UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Mathematical Approaches to Color Denoising and Image Inpainting Problems (Ph.D. Thesis)

Sung Ha Kang

May 2002

CAM Report 02-25

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

University of California

Los Angeles

Mathematical Approaches to Color Denoising and Image Inpainting Problems

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Sung Ha Kang

2002

The dissertation of **Sung Ha Kang** is approved.

Stanley J. Osher

Russel E. Caflisch

Stefano Soatto

Tony F. Chan, Committee Chair

University of California, Los Angeles

2002

To my mother

Table of Contents

List of Figures

Acknowledgments

Chapter 2 was published in *Journal of Visual Communication and Image Representation*, co authored with Tony F. Chan and Jianhong Shen [CKS01b].

Revised version of Chapter 4 is submitted to *SIAM Journal of Applied Mathematics*, co authored with Tony F. Chan and Jianhong Shen [CKS01a].

Short version of Chapter 5 is to appear in the *Proceedings of 3D Data Processing Visualization Transmission*, June 2002 and complete version is submitted to *IEEE PAMI, Pattern Analysis and Machine Intelligence*, co authored with Tony F. Chan and Stefano Soatto [CKS02]

This research was supported in part by NSF grant, number DMS-9973341 and from ONR under N00014-96-1-0277.

I wish to express my deepest appreciation to my advisor, Prof. Tony F. Chan, for all his advice, guidance and support throughout my graduate studies at UCLA. His enthusiasm, constant encouragement and sincere considerations have helped me achieve more in my studies. I would also like to thank Prof. Stanley Osher, Prof. Stefano Soatto and Prof. Luminita Vese for their valuable feedbacks, constructive suggestions and kind encouragements, and additional thanks to my committee member Prof. Russel Caflisch for his guidance. Especially, I wish to express my sincere gratitude to Prof. Jianhong (Jackie) Shen for being a best mentor a graduate student can hope for, and for all his guidance and encouragements throughout my studies.

I would also like to thank my colleagues, especially UCLA imagers, Chiu-Kwong Wong, Haomin Zho, Berta Sandberg, Bing Song, Wei Zhu, Ming-Ham (Andy) Yip for valuable discussions and support. I am grateful to Myungjoo

Kang and Heasun Shim for their generous considerations, and Howard Lee, Doreen Deleon, Suzanne Nezzar, Jared Tanner, Andrew Nevai-Tucker for their friendship and encouragements. Additional thanks to my officemates Richard Tasi, Li-Tien Cheng, Brain Walter and Chiu-Yen Kao. I like to acknowledge my brother, all my friends from Korea and colleagues from Yonsei University for their sincere support. I extend my thanks to all faculty, staff and students in UCLA math department.

Vita

1974	Born, Seoul, Korea
1997	B.S. Mathematics Yonsei University, Seoul, Korea
1997 - 2001	Teaching Assistant and Research Assistant UCLA, Los Angeles, California.
2001 - 2002	Dissertation Year Fellowship Graduate Division UCLA, Los Angeles, California

Publications

Tony F. Chan, Sung Ha Kang, and Jianhong Shen, *Total Variation Denoising and Enhancement of Color Images Based on the CB and HSV Color Models* Journal of Visual Communication and Image Representation, Volume 12, 422-435, 2001, Copyright 2001, Elsevier Science (USA). All rights reserved

Tony F. Chan, Sung Ha Kang, and Jianhong Shen, *Euler's Elastica and Curvature Based Inpaintings*, Submitted to SIAM Journal of Applied Mathematics, 2001

Sung Ha Kang, Tony F.Chan and Stefano Soatto, *Inpainting from Multiple Views*,

To appear in the Proceedings of 3DPVT, Proceedings of 3D Data Processing Visualization Transmission, June 2002

Sung Ha Kang, Tony F.Chan and Stefano Soatto, *Landmark based Inpainting from Multiple Views*, Submitted to IEEE PAMI, Pattern Analysis and Machine Intelligence

Abstract of the Dissertation

Mathematical Approaches to Color Denoising and Image Inpainting Problems

by

Sung Ha Kang

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2002

Professor Tony F. Chan, Chair

This dissertation contains studies on using Partial Differential Equations and mathematical methods for color denoising and image inpainting problems. For the color denoising method, we study the total variation (TV) restoration based on the two nonlinear (or non-flat) color models: the Chromaticity-Brightness (CB) model and Hue-Saturation-Value (HSV) model. We make use of recent works on the variational/PDE method for non-flat features to denoise the chromaticity and hue components directly.

Inpainting problem arise when we have damaged old photo or art pictures which we need to revise. Motivated by earlier work by Bertalmio, Sapiro, Caselles and Ballester [BSC00], we have studied PDE based inpainting problems. From the variational TV inpainting model suggested by Chan and Shen, we have improved the results by adding a Curvature term to the regularization functional. In addition, we consider more global inpainting problems in which the missing region are so large that most local inpainting methods fail. As an alternative to the local principle, we assume that there are other images with related global information to enable a reasonable inpainting. As one possible method, we propose

using Landmark matching, interpolation, and inpainting. Finally, we study how the inpainting error depends on the shape and size of the inpainting domain. In practice, the error is more local than global and the error can be refined by local width. The error bound is dependent on the image space assumed, therefore, we will investigate two cases, first assuming image space is smooth $u \subset \mathcal{C}^2$ and using Harmonic inpainting and secondly for binary image u with TV inpainting.

CHAPTER 1

Introduction

Image processing is a broad field with many interesting tasks. Typical image processing task includes image restoration, compression and segmentation. In this dissertation, we are focusing our attention to image restoration, which includes denoising and inpainting problems.

As various subject areas are doing research on image processing, there are various frameworks for processing images. For example, statistical methods for averaging to texture analysis, Fourier or Wavelet transformations including recent improved curvelet, ridgele, and bandlets. We are interested in using PDE method to deal with image processing. There are some good features for using PDE imaging. It treats images as continuous functions and use PDE concepts like gradients, diffusion, curvature, level sets and more. In addition to its flexibility in modeling, it also have some of advantages over getting sharper edges and better geometric properties. Applications can be widely found in image segmentation, denoising, deblurring, enhancement, inpainting, and motion estimation

1.1 Image Restoration

Image restoration, particularly image denoising starts from image model like following,

$$u_o = K * u + \nu$$

where u_o is given observed image, u true image, K blurring kernel and ν is the noise term. The restoration problem is to find the true image u, only from observed image u_o. We will assume $K = I$, therefore, there is no blurring.

It has been known that by properly choosing the right cost or energy functionals in the variational formulation in the PDE formulation, edges in noisy images can be well restored and enhanced. Among the many possible choices, the *Total Variation* (TV) is the simplest but sufficiently efficient measurement for enhancement or denoising. Its PDE form (via the Euler-Lagrange equation) is interestingly connected to anisotropic diffusions and the mean curvature motion. Ever since the first explicit demonstration of its successful performance in image restoration was made by Rudin, Osher and Fatemi [RO94, ROF92], the TV model has been studied by many authors. Given the noisy image $u^0 : \Omega \to R$, Total Variation minimizing is to find u which satisfies the following,

$$\min_u f(u) \quad = \quad \int_\Omega |\nabla u| \, dx \, dy + \frac{\lambda}{2} \int_\Omega |u - u^0|^2 \, dx \, dy.$$

Throughout this dissertation, many models are based on this Total Variation Minimizing method.

1.2 Contributions and Organization of This Dissertation

In this dissertation, we are focusing our attention on color denoising and image inpainting problems.

Most denoising and enhancement methods for color images have been formulated on linear color models, namely, the channel-by-channel model and vectorial model. In Chapter 2, we study the total variation (TV) restoration based on the two nonlinear (or non-flat) color models: the Chromaticity-Brightness (CB) model and Hue-Saturation-Value (HSV) model. These models are known to be

closer to human perception. We made use of recent works on the variational/PDE method for non-flat features and denoise the chromaticity and hue components directly. We present both the mathematical theory and digital implementation for the TV method. Comparison to the traditional TV restorations based on linear color models is made through various experiments. The CB model seems to give better color control and detail control, also separating Hue and Saturation is not as good as using combined chromaticity.

Inpainting Problem arise when we have damaged old photo or art pictures which we need to revise. Recently, Bertalmio,Sapiro, Caselles and Ballester [BSC00] and Masnou and Morel [MM98] draw our attention to the inpainting problem. In Chapter 3, we show a couple of results for color image inpainting. From the TV inpainting model suggested by Chan and Shen [CS01a], we are able to extend the model to 3-D color space and show some successful results. In Chapter 4, we improve the TV inpainting model by adding the curvature term to the regularization functional. This is a variational inpainting method withe Euler's Elastica term. The curvature term is the functional helps to avoid a sharp edge (kink) which has very big curvature, and a result in getting connected result or curved result instead of straight line. With curvature term, the Euler-Lagrange equation becomes forth order highly nonlinear equation, so we use the normal and tangent vectors to better represent this Euler-Lagrange equation. We present the Mathematical Model using geometrical vectors as well as successful results using min-mod and upwind schemes.

Most existing inpainting algorithms are local in nature and extrapolate information from neighboring pixels into the inpainting regions. In Chapter 5, we are interested in the inpainting problem where the missing region are so large that these local inpainting methods fail. As an alternative to the local principle, we

assume that there are other images with related global information to enable a reasonable inpainting. These additional images could be from a movie sequence, an image of the same object from a different time and a different viewpoint, or an image of a similar object. We propose a method with roughly three phases: landmark matching, interpolation, and copying. For the landmark matching, modified shape context information is used to exploit the global information. Then matched information is interpolated (and regularized) using thin plate splines. Finally, we copy the information from one image to another. Using landmark matching and interpolation, allows the missing regions to be significantly larger compared to the local inpainting methods, and can be used when the object is distorted from one image to another.

There have been many recent developments in image inpainting problems and most inpainting domain is narrow or small enough to get good results. Therefore, in Chapter 6, we are interested in getting a local error bound for these inpainting problems. The error bound is depended on the image space assumed and we consider two cases, first assuming image space is smooth $u \subset C^2$ and using Harmonic inpainting and secondly for binary image u with TV inpainting. In both cases, we get explicit form for the error bound. For the continuous case, we are able to reduce the error bound specially when the inpainting domain is shape of an ellipse. For binary case, we use level line to analyze the error bound and show the error is bounded by the distance between the level lines and the normal direction of the level lines.

CHAPTER 2

Total Variation Denoising and Enhancement of Color Images Based on the CB and HSV Color Models

2.1 Introduction

Any tool that attempts to denoise and enhance digital color images must rely on two correlated ingredients — the representation of colors (i.e. the color model) and the restoration methodology applied to the representation. Therefore, this chapter starts with a brief introduction to these two components.

2.1.1 RGB, CB, and HSV Color Models

In image processing, color has been represented or modeled in various ways [GW92]. In this chapter we shall focus on the RGB model and HSV model.

In the RGB representation of color images, at each pixel $p = (x, y)$, the vectorial value $I(p) = (u_1(p), u_2(p), u_3(p))$ represents the intensity of the three primary colors separately. Each monochromatic component u_i is called a channel. The RGB model has led to the CB model, which decomposes an RGB pixel value $I(p)$ into two components — the brightness component $u(p) = \|I(p)\|$ (Euclidean length), and the chromaticity component $f(p) = I(p)/u(p)$. The chromaticity

component lives on the unit sphere S^2. Features that live on nonlinear manifolds are said to be *non-flat* [CS00b, TSC00]. It has been shown by several authors that this CB model is well suited for denoising, edge detection and enhancement, and segmentation [CS00b, TSC99, TSC00, TKV96, TV93].

HSV is a different color system that is believed to be more natural than the RGB system for human perception. The three variables are: hue H, saturation S, and value V. The S and V are linear features and take values in the interval $[0, 1]$. S encodes the "purity" of color: larger S corresponds to purer color. The value V stores the intensity information; larger V value means brighter color. The hue component H, though also taking values in $[0, 1]$, is a circular or periodic feature: as H increases from 0 to 1, the color spectra revolve from red, yellow, green, cyan, blue, to magenta, and eventually back to red. Therefore, the hue can be thought of non-flat features [KS02] lives on the unit circle S^1. The circularization is easily made by the exponential mapping $H \to \exp(i2\pi H)$.

The precise transition from the $[r, g, b]$ variables to $[H, S, V]$ is realized by:

$$\begin{aligned} V &= r \vee g \vee b \\ S &= 1 - \tfrac{r \wedge g \wedge b}{r \vee g \vee b}, \end{aligned}$$

where for a pair of real numbers a and b: $a \vee b := \max(a, b)$, $a \wedge b := \min(a, b)$. The formula for H appears to be more complicated, but it is a piecewise linear function along any iso-S & V line. On the hexagonal iso-S & V line in Figure 2.1, as point moves from $a1, a2, \cdots$ to $a6, a1$, the H value increases linearly from 0 to 1. For Example, along the first segment $[a1, a2]$ (along which r and g are fixed)

$$H = H(\cdot, g, \cdot) = \frac{1}{6}\left[\frac{g-b}{r-b}\right];$$

and along $[a3, a4]$ (along which r and b are fixed)

$$H = H(\cdot, \cdot, b) = \frac{2}{6} + \frac{1}{6}\left[\frac{b-r}{g-r}\right].$$

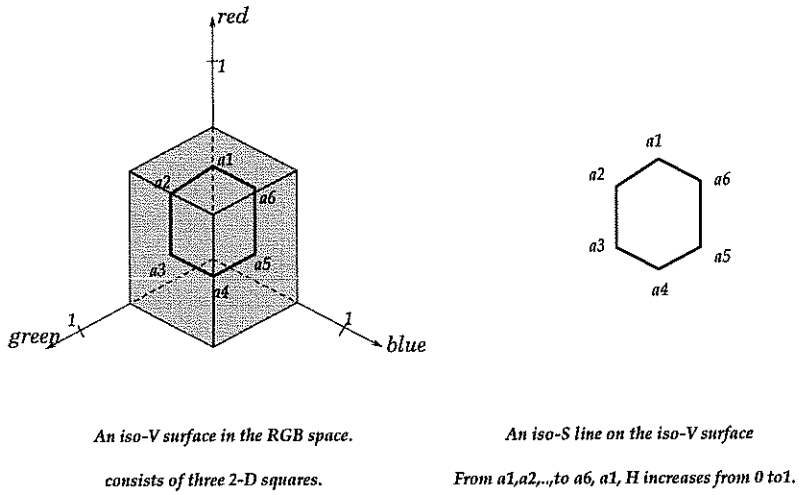


Figure 2.1: An iso-V surface and an iso-S line on it in the RGB space. See Eq. (2.1,2.1) for the definition. Notice that an iso-V & S line is a 3-D "hexagon," as shown separately on the right panel. From $a1, a2, \cdots$ to $a6, a1$, the hue value H increases from 0 to 1.

Many classical processing tools for color images are based on the RGB model, mostly because linear spaces are easy to work with. As a result, despite their similarity to human color perception, the CB and HSV color models have been less favored due to their non-flatness. This chapter intends to construct restoration models based on such non-flat color models.

2.1.2 The Methodology of Variational Restoration

The variational and PDE method has attracted much attention in image processing because of its flexibility in modeling and numerous advantages in numerical computation. Applications are widely found in image denoising, deblurring, enhancement, inpainting, segmentation, and motion estimation (see the two monographs [MS95, Wei98], for examples).

The general framework of the variational approach for denoising and enhancing color images based on the linear RGB color models can be classified into two categories — the channel-by-channel approach and the vectorial approach.

In the channel-by-channel approach, each channel u_i is assumed to be contaminated by noise η_i so that the observation becomes $u_i^0(x,y) = u_i(x,y) + \eta_i(x,y)$. A typical channel-by-channel denoising model carries the form of

$$\min_{u_i} R_i(u_i) \quad \text{subject to} \quad \frac{1}{|\Omega|}\int_\Omega |u_i - u_i^0|^2 = \sigma_i^2, \tag{2.1}$$

where Ω is the image domain, $|\Omega|$ its area, R_i the regularity functional, and σ_i the noise level. One example of R_i is the total variation approach (Rudin, Osher and Fatemi [RO94, ROF92]),

$$R_i(u_i) = \int_\Omega |\nabla u_i| dxdy.$$

The TV measure has been proven both mathematically and computationally to recognize and enhance edges embedded in noise [CL97, ROF92].

In the vectorial denoising approach, the cost functional is on the vectorial function $I = (u_1, u_2, u_3)$:

$$\min_I R_{3D}(I) \quad \text{subject to} \quad \frac{1}{|\Omega|}\int_\Omega \|I - I^0\|^2 = \sigma^2, \tag{2.2}$$

where R_{3D} is a regularity functional for vector-valued functions. Typically in applications, suppose R_i is a suitable 1-dimensional regularity functional for the i-th channel, then one simply takes $R_{3D} = \sqrt{R_1^2 + R_2^2 + R_3^2}$. In the case when R_{3D} is connected to the total variation and anisotropic diffusions, we observe the work of Sapiro and Ringach [SR96] and Blomgren and Chan [Blo98, BC98].

In practice, both constrained optimization problems (2.1) and (2.2) are replaced by their unconstrained forms. For example, for the vectorial case, we

solve

$$\min_I \left(R_{3D}(I) + \frac{\lambda}{2} \int_\Omega \|I - I^0\|^2 \right).$$

Here λ is an appropriate penalty weight, which depends on the noise level. In practice, it is often estimated or chosen *a priori* [COS01, ROF92].

2.1.3 Variational Restoration for Non-Flat Features

Recently, the variational and PDE method has been generalized to non-flat features by Sochen, Kimmel and Malladi [SKM98], Perona [Per98], Tang, Sapiro and Casselles [TSC99, TSC00], Chan and Shen [CS00b], and most recently by Kimmel and Sochen [KS02]. Tang et al. and Chan et al. generalize the total variational model, while Kimmel et al.'s model profits from their previous general framework of Polykov action and Beltrami operators [SKM98]. All these works intend to restore edges of non-flat features. Chan and Shen [CS00a] also proved that near edges, the models in [CS00b] and [KS02, SKM98] are equivalent. But the differential equations based on the total variation norm are much simpler and thus easier for computation [CS00b, TSC99, TSC00].

This chapter is the extension and completion of the works of [CS00b, TSC99, TSC00] on the total variation approach for enhancing and denoising non-flat image features. We study in details the TV model and its numerical implementation for restoring color images based on the CB and HSV color models. Previous works on general non-flat features enable us to work with the chromaticity feature f (on the sphere S^2) and hue feature H (along the circle S^1) directly. Detailed comparison to the channel-by-channel and vectorial approaches are illustrated through numerical examples. Our results show convincingly the advantages of the CB and HSV color models over linear ones.

The chapter is organized as follows. Section 2.2 introduces the mathemati-

cal models, and Section 2.3 details their numerical implementations. Numerical experiments and comparison are explained in Section 4.

2.2 The TV Formulation on CB and HSV Models

In this section, we explain the total variation formulation for the two nonlinear color models.

2.2.1 TV for the CB model

In the RGB representation, a color image is a mapping

$$I : \Omega \to R^3_+ = \{(r, g, b) : r, g, b > 0\}.$$

I can be separated into the *brightness component* $u = \|I\|$, and the *chromaticity component* $f = I/\|I\| = I/u$. The brightness u can be treated as a gray image, thus any scalar denoising model can be applied. However, the chromaticity component f stores the color information, and takes values on the unit sphere S^2. We need to apply the general framework of non-flat models to denoise the chromaticity, here we have used the non-flat TV model studied by Chan and Shen [CS00b].

The idea of Chromaticity and Brightness approach is, from the RGB represented image, we first separate it to brightness and chromaticity components, then denoise those separately; using gray scale TV model for brightness, and non-flat TV model for chromaticity component. Finally, we assemble two component to get restored image I;

$$I(p) = u(p) \times f(p).$$

For the Brightness denoising model, from the given a noisy image I^0, let

$u^0 : \Omega \to R$ be its brightness component. The scalar TV restoration model applied to u^0 is

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

The associated Euler-Lagrange equation (in a formal level) of this cost functional is

$$-\nabla \cdot (\frac{\nabla u}{|\nabla u|}) + \lambda(u - u^0) = 0. \tag{2.3}$$

Since $|\nabla u|$ is in denominator, in order to avoid possible singularity, we condition it to $|\nabla u|_a = \sqrt{|\nabla u|^2 + a^2}$ for some small constant a in numerical implementation. This is equivalent to minimizing directly

$$\int_\Omega |\nabla u|_a \, dx \, dy + \frac{\lambda}{2} \int_\Omega (u - u^0)^2 \, dx \, dy.$$

The total variation as a cost functional legalizes the existence of edges or sharp jumps in the brightness component, which usually correspond to the real physical boundaries of objects [ROF92]. (As in the literature, if $a = 1$ (or after being linearly scaled), the first term measures the total area of the graph of $z = u(x, y)$.)

For Chromaticity Denoising model, let $f : \Omega \to S^2$ be a chromaticity feature. Assume f is smooth. This f lies on the unit sphere, so let $\partial_x f(p)$ and $\partial_y f(p)$ be two tangent vectors in the tangent space $T_{f(p)}S^2$ of S^2. Let $\| \cdot \|$ be the induced Riemannian norm of $T_{f(p)}S^2$ in R^3. Then the *total variation* model for chromaticity becomes [CS00b],

$$\mathcal{E}^{TV}(f) = \int_\Omega e(f; p) \, dp = \int_\Omega \sqrt{\|\partial_x f(p)\|^2 + \|\partial_y f(p)\|^2} dp, \quad dp = dx \, dy.$$

Moreover, since the unit sphere S^2 is embedded in R^3 and $f = (f_1, f_2, f_3) \in R^3$, we can express the energy with ordinary differentials of vectors in R^3,

$$e(f; p) = \|\nabla f\| = \sqrt{\|f_x(p)\|^2 + \|f_y(p)\|^2} = \sqrt{\|\nabla f_1\|^2 + \|\nabla f_2\|^2 + \|\nabla f_3\|^2}.$$

For the constrain term, let d be any appropriate distance function on S^2 [CS00b]. One can choose the *embedded distance*, the cord distance

$$d(f,g) := \|f-g\|_{R^3} = \sqrt{(f-g)^2}, \quad f,\ g \in S^2. \tag{2.4}$$

or the *geodesic distance*, the arc distance

$$d(f,g) = \arccos\ \langle \mathrm{f}, \mathrm{g} \rangle,$$

where $\langle f,g \rangle$ denotes the inner product in R^3. Here we have used embedded distance (2.4). The TV restoration model is

$$\min_f \ \mathcal{E}^{TV}(f) \quad \text{subject to} \quad \frac{1}{|\Omega|}\int_\Omega d^2(f^0,f)dp = \sigma^2$$

and the fitted TV energy is

$$\min_f \ \mathcal{E}^{TV}(f) \quad + \frac{\lambda}{2}\int_\Omega d^2(f^0,f)dp$$

The Euler-Lagrange equation for this unconstrained form becomes:

$$-\partial_x^*\left(\frac{\partial_x f}{\|\nabla f\|}\right) - \partial_y^*\left(\frac{\partial_y f}{\|\nabla f\|}\right) + \frac{\lambda}{2}\mathrm{grad}_f\ d^2(f^0,f) = 0,$$

where $\mathrm{grad}_f d^2(f^0,f)$ denotes the gradient vector of the scalar function $d^2(f^0,f)$ on S^2 and ∂_x^* and ∂_y^* the covariant derivatives acting on vector fields on the sphere. According to [CS00b], when the chromaticity sphere is naturally endowed with the restricted metric of R^3, the covariant derivatives are the ordinary derivatives (in R^3) followed by an orthogonal projection onto the sphere. As worked out step by step in [CS00b],

$$\partial_x^*\left(\frac{\partial_x f}{\|\nabla f\|}\right) + \partial_y^*\left(\frac{\partial_y f}{\|\nabla f\|}\right) = \nabla\cdot\left(\frac{\nabla f}{\|\nabla f\|}\right) + \|\nabla f\| f,$$

Furthermore, since we choose the distance (2.4) which is $d = \sqrt{(f-g)^2}$, as shown in [CS00b],

$$\frac{1}{2}\mathrm{grad_f d^2(f^0, f)} = \Pi_\mathrm{f}(\mathrm{f}-\mathrm{f}^0) = -\Pi_\mathrm{f}\mathrm{f}^0.$$

Where Π_f is the orthogonal projection from $T_f R^3$ onto the tangent plane $T_f S^2$:

$$\Pi_f g = g - \langle g, f \rangle f, \quad \text{for any vector} \quad g \in T_f R^3.$$

The projection operator appears due to the fact that we are working exclusively on the chromaticity sphere, thus "blind" to any changes occuring perpendicularly to the sphere. Combining the two equations, TV energy and fitting term,the restoration equation for the TV model becomes:

$$f_t \ \ (\text{or } \ 0) \ \ = \nabla \cdot \left(\frac{\nabla f}{\|\nabla f\|} \right) + \|\nabla f\| f + \lambda \Pi_f f^0. \tag{2.5}$$

We shall explain in Section 3 the numerical implementation and the digital version of the nonlinear restoration equations (2.3) and (2.5). In particular, our numerical scheme works directly on the steady equation, thanks to the digital TV filter recently proposed by Chan, Osher and Shen [COS01]. It avoids the complexity of keeping the nonlinear flow (2.5) strictly on the chromaticity sphere. Numerical methods for the direct flow are also discussed in [TSC99, TSC00].

