Reconstruction of High-Resolution Image from Movie Frames

by

Ling Kai Tung

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Mathematics

August 2003

Abstract

Abstract of thesis entitled:

Reconstruction of High-Resolution Image from Movie Frames

Submitted by Ling Kai Tung

for the degree of Master of Philosophy in Mathematics
at The Chinese University of Hong Kong in August 2003

Image reconstruction is generally an inverse problem, which intends to recover the original ideal image from its given degraded version. Many methods have been proposed to reconstruct a high-resolution image from several undersampled, shifted, degraded still images with subpixel displacements. This thesis will demonstrate the reconstruction of high-resolution image from low-resolution movie frames. From the movement of the video camera, we can obtain the target object located in the different positions in movie frames. With these captured movie frames, we can get different still images with displacements. We align the target object in the still images so that they are shifted from each other by approximately one pixel and these images serve as the input images for existing methods. This technique is illustrated with some examples. We can see from these examples that we can reconstruct an improved image from the movie frames. The result can be further improved if we can estimate the exact subpixel movement of the target object and we would like to apply blind deconvolution to achieve this task.

摘要

從電影畫格重構高解析度圖片

高解析度圖片重構普遍是一個逆問題，是希望從一些低解析度的圖片中重構一幅高解析度圖片。很多方法利用一些低解析度並且有非整像素移位的圖片作重構。此論文將會示範從電影畫格重構高解析度圖片。由鏡頭的移動，我們可以獲得目標事物位於畫格中不同的位置，從這些畫格中，我們欲將重構之事物截取並作特定的排位，令它們大概與其他截取的事物有一像素之距離，並利用現有的方法應用到這些圖片上，我們會給予例子去示範。從這些例子中，我們可見重構之圖片比原來的圖片爲佳，我們更會嘗試去估計實際移位的距離，並希望重構出更佳之高解析度圖片。

ACKNOWLEDGMENTS

I am greatly indebted to my supervisor, Prof. Raymond H. Chan for his continual guidance and constant encouragement throughout the period of my postgraduate studies. I would like to thank Profs. X.Q. Jin, Michael Ng, my colleagues Mr. Z.J. Bai, Y. Chen, H.L. Chung, C.W. Ho, C. Hu, K.C. Ma, Y.H. Tam and C.Y. Wong for their helpful discussions and encouragements.

Contents

Chapter 1

Introduction

In this thesis, we consider the reconstruction of high-resolution images from several degraded low-resolution frames obtained from a movie. These low-resolution frames are shifted from each other by sub-pixel displacements in order to gather enough information for the reconstruction of the high resolution images.

First, we will have a brief introduction to what a digital image is and what is the differences between high-resolution and low-resolution images. Some examples are illustrated and a common phenomenon of image defect called "blurring" is introduced at the end of Chapter 2.

Image reconstruction problem is generally an inverse problem, a least-squares approach is often used to solve the problem. In Chapter 3, we survey some of the common methods for solving linear and nonlinear least-squares problems.

In Chapter 4, we will use the model proposed by [2] since they required the degraded low-resolution images to be shifted by sub-pixel displacements and these displacements can be obtained by the movement of the target object in a movie. Although the sub-pixel displacements may not be the ideal one, we can see from the illustrated results that improvement of the quality of the image is achieved.

Since the model introduced in Chapter 4 is not the ideal one, and the sub-pixel displacements we obtained is not exact, there may be some errors in it. In Chapter 5, we try to impose the errors into the model and we will use the model proposed by [5] to see whether more improvement of the reconstructed image can be made.

Chapter 2

Fundamentals

2.1 Digital image representation

A digital image is a 2-D array of integers. It is an approximation to the continuous image. Each element of the array is called a pixel and the value of each element is called a pixel value.

For a monochrome image, it consists of one array of integers and these integers represent the gray-levels, see Figure 2.1. For color images, there are different kind of representations. The most common one is the **RGB** representations, it represents a color image using three arrays of integers. The first, second and the third arrays represent the intensities of **R**ed, **G**reen and **B**lue color respectively, see Figure 2.2. For other representations of a color image, please refer to [1].

