4-14-2017

Mathematical Approaches to Digital Image Inpainting

Sumudu S. Kalubowila
sskw29@umsl.edu

Mathematical Approaches to Digital Image Inpainting

Sumudu Samanthi Kalubowila

M.Sc., Mathematics, University of North Florida, Jacksonville, 2011

B.Sc., Com. Mathematics, University of Colombo, Sri Lanka, 2008

A dissertation submitted to the Graduate School of the
University of Missouri-St. Louis
in partial fulfillment of the requirments for the degree
Doctor of Philosophy in Applied Mathematics

May 2017

Advisory Committee:

Qingtang Jiang,Ph. D. (Chair)

Haiyan Cai,Ph. D.

Wenjie He,Ph. D.

Yuefeng Wu,Ph. D.

Acknowledgements

First I would like to thank my advisor Professor Charles K. Chui for introducing me to the field of image inpainting, providing me some challenging problems to the work on. I would like to pay my heartfelt gratitude to Professor Chui for his guidance throughout my dissertation. Without Professor Chui's help and guidance, I would not have completed my dissertation. I also thank Professor Qingtang Jiang for his kind help and guidance with my dissertation. I believe that I learned at least a bit from both renowned professors about how to be innovative in applying mathematics in real life applications. When looking back, there were many brainstorming sessions with two ingenious professors, and it is so impressive to recall those moments.

I'd like to thank all of the committee members for being part of my dissertation. I'd like to pay my special thanks to Professor Wenjie He for direction in computer algorithms. I must appreciate and cannot understate the invaluable support and kindness given by course coordinator for my teaching Shala Peterman throughout the journey to my graduation. Also, I would like to thank my current course coordinator, Qiang Dotzel. The current administrative staff of Kimberly Stranger, Alice Canavan and former administrative staff of Raina Traore-Gress gave their tiresome support in paperwork which relieved me in stressful

situations. Also I want to thank professor Ravindra Girivaru for encouraging me in stressful situations.

My dream of graduating would not be real without the financial support of the department. I have the deepest appreciation for the department and also feel pleased to be able to complete the research work. On the other hand, I do appreciate the department for giving me the opportunity to teach, which gave me so much inspiration for my future career. Furthermore, I must appreciate the department chair for offering me an adjunct position in the department to get the experience as a staff member which is really helpful to my career.

Last but not least, I give my thanks to my family for their loving kindness throughout my graduation process. They eased my studies by caring for me in my stressful situations and keeping my mind away from day to day work.

Abstract

Image inpainting process is used to restore the damaged image or missing parts of an image. This technique is used in some applications, such as removal of text in images and photo restoration. There are different types of methods used in image inpainting, such as non-inear partial differential equations(PDEs), wavelet transformation and framelet transformation.

We studied the usage of the current image inpainting methods and solved the Poisson equation using a five-point stencil method. We used a modified five-point stencil method to solve the same equation. It gave better results than the standard five-point stencil method. Using modified five-point stencil method results as the initial condition, we solved the iterative linear and non-linear diffusion PDE. We considered different types of diffusion conductivity and compared their results. When compared with PSNR values, the iterative linear diffusion PDE method gave the best results where as constant diffusion conductivity PDE gave the worst result. Furthermore, inverse diffusion conductivity PDE had given better results than that of the constant diffusion PDE. However, it was worse than the Gaussian and Lorentz diffusion conductivity PDE. Gaussian and Lorentz diffusion conductivity iterative linear PDE had given a better result for image inpainting.

When we use any inpainting technique, we cannot restore the original image. We studied the relationship between the error of the image inpainting and the inpainted domain. Error is proportional to the value of the Greens function. There are two types of methods to find the Greens function. The first method is solving a Poisson equation for a different shape of domain, such as a circle, ellipse, triangle and rectangle. If the inpainting domain has a different shape, then it is difficult to find the error. We used the conformal mapping method to find the error. We also developed a formula for transformation from any polygon to the unit circle. Moreover, we applied the Schwarz Christoffel transformation to transform from the upper half plane to any polygon.

Contents

Acknowledgements **i**

Abstract **iii**

Contents **v**

List of Figures **viii**

List of Table **xii**

List of Abbreviation **xiv**

1 Introduction **1**

2 Background **4**

2.1 Texture Image inpainting 4

2.2 Local non-linear PDE image inpainting 5

2.3 Digital Image de-noising using Wavelet based minimal energy 6

2.4 Tensor product complex tight framelets 6

2.5 Fast Digital Image inpainting 6

2.6 Anisotropic Diffusion Linear PDE 8

2.7 Domain Transformation 8

2.7.1 Wavelet Decomposition 8

2.8 Transformation and Optimization 10

2.8.1 Complex wavelet transformation 10

2.8.2 Framelet transformation 11

2.9 Applications 13

2.9.1 Object Removal 13

2.9.2 Photo Restoration 14

2.9.3 Text Removal 15

2.10 Error of Image Inpainting 15

2.11 Extension of Digital image inpainting 16

2.11.1 Video Inpainting 16

2.11.2 Inpainting in 3 dimensions 17

3 Finite Difference Method(FDM) 18

4 Diffusion Partial Differential Equation 22

4.1 Non Linear Diffusion PDE 23

4.2 Diffusion Conductivity 24

5 The Green Function of a Boundary Value problem 27

6 Mapping 30

6.1 Harmonic Function 30

6.1.1 Simply connected domain 30

6.1.2 Analytic Function 31

6.2 Composition of Function 31

6.2.1 Chain Rule 31

6.3 Conformal Mapping 32

6.3.1 Mӧbius Transformation 33

6.3.2 Joukowski Transformation 35

6.4 Schwarz Christoffel Mapping 37

6.4.1 Schwarz Christoffel Mapping for a Triangle 39

6.5 Application of Conformal Mapping 43

7 Multi Resolution Approximation for Image Inpainting 44

7.1 Process of MRA 48

8 Mathematical Approaches 51

8.1 Image Inpainting Methods 51

8.1.1 Initial Value 51

8.1.2 Non-Linear Diffusion PDE and Iterative Linear Diffusion PDE 58

8.2 Error Analysis of Image Inpainting 68

8.3 Error using Poisson Equation 69

8.3.1 Circle,[20] 70

8.3.2 Ellipse 71

8.3.3 Triangle 73

8.3.4 Rectangle 76

8.4 Schwarz Christoffel Transformation from any polygon to unit disk 82

8.4.1 Erros using conformal mapping 84

9 Future Work 86

List of Figures

1.1 $u\Big|_{D^c}$ is the known data and D is the inpainting domain D [28] 2

2.1 Texture Oriented Image Inpainting Algorithm [19, 42] . 5

2.2 Application of Texture Oriented Image Inpainting [19, 42] 5

2.3 Inpainting domain (left bottom). Result of diffusion process without *barriers*. It has blury edges (left top). Define diffusion barriers (right bottom). Result of diffusion process with barriers. It has sharp edges (right top). [17] 7

2.4 Line is the inpainting domain (left). Define two *diffusion barriers* at the end of hair (middle). Result of fast digital image inpainting (right) [17] 7

2.5 (a) the original image; (b) damaged image; (c) the result of discrete cosine transformation decomposition ; (d) the result of wavelet decomposition. [11] 9

2.6 (a) Damage image, (b) results of method [39], (c) results of method [40], (d) results of method [12]. 10

2.7 (a) The damage image. (b) Inpainted image by using DWT (c) Inpainted image by using framelet(d) Inpainted image by using the [16] 11

2.8 Left: original image. Middle: inpainting mask. Right: result of image ipainting [13] 13

2.9 The Commissar vanishes in second picture. [43] 14

2.10 Left: damage picture of Cornelia, Mother of the Gracchi, Right: result of image inpainting. [43] 14

2.11 Left: damage photo, Right: result of image inpainting. [43] 14

2.12 Left :Original Image. Right: After removal of text [28] . 15

2.13 1st row: damage images. 2nd row: inpainted result [23] . 15

2.14 1st row: frames of video. 2nd row: user-defned mask in black. 3rd row: results video inpainting using method [42] 16

2.15 3D inpainting 17

3.1 Finite Difference along x-axis 18

3.2 5-point stencil for Laplace equation 20

3.3 2D Grid with boundary conditions 20

4.1 Isotropic Diffusion PDE Inpainting 24

4.2 TV Inpainting method. Top: Orginal Image and Bottom : Inpainted Image 25

4.3 CDD Inpainting method. Top: Orginal Image and Bottom : Inpainted Image 25

6.1 simply connected domain 30

6.2 Compostion of Function, from A to C. $g \circ f : A \to C$ . . 31

6.3 preserves angle 32

6.4 Transformation from upper half plane to the unit circle . 33

6.5 Transformation from any circle to ellipse 35

6.6 Transformation from upper half plane to any polygon . . 37

6.7 Sum of the angles on the straight line is eqaul to the π . 38

6.8 Schwarz Christoffel Mapping form upper Half plane to Triangle 39

8.1 General 2D Grid with n rows and m column 51

8.2 5-Point apply Level by level 53

8.3 5-Point apply to Level 1 53

8.4 5-Point apply to Level 2 55

8.5 Apply 5-point method level by level. Purple is the Boundary data. 55

8.6 Standard 5-Point Stencil method and Modified 5-point Stencil 57

8.7 Image Inpainting methods with Constant Diffusion . . . 61

8.8 Image Inpainting methods with inverse proportional diffusion conductivity 63

8.9 Image Inpainting methods with inverse proportional diffusion conductivity 63

8.10 Image Inpainting methods with Gaussian Diffusion Conductivity 65

8.11 Image Inpainting methods with Gaussian Diffusion Conductivity 65

8.12 Image Inpainting methods with Lorentz Diffusion Conductivity 67

8.13 Image Inpainting methods with Lorentz Diffusion Conductivity 67

8.14 circle with center zero and radius r 70

8.15 Ellipse with center zero and a $>$ b 72

8.16 Triangle with each side is 2a 74

8.17 Rectangle with a>b 77

8.18 Schwarz Christoffel Transformation from any unit disk to polygon 82

List of Tables

2.1 PSNR values and processing times for each methods . . . 11

8.1 PSNR value for Standard 5-Point Stencil method and Modified 5-point Stencil 57

8.2 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Constant Diffusion Conductivity . 61

8.3 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Inverse Proportional Diffusion Conductivity 63

8.4 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Inverse Proportional Diffusion Conductivity 64

8.5 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Gaussian Proportional Diffusion Conductivity 65

8.6 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Gaussian Proportional Diffusion Conductivity 66

8.7 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Lorentz Proportional Diffusion Conductivity 67

8.8 PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Lorentz Proportional Diffusion Conductivity 68

List of Abbreviation

PDE : Partial Differential Equation

MRA: Multi Resolution Approximation

TV : Total Variation

FD : Finite Difference

CDD : Curvature Driven Diffusion

CWT : Complex Wavelet Transformation

DWT : Discreate Wavelet Transformation

PSNR : Peak Signal Noise Ratio

FDM : Finite Difference Method

MAX : MAXimum posible pixel value of the image

MSE : Mean Square Error

Chapter 1

Introduction

Inpainting has been carried out by professional artists for many years. When done manually, it is a very time-consuming process. The basic idea of this process is to reconstruct damaged or missing parts of an image. It has important value in restoration of old photographs, the removal of artifacts in a film, the removal of red eye, the removal of superimposed text, and the removal of redundant objects. At the 2000 SIGGRAPH conference, the idea of digital inpainting was established by Bertalmio-Sapiro-Caselles-Ballester [13]. Image inpainting has been expanding very fast. It is a very important topic in the field of Digital Image Processing.

Nowadays, data exchange has become popular. Since time and skill are required to do image inpainting manually, it is important to find an automatic and efficient method. Therefore, different types of successful inpainting techniques were developed in the last few yaers.

The idea of the computer algorithm of image inpainting is to fill the missing data with known data surrounding D. D is the *inpainting domain.*

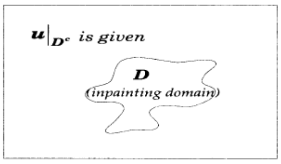


Figure 1.1: $u\Big|_{D^c}$ is the known data and D is the inpainting domain D [28]

Our first approach is to solve linear diffusion PDE. When we solve this, we need the initial values as boundary conditions. We have to solve the Laplace equation for Dirichlet boundary conditions in the rectangular domain. We use the five-point stencil method to find them. Also, we did some modifications of the five-point stencil method. Using this solution, we solved the linear PDE. We have a numerical method to solve non-linear PDE. Here we develop that technique to solve linear diffusion PDE. In 1990, Perona and Malik considered the non-constant diffusion conductivity; we extended our PDE method for constant and non-constant diffusion conductivity. We wrote a MATLAB program for these linear and non-linear PDE with different diffusion conductivity. We took an original image and removed small areas; then, using these inpainting methods tried to inpaint them. After that, we compared each method outputs with the original image.

Normally, we can't get the original image; we try to find the relationship between inpainting domain and error of the image inpainting. Here, we consider two different methods. For a specific shape of domain, such as circle, ellipse, triangle and rectangle, I solve Poisson PDE to estimate the error of image inpaiting. If we have a different shape, such as a polygon, we consider conformal mapping method to solve PDE. We extend the Schwarz Christoffel method to find a conformal map from polygon to unit disk. Using this method, we can also develop a relationship between the error and the inpainting domain.

This dissertation is organized in nine sections. Chapter 2 describes the background of image inpainting. In this chapter, we can get an idea about current methods of image inpainting. Also here we explain application of image inpainting. When we use inpainting methods, we can't get the original image. So in chapter 2, we discussed the error of the image inpainting. We can develop this inpainting technique for 3D inpainting and video inpainting. Chapter 3 reviews the elliptic partial differential equation. Here we can get an idea about solving the Poisson equation with Dirichlet boundary conditions. When we solve PDE, we use FDM(Finite Difference method). In chapter 4, we discuss the Diffusion PDE(Partial Differential Equation). In that section, we can get an idea of linear and non-linear diffusion PDE and numerical methods of solving non-linear diffusion PDE. We also discuss different types of diffusion conductivity.

The next three chapters(5, 6 and 7) are important for calculation of error analysis of image inpainting. In chapter 5, we consider the relationship between green function and PDE. Different types of conformal mapping details are presented in chapter 6. In chapter 7, we discuss the process of the Multi Resolution Approximation method in image inpainting. In Chapter 8, we explain our methods of image inpainting and their outputs, and Chapter 9 has final remarks and our conclusions.

Chapter 2

Background

The success of the image inpainting techniques depend on type of image, size of image, and characteristics of the lost data. A lot of appropriate techniques have been developed in recent years.

2.1 Texture Image inpainting

In a 1995 paper, Heeger and Bergen [25] proposed a method related to the texture of image inpainting by matching marginal histograms of subband transformation. Efros and Leung[27] developed a technique for texture synthesis in 1999. In this method, they tried to find an approximately similar part of the image to fill the missing area. First, they chose a small part of the missing area, close to the boundary of D (Figure 1). Then they considered the known data closer to that pixel. Then they scanned the whole image to find similar data matching that information. They used that known data to fill that missing information. This technique is perfect when the image has patterns[35].

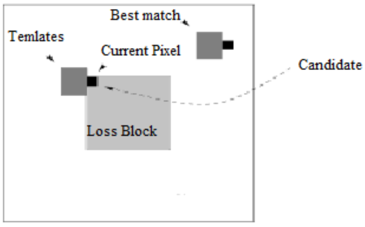


Figure 2.1: Texture Oriented Image Inpainting Algorithm [19, 42]

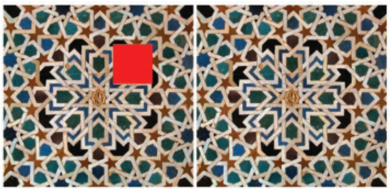


Figure 2.2: Application of Texture Oriented Image Inpainting [19, 42]

2.2 Local non-linear PDE image inpainting

In local inpainting, the missing data is to be inpainted by using image data available in the neighborhood of the missing area[1]. Heat, gas and fluid are immediately moved from the area of higher concentration to the area of lower concentration.Therefore, we can compare them with the process of image inpainting. The major idea of image inpainting is the numerical result of partial differential equation, such as Isotropic Diffusion, TV(Total variation) inpainting [28, 15, 29, 30, 1] and CDD(Curvature-Driven Diffusion) [15, 1], or Navier-Stokes equation for incompressible fluids [18].

2.3 Digital Image de-noising using Wavelet based minimal energy

In 2007, Charles K. Chui and J. Wang considered the total energy function in the wavelet domain. Total energy is the sum of internal energy and external energy. Internal energy describes the quality of image.(i.e smoothness and feature of the image.)

$$TotalEnergy = E_i + \lambda E_e = \int_\Omega \rho(\mid \bigtriangledown u \mid) \,\mathrm{d}x + \frac{\lambda}{2} \int_\Omega (u - u_0)^2 \,\mathrm{d}x \quad (2.3.0.1)$$

Where λ is the adjusting term between internal energy and external energy. In this paper [21], Charles K. Chui and J. Wang used some wavelet transformation as a gradient operator in the internal energy. Using that, they got a better feature. They considered constant weight, soft wavelet thresholding and hard thresholding as energy density functions.

2.4 Tensor product complex tight framelets

Bin Han discussed a comprehensive theory and construction of directional complex tight framelets. To increase directionality, he used a family of tensor product complex tight framelets. Also, he considered orthogonal wavelet filter banks. He used this technique for image denoising and got very good output.

2.5 Fast Digital Image inpainting

We can reconstruct a damaged image quickly by using a fast digital image inpainting algorithm [17]. This method can be used to get perfect edge reconnection. Here we used *diffusion barriers*, which are used to separate two different areas. It is a very tiny line, only two pixels wide. First we want to put *diffusion barriers* to our image. Then we can apply a diffusion process for each component separately. The process stops when it is close to *diffusion barriers* . Finally, we use the diffusion process to fill that line segment of *diffusion barriers*. If we apply the diffusion process without *diffusion barriers*, then we have blury edges.

