Staircase like noise around image edges, Blocky artifacts and image edges are also often distorted. Image regularization can help regularize those heavily compressed images to make them easier for human perception or future processing. A quick example has presented in Figure 5-4 to show the potential of our proposed image regularization framework for image compression artifacts. One can see that the proposed did remove some blocky artifacts in the image and some distorted edges are also more regular, the text inside image also turns sharper.

Figure 5-4: A quick example showing the potential of the proposed image regularization framework in regularizing a heavily compressed jpeg image. (a) Original image; (b) Regularized image.

5.3 Possible applications of our proposed regularization framework

Besides these above mentioned applications, other applications such as color image deblurring [4, 52], color image segmentation [78], color image magnification [58, 88], etc. can also be achieved with some modification of the proposed regularization framework. In general, there are many color image processing applications can be related to PDE-based regularization frameworks [81, 88], our proposed adaptive edge-preserving framework can achieve good results in most of them because edges are playing important roles in many applications.

Chapter 6. Conclusions and Future Work

6.1. Conclusions

In this thesis, we have studied the problem of edge-preserving color image regularization, which is a low-level process and can be used as a pre-processing stage in many image processing applications. Most of these applications require both good noise removal and edge preservation, which is difficult to be simultaneously achieved by the existing regularization methods. Our objective is to design an adaptive regularization framework, which can adaptively preserve important image features (edges, corners) better, while still being able to effectively remove image noise.

Our main contributions include:

In Chapter 3, we compared edge-preserving property of different regularization terms, and proposed to construct a locally adaptive regularization term based on the local edge information. Besides this, we also proposed to automatically calculate an adaptive data fidelity term based on the edge information as well to help preserve edges. Experimental results showed that our proposed adaptive edge-preserving regularization framework can preserve edges better while still removing noise well.

In Chapter 4, to deal with impulse noise, we further extended our regularization framework by extending a grayscale *impulse noise detection* method to color images and used together with our propose regularization framework. We also considered the more difficult case of mixed impulse and Gaussian noise by proposing an innovative two-phase regularization framework. Impulse noise corrupted pixels were first detected and reconstructed by a modified version of our regularization framework using the *color image inpainting* principle, then the original version of our proposed framework

6.2 Future research directions

was applied to the reconstructed images to finish final denoising. We also presented experimental results to show the denoising performance of our proposed framework under different noise conditions.

In Chapter 5, we proposed to use a semi-local Zernike moments instead of local image derivatives to design a more robust edge indicator function for our regularization framework. Experimental results were also presented to show the improvement in performance especially for highly-noisy images. The possible extension of our proposed regularization framework to the nonlocal version was also discussed and suggested as future research directions.

6.2. Future research directions

In this thesis, our proposed image regularization framework was mainly used for image denoising under different noise conditions. In the future, more related image processing applications such as color edge detection, color image deblurring, and color image inpainting etc., which can be directly integrated into our framework or can use our framework as a preprocessing stage, could be explored in details.

In Section 5.1, we have proposed to use a *semi-local* Zernike moments to construct a more robust color edge indicator function and have reached good experimental results for highly-noisy images. Future research directions could be investigating the performance of constructing diffusion tensors directly based on the vector geometry information given by those *semi-local* moments-based operators.

As discussed in Chapter 5, our proposed adaptive edge-preserving color image regularization framework is mainly based on traditional local derivative-based image operators. Extending our proposed framework to the nonlocal version using nonlocal image operators or directly on the patch-space should be a good future research

6.2 Future research directions

direction. Regularization performance improvements of our proposed framework for complicated textured images and highly-noisy images can be expected. Furthermore, it would be interesting to better understand the differences between local and nonlocal processes, and to design a new framework which can adaptively apply local or nonlocal process depending on image characteristics to maximize regularization performance.