The degree of discernible detail of an image is controlled by the image's resolution. The higher the resolution, more points are taken to approximate the continuous image and the larger the size of the array(s), see Figures 2.3 and 2.4. As a result, the storage and processing requirements increase rapidly as resolution increases.

Figure 2.1: A gray-level image with some pixel values listed.

Figure 2.2: A color image (left) and its different channels (right).

Figure 2.3: The same scene using 64×64 resolution (left), 128×128 resolution (center) and 256×256 resolution (right).

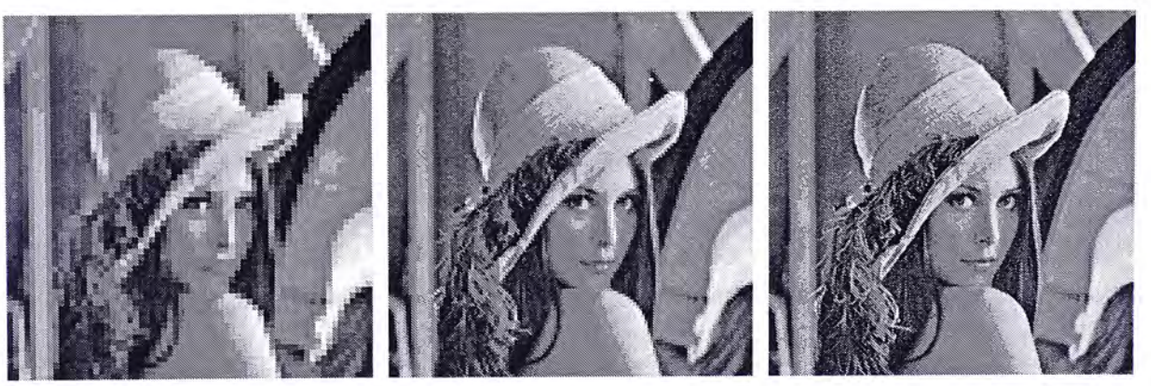

Figure 2.4: The same scene using 64×64 resolution (left), 128×128 resolution (center) and 256×256 resolution (right) with the first 2 images resampled to 256×256 pixels.

$$f(x,y,z) = \begin{pmatrix} f(1,1,1) & f(1,2,1) & \dots & f(1,N,1) \\ f(2,1,1) & f(2,2,1) & \dots & f(2,N,1) \\ \vdots & \vdots & & \vdots \\ f(M,1,1) & f(M,2,1) & \dots & f(M,N,1) \end{pmatrix} .2)$$
$$\begin{pmatrix} f(2,1,2) & f(2,2,2) & \dots & f(2,N,2) \\ \vdots & \vdots & & \vdots \\ f(M,1,2) & f(M,2,2) & \dots & f(M,N,2) \end{pmatrix} .3)$$
$$\begin{pmatrix} f(2,1,3) & f(2,2,3) & \dots & f(2,N,3) \\ \vdots & \vdots & & \vdots \\ f(M,1,3) & f(M,2,3) & \dots & f(M,N,3) \end{pmatrix}$$

Figure 2.5: Matrix representation of a color image.

In mathematical language, a gray-level image is a matrix with entries representing the intensities of the gray-level at different pixels. Similarly, a color image using RGB representation consists of three matrices with each matrix representing the intensities of each color. A monochrome image can be written in the following matrix form:

$$f(x,y) = \begin{pmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{pmatrix}.$$

A color image can be written similarly with three matrices instead of one matrix, see Figure 2.5.

Figure 2.6: A motion blurred image.

2.2 Motion Blur

Motion blur is a common image defect. It occurs when the exposure time of a camera is longer than the optimal one. Because the shutter remains open for an extended period of time, all the motion occurred during this time interval is recorded in the image and makes the image look hazy, see Figure 2.6.