2.2.2 TV for the HSV model

In the HSV color model, the hue H is a non-flat feature lives on the unit circle S^1, while the other two, S and V, are both linear features. As for brightness component in CB model, for saturation S and value V, we apply the scalar TV denoising model (2.3). For the circular feature H, we used equation for chromaticity (2.5) with the straightforward modification from S^2 to S^1:

$$H = (H_1, H_2) \in S^1 \subset R^2 \quad \text{and} \quad \|\nabla H\| = \sqrt{\|H_x\|^2 + \|H_y\|^2}.$$

However, our experiment (Fig. 2.6) shows that denoising the hue H and the saturation S separately does not produce satisfactory visual results. A possible

explanation for this deficiency is that to human visual perception the two components, H and S, are highly correlated. For example, for color 'pink', H should be around Red ($H = 0$) and S around 0. Therefore, a better approach is to restore the combination of H and S, which is also called chromaticity in the classical literature of color image processing.

To combine H and S components, let's consider a disk transformation. From the RGB color space (Fig. 2.1) for a fixed value V, the two other variables H and S span the iso-V surface which are three 2-D squares; Figure 2.2, surface $ABGF$, $BCDG$ and $GDCF$. This iso-V surface is piecewise smooth with a corner and three folding lines, thus we apply the *disk transform* to "straighten" this folded surface as follows. For each point on the iso-V surface with the hue and saturation (H, S), define a complex number

$$Z = S \times \exp(i2\pi H). \tag{2.6}$$

Since both H and S take all values in $[0, 1]$, the image of each iso-V surface is the unit disk in the complex plane, Figure 2.2. Under this disk transform, each iso-V & S lines (the 3-D hexagon) is mapped onto a circle centered at the origin.

The unit disk is a smooth and convex domain, thus much easier to work with. To restore th combination of H^0 and S^0 in the HSV color model, is equivalent to restore a complex-valued *scalar* feature Z^0. As a conclusion, from any given noisy image under the HSV model

$$(H^0(x, y), S^0(x, y), V^0(x, y)), \quad (x, y) \in \Omega,$$

we denoise two *scalar* components — the real-scalar-valued $V^0(x, y)$, and the complex-valued $Z^0(x, y)$ which is the color component; combination of H and S. We apply the scalar TV model (2.3) for both V and Z separately. The numerical

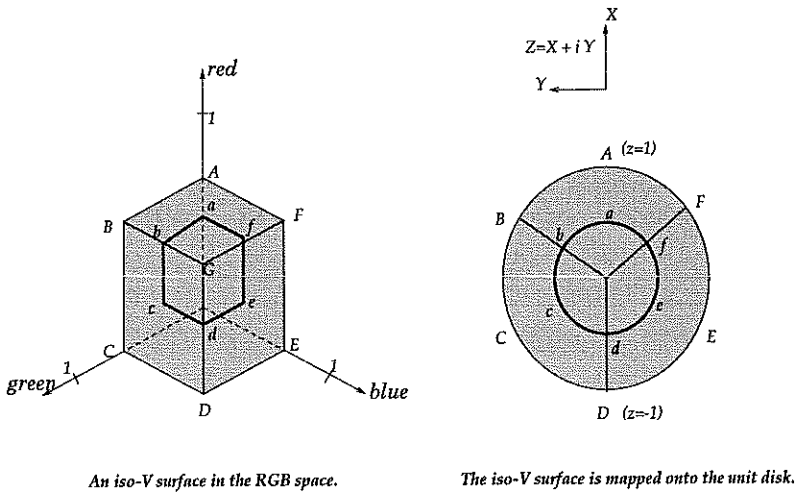


Figure 2.2: The disk transform of an iso-V surface: $Z = S\exp(i2\pi H)$.

results show the advantage of working with the combination of H and S than separating H and S.

2.3 Numerical Implementation

To digitally implement the nonlinear differential equations, we have applied the approach of *digital TV filtering* as proposed in [COS01]. The digital TV filter can be considered as the finite difference realization of the differential equations, but is simpler and more self-contained in the general framework of graph theory [COS01].

Let Ω be a discrete digital domain or a graph. Pixels in Ω are denoted by $\alpha, \beta, \cdots$. In the conventional rectangular setting, it is also denoted by $\alpha = (ij)$. Denote N_α the neighbors of the pixel α. For instance, in the rectangular setting,

$$N_\alpha = N_{(ij)} = \{\ (i+1, j),\ (i-1, j),\ (i, j-1),\ (i, j+1)\ \}.$$

There is much freedom in defining the neighbors, for example one may include $(i \pm 1, j \pm 1)$ in $N_{(ij)}$. Here we used above definition of neighbors and if $\beta \in N_\alpha$,

we also write as $\beta \sim \alpha$.

Let $u : \Omega \to R$ be the brightness component. We define its local variation to be $|\nabla_\alpha u| = \sqrt{\sum_{\beta\sim\alpha}(u_\beta - u_\alpha)^2}$, and its conditioned form to be $|\nabla_\alpha u|_a = \sqrt{\|\nabla_\alpha u\|^2 + a^2}$ for some small a (Sec 2.1).

The digital form of Euler-Lagrange equation (2.3) is a system of nonlinear algebraic equations, Chan, Osher, and Shen [COS01],

$$\sum_{\beta\sim\alpha} \omega_{\alpha\beta}(u)\ (u_\beta - u_\alpha) + \lambda(u - u^0) = 0, \quad \alpha \in \Omega.$$

where the weights are $\omega_{\alpha\beta}(u) = \frac{1}{|\nabla_\alpha u|_a} + \frac{1}{|\nabla_\beta u|_a}$. The techniques of linearization and iteration scheme lead to the so-called digital TV filter [COS01]. The digital TV filter $\mathcal{F}$ is nonlinear data dependent filter $\mathcal{F} : u \to v$. For the given noisy image u^0, at any pixel $\alpha \in \Omega$,

$$v_\alpha = \mathcal{F}_\alpha(u) = \sum_{\beta\sim\alpha} h_{\alpha\beta}(u)u_\beta + h_{\alpha\alpha}(u)u^0_\beta, \tag{2.7}$$

where the low-pass filter coefficients are

$$h_{\alpha\beta}(u) = \frac{\omega_{\alpha\beta}(u)}{\lambda + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u)}, \qquad h_{\alpha\alpha}(u) = \frac{\lambda}{\lambda + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u)}.$$

Apply this digital TV filter in an iterative fashion. To restore and denoise the brightness component u^0, one starts with a initial guess $u^{(0)}$ (for convenience $u^{(0)} = u^0$), and the TV filter generates $u^{(n)} = \mathcal{F}(u^{n-1})$ for $n = 1, 2, \cdots$. $u^{(n)}$ converges to the optimal restoration u.

For the Numerical implementation for Chromaticity component, we have used digitized version of TV minimizing Model. Let $f^0 : \Omega \to S^2$ be the noisy chromaticity. Taking the embedded distance $d(f, g) = \sqrt{(f - g)^2}$ for any $f, g \in S^2$, we define the local variation at a pixel α of a chromaticity feature $f : \Omega \to S^2$ to be:

$$e(f; \alpha) = [\sum_{\beta\sim\alpha} d^2(f_\beta, f_\alpha)]^{\frac{1}{2}}.$$

The digitized total variation with the fitting constraint becomes [COS01]

$$\mathcal{E}^{TV}(f,\lambda) = \sum_{\alpha\in\Omega_n} e(f;\alpha) + \lambda \sum_{\alpha\in\Omega_n} \frac{1}{2} d^2(f_o, f),$$

which provides the digital cost function for optimization. Then the digital Euler-Lagrange equation is shown to be [CS00b]:

$$0 = \sum_{\beta\sim\alpha} \Pi_{f_\alpha}(f_\beta) \left(\frac{1}{e(f;\alpha)} + \frac{1}{e(f;\beta)} \right) + \lambda \Pi_{f_\alpha}(f^0_\alpha), \quad \alpha \in \Omega. \tag{2.8}$$

Where Π_f is the orthogonal projection defined in the previous section. This Π_f is needed to insure the chromaticity to lie on the unite sphere. By setting $\omega_{\alpha\beta}(f) = \frac{1}{e(f;\alpha)} + \frac{1}{e(f;\beta)}$, the restoration equation (2.8) can be rewritten as

$$\Pi_{f_\alpha}(\sum_{\beta\sim\alpha} \omega_{\alpha\beta} f_\beta + \lambda f^0_\alpha) = 0, \quad \alpha \in \Omega.$$

Chan and Shen [CS00b] showed that this equation on the unknown optimal restoration $f : \Omega \to S^2$ can be similarly solved by the digital TV filter $\mathcal{F}$. Since the feature lives on the unit sphere, $g = \mathcal{F}(f)$ now needs an extra step of normalization (projection to sphere) :

$$\begin{aligned} \tilde{g}_\alpha &= \textstyle\sum_{\beta\sim\alpha} h_{\alpha\beta} f_\beta + h_{\alpha\alpha} f^0_\alpha; \\ g_\alpha &= \tilde{g}_\alpha / \|\tilde{g}_\alpha\|. \end{aligned} \tag{2.9}$$

One starts the iterative filtering process with an initial guess $f^{(0)}$, then generate $f^{(n)} = \mathcal{F}(f^{(n-1)})$. Chan and Shen [CS00b] showed that the limit of $f^{(n)}$ indeed solves the restoration equation (2.8).

In the HSV color system, for denoising H, S and V separately, we have used scalar TV filter (2.7) for S and V, and for hue H, we used simple modification of Chromaticity TV filter (2.9) from S^2 to S^1. For denoising Z and V, where Z is a combination of H and S, we used scalar TV filter (2.7) for both the real scalar function V^0, and the complex scalar function Z^0. Since the both unit interval and

unit disk are convex domains, the *maximum principle* of the TV filter [COS01] guarantees that the restored V takes values in $[0, 1]$, and Z on the unit disk. Thus, the restored H and S can be well reconstructed from Z.

2.4 Numerical Experiments and Comparison Conclusion

This section summarizes the performance of the TV models based on the CB and HSV color models.

TV restoration based on the CB model: In Fig. 2.3, we demonstrate the result of TV restoration applied to the chromaticity-brightness representation. The top image is the original noisy image, the middle panel is the image with only chromaticity restored and at the bottom, the image with both chromaticity and brightness restored. After denoising chromaticity, the visible noisy red and green dots have been swept out and the eyes and dark lines resume their original black color. In the final result, the nose and lips become smoothly red as they should be.

Comparison of TV's on the CB and linear models : In Fig. 2.4, we compare the restoration results by the CB-based TV and linear TV; channel-by-channel TV and vectorial TV. The images in the first column (left) are the original image and the two right columns are zoom-in to show the details of the first column. Compared to the channel-by-channel TV and vectorial TV, the chromaticity-brightness TV restoration seems to give better color control. Inside the white arrow, while channel-by-channel TV and vectorial TV have some red, green, and different color dots left, the chromaticity-brightness TV is closer to white and closer to human color perception.

Fig. 2.5 shows another example for the comparison between the linear TV's

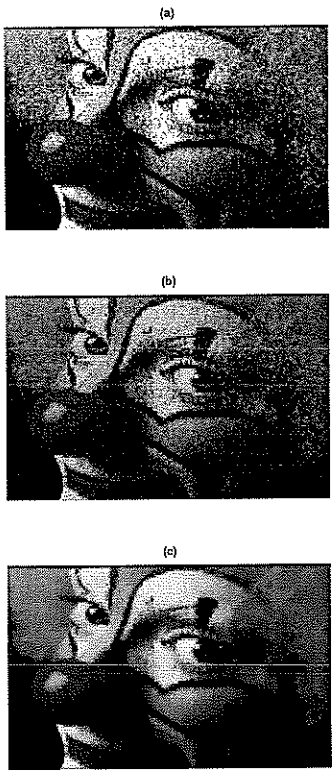


Figure 2.3: Spherical TV on the chromaticity component and scalar TV on the brightness component. (a) is original noisy image. In (b), only the chromaticity is restored, while in (c), both the chromaticity and brightness components are restored.

and the CB-based TV. The first column is the original image and second column is zoom-in to show the letters at the bottom right corner of first column image. We see that the the designer's name has been best restored by the chromaticity-brightness TV, again showing better color control. The channel-by-channel TV and vectorial TV have blurred the letters, mostly due to their inefficiency to deal with the chromaticity component. The third column of Fig. 2.5 is an example showing that the CB model is better at recovering details. While channel-by-channel TV and vectorial TV lose details to recover color, the CB model can

control both by denoising separately.

TV restoration directly based on the HSV model : In Fig. 2.6, we show the restoration result from the direct application of the TV models on the three components of the HSV representation. We are denoising H with non-flat (S^1) TV model, S and V with scalar TV models. The zoom-in image shows the unsatisfactory behavior of such an approach of the approach. The underlying reason, we believe, is the high correlation between the hue component H and the saturation S for human perception.

TV restoration on HSV and its disk transform : The preceding sections and the previous numerical example lead us to considering Z, the combination of H and S via the disk transform (2.6). In Fig. 2.7, we display the restoration result from the scalar TV on both the real function V and the complex function Z. From the zoom-in image, it is clear that such combination is much closer to human perception and results in better color restoration then denoising H, S and V separately.

TV on CB and HSV with the disk transform : The disk transform Z in the HSV representation encodes both the hue and saturation, and thus is similar to the chromaticity information in the CB representation. Meanwhile, the "value" component V apparently plays the same role as the brightness component in the CB model. As a result, in Fig. 2.8 and Fig. 2.9 the performance of these two approaches are very close. However, CB model result have sharper and clearer black recovery.

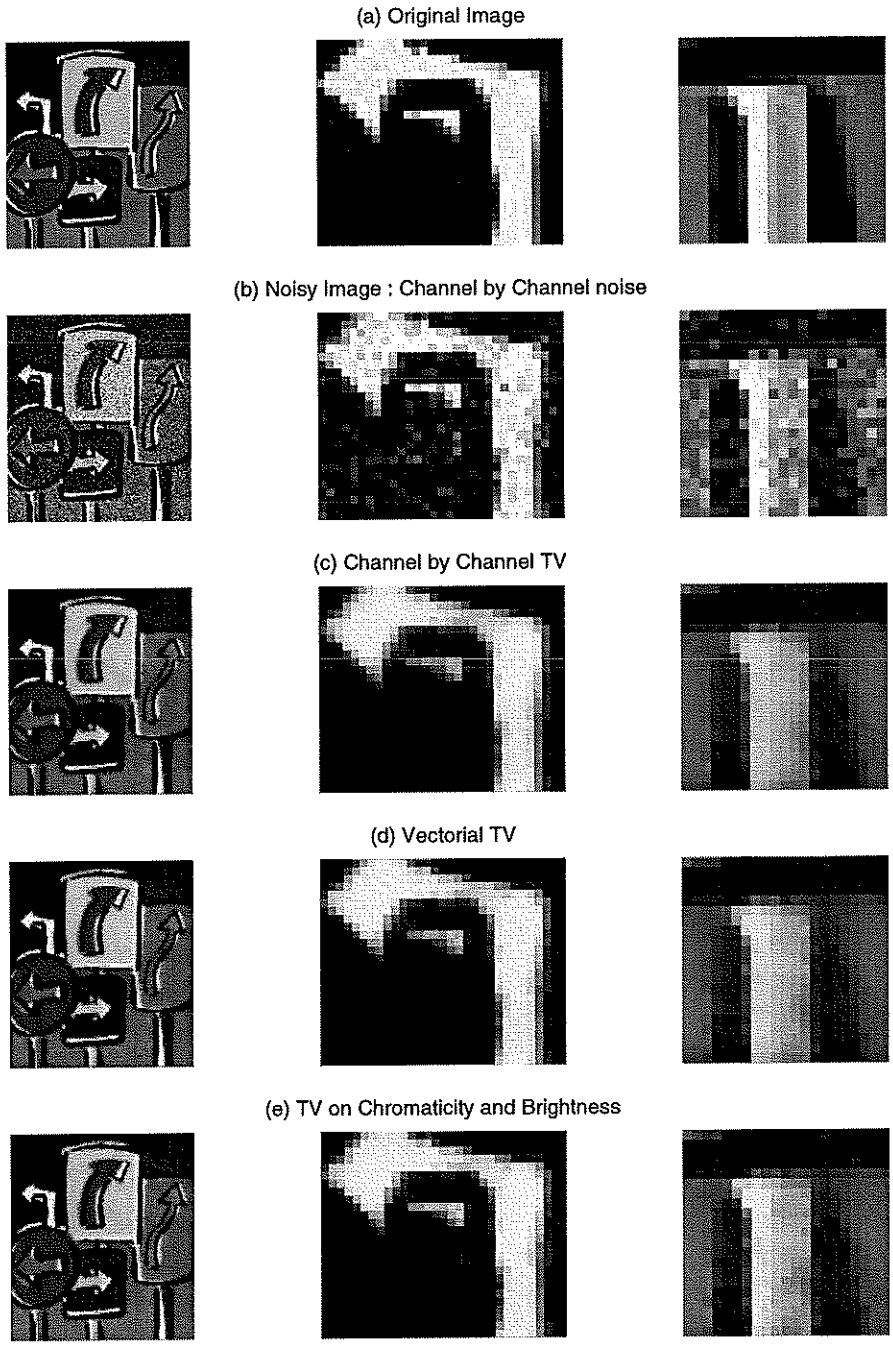


Figure 2.4: Comparison of the CB based TV and linear TV's (I). The first column is original image and right two columns are zoom-in of the first column to show details. CB leads to better color control.

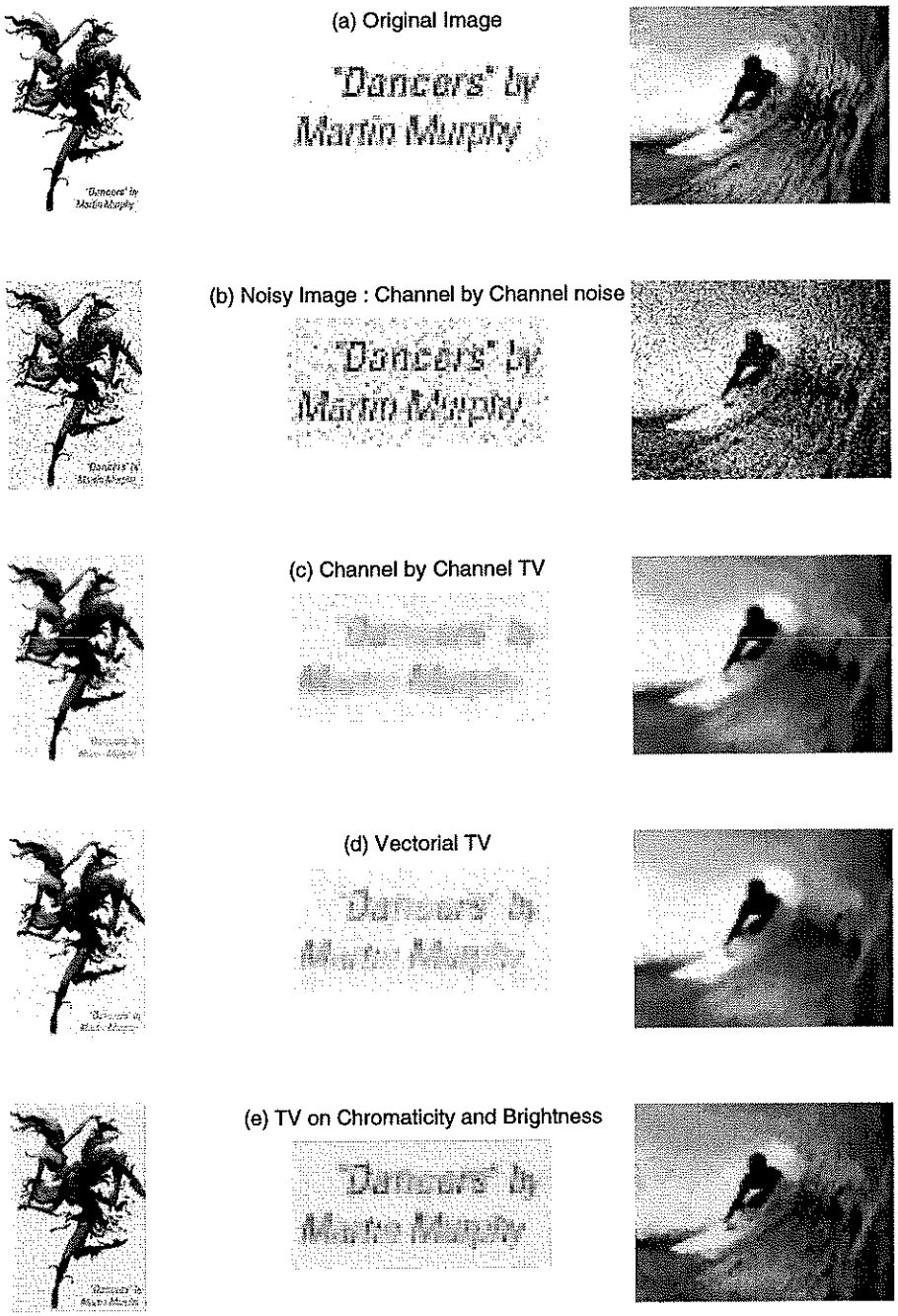


Figure 2.5: Comparison between the CB based TV and linear TV's (II) and (III). (II) The middle column is the zoom-in of bottom right corner of first column. (III) The third column is another image. CB leads to better color control and better detail recovery.