6.2 Future research directions

Bibliography

- [1] L. Alvarez, P. L. Lions, and J. M. Morel, "Image Selective Smoothing and Edge-Detection by Nonlinear Diffusion," *Siam Journal on Numerical Analysis*, vol. 29, pp. 845-866, Jun 1992.
- [2] L. Alvarez and L. Mazorra, "Signal and Image-Restoration Using Shock Filters and Anisotropic Diffusion," *Siam Journal on Numerical Analysis*, vol. 31, pp. 590-605, Apr 1994.
- [3] L. Ambrosio and V. M. Tortorelli, "Approximation of Functionals Depending on Jumps by Elliptic Functionals Via Gamma-Convergence," *Communications on Pure and Applied Mathematics*, vol. 43, pp. 999-1036, Dec 1990.
- [4] L. Bar, A. Brook, N. Sochen, and N. Kiryati, "Deblurring of Color Images Corrupted by Impulsive Noise," *IEEE Transactions on Image Processing*, vol. 16, pp. 1101-1111, 2007.
- [5] L. Bar, N. Kiryati, and N. Sochen, "Image Deblurring in the Presence of Impulsive Noise," *International Journal of Computer Vision*, vol. 70, pp. 279-298, Dec 2006.
- [6] L. Bar, N. Sochen, and N. Kiryati, "Image Deblurring in the Presence of Salt-and-Pepper Noise," in *Proceedings of Scale Space and Pde Methods in Computer Vision*, vol. 3459, R. Kimmel, et al., Eds., ed Berlin: Springer-Verlag Berlin, pp. 107-118.
- [7] D. Barash, "Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 844-847, 2002.
- [8] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, and A. A. Acm, "Image Inpainting," in *Proceedings of Siggraph 2000* ed New York: Assoc Computing Machinery, pp. 417-424.
- [9] J. Bigun and J. M. Hans du Buf, "Geometric Image Primitives by Complex Moments in Gabor Space and the Application to Texture Segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648-650, 1992.
- [10] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust Anisotropic Diffusion," *IEEE Transactions on Image Processing*, vol. 7, pp. 421-432, Mar 1998.
- [11] P. Blomgren and T. F. Chan, "Color Tv: Total Variation Methods for Restoration of Vector-Valued Images," *IEEE Transactions on Image Processing*, vol. 7, pp. 304-309, Mar 1998.
- [12] A. Buades, B. Coll, and J. Morel, "Image Denoising by Non-Local Averaging," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 25-28, 2005.
- [13] A. Buades, B. Coll, and J. M. Morel, "A Non-Local Algorithm for Image Denoising," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 60-65 vol. 2, 2005.
- [14] A. Buades, B. Coll, and J. M. Morel, "A Review of Image Denoising Algorithms, with a New One," *Multiscale Modeling & Simulation*, vol. 4, pp. 490-530, 2005.
- [15] J. F. Cai, R. H. Chan, and M. Nikolova, "Two-Phase Approach for Deblurring Images Corrupted by Impulse Plus Gaussian Noise," *Inverse Problems and Imaging*, vol. 2, pp. 187-204, May 2008.

Bibliography

- [16] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, 1986.
- [17] A. Chambolle and P. L. Lions, "Image Recovery Via Total Variation Minimization and Related Problems," *Numerische Mathematik*, vol. 76, pp. 167-188, Apr 1997.
- [18] R. H. Chan, H. Chen, and M. Nikolova, "An Iterative Procedure for Removing Random-Valued Impulse Noise," *IEEE Signal Processing Letters*, vol. 11, pp. 921-924, 2004.
- [19] R. H. Chan, H. Chung-Wa, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Transactions on Image Processing*, vol. 14, pp. 1479-1485, 2005.
- [20] T. F. Chan, "Nontexture Inpainting by Curvature-Driven Diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, pp. 436-449, Dec 2001.
- [21] T. F. Chan and J. H. Shen, "Mathematical Models for Local Nontexture Inpaintings," *Siam Journal on Applied Mathematics*, vol. 62, pp. 1019-1043, Feb 2002.
- [22] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic Edge-Preserving Regularization in Computed Imaging," *IEEE Transactions on Image Processing*, vol. 6, pp. 298-311, 1997.
- [23] P. Chatterjee and P. Milanfar, "Is Denoising Dead?," *IEEE Transactions on Image Processing*, vol. 19, pp. 895-911, 2010.
- [24] J. Cisternas, T. Asahi, M. Galvez, and G. Rojas, "Regularization of Diffusion Tensor Images," in *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pp. 935-938, 2008.
- [25] CVG. University of Granada Test Image Database [Online]. Available: <http://decsai.ugr.es/cvg/dbimagenes/>
- [26] R. Deriche, "Fast Algorithms for Low-Level Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 78-87, 1990.
- [27] S. Dizenzo, "A Note on the Gradient of a Multiimage," *Computer Vision Graphics and Image Processing*, vol. 33, pp. 116-125, Jan 1986.
- [28] S. Durand, J. Fadili, and M. Nikolova, "Multiplicative Noise Removal Using L1 Fidelity on Frame Coefficients," *Journal of Mathematical Imaging and Vision*, vol. 36, pp. 201-226, Mar 2010.
- [29] A. A. Efros and T. K. Leung, "Texture Synthesis by Non-Parametric Sampling," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1033-1038 vol.2, 1999.
- [30] M. Elad, "On the Origin of the Bilateral Filter and Ways to Improve It," *Image Processing, IEEE Transactions on*, vol. 11, pp. 1141-1151, 2002.
- [31] S. Esedoglu, "An Analysis of the Perona-Malik Scheme," *Communications on Pure and Applied Mathematics*, vol. 54, pp. 1442-1487, Dec 2001.
- [32] M. Felsberg, "Autocorrelation-Driven Diffusion Filtering," *IEEE Transactions on Image Processing*, vol. 20, pp. 1797-1806, 2011.
- [33] J. B. Garnett, T. M. Le, Y. Meyer, and L. A. Vese, "Image Decompositions Using Bounded Variation and Generalized Homogeneous Besov Spaces," *Applied and Computational Harmonic Analysis*, vol. 23, pp. 25-56, Jul 2007.
- [34] R. Garnett, T. Huegerich, C. Chui, and H. Wenjie, "A Universal Noise Removal Algorithm with an Impulse Detector," *IEEE Transactions on Image Processing*, vol. 14, pp. 1747-1754, 2005.
- [35] T. Gasser, L. Sroka, and C. Jennensteinmetz, "Residual Variance and Residual Pattern in Nonlinear-Regression," *Biometrika*, vol. 73, pp. 625-633, Dec 1986.