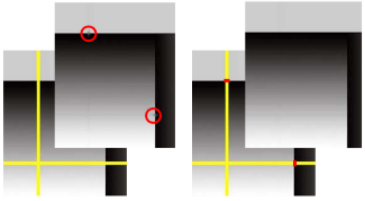


Figure 2.3: Inpainting domain (left bottom). Result of diffusion process without *barriers*. It has blury edges (left top). Define diffusion barriers (right bottom). Result of diffusion process with barriers. It has sharp edges (right top). [17]

Application of fast digital image inpainting with two barriers are shown below.

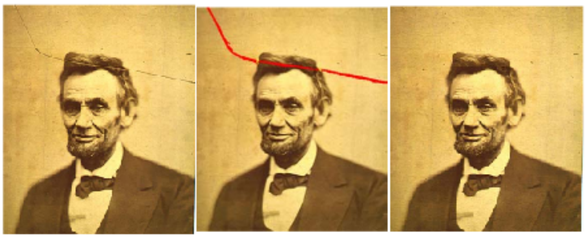


Figure 2.4: Line is the inpainting domain (left). Define two *diffusion barriers* at the end of hair (middle). Result of fast digital image inpainting (right) [17]

2.6 Anisotropic Diffusion Linear PDE

In 2009, [20] C. K. Chui developed a multi-resolution approximation method for image inpainting and surface completion. Here he use partial differential equation of anisotropic diffusion to known data .

$$\left.\begin{aligned} \frac{\partial}{\partial t}u_j &= \bigtriangledown \cdot (c(\mid \bigtriangledown u_{j-1} \mid) \bigtriangledown u_j) \quad in\ D, t \geq 0 \\ \frac{\partial}{\partial \mathbf{n}}u_j\Big|_{\partial D} &= 0 \\ u_j(z,0) &= u_0(z), \qquad z\ \in\ D \end{aligned}\right\}$$

where j=1, 2, ... and $u^0 = u_0$

Here we have a set of linear partial differential equations. This is also a local image inpainting method.

In that paper, they set up an error of inpainting using the heat kernels.

2.7 Domain Transformation

We can transform image data into a different domain by using a transformation method such as orthonormal wavelet transformation, discrete Fourier transformation, discrete cosine transformation, discrete sine transformation, and tight framelets transformation. Then, we apply any inpainting technique for that transformed data. Finally, we apply the inverse transformation for that corresponding transformation method to get back the original data.

2.7.1 Wavelet Decomposition

In 2012, [11] Hongying Zhang and Shimei found a method for image inpainting by using wavelet decomposition. It had three steps, namely

Image decomposition, structure inpainting, and texture inpainting. In the first step, they decomposed the original image into four parts. They used the wavelet transformation for this process because it was not losing image information and it was reconstructing quality data. They used those four parts as one section for texture inpainting and three sections for structure inpainting. Here they used a texture synthesis algorithm for inpainting texture parts. We can use Chan T F.and Shen J H's. [15] method for structure inpainting. We apply it for three directions, such as horizontal, vertical and diagonal. Finally, we apply inverse wavelet transformation to get back the original image.

We can use discrete cosine transformation for image decomposition. Here they decomposed images in frequency domains. In figure 6, we compare these two decomposition methods.

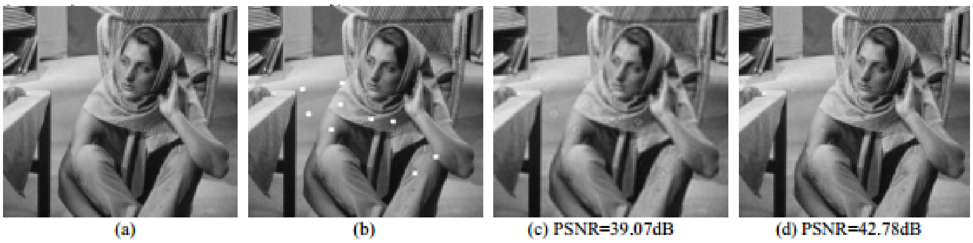


Figure 2.5: (a) the original image; (b) damaged image; (c) the result of discrete cosine transformation decomposition ; (d) the result of wavelet decomposition. [11]

Projected Onto Convex Sets (POCS)

In 2009, [39] Hirani and Totsuka developed a technique for image denoising and removal of scratches in the image. Here they joined frequency and spatial domain data. In 2003, [40] Patwardhan and Sapiro developed a method for image inpainting using the wavelet transformation. In both methods, they used the idea of Projected Onto Convex Sets. It helps to improve the texture of the image. Fast marching method can restore the known image data first. In a 2001 [12] paper, Patwardhan and Sapiro considered fast marching method and Patwardhan, Sapiro's method [40]. In that paper, they mentioned their method processing time is less than half of Patwardhan, Sapiro's method [40].

Here we compare three inpainting methods outputs,

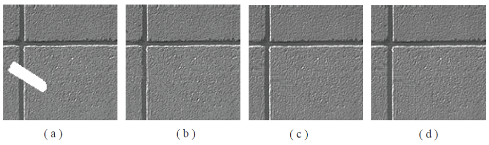


Figure 2.6: (a) Damage image, (b) results of method [39], (c) results of method [40], (d) results of method [12].

2.8 Transformation and Optimization

The out-put of a wavelet transformation or framelet transformation is not absolutely sparse. It becomes sparse after thresholding, in the sense that most of the elements of the vectors recieved from such a transformation become zero when the small values are replaced by zero.

2.8.1 Complex wavelet transformation

The Complex Wavelet Transformation($\mathbb{C}$WT) and real wavelet transformation are different because they have different shift-invariance (small shift into input signal) , anti-aliasing and directional selectivity [31]. The dual tree $\mathbb{C}$WT method can be used for image denoising, classification and segmentation. $\mathbb{C}$WT is faster than other methods, when $\mathbb{C}$WT and other wavelet-based image processing methods have the same wavelet [16].

In figure 2.7 and table, we can compare different methods with their processing time, number of iterations and PSNR values [16].

image	number of iterations	PSNR	processing time(seconds)
b	500	30.56	18.00
c	213	32.06	86.29
d	271	32.124	38.10

Table 2.1: PSNR values and processing times for each methods

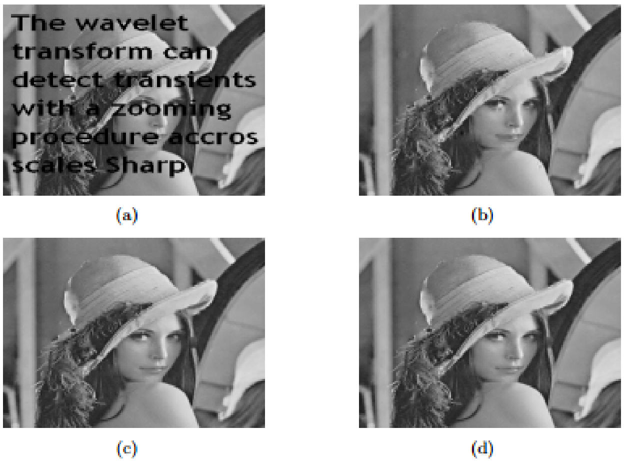

The wavelet transform can detect transients with a zooming procedure accros scales Sharp
(a)
(b)
(c)
(d)

Figure 2.7: (a) The damage image. (b) Inpainted image by using DWT (c) Inpainted image by using framelet(d) Inpainted image by using the [16]

2.8.2 Framelet transformation

When we are doing tight frame transformation for image inpainting, we assume images have sparse approximation.

Simultaneous cartoon and texture inpainting

There are two different characteristic layers in images. They are the cartoon layer and the texture layer. Cartoon layers have piecewise smooth function, and texture layers have oscillating function. Both of them have sparse representation when we use framelet transformation. In [32], Z.Shen applied tight framelet transformations for both cartoon and texture layers separately. In their paper, they developed algorithms to find solutions of minimization problems. Also, they proved those algorithms were convergent They used the proximal forward-backward splitting method , when they found numerical solutions of the minimization problem.

Split Bregman methods

We can use L_1 regularization for signal processing, statistics and computer science. Compressed sensing, wavelet thresholding and signal recovering are the applications of signal processing. Normally, L_1-regularized optimization problems are hard to solve. In 2009 [33], S. Osher and T. Goldsten used the "split bregman" method to solve these type of problems.

In [14] used Split Bregman was udes method for image inpainting. First, they transform the known data into the framelet domain,

$$f_{n+1} = P_\Lambda g + (I - P)A^* A A f_n \tag{2.8.2.1}$$

Then they applied the soft thresholding operator τ_λ to our equation to get the framelet inpainting algorithm:

$$f_{n+1} = P_\Lambda g + (I - P)A^* A \tau_\lambda (A f_n) \tag{2.8.2.2}$$

Now data is in the framelet domain. They have an optimization problems, when they are transforming data into framelet domain to image domain. Different types of methods were used to solve this optimization problem such as Bergman and split Bergman methods [8, 9]. This is a convergent algorithm. It is proved in [7, 34, 36].

There are two types of thresholding methods. They are soft-thresholding and hard-thresholding. Normally we use hard-thresholding for linear approximation schemes and soft thresholding for nonlinear approximation schemes. Thresholding operators can remove noise in the image and distribute the framelet coefficient.

2.9 Applications

A few applications of image inpainting are restoration of damaged film, removal of dust spots in films, removal of scratches from old photographs, removal of red eye, removal of superimposed text and the removal of entire objects .

2.9.1 Object Removal

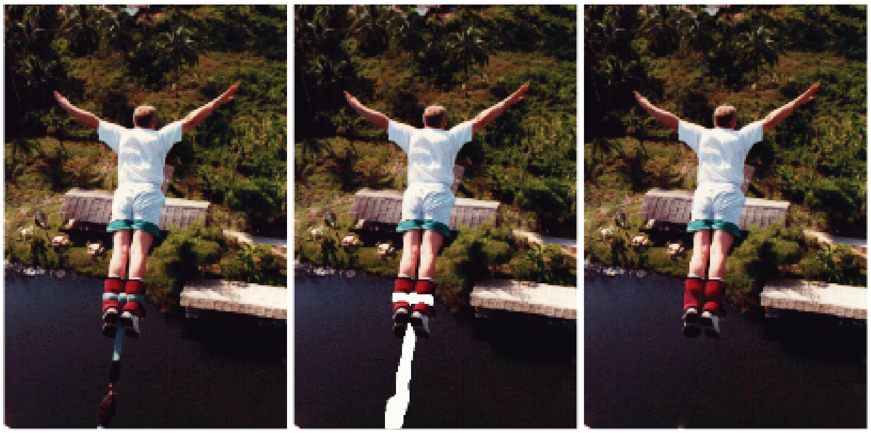


Figure 2.8: Left: original image. Middle: inpainting mask. Right: result of image ipainting [13]

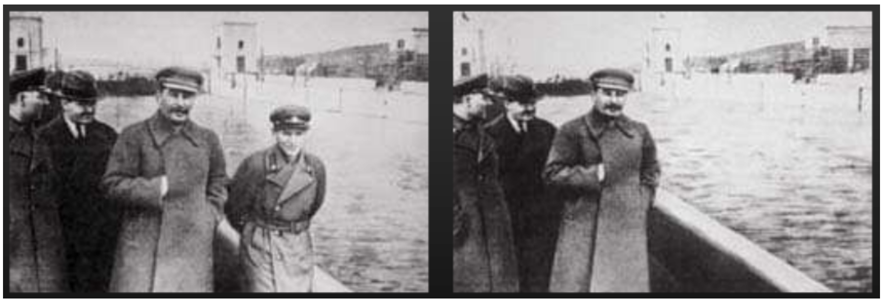


Figure 2.9: The Commissar vanishes in second picture. [43]

2.9.2 Photo Restoration

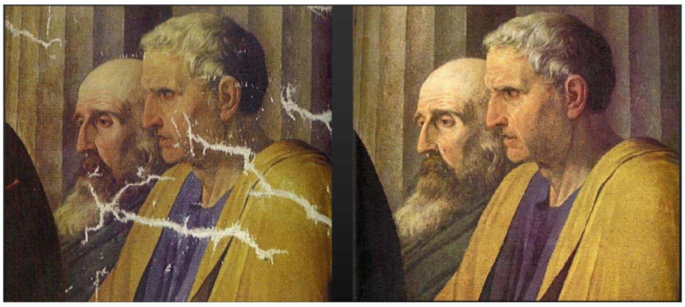


Figure 2.10: Left: damage picture of Cornelia, Mother of the Gracchi, Right: result of image inpainting. [43]

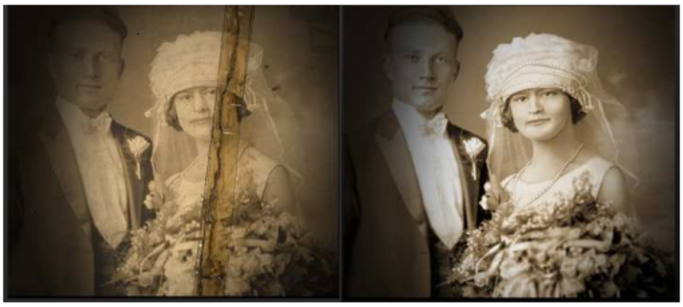


Figure 2.11: Left: damage photo, Right: result of image inpainting. [43]

2.9.3 Text Removal

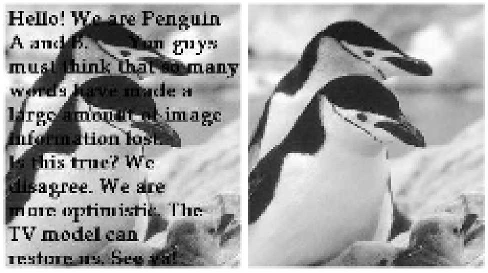


Figure 2.12: Left :Original Image. Right: After removal of text [28]

2.10 Error of Image Inpainting

Error of the image inpainting depends on the width of the inpainting domain. A lot of inpainting methods give perfect output when the inpaiting domain is narrow. In figure 14, both inpainting images and total area of inpainting domain are the same. But we have different outputs.

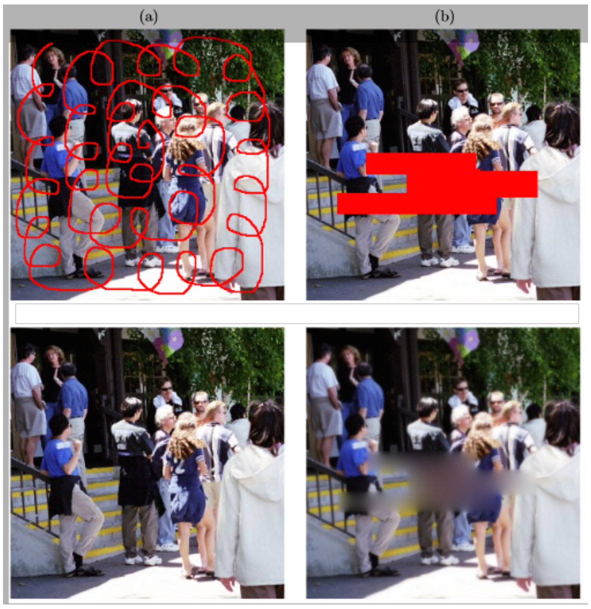


Figure 2.13: 1st row: damage images. 2nd row: inpainted result [23]

In [23],Tony F. Chan and Sung Ha Kang find the error using this formula.

$$error \leq kd^2 \tag{2.10.0.1}$$

where d is the radius of a disk covering the inpainting domain D and k is a constant .

In 2010 [22], Charles K. Chui developed a formula for the error in terms of the local volume (i.e.Width of the inpainting domain).

2.11 Extension of Digital image inpainting

2.11.1 Video Inpainting

First we separate video into frames and apply image inpainting technique for each frame. Finally, we add those frames together [18].

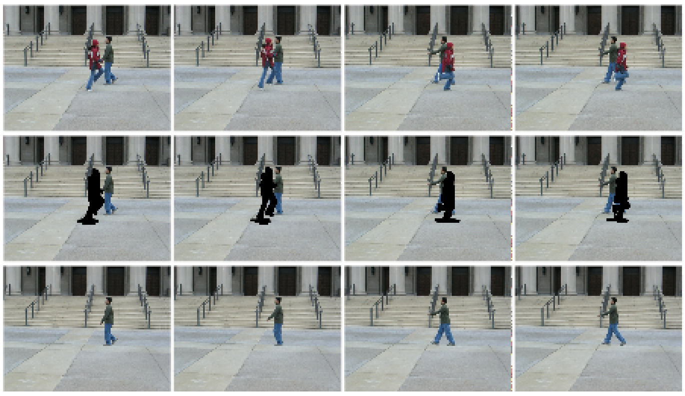


Figure 2.14: 1st row: frames of video. 2nd row: user-defned mask in black. 3rd row: results video inpainting using method [42]

2.11.2 Inpainting in 3 dimensions

The surface Inpanting algorithm is the same as Image Inpainting algorithm. The only difference is, surface inpainting has 3 dimensions.[41]

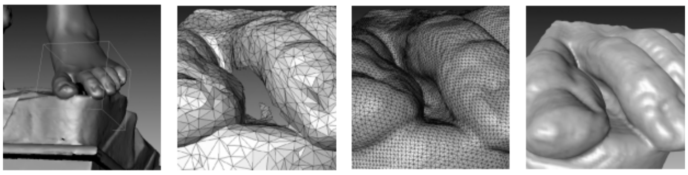


Figure 2.15: 3D inpainting

Chapter 3

Finite Difference Method(FDM)

We consider the Laplace equation with Dirichlet boundary condition.

$$\begin{aligned} \bigtriangledown^2 u(z) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad u \in D \\ u = f \quad on \quad \partial D \end{aligned} \tag{3.0.2.1}$$

Consider any three points on the x-axis with distance h, such as

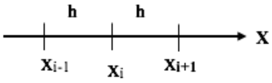


Figure 3.1: Finite Difference along x-axis

We use the taylor series expansion for x at x_i. Consider the centered-difference formula,

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \tag{3.0.2.2}$$

Consider any three points on the y-axis with distance h.