We can model the motion blur phenomenon by using a sequence of shifted images (see Figure 2.7) and then take the average of these images.

$$\frac{f_1 + f_2 + \ldots + f_n}{n} = \bar{f}$$

Figure 2.7: Part of the image sequence that Figure 2.6 is manufactured from and they are shifted from each other by a pixel to the right.

Chapter 3

Methods for Solving Nonlinear Least-Squares Problem

3.1 Introduction

In image processing problems, one always have to solve linear or nonlinear least-squares problem. In this chapter, we are going to survey some of the methods for solving unconstrained and constrained nonlinear least-squares problems. For details analysis of the methods introduced in this chapter, see [7] and [8].

3.2 Nonlinear Least-Squares Problem

The nonlinear least-squares problem is

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \quad f(\mathbf{x}) = \frac{1}{2}R(\mathbf{x})^t R(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} r_i(\mathbf{x})^2$$

where $m > n$, $R(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \cdots, r_m(\mathbf{x}))^t$ and $R(\mathbf{x})$ is nonlinear in $\mathbf{x}$.

This kind of problem arises commonly in data-fitting applications. In data-fitting problem, we have a set of data (t_i, y_i), $i = 1, 2, \cdots, m$ and we would like to fit a model $k(\mathbf{x}, t)$ that is nonlinear in $\mathbf{x}$ such that the distance between the points y_i and $k(\mathbf{x}, t_i)$, $i = 1, 2, \cdots, m$ is minimized. If we let $r_i(\mathbf{x}) = k(\mathbf{x}, t_i) - y_i$, then the problem becomes

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \quad f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} r_i(\mathbf{x})^2 = \frac{1}{2}R(\mathbf{x})^t R(\mathbf{x}).$$

3.3 Gauss-Newton-Type Methods

3.3.1 Gauss-Newton Method

The Gauss-Newton methods use a linear model to approximate $R(\mathbf{x})$ in a neighborhood of a given point $\mathbf{x_c}$,

$$M_c(\mathbf{x}) = R(\mathbf{x_c}) + J(\mathbf{x_c})(\mathbf{x} - \mathbf{x_c})$$

where $J(\mathbf{x})$ is the Jacobian matrix of $R(\mathbf{x})$. Then we can use the linear least-squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|R(\mathbf{x_c}) + J(\mathbf{x_c})(\mathbf{x} - \mathbf{x_c})\|_2^2 \tag{3.1}$$

and assume $J(\mathbf{x_c})$ has full column rank, then the minimizer of the above problem is given by

$$\mathbf{x} = \mathbf{x_c} - (J(\mathbf{x_c})^t J(\mathbf{x_c}))^{-1} J(\mathbf{x_c})^t R(\mathbf{x_c}).$$

The method described above is linear and requires the first derivative of $R(\mathbf{x})$ at $\mathbf{x_c}$ only. It has fast local convergence on mildly nonlinear and nearly consistent problems, but it may not converge locally for highly nonlinear problems.

3.3.2 Damped Gauss-Newton Method

Sometimes the Gauss-Newton method takes too long steps length ($J(\mathbf{x_c})(\mathbf{x}-\mathbf{x_c})$) which makes the approximated solution diverges away from the exact solution. In order to control the step length, a parameter α_k is added to (3.1) in order to control the step length.

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|R(\mathbf{x_c}) + \alpha_k J(\mathbf{x_c})(\mathbf{x}-\mathbf{x_c})\|_2^2$$

The minimizer of the above problem is

$$\mathbf{x}-\mathbf{x_c} = -\alpha_k(J(\mathbf{x_c})^t J(\mathbf{x_c}))^{-1} J(\mathbf{x_c})^t R(\mathbf{x_c}),$$

First, we calculate the descent direction by

$$\mathbf{x}-\mathbf{x_c} = -(J(\mathbf{x_c})^t J(\mathbf{x_c}))^{-1} J(\mathbf{x_c})^t R(\mathbf{x_c}),$$

and then α_k is obtained by solving the following minimization problem

$$\min_{\alpha_k} \|r(\mathbf{x_k} + \alpha_k(\mathbf{x}-\mathbf{x_c}))\|_2^2.$$

A theoretical analysis of the above step length principle has been given in [6].