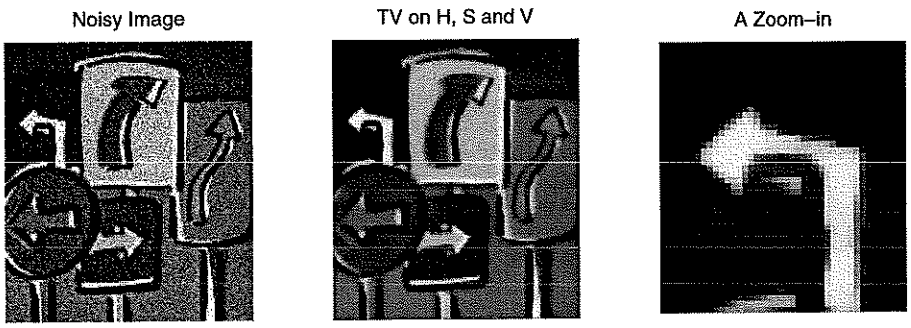


Figure 2.6: Circular TV on H, and scalar TV's on S and V. The unsatisfactory performance shows the inappropriateness of treating the H and S components separately.

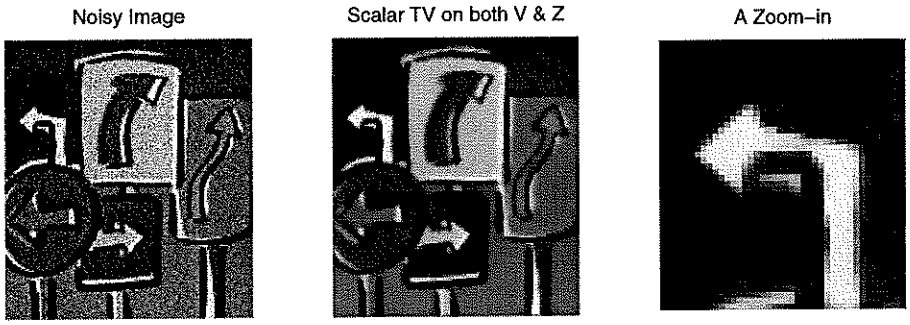


Figure 2.7: Scalar TV's on both the real function V and the complex function Z, where Z is the disk transform of H and S.

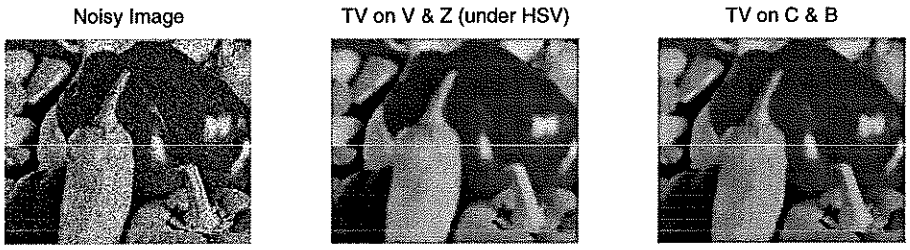


Figure 2.8: TV's based on the CB and HSV (using V and Z) color models have the similar performance.

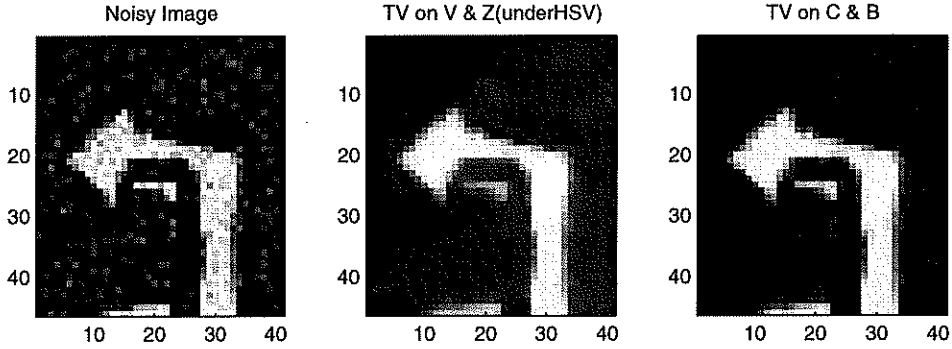


Figure 2.9: TV's based on the CB and HSV (using V and Z), CB model seems to have better result.

CHAPTER 3

Color Image Inpainting

3.1 Introduction

The color image denoising can easily be extended to color image inpainting problems. Inpainting problem arise from restoring the image with a crack which needed to be filed in. More detail about inpainting will follow in Chapter 4.

Inspired by Bertalmio, Sapiro, Caselles and Ballester [BSC00], Chan and Shen proposed TV inpainting (Total Variation minimizing inpainting) method. This TV inpainting is closely related to Total Variation denoising suggested by Rudin, Osher and Fatemi [ROF92]. Let D be the region with missing information, this is called *Inpainting Domain.* Then, the TV inpainting model is to find image u which minimizes Total Variation on the domain Ω while fitting the information only from outside the inpainting domain, $\Omega \backslash D$,

$$J_\lambda(u) = \int_\Omega |\nabla u| dxdy + \frac{\lambda}{2} \int_{\Omega \backslash D} |u - u_o|^2 dxdy.$$

Here, u_o is original image, ∇u is the gradient and constraint is only on $\Omega \backslash D$ with λ as the Lagrange relaxation parameter.

In this chapter, we extend this TV inpainting to RGB color space. From 3-D RGB color vector, we use channel by channel inpainting as well as vectorial inpainting. In following sections, we present the model and numerical results.

3.2 The Model and Numerical Implementation

For a given image $u : \Omega \to R^3_+ = \{(r,g,b) : r,g,b > 0\}$, in channel-by-channel inpainting, each channel u_i is inpainted separately using TV inpainting [CS01a],

$$\min_{u_i} = \int_\Omega |\nabla u_i| dxdy + \frac{\lambda}{2} \int_{\Omega \backslash D} |u_i - u_i^o|^2 dxdy.$$

For the vectorial inpainting, the cost functional is on the vectorial function $u = (u_1, u_2, u_3)$,

$$\min_{u} = \int_\Omega \|\nabla u\|_{3D} dxdy + \frac{\lambda}{2} \int_{\Omega \backslash D} \|u - u^o\|^2_{3D} dxdy$$

where $\|\cdot\|_{3D}$ is for vector-valued functions, typically in applications, $\|u\|_{3D} = \sqrt{|u_1|^2 + |u_2|^2 + |u_3|^2}$.

For numerical implementation for channel by channel inpainting, we use *digital TV filtering* as proposed in [COS01]. Similar to Chapter 2, We define local variation to be $|\nabla_\alpha u_i| = \sqrt{\sum_{\beta \in N_\alpha} (u_{i\beta} - u_{i\alpha})^2 + a^2}$, where a is a small constant, and $N_\alpha = N_{(ij)} = \{ (i+1,j), (i-1,j), (i,j-1), (i,j+1) \}$. The digital form of Euler-Lagrange equation for inpainting becomes,

$$\sum_{\beta \in N_\alpha} \omega_{\alpha\beta}(u_i) \, (u_{i\beta} - u_{i\alpha}) + \lambda_D (u_i - u_i^0) = 0, \quad \alpha \in \Omega.$$

where the weights are $\omega_{\alpha\beta}(u_i) = \frac{1}{|\nabla_\alpha u_i|} + \frac{1}{|\nabla_\beta u_i|}$, and λ_D is,

$$\lambda_D = \begin{cases} \lambda & z \in \Omega \backslash D \\ 0 & z \in D \end{cases}$$

Then, the digital TV filter $\mathcal{F}$ is $\mathcal{F} : u_i \to v_i$,

$$v_{i\alpha} = \mathcal{F}_\alpha(u_i) = \sum_{\beta \sim \alpha} h_{\alpha\beta}(u_i) u_{i\beta} + h_{\alpha\alpha}(u_i) u^0_{i\beta}, \tag{3.1}$$

similar to denoising filter (2.7), however, the low-pass filter coefficients with λ_D are,

$$h_{\alpha\beta}(u_i) = \frac{\omega_{\alpha\beta}(u_i)}{\lambda_D + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u_i)}, \qquad h_{\alpha\alpha}(u_i) = \frac{\lambda_D}{\lambda_D + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u_i)}.$$

The digital TV filter is applied in iterative fashion.

For Vectorial inpainting method, we also use digital TV filter. The digital form of Euler-Lagrange equation is,

$$\sum_{\beta\in N_\alpha} \omega_{\alpha\beta}(u)\ (u_\beta - u_\alpha) + \lambda_D(u - u^0) = 0, \quad \alpha \in \Omega.$$

where the weights are $\omega_{\alpha\beta}(u) = \frac{1}{\|\nabla_\alpha u\|_{3D}} + \frac{1}{\|\nabla_\beta u\|_{3D}}$, and

$$\|\nabla_\alpha u\|_{3D} = \sqrt{\sum_{\beta\in N_\alpha}(u_{1\beta} - u_{1\alpha})^2 + \sum_{\beta\in N_\alpha}(u_{2\beta} - u_{2\alpha})^2 + \sum_{\beta\in N_\alpha}(u_{3\beta} - u_{3\alpha})^2 + a^2}.$$

Then, the digital TV filter $\mathcal{F}$ is $\mathcal{F} : u \to v$,

$$v_\alpha = \mathcal{F}_\alpha(u) = \sum_{\beta\sim\alpha} h_{\alpha\beta}(u)u_\beta + h_{\alpha\alpha}(u)u^0_\beta,$$

with low-pass filter coefficients,

$$h_{\alpha\beta}(u) = \frac{\omega_{\alpha\beta}(u)}{\lambda_D + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u)}, \qquad h_{\alpha\alpha}(u) = \frac{\lambda_D}{\lambda_D + \sum_{\gamma\sim\alpha}\omega_{\alpha\gamma}(u)}.$$

3.3 Numerical Results

We present a couple of results demonstrating the result of channel by channel inpainting and vectorial inpainting. Fig. 3.1 is an image of Kyung Bok Palace in Seoul, Korea. Both channel by channel inpainting and vectorial inpainting show successful results. The yellow letters are successfully removed while leaving the complicated texture. Fig. 3.2 is an image of White house and the pink letter are the inpainting domain. The results looks better for the vectorial case, since vectorial model is closer to chromaticity control than channel by channel inpainting.

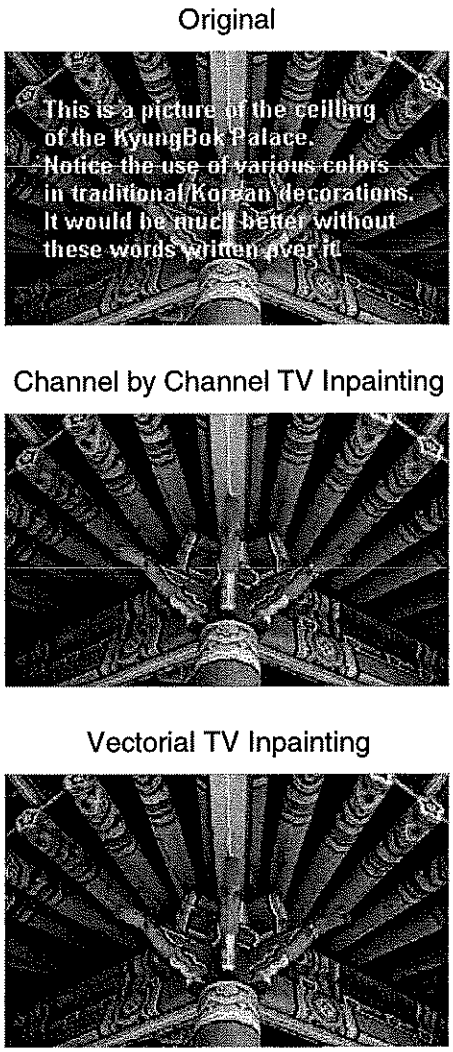


Figure 3.1: In the top image, the yellow letters are the inpainting domain. The middle image is channel by channel inpainting and bottom image is the vectorial inpainting. Both, channel by channel inpainting and vectorial inpainting show successful results.

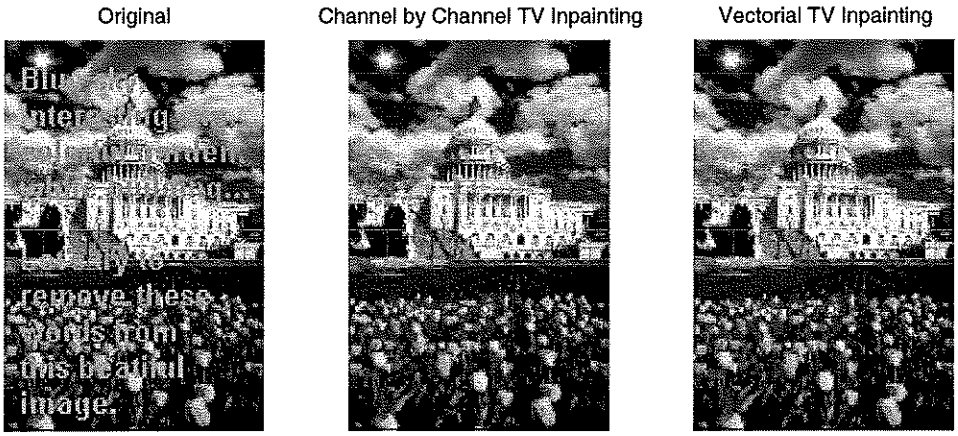


Figure 3.2: In the top image, the pink letters are the inpainting domain. The middle image is channel by channel inpainting and bottom image is the vectorial inpainting. Vectorial inpainting has better result.

CHAPTER 4

Euler's Elastica and Curvature Based Inpaintings

4.1 Introduction

When you have damaged image, e.g. scratched or some portion is missing, we try to refill the gaps to reconstruct the better image. This is called *retouching* or *inpainting*. In Fig. 4.1, couple of example of inpainting is shown. From many subjects in image processing, this is a part of image restoration, reconstruction problems which is also related to disocclusion problem from vision analysis.

The Digital inpainting, instead of manual re-touchings, was proposed by Bertalmio,Sapiro, Caselles and Ballester [BSC00]. In their paper, they use transportation mechanism which is from the boundary of the missing region, it extends the information from boundary to inside, step by step.

Another approach based on solving PDEs from vision analysis is Masnou and Morel [MM98]'s level line based disocclusion. They use dynamical programming to to interpolate the level lines from boundary level lines. Closely related approach based on using PDE from Inpainting point of view, is TV minimizing inpainting suggested by Chan and Shen [CS01a]. This Total Variation minimizing inpainting is in spirit very similar to the classical restoration model of Rudin, Osher and Fatemi [ROF92] and also reduces good results for inpainting problems.

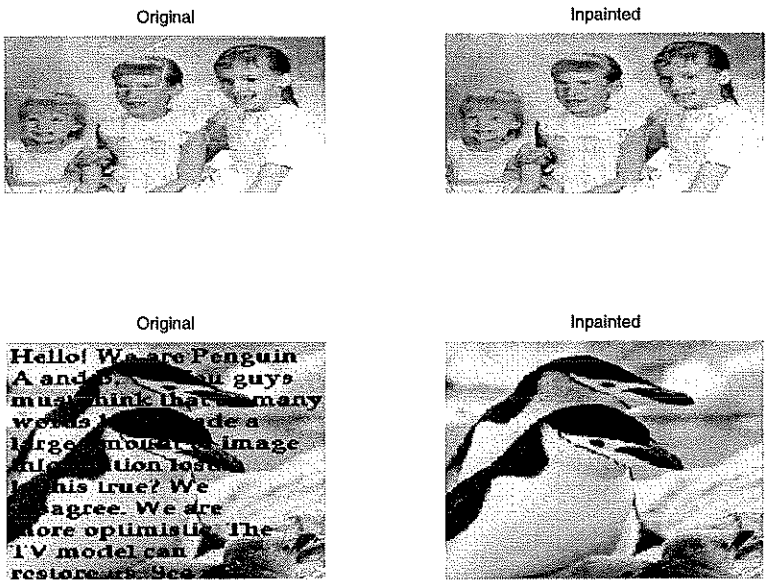


Figure 4.1: Inpainting problem examples. In the first row, the cracks in old photo is inpainted and in the second row, the black letters are inpainted.

The TV inpainting model (4.1) is to find image u which minimizes Total Variation on the domain $E \cup D$. As shown in Fig. 4.2, D is the inpainting domain with piecewise smooth boundary, and E be extended outside domain of D where only the boundary of D lies in.

$$J_\lambda(u) = \int_{E \cup D} |\nabla u| dxdy + \frac{\lambda}{2} \int_E |u - u_o|^2 dxdy \tag{4.1}$$

where u_o is original image and ∇u is the gradient and constraint is only on E with λ as the Lagrange relaxation parameter. The Euler-Lagrange equation is,

$$-\nabla \cdot (\frac{\nabla u}{|\nabla u|}) + \lambda_E (u - u_o) = 0 \quad where \quad \lambda_E = \begin{cases} \lambda & z \in E \\ 0 & z \in D \end{cases}$$

This model is very close to Total Variation Minimizing denoising model except for λ_E which only enforces the information from E, outside D, rather then from

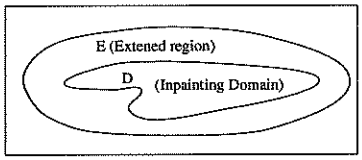


Figure 4.2: Inpainting Domain D

the hold image $E \cup D$. This TV inpainting is very useful when we have small inpainting domain D or we want to recover a straight line. However, it will get a straight line or disconnected results, even when the missing region looks connected or curved. As in Fig. 4.3, the gray noisy portion is the inpainting domain D, it is reasonable to think the black line is connected. However, TV inpainting will result in disconnected result (b). This is because at the boundary of inpainting domain D the black line is much narrower then the white background, therefore, TV minimization will restore the inpainting region as white, the same color as background.

To get connected result, Chan and Shen [CS01c] suggested Curvature Driven Diffusions (CDD). By having the diffusion coefficient which depends on the curvature of the isophotes, this can give connected results.

$$-\nabla \cdot \left(\frac{G(\kappa, z)}{|\nabla u|}\nabla u\right) + \lambda_E(u - u_o) = 0 \quad where \quad G(\kappa, z) = \begin{cases} 1 & z \in E \\ g(|\kappa|) & z \in D \end{cases}$$

above $g(s)$ is appropriate function that reduces the large curvature and stabilized small curvatures inside the inpainting domain, for example one can choose $g(s) = s^\alpha$ for some $\alpha \geq 1$.

In this chapter, we are following Masnou and Morel's [MM98] proposal of variational model to improve the TV inpainting. By adding the curvature term to the regularization functional, it will try to avoid a sharp edge (kink) which has very big curvature, and get connected result for Fig. 4.3. Also this allows to

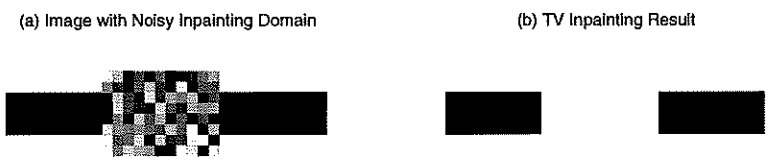


Figure 4.3: Disconnected result.

get curved result instead of straight line. However, with curvature term, Euler-Lagrange equation becomes forth order highly nonlinear equation, so here we use Normal and Tangent vectors to better represent this Euler-Lagrange equation. Using Normal and Tangent vectors representation, this result in requiring only simple Numerical Schemes. In following sections, we'll present the Mathematical Model with detailed derivation of Euler-Lagrange equation, show numerical implementations and present some results.

4.2 Mathematical Model

In [MM98], Masnou and Morel proposed the disocclusion functional,

$$\int_{\Omega} |\mathcal{D}v|(1+|\text{curv } v|^p)dx, \quad p>1 \qquad v=u \ \ outside \ \ D$$

where Ω is the entire image domain, and D the occluded image. Using this functional with TV inpainting, we use regularization functional,

$$J_\lambda(u) = \int_{E\cup D} |\nabla u|(a+b\kappa^2)dxdy + \frac{\lambda}{2}\int_E |u-u_o|^2 dxdy \tag{4.2}$$

where $E \cup D$ is the extended inpainting domain, ∇u is the gradient, constraint is only on E with λ. Here κ is curvature, $\kappa = \nabla \cdot (\frac{\nabla u}{|\nabla u|})$. a and b are constants for TV term and curvature term respectively.

A curve γ is said to be *Euler's Elastica* if it is the equilibrium of the elasticity

energy,

$$\int_\gamma (a + b\kappa^2) ds.$$

Here, ds is the arc length element, $\kappa(s)$ the scalar curvature, and a, b are two positive constant weights. Euler obtained the energy while studying the steady shape of a thin and torsion free rod under external forces [Lov27].

A naive derivation of Euler-Lagrange equation, using $\frac{\partial f}{\partial u} = \frac{\partial}{\partial x}\frac{\partial f}{\partial u_x} + \frac{\partial}{\partial y}\frac{\partial f}{\partial u_y}$ is,

$$(\nabla \cdot \frac{\nabla u}{|\nabla u|})\{a + b\kappa^2 + 2b|\nabla u|(\star)\} - \lambda_E(u - u^o) = 0 \tag{4.3}$$

where $(\star)$ is,

$$\begin{aligned}(\star) \quad &= \Delta(\tfrac{1}{|\nabla u|}) - \{\tfrac{\partial^2}{\partial x^2}(\tfrac{u_x^2}{|\nabla u|^3}) + \tfrac{\partial^2}{\partial x \partial y}(\tfrac{2u_y u_x}{|\nabla u|^3}) + \tfrac{\partial^2}{\partial y^2}(\tfrac{u_y^2}{|\nabla u|^3})\} \\ &= \tfrac{\partial^2}{\partial x^2}(\tfrac{u_y^2}{|\nabla u|^3}) + \tfrac{\partial^2}{\partial x \partial y}(\tfrac{-2u_y u_x}{|\nabla u|^3}) + \tfrac{\partial^2}{\partial y^2}(\tfrac{u_x^2}{|\nabla u|^3}).\end{aligned}$$

The curvature κ is second order feature of v, which leads to fourth order highly nonlinear Euler Lagrange equation. This is not trivial to drive and the resulting Euler-Lagrange equation is very complicated and highly nonlinear. This derivation is also done independently by Masnou [Mas00].