Similarly we can write,

$$\frac{\partial^2 u}{\partial y^2} = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2} \tag{3.0.2.3}$$

adding equation(3.0.2.2)and equation(3.0.2.3),

$$\begin{aligned}\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial y^2} = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \\ + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2}\end{aligned}$$

Substitute this values into the equation(3.0.2.1),

$$\begin{aligned}\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + \\ \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2} = 0 \\ u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j) + \\ u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}) = 0\end{aligned}$$

$$\begin{aligned}4u(x_i, y_j) &= u(x_{i+1}, y_j) + u(x_{i-1}, y_j) \\ &\quad + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) \\ u(x_i, y_j) &= \frac{u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1})}{4}\end{aligned}$$

Now consider the square domain with boundary conditions.,

Apply the 5-point stencil method for each u_i, where i=1,2,3,...,9.

For the first row,

$$-4u_1 + u_2 + u_4 = -F_1 - F_4$$

$$u_1 - 4u_2 + u_3 + u_5 = -F_2$$

$$u_2 - 4u_3 + u_6 = -F_3 - F_5$$

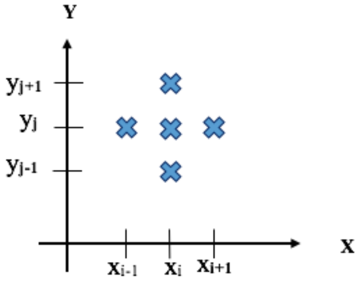


Figure 3.2: 5-point stencil for Laplace equation

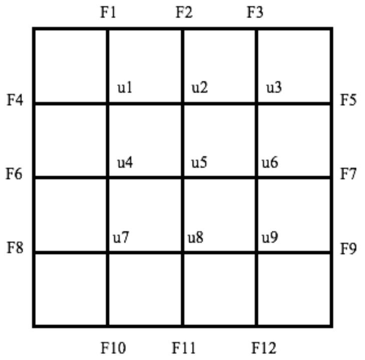


Figure 3.3: 2D Grid with boundary conditions

For the second row,

$$u_1 - 4u_4 + u_5 + u_7 = -F_6$$

$$u_2 + u_4 - 4u_5 + u_6 + u_8 = 0$$

$$u_3 + u_5 - 4u_6 + u_9 = -F_7$$

For the third row,

$$u_4 - 4u_7 + u_8 = -F_8 - F_{10}$$

$$u_5 + u_7 - 4u_8 + u_9 = -F_{11}$$

$$u_6 + u_8 - 4u_9 = -F_9 - F_{12}$$

Now we can convert this system of linear equations into the matrix.

$$\left[\begin{array}{ccc|ccc|ccc} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{array}\right] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} -F_1 - F_4 \\ -F_2 \\ -F_3 - F_5 \\ -F_6 \\ 0 \\ -F_7 \\ -F_8 - F_{10} \\ -F_{11} \\ -F_9 - F_{12} \end{bmatrix}$$

$$Au = F$$

Where A is a invertible square matrix. Therfore value of u_i is given by,

$$u = A^{-1}F$$

Also A is a block matrix. We can rewrite this matix as,

$$A = \begin{bmatrix} B & I & O \\ I & B & I \\ O & I & B \end{bmatrix}$$

where,

$$B = \begin{bmatrix} -4 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & -4 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Chapter 4

Diffusion Partial Differential Equation

Diffusion equation is a partial differential equation(PDE). It is used to understand the density fluctuation in a material.

$$\frac{\partial}{\partial t}u = \bigtriangledown \cdot (c(| \bigtriangledown u |) \bigtriangledown u) \quad in\ D, t \geq 0$$

where, c(| $\bigtriangledown u$ |) is the diffusion coefficient.
If the diffusion coefficient is a constant then it is called *Isotropic diffusion PDE.*
If the diffusion coefficient is non constant then it is called *Anisotropic diffusion PDE.*
In 1990, Perona and Malik introduced the non-constant diffusion conductivity [24].

Consider the anisotropic diffusion equation,

$$\bigtriangledown \cdot (c(| \bigtriangledown u |) \bigtriangledown u) = \bigtriangledown(c(| \bigtriangledown u |)) \cdot \bigtriangledown u + c(| \bigtriangledown u |)\Delta u \qquad (4.0.2.1)$$

4.1 Non Linear Diffusion PDE

Here, we consider the PDE with Neumann boundary condition.

$$\begin{aligned}\frac{\partial u}{\partial t} &= -f(u), \quad in \quad D \quad t \geq 0,\\ u(z,0) &= u_0(z), \quad z \quad in \quad D,\\ \left.\frac{\partial u}{\partial \mathbf{n}}\right|_{\partial D} &= 0,\end{aligned}$$

Here, **n** is the unit inner normal vector.

Therefore, numerical solution of this PDE is,

$$u(z,t_{k+1}) = u(z,t_k) - \tau_k f(u(z,t_k)) \quad k = 0,1,2,3,... \tag{4.1.0.2}$$

where τ_k is the step size.

Now we consider the heat diffusion equation with heat conductivity $c(\mid \bigtriangledown u \mid)$.

$$\begin{aligned}\frac{\partial}{\partial t}u &= \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) \quad in\ D, t \geq 0\\ \left.\frac{\partial}{\partial \mathbf{n}}u\right|_{\partial d} &= 0\\ u(z,0) &= u_0(z), \qquad z\ \in\ D\end{aligned}$$

Now apply the equation(4.0.2.1) for this equation,

$$\bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown(c(\mid \bigtriangledown u \mid) \cdot \bigtriangledown u + c(\mid \bigtriangledown u \mid)\Delta u$$

$u_0(z)$ is given by the solution of this PDE,

$$\begin{aligned}\bigtriangledown^2 u(z) = 0 \quad u \in D\\ u = f \quad on \quad \partial D\end{aligned}$$

Now we are going to solve this non-linear diffusion PDE using equation(4.1.0.2).

Therefore,

$$u(z,t_{k+1}) = u(z,t_k) - \tau_k \bigtriangledown \cdot (c(\mid \bigtriangledown u(z,t_k) \mid) \bigtriangledown u(z,t_k)) \quad k = 0,1,2,3,...$$

$$when \quad k = 0 \quad ; \quad u(z,t_1) = u(z,t_0) - \tau_0 \bigtriangledown \cdot (c(\mid \bigtriangledown u(z,t_0) \mid) \bigtriangledown u(z,t_0))$$

$$when \quad k = 1 \quad ; \quad u(z,t_2) = u(z,t_1) - \tau_1 \bigtriangledown \cdot (c(\mid \bigtriangledown u(z,t_1) \mid) \bigtriangledown u(z,t_1))$$

$$when \quad k = 2 \quad ; \quad u(z,t_3) = u(z,t_2) - \tau_2 \bigtriangledown \cdot (c(\mid \bigtriangledown u(z,t_2) \mid) \bigtriangledown u(z,t_2))$$

....

We can continue this process and get a value of u.

4.2 Diffusion Conductivity

Diffusion conductivity can be a constant or non-constant. Here, we consider PDE with different type of diffusion conductivities.

1. Linear Conductivilty.[21]

$c(\mid \bigtriangledown u \mid) = c$, where c is a positive constant.

$$\frac{\partial}{\partial t} u = \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown \cdot (c \bigtriangledown u) = c \bigtriangledown \cdot \bigtriangledown u = c\Delta u$$

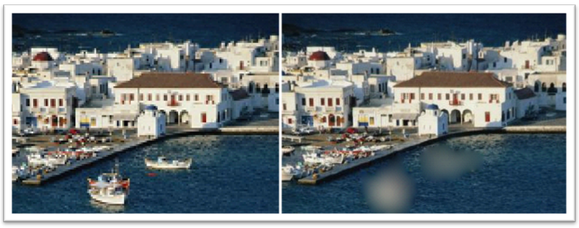


Figure 4.1: Isotropic Diffusion PDE Inpainting

2. Inverse proportional conductivity / TV Inpainting [21]

$c(\mid \bigtriangledown u \mid) = \frac{1}{\mid \bigtriangledown u \mid}$,

$$\frac{\partial}{\partial t} u = \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown \cdot \left(\frac{1}{\mid \bigtriangledown u \mid} \bigtriangledown u \right) = \bigtriangledown \cdot \left(\frac{\bigtriangledown u}{\mid \bigtriangledown u \mid} \right)$$

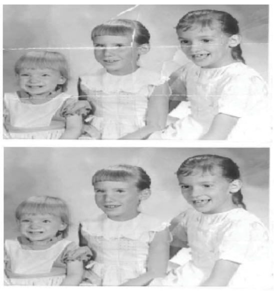


Figure 4.2: TV Inpainting method. Top: Orginal Image and Bottom : Inpainted Image

3. Curvature Driven Diffusion (CDD)Inpainting

$c(\mid \bigtriangledown u \mid) = \frac{\bigtriangledown u}{\mid \bigtriangledown u \mid}$,

$$\frac{\partial}{\partial t} u = \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown \cdot \left(\frac{\bigtriangledown u}{\mid \bigtriangledown u \mid} \bigtriangledown u \right)$$

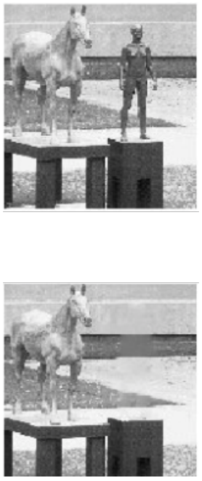


Figure 4.3: CDD Inpainting method. Top: Orginal Image and Bottom : Inpainted Image

4. Gaussian Conductivity.[21]

$c(\mid \bigtriangledown u \mid) = e^{-(\frac{|\bigtriangledown u|}{k})^2}$, where k is a constant.

$$\frac{\partial}{\partial t}u = \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown \cdot \left(e^{-(\frac{|\bigtriangledown u|}{k})^2} \bigtriangledown u \right)$$

5. Lorentz conductivity.[21]

$c(\mid \bigtriangledown u \mid) = \frac{1}{1+(\frac{|\bigtriangledown u|}{k})^2}$, where k is a constant.

$$\frac{\partial}{\partial t}u = \bigtriangledown \cdot (c(\mid \bigtriangledown u \mid) \bigtriangledown u) = \bigtriangledown \cdot \left(\frac{1}{1+(\frac{|\bigtriangledown u|}{k})^2} \bigtriangledown u \right)$$

Chapter 5

The Green Function of a Boundary Value problem

When we solve differential equation of the form *Lu=f*, with homogeneous boundary conditions, we use Green's function where *L* is a linear diffrentiable operator. This is a very important method to solve diffusion equation and wave equation.

Theorem : 5.1 [45]

Consider a continuous function f in [0, l]. Then u(z) satifies the *that satisfies*

Lu=f

Bu=0

also, is can be written by

$$u(z_0) = \int_0^l f(z)G(z_0, z)dz$$

where

G(z_0,z) is a Green's function and is satisfied by the following conditions:

a) G(z_0,z) is a continuous in z and z_0

b) G(z_0,z) = 0 whenever $z_0 \neq z$,

c) G(z_0,z) does satisfy the given boundary conditions at each end point

d)Symmentry: G(z_0,z) = G(z,z_0)

L is the Sturm-Liouville operator, it is a linear differential operator of the form.

$$L = \frac{d}{dz}\left[c(z)\frac{d}{dz}\right]$$

In [28], Tony F. Chan and J. Shen used the Green's second function for 2-D smooth image inpainting.

Consider the Green's second formula,

$$\int_D (u\Delta v - v\Delta u)dxdy = \int_{\partial D}\left(u\frac{\partial v}{\partial \mathbf{n}} - v\frac{\partial u}{\partial \mathbf{n}}\right)ds \tag{5.0.0.1}$$

where

a) The outward normal of ∂D, $\mathbf{n}$

b) The length parameter, s

c) u and v are any complex function.

d) Δ denote the Laplacian operator such that

$$\Delta u := \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2}$$

Consider z, z_0 $\in$ D. Where z_0 is a source point and z is a field point on D.

Consider u = f(z) and v = - G(z_0,z)

where

G(z_0,z) is the Green's function.

$$-\Delta G = \delta(z - z_0)$$
$$G\Big|_{\partial D} = 0$$

where f(z) is a image function defne on 2-D domain D.

$$\int_D (u\Delta v - v\Delta u)dxdy = \int_{\partial D}\left(u\frac{\partial v}{\partial \mathbf{n}} - v\frac{\partial u}{\partial \mathbf{n}}\right)ds$$
$$\int_D \Big(f(z)\Delta G(z_0, z) + G(z_0, z)\Delta f(z)\Big)dz = \int_{\partial D}\left(f(z)\frac{\partial G(z_0, z)}{\partial \mathbf{n}} + G(z_0, z)\frac{\partial f(z)}{\partial \mathbf{n}}\right)ds$$
$$-\int_D f(z)\delta(z_0 - z)dz + \int_D G(z_0, z)\Delta f(z)dz = \int_{\partial D} f(z)\frac{\partial G(z_0, z)}{\partial \mathbf{n}}ds + \int_{\partial D} G(z_0, z)\frac{\partial f(z)}{\partial \mathbf{n}}ds$$

But we know that $G\Big|_{\partial D} = 0$ and

$$\delta(z_0 - z) = \begin{cases} 1 & \text{if } z_0 = \text{z} \\ 0 & \text{if } z_0 \neq \text{z} \end{cases}$$

Therefore

$$-\int_D f(z_0)dz + \int_D G(z_0, z)\Delta f(z)dz = \int_{\partial D} f(z)\frac{\partial G(z_0, z)}{\partial \mathbf{n}}ds$$
$$\int_D f(z_0)dz = \int_D G(z_0, z)\Delta f(z)dz - \int_{\partial D} f(z)\frac{\partial G(z_0, z)}{\partial \mathbf{n}}ds$$
$$f(z_0) = \int_D G(z_0, z)\Delta f(z)dz + \int_{\partial D} f(z)\frac{-\partial G(z_0, z)}{\partial \mathbf{n}}ds$$

First term of this expression is called the *anti-harmonic inpainting*, also this is called *error term.* Second term is called the *Harmonic inpainting.*

Chapter 6

Mapping

6.1 Harmonic Function

A function $u(x, y)$ is said to be harmonic in a domain D
if the partial derivatives, $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$, $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$ exist and continue,
and if $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$ at all points of D.

6.1.1 Simply connected domain

A region D is said to be a simply connected domain when we consider any simple closed curve which is keep in D. For a two dimentional region, if there is no holes then we consider it is a simply connected domain.

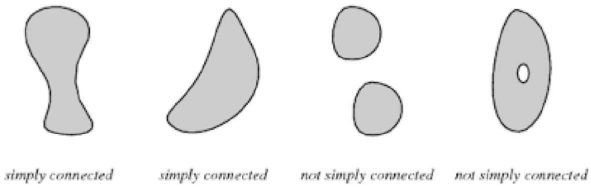


Figure 6.1: simply connected domain

6.1.2 Analytic Function

If the function f has a derivative at each point in some neighborhoods of z_0 then it is analytic at z_0.[3], [4], [5], [6]

6.2 Composition of Function

If there is a function $f : A \to B$ and $g : B \to C$ then there is a function from A to B. We can write it as $g \circ f : A \to C$. This is called composition of function.[3], [4], [5]

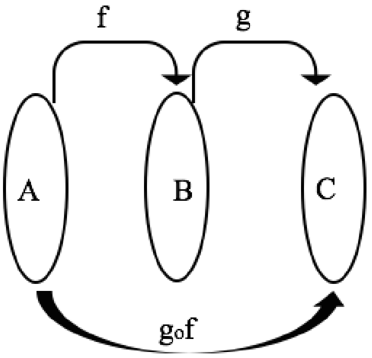


Figure 6.2: Compostion of Function, from A to C. $g \circ f : A \to C$

6.2.1 Chain Rule

Suppose that a function $f(z)$ is analytic on domain A and function g is analytic on domain A. We know that composition of two analytic functions is analytic. Therefore $g[f(z)]$ is analytic on domain A.[3], [4]

$$\frac{d}{dz} g[f(z)] = g'[f(z)]f'(z)$$

6.3 Conformal Mapping

If α_1 and α_2 are the angle of arc A_1 and A_2, which goes through z_0, and if β_1 and β_2 are the angles of arc B_1 and B_2, which goes through w_0 then,

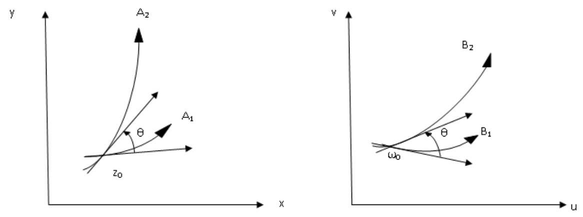


Figure 6.3: preserves angle

α_2 - α_1 = β_2 - β_1 = θ. The angel from B_1 to B_2 is the same as the angle from A_1 and A_2. This is called *preserves angle.*

A mapping that preserves an angle in that manner between every pair of curves at each point of some domain is said to be *conformal* in the domain.

Theorem:

At each point z of a domain where f is analytic and f'(z)≠0 the mapping w=f(z) is conformal.

Therefore, *conformal mapping* and *conformal transformation* will be used for transformation of analytic function.[3], [4], [5], [6]

Riemann Mapping Theorem

Suppose that D is a simply connected domain on the z-plane, which contains neither the point at ∞ *nor a given finite point a. Let* $z_0 \in$ *D. Then there exists a univalent analytic function w=f(z) which satisfies the following conditions:*[3], [4], [5], [6]

1) w= f(z) conformally maps D onto the unit disk $G \equiv \{|w| < 1\}$

2) $f(z_0) = 0, \acute{f}\ (z_0) > 0$

6.3.1 Möbius Transformation

The *linear fractional transformation*,

$$w = \frac{az+b}{cz+d} \quad , such \quad that \quad (ad - bc \neq 0) \tag{6.3.1.1}$$

a,b,c and d are complex constant [47].

Consider, the special case of transformation from upper half plane to unit circle.