This method is locally convergent on almost all nonlinear least-squares problems, including problems which are higher nonlinear but the convergent rate is slow compare to Newton's method.

3.4 Full Newton-Type Methods

For the Full Newton-Type Methods, the approximation is based on Taylor series of $f(x)$ up to the quadratic term.

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{R}^n} f_c(\mathbf{x}), \quad f_c(\mathbf{x}) =& f(\mathbf{x_c}) + \nabla f(\mathbf{x_c})^t(\mathbf{x}-\mathbf{x_c}) + \frac{1}{2}(\mathbf{x}-\mathbf{x_c})^t\nabla^2 f(\mathbf{x_c})(\mathbf{x}-\mathbf{x_c}) \\ =& \frac{1}{2}R(\mathbf{x_c})^t R(\mathbf{x_c}) + R(\mathbf{x_c})^t J(\mathbf{x_c})(\mathbf{x}-\mathbf{x_c}) \\ &+ \frac{1}{2}(\mathbf{x}-\mathbf{x_c})^t(J(\mathbf{x_c})^t J(\mathbf{x_c}) + S(\mathbf{x_c}))(\mathbf{x}-\mathbf{x_c})\end{aligned}$$

where

$$S(\mathbf{x}) = \sum_{i=1}^{m} r_i(\mathbf{x}) G_i(\mathbf{x}), \quad G_i(\mathbf{x})_{jk} = \frac{\partial^2 r_i(\mathbf{x})}{\partial \mathbf{x}_j \partial \mathbf{x}_k}, \quad i = 1, \cdots, m.$$

The minimizer of the above problem is

$$\mathbf{x} = \mathbf{x_c} - (J(\mathbf{x_c})^t J(\mathbf{x_c}) + S(\mathbf{x_c}))^{-1} J(\mathbf{x_c})^t R(\mathbf{x_c}).$$

This method is locally quadratically convergent as long as

$$\nabla^2 f(\mathbf{x}) = J(\mathbf{x_c})^t J(\mathbf{x_c}) + S(\mathbf{x_c})$$

is Lipschitz continuous around $\mathbf{x_c}$, and $\nabla^2 f(\mathbf{x}_*)$ is positive definite, where $\mathbf{x}_*$ is the minimizer of $f(\mathbf{x})$, see [7] for details.

The convergent rate of this method is much superior than the previous methods, but it requires the calculation of $J(\mathbf{x_c})^t J(\mathbf{x_c}) + S(\mathbf{x_c})$ which is rarely available. If it is not analytically available, one should approximate $J(\mathbf{x_c})^t J(\mathbf{x_c}) + S(\mathbf{x_c})$ by finite difference which maybe very costly to do so.

3.4.1 Quasi-Newton methods

In quasi-Newton method, one have to calculate an approximation to the second derivative matrix successively by evaluations of the gradient.

Let S_{k-1} be a symmetric approximation to the Hessian at step k. S_k approximates the change of the gradient of f along the direction $\mathbf{x_k} - \mathbf{x_{k-1}}$, that is,

$$S_k(\mathbf{x_k} - \mathbf{x_{k-1}}) = y_k, \quad y_k = J(\mathbf{x_k})^t r(\mathbf{x_k}) - J(\mathbf{x_{k-1}})^t r(\mathbf{x_{k-1}}),$$

which is called the quasi-Newton relation.

The search direction $\mathbf{p_k}$ is then calculated by

$$S_k \mathbf{p_k} = -J(\mathbf{x_k})^t r(\mathbf{x_k}).$$

In common practice, the starting approximation of S_0 is taken as $S_0 = J(\mathbf{x_0})^t J(\mathbf{x_0})$.

3.5 Constrained problems

Sometimes the nonlinear least-squares problem may be subject to constraints. In case of nonlinear equality constraints, the problem can be stated as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2} R(\mathbf{x})^t R(\mathbf{x}), \quad \text{subject to} \quad h(\mathbf{x}) = 0,$$

where $R(\mathbf{x}) \in \mathbb{R}^m$, $h \in \mathbb{R}^p$, and $p < n$.