Here, we analysis this better by using normal and tangent vectors. This will reduced to simple Euler-Lagrange equation representation only with ∇ and $\nabla\cdot$ avoiding second or higher order derivatives. From equation (4.2), let

$$R[u] = \int_{E \cup D} |\nabla u|(a + b\kappa^2) dxdy \quad \textit{and} \quad F[u] = \frac{\lambda}{2}\int_E \|u - u_o\|^2 dxdy$$

Denote $< f >= \int_\Omega f dxdy$ and compute its variation,

$$\begin{aligned}\delta R \; &=< \delta\{|\nabla u|(a + b\kappa^2)\} >=< a\,\delta(|\nabla u|) + b\,\delta(\kappa^2|\nabla u|) > \\ &= a < \delta|\nabla u| > + b < \kappa^2 \delta|\nabla u| > + 2b < \kappa|\nabla u|\delta\kappa > \\ &=< (a + b\kappa^2)\,\delta|\nabla u| > + 2b < \kappa|\nabla u|\,\delta\kappa > \\ &=< (a + b\kappa^2)\tfrac{\nabla u}{|\nabla u|}\,\delta \cdot \nabla u > + 2b < \kappa|\nabla u|\,\delta\kappa > \\ &= - < \nabla \cdot [(a + b\kappa^2)\vec{\eta}]\,\delta u > + 2b < \kappa|\nabla u|\,\delta\kappa >\end{aligned}$$

here $\vec{\eta} = \frac{\nabla u}{|\nabla u|} = (\frac{u_x}{|\nabla u|}, \frac{u_y}{|\nabla u|})$. To work out the second term, first compute the variation of κ i.e. $\delta\kappa$. Using $\kappa = \nabla \cdot \vec{n}$ and $\vec{\eta} = \frac{\nabla u}{|\nabla u|}$,

$$\begin{aligned}
\delta\kappa \quad &= \delta(\nabla \cdot \tfrac{\nabla u}{|\nabla u|}) = \nabla \cdot [\tfrac{1}{|\nabla u|}\ \nabla(\delta u) + \nabla u\ \delta(\tfrac{1}{|\nabla u|})] \\
&= \nabla \cdot [\tfrac{1}{|\nabla u|}\nabla(\delta u) - (\nabla u\ \tfrac{\nabla u}{|\nabla u|^3})\nabla(\delta u)] \\
&= \nabla \cdot [\tfrac{1}{|\nabla u|}\{\mathrm{I} - \vec{\eta} \otimes \vec{\eta}\}\nabla(\delta u)]
\end{aligned}$$

$$\begin{aligned}
< \kappa|\nabla u|\ \delta\kappa > \quad &=< \kappa|\nabla u|\ \nabla \cdot [\tfrac{1}{|\nabla u|}\{\mathrm{I} - \vec{\eta} \otimes \vec{\eta}\}\nabla(\delta u)] > \\
&=< -\nabla(\kappa|\nabla u|)\ [\tfrac{1}{|\nabla u|}\{\mathrm{I} - \vec{\eta} \otimes \vec{\eta}\}\nabla(\delta u)] > \\
&=< -\{\mathrm{I} - \vec{\eta} \otimes \vec{\eta}\}\{\tfrac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\}\nabla(\delta u) > \\
&=< \nabla \cdot [(\mathrm{I} - \vec{\eta} \otimes \vec{\eta})\{\tfrac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\}]\ (\delta u) >
\end{aligned}$$

Combining all above for δR, we have

$$\begin{aligned}
\tfrac{\partial R}{\partial u} \quad &= -\nabla \cdot [(a + b\kappa^2)\vec{\eta} - 2b(\mathrm{I} - \vec{\eta} \otimes \vec{\eta})\tfrac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)] \\
&= -\nabla \cdot [(a + b\kappa^2)\vec{\eta} - \tfrac{2b}{|\nabla u|}\tfrac{\partial(\kappa|\nabla u|)}{\partial \vec{\tau}}\vec{\tau}] \\
&= -\nabla \cdot \vec{V} \\
\tfrac{\partial F}{\partial u} \quad &= \lambda_E(u - u_o)
\end{aligned}$$

Here $\vec{\eta} = (\frac{u_x}{|\nabla u|}, \frac{u_y}{|\nabla u|})$ is normal vector and $\vec{\tau} = (\frac{-u_y}{|\nabla u|}, \frac{u_x}{|\nabla u|})$ is tangent vector. For 2D image domain, Fig. 4.4, $\mathrm{I} = \vec{\eta} \otimes \vec{\eta} + \vec{\tau} \otimes \vec{\tau}$ and using the fact $\vec{\tau} \cdot \nabla = \frac{\partial}{\partial \vec{\tau}}$, we get above result. Here $\vec{V}$ is represented by two orthonormal vectors $\vec{\eta}$ and $\vec{\tau}$.

Thus Euler-Lagrange equation becomes,

$$\begin{aligned}
\tfrac{\partial J}{\partial u} \quad &= \tfrac{\partial R}{\partial u} + \tfrac{\partial F}{\partial u} = -\nabla \cdot \vec{V} + \lambda_E(u - u_o) \\
\vec{V} \quad &= (a + b\kappa^2)\vec{\eta} - \tfrac{2b}{|\nabla u|}\tfrac{\partial(\kappa|\nabla u|)}{\partial \vec{\tau}}\vec{\tau}
\end{aligned} \tag{4.4}$$

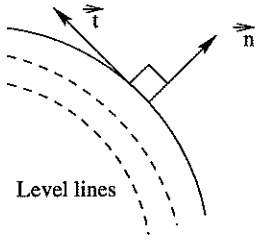


Figure 4.4: Vector field of Normal and Tangent vector

This equation is better to work with then equation(4.3), since it only has $\nabla\cdot$ and ∇s instead of second or higher order derivative terms. This enable us to use only simple numerical schemes (next section).

Theorem The vector field $\vec{V}$ is morphologically invariant. (i.e. $\vec{V}$ does not depend on relative intensity.)

Proof For a morphological transform g:

$$u \rightarrow g(u) \qquad where \quad g(0) = 0, \quad g(1) = 1$$
$$g'(u) > 0 \quad u \in (0,1)$$

We have to show, $\vec{V}(g(u)) = \vec{V}(u)$.

In (4.4), the first term is invariant since both $\vec{\eta}$ and $\kappa = \nabla \cdot \vec{\eta}$ are morphologically invariant. For example,

$$\vec{\eta} = \frac{\nabla g(u)}{|\nabla g(u)|} = \frac{g'(u)\nabla u}{|g'(u)||\nabla u|} = \frac{\nabla u}{|\nabla u|}$$

For the second term,

$$\frac{1}{|\nabla g(u)|}\frac{\partial(\kappa|\nabla g(u)|)}{\partial\vec{\tau}}\vec{\tau} = \frac{1}{g'(u)|\nabla u|}\frac{\partial(g'(u)\kappa|\nabla u|)}{\partial\vec{\tau}}\vec{\tau}$$

$$= \frac{g'(u)}{g'(u)|\nabla u|}\frac{\partial(\kappa|\nabla u|)}{\partial\vec{\tau}}\vec{\tau} = \frac{1}{|\nabla u|}\frac{\partial(\kappa|\nabla u|)}{\partial\vec{\tau}}\vec{\tau}$$

since in the tangent direction $\vec{\tau}$, u, $g(u)$ and $g'(u)$ are all constants $\square$

4.3 Numerical Implementation

For the numerical implementation, let $u : \Omega_n \rightarrow R$ be digital gray scale image. From the equation (4.4), we used time marching,

$$\begin{aligned} \frac{\partial u}{\partial t} &= \nabla \cdot \vec{V} - \lambda_E(u - u_o) \;\; where \;\; \lambda_E = \begin{cases} \lambda & z \in E \\ 0 & z \in D \end{cases} \\ \vec{V} &= (a + b\kappa^2)\vec{\eta} - \frac{2b}{|\nabla u|}\frac{\partial(\kappa|\nabla u|)}{\partial \vec{\tau}}\vec{\tau} \end{aligned} \tag{4.5}$$

This Euler-Lagrange equation is nonlinear and ill-conditioned, and both terms in $\vec{V}$ have $|\nabla u|$ in the denominator, therefore, we multiply the magnitude of the gradient of u, suggested by Marquina and Osher [MO99]. This allows the level contours to move faster and the gradient regularizes the mean curvature term.

$$\begin{aligned} \frac{u^{n+1}_{(i,j)} - u^n_{(i,j)}}{dt} &= |\nabla u^n_{(i,j)}|\, F(u^n_{(i,j)}) \; - \; |\nabla u^n_{(i,j)}|\lambda_E(u^n_{(i,j)} - u^o_{(i,j)}) \\ where \qquad & F(u^n_{(i,j)}) = \nabla \cdot \vec{V}^n_{(i,j)} \end{aligned}$$

For the $|\nabla u^n_{(i,j)}|$ in front of $F(u^n_{(i,j)})$,

$$|\nabla u^n_{(i,j)}| = \sqrt{\frac{u^n_{(i+1,j)} - u^n_{(i-1,j)}}{2} + \frac{u^n_{(i,j+1)} - u^n_{(i,j-1)}}{2}} \tag{4.6}$$

For the $|\nabla u^n_{(i,j)}|$ in front of $\lambda_E(u^n_{(i,j)} - u^o_{(i,j)})$,

$$|\nabla u^n_{(i,j)}| = \sqrt{(upwind\ d_x u)^2 + (upwind\ d_y u)^2}$$

$$upwind\ d_x u = \begin{cases} \frac{u^n_{(i,j)} - u^n_{(i-1,j)}}{2} & if \;\; (\frac{u^n_{(i+1,j)} - u^n_{(i-1,j)}}{2})(u^n_{(i,j)} - u^o_{(i,j)}) > 0 \\ \frac{u^n_{(i+1,j)} - u^n_{(i,j)}}{2} & if \;\; (\frac{u^n_{(i+1,j)} - u^n_{(i-1,j)}}{2})(u^n_{(i,j)} - u^o_{(i,j)}) < 0 \end{cases}$$

Similar for $upwind\ d_y u$, this work is from [MO99].

This Euler-Lagrange equation (4.4) has advantage of only having $\nabla\cdot$ and ∇ and for which we used half steps. For $F(u^n_{(i,j)}) = \nabla \cdot \vec{V}^n_{(i,j)}$, this V is a vector i.e.

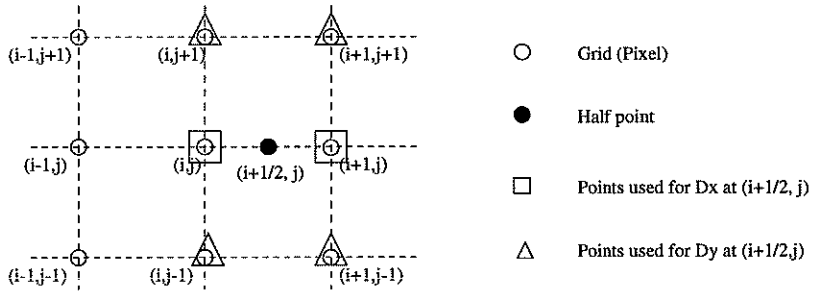


Figure 4.5: Grid for finite different scheme

$V = (V^1, V^2)$ and taking half steps for the difference, (dropping index n)

$$F(u_{(i,j)}) = \nabla \cdot \vec{V}_{(i,j)} = D_x V^1_{(i,j)} + D_y V^2_{(i,j)}$$

$$= V^1_{(i+\frac{1}{2},j)} - V^1_{(i-\frac{1}{2},j)} + V^2_{(i,j+\frac{1}{2})} - V^2_{(i,j-\frac{1}{2})}$$

Here this V^1 is,

$$V^1 = (a + b\kappa^2)(\eta^1) - \frac{2b}{|\nabla u|}\{(D_x(\kappa|\nabla u|), D_y(\kappa|\nabla u|)) \cdot (\tau^1, \tau^2)\}(\tau^1)$$

$$= (a + b\kappa^2)(\eta^1) - \frac{2b}{|\nabla u|}(D_x(\kappa|\nabla u|)\ \tau^1 + D_y(\kappa|\nabla u|)\ \tau^2)(\tau^1)$$

$$= (a + b\kappa^2)(\frac{u_x}{|\nabla u|}) - \frac{2b}{|\nabla u|}(D_x(\kappa|\nabla u|)\ (-\frac{u_y}{|\nabla|}) + D_y(\kappa|\nabla u|)\ (\frac{u_x}{|\nabla|}))(-\frac{u_y}{|\nabla|})$$

we used $\frac{\partial}{\partial \vec{\tau}} = \vec{\tau} \cdot \nabla$ and $\vec{\eta}$ is normal vector and η^1 is fist component of it, $\vec{\eta} = (\eta^1, \eta^2) = (\frac{u_x}{|\nabla u|}, \frac{u_y}{|\nabla u|})$ and τ^1, τ^2 are first and second component of tangent vector $\vec{\tau} = (\tau^1, \tau^2) = (\frac{-u_y}{|\nabla u|}, \frac{u_x}{|\nabla u|})$. Notice the index here is at point $(i + \frac{1}{2}, j)$, so for $\kappa_{(i+\frac{1}{2},j)}$ we took min-mod between $(i + 1, j)$ and (i, j),

$$\kappa_{(i+\frac{1}{2},j)} = minmod(\kappa_{(i+1,j)}, \kappa_{(i,j)})$$

$$where \quad minmod(\alpha, obeta) = \frac{sign(\alpha)+sign(\beta)}{2} min(|\alpha|, |\beta|).$$

For u_x and D_x at point $(i+\frac{1}{2}, j)$, we used difference of points $(i+1, j)$ and (i, j), shown in Fig. 4.5,

$$u_x|_{at\ (i+\frac{1}{2},j)} = \frac{u_{(i+1,j)}-u_{(i,j)}}{2}$$

$$D_x(\kappa|\nabla u|)|_{at\ (i+\frac{1}{2},j)} = \frac{(\kappa_{(i+1,j)}\ |\nabla u|_{(i+1,j)})-(\kappa_{(i,j)}\ |\nabla u|_{(i,j)})}{2}$$

For $|\nabla u|_{(i,j)}$ we used central differencing in x and y-directions as (4.6).

For u_y and D_y at point $(i+\frac{1}{2}, j)$, Fig. 4.5, we used central differencing of point $(i+1, j+1), (i+1, j-1)$ and central differencing of points $(i-1, j+1), (i-1, j-1)$. For better approximation of D_y, we min-mod of these two central differencing,

$$u_y|_{at\ (i+\frac{1}{2},j)} = minmod(\frac{u_{(i+1,j+1)} - u_{(i+1,j-1)}}{2}, \frac{u_{(i-1,j+1)} - u_{(i-1,j-1)}}{2})$$

and

$$\begin{aligned} &D_y(\kappa|\nabla u|)|_{at\ (i+\frac{1}{2},j)} \\ &= minmod(\frac{(\kappa|\nabla u|)_{(i+1,j+1)}-(\kappa|\nabla u|)_{(i+1,j-1)}}{2}, \frac{(\kappa|\nabla u|)_{(i-1,j+1)}-(\kappa|\nabla u|)_{(i-1,j-1)}}{2}) \end{aligned} \tag{4.7}$$

For V^2, do the same for half point $(i, j+\frac{1}{2})$.

We used min-mod to get better results. Fig. 4.6 shows the comparison between different schemes which can be used for approximation for D_y at point $(i+\frac{1}{2}, j)$. Original image without inpainting Domain is (a), this is the optimal image for the result. Forward for (b) means using Central Differencing at $i+1$ for D_y at point $(i+\frac{1}{2}, j)$, i.e. $\frac{u_{(i+1,j+1)}-u_{(i+1,j-1)}}{2}$. Another commonly used method is using average of $i+1$ and i for point $(i+\frac{1}{2}, j)$, this is shown is (c) average, $\frac{u_{(i+1,j+1)}-u_{(i+1,j-1)}+u_{(i,j+1)}-u_{(i,j-1)}}{4}$. (d) min-mod is what was have used for this chapter and the result shows, this has best edge recovery compared to other methods.

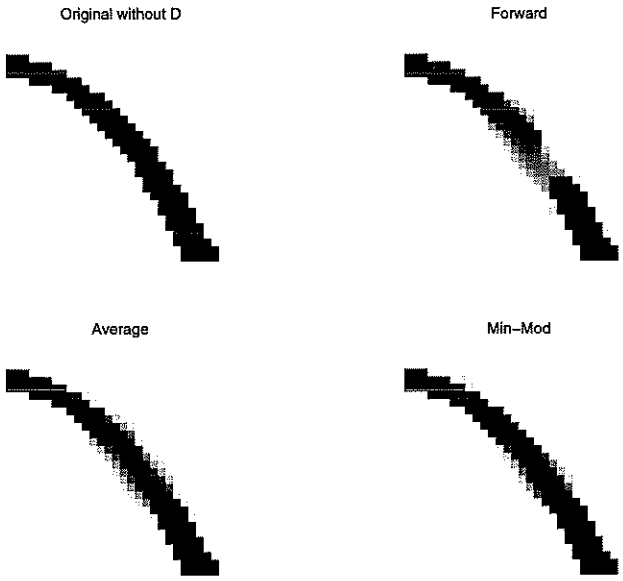


Figure 4.6: Reason for using Min-Mod : (a) Original image without inpainting Domain i.e. optimal case for the result. (b) using Central Differencing at $i+1$ for u_y and D_y at point $(i+\frac{1}{2}, j)$, (c) Using average of Central differencing at $i+1$ and i. (d) Using min-mod of Central differencing at $i+1$ and i equation (4.7). Using min-mod give much clearer and closer to optimal case result.

4.4 Analysis and Results

In Fig. 4.7, we analyze the results intuitively. The image in the left, Image 1, has disconnected object, the image on the right, Image 2, has connected object and the dotted line are the inpainting domain. Total variation of image 1 inside the

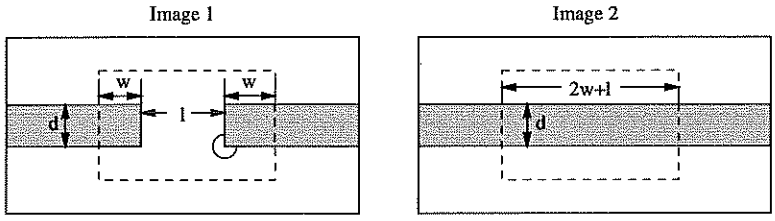


Figure 4.7: Comparing Total Variation and Curvature value to understand why we get different results.

inpainting domain (dotted box), is $(4w + 2d)M_1$ where this M_1 is the difference between two color values (background color and the strip color). The same way, Total variation of image 2 is $(4w + 2l)M_1$. So, $(4w + 2d)M_1 < (4w + 2l)M_1$ if and only if $d < l$. This means, if $d < l$ than Total variation for image 1 is smaller than image 2, so TV inpainting prefers result as image 1 then image 2. Now considering the curvature, the curvature term $|\nabla u|\kappa^2$ for image 1 is $(4w + 2d + 4v)M_2$ where M_2 is the curvature around the straight line and v is the value for the square edge (circle in Fig. 4.7, image 1). And curvature term $|\nabla u|\kappa^2$ for image 2 is $(2l + 4w)M_2$, i.e. $(4w + 2d + 4v)M_2$ vs $(2l + 4w)M_2$, this is same as $2d + 4v$ vs $2l$. If l is small (two bars are close), then $(2d + 4v) > 2l$, therefore, by enforcing curvature it will try to connect the result. And if l is vary large i.e. $(2d+4v) < 2l$, then even with curvature term the result will be disconnected. However, this v value is very big so minimizing curvature will try to circle out the sharp square edge to circular edge.

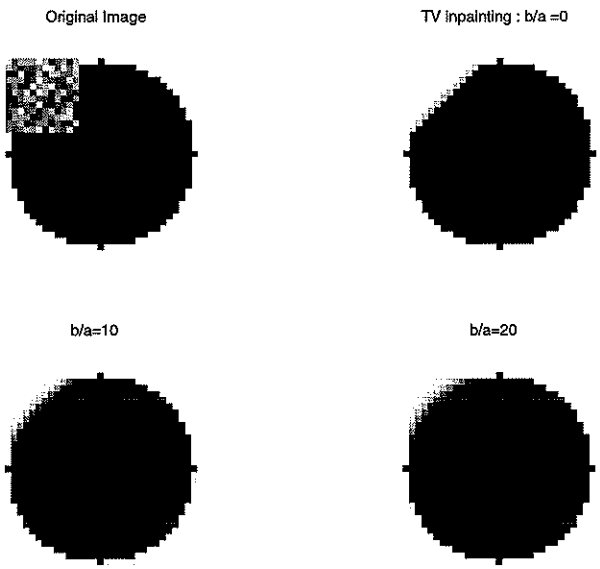


Figure 4.8: Ratio b/a : (a) Original image with Inpainting domain, (b) TV Inpainting ($b/a = 0$) (c) $b/a \sim 10$ (d) Curvature Inpainting $b/a \sim 20$. Ratio b/a tell how close you want to get your result either closer to TV or curvature Inpainting.

We can also see similar result by changing values for constant a and b from the equation (4.5). There are three constant to choose, Lagrange multiplier λ, TV inpainting term a and curvature inpainting term b. λ is the amount for enforcing outside information of the inpainting domain, therefore, when there is no noise we can use as big λ as possible and if we do have some noise outside E we can enforce λ less. The constants a and b is for the amount we are enforcing TV term and curvature term. Let's consider the ratio b/a. If this b/a is close to 0, it is TV inpainting, and if b/a is big, it is more like curvature inpainting. For our experiment, even for curvature inpainting, we've used small a instead of $a = 0$. This speeds up the convergence and gives better results, since minimizing Total Variation helps getting ride of the noisy gray inpainting domain D. For most of

our experiment we have used same a, b and λ, the ratio was about $b/a \sim 20$. Fig. 4.8, 4.9 shows, relation between a and b i.e. ratio b/a.

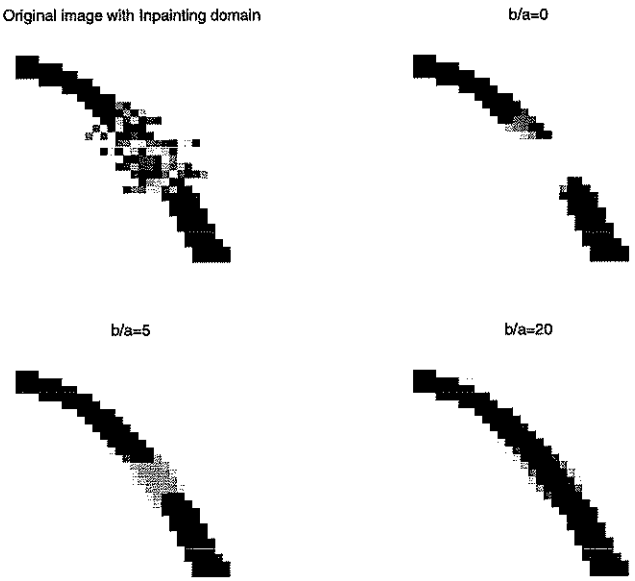


Figure 4.9: Ratio b/a (a) Original image with Inpainting domain, (b) TV Inpainting ($b/a = 0$) (c) $b/a \sim 5$ (d) curvature Inpainting $b/a \sim 20$. Ratio b/a tell how close you want to get your result either closer to TV or curvature Inpainting.

We have improve TV inpainting by adding curvature to our minimization function. First, we can recover the result connected, as in Fig. 4.10. Using TV inpainting, it will take the values of the bigger area connected to the inpainting domain. By enforcing the curvature term, we can recover the image connected, since the curvature is very large near the shape edge (Kink), so by minimizing it will try to avoid having sharp kink. Another result can be seen in Fig. 4.11, from the original image, the result could be three triangles but also one big triangle. Freedom of choosing constant a, b allows to get two different results depending on the original image Fig. 4.12

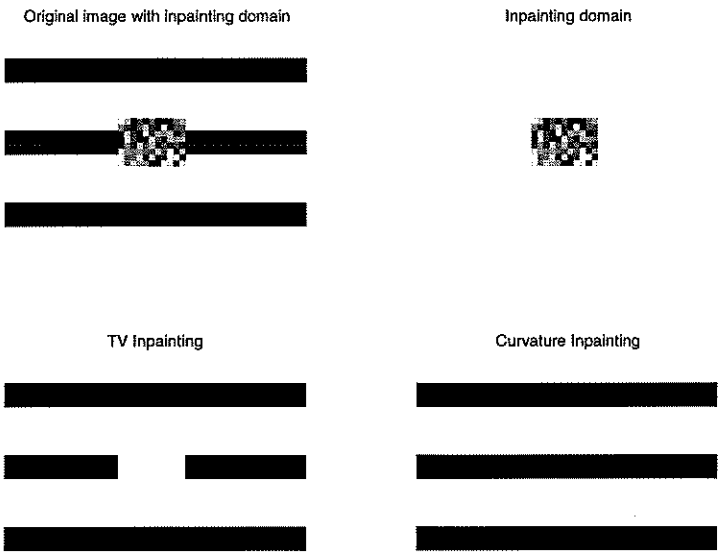


Figure 4.10: From the original image it is reasonable to think the black line is connected, but TV inpainting will recover to disconnected result. But curvature inpainting will recover as what we expected.