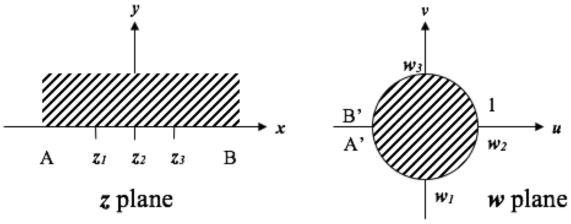


Figure 6.4: Transformation from upper half plane to the unit circle

Consider the mapping points,

$z_1 = -1$, $z_2 = 0$ and $z_3 = 1$

onto the points

$w_1 = -i$, $w_2 = 1$ and $w_3 = i$

Consider the, $z_2 = 0$ and $w_2 = 1$ and substitute them into the equation (6.3.1.1) then we can find,

$1 = \frac{b}{d}$ $\quad \therefore$ b=d.

$$w = \frac{az+b}{cz+b} \quad (b(a-c) \neq 0) \tag{6.3.1.2}$$

Consider the, $z_1 = -1$ and $w_1 = -i$,

$$\begin{aligned} -i &= \frac{-a+b}{-c+b} \\ ic - ib &= -a+b \end{aligned} \tag{6.3.1.3}$$

Now consider the, $z_3 = 1$ and $w_3 = i$,

$$\begin{aligned} i &= \frac{a+b}{c+b} \\ ic + ib &= a + b \end{aligned} \tag{6.3.1.4}$$

adding equations (6.3.1.3) and (6.3.1.4) are given by, c=-ib.

Substracting equations (6.3.1.3) and (6.3.1.4) are given by, a=ib.

Substitute both values into the equation (6.3.1.2)

$$\begin{aligned} \therefore w &= \frac{ibz + b}{-ibz + b} \\ &= \frac{b(iz+1)}{b(-iz+1)} \\ &= \frac{(iz+1)}{(-iz+1)} \\ &= \frac{(iz+1)}{(-iz+1)}\frac{i}{i} \\ &= \frac{i-z}{i+z} \end{aligned}$$

Therefore, transformation from upper halp plane to unit circle is,

$$w = \frac{i-z}{i+z} \tag{6.3.1.5}$$

Using equation (6.3.1.5) we can find transformation from unit circle to upper half plane.

$$\begin{aligned} w &= \frac{i-z}{i+z} \\ w(i+z) &= i - z \\ iw + zw &= i - z \\ zw + z &= i - iw \\ z(w+1) &= i - iw \\ z &= i\frac{1-w}{1+w} \end{aligned}$$

Therefore, transformation from unit circle to upper halp plane is,

$$z = i\frac{1-w}{1+w} \tag{6.3.1.6}$$

6.3.2 Joukowski Transformation

Transformation from any circle to ellipse is define by Joukowski Transformation.

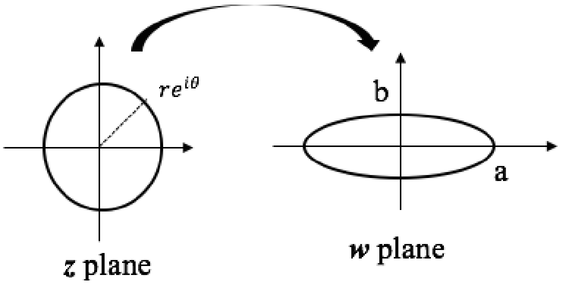


Figure 6.5: Transformation from any circle to ellipse

$$w = z + \frac{k^2}{z} \quad ,k \quad is \quad a \quad constant \tag{6.3.2.1}$$

$z = re^{i\theta}$ $w = \zeta + i\eta$ substitute them in to the equation (6.3.2.1).

$$\begin{aligned}\zeta + i\eta &= re^{i\theta} + \frac{k^2}{re^{i\theta}}\\ &= re^{i\theta} + k^2 re^{-i\theta}\\ &= r(cos\theta + isin\theta) + \frac{k^2}{r}(cos\theta - isin\theta)\\ &= \left(r + \frac{k^2}{r}\right)cos\theta + i\left(r - \frac{k^2}{r}\right)sin\theta\end{aligned}$$

On both sides we have comlex numbers. Then, their real parts and imaginary parts are equal separately.

Therfore,

$$\zeta = \left(r + \frac{k^2}{r}\right)cos\theta, \quad cos\theta = \frac{\zeta}{\left(r + \frac{k^2}{r}\right)}$$
$$\eta = \left(r - \frac{k^2}{r}\right)sin\theta, \quad sin\theta = \frac{\eta}{\left(r - \frac{k^2}{r}\right)} \tag{6.3.2.2}$$

Also, we know that,

$$cos^2\theta + sin^2\theta = 1$$

Substitute into the (6.3.2.2). Therfore,

$$\frac{\zeta^2}{\left(r + \frac{k^2}{r}\right)^2} + \frac{\eta^2}{\left(r - \frac{k^2}{r}\right)^2} = 1 \tag{6.3.2.3}$$

when, $r = b$, euqation (6.3.2.2) becomes,

$$\zeta = 2rcos\theta$$
$$\eta = 0$$

We have,

$z = x + iy$

$z = rcos\theta + irsin\theta \quad where, \quad x = rcos\theta \quad and \quad y = rsin\theta$

Therefor, $x = \frac{\zeta}{2}$.

Consider, $r + \frac{k^2}{r} = A$ and $r - \frac{k^2}{r} = B$ Therfore,

$$\frac{\zeta^2}{A^2} + \frac{\eta^2}{B^2} = 1$$

Now consider,

$$\begin{aligned} A^2 - B^2 &= \left(r + \frac{k^2}{r}\right)^2 - \left(r - \frac{k^2}{r}\right)^2 \\ &= r^2 + 2k^2 + \frac{k^4}{r^2} - r^2 + 2k^2 - \frac{k^4}{r^2} \\ &= 4k^2 \end{aligned}$$

We know that $A^2 - B^2 = C^2$ where C is a foci of the ellipse. Therefore, we can find, $k^2 = \frac{C^2}{4}$. We can rewrite the transformation from any circle to ellipse,

$$w = z + \frac{c^2}{4z}$$

6.4 Schwarz Christoffel Mapping

Schwarz Christoffel Mapping is the mapping for the real axis onto a polygon. This method was founded by $ElwinBrunoChristoffel$ and $HermannAmandusSchwarz$ [44]. Consider any points t_1, t_2, . . . ,t_n on the x-axis. Now we can map these points onto a polygon with points w_1, w_2, . . . ,w_n.

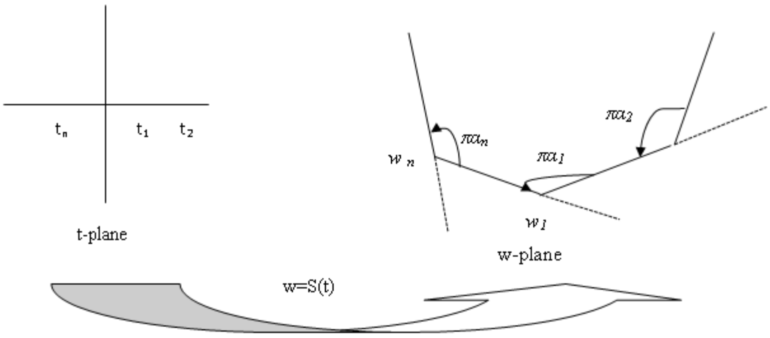


Figure 6.6: Transformation from upper half plane to any polygon

Theorem

Let P be the interior of a polygon Γ *having vertices* w_1,w_2, . . . ,w_n *and interior angles* $\alpha_1\ \pi$, $\alpha_2\ \pi$, $\alpha_3\ \pi$, . . . , $\alpha_n\ \pi$ *in counter clockwise* *order. Let f be any conformal map from the upper half plane H^+ to P. Then* [3], [4]

$$S'(t) = A(t-t_1)^{-\beta_1}(t-t_2)^{-\beta_2}...(t-t_n)^{-\beta_n} \tag{6.4.0.4}$$

We can write this as,

$$S(t) = A\int_{t_0}^{t}(t'-t_1)^{-\beta_1}(t'-t_2)^{-\beta_2}...(t'-t_n)^{-\beta_n}dt' + B \tag{6.4.0.5}$$

where $\beta_k\pi$ is the exterior angle of polygon such that,

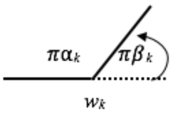


Figure 6.7: Sum of the angles on the straight line is eqaul to the π

Therfore,

$$\begin{aligned} \pi\alpha_k + \pi\beta_k &= \pi, \quad k = 1,2,3,...,n \\ \alpha_k + \beta_k &= 1 \\ \beta_k &= 1-\alpha_k \end{aligned}$$

Also we know that any closed polygon sum of the exterior angle of is equal to 2π.

$$\begin{aligned} \pi\beta_1 + \pi\beta_2 + ... + \pi\beta_n &= 2\pi \\ \beta_1 + \beta_2 + ... + \beta_n &= 2 \\ \sum_{k=1}^{n}\beta_k &= 2 \end{aligned}$$

use the previouse equations answer here. Then, we have

$$\sum_{k=1}^{n} 1 - \alpha_k = 2 \quad where \quad -1 < \alpha_k < 1 \tag{6.4.0.6}$$

6.4.1 Schwarz Christoffel Mapping for a Triangle

[3]

Consider the transformation from upper half plane to any triangle.

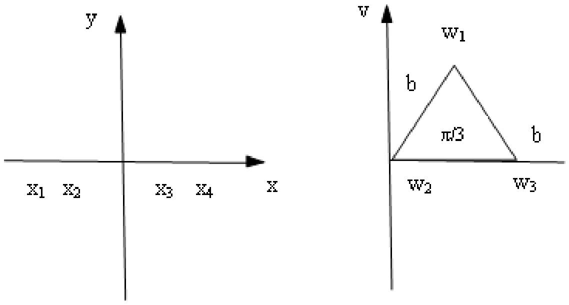


Figure 6.8: Schwarz Christoffel Mapping form upper Half plane to Triangle

Using Schwarz Christoffel Mapping equation (6.4.0.5), we can can write,

$$\omega = A\int_{t_0}^{t} (t' - x_1)^{-\beta_1}(t' - x_2)^{-\beta_2}(t' - x_3)^{-\beta_3}dt' + B$$

Consider the outside angles are $\beta_1\pi$, $\beta_2\pi$ and $\beta_3\pi$.

$$\beta_1\pi + \beta_2\pi + \beta_3\pi = 2\pi$$

$$\beta_1 + \beta_2 + \beta_3 = 2$$

We can consider the w_3 is an infinite point. Then we can rewrite the transformation,

$$\omega = A\int_{t_0}^{t} (t' - x_1)^{-\beta_1}(t' - x_2)^{-\beta_2}dt' + B$$

When we consider the equilateral triangle,

$\beta_1\pi = \beta_2\pi = \beta_3\pi = \frac{2\pi}{3}$

$\beta_1 = \beta_2 = \beta_3 = \frac{2}{3}$

Therefore we can write,

$$\omega = A\int_{t_0}^{t}(t' - x_1)^{-\frac{2}{3}}(t' - x_2)^{\frac{2}{3}}dt' + B$$

Substitute $x_1 = -1$ and $x_2 = 1$. Also $t_0 = 1$ when A=1 and B=0.

$$\omega = \int_{1}^{t}(t' + 1)^{-\frac{2}{3}}(t' - 1)^{\frac{2}{3}}dt' \tag{6.4.1.1}$$

When we are finding the position of the w_1, w_2 and w_3

Case 1: t=1;

Now we have $\omega = 0$, therefore $\omega_2 = 0$.

Case 2: t=-1;

Now our integral limit t=-1,

We know that when the points of x-axis map to points of polygon, it satisfies the ,

$$(t - t_k)^{-\beta_k} = |(t - t_k)|^{-\beta_k}e^{-i\beta_k\gamma_k\pi}$$

where $\gamma_k = arg(t - t_k)$

consider the point, x_1 = -1, then arg(t+1) = 0 and x_2 = 1, then arg(t-1) = π. Therfore,

$$\begin{aligned}(t + 1)^{-\frac{2}{3}} &= |(t + 1)|^{-\frac{2}{3}}e^{-i(\frac{2}{3})(0)}\\ &= |(t + 1)|^{-\frac{2}{3}}\end{aligned}$$

$$\begin{aligned}(t-1)^{-\frac{2}{3}} &= |(t-1)|^{-\frac{2}{3}}e^{-i(\frac{2}{3})(\pi)}\\ &= |(t-1)|^{-\frac{2}{3}}e^{-\frac{2}{3}\pi i}\end{aligned}$$

Now rewrite the equation (6.4.1.1) with these information,

$$\begin{aligned}\omega_1 &= \int_1^{-1} |(t+1)|^{-\frac{2}{3}}|(t-1)|^{-\frac{2}{3}}e^{-\frac{2}{3}\pi i}dt\\ &= \int_1^{-1}(1+t)^{-\frac{2}{3}}(1-t)^{-\frac{2}{3}}e^{-\frac{2}{3}\pi i}dt\\ &= e^{-\frac{2\pi i}{3}}\int_{-1}^{1}\frac{dt}{(1-t^2)^{2/3}}\\ &= 2e^{\frac{1\pi i}{3}}\int_0^1\frac{dt}{(1-t^2)^{2/3}}\end{aligned}$$

Subsitute $y=t^2$

$1-t^2$=1-y, if t=0 then y=0 and if t=-1then y=1.

dy=2t dt, dt=$\frac{dy}{2t}$=$\frac{dy}{2y^{\frac{1}{2}}}$

$$\begin{aligned}w_1 &= 2e^{\frac{1\pi i}{3}}\int_0^1\frac{\frac{dy}{2y^{\frac{1}{2}}}}{(1-y)^{2/3}}\\ &= e^{\frac{1\pi i}{3}}\int_0^1\frac{dy}{(1-y)^{2/3}y^{\frac{1}{2}}}\end{aligned}$$

When we solve this equation, we can use ***Beta function***,

$$B(g,h)=\int_0^1 y^{g-1}(1-y)^{h-1}dy$$

Consider, $g=\frac{1}{2}$ and $h=\frac{1}{3}$ then,

$$\begin{aligned}B\left(\frac{1}{2},\frac{1}{3}\right) &= \int_0^1 t^{\frac{1}{2}-1}(1-t)^{\frac{1}{3}-1}dy\\ &= \int_0^1 t^{-\frac{1}{2}}(1-t)^{-\frac{2}{3}}dt\end{aligned}$$

Consider this $B(\frac{1}{2},\frac{1}{3})$ = a, then we have,

$$w_1 = ae^{\frac{1\pi i}{9}}$$

Case 3: t=infinity

Case 3a: t=∞ limit t=∞, if w_3 is in positive infinity then the equation (6.4.1.1) can write as,

$$\begin{aligned} w_3 &= \int_1^\infty |(t+1)|^{\frac{-2}{3}} |(t-1)|^{\frac{-2}{3}} dx \\ &= \int_1^\infty (t+1)^{\frac{-2}{3}} (t-1)^{\frac{-2}{3}} dx \\ &= \int_1^\infty \frac{dx}{(x^2-1)^{2/3}} \end{aligned}$$

Case 3b: t=$-\infty$ if w_3 is in negative infinity then the equation (6.4.1.1) limit t=$-\infty$. Now we can write that equation,

$$\omega_3 = \int_1^{-1} |(t+1)|^{-\frac{2}{3}} e^{-\frac{2}{3}\pi i} |(t-1)|^{-\frac{2}{3}} dt + \int_{-1}^{-\infty} |(t+1)|^{-\frac{2}{3}} e^{-\frac{4}{3}\pi i} |(t-1)|^{-\frac{2}{3}} dt$$

We can solve this similarly as w_1 and also first part of this integral is equal to exactly the w_1. So we can rewite our equation as,

$$\begin{aligned} \omega_3 &= w_1 + e^{-\frac{4}{3}\pi i} \int_{-1}^{-\infty} (t+1)^{-\frac{2}{3}} (t-1)^{-\frac{2}{3}} dt \\ &= w_1 + e^{-\frac{1}{3}\pi i} \int_{-1}^{-\infty} (t^2-1)^{-\frac{2}{3}} dt \end{aligned}$$

second integral is an even function, so we can change the integral limits.

$$w_3 = w_1 + e^{-\frac{1}{3}\pi i} \int_1^\infty (t^2-1)^{-\frac{2}{3}} dt$$

Use the answer of case 1 value of w_3,

$$w_3 = ae^{\frac{\pi i}{3}} + e^{-\frac{1}{3}\pi i} w_3$$

Now solve for w_3, consider

$$e^{i\theta} = cos(\theta) + isin(\theta)$$

$$\begin{aligned} w_3 &= a\left(cos\frac{\pi}{3} + isin\frac{\pi}{3}\right) + w_3\left(cos\frac{-\pi}{3} + isin\frac{-\pi}{3}\right) \\ &= a\left(\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) + w_3\left(\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \\ w_3 &= a \end{aligned}$$

6.5 Application of Conformal Mapping

The Dirichlet and Neumann problems can be solved for any simply connected domain D, which can be mapped conformally by an analytic function on to the interior of a unit circle or half plane. Schwarz Christoffel Mapping is used for magnetic motors, crack detection, microwave waveguides, etc [2].

Chapter 7

Multi Resolution Approximation for Image Inpainting

In [20] this paper, Charles K. Chui introduced a new method for image inpainting.
Let Ω be the simply connected domain in $\mathbb{R}^2$. Here, we consider inpainting domain D be a simply connected domain of Ω.