The Gauss-Newton method can be generalized to solve constrained problems by linearizing at a point $\mathbf{x_k}$. A line search direction $\mathbf{p_k}$ is computed as a solution to the linearly constrained problem

$$\min_{\mathbf{p}} \|r(\mathbf{x_k}) + J(\mathbf{x_k})\mathbf{p}\|_2 \quad \text{subject to} \quad h(\mathbf{x_k}) + C(\mathbf{x_k})\mathbf{p} = 0,$$

where J and C are the Jacobian matrices for $r(\mathbf{x})$ and $h(\mathbf{x})$, respectively. Then this problem can be solved by the method of weighting. The method of weighting is to assign a weight in the constraint and minimize it together with the original function, i.e.

$$\min_{\mathbf{p}} \|r(\mathbf{x_k}) + J(\mathbf{x_k})\mathbf{p}\|_2^2 + \alpha\|h(\mathbf{x_k}) + C(\mathbf{x_k})\mathbf{p}\|_2^2$$

By using this method the constrained least-squares problem becomes an unconstrained least-squares problem. This method is attractive because of its simplicity. It allows the use of programs for solving unconstrained least-squares problem to solve the constrained problems. But α should be carefully chosen, otherwise the system may becomes ill-conditioned.

Chapter 4

Reconstruction of High-Resolution Images from Movie Frames

4.1 Introduction

Electronic surveillance system has a wide-range of usage, for example to enhance security. Reconstruction of the high-resolution image from the recorded movie in this kind of systems is of great importance. In this chapter, we consider the reconstruction of high-resolution images from multiple undersampled, shifted, degraded and noisy color movie frames which are obtained from a movie of a target object. There are a lot of researches regarding the reconstruction of high-resolution images from low-resolution images taken by camera, see [10] and [11]. We would like to find a method that can use the low-resolution movie frames from video and reconstruct a high-resolution image from them. The difference between the source

images from camera and from movie frames is that the subpixel displacements can be controlled from camera, but not from movie frames.

We separated the color movie frames into a set of three images in their primary color components: red, green and blue. The reconstruction of high-resolution color images can be modeled as solving

$$\vec{g} = \mathcal{A}\vec{f} + \vec{\eta}, \tag{4.1}$$

where $\mathcal{A}$ is the reconstruction operator, $\vec{\eta}$ represents unknown Gaussian noise or measurement errors, $\vec{g}$ is the observed high-resolution color image formed from the low resolution color movie frames and $\vec{f}$ is the desired high-resolution color image. The observed and original color images can be expressed as

$$\vec{g} = \begin{pmatrix} g^{(r)} \\ g^{(g)} \\ g^{(b)} \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} f^{(r)} \\ f^{(g)} \\ f^{(b)} \end{pmatrix},$$

where $g^{(i)}$ and $f^{(i)}$ $(i \in \{r, g, b\})$ are the observed and the original color images from the red, green and blue channels respectively.

Since the system (4.1) is ill-conditioned and generally not positive definite, we solve it by using a minimization and regularization technique:

$$\min_f \left\{ \sum_{i\in\{r,g,b\}} \alpha_i \left\| \sum_{j\in\{r,g,b\}} \mathcal{A}^{ij} f^{(j)} - g^{(i)} \right\|_2^2 + \mathcal{R}(f^{(r)}, f^{(g)}, f^{(b)}) \right\}. \tag{4.2}$$

Here the operators $\mathcal{A}^{ii}$ and $\mathcal{A}^{ij} (i \neq j)$ are the entries of $\mathcal{A}$ and represent the within-channel and the cross-channel degradation operators respectively, $\mathcal{R}$ is a functional which measures the regularity of f, and the regularization parameter α_i is to control the degree of regularity of the solution for the i-th channel.