Secondly, adding curvature gives better results in recovering circular images, Fig. 4.13. Using TV inpainting, it prefers straight results which minimizes Total Variation. However, by curvature inpainting, it tries to keep the curvature for the inpainting domain, recovering circular image as we expect.

Finally, Fig. 4.15 and Fig. 4.16 shows more challenging images, recovering a curve instead of straight line or disconnected image.

4.5 Conclusion

By adding the curvature term to the functional, and using Euler's Elastica model, we are able to get connected and curved results, as shown in previous section. This is a variational inpainting method improving TV inpainting.

We are able to express the Euler-Lagrange equation using Tangent and Normal vectors, which allows to use simple numerical methods. However, the numerical computation is very slow, and improvements can be farther studied.

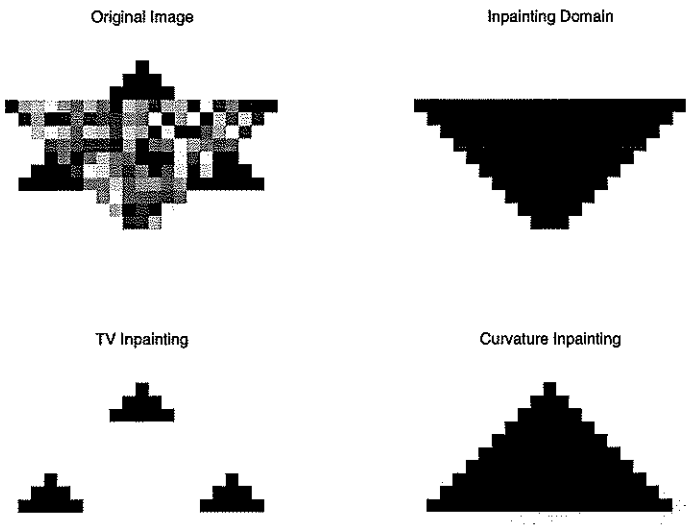


Figure 4.11: From the original image the result could be three small triangles but also one big triangle. By TV inpainting, you'll get three small triangles and by curvature inpainting, one big triangle.

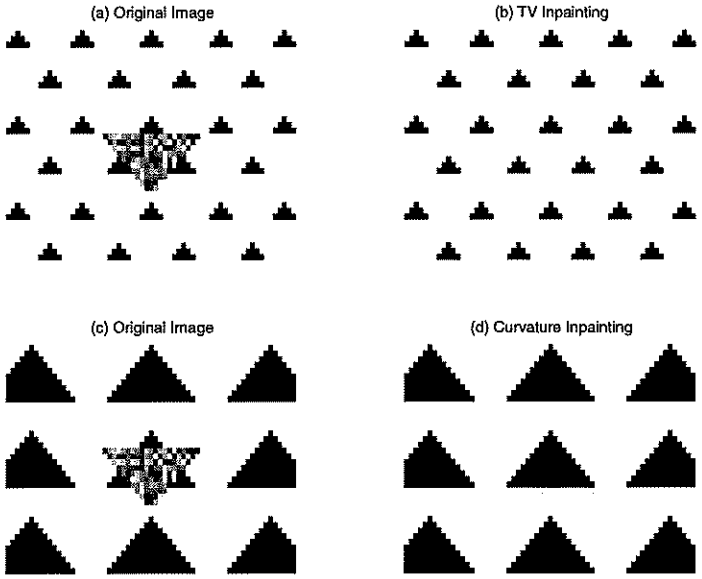


Figure 4.12: We can choose to use TV inpainting or curvature inpainting for Fig. 4.11, depending on what Original image is. If (a) is given, we use TV inpainting to get result (b), and if (c) is given, we use curvature inpainting to get result (d).

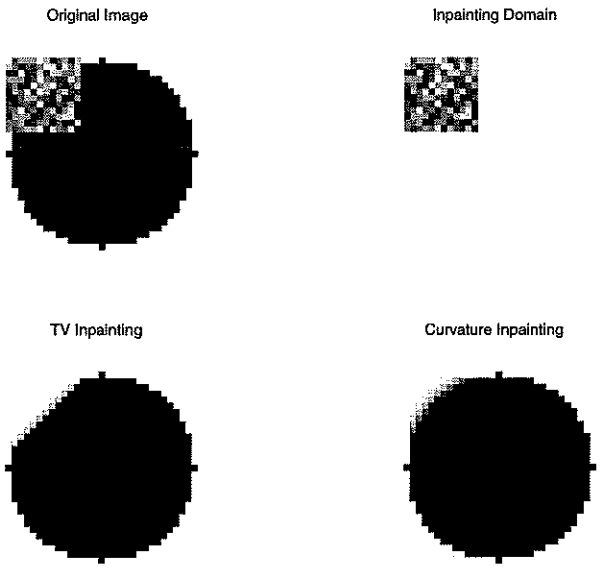


Figure 4.13: From the original image we expect the results to be circle, but by TV inpainting it recovers to straight line. curvature inpainting, gives circular result closer to what we expect.

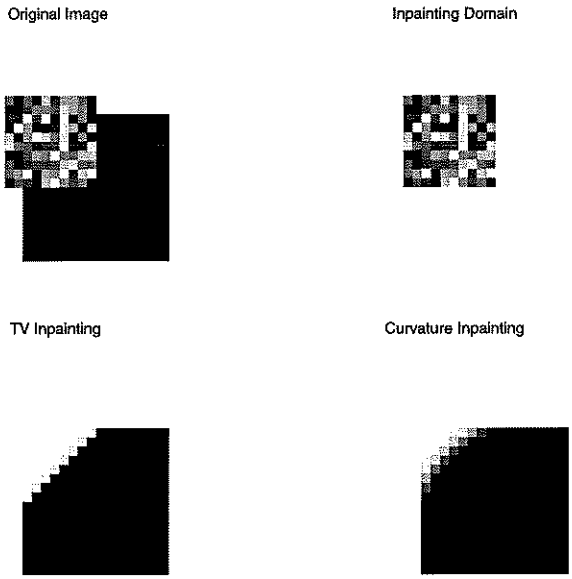


Figure 4.14: From the original image we expect the result as perfect square. By using curvature inpainting, the results is improved then TV inpainting.

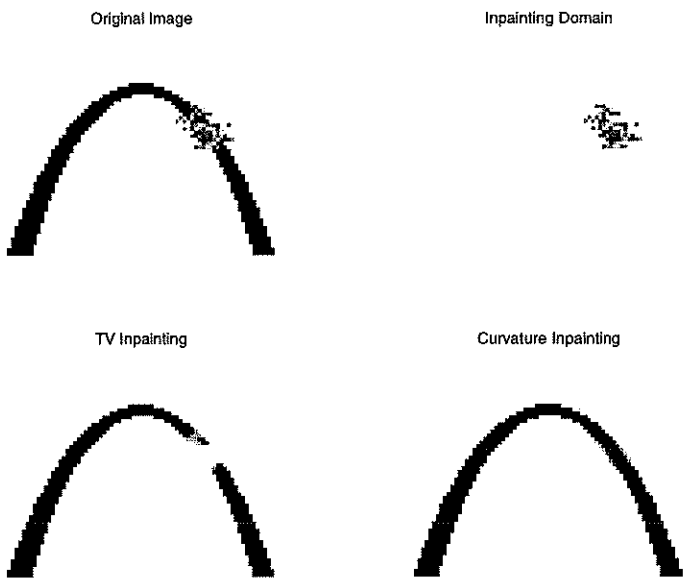


Figure 4.15: By using second order inpainting we improve the result. Result is connected with curve as expected.

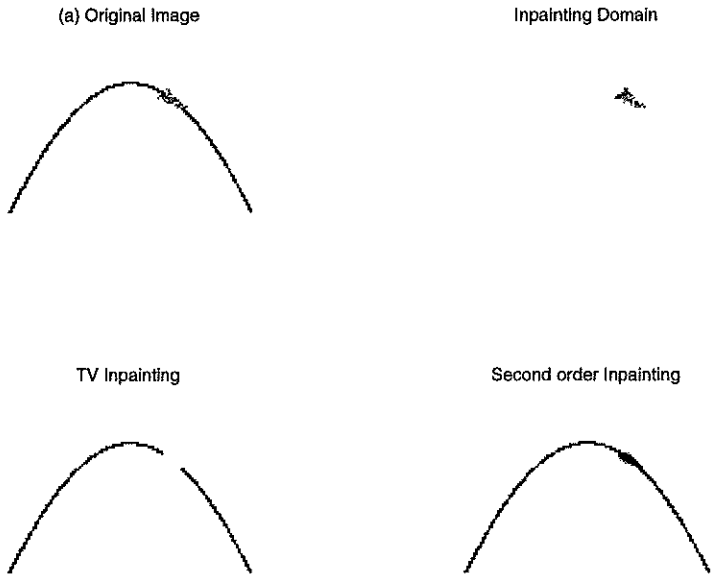


Figure 4.16: Even for thin curve, second order inpainting works. Also gives better result then first order disconnected result.

CHAPTER 5

Landmark based Inpainting from Multiple Views

5.1 Introduction

Inpainting refers to the specific image restoration task of reconstructing an image with missing or damaged regions. For example, when part of a painting is damaged with a crack or a part of a photograph is covered by some letters, inpainting algorithms use local information to fill in the crack, gap and undesirable letters.

Digital inpainting was first proposed by Bertalmio et. al. [BSC00], and many approaches followed. There are some approaches based on solving PDEs including Masnou and Morel's level line based disocclusion [MM98] and Chan and Shen's TV minimizing inpainting [CS01a]. More literature can be found in [BBC01, CKS01a, CS01b, CS01c].

These inpainting methods only make use of local information. Therefore, the missing region needs to be small enough to have good results. Fig. 5.1 is one of the results from curvature based inpainting [CKS01a], which is an improved version of *Total Variation* minimizing inpainting [CS01a]. With Fig. 5.1 (a), it is reasonable to think the object is a perfect square, however, even with curvature inpainting in (c), it is not possible to recover the square corner. In order to recover the corner, global information or additional shape information is needed.

These are the problems we are interested in this chapter.

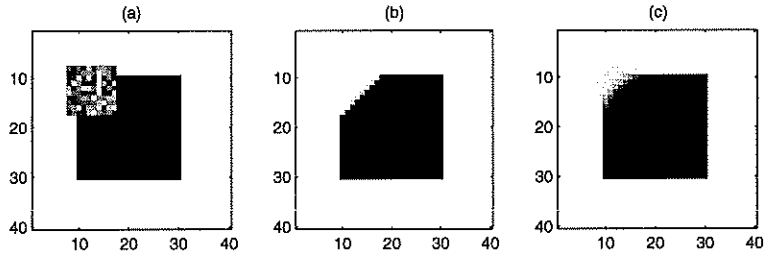


Figure 5.1: (a) is the original image with the square noisy area as the region to be filled, (b) is the result using Total Variation inpainting [CS01a], and (c) is the result using curvature based inpainting [CKS01a].

We are proposing a problem with large missing domains such that it is not enough to recover the region only using local information. However, the recovery process does assume there is additional information available. This is reasonable since when humans look at an image with a missing domain, we automatically recover image without any difficulty. As in Fig. 5.2, it is reasonable to think that (a) is an image of a checkerboard box, therefore, (b) can be used as the reference. In addition, when there is a damaged image of a known object or a building, similar photo can be used as a reference. An image of reconstructed building (d) can be used to retouch the same building in image (c).

There are some restoration and inpainting studies done for movies (image sequence). Guichard [Gui98] introduced AMG model to deal with movie images to successfully denoise and inpaint movies. Kokaram et al. [KMF95a, KMF95b] have studied detection and interpolation of missing data in image sequences. Chanas et al. [CCB01] proposed using 3D diffusion for recovering image sequences, considering time as the third axis. However, in our proposed problem, the reference image and the damaged images are not assumed to be in any sequence. We are

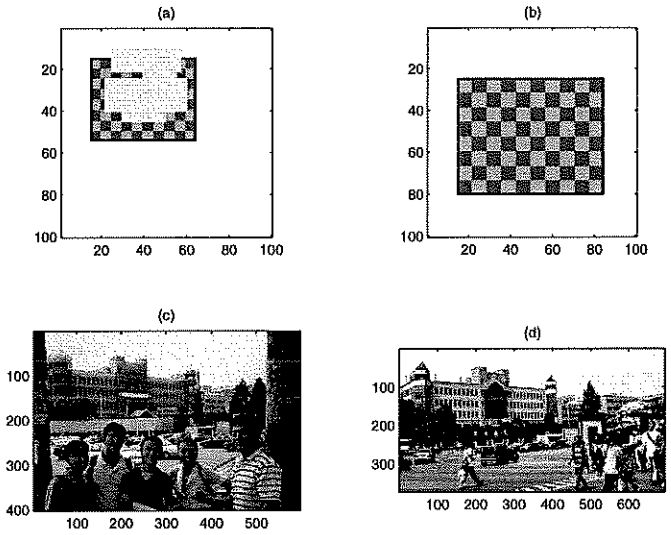


Figure 5.2: (a) has a gray missing region, and (b) is an image of the same object in different scale. (b) can be used for inpainting image (a). (c) and (d) are images of the same building taken at different times. In (c) the building is under construction and (d) is after the construction is completed. (d) can be used as reference image to replace the building in (c).

allowing the additional image to be distorted, image from any different viewpoint, or image of similar object, not the same.

As one possible method, we propose using Landmark matching, interpolation, and inpainting. Landmark matching is often used in image registrations, image morphing, and shape matching. In this way, we can enforce the relative position information explicitly, allowing the objects to move far and to be distorted. After identifying high variance points, the matching of these landmarks is achieved by using modified shape context information [BMP01]. After the (matching) correspondence is assigned for each extracted point, we interpolate the transform

information to the whole image by thin plate spline [BMP01, Pow95, Wah90]. For the final step of inpainting, we copy the information from one image to another using the interpolated transformation.

One alternative method in the literature for point extraction to affine transformations is RANSAC [FB81]. RANSAC will automatically extract points, pick a small number of points from that set of extracted points, get the affine transformation, and then choose the best affine transformation which minimizes the difference. This method is useful for the case with a small number of outcast points and for affine transformation. However, with RANSAC, many different affine transforms need to be tested to get the best match. We carefully looked into each steps, points extraction, matching and interpolation to better understand the image information and carefully take care of the inpainting region. In addition, when the object is deformed it is better to add spline interpolation together with affine transformation.

The outline of the chapter is as follows. In Section 5.2, we present the model formulation and explain why optical flow analysis is not used. In Section 5.3, we describe the method in detail. Examples of reconstructed results are shown in Section 5.4, followed by concluding remarks in Section 5.5.

5.2 Problem Formulation

In this chapter, we assume there are only two images: one image, I_1, with missing information and another, I_2, with the information missing from I_1. Let $I_1, I_2 : \Omega \longrightarrow \mathbb{R}_+$ be those two images, where $\Omega \subset \mathbb{R}^2$ is the image domain. I_1 has a domain D representing the missing information. This D is called the *inpainting domain*, and we assume it has already been identified (Fig. 5.3). I_2 is the second

image (reference image), supposedly undamaged.

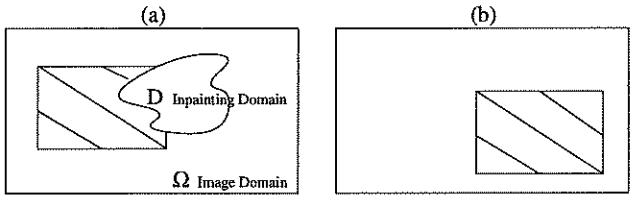


Figure 5.3: Setting of the problem : (a) is I_1, the original image with D as the inpainting domain. (b) is I_2 with a complete object information, which can be used to inpaint I_1.

We are looking for a map $g : \Omega \longrightarrow \mathbb{R}^{\not\in}$ which satisfies,

$$I_1(\vec{x}) = I_2(g(\vec{x})) \quad \forall\, x \in \Omega \cap g^{-1}(\Omega). \tag{5.1}$$

We can find this g as the minimizer of a functional for measuring how well (5.1) is satisfied. Adding the regularizing term to the minimizing Total Variation of g, the functional becomes

$$\vec{g}(\cdot) = \arg\min \int_{\Omega \setminus D} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\mu(\vec{x}) + \int_{\Omega} |\nabla \vec{g}| d\mu(\vec{x}) \tag{5.2}$$

One approach to solving (5.2), when the images come from a movie sequence, is optical flow. Equation (5.1), and the first term in equation (5.2),can be seen as the brightness constancy constraint in computer vision. When the two images are from a movie sequence and are consecutive, we can define I_1, I_2 to be $I_1 = I(\vec{x}, t)$ and $I_2 = I(g(\vec{x}), t+1)$, where t is time. In the simplest case when g is of the form $g(\vec{x}) = \vec{x} + \vec{u}$, the vector field $\vec{u}$ is called the *optical flow field* [BB95, BIK01, HS81, KTB96], and it satisfies the following optical flow equation : $\nabla I \cdot \vec{u} + I_t = 0$. One approach to this problem can be computing u outside the inpainting domain D,

and enforce the smoothness of u throughout the whole domain Ω:

$$\vec{u}(\cdot) = \arg\min \int_{\Omega \backslash D} |\nabla I \cdot \vec{u} + I_t| d\mu(\vec{x}) + \int_{\Omega} |\nabla \vec{u}| d\mu(\vec{x}).$$

Then, once the vector field $\vec{u}(x)$ in $\Omega \backslash D$ has been filled in, the restored image can be obtained as $I(t, \vec{x}) = I(t+1, g^{-1}(x)), \forall\ x \in D$ where $g(\vec{x}) = \vec{x} + \vec{u}(\vec{x})$.

Using optical flow might work for movie sequence, when $I_1 = I(t)$ and $I_2 = I(t+1)$ are not very different. However, optical flow is not very useful for large domain inpainting problems when the two images are very different. One major disadvantage is the local aperture problem: "Flow" related methods depend only on local information, not on global position information (see Fig. 5.4). If the object moves too far, even with multiscale analysis [KWM99, LKW94], it may be difficult to capture the right positions. In addition, in order to get the directional information with discontinuities, sophisticated numerical schemes like min-mod or Eno-like scheme are needed to better capture the accurate flow estimation.

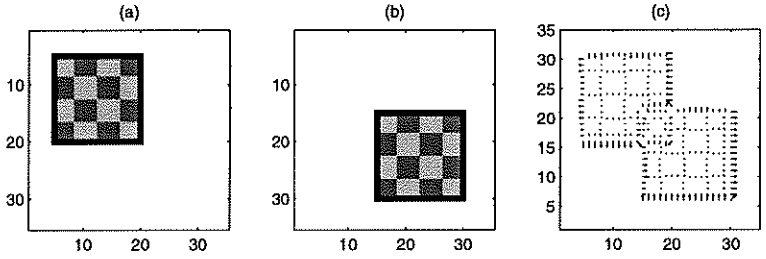


Figure 5.4: (a) is I_1, (b) is I_2 and (c) is optical flow. From I_1 to I_2, the ideal directional field should be uniform arrows of length $10\sqrt{2}$, in down right direction. However, calculated optical flow result (c) is far from what is expected.

In addition, the g from equation (5.2) could be a finite-dimensional group, therefore, there may not be a need to regularize with respect to g. The second

term from the cost functional, equation (5.2), can introduce a bias towards small displacements.

In our approach, instead of using the optical flow model, we directly use the translational model and minimize

$$|I_1(\vec{x}) - I_2(g(\vec{x}))| \tag{5.3}$$

and get g independently. To find the appropriate g for equation (5.3), we use landmark matching, and spline interpolation for regularization (thin plate spline approximation). Landmarks are the points extracted from the images which have the directional information and thin plate spline is needed to extend those directional information to the whole domain. This is equivalent to minimizing the following functional:

$$\vec{g}(\cdot) = \arg\min \int_{Landmarks} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\vec{x} + \int_{\Omega} [g_{xx}^2(\vec{x}) + g_{xy}^2(\vec{x}) + g_{yy}^2(\vec{x})] d\vec{x}. \tag{5.4}$$

The first term is the fitting term for landmark matching, and the second regularizing term corresponds to thin plate spline interpolation. In our actual implementation, instead of directly finding $\vec{g}$ by solving the minimization problem (5.4), we decouple the two terms and first identify the landmark correspondence separately and then use this information for the spline interpolation.

5.3 Description of the Method

In this section, we describe each of the three main phases of our method: landmark matching, interpolation and inpainting.

First, any method for extracting points can be used for landmark extraction. After the landmark points are extracted, we carefully match the landmarks

between I_1 and I_2. We use shape context information as well as intensity information to match the points. The shape context information contains global position information for each point relative to all the other extracted points.

The number of extracted points are small compared to entire discretized image domain; yet, these are the only points with correspondence information. Therefore, we need to interpolate the directional information to the whole domain. This is similar to sparse data interpolation, for which radial basis functions are commonly used. We use thin plate spline for this interpolation.

After determining the interpolation, which gives the transformation for every points in the image domain, we finally copy the information from one image to another. When there are more than one object or the object and the background are moving in separate direction, mask of the object is needs to be identified to get better results. Also, local inpainting methods on I_1 can be combined with the copying information from I_2 using interpolated transformation.

5.3.1 Landmark Matching with Modified Shape Context

Landmark Extraction : There is a significant literature on landmark extraction, and any effective method can be used for this purpose. We tried to use salient features like corners, intersections and high curvature points by finding high local variance points and thresholding it with some constant. However, points near the inpainting domain D of the damaged image I_1 can be mistaken as salient features. Therefore, we set the values on D to be zero which prevents the points next to the inpainting domain from being extracted as feature points. We extract the high variance points by thresholding the following V with some

constant,

$$V(i,j) = \sqrt{\sum_{(k,l)\in N} \chi(k,l) \cdot \chi(i,j) \cdot (I_1(k,l) - I_1(i,j))^2}$$

$$where \quad \chi = \begin{cases} 1 & (i,j) \in \Omega \backslash D \\ 0 & (i,j) \in D \end{cases},$$

and $N = \{(i+1,j),(i-1,j),(i,j+1),(i,j-1)\}$.

Shape Context : After the landmark extraction, we need to incorporate the global positional information to match landmarks. Therefore, we assign correspondence using modified *Shape Context* profiles. Shape context was introduced by Belongie et al. [BMP01], to incorporate global information of shape to better match and compare shapes. First the shapes are represented using points along the boundary. Then for each point p, the shape context is calculated storing relative positions of other points to itself. The simplified version is illustrated in Fig. 5.5. From the sectors in (a), the number of points contained in each sectors are stored at each corresponding cells in (b). By using histogram comparison of these profiles for all the points, the authors were able to find correspondence between the points from two shapes.