His method is called MRA.

$$u_n = u_0 + w_1 + ... + w_n \quad in \quad D \tag{7.0.0.1}$$

where $u_0 = P_0(F|_{\partial D})$ and $w = w_i$ is a solution of the PDE

$$\left.\begin{array}{l} L_0 w = (T_1 ... T_{i-1} P_i)(d_i(F)) \quad in\ D \\ w|_{\partial D} = 0 \end{array}\right\}$$

$$w_i = E_i(d_i(F)) \qquad i = 1, ..., n \tag{7.0.0.2}$$

$$E_i = T_0 T_1 ... T_{i-1} P_i, \qquad i = 1, ..., n \tag{7.0.0.3}$$

$$d_i(F) := L_{i-1} ... L_0 F \quad on \quad \partial D \tag{7.0.0.4}$$

Where L is the lagged diffucivity operator.

$$(L_i f)(z_1) = \bigtriangledown \cdot (c_{i-1}(z_1) \bigtriangledown f(z_1)), \quad z_1 \in D \tag{7.0.0.5}$$

Where $\quad c_i(z_1) = c(| \bigtriangledown u^i(z_1)|), \quad z_1 \in D$

These are called the data propoagation operators,

$$(T_i f)(z_0) = \int_D f(z) G_i(z_0, z) dz \tag{7.0.0.6}$$

$$(P_i v)(z_0) = \int_{\partial D} v(z(s)) g_i(z_0, z(s)) ds \tag{7.0.0.7}$$

$$g_i(z_0, z) = -c_{i-1}(z) \left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}} \right)(z) \tag{7.0.0.8}$$

$$\frac{\partial h(z)}{\partial \mathbf{n}} = (\bigtriangledown h(z)) \cdot \mathbf{n}, \quad since \quad h|_{\partial D} = 0 \tag{7.0.0.9}$$

Lemma 7.1[20] For a smooth function f,h,$c_{i-1} \in D$

$$\int_D f(z)(L_i h)(z) dz = \int_D h(z)(L_i f)(z) dz - \int_{\partial D} c_{-1}(z) f(z) \left(\frac{\partial h(z)}{\partial \mathbf{n}} \right)(z) ds$$

Theorem 7.1[20]: Let F be a sufficiently smooth function in Ω with missing section $F_D := F|_D$, and let w_i be the i^{th} level details with boundary data $d_i(F)$. Then the error of the recovery portion of F_D by u_n is given by

$$F_D(z_0) - u_n(z_0) = (T_0 T_1 ... T_{n-1})(L_n L_{n-1} ... L_0 F_D)(z_0) \quad z_1 \in D \tag{7.0.0.10}$$

Proof:

Consider,

$$\left.\begin{aligned} L_i G_i(z_0, z) &= \delta(z_0, z), \quad z_0, z \in D \\ G_i(z_0, z)|_{z \in \partial D} &= 0 \end{aligned}\right\}$$

$$\delta(z_0 - z) = \begin{cases} 1 & \text{if } z_0 = \text{z} \\ 0 & \text{if } z_0 \neq \text{z} \end{cases}$$

consider,

$$\int_D F_D(z) L_0 G_0(z_0, z) dz = \int_D F_D(z) \delta(z_0, z) dz$$

$$if \quad z = z_0 \quad then \quad \delta(z_0, z) = 1$$

$$= \int_D F_D(z_0) dz$$

$$= F_D(z_0)$$

Therefore

$$F_D(z_0) = \int_D F_D(z) L_0 G_0(z_0, z) dz \tag{7.0.0.11}$$

Use the Lemma 7.1 such that $h(z_0) = G_0(z_0, z)$ and $z_0 \in D$. Now we have

$$\int_D F_D(z) L_0 G_0(z_0, z) dz = \int_D G_0(z_0, z)(L_0 F_D(z)) dz - \int_{\partial D} c_{-1}(z) F(z) \left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}} \right)(z) ds$$

which replaced the left hand side of the equation using the value of the equation (7.0.0.11)

$$F_D(z_0) = \int_D G_0(z_0, z)(L_0 F_D(z))dz - \int_{\partial D} c_{-1}(z)F(z)\left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}}\right)(z)ds$$

$$= \int_D (L_0 F_D(z))G_0(z_0, z)dz + \int_{\partial D} F(z)(-c_{-1}(z)\left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}}\right)(z))ds \tag{7.0.0.12}$$

use the equation (7.0.0.6)

$$T_0(L_0 F_D))(z_0) = \int_D (L_0 F_D(z))G_0(z_0, z)dz \tag{7.0.0.13}$$

Also consider the equation (7.0.0.8)

$$g_0(z_0, z) = -c_{-1}(z)\left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}}\right)(z)$$

Therefore,

$$\int_{\partial D} F(z)(-c_{-1}(z)\left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}}\right)(z))ds = \int_{\partial D} F(z)(g_0(z_0, z))ds$$

Also consider definition of

$$(P_0 F)(z_0) = \int_{\partial D} F(z)(g_0(z_0, z))ds$$

Therfore,

$$(P_0 F)(z_0) = \int_{\partial D} F(z)(-c_{-1}(z)\left(\frac{\partial G_i(z_0, .)}{\partial \mathbf{n}}\right)(z))ds \tag{7.0.0.14}$$

Substitute equation (7.0.0.13) and (7.0.0.14) into the equation (7.0.0.12)

$$F_D(z_0) = (T_0(L_0F_D))(z_0) + (P_0F)(z_0)$$

use the definition of u_0=$(P_0F)(z_0)$

$$F_D(z_0) = (T_0(L_0F_D))(z_0) + u_0(z_0)$$

$$F_D(z_0) - u_0(z_0) = (T_0(L_0F_D))(z_0) \tag{7.0.0.15}$$

Therfore, when i=0, we prove equation (7.0.0.10)

For 1 $\leq$ i $\leq$ n, we can do this calculation the same as we did earlier . We can prove the error formula.

7.1 Process of MRA

$u_1 = u_0 + w_1$

$u_2 = u_0 + w_1 + w_2$

$u_3 = u_0 + w_1 + w_2 + w_3$

.

$u_n = u_0 + w_1 + ... + w_n$

by the definition of w_i in the equation (7.0.0.2)

$w_1 = E_1(d_1(F))$

$w_2 = E_2(d_2(F))$

$w_3 = E_3(d_3(F))$

.

$w_n = E_n(d_n(F))$

by the definition of $d_i(F)$ in the equation (7.0.0.4)

$w_1 = E_1L_0F$

$w_2 = E_2L_1L_0F$

$w_3 = E_3L_2L_1L_0F$

.

$w_n = E_nL_{n-1}...L_0F$

by the definition of E_i in the equation (7.0.0.3)

$E_1 = T_0P_1$

$E_2 = T_0T_1P_2$

$E_3 = T_0T_1T_2P_3$

.

$E_n = T_0T_1...T_{n-1}P_n$

apply this is for w_i.

by the definition of MRA details extension

$$w_1 = T_0P_1L_0F$$

$$w_2 = T_0T_1P_2L_1L_0F$$

$$w_3 = T_0T_1T_2P_3L_2L_1L_0F$$

$$.$$

$$w_n = T_0T_1...T_{n-1}P_nL_{n-1}...L_0F$$

We consider special case such that

1.$c_{i-1}(z_0) = 1$ then

$$(L_if)(z_0) = \Delta f(z_0), \quad z_0 \in D$$

2. $G_i(z_0, z) = G(z_0, z)$ then

$$(T_if)(z_0) = (Tf)(z_0) = \int_D f(z)G(z_0, z)dz \quad and$$

$$g_i(z_0, z) = g(z_0, z) = -\left(\frac{\partial G(z_0, .)}{\partial \mathbf{n}}\right)(z) \quad and$$

$$(P_i v)(z_0) = (Pv)(z_0) = \int_{\partial D} v(z(s))g(z_0, z(s))ds$$

3. $G(z_0, z) = G(z, z_0)$

Therefore,

$$\begin{aligned}
w_1 &= TP\Delta F & &= TP\Delta F \\
w_2 &= TTP\Delta\Delta F & &= TTP\Delta^2 F \\
w_3 &= TTTP\Delta\Delta\Delta F & &= TTTP\Delta^3 F \\
&. & & \\
w_n &= TT...TP\Delta...\Delta F & &= TT...TP\Delta^n F
\end{aligned}$$

apply equation (6) and (7) for these

$$\begin{aligned}
w_1 &= TP\Delta F(z) \\
&= \int_D P\Delta F(z_1)G(z_0, z)dz \\
&= \int_D \int_{\partial D} \Delta F(z(s))g(z_0, z(s))dsG(z_0, z)dz_0
\end{aligned}$$

$$\begin{aligned}
w_2 &= TTP\Delta^2 F(z) \\
&= \int_D TP\Delta^2 F(z_0)G(z_0, z)dz_0 \\
&= \int_D \int_D P\Delta^2 F(z)G(z_0, z)dzG(z_0, z)dz_0 \\
&= \int_D \int_D \int_{\partial D} \Delta^2 F(z_0(s))g(z_0, z)dsG(z_0, z)dzG(z_0, z)dz_0
\end{aligned}$$

$$\begin{aligned}
w_3 &= TTTP\Delta^3 F(z) \\
&= \int_D TTP\Delta^3 F(z_0)G(z_0, z)dz \\
&= \int_D \int_D TP\Delta^3 F(z)G(z_0, z)dzG(z_0, z)dz_0 \\
&= \int_D \int_D \int_D P\Delta^3 F(z_0)G(z_0, z)dz_0 G(z_0, z)dzG(z_0, z)dz_0 \\
&= \int_D \int_D \int_D \int_{\partial D} \Delta^3 F(z(s))g(z_0, z(s))dsG(z_0, z)dz_0 G(z_0, z)dzG(z_0, z)dz_0
\end{aligned}$$

and continue this process.

Chapter 8

Mathematical Approaches

8.1 Image Inpainting Methods

8.1.1 Initial Value

When we find the initial value of the inpainting domain, we use a five-point stencil method. So the inpainting domain has m rows and n colums. Then our image looks like this.

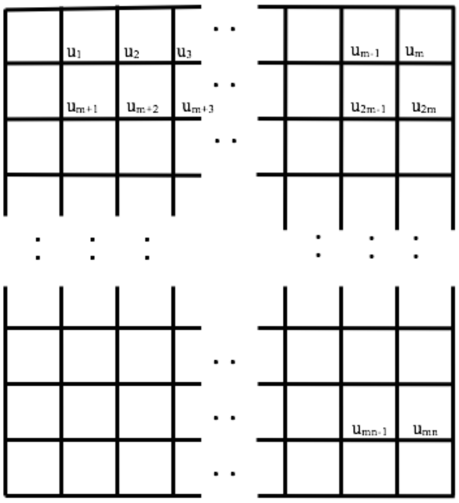


Figure 8.1: General 2D Grid with n rows and m column

Now consider the general matrix for A for the size of inpainting domain.

$$A = \begin{bmatrix} B & I & O & ... & O \\ I & B & I & ... & O \\ O & I & B & ... & O \\ O & ... & ... & ... & O \\ & ... & ... & ... & ... \\ & ... & ... & ... & ... \\ & ... & I & B & I \\ & ... & ... & ...B & I \end{bmatrix}$$

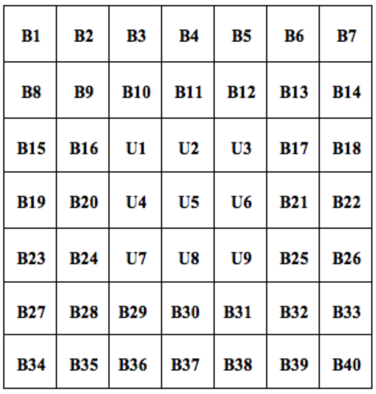
where A is a matrix with nm × nm. It has n number of block martix in each row and column.

$$B = \begin{pmatrix} -4 & 1 & & & & \\ 1 & -4 & 1 & & & \\ 0 & 1 & -4 & 1 & & \\ & & & \ddots & & \\ & & & 1 & -4 & 1 \\ & & & & 1 & -4 \end{pmatrix}$$

B is a m × m matrix. Where m is a number of columns in inside of the grid.

$$I = \begin{pmatrix} 1 & 0 & & & \\ 0 & 1 & 0 & & \\ & & \ddots & & \\ & & 0 & 1 & 0 \\ & & & 0 & 1 \end{pmatrix} \quad O = \begin{pmatrix} 0 & 0 & & & \\ 0 & 0 & 0 & & \\ & & \ddots & & \\ & & 0 & 0 & 0 \\ & & & 0 & 0 \end{pmatrix}$$

When we apply this method to our image, we did some changes. I copleted image inpainting technique level by level. That is,

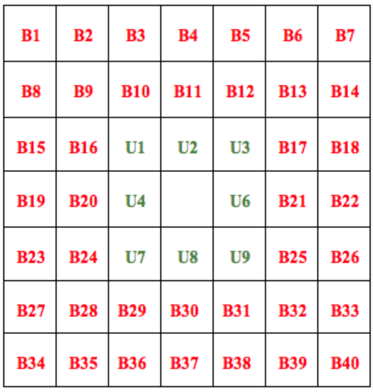


B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4	U5	U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 8.2: 5-Point apply Level by level

We have the boundary information and we have to find $u_1, \ldots,$ u_9. Normally in a 5-point method we need only one adjacent bounday level data. But in this method, we need two adjacent bondary level data to fill inside date.

Step 1:



B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4		U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 8.3: 5-Point apply to Level 1

Using the boundary data (**red data**) we can find adjacent level data (**green data**). Here we apply the 5-point stencil method in different way.

For U_2

$$U_2 = 4B_{11} - B_4 - B_{10} - B_{12}$$

For U_4

$$U_4 = 4B_{20} - B_{16} - B_{19} - B_{24}$$

For U_6

$$U_6 = 4B_{21} - B_{17} - B_{22} - B_{25}$$

For U_8

$$U_8 = 4B_{30} - B_{29} - B_{31} - B_{37}$$

Step 2:

We write two different formulas for corner points and consider the average value. Such as,

For U_1

$$U_1^1 = 4B_{16} - B_9 - B_{15} - B_{20}$$
$$U_1^2 = 4B_{10} - B_3 - B_9 - B_{11}$$
$$U_1 = \frac{U_1^1 + U_1^2}{2}$$

For U_3

$$U_3^1 = 4B_{12} - B_5 - B_{11} - B_{13}$$
$$U_3^2 = 4B_{17} - B_{13} - B_{18} - B_{21}$$
$$U_3 = \frac{U_3^1 + U_3^2}{2}$$

For U_7

$$U_7^1 = 4B_{24} - B_{20} - B_{23} - B_{28}$$
$$U_7^2 = 4B_{29} - B_{28} - B_{30} - B_{36}$$
$$U_7 = \frac{U_7^1 + U_7^2}{2}$$

For U_9

$$U_9^1 = 4B_{25} - B_{21} - B_{26} - B_{32}$$

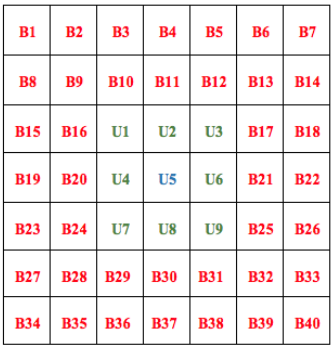
$$U_9^2 = 4B_{31} - B_{30} - B_{32} - B_{38}$$

$$U_9 = \frac{U_9^1 + U_9^2}{2}$$

Step 3:

When we are finding the next level of data (**blue data**), we use the adjacent level data (**green data**)

$$U_5 = \frac{U_2 + U_4 + U_6 + U_8}{4}$$



B1	B2	B3	B4	B5	B6	B7
B8	B9	B10	B11	B12	B13	B14
B15	B16	U1	U2	U3	B17	B18
B19	B20	U4	U5	U6	B21	B22
B23	B24	U7	U8	U9	B25	B26
B27	B28	B29	B30	B31	B32	B33
B34	B35	B36	B37	B38	B39	B40

Figure 8.4: 5-Point apply to Level 2

Number of iterations

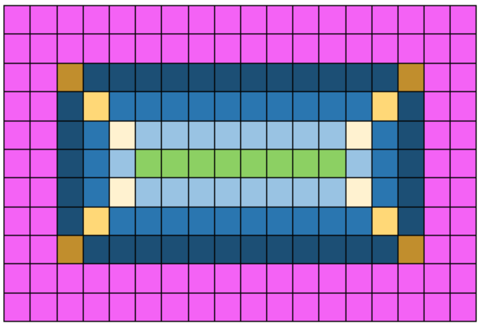


Figure 8.5: Apply 5-point method level by level. Purple is the Boundary data.

$$niter = \left\lfloor \frac{min(nrows, mcolms)}{2} \right\rfloor$$

if *niter* is odd and nrows >mcolms

$$niter1 = niter$$

$$niter2 = niter + 1$$

$$niter3 = niter + 1$$

if *niter* is odd and nrows<mcolms

$$niter1 = niter + 1$$

$$niter2 = niter$$

$$niter3 = niter + 1$$

if *niter* is even and nrows>mcolms

$$niter1 = niter - 1$$

$$niter2 = niter$$

$$niter3 = niter$$

if *niter* is even and nrows<mcolms

$$niter1 = niter$$

$$niter2 = niter - 1$$

$$niter3 = niter$$

Where ,

niter : Number of iterartions

nrows :Number of rows in the domain

mcolms : Number of colums in the domain

niter1: Number of iterartions from left side or right side of the domain

niter2 : Number of iterartions from top side or bottom side of the domain

niter3 : Number of iterartions from corners of the domain

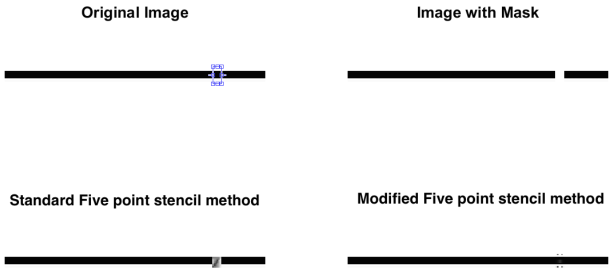

Original Image
Image with Mask
Standard Five point stencil method
Modified Five point stencil method

Figure 8.6: Standard 5-Point Stencil method and Modified 5-point Stencil

Inpainting Methods	Standard 5-point Stencil	Modified 5-point Stencil
PSNR	29.1787	39.2958

Table 8.1: PSNR value for Standard 5-Point Stencil method and Modified 5-point Stencil

When we compare the inpainted image with the original image, we use PSNR values. This is Peak Signal Noise Ratio. We define PSNR using mean squared error (MSE) and formula is given by, [46]

$$PSNR = 20 \cdot log_{10} \frac{MAX}{\sqrt{MSE}}$$

Where MAX is the maximum posible pixel value of the image.

When we check the two different approach of 5-point stancil methods. We can see modified method PSNR value is larger than the standard 5-point stencil method. So we use the modified the 5-point stancil method values for our further calculations.