Bibliography

- [36] S. Ghosal and R. Mehrotra, "Zernike Moment-Based Feature Detectors," in *Proceedings of IEEE International Conference on Image Processing*, pp. 934-938 vol.1, 1994.
- [37] S. Ghosal and R. Mehrotra, "A Moment-Based Unified Approach to Image Feature Detection," *IEEE Transactions on Image Processing*, vol. 6, pp. 781-793, 1997.
- [38] G. Gilboa, "Nonlinear Scale Space with Spatially Varying Stopping Time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 2175-2187, Dec 2008.
- [39] G. Gilboa and S. Osher, "Nonlocal Linear Image Regularization and Supervised Segmentation," *Multiscale Modeling & Simulation*, vol. 6, pp. 595-630, 2007.
- [40] G. Gilboa and S. Osher, "Nonlocal Operators with Applications to Image Processing," *Multiscale Modeling & Simulation*, vol. 7, pp. 1005-1028, 2008.
- [41] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Forward-and-Backward Diffusion Processes for Adaptive Image Enhancement and Denoising," *IEEE Transactions on Image Processing*, vol. 11, pp. 689-703, Jul 2002.
- [42] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Image Enhancement and Denoising by Complex Diffusion Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1020-1036, Aug 2004.
- [43] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Estimation of Optimal Pde-Based Denoising in the Snr Sense," *IEEE Transactions on Image Processing*, vol. 15, pp. 2269-2280, Aug 2006.
- [44] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Variational Denoising of Partly Textured Images by Spatially Varying Constraints," *IEEE Transactions on Image Processing*, vol. 15, pp. 2281-2289, Aug 2006.
- [45] A. Haddad and Y. Meyer, "Variational Methods in Image Processing," *UCLA CAM Report 04-52*, 2004.
- [46] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms and Results," *IEEE Transactions on Image Processing*, vol. 4, pp. 499-502, 1995.
- [47] M. S. Islam, A. Sluzek, and L. Zhu, "Detecting and Matching Interest Points in Relative Scale," *Machine Graphics and Vision*, vol. 14, pp. 259-283, 2005.
- [48] S. Islam and L. Zhu, "Matching Interest Points of an Object," in *Proceedings of IEEE International Conference on Image Processing*, pp. I-373-6, 2005.
- [49] Z. Junmei and S. Huifang, "Wavelet-Based Multiscale Anisotropic Diffusion with Adaptive Statistical Analysis for Image Restoration," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 55, pp. 2716-2725, 2008.
- [50] S. Kichenassamy, "The Perona-Malik Paradox," *Siam Journal on Applied Mathematics*, vol. 57, pp. 1328-1342, Oct 1997.
- [51] R. Kimmel, R. Malladi, and N. Sochen, "Images as Embedded Maps and Minimal Surfaces: Movies, Color, Texture, and Volumetric Medical Images," *International Journal of Computer Vision*, vol. 39, pp. 111-129, Sep 2000.
- [52] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and Denoising of Images by Nonlocal Functionals," *Multiscale Modeling & Simulation*, vol. 4, pp. 1091-1115, 2005.
- [53] S. J. Ko and Y. H. Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," *IEEE Transactions on Circuits and Systems*, vol. 38, pp. 984-993, 1991.
- [54] Kodak. Kodak Lossless True Color Image Suite [Online]. Available: <http://r0k.us/graphics/kodak/>
- [55] J. J. Koenderink, "The Structure of Images," *Biological Cybernetics*, vol. 50, pp. 363-370, 1984.