Modified Shape Context : Despite the heavy amount of calculations and much information to be stored, using relative position information is useful in our problem. Therefore, we modify the shape context to better serve our purpose. First, since we are using only salient features, for I_1 and I_2 the numbers of points extracted are much smaller than in [BMP01]. Therefore, instead of 60 cells in a sector, we only use 6. In addition, we add intensity for better matching the correspondence. (Belongie et al. [BMP01] did not use the intensity information. The points are representing the boundary of the shape and intensity information is not necessary for shape matching.)

As an illustration, let H_p be the shape information for a point $p = (i,j)$;

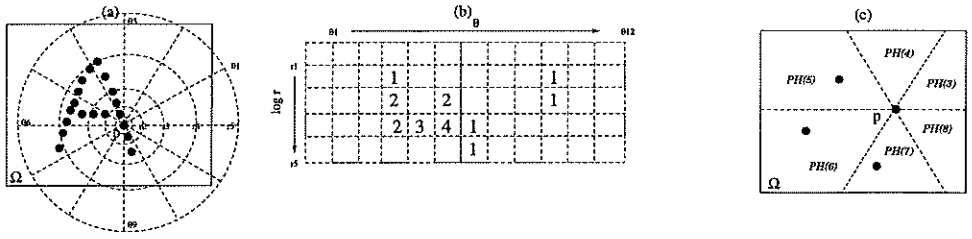


Figure 5.5: (a) sector used in [BMP01]. Each sector in (a) corresponds to one square cell in (b). The number in each cell in (b) is the number of points in each corresponding sector. (In [BMP01], histogram is used instead of the number of points.) (c) is the sector used for our shape context, for point p, the profile vector is $H_p = (I(p), Av(p), 0, 0, 1, 1, 1, 0)$. (The third through eighth components store the number of points in the each sectors.)

H_p is a vector of size 8. The first component stores the intensity at point p, $H_p(1) = I(p)$, and the second component $H_p(2) = Av(p)$ stores the intensity average of 8 neighborhood $\frac{1}{8}\sum_{(k,l)\in A} I_{(k,l)}$ where $A = \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$. The remaining six components are shape context like information storing the number of points from each corresponding sections. For example, from Fig. 5.5(c), the profile for point p is $H_p = (I(p), Av(p), 0, 0, 1, 1, 1, 0)$.

After each profile H_p for all extracted points p from I_1, and each profile H_q for all extracted points q from I_2 are calculated, we find the best match by comparing the vector profile and comparing the distance. For each point p_i from I_1, we find

the point q_j from I_2 which minimizes the following energy:

$$c_1|H_p(p_i) - H_q(q_j)| + c_2\|p_i - q_j\| + c_3\|f(p_i) - f(q_j)\| \tag{5.5}$$

$$where \quad f(p_i) = \frac{p_i - \frac{1}{N_p}\sum_{N_p} p_k}{Max_{j \in N_p}\{p_j - \frac{1}{N_p}\sum_{N_p} p_k\}}$$

The first term is an L^1-norm for comparing vector profile, the second term is a regular Euclidean distance L^2 for keeping the points from moving very far between I_1 to I_2. The third term[1] is for comparing the relative distance of points in I_1 and I_2. The constants c_1, c_2 and c_3 are weights chosen to prevent the problem of having more than one minimum for each point.

After the best match is found for each point p_i in I_1, it is stored as $Match = \{(p_i, q_j)\}$, i.e. point p_i is matched with point q_j in I_2. The number of correspondence is small compared to the whole image domain, since we only extract a small number of salient features, and not all the points extracted are matched between I_1 and I_2. Therefore, we need to interpolate the information to the whole domain (next section). Defining $U(p_i) = \vec{q_j} - \vec{p_i}$ to be the transformation from point p_i in I_1 to point q_j in I_2, the transformation $U(p_i)$ is assigned.

Remarks

1. Depending on the size of the image,'block' instead of landmark point may be used. Size of the block needs to be determined, which will guarantee all the points inside the block move to same direction. In addition, when the two images are taken from movie sequence, information of extracted points

[1] $f(p_1)$ is the relative distance from point p_1 to the center of all p_is, i.e. $\frac{1}{N_p}\sum_{N_p} p_i$. This is scaled with maximum distance from center to furthest extracted points. Therefore, $0 \leq f \leq 1$. This term is effective when the object moves very far, i.e. all the points move far, yet, keeping the similar distance relative to each other. The scaling term $Max_{j \in N_p}\{p_j - \frac{1}{N_p}\sum_{N_p} p_i\}$ in the denominator is for scale invariance.

from I_1 can be used to better extract the points from I_2.

2. Since I_1 have a large inpainting domain D, it is efficient to work with a smaller number of extracted points. For example, in Fig. 5.6, let the point p in I_1 be the same point with point q in I_2. If we extract many number of points, I_1 is missing many points around the inpainting domain, thus, the point p and q will have very different shape context profiles and result in the point p not corresponding to the right match q. Therefore, it is best to work with a smaller number of points.

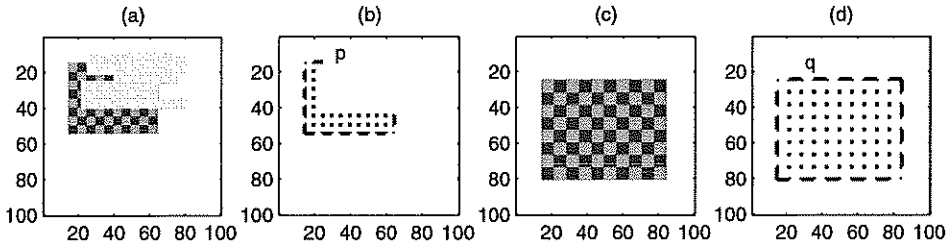


Figure 5.6: (a) is I_1 with a gray inpainting domain and (b) shows landmarks extracted from I_1. (c) is I_2 and (d) landmarks from I_2. The point p in (b) and q in (d) are the same point from the object, therefore, need to be matched. However, with too many landmarks extracted, the shape context information will be highly different from point p to point q. It is better to extract a smaller number of landmarks.

3. For the matching landmark correspondence, in some literatures, the Hungarian method is used. From two sets of same number of points, by trying possible permutations, the method looks for the best match which minimizes a certain energy functional. However, in our problem the number of points extracted from the two images I_1 and I_2 are most likely different due to the existence of D, also adding dummy points will result in unstable

solutions. Therefore, instead of using permutation, we pick the best match for each landmark p for I_1, that is, for each p, all landmarks q from I_2 is compared and get only the best match. Not all points will necessarily be assigned but only the best match.

4. When picking the best match for each landmarks, there are cases when one point p_1 from I_1 corresponds to multiple q's from I_2 (or vise versa) or due to occlusion, the appropriate match q_{right} from I_2 is hidden and another point q_{wrong} is matched for the best correspondence to p. Therefore, we also match the points from I_2 to I_1 as well, i.e. get both $MatchP = \{(p_i, q_j)\}$ and $MatchQ = \{(q_k, p_l)\}$ and use the correspondence which in both sets of matches (I_1 to I_2, $MatchP$ and from I_2 to I_1, $MatchQ$). For example, when the set $MatchP$ includes multiple correspondence and a wrong correspondence $\{(p_1, q_1), (p_2, q_1), (p_3, q_{wrong})\}$, the set $MatchQ$ including $\{(q_1, p_1), (q_{wrong}, p_4)\}$ will help eliminate the incorrect correspondence (p_2, q_1) and (p_3, q_{wrong}).

5. From I_1 to I_2, when the object and background are moving independently, the object and background points need to be separated to get better transformation. When the object moves far away in a non-uniform background, one transformation is not enough, Fig. 5.7 illustrates this point. Therefore after the landmark extraction, we need to separate the points into two classes : object points and background points. One way is to use intensity similarity and group the them to object points and background points, then find the correspondence only from each group (of same or similar intensity). For some cases, when the objects and the background have similar intensities, the landmarks may need to be manually identified and separated.

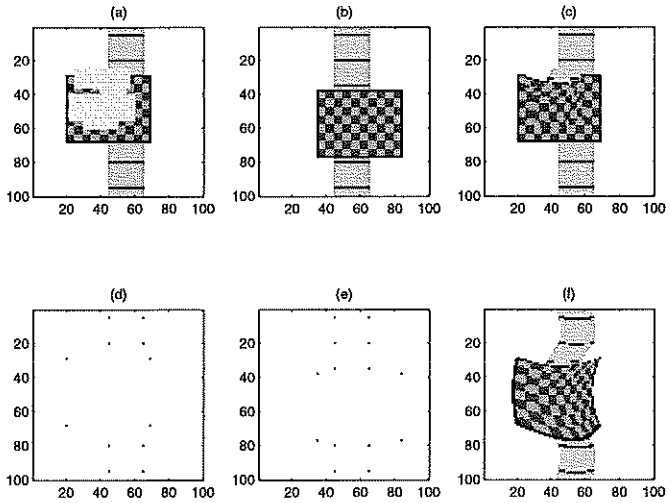


Figure 5.7: (a) is I_1 with the irregular gray area as the inpainting domain and (b) is I_2. The checkerboard is the object and the gray bar with horizontal strips is a part of the background. (c) is the recovered I_1 using (d) and (e) as the landmarks for I_1 and I_2 respectively. (f) shows the transformation U. Since the background is not uniform, if we use one transformation for all the extracted landmarks, the spline interpolation (f) gets highly distorted and the result in unsatisfactory recovery of I_1 in (c).

5.3.2 Affine Mapping and Interpolation

Compared to the image domain Ω, the number of extracted landmarks is relatively small, yet only the matched landmarks have directional information $U(p_i)$. Therefore, we need to interpolate the matched point directions to the whole image domain. This is similar to sparse data interpolation, for which radial basis functions are commonly used. In particular, we used thin plate spline for this interpolation.

Thin plate splines [BMP01, Pow95, Wah90] are commonly used for smooth

interpolations. They minimizes the bending energy of the embedded function: $E = \int_\Omega [f_{xx}^2(\vec{x}) + f_{xy}^2(\vec{x}) + f_{yy}^2(\vec{x})] d\vec{x}$. Solving this thin plate spline is equivalent to interpolating with bi-harmonic radial basis functions [Pow90]. We used $K(r) = r^2 log|r*r|$ for the radial basis function. Since $U(p_i)$ is a $2D$ directional vector, thin plate spline interpolation is applied twice to each x and y coordinates separately.

The known information is $U(\vec{p_i}) : R^2 \rightarrow R^2$ which have directional information from point p_i in I_1 to I_2. The interpolation function is given as :

$$U(\vec{x}) = \mathbf{A}\vec{x} + \mathbf{t} + \sum_{i=1}^{n} w_i K(|\vec{x} - p_i|),$$

where n is the number of correspondence, $\vec{x}$ is the point (x, y) in I_1 and $\mathbf{A}$, $\mathbf{t}$, and w_i are unknowns. We first calculate $\mathbf{A}$ and $\mathbf{t}$, independent of w_i, i.e. solve $U(\vec{p_i}) = \mathbf{A}\vec{p_i} + \mathbf{t}$ using only the known values of $U(\vec{p_i})$ and $\vec{p_i}$. (We used Gaussian elimination and, when the matrix is not square, the solution in the least squares sense.) Then the weights w_i are calculated using $\mathbf{A}$, $\mathbf{t}$, K and $U(p_i)$ for x and y coordinates separately [BMP01, Pow95, Wah90].

5.3.3 Inpainting - Filling-in The Information

Once the transformation U is found, what remains is copying the information from I_2 to I_1. This U is the transformation from I_1 to I_2, defined by the interpolation of $U(p_i)$. The new recovered image is

$$I_{NEW}(\Omega) = (1 - \chi_D)\ I_1 + \chi_D\ I_2(-U)$$

$$where\ \ \chi_D = \begin{cases} 0 & (i,j) \in \Omega \backslash D \\ 1 & (i,j) \in D \end{cases}$$

χ_D is the characteristic function of D. The first term is to keep the information of $\Omega \backslash D$ from image I_1 and second term is to copy the missing information using

inverse transformation $-U$ from image I_2.

In some cases, the recovered image might have some intensity discontinuity due to simply copying the information. For such cases, combining local inpainting techniques will help improve the results. We can incorporate *Total Variation*(TV) inpainting in addition to the transformation interpolation by minimizing following:

$$a_1 \int_{\Omega} |\nabla I| + a_2 \int_{\Omega/D} |I - I_1| + a_3 \int_{D} |I - I_2(-U(\vec{x}))|.$$

The first term is the TV minimization enforced in the whole image domain Ω. The second term is the fitting term so that the image outside the inpainting domain Ω/D will be close to the original I_1. The third term is the transformation inpainting, which copies information from I_2 to I_1.

As in Remark 5 of Section 3.1, when the background is non-uniform and moving independently to the object or there are multiple objects moving in different directions, one transformation is not enough to capture the different movements of the object and the background. For these cases, we need to consider the object and the background (or multiple objects) separately. We first separate the extracted points into background points and object points, then get two or more different transformations. In this way, we can insure better transformations corresponding to appropriate movements of the objects and the background separately.

With two (or more) different transformations, when we fill in the information, extra caution is needed. The inpainting domain should be separated into two regions, corresponding to the object and the background, see Fig. 5.8. Some techniques like optical flow segmentation or motion estimation [KD92, PD99, SB95] can be used for this purpose. However, these techniques are computationally very expensive and we prefer to use a heuristic method that gives good experimental

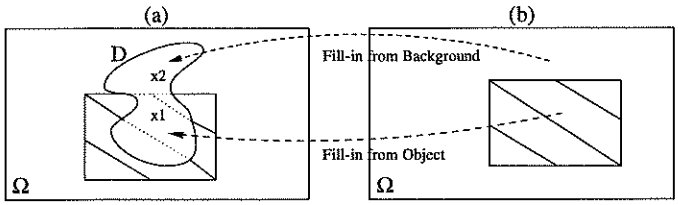


Figure 5.8: (a) is I_1 and (b) is I_2. When the inpainting domain in (a) covers both the object and the background, extra caution is need to fill-in the region with appropriate information either with the object transformation or the background transformation.

results. We use, segmentation [CV01] or masks of the object in I_2. With this mask of the object, we use the object transformation U_{obj} and inverse transform this mask back to I_1. This transferred mask shows the approximated object region in I_1, also inside the inpainting domain D. Therefore, the transformation inpainting for the inpainting domain D becomes,

$$I_{NEW}(D) = \chi_{object \cap D}\ I_2(-U_{obj}) + \chi_{background \cap D}\ I_2(-U_{bk})$$

copying the information from the object and the background separately – using two separate transformations U_{obj} (object transformation) and U_{bk} (background transformation).

5.4 Experimental Results

We presents some experimental results in this section. Some results are simple synthetic objects over an uniform background, some are objects in non-uniform background, and some examples using real images are presented.

Fig. 5.9 shows a couple of results when I_1, (a) and (f), are a textured box with a large inpainting domain. The large inpainting domain covers most of the

texture as well as one corner of the box. The first row shows results using (c) as reference image and the second row using (h). This object in (c) is in different position and the object in (f) is in different scale and position. The recovered results (e) and (j) show that the method can recover the checkerboard texture as well as the missing corner. By using modified shape context information without distance sectors, the scale change is automatically taken care of.

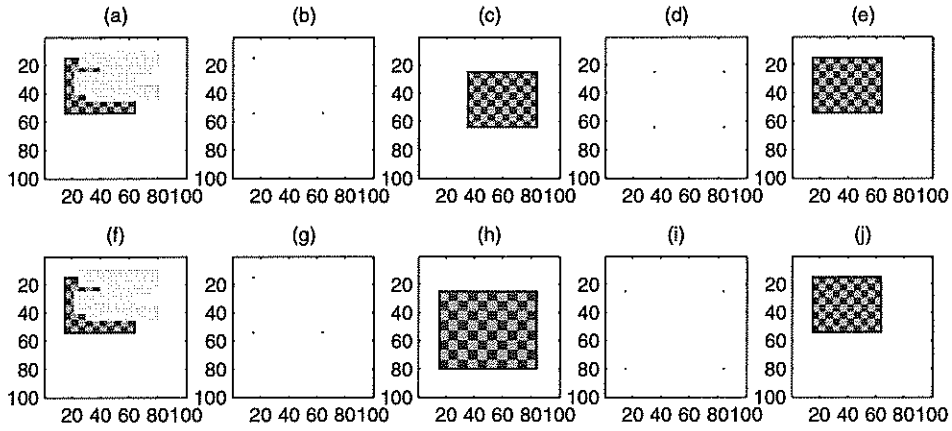


Figure 5.9: (a) is I_1 with the gray inpainting domain covering most of the texture and one corner of the box. (b) is the landmarks from I_1. Using (c) as reference image I_2 and (d) as landmarks of I_2, the recovered result is shown in (e). Images (f) to (j) shows another results using (h) as I_2. The method successfully recovers the texture and the corner of the box.

Our method can also be used for missing shape matching. Fig. 5.10 shows an example of how a given image is restored using the reference image I_2. From Fig. 5.10 (a), if we use local inpainting method, the result will look like an arrow heading up to the right corner. Instead, by using additional information from I_2, star shape can be recovered, even though the star in I_2 is different from the star

in I_1. However, as in Fig. 5.11, when the reference image, I_2, is very different from I_1, landmark matching fails to find a good correspondence. Fig. 5.11 (c) shows a different result compared to Fig. 5.10 (c). This is reasonable, since I_1 and I_2 are very different, and hard to be seen as the same object.

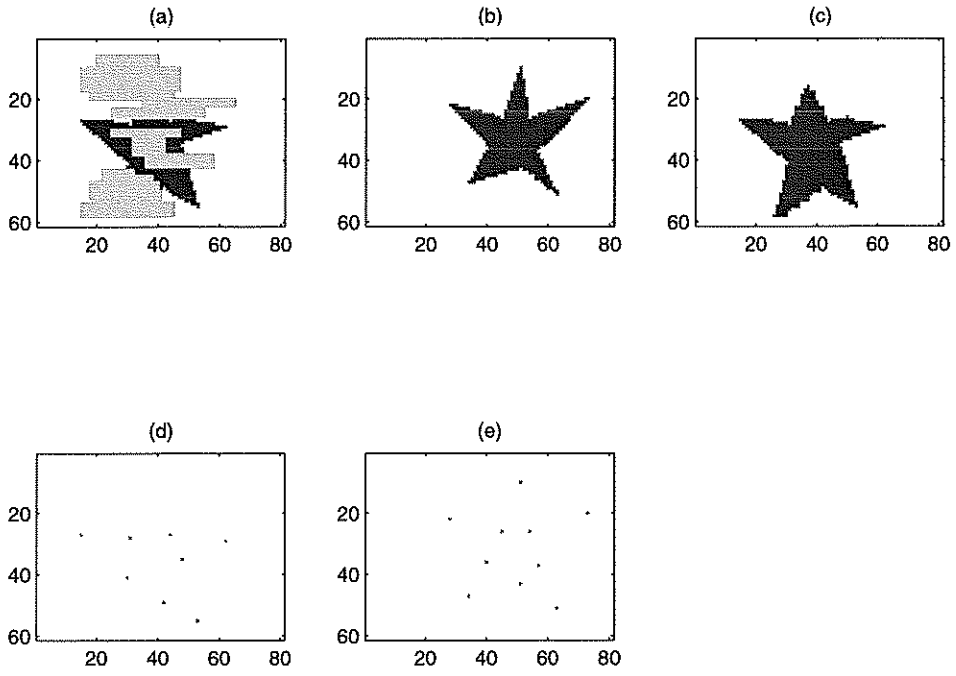


Figure 5.10: (a) I_1 can be recovered relatively similar to I_2 in (c). (e) is the recovered result. (b) and (d) are landmarks from I_1, I_2 respectively. The star in I_1 is different from the star in I_2, however, the method recovered the result (e) as a star shape.

In Fig. 5.12, 5.13,and 5.14, it shows results for images with object and background. After the landmarks are extracted, the points are separated into object points and background points. (The black dots are from the background and the 'O' points are from the object.) In Fig. 5.12, the checkerboard is in different position (c) and in different scale (h) in steady background. Using two separate

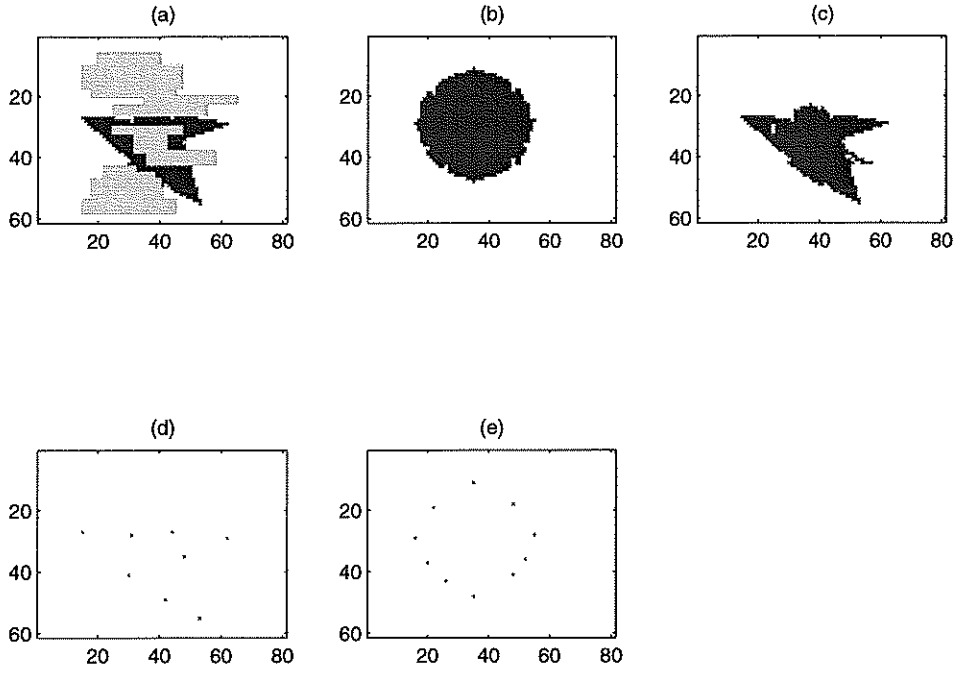


Figure 5.11: (a) is the same I_1 as in Fig. 5.10. However, due to the highly different I_2 in (c), the result (e) is unsatisfactory. (b) and (d) are landmarks from I_1, I_2 respectively.

transformations for the object and the background, the results successfully recovered the checkerboard texture and the corner. The stars in Fig. 5.13 (a) and (c) are different, and the background in more complicated. However, the method recovers the object similar to star in I_2, and the text background is successfully recovered using information from I_2. In Fig. 5.14, the stars in (a) and (b) are very different and the background is also shifted independent to the object. Both the object and the background are successfully recovered, using object and background transformations.