8.1.2 Non-Linear Diffusion PDE and Iterative Linear Diffusion PDE

In 2009, [20] C. K. Chui used a partial differential equation of anisotropic diffusion to known data .

$$\left.\begin{aligned} \frac{\partial}{\partial t}u_j &= \bigtriangledown \cdot (c(\mid \bigtriangledown u_{j-1} \mid) \bigtriangledown u_j) \quad in\ D, t \geq 0 \\ \frac{\partial}{\partial \mathbf{n}}u_j\Big|_{\partial D} &= 0 \\ u_j(z,0) &= u_0(z), \qquad z\ \in\ D \end{aligned}\right\}$$

where j=1, 2, ... and $c(\mid \bigtriangledown u_j(z) \mid$ is the diffusion conductivity. Here we have a set of linear partial differential equations. This is also local image inpainting method.

Now we are going to solve this linear equation.

when j=1;

$$\begin{aligned} \frac{\partial}{\partial t}u_1 &= \bigtriangledown \cdot (c(\mid \bigtriangledown u_0 \mid) \bigtriangledown u_1) \quad in\ D, t \geq 0 \\ \frac{\partial}{\partial \mathbf{n}}u_1\Big|_{\partial d} &= 0 \\ u_1(z,0) &= u_0(z), \qquad z\ \in\ D \end{aligned}$$

Now apply the equation (6.3.2.1) for this PDE,

$$u_1(z,t_{k+1}) = u_1(z,t_k) + \tau_k f(u_1(z,t_k)) \quad k = 0,1,2,3,...$$

Now $f(u_1(z,t_k)) = \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_k))$

Therefore,

$$u_1(z,t_{k+1}) = u_1(z,t_k) + \tau_k \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_k)) \quad k = 0,1,2,3,...$$

$$when \quad k=0 \quad ; \quad u_1(z,t_1) = u_1(z,t_0) + \tau_0 \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_0))$$

$$when \quad k=1 \quad ; \quad u_1(z,t_2) = u_1(z,t_1) + \tau_1 \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_1))$$

$$when \quad k=2 \quad ; \quad u_1(z,t_3) = u_1(z,t_2) + \tau_2 \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_2))$$

$$.$$

$$.$$

$$.$$

$$when \quad k=N-1 \quad ; \quad u_1(z,t_N) = u_1(z,t_{N-1}) + \tau_{N-1} \bigtriangledown \cdot (c(\mid \bigtriangledown u_0(z) \mid) \bigtriangledown u_1(z,t_{N-1}))$$

We can get a value of $u_1(z,t_N)$ and use this value to find u_2.

when j=2;

$$\frac{\partial}{\partial t} u_2 = \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2) = \quad in\ D, t \geq 0$$

$$\frac{\partial}{\partial \mathbf{n}} u_2 \Big|_{\partial d} = 0$$

$$u_2(z,0) = u_0(z), \qquad z \ \in \ D$$

Now consider $f(u_2) = \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2)$

Therefore,

$$u_2(z,t_{k+1}) = u_2(z,t_k) + \tau_k \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2(z,t_k)) \quad k=0,1,2,3,...$$

$$when \quad k=0 \quad ; \quad u_2(z,t_1) = u_2(z,t_0) + \tau_0 \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2(z,t_0))$$

$$when \quad k=1 \quad ; \quad u_2(z,t_2) = u_2(z,t_1) + \tau_1 \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2(z,t_1))$$

$$when \quad k=2 \quad ; \quad u_2(z,t_3) = u_2(z,t_2) + \tau_2 \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2(z,t_2))$$

$$\vdots$$

$when \quad k = N-1 \quad ; \quad u_2(z,t_N) = u_2(z,t_{N-1}) + \tau_{N-1} \bigtriangledown \cdot (c(\mid \bigtriangledown u_1(z,t_N) \mid) \bigtriangledown u_2(z,t_{N-1}))$

We can get a value of $u_2(z,t_N)$ and use this value to find u_3.

When j=3;

$$\frac{\partial}{\partial t} u_3 = \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3) = \quad in\ D, t \geq 0$$

$$\frac{\partial}{\partial \mathbf{n}} u_3 \Big|_{\partial d} = 0$$

$$u_3(z,0) = u_0(z), \qquad z \in D$$

Now consider $f(u_3) = \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3)$

Therefore,

$u_3(z,t_{k+1}) = u_3(z,t_k) + \tau_k \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3(z,t_k)) \quad k = 0,1,2,3,...$

$when \quad k = 0 \quad ; \quad u_3(z,t_1) = u_3(z,t_0) + \tau_0 \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3(z,t_0))$

$when \quad k = 1 \quad ; \quad u_3(z,t_2) = u_3(z,t_1) + \tau_1 \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3(z,t_1))$

$when \quad k = 2 \quad ; \quad u_3(z,t_3) = u_3(z,t_2) + \tau_2 \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3(z,t_2))$

$$\vdots$$

$when \quad k = N-1 \quad ; \quad u_3(z,t_N) = u_3(z,t_{N-1}) + \tau_{N-1} \bigtriangledown \cdot (c(\mid \bigtriangledown u_2(z,t_N) \mid) \bigtriangledown u_3(z,t_{N-1}))$

We can get a value of $u_3(z,t_N)$.

Using this method, we can solve iterative linear PDE.

Initial value of the non-Linear and iterative linear PDE method is the 5-point stencil method values. Using a MATLAB program we inpainted the damaged image. Here we comapre MATLAB outputs different inpainting methods with different diffusion conductivity.

Case 1: Iterative Linear and Non-Linear Image Inpainting PDE with Constant Diffusion Conductivity

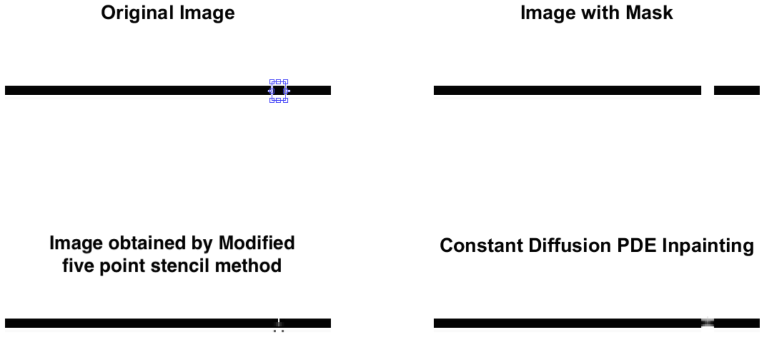

Original Image
Image with Mask
Image obtained by Modified five point stencil method
Constant Diffusion PDE Inpainting

Figure 8.7: Image Inpainting methods with Constant Diffusion

Inpainting Methods	5-point Stencil	Constant conductivity PDE 3
PSNR	35.5747	29.9493

Table 8.2: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Constant Diffusion Conductivity

Here, we consider the constant conductivity. That is,

c(p)= c. When c is a constant, there is a no difference between linear and non-linear PDE.

That is, Linear PDE,

$$\frac{\partial}{\partial t}u_i = \bigtriangledown \cdot (c) \bigtriangledown u_i) = c\Delta u_i$$

Non-Linear PDE,

$$\frac{\partial}{\partial t}u = \bigtriangledown \cdot (c) \bigtriangledown u) = c\Delta u$$

When c is a constant, we have worst inpainted image. Therefore, the 5-point stencil method is better than the constant diffusion conductivity PDE.

Case 2: Iterative Linear and Non-Linear Image Inpainting PDE with Inverse Proportional Diffusion Conductivity

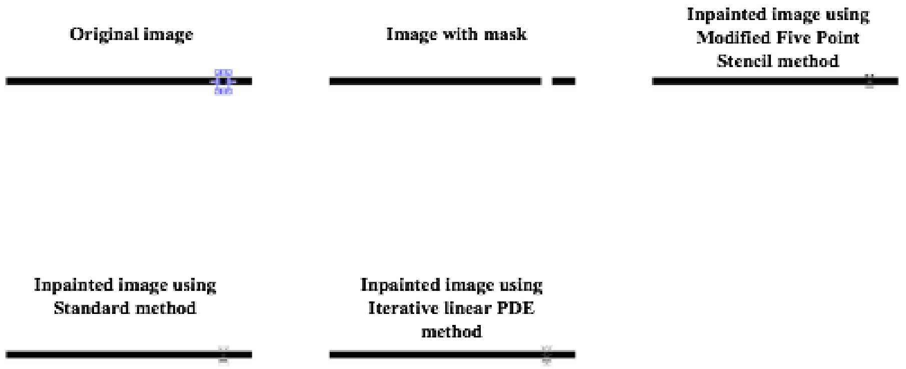

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard method
Inpainted image using Iterative linear PDE method

Figure 8.8: Image Inpainting methods with inverse proportional diffusion conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	30.3192	32.8715	32.9496

Table 8.3: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Inverse Proportional Diffusion Conductivity

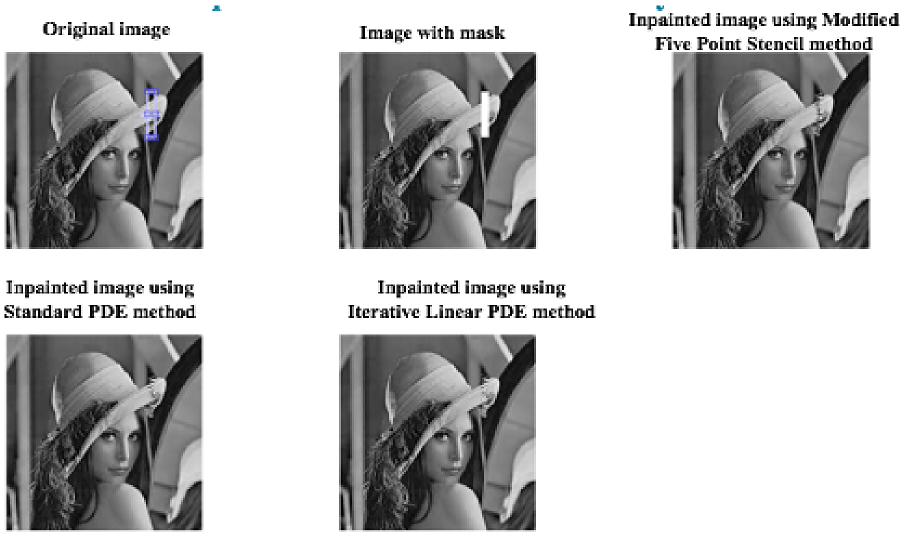

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard PDE method
Inpainted image using Iterative Linear PDE method

Figure 8.9: Image Inpainting methods with inverse proportional diffusion conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	27.7033	29.1043	29.5152

Table 8.4: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Inverse Proportional Diffusion Conductivity

Here, we consider the inverse proportional conductivity. That is,

c(p)= $\frac{1}{p}$.

With this diffusion conductivity,diffusion PDE is called TV inpainting method.

When we use this in MATLAB , we consider

c(p)= $\frac{1}{\varepsilon+p}$. Because we want to ignore the value of p=0.

We use the 5-point stencil values as a initial value of iterative linear and non-linear PDE. When we study this table, we can see the PSNR value increases to 5-point stencil method to iterative linear PDE method. Also PSNR value of the standard PDE method to iterative linear PDE method is increases. Therefore, iterative linear PDE method generates better inpainted image.

Case 3: Iterative Linear and Non-Linear Image Inpainting PDE with Gaussian Diffusion Conductivity

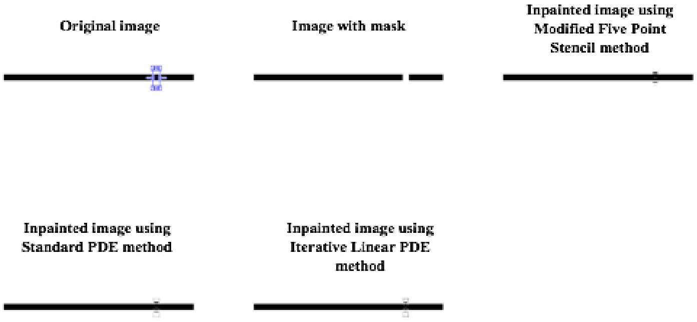

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard PDE method
Inpainted image using Iterative Linear PDE method

Figure 8.10: Image Inpainting methods with Gaussian Diffusion Conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	31.5166	33.5207	34.6632

Table 8.5: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Gaussian Proportional Diffusion Conductivity

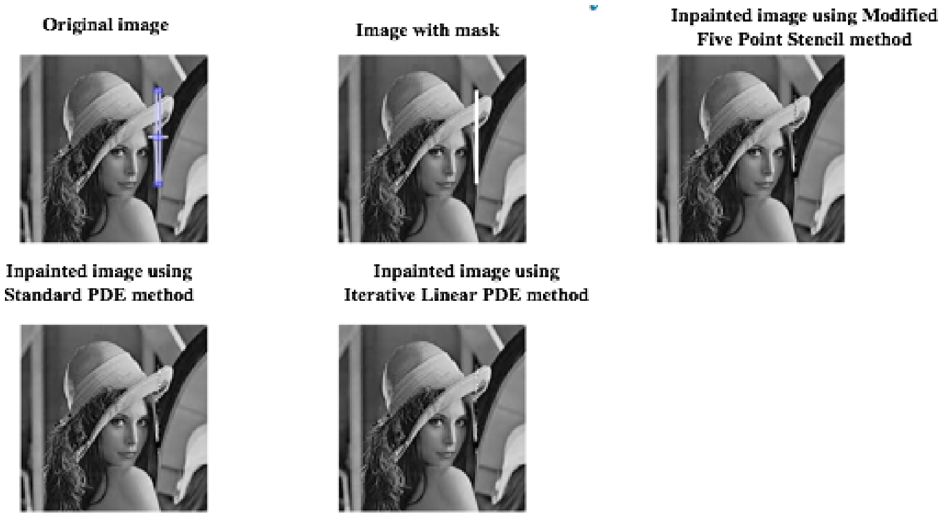

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard PDE method
Inpainted image using Iterative Linear PDE method

Figure 8.11: Image Inpainting methods with Gaussian Diffusion Conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	28.8209	31.0668	32.0291

Table 8.6: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Gaussian Proportional Diffusion Conductivity

Here, we consider the Gaussian Diffusion Conductivity. That is, c(p)= $e^{-\frac{p^2}{k^2}}$.

Here also, we use the 5-point stencil values as the initial value of iterative linear and non-linear PDE.

When we study this table, we can see PSNR value is increasing to the 5-point stencil method to iterative linear PDE method. But PSNR value of 5-point stencil method is decreasing in the non-Linear PDE method. That is the non-linear method offer the worst inpainted image. Therefore iterative linear PDE method generates best inpainted image.

Case 4: Iterative Linear and Non-Linear Image Inpainting PDE with Lorentz Diffusion Conductivity

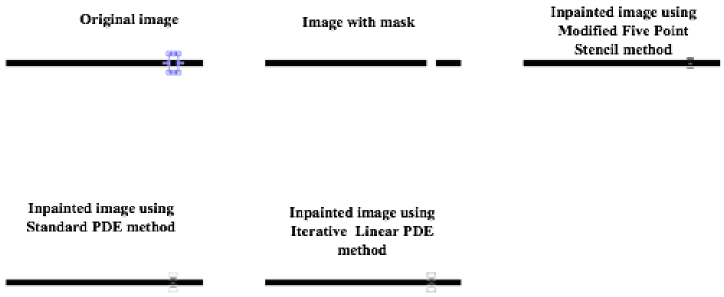

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard PDE method
Inpainted image using Iterative Linear PDE method

Figure 8.12: Image Inpainting methods with Lorentz Diffusion Conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	29.5464	32.3549	34.0114

Table 8.7: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Lorentz Proportional Diffusion Conductivity

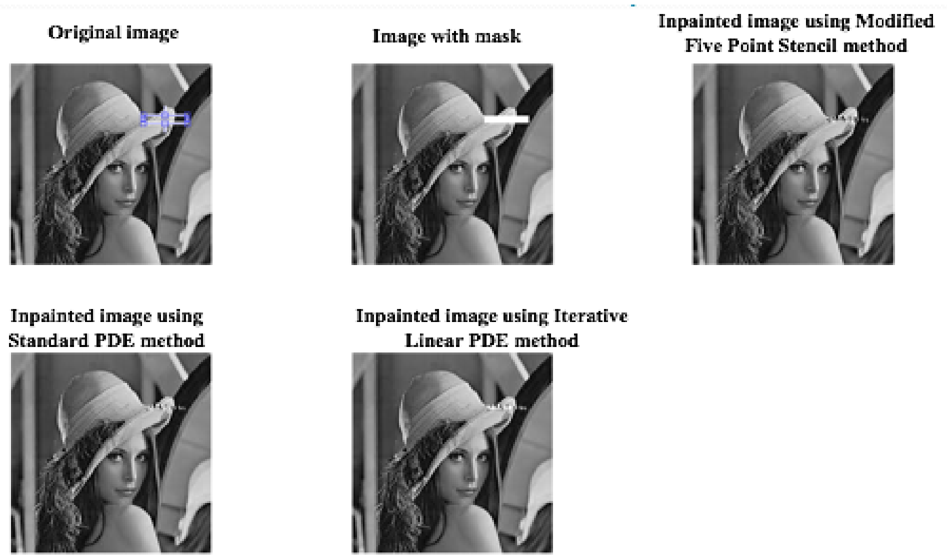

Original image
Image with mask
Inpainted image using Modified Five Point Stencil method
Inpainted image using Standard PDE method
Inpainted image using Iterative Linear PDE method

Figure 8.13: Image Inpainting methods with Lorentz Diffusion Conductivity

Inpainting Methods	5-point Stencil	Standard PDE	Iterative Linear PDE
PSNR	29.6713	31.5952	32.7003

Table 8.8: PSNR value for Iterative Linear and Non-Linear Image Inpainting PDE with Lorentz Proportional Diffusion Conductivity

Here we consider the Lorentz Diffusion Conductivity. That is,

c(p)= $\frac{1}{1+\frac{p^2}{k^2}}$.