The main aim of this chapter is to extend the results in [2] from still images to movie frames. We apply the fast and stable image processing algorithm in [2] to the color image reconstruction problems.

The outline of this chapter is as follows. In Section 2, we give a mathematical formulation of the problem. We describe the acquisition of the movie frames in Section 3. In Section 4, experimental results are presented to demonstrate the effectiveness of our method. We have the concluding remarks in the last section.

4.2 The Mathematical Model

The mathematical model is the same as [2]. Although we are using movie frames rather than taking still images from a sensor array, when the movie is taken with sufficient frame rate, we can extract the movie frames with approximately the same subpixel displacement as taking still images from the sensor array by taking advantage of the movement of the target object in the movie. Therefore we can still use the model in [2], i.e., with a sensor array with $L_1 \times L_2$ sensors, each sensor has $N_1 \times N_2$ sensing elements, and the size of each sensing element is $T_1 \times T_2$. We would like to reconstruct a high-resolution image of resolution $M_1 \times M_2$ where $M_1 = L_1 \times N_1$ and $M_2 = L_2 \times N_2$. To maintain the aspect ratio of the reconstructed image, we only consider the case where $L_1 = L_2 = L$. For simplicity, we assume L is an even number.

Ideally, the sensors are shifted from each other by a value proportional to $T_1/L \times T_2/L$. For simplicity, we assume there is no imperfection of the imaging system, i.e., there are no perturbations around ideal subpixel location. Thus, for $l_1, l_2 = 0, 1, \ldots, L-1$ with $(l_1, l_2) \neq (0, 0)$, the horizontal and vertical displacements $d^x_{l_1 l_2}$ and $d^y_{l_1 l_2}$ are given by $d^x_{l_1 l_2} = \frac{T_1 l_1}{L}$ and $d^y_{l_1 l_2} = \frac{T_2 l_2}{L}$.

Let $f^{(r)}, f^{(g)}$ and $f^{(b)}$ be the original scene in red, green and blue channels respectively. The high-resolution reconstruction problem can be modeled as

$$g^{(i)} = \sum_{j \in \{r,g,b\}} w_{ij} \mathcal{H} f^{(j)} + \eta^{(i)}, \quad i \in \{r, g, b\}. \tag{4.3}$$

Here $g^{(i)}$ is the observed high-resolution image, $\mathcal{H}f^{(j)}$ is the formation of the low-resolution images. See [2] for more details. The parameter $\eta^{(i)}$ is the noise in the i-th channel, and w_{ii} and w_{ij} $(i \neq j)$ are the within-channel and the cross-channel degradation parameters. We note that

$$w_{ij} \geq 0, \quad i,j \in \{r,g,b\} \quad \text{and} \sum_{j=r,g,b} w_{ij} = 1, \quad i \in \{r,g,b\}. \tag{4.4}$$

4.2.1 The Discrete Model

For $i \in \{r,g,b\}$, let $\mathbf{g}^{(i)}$ and $\mathbf{f}^{(i)}$ be respectively the discretization of $g^{(i)}$ and $f^{(i)}$ using a column by column ordering. Let

$$\mathbf{g} = [\mathbf{g}^{(r)} \quad \mathbf{g}^{(g)} \quad \mathbf{g}^{(b)}]^t \quad \text{and} \quad \mathbf{f} = [\mathbf{f}^{(r)} \quad \mathbf{f}^{(g)} \quad \mathbf{f}^{(b)}]^t. \tag{4.5}$$

Under the Neunmann boundary condition assumption, the resulting matrices, denoted by $H^x_{l_1l_2}$ and $H^y_{l_1l_2}$ have a Toeplitz-plus-Hankel structure:

$$\mathbf{H}^x_{l_1l_2} = \frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & h^{x+}_{l_1l_2} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & \ddots & h^{x+}_{l_1l_2} \\ h^{x-}_{l_1l_2} & \ddots & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & & h^{x-}_{l_1l_2} & 1 & \cdots & 1 \end{pmatrix} + \frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & h^{x-}_{l_1l_2} & & 0 \\ \vdots & \iddots & \iddots & & & \\ 1 & \iddots & & & & h^{x+}_{l_1l_2} \\ h^{x-}_{l_1l_2} & & & & \iddots & 1 \\ & & & \iddots & \iddots & \vdots \\ 0 & & h^{x+}_{l_1l_2} & 1 & \cdots & 1 \end{pmatrix} \tag{4.6}$$