Next, we show some experiments using real images. Fig. 5.15 shows images of UCLA Royce Hall, (a) with couple of white disks as inpainting domains. Using

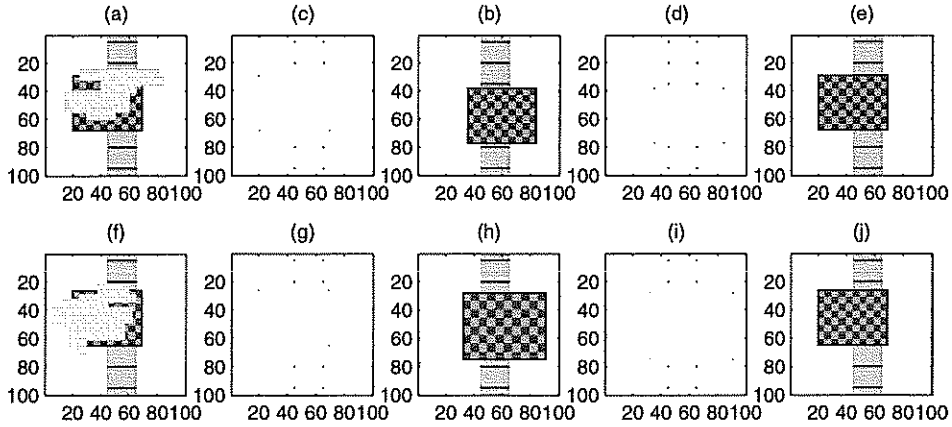


Figure 5.12: Two results are shown for inpainting an object moving in a non-uniform steady background. In the first row, (c) shows I_2 with the checkerboard box moved, and the second row, (h) shows the I_2 with the object moved and scaled. In I_1, (a) and (f), the large inpainting domain covers most of the checkerboard texture, one corner of the box as well as some part of the background. Using separate transformations for the object and the background, the inpainted results successfully recovered the texture and the corner.

(b), an image of Royce Hall taken from different time and different viewpoint, as reference image, (c) shows the recovered result. Missing windows at top left corner of the building is recovered and the three missing pole in front of the building is also added using information from I_2. In Fig. 5.16, I_1 and I_2 are two photo of same person taken at different places and at different times. With local inpainting methods, the missing eye can not be recovered. However, using information from I_2, the missing eye and the teeth are successfully recovered. In addition, in Fig. 5.17, two photos are of two different people, taken at different time and place. However, using the information from the other photo, I_1 can

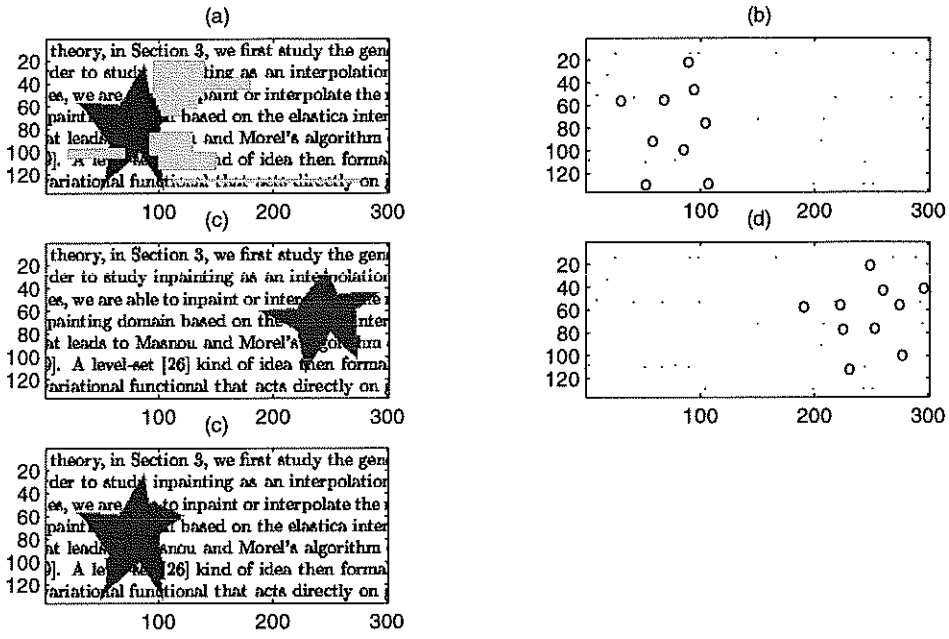


Figure 5.13: The result with deformed objects are shown. (a) I_1, (c) I_2, (b) and (d) the landmarks from I_1 and I_2 respectively, and (c) is the recovered result. The object in I_1 and I_2 are in different positions and deformed. The method recovers the object according to the given object in I_2, and text background is successfully recovered as well, using information from I_2.

recover the shape of the eyes. However, due to the intensity difference between I_1 and I_2, the recovered results show some discontinuities, post processing with local inpainting method will help improving these results.

We can apply this method to modify the undesirable photos. In Fig. 5.18, I_1 and I_2 are taken in the year 1998 and 2001 respectively (at Yonsei University, Seoul, Korea). I_1 has friends but the building is under construction. Using the whole building as inpainting domain D, I_1 is recovered using the more recent picture, I_2, of the completed building. The two images are taken from two different

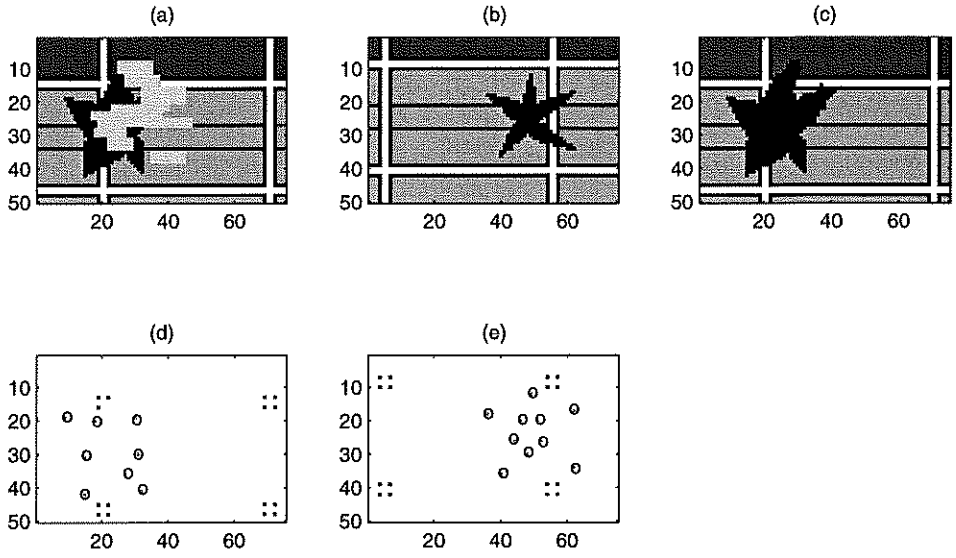


Figure 5.14: (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. The object is in different positions highly deformed and the background is also moved in a separate direction. Both the object and the background are successfully recovered.

time and positions. In Fig. 5.19, 5.20, I_1 and I_2 are taken consecutively. In I_1, the person in the middle has an odd facial expression and in I_2, the person in the middle is slightly occluded by a person from left. Using I_2 as reference and using the whole face as the mask to be inpainted, I_1 can be modified.

5.5 Conclusion

This method allows the missing region to be significantly larger compared to local inpainting methods, since there is additional information available. Using the modified shape context method [BMP01] helps to incorporate the global shape information. Thin plate spline includes affine transformation with spline

interpolations, which help to find transformations for distorted objects from one image to another.

Given a good set of landmarks, this method will work well, even when the two images I_1 and I_2 are quite different and with significantly large inpainting domains. The method is scale invariant and rotational invariance can be also achieved by rotating the shape context sectors.

One shortcoming of the method is in getting appropriate landmarks. Identifying feature points may not be easy, and sometimes manual identification may be needed. Our method is as accurate as the landmark matching is accurate. However, when the points are detected, using our modified shape context method gives good correspondence making use of the global information about the object and image.

For further studies, we can combine this method with segmentation to better capture the texture of the copied quality. When copying the information, the two images' texture or lighting may be different, therefore, by segmenting the objects in both images and modifying the intensity information inside the region, better restoration may be possible.

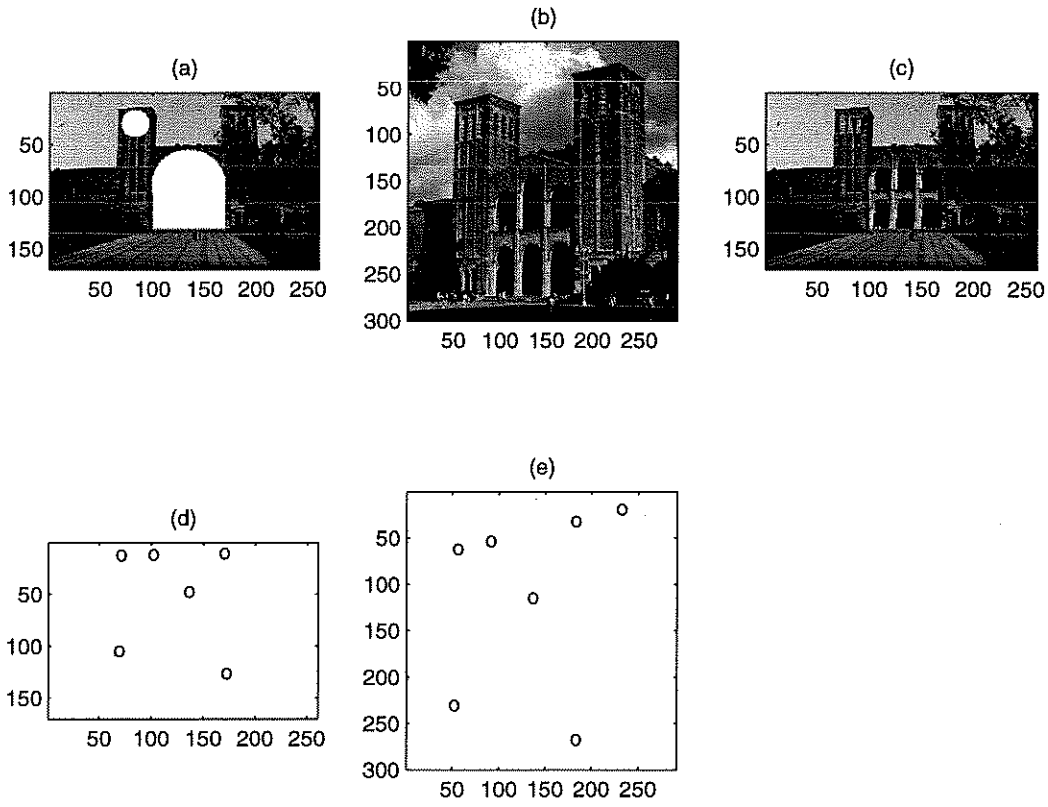


Figure 5.15: Application to a real image. (a) I_1 with white inpainting domain, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note that, I_1 and I_2 are taken in different time and from different viewpoints. However, the missing windows at the top left of the building are recovered and three poles are recovered.

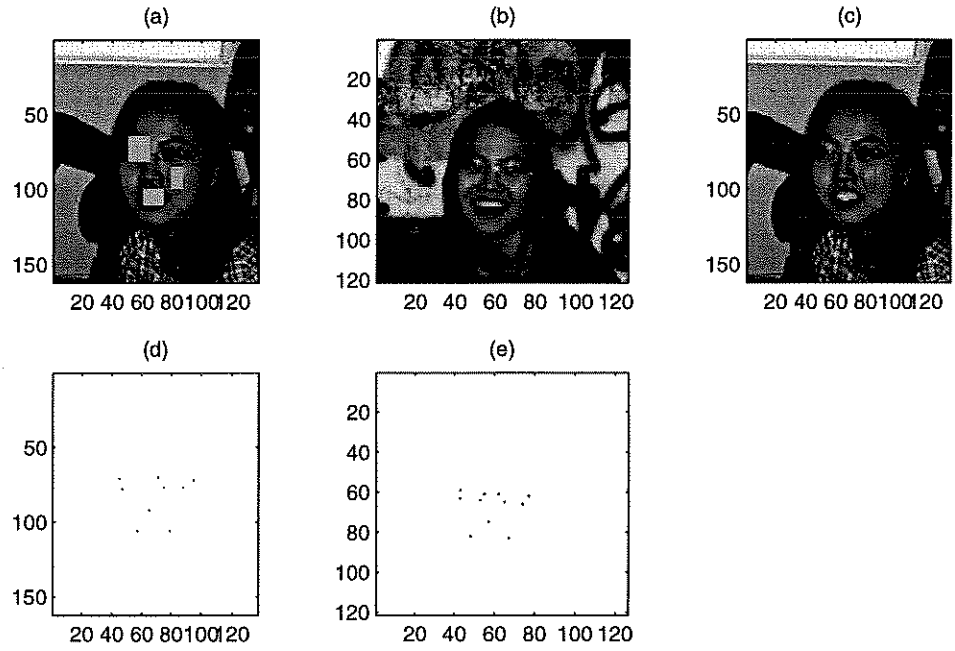


Figure 5.16: Application to a real image. (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note that, I_1 and I_2 are taken in different place and different time. However, the missing eye and the teeth are successfully recovered.

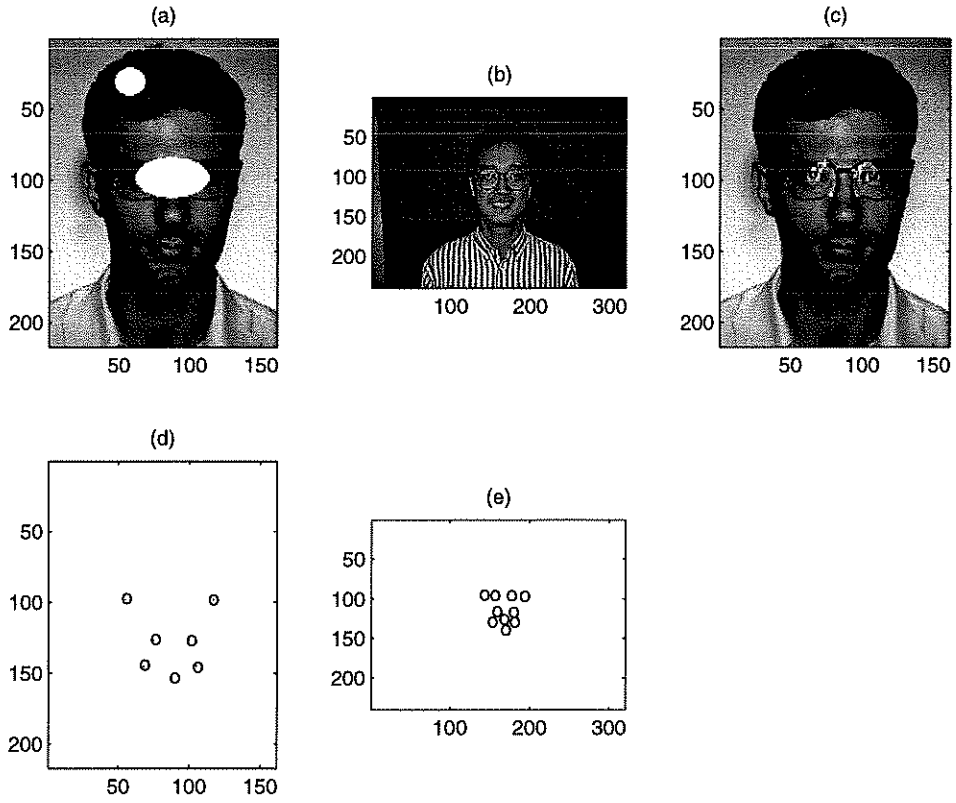


Figure 5.17: Application to a real image. (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note, I_1 and I_2 are photos of two different people. However, information from I_2 can be used to recover I_1.

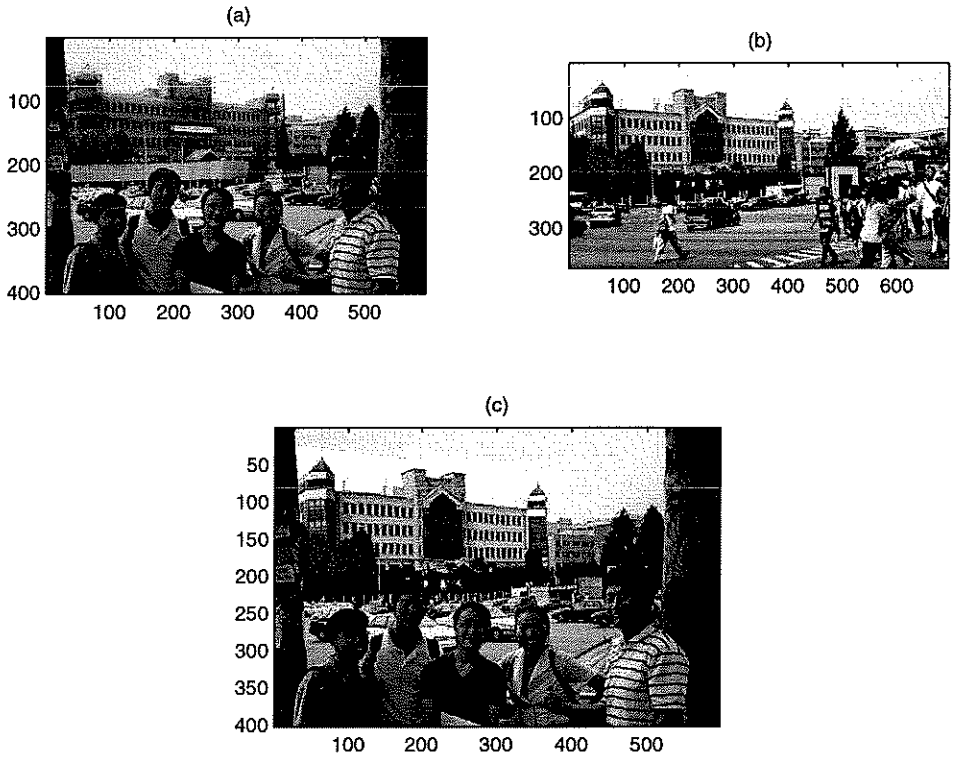


Figure 5.18: (a) I_1, (b) I_2, (c) recovered result. I_1 is from the summer of 1998 when the building was under construction, and I_2 was taken in the summer of 2001 when the construction is completed. By keeping the people from the old photo, only the building is successfully replaced by the new building. We used the whole building as inpainting domain in I_1.

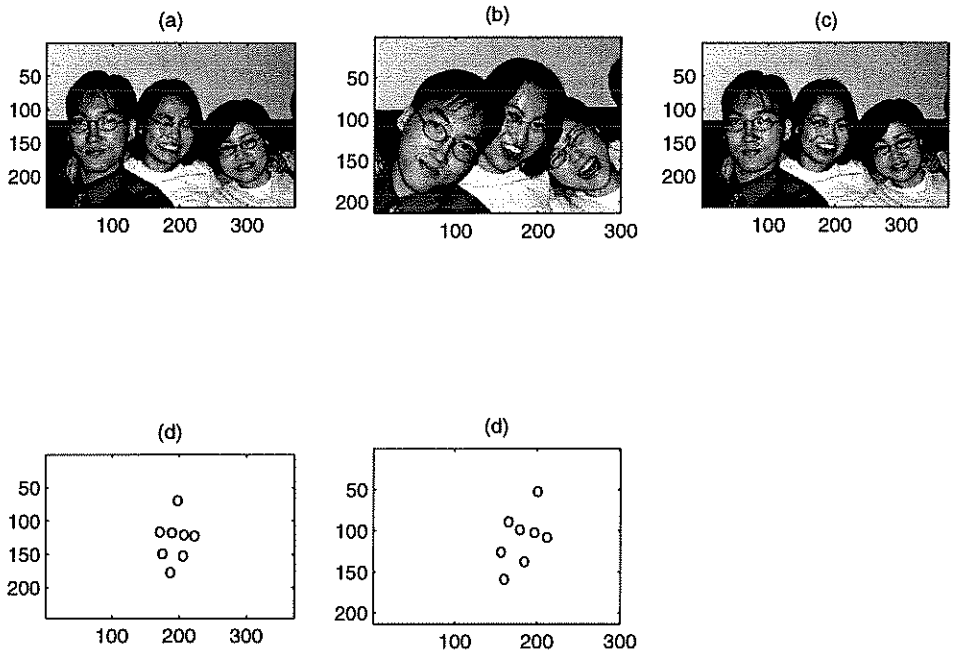


Figure 5.19: Some photos with odd facial expression can be modified. The middle person in I_1 (a), is modified using the other image with smiling face I_2 in (b). (c) shows the result which is modified from I_1 middle person is now smiling. We used the whole face as inpainting domain in I_1.

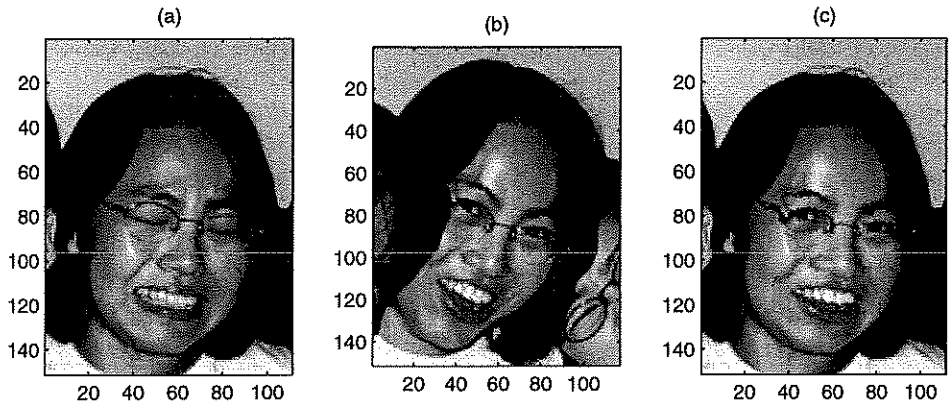


Figure 5.20: This is a close-up of Fig. 5.19, showing detailed recovered result. The mask refilled was the whole face. Comparing I_1(a) and (c) result, the eyes are opened and the mouth is modified according to I_2 in (b).

CHAPTER 6

Error Analysis for Inpainting Problems

6.1 Introduction

Ever since digital inpainting was proposed, many successful studies followed on different approaches and new methods. However, most inpainting domain is narrow or small enough to get good results and the the local inpainting results depend on the shape of the inpainting domain. As shown in Fig. 6.1, the two images (a) and (b) have same total area of inpainting domain (10875 pixels for each inpainting domain), however, the result for the local inpainting results are quite different. (c) is successfully recovered, while (d) is blurry even after many more iterations. (This experiment is done using TV inpainting method (4.1).) The inpainted results are depended on the local shape of the inpainting domain.

In addition, the error bound is highly dependent on the image space u is assumed to be in. We investigate two cases, first assuming image space is smooth, $u \subset C^2$, and using Harmonic inpainting and secondly for for binary image u with TV inpainting. For the smooth function, we use Green's function and show the error is depended on the local thickness. For the binary function u, we use level line analysis to show the error bound is depend on the distance between the level lines.

The outline of the chapter is as follows. In Section 6.2, we investigate the the case for smooth function, $u \subset C^2$ and harmonic inpainting, with review of some

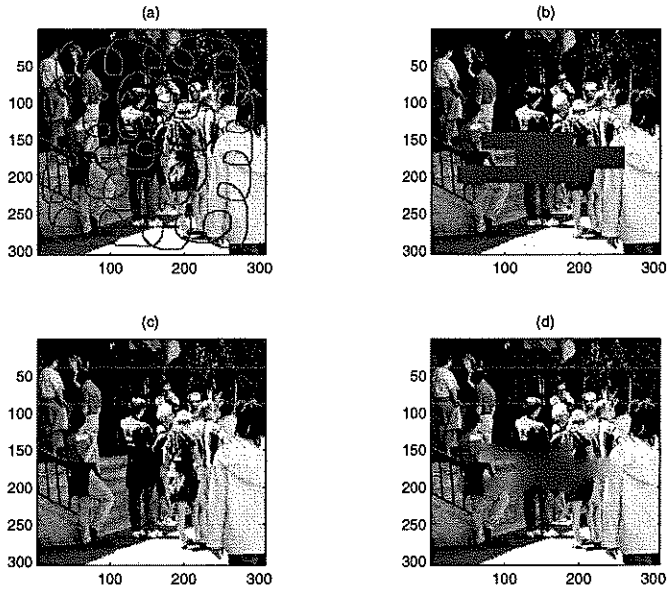


Figure 6.1: (a) and (c) gray area are the inpainting domains, both have exactly same area of inpainting domain (Pixel of 10875) (c) and (d) are the inpainted result of (a) and (b) respectively. (d) needed much more iteration than (c), still the result is not as good as (c). This due to the shape of the inpainting domain.

properties of Green's function. In Section 6.3, we will consider binary image u by level line analysis, followed by conclusion in Section 6.4.