Here also, we use the 5-point stencil values as a initial value of iterative linear and non-linear PDE.

When we study this table, we have same idea of previous methods. That is, the iterative linear PDE method gives the best inpainted image.

8.2 Error Analysis of Image Inpainting

When we use the inpainting method to inpaint a damage image, our final result is not exactly same as the original image. In this section, we are trying to find a relationship between error of the image inpainting with the inpainting domain.

Consider the **Theorem 7.1**,

$$\|F_D(z_0) - u_n(z_0)\|_D = \|(T_0T_1...T_{n-1})\|\|(L_nL_{n-1}...L_0F_D)\|_D \quad z_1 \in D$$

Now, apply the diffusion conductivity ,c(p)=1, then,

Lagged Diffusivity operator , L_i=Δ ,

T_i=T and

$G_i(x,y)$=G(x,y)

$$\begin{aligned}
\|F_D(z_0) - u_n(z_0)\|_D &= \|(T_0 T_1 ... T_{n-1})\| \quad \|(L_n L_{n-1} ... L_0 F_D)\|_D \\
&= \|(TT...T)\| \quad \|(\Delta\Delta...\Delta)F_D)\|_D \\
&= \|(TT...T)\| \quad \|\Delta^{n+1} F_D)\|_D \\
&\leq \|T\|^{n+1} \quad \|\Delta^{n+1} F_D)\|_D
\end{aligned}$$

$$\|F_D - u_n\|_D \leq \|\Delta^{n+1} F_D\|_D \quad \|T\|^{n+1}$$

$$\|T\| = \underset{x\in D}{sup} \int_D |G(z, z_0)dz_0| = \underset{x\in D}{sup} \left| \int_D G(z, z_0)dz_0 \right|$$

If we can find a value for Green's function then we can find a estimator for error of the image inpainting. In the next two sections, we use two different methods to find a Green's function.

8.3 Error using Poisson Equation

Consider the solution of Poisson equation,

$$\begin{aligned}
\Delta u(z) &= 1, \quad z \in D \\
u(z) &= 0, \quad z \in \partial D
\end{aligned}$$

We know that solution is given by,

$$u(z) = \int_D G(z, z_0)dz_0$$

Now we are solve this Poisson equation for different types of domain.

8.3.1 Circle,[20]

Where D is a circle with center zero and radius r

Consider the equation of the circle. It is,

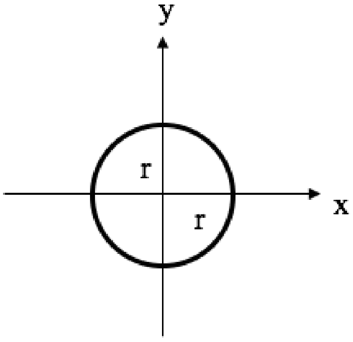


Figure 8.14: circle with center zero and radius r

$$x^2 + y^2 = r^2$$

We can guess the solution is,

$$u(z) = c(x^2 + y^2 - r^2)$$

This satisfies the boundary condition. Now we can find the constant c,

$$\begin{aligned}
\frac{\partial u}{\partial x} &= 2cx \quad , \quad \frac{\partial^2 u}{\partial x^2} = 2c \\
\frac{\partial u}{\partial y} &= 2cy \quad , \quad \frac{\partial^2 u}{\partial y^2} = 2c \\
\Delta u(z) &= 1 \\
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 1 \\
2c + 2c &= 1 \\
c &= \frac{1}{4}
\end{aligned}$$

Therefore,

$$u(z) = \frac{x^2 + y^2 - r^2}{4}$$
$$u(z) = \frac{|z|^2 - r^2}{4}$$

Consider Green's second formula,

$$\begin{aligned} u(z_0) &= \int_D G(z_0, z)(\Delta u(z))dxdy = \int_D G(z_0, z)dxdy \\ &= \frac{|z|^2 - r^2}{4} \\ \therefore \|T\| &= \sup_{x \in D} \left| \int_D G(z, z_0)dz_0 \right| = \left| \frac{|z|^2 - r^2}{4} \right| \leq \frac{r^2}{4} \end{aligned}$$

$$\begin{aligned} \therefore \|F_D - u_n\|_D &\leq \|\Delta^{n+1} F_D\|_D \quad \|T\|^{n+1} \\ &\leq \|\Delta^{n+1} F_D\|_D \quad \left(\frac{r^2}{4}\right)^{n+1} \end{aligned}$$

8.3.2 Ellipse

Theorem 8.3.2

Suppose D(a,b), where $a \geq b$, is an ellipse centered at zero, Then,

$$\|F_D - u_n\|_D \leq \|\Delta^{n+1} F_D\|_D \quad \left(\frac{1}{2}\left(\frac{a^2 b^2}{a^2 + b^2}\right)\right)^{n+1}$$

Proof: Consider the poisson equation on the ellipse,

$$\Delta u(z) = 1 \quad z \in D$$

$$u(z) = 0 \quad z \in \partial D$$

Where D is an ellipse with center zero and (a, b) such that $a \geq b$.

Consider the equation of the ellipse. It is,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

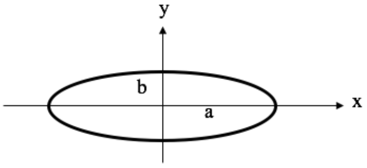


Figure 8.15: Ellipse with center zero and a > b

We can guess the solution is,

$$u(z) = c(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2})$$

This satisfies the boundary condition. Now we can find the constant c,

$$\begin{aligned} \frac{\partial u}{\partial x} &= -\frac{2cx}{a^2} \quad , \quad \frac{\partial^2 u}{\partial x^2} = -\frac{2c}{a^2} \\ \frac{\partial u}{\partial y} &= -\frac{2cy}{b^2} \quad , \quad \frac{\partial^2 u}{\partial y^2} = -\frac{2c}{b^2} \\ \Delta u(z) &= 1 \\ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 1 \\ -\frac{2c}{a^2} - \frac{2c}{b^2} &= 1 \\ c &= \frac{1}{-\frac{2}{a^2} - \frac{2}{b^2}} \end{aligned}$$

Therefore,

$$u(z) = \frac{1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}}{-\frac{2}{a^2} - \frac{2}{b^2}}$$

Consider Green's second formula,

$$\begin{aligned}
u(z_0) &= \int_D G(z_0,z)(\Delta u(z))dxdy = \int_D G(z_0,z)dxdy \\
&= \frac{1-\frac{x^2}{a^2}-\frac{y^2}{b^2}}{-\frac{2}{a^2}-\frac{2}{b^2}} \\
\therefore \|T\| &= \sup_{x\in D}\left|\int_D G(z,z_0)dz_0\right| = \left|\frac{1-\frac{x^2}{a^2}-\frac{y^2}{b^2}}{-\frac{2}{a^2}-\frac{2}{b^2}}\right| \\
&\le \left|\frac{1}{-\frac{2}{a^2}-\frac{2}{b^2}}\right| \\
&= \left|\frac{1}{\frac{2}{a^2}+\frac{2}{b^2}}\right| = \left|\frac{a^2b^2}{2(a^2+b^2)}\right|
\end{aligned}$$

$$\begin{aligned}
\therefore \|F_D-u_n\|_D &\le \|\Delta^{n+1}F_D\|_D \quad \|T\|^{n+1} \\
&\le \|\Delta^{n+1}F_D\|_D \quad \left(\frac{1}{2}\left(\frac{a^2b^2}{a^2+b^2}\right)\right)^{n+1}
\end{aligned}$$

8.3.3 Triangle

Theorem 8.3.3

Suppose D is an equilateral triangle with each side 2a, then

$$\|F_D-u_n\|_D \le \|\Delta^{n+1}F_D\|_D \quad \left(\frac{3a^2}{4}\right)^{n+1}$$

Proof

Consider the equation of each side. They are ,

$$\begin{aligned}
y &= 0 \\
y &= \sqrt{3}x+\sqrt{3}a \\
y &= -\sqrt{3}x+\sqrt{3}a
\end{aligned}$$

We can guess the solution is,

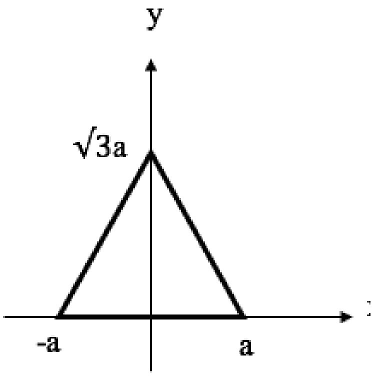


Figure 8.16: Triangle with each side is 2a

$$u(z) = cy(y - \sqrt{3}x - \sqrt{3}a)(y + \sqrt{3}x - \sqrt{3}a)$$

This satisfies the boundary condition. Now we can find the constant c,

$$\begin{aligned}
u(z) &= cy((y - \sqrt{3}a) - \sqrt{3}x)((y - \sqrt{3}a) + \sqrt{3}x) \\
&= cy((y - \sqrt{3}a)^2 - (\sqrt{3}x)^2) \\
&= cy(y^2 - 2\sqrt{3}ay + 3a^2 - 3x^2) \\
&= cy^3 - 2c\sqrt{3}ay^2 + 3ca^2y - 3cx^2y \\
&\frac{\partial u}{\partial x} = -6cxy \quad , \quad \frac{\partial^2 u}{\partial x^2} = -6cy \\
&\frac{\partial u}{\partial y} = 3cy^2 - 4c\sqrt{3}ay + 3ca^2 - 3cx^2 \quad , \quad \frac{\partial^2 u}{\partial y^2} = 6cy - 4ca\sqrt{3} \\
\Delta u(z) &= 1 \\
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 1 \\
-6cy + 6cy - 4ca\sqrt{3} &= 1 \\
-4ca\sqrt{3} &= 1 \\
c &= -\frac{\sqrt{3}}{12a}
\end{aligned}$$

Therefore,

$$u(z) = -\frac{\sqrt{3}}{12a}y(y-\sqrt{3}x-\sqrt{3}a)(y+\sqrt{3}x-\sqrt{3}a)$$

Consider Green's second formula,

$$\begin{aligned}
u(z_0) &= \int_D G(z_0,z)(\Delta u(z))dxdy = \int_D G(z_0,z)dxdy \\
&= -\frac{\sqrt{3}}{12a}y(y-\sqrt{3}x-\sqrt{3}a)(y+\sqrt{3}x-\sqrt{3}a) \\
\therefore \|T\| &= \underset{x\in D}{sup}\left|\int_D G(z,z_0)dz_0\right| = \left|-\frac{\sqrt{3}}{12a}y(y-\sqrt{3}x-\sqrt{3}a)(y+\sqrt{3}x-\sqrt{3}a)\right| \\
&\leq \left|\frac{\sqrt{3}}{12a}y[(y-\sqrt{3}a)^2-(\sqrt{3}x)^2]\right| \\
&= \left|\frac{\sqrt{3}}{12a}y[(y-\sqrt{3}a)^2-3x^2]\right| \\
&\leq \left|\frac{\sqrt{3}}{12a}y[(y-\sqrt{3}a)^2]\right| \\
&\leq \left|\frac{\sqrt{3}}{12a}y^3\right| \\
&\leq \left|\frac{\sqrt{3}}{12a}(\sqrt{3}a)^3\right| \quad where \quad 0\leq y\leq\sqrt{3}a
\end{aligned}$$

$$= \frac{9a^3}{12a} = \frac{3a^2}{4}$$

$$\begin{aligned}
\therefore \|F_D-u_n\|_D &\leq \|\Delta^{n+1}F_D\|_D \quad \|T\|^{n+1} \\
&\leq \|\Delta^{n+1}F_D\|_D \quad \left(\frac{3a^2}{4}\right)^{n+1}
\end{aligned}$$

8.3.4 Rectangle

Theorem 8.3.4

Suppose D(a,b), where $a \geq b$, is a rectangel. Then,

$$\|F_D - u_n\|_D \leq \|\Delta^{n+1} F_D\|_D \quad \left(\frac{abN^2}{2}\right)^{n+1}$$

where,

$$N^2 = \sum_{k=1}^{\infty} \frac{1}{k^2}$$

Proof

Consider the poisson equation on rectangle,

$$\begin{aligned} \Delta u(z) &= f(z) \quad z \in D \\ u(z) &= 0 \quad z \in \partial D \end{aligned} \tag{8.3.4.1}$$

First, we have to solve **eigenvalue problem** when we are solving this equation. Here we are finding function $v(z)$, and and constant λ such that

$$\Delta v(z) = \lambda v(z) \quad z \in D$$
$$v(z) = 0 \quad z \in \partial D$$

When we solve this problem, we use the separation of variables method.

$$v(z) = v(x, y) = X(x)Y(y)$$

consider the boundary condition for ractangle $D = [0, a] \times [0, b]$ such that $a > b$,

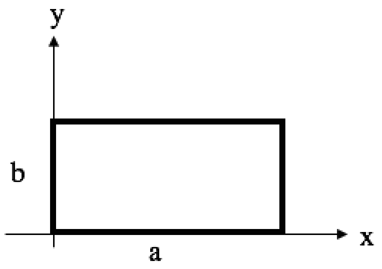


Figure 8.17: Rectangle with a>b

$$v(x,y) = 0$$

$$X(x)Y(y) = 0$$

$$X(0)Y(x) = 0 \quad X(x)Y(0) = 0 \quad X(a)Y(y) = 0 \quad X(x)Y(b) = 0$$

X(x) or Y(y) can not be equal to zero. Therefore,

$$X(0) = 0, \quad Y(0) = 0, \quad X(a) = 0, \quad Y(b) = 0$$

$$\Delta v(z) = \lambda v(z)$$

$$v_{xx}(x,y) + v_{yy}(x,y) = \lambda v(x,y)$$

$$X''(x)Y(y) + X(x)Y''(y) = \lambda X(x)Y(y)$$

$$\frac{X''(x)}{X(x)} + \frac{Y''(y)}{Y(y)} = \lambda$$

$\frac{X''(x)}{X(x)}$ depends on x and $\frac{Y''(y)}{Y(y)}$ depends on y. Therefore, they must be a constant.

$$\frac{X''(x)}{X(x)} = \alpha, \qquad \frac{Y''(y)}{Y(y)} = \beta, \quad \alpha + \beta = \lambda$$

$$X''(x) = \alpha X(x), \quad Y''(y) = \beta Y(y)$$

Consider each equation with boundary conditions,

$$X''(x) = \alpha X(x), \quad X(0) = 0, \quad X(a) = 0$$

$$X_m(x) = sin\left(\frac{m\pi x}{a}\right) \quad where \quad \alpha_m = \left(\frac{m\pi}{a}\right)^2 \quad m = 1, 2, 3, 4, ...$$

similarly we can find $Y(y)$,

$$Y''(y) = \beta Y(y), \quad Y(0) = 0, \quad Y(b) = 0$$

$$Y_n(y) = sin\left(\frac{n\pi y}{b}\right) \quad where \quad \beta_n = \left(\frac{n\pi}{b}\right)^2 \quad n = 1, 2, 3, 4, ...$$

Therefore, the solution of the Poisson equation on a rectangle is,

$$u(z) = u(x, y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} u_{mn} v_{mn}(x, y)$$

where,

$$u_{mn} := \lambda_{mn}^{-1} \frac{\langle f, v_{mn} \rangle}{\langle v_{mn}, v_{mn} \rangle}$$

$$v_{mn}(x, y) := X_m(x) Y_n(y) \quad \lambda_{mn} := \alpha_m + \beta_n$$

Consider our problem

$$\Delta u(z) = 1 \quad z \in D$$

$$u(z) = 0 \quad z \in \partial D$$

Here, we consider , $f = 1$.

$$
\begin{aligned}
\langle v_{mn}, v_{mn} \rangle &= \int_0^a \int_0^b X_m(x)^2 Y_n(y)^2 \, dx \, dy \\
&= \int_0^a X_m(x)^2 dx \quad \int_0^b Y_n(y)^2 dy \\
&= \int_0^a sin^2\left(\frac{m\pi x}{a}\right) dx \quad \int_0^b sin^2\left(\frac{n\pi y}{b}\right) dy \\
&= \frac{1}{2}\int_0^a 1 - cos\left(\frac{2m\pi x}{a}\right) dx \quad \frac{1}{2}\int_0^b 1 - cos\left(\frac{2n\pi y}{b}\right) dy \\
&= \frac{1}{2}\left[x - \frac{a}{2m\pi} sin\left(\frac{2m\pi x}{a}\right)\right]_0^a \quad \frac{1}{2}\left[y - \frac{b}{2n\pi} sin\left(\frac{2n\pi y}{b}\right)\right]_0^b \\
&= \frac{1}{2}\left[a - \frac{a}{2m\pi} sin(2m\pi)\right] \quad \frac{1}{2}\left[b - \frac{b}{2n\pi} sin(2n\pi)\right] \\
&= \frac{a}{2}\frac{b}{2} = \frac{ab}{4}
\end{aligned}
$$

$$
\begin{aligned}
\langle f, v_{mn} \rangle = \langle 1, v_{mn} \rangle &= \int_0^a \int_0^b 1 X_m(x) Y_n(y) \, dx \, dy \\
&= \int_0^a X_m(x) dx \quad \int_0^b Y_n(y) dy \\
&= \int_0^a sin\left(\frac{m\pi x}{a}\right) dx \quad \int_0^b sin\left(\frac{n\pi y}{b}\right) dy \\
&= \left[-\frac{a}{m\pi} cos\left(\frac{m\pi x}{a}\right)\right]_0^a \quad \left[-\frac{b}{n\pi} cos\left(\frac{n\pi y}{b}\right)\right]_0^b \\
&= \left[\frac{a}{m\pi}(cos(m\pi) - 1)\right] \quad \left[\frac{b}{n\pi}(cos(n\pi) - 1)\right]
\end{aligned}
$$

Consider one of these,

$$
(cos(m\pi) - 1) = \begin{cases} -2 & \text{if} \quad m \quad is \quad odd \\ 0 & \text{if} \quad m \quad is \quad even \end{cases}
$$