and $\mathbf{H}^y_{l_1l_2}$ is defined similarly. The degradation matrix corresponding to the (l_1,l_2)-th sensor under the Neunmann boundary condition is given by $\mathbf{H}_{l_1l_2} = \mathbf{H}^x_{l_1l_2} \otimes \mathbf{H}^y_{l_1l_2}$. The degradation matrix for the whole sensor array is made up of degradation matrices from each sensor:

$$\mathbf{H}_L = \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \mathbf{D}_{l_1l_2}\mathbf{H}_{l_1l_2}, \quad i,j \in \{r,g,b\}. \tag{4.7}$$

Here $\mathbf{D}_{l_1 l_2}$ are diagonal matrices with diagonal elements equal to 1 if the corresponding component of the observed low resolution image comes from the (l_1, l_2)-th sensor and zero otherwise, see [3] for more details. From (4.3), we see that we have the same matrix $\mathbf{H}_L$ within the channels and across the channels, the overall degradation matrix is given by

$$\mathbf{A}_L = \begin{pmatrix} w_{rr} & w_{rg} & w_{rb} \\ w_{gr} & w_{gg} & w_{gb} \\ w_{br} & w_{bg} & w_{bb} \end{pmatrix} \otimes \mathbf{H}_L \equiv \mathbf{W} \otimes \mathbf{H}_L. \tag{4.8}$$

4.2.2 Regularization

We use the following weighted discrete Laplacian matrix $\mathbf{R}$ proposed by Galatsanos et al. in [4] with $\|\tilde{\mathbf{f}^{(i)}}\|_2$ replaced by $\|\mathbf{g}^{(i)}\|_2$ as the regularization matrix.

$$\mathbf{R} = \begin{pmatrix} 2 & -\frac{\|g^{(r)}\|_2}{\|g^{(g)}\|_2} & -\frac{\|g^{(r)}\|_2}{\|g^{(b)}\|_2} \\ -\frac{\|g^{(g)}\|_2}{\|g^{(r)}\|_2} & 2 & -\frac{\|g^{(g)}\|_2}{\|g^{(b)}\|_2} \\ -\frac{\|g^{(b)}\|_2}{\|g^{(r)}\|_2} & -\frac{\|g^{(b)}\|_2}{\|g^{(g)}\|_2} & 2 \end{pmatrix} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L} = \mathbf{S} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L}, \tag{4.9}$$

where $\mathbf{L}$ is the 2-dimensional discrete Laplacian matrix with the Neumann boundary condition.

Using Tikhonov regularization in (4.2), our discretization problem becomes:

$$(\mathbf{A}_L^t \Upsilon \mathbf{A}_L + \mathbf{R}^t \mathbf{R})\mathbf{f} = \mathbf{A}_L^t \Upsilon \mathbf{g}, \tag{4.10}$$

where

$$\Upsilon = \begin{pmatrix} \alpha_r \mathbf{I} & 0 & 0 \\ 0 & \alpha_g \mathbf{I} & 0 \\ 0 & 0 & \alpha_b \mathbf{I} \end{pmatrix} = \begin{pmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & a & \alpha_b \end{pmatrix} \otimes \mathbf{I} \equiv \Omega \otimes \mathbf{I}, \tag{4.11}$$

and α_r, α_g and α_b are the regularization parameters which are assumed to be positive scalars. According to (4.9), (4.11), and when the Neunmann boundary condition is used for both $\mathbf{H}_L$ and $\mathbf{L}$, (4.10) can be simplified to

$$[\mathbf{W}^t\Omega\mathbf{W}\otimes\Lambda^2+(\mathbf{S}\otimes\mathbf{I