6.2 Smooth Function $u \in C^2$, Harmonic Inpainting

Let u be a smooth image function defined on a 2-D domain Ω, and D be the inpainting domain. Let Δ be the Laplacian operator, $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$. Then for C^2 functions u and v, the *Green's Second Formula* on D is

$$\int_D (u\Delta v - v\Delta u)dxdy = \int_{\partial D} (u\frac{\partial v}{\partial \eta} - v\frac{\partial u}{\partial \eta})ds \tag{6.1}$$

where, η is outward normal direction of ∂D, and s arc length parameter of ∂D.

Let $G(z_o, z) = -v(z)$ be the **Green's function** for the grounded Poisson equation on D. Then G satisfies,

$$-\Delta G = \delta(z - z_o), \quad G\Big|_{\partial D} = 0.$$

Then by (6.1), Green's second Formula,

$$\int_D (u\Delta(-G) - (-G)\Delta u)dxdy = \int_{\partial D} (u\frac{\partial(-G)}{\partial \eta} - (-G)\frac{\partial u}{\partial \eta})ds$$

$$u(z_o) = \int_{\partial D} u(z(s))\frac{\partial(-G(z_o, z))}{\partial \eta}ds + \int_D G(z_o, z)(-\Delta u(z))dz \qquad (6.2)$$

For smooth function u, we assume we use Harmonic inpainting, which is using variational model with $\mathcal{H}^1$ regularization,

$$min \int_D \|\nabla v\|^2 dx \;\; such\ that \;\; v|_{\partial D} = u|_{\partial D}.$$

On the other hand, the first term in (6.2) is exactly the harmonic extension of u along the boundary ∂D, which is equivalent to harmonic inpainting result. Therefore, for harmonic inpainting in 2D, the first term doesn't have any error and the error bound only comes from second anti-harmonic term [CS01a]. Using the Lemmas and Theorem from [CS01a], we farther investigate the property of the second term, $\int_D G(z_o, z)(-\Delta u(z))dz$.

Lemma 1 *(Comparison Lemma) Suppose $D_1 \subset D_2$, and $G_1(z_o, z)$ and $G_2(z_o, z)$ are their associated Green's functions. Then for all $z_o, z \in D_1$,*

$$G_1(z_o, z) \leqq G_2(z_o, z).$$

Lemma 2 *Suppose B_1 is the unit disk centered at 0, and $G_1(z_o, z)$ its Green's function. Then*

$$\int_{B_1} G_1(z_o, z)dxdy = \frac{1 - |z_o|^2}{4},$$

for all $z_o \in B_1$.

Theorem 1 *Let d denote the diameter of a domain D and $G(z_o, z)$ the associated Green's function for the Poisson equation. Then*

$$\int_D G(z_o, z)dxdy \leqq \frac{d^2}{4}.$$

This theorem is proved using Lemma 1, 2. In addition, since u is smooth function, there exits M such that $|\Delta u(z)| \leqq M$, for all $z \in D$. Therefore, for harmonic inpainting in D, Chan and Shen [CS01a] showed, for any $z_o \in D$,

$$\|u_D - u^h\| \leqq M \int_D G(z_o, z)dz \leqq \frac{Md^2}{4}. \tag{6.3}$$

However, this d in equation(6.3) is the diameter of the inpainting domain which is big enough to cover whole inpainting domain. Therefore, for the case like Fig. 6.2, image (b) have bigger d then that of (c), while the inpainted result (d) looks better in (e). (d) is after 10 iteration, while (e) is after 2000 iteration. (2392 pixels for each inpainting domain.)

Therefore, we refine the error depending on the local shape of the inpainting domain. First, by using ellipse, we prove when D can be covered with ellipse, the error bound can be reduced.

Lemma 3 *Suppose B_2 is an ellipse centered at 0, $\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1$, where $b > a$ (otherwise rotate), and $G_2(z_o, z)$ be it's Green's function. Then*

$$\int_{B_2} G_2(z_o, z)dxdy \leqq \frac{a^2 - y_o^2}{2}$$

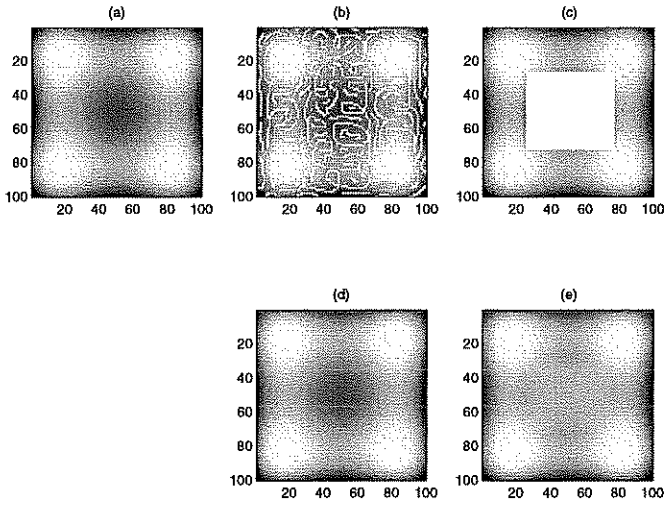
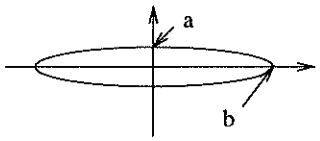


Figure 6.2: Example of Harmonic extension: (a) Original image. In (b) and (c), the white area are the inpainting domain. (d) and (e) are inpainting results for (b) and (c), respectively. (d) have better result than (e) compared to original image in (a).



Proof 1 *Consider a Poisson equation on ellipse* B_2,

$$-\Delta u = 1, \quad u\Big|_{\partial B_2} = 0.$$

Then the unique solution becomes,

$$u(z) = \frac{1 - \frac{x^2}{b^2} - \frac{y^2}{a^2}}{\frac{2}{b^2} + \frac{2}{a^2}} = \frac{a^2 - \frac{a^2}{b^2}x^2 - y^2}{2\frac{a^2}{b^2} + 2}. \tag{6.4}$$

Meanwhile, from Green's function,

$$u(z_o) = \int_{B_2} G_2(z_o, z)(-\Delta u(z))dxdy = \int_{B_2} G_2(z_o, z)dxdy.$$

Therefore,

$$\int_{B_2} G_2(z_o, z)dxdy = \frac{1 - \frac{x_o^2}{b^2} - \frac{y_o^2}{a^2}}{\frac{2}{b^2} + \frac{2}{a^2}} \leqq \frac{1 - \frac{y_o^2}{a^2}}{\frac{2}{a^2}} = \frac{a^2 - y_o^2}{2}. \quad \square$$

In addition, as $b \to \infty$, from Equ. 6.4, $u(z) \to \frac{a^2 - y_o^2}{2}$, and as $b \to a$, ellipse becomes circle, $u(z_o) \to \frac{a^2 - |z_o|^2}{4}$ which is as Lemma 2.

The solution $u(z)$ in (6.4), decreases faster in y direction than in x direction. Since $b > a$ and

$$\frac{\partial u}{\partial x} = \frac{-2\frac{x}{b^2}}{\frac{2}{b^2} + \frac{2}{a^2}}, \quad \frac{\partial u}{\partial y} = \frac{-2\frac{y}{a^2}}{\frac{2}{b^2} + \frac{2}{a^2}}.$$

Thus, u decreases faster in y-direction by factor of $\frac{b^2}{a^2}$. Therefore, the value $u(z_o)$ (where $z_o = (x_o, y_o)$) is more depended on the y_o then value x_o.

Therefore, for the bound for the ellipse shape inpainting domain, the bound is as follows.

Theorem 2 *Given inpainting domain D, for any $p \in D$, find the ellipse which covers D with maximum aspect ratio between a and b, $(b > a)$. Then,*

$$\int_D G(z_o, z)dxdy \leqq \frac{a^2}{2}$$

Proof 2 *Let B_e be an ellipse covering D with maximum aspect ratio between a and b, (b¿a). Let Green's function for B_e be G_e, and assume $p = 0$ for simplicity, then*

$$\begin{aligned} \int_D G(z_o, z)dxdy &\leqq \int_{b_e} G(z_o, z)dxdy (by\ Lemma\ 1) \\ &\leqq \frac{a^2 - y_o^2}{2} (by\ Lemma\ 3) \\ &\leqq \frac{a^2}{2}. \quad \square \end{aligned}$$

Therefore, with $|\Delta u(z)| \leqq M$ for all $z \in D$, the error becomes

$$\|u_D - u^h\| \leqq M \int_D G(z_o, z)dz \leqq \frac{Ma^2}{2}.$$

For inpainting domain D like ellipse the error is bounded by $\mathcal{O}(a^2)$.

6.3 Binary Function u, TV Inpainting

The image is generally thought to lie on *Bounded Variation* with discontinuities and jumps. This case, the error bound is always the area of the whole inpainting domain, since there could be an object totally hidden by the inpainting domain.

However, not considering the hidden factor, we can consider an error bound which could be reduced from information around the inpainting domain. Let F be extended domain outside of D, such that $F \cap D = \partial D$, Fig 6.3. We are

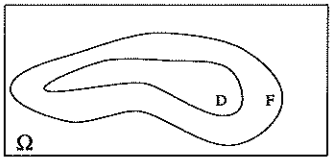


Figure 6.3: Inpainting domain D and extended domain F

interested in getting a bound, $\mathcal{E}(F)$, which is not considering the hidden factor, and depended on information from domain F.

$$\|u_{inpainted}(D) - u_{true}(D)\| \leqq \mathcal{E}(F) \leqq (I_{max} - I_{min})|D|$$

$|D|$ is the area of the inpainting domain, and $(I_{max} - I_{min})|D|$ is error of the hidden factor. Notice, that the error is no longer point-wise, but for whole image, i.e. difference in inpainting domain D.

For the TV inpainting method [CS01a], the error is the area from the shortest distance to the intersection of the two lines following the curvature of the level lines. Since, minimizing total variation will result in getting shortest distance, however, the original object and extend as far as the intersection of two level lines. It is illustrated in Fig. 6.4, dashed triangle is the area of error.) Some of analysis for the perception can be found at [CS01a]. In Fig. 6.4, (a) and (b) have

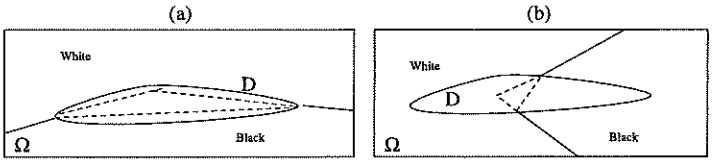


Figure 6.4: Example of the error in inpainting domain for piece-wise constant function.

same inpainting domain, however, unlike the case for smooth function, (a) and (b) have different error.

The error of the inpainting domain is from shortest distance to the intersection of the curvature extension. For straight line case, the area of error is calculated using vectors and distance. In Fig. 6.5, $\vec{n_1}$ is unit tangent vector from point p, $\vec{n_2}$ for point q, and $\vec{l}$ be unit vector in $q-p$ direction.

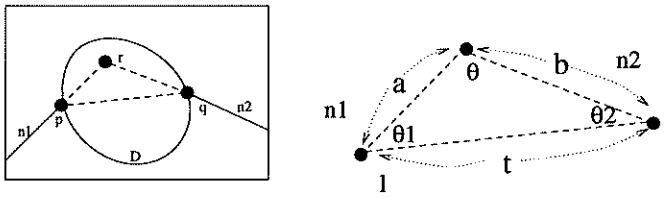


Figure 6.5: Dotted triangle is the possible error.

Then the area of the triangle is,

$$\begin{aligned}
\tfrac{1}{2}ab\sin\theta &= \tfrac{1}{2}ab|\vec{n_1}\times\vec{n_2}| \\
&= \tfrac{1}{2}\left(\tfrac{t\sin\theta_2}{\sin\theta}\right)\left(\tfrac{t\sin\theta_1}{\sin\theta}\right)|\vec{n_1}\times\vec{n_2}| \\
&\qquad \left(by\ \ \tfrac{t}{\sin\theta}=\tfrac{a}{\sin\theta_2}=\tfrac{b}{\sin\theta_1}\right) \\
&= \tfrac{1}{2}\left(\tfrac{t|\vec{n_2}\times\vec{l}|}{|\vec{n_1}\times\vec{n_2}|}\right)\left(\tfrac{t|\vec{n_1}\times\vec{l}|}{|\vec{n_1}\times\vec{n_2}|}\right)|\vec{n_1}\times\vec{n_2}| \\
&= \tfrac{1}{2}\left(\tfrac{|\vec{n_2}\times\vec{l}||\vec{n_1}\times\vec{l}|}{|\vec{n_1}\times\vec{n_2}|}\right)t^2
\end{aligned}$$

Therefore, the error becomes,

$$\mathcal{E}(F) = \frac{1}{2}(\frac{|\vec{n_2} \times \vec{l}||\vec{n_1} \times \vec{l}|}{|\vec{n_1} \times \vec{n_2}|}t^2)(I_{max\ \partial D} - I_{min\ \partial D})$$
$$\leqq (I_{max} - I_{min})|D|.$$

For the case, when vectors $\vec{n_1}$ and $\vec{n_2}$ have the same direction as l, then $|\vec{n_1} \times \vec{l}| = 0$, therefore $\mathcal{E} = 0$. And for the case, the two level lines at the boundary is the same if the distance between the two level lines are far, big t, then the error is bigger. The error term for binary function u is bounded by $\mathcal{O}(t^2)$, where this t is the minimum distance between two level line. In addition, the error depends on the direction $\vec{n_1}$ and $\vec{n_2}$.

6.4 Conclusion

We study the error bound for the inpainting domain for two space, for smooth function $u \in C^2$, Harmonic inpainting and binary function u, TV inpainting. We show the explicit form for each error bound.

First for the smooth function, harmonic inpainting case, when the inpainting domain can be covered by ellipse, the error is bounded by the minimum axis a with $\mathcal{O}(a^2)$. The error term have been reduced for general case using d and specially for inpainting domain with narrow region.

As for the binary image u and TV inpainting, the error is bounded by the shortest distance, t, between two level lines and and $\vec{n_1}$, $\vec{n_2}$ direction of the two level lines. Even if the inpainting domain is narrow with big aspect ratio, if the distance between two level lines are large, the error could be bigger. The error is bounded by $\mathcal{O}(t^2)$. However, for binary case, there always could be a hidden factor, therefore, the error term can not be any smaller than total area of the inpainting domain.

References

[BB95] S. S. Beauchemin and J. L. Barron. "The Computation of Optical Flow." *ACM Computing Surveys*, **27**(3):433–467, 1995.

[BBC01] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Vergera. "Filling-In by Joint Interpolation of vector Fields and Gray levels." *IEEE Trans. Image Processing*, **10**(8):1200–1211, 2001.

[BC98] P. V. Blomgren and T. F. Chan. "Color TV: Total Variation Methods for Restoration of Vector Valued Images." *IEEE Trans. Image Processing*, **7**(3):304–309, 1998.

[BIK01] A. Borzi, K. Ito, and K. Kunisch. "Optimal Control Formulation for Determining Optical Flow." *preprint*, 2001.

[Blo98] P. V. Blomgren. "Total Variation Methods for Restoration of Vector Valued Images, (Ph.D. thesis)." *UCLA Dept. of Math. CAM 98-30*, 1998.

[BMP01] S. Belongie, J. Malik, and J. Puzicha. "Matching Shapes." *Eight IEEE Int. Conf. on Computer Vision*, July 2001.

[BSC00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Balleste. "Image Inpainting." *Proceedings of SIGGRAPH 2000, New Orleans, USA*, 2000.

[CCB01] L. Chanas, J.P. Cocquerez, and J. Blanc-Talon. "Highly degraded sequences restoration and inpainting." *preprint*, 2001.

[CKS01a] T. F. Chan, S. H. Kang, and J. Shen. "Euler's Elastica and Curvature Based Inpaintings." *UCLA Dept. of Math. CAM 01-12*, 2001.

[CKS01b] T. F. Chan, S. H. Kang, and J. Shen. "Total Variation Denoising and Enhancement of Color Images Based on the CB and HSV Color Models." *Journal of Visual Communication and Image Representation*, **12**:422–435, 2001.

[CKS02] T. F. Chan, S. H. Kang, and S. Soatto. "Landmark based Inpainting from Multiple Views." *UCLA Math dep, CAM 02-11*, 2002.

[CL97] A. Chambolle and P. L. Lions. "Image recovery via Total Variational minimization and related problems." *Numer. Math.*, **76**:167–188, 1997.

[COS01] T. F. Chan, S. Osher, and J. Shen. "The Digital TV filter and Nonlinear Denoising." *IEEE Trans. Image Processing*, **10**(2):231–241, 2001.

[CS00a] T. F. Chan and J. Shen. "Rotational Invariance for Prior Models in Image Analysis." *Let. Applied Math.*, submitted, 2000.

[CS00b] T. F. Chan and J. Shen. "Variational Restoration of Non-Flat Image Features: Models and Algorithms." *SIAM Journal of Applied Mathematics*, **61**(4):1338–1361, 2000.

[CS01a] T. F. Chan and J. Shen. "Mathematical Models for Local Deterministic Inpaintings." *SIAM Journal of Applied Mathematics*, **62**(3):1019–1043, 2001.

[CS01b] T. F. Chan and J. Shen. "Morphology Invariant PDE Inpaintings." *UCLA Dept. of Math. CAM 01-15*, 2001.

[CS01c] T. F. Chan and J. Shen. "Non-Texture Inpainting by Curvature-Driven Diffusion (CDD)." *Journal of Visual Communication and Image Representation*, **12**(4):436–449, 2001.

[CV01] T. Chan and L. Vese. "Active Contours without Edges." *IEEE Trans. Image Processing* , **10**(2):266–277, 2001.

[FB81] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." *Communications of the ACM*, **24**(6):381–395, 1981.

[Gui98] F. Guichard. "A morphological, Affine and Galilean Invariant Scale-Space for Movies." *IEEE Trans. Image Process.*, **7**(3):444–456, 1998.

[GW92] R. Gonzalez and R. Wood. *Digital Image Processing.* Addison-Wesley, 1992.

[HS81] B.K.P. Horn and B. G. Schunck. "Determining Optical Flow." *Artificial Intelligence*, **17**:185–203, 1981.

[KD92] J. Konrad and E. Dubois. "Bayesian Estimation of Motion Vector Fields." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **14**(9):910–927, 1992.

[KMF95a] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. "Detection of Missing Data in Image Sequences." *IEEE Trans. Image Processing*, **11**(4):1496–1508, 1995.

[KMF95b] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. "Interpolation of Missing Data in Image Sequences." *IEEE Trans. Image Processing*, **11**(4):1509–1519, 1995.

[KS02] R. Kimmel and N. Sochen. "Orientation Diffusion or How to Comb a Porcupine ?" *Special issue on PDEs in Image Processing, Computer Vision, and Computer Graphics, J. Visual Comm. Image Representation*, **13**:238–248, 2002.

[KTB96] A. Kumar, A. R. Tannenbaum, and G.J. Balas. "Optical Flow: A Curve Evolution Approach." *IEEE Trans. Image Processing*, **5**(4):598–610, 1996.

[KWM99] R. Krishnamurthy, J. W. Woods, and P. Moulin. " Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields." *IEEE Trans. on Circuits and Systems for Video Technology*, **9**(5):713 –726, 1999.

[LKW94] M. R. Luettgen, W. C. Karl, and A. S. Willsky. "Efficient Multiscale Regularization with Applications to the Computation of Optical Flow." *IEEE Trans. Image Processing*, **3**(1):41–64, 1994.

[Lov27] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity.* Dover, New York, 1927.

[Mas00] S. Masnou. "Variational Formulation of the Disocclusion problem." *The abdus salam, int'l centre for theoretical physics, school on mathematical problems in image processing*, 2000.

[MM98] S. Masnou and J. M. Morel. "Level Lines based Disocclusion." *Proceedings of 5th IEEE Int'l Conf. on Image Process.,Chicago*, **3**:259–263, 1998.

[MO99] A. Marguina and S. Osher. "Explicit Algorithms for a New Time Dependent Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal." *UCLA Dept. of Math. CAM 99-5*, 1999.

[MS95] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*, volume 14 of *Progress in Nonlinear Differential Equations and Their Applications.* Birkhäuser, Boston, 1995.

[PD99] N. Paragios and R. Deriche. "Geodesic Active Regions for Motion Estimation and Tracking." *Proceedings of IEEE Int'l Conf. on Computer Vision*, **1**:688–694, 1999.

[Per98] P. Perona. "Orientation diffusion." *IEEE Trans. Image Processing*, **7**(3):457–467, 1998.

[Pow90] M. J. D. Powell. "The Theory of Radial Basis Function Approximation." *Advances in Numerical Analysis*, **II**:105–210, 1990.

[Pow95] M.J.D. Powell. "A thin Plate spline method for mapping curves into curves in two dimensions." *Computational Techniques and Applications (CTAC95)*, 1995.

[RO94] L. Rudin and S. Osher. "Total variation based image restoration with free local constraints." *Proc. 1st IEEE ICIP*, **1**:31–35, 1994.

[ROF92] L. Rudin, S. Osher, and E. Fatemi. "Nonlinear Total Variation based Noise Removal Algorithm." *Physica D*, **60**:259–268, 1992.

[SB95] S. M. Smith and J. M. Brady. "ASSET-2: real-time motion segmentation and shape tracking." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **17**(8):814 –820, 1995.

[SKM98] N. Sochen, R. Kimmel, and R. Malladi. "A geometrical framework for low level vision." *IEEE Trans. Image Processing*, **7**(3):310–318, 1998.

[SR96] G. Sapiro and D. Ringach. "Anisotropic diffusion of multivalued images with applications to color filtering." *IEEE Trans. Image Processing*, **5**:1582–1586, 1996.

[TKV96] P. E. Trahanias, D. Karako, and A. N. Venetsanopoulos. "Directional processing of color images: theory and experimental results." *IEEE Trans. Image Processing*, **5**(6):868–880, 1996.

[TSC99] B. Tang, G. Sapiro, and V. Caselles. "Color image enhancement via Chromaticity Diffusion." Technical report, ECE University of Minnesota, 1999.

[TSC00] B. Tang, G. Sapiro, and V. Caselles. "Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case." *Int. Journal Computer Vision*, **36**(2):149–161, 2000.

[TV93] P. E. Trahanias and A. N. Venetsanopoulos. "Vector directional filters — a new class of multichannel image processing filters." *IEEE Trans. Image Processing*, **2**(4):528–534, 1993.

[Wah90] G. Wahba. "Spline Models for Observational Data." *SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia*, **59**, 1990.

[Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner-Verlag, Stuttgart, Germany, 1998.