Therefore,

$$
\langle f, v_{mn} \rangle = \langle 1, v_{mn} \rangle = \begin{cases} \frac{2a}{m\pi}\frac{2b}{n\pi} & \text{if} \quad m, n \quad are \quad odd \\ 0 & \text{if} \quad m, n \quad are \quad even \end{cases}
$$

$$
= \frac{4ab}{mn\pi^2}, \quad when \quad m, n \quad are \quad odd
$$

$$\begin{aligned}\lambda_{mn} &= \alpha_m + \beta_n \\ &= \left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 \\ &= \left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)\pi^2 \\ \therefore \quad u_{mn} &= \lambda_{mn}^{-1}\frac{\langle f, v_{mn}\rangle}{\langle v_{mn}, v_{mn}\rangle} \\ &= \frac{1}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)\pi^2}\frac{\frac{4ab}{mn\pi^2}}{\frac{ab}{4}} \\ &= \frac{16}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)mn\pi^4}\end{aligned}$$

Therefore, the solution is given by,

$$\begin{aligned}u(z) = u(x,y) &= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} u_{mn}v_{mn}(x,y) \\ &= \sum_{m=1,3,5,...}^{\infty}\ \sum_{n=1,3,5,...}^{\infty} \frac{16}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)mn\pi^4} sin\left(\frac{m\pi x}{a}\right) sin\left(\frac{n\pi y}{b}\right)\end{aligned}$$

Consider Green's second formula,

$$\begin{aligned}u(z_0) &= \int_D G(z_0,z)(\Delta u(z))dxdy = \int_D G(z_0,z)dxdy \\ &= \sum_{m=1,3,5,...}^{\infty}\ \sum_{n=1,3,5,...}^{\infty} \frac{16}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)mn\pi^4} sin\left(\frac{m\pi x}{a}\right) sin\left(\frac{n\pi y}{b}\right), \quad a > b \\ &\leq \sum_{m=1,3,5,...}^{\infty}\ \sum_{n=1,3,5,...}^{\infty} \frac{16}{\left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right)mn\pi^4}\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{m=1,3,5,\ldots}^{\infty} \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{mn\left(\frac{m^2}{a^2}+\frac{n^2}{b^2}\right)} \\
&\leq \sum_{m=1,3,5,\ldots}^{\infty} \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{mn}\frac{1}{\frac{2mn}{ab}} \\
&\leq \frac{ab}{2} \sum_{m=1,3,5,\ldots}^{\infty} \frac{1}{m^2} \sum_{n=1,3,5,\ldots}^{\infty} \frac{1}{n^2} \\
&\leq \frac{ab}{2} \sum_{m=1}^{\infty} \frac{1}{m^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \\
&\leq \frac{abN^2}{2} \quad , where \quad M, N \quad are \quad constant
\end{aligned}$$

Note:

(1)

$$\begin{aligned}
\frac{1}{2AB} - \frac{1}{A^2+B^2} &= \frac{A^2+B^2-2AB}{2(A^2+B^2)AB} \\
&= \frac{(A-B)^2}{2(A^2+B^2)AB} \geq 0 \\
\therefore \frac{1}{2AB} - \frac{1}{A^2+B^2} &\geq 0 \\
\frac{1}{2AB} &\geq \frac{1}{A^2+B^2}
\end{aligned}$$

(2)

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

It is a convergent series.

$$\begin{aligned}
\therefore \|T\| &= \sup_{x\in D}\left| \int_D G(z,z_0)dz_0 \right| \\
&\leq |\frac{abN^2}{2}|
\end{aligned}$$

$$\begin{aligned}
\therefore \|F_D - u_n\|_D &\leq \|\Delta^{n+1}F_D\|_D \quad \|T\|^{n+1} \\
&\leq \|\Delta^{n+1}F_D\|_D \quad \left(\frac{abN^2}{2}\right)^{n+1}
\end{aligned}$$

8.4 Schwarz Christoffel Transformation from any polygon to unit disk

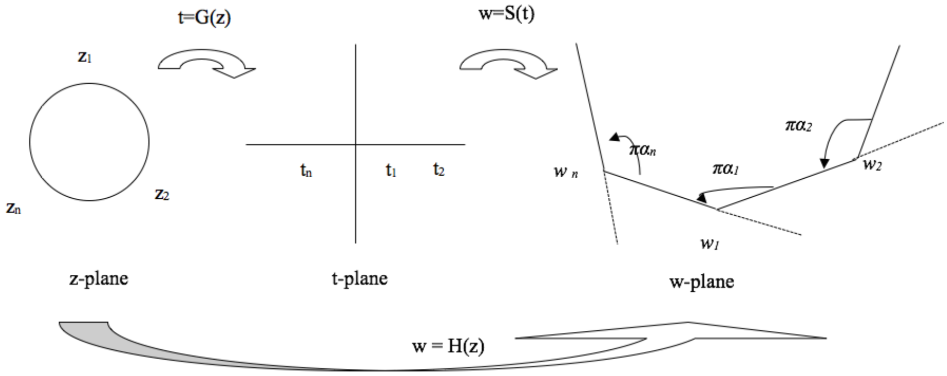


Figure 8.18: Schwarz Christoffel Transformation from any unit disk to polygon

First, we consider the transformation from unit disk to upper half plane. In this transformation, we consider the Mobius Transformation G(z) and then we consider the Schwarz Christoffel Transformation for upper half plane to polygon. In these two transformations, we apply composition function. But we have Schwarz Christoffel Transformation for derivative of the transformation. So, first we consider the derivative of composite function. Here we apply chain rule of the composite function.

We know the transformation from unit disk to upper-half plane is,

$$G(z) = i\frac{1-z}{1+z}$$

$$\begin{aligned} G'(z) &= i\frac{(1+z)(-1)-(1-z)(1)}{(1+z)^2} \\ &= i\frac{-1-z-1+z}{(1+z)^2} \\ &= i\frac{-2}{(1+z)^2} \end{aligned}$$

$$H(z) = S(G(z))$$

$$\begin{aligned}
H'(z) &= G'(z)S'(G(z)) \\
&= G'(z)C_0\prod_{k=1}^{n}\left(G(z)-G(z_k)\right)^{\alpha_k-1} \\
&= \frac{-2}{(1+z)^2}C_0\prod_{k=1}^{n}\left(i\frac{1-z}{1+z}-i\frac{1-z_k}{1+z_k}\right)^{\alpha_k-1} \\
&= \frac{-2C_0}{(1+z)^2}\prod_{k=1}^{n}\left(i\frac{1-z}{1+z}-i\frac{1-z_k}{1+z_k}\right)^{\alpha_k-1} \\
&= \frac{-2C_0}{(1+z)^2}\prod_{k=1}^{n}i\left(\frac{(1+z_k)(1-z)-(1-z_k)(1+z)}{(1+z)(1+z_k)}\right)^{\alpha_k-1} \\
&= \frac{-2C_0}{(1+z)^2}\prod_{k=1}^{n}i\left(\frac{1+z_k-z-zz_k-1-z+z_k+zz_k}{(1+z)(1+z_k)}\right)^{\alpha_k-1} \\
&= \frac{-2iC_0}{(1+z)^2}\prod_{k=1}^{n}\left(\frac{-2(z_k-z)}{(1+z)(1+z_k)}\right)^{\alpha_k-1} \\
&= \frac{4iC_0}{(1+z)^2}\prod_{k=1}^{n}\left(\frac{1}{1+z}\right)^{\alpha_k-1}\prod_{k=1}^{n}(z_k-z)^{\alpha_k-1}\prod_{k=1}^{n}\left(\frac{1}{1+z_k}\right)^{\alpha_k-1} \\
&= \frac{4iC_0}{(1+z)^2}\left(\frac{1}{1+z}\right)^{\sum_{k=1}^{n}\alpha_k-1}\prod_{k=1}^{n}(z_k-z)^{\alpha_k-1}\prod_{k=1}^{n}\left(\frac{1}{1+z_k}\right)^{\alpha_k-1} \\
&= \frac{4iC_0}{(1+z)^2}\left(\frac{1}{1+z}\right)^{2}\prod_{k=1}^{n}(z_k-z)^{\alpha_k-1}\prod_{k=1}^{n}\left(\frac{1}{1+z_k}\right)^{\alpha_k-1} \\
&= C\prod_{k=1}^{n}(z_k-z)^{\alpha_k-1}\prod_{k=1}^{n}\left(\frac{1}{z_k}\right)^{\alpha_k-1} \\
&= C\prod_{k=1}^{n}\left(\frac{z_k-z}{z_k}\right)^{\alpha_k-1} \\
&= C\prod_{k=1}^{n}\left(1-\frac{z}{z_k}\right)^{\alpha_k-1}
\end{aligned} \tag{8.4.0.2}$$

$$\begin{aligned} \frac{dw}{dz} = H'(z) = C \prod_{k=1}^{n} \left(1 - \frac{z}{z_k}\right)^{\alpha_k - 1} \\ H(z) = A + C \int_{z_0}^{z} \prod_{k=1}^{n} \left(1 - \frac{\zeta}{\zeta_k}\right)^{\alpha_k - 1} d\zeta \end{aligned} \tag{8.4.0.3}$$

Here, A and C are complex constant. This is conformal mapping from unit disc to unit circle.

Consider the inverse of equation (6.3.2.1). It is possible, because w= w(z) is a conformal mapping .That is $|\frac{dw}{dz}| > 0$, everywhere [38].

$$\frac{dz}{dw} = \frac{1}{C} \prod_{k=1}^{n} \left(1 - \frac{z}{z_k}\right)^{1-\alpha_k}$$

Therefore, conformal mapping from polygon to unit disc is

$$B + \frac{1}{C} \int_{z_0}^{z} \prod_{k=1}^{n} \left(1 - \frac{\zeta}{\zeta_k}\right)^{1-\alpha_k} d\zeta$$

8.4.1 Erros using conformal mapping

When we calculate the error, we have to find Green's function. Now we try to define Green's function using conformal mapping, such that,[20]

$$G(z_0, z) = \frac{1}{2\pi} ln|H_{z_0}(z)|$$

Where $H_{z_0}(z)$ is a conformal mapping from any polygon to unit disk, such that

a) D is a one-to-one and onto onto map in the unit disk with center at origin

b) $H_{z_0}(z_0)$=0

Use the error formula,

$$F(z_0) - u_n(z_0) = \int_D K_{n+1}(z_0, z)\Delta^{n+1}F(z)dz, \quad z_0 \in D$$

Where, [20]

$$K_1(z_0, z) = G(z_0, z) \quad and$$

$$K_i(z_0, z) = \int_{D^i} G(z_0, z_1)...G(z_0, z_{i+1})dz_1...dz_{i+1}$$

Chapter 9

Future Work

In this reaserch work, we were able to form a formula for a transformation from any polygon to unit circle. In future work, we can find the error of image inpainting by using this result. Going foraward here, a reasercher can extend this result to image inapinting on manifolds.

Bibliography

[1] J. Shen and T. F. Chan. *Image processing and Analysis.* SIAM, 2005

[2] T. A. Driscoll. *Schwarz Christoffel Mapping.* Cambridge University Press, 2002.

[3] J. Brown and R. Churchill. *Complex Variables and Applications.* McGraw-Hill.

[4] V. I. Ivanov and M. K. Trubetskov. *Handbook of conformal mapping with computer aided visualization.* CRC press.

[5] G. Wen. *Conformal Mapping and Boundary Value Problems, translations of Mathematical Monographs.* AMS, Vol. 106.

[6] Z. Nehari. *Conformal Mapping.* McGraw-Hill, 1952.

[7] J. Cai, R. H. Chan, L. Shen and Z. Shen. *Simultaneously Inpainting in Image and Transformed Domains*, Numerishe Mathematik, pages 509-533, 2009.

[8] S. Osher, J. Cai and Z. Shen. *Split Bergman Methods and frame based image restoration*, SIAM Interdiciplinary, Vol. 2, pages 337-369, 2009,.

[9] L. Shen, Q. Li and L. Yang. *Split-Bergman iteration for farmalet based image inpainting*, Computational Harmonic Analysis, Vol. 32, pages 145-154, 2012.

[10] H. Hosseini, N.B. Marvasti and F. Marvasti. *Image Inpainting Using Sparsity of the Transform Domain*,

[11] H. Zhang and S. Dai. *Image Inpainting Based on Wavelet Decomposition*, 2012 International Workshop on Information and Electronics Engineering, Elsevier Science, Vol. 29, pages 3674-3678, 2012

[12] F. Wang, D. Liang, N. Wang, Z. Cheng and J. Tang. *An new method for image inpainting using wavelets*, Multimedia Technology ICMT, 2011 International Conference, IEEE, pages 201-204, 2011.

[13] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester. *Image Inpainting.* SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques, Wesley Publishing Co. New York, NY, USA, pages 417-424, 2000.

[14] J. Cai, R. Chan, L. Shen and Z. Shen. *A Framelet based image inpainting algorithm* , Applied and computational harmonic analysis, Elsevier Science, Vol. 24, pages 131-149, March 2008.

[15] T. F. Chan and J. H. Shen. *Non-texture inpainting by Curvature-Driven Diffusions(CDD)* , Visual Communication and Image Representation, Vol. 12, pages 436-449, 2001.

[16] J. Yang. *Image inpainting using complex 2-D dual-tree wavelet transform* , Appl. Math. J. Chinese Univ, Vol. 26, pages 70-76, 2011.

[17] M. M. Oliveira, B. Bowen, R. McKenna and Y. Chang. *Fast Digital Image Inpainting*, VIIP 200.

[18] M. Bertalmio, A.L.Bertozz and G.Sapiro. *Navier Stokes Fluid Dynamics and Image and Video Inpainting*, In computer Vision and Pattern Recognition, Vol. 1, pages I-355 - I-362, 2011.

[19] O. Sher, G. Sapiro, M. Bertalmo and L. Vese. *Simultaneous structure and texture image inpainting*, IEEE Transactions on Image Processing, Vol. 12, pages 882-889, 2003

[20] C. K. Chui. *An MRA approach to surface completion and image inpainting*, Appllied and Computational Harmonic Analysis, Elsevier Science, Vol. 26, pages 270-276, 2009.

[21] C. K. Chui and J. Wang. *Wavelet-based minimum-energy approach to image restoration* , Appllied and Computational Harmonic Analysis, Elsevier Science Vol. 23, pages 114-130, 2007.

[22] C. K. Chui and H.N. Mhaskar. *MRA Contextual-Recovery Extension of Smooth Functions on Manifolds*, Appllied and Computational Harmonic Analysis, Elsevier Science, Vol. 28, 104-113, 2010.

[23] T. F. Chan ,S. H. Kang. *Error Analysis for Image Inpainting*, Mathematical Imaging and Vision, Vol. 26, pages 85-103, November 2006.

[24] P. Perona and J. Malik. *Scale space and edge detection using anisotropic diffusion* , IEEE PAML, Vol. 12, pages 629-639, 1990.

[25] D. Heeger and J. Bergen. *Pyramid-based texture analysis/synthesis* , SIDDGRAPH95, pages 229-238, July 1995.

[26] J. S. DeBonet. *Multiresolution sampling procedure for analysis and synthesis of texture images* , SIDDGRAPH97, pages 361-368, 1997.

[27] A. Efros and T. Leung. *Texture Synthesis by Non-parametric sampling*, International Conference on computer Vision 2, pages 1033-1038, September 1999.

[28] T. F. Chan and J. Shen. *Mathematical Models for local non-texture inpaintings* , SIAM, Vol. 62(3), 2002.

[29] S. Osher, X. Tai and T. Rahman. *TV stokes denoising algorithm*

[30] T. F. Chan, J. Shen and L. Vese. *Variational PDE models in image processing* , Notices of the American Mathematical Society, pages 14-26, 2003.

[31] N. G. Kingsbury, I. W. Selesnick and R.G. Baranuik. *The Dual-Tree complex wavelet transform* , IEEE signal processing magazine, page 123, 2005.

[32] Z. Shen, R. H. Chan and J. Cai. *SIMULTANEOUS CARTOON AND TEXTURE INPAINTING* , AMIS'

[33] S. Osher and T. Goldstein. *The Split Bregman Method for L1-Regularized Problems* , SIAM Imaging Science, Vol. 2, pages 323-343, 2009.

[34] Z. Shenx, J. Cai, R. H. Chany and L. Shenz. *Convergence Analysis of Tight Framelet Approach for Missing Data Recovery*, Advanced in Computational Analysis, Vol. 31, pages 87-113, 2009.

[35] V. Caselles. *Exemplar-based Image Inpainting and Applications*, SIAM, Vol. 44, 2011.

[36] Z. Shenx, J. Cai and S. Osher. *CONVERGENCE OF THE LINEARIZED BREGMAN ITERATION FOR L1-NORM MINIMIZATION*, Mathematics of Computation, Vol. 2, pages 323-343, 2009.

[37] C. R. Vogel and M. E. Oman. *Iterative methods for total variation denoising*, SIAM, Vol. 17, pages 227-238, 1996.

[38] L. Trefethen. *NUMERICAL COMPUTATION OF THE SCHWARZ-CHRISTOFFEL TRANSFORMATION*, STAN-CS-79-710, March 1979.

[39] A. N. Hirani and T. Totsuka. *Combining Frequency and Spatial Domain Information for Fast Interactive Image Noise Removal*, SIGGRAPH, Vol. 78, pages 2127-2136, 2009.

[40] G. Sapiro and K. A. Patwardhan. *Projection based image and video inpainting using wavelets*, In Proc. ICIP, pages 857–860, 2003.

[41] J. Verdera, V. Caselles, M. Bertalmio and G. Sapiro. *Inpainting Surface Holes*, In Int. Conference on Image Processing, pages 903-906, 2003.

[42] M. Bertalmio, V. Caselles, S. Masnou and G. Sapiro. *Inpainting*, http://math.univ-lyon1.fr/ masnou/fichiers/publications/survey.pdf

[43] G. Sapiro. *Image Inpainting*, iie.fing.edu.uy/investigacion/grupos/gti/cursos/slides/upc3.p

[44] *Schwarz Christoffel mapping*, www.wikipedia.org

[45] *Greens Function*, www.wikipedia.org

[46] *PSNR*, www.wikipedia.org

[47] *Mobius Transformation*, www.wikipedia.org