Bibliography

- [56] P. Kornprobst, R. Deriche, and G. Aubert, "Nonlinear Operators in Image Restoration," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 325-330, 1997.
- [57] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [58] Z. X. Liu, H. J. Wang, and S. L. Peng, "Image Magnification Method Using Joint Diffusion," *Journal of Computer Science and Technology*, vol. 19, pp. 698-707, Sep 2004.
- [59] M. Lysaker, A. Lundervold, and T. Xue-Cheng, "Noise Removal Using Fourth-Order Partial Differential Equation with Applications to Medical Magnetic Resonance Images in Space and Time," *IEEE Transactions on Image Processing*, vol. 12, pp. 1579-1590, 2003.
- [60] M. Lysaker, S. Osher, and T. Xue-Cheng, "Noise Removal Using Smoothed Normals and Surface Fitting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1345-1357, 2004.
- [61] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Subpixel Measurements Using a Moment-Based Edge Operator," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1293-1309, 1989.
- [62] R. Malladi and J. A. Sethian, "Image Processing: Flows under Min/Max Curvature and Mean Curvature," *Graphical Models and Image Processing*, vol. 58, pp. 127-141, Mar 1996.
- [63] Y. Meyer, "Oscillating Patterns in Image Processing and Nonlinear Evolution Equations," *University Lecture Series Volume 22*, AMS, 2002.
- [64] P. Mrazek and M. Navara, "Selection of Optimal Stopping Time for Nonlinear Diffusion Filtering," *International Journal of Computer Vision*, vol. 52, pp. 189-203, May-Jun 2003.
- [65] D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational-Problems," *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577-685, Jul 1989.
- [66] M. Nikolova, "Minimizers of Cost-Functions Involving Nonsmooth Data-Fidelity Terms. Application to the Processing of Outliers," *Siam Journal on Numerical Analysis*, vol. 40, pp. 965-994, Sep 2002.
- [67] M. Nikolova, "A Variational Approach to Remove Outliers and Impulse Noise," *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 99-120, Jan-Mar 2004.
- [68] M. Nikolova, "Weakly Constrained Minimization: Application to the Estimation of Images and Signals Involving Constant Regions," *Journal of Mathematical Imaging and Vision*, vol. 21, pp. 155-175, Sep 2004.
- [69] M. Nitzberg and T. Shiota, "Nonlinear Image Filtering with Edge and Corner Enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 826-833, 1992.
- [70] S. Osher and L. I. Rudin, "Feature-Oriented Image-Enhancement Using Shock Filters," *Siam Journal on Numerical Analysis*, vol. 27, pp. 919-940, Aug 1990.
- [71] S. Osher, A. Sole, and L. Vese, "Image Decomposition and Restoration Using Total Variation Minimization and the H-1 Norm," *Multiscale Modeling & Simulation*, vol. 1, pp. 349-370, 2003.
- [72] Y. Oshima, "On an Optimal Stopping Problem of Time Inhomogeneous Diffusion Processes," *Siam Journal on Control and Optimization*, vol. 45, pp. 565-579, 2006.
- [73] G. A. Papakostas, D. A. Karras, and B. G. Mertzios, "Image Coding Using a Wavelet Based Zernike Moments Compression Technique," in *Proceedings of*

Bibliography

- International Conference on Digital Signal Processing*, pp. 517-520 vol.2, 2002.
- [74] P. Perona and J. Malik, "Scale-Space and Edge-Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629-639, Jul 1990.
 - [75] I. Pollak, A. S. Willsky, and H. Krim, "Image Segmentation and Edge Enhancement with Stabilized Inverse Diffusion Equations," *IEEE Transactions on Image Processing*, vol. 9, pp. 256-266, 2000.
 - [76] Y. D. Qu, C. S. Cui, S. B. Chen, and J. Q. Li, "A Fast Subpixel Edge Detection Method Using Sobel-Zernike Moments Operator," *Image and Vision Computing*, vol. 23, pp. 11-17, Jan 2005.
 - [77] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, vol. 60, pp. 259-268, Nov 1992.
 - [78] G. Sapiro, "Color Snakes," *Computer Vision and Image Understanding*, vol. 68, pp. 247-253, Nov 1997.
 - [79] G. Sapiro and D. L. Ringach, "Anisotropic Diffusion of Multivalued Images with Applications to Color Filtering," *IEEE Transactions on Image Processing*, vol. 5, pp. 1582-1586, Nov 1996.
 - [80] J. Shah, "A Common Framework for Curve Evolution, Segmentation and Anisotropic Diffusion," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136-142, 1996.
 - [81] N. Sochen, R. Kimmel, and R. Malladi, "A General Framework for Low Level Vision," *IEEE Transactions on Image Processing*, vol. 7, pp. 310-318, 1998.
 - [82] X.-C. L. Tai, Knut-Andreas; Chan, Tony F., *Image Processing Based on Partial Differential Equations* Heidelberg: Springer Verlag, 2006.
 - [83] A. N. Tikhonov, "Regularization of Incorrectly Posed Problems," *Doklady Akademii Nauk Sssr*, vol. 153, pp. 49-&, 1963.
 - [84] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 839-846, 1998.
 - [85] D. Tschumperle, "Pdes Based Regularization of Multivalued Images and Applications," PhD Thesis, Univ. Nice–Sophia Antipolis, Sophia Antipolis, France, 2002.
 - [86] D. Tschumperle, "Curvature-Preserving Regularization of Multi-Valued Images Using Pde's," in *Proceedings of European Conference on Computer Vision, Pt 2*, vol. 3952, A. Leonardis, et al., Eds., ed, pp. 295-307.
 - [87] D. Tschumperle and L. Brun, "Non-Local Image Smoothing by Applying Anisotropic Diffusion Pde's in the Space of Patches," in *Proceedings of IEEE International Conference on Image Processing*, pp. 2957-2960, 2009.
 - [88] D. Tschumperle and R. Deriche, "Vector-Valued Image Regularization with Pdes: A Common Framework for Different Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 506-517, Apr 2005.
 - [89] USC. Use Sipi Test Image Database [Online]. Available: <http://sipi.usc.edu/services/database/>
 - [90] L. Vese and S. Osher, "Modelling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing," *UCLA CAM Report 02-19*, 2002.
 - [91] J. Weickert, "Anisotropic Diffusion in Image Processing," PhD Thesis, Laboratory of Technomathematics, University of Kaiserslautern, Germany, 1996.
 - [92] J. Weickert, "A Review of Nonlinear Diffusion Filtering," in *Scale-Space Theory in Computer Vision*. vol. 1252, B. terHaarRomeny, et al., Eds., ed, pp. 3-28.

Bibliography

- [93] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart: Teubner-Verlag, 1998.
- [94] J. Weickert, "Coherence-Enhancing Diffusion Filtering," *International Journal of Computer Vision*, vol. 31, pp. 111-127, Apr 1999.
- [95] J. Weickert and H. Scharr, "A Scheme for Coherence-Enhancing Diffusion Filtering with Optimized Rotation Invariance," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 103-118, Mar-Jun 2002.
- [96] J. F. Yang, Y. Zhang, and W. T. Yin, "An Efficient Tvll Algorithm for Deblurring Multichannel Images Corrupted by Impulsive Noise," *Siam Journal on Scientific Computing*, vol. 31, pp. 2842-2865, 2009.
- [97] Y. Yu-Li, X. Wenyuan, A. Tannenbaum, and M. Kaveh, "Behavioral Analysis of Anisotropic Diffusion in Image Processing," *IEEE Transactions on Image Processing*, vol. 5, pp. 1539-1553, 1996.
- [98] v. F. Zernike, "Beugungstheorie Des Schneidenver-Fahrens Und Seiner Verbesserten Form, Der Phasenkontrastmethode," *Physica*, vol. 1, pp. 689-704, 1934.
- [99] L. Zhu and S. Islam, "An Adaptive Edge-Preserving Variational Framework for Color Image Regularization," in *Proceedings of IEEE International Conference on Image Processing*, pp. I-101-4, 2005.
- [100] L. Zhu and A. Sluzek, "Color Similarity Distribution Based Edge Detection," in *International Workshop on Advanced Image Technology (IWAIT'04)*, Singapore, 2004.
- [101] L. Zhu, A. Sluzek, and M. D. Saiful Islam, "An Adaptive Edge Preserving Variational Method for Color Image Regularization," in *Visual Communications and Image Processing 2005, 12-15 July 2005, USA*, pp. 59605-1, 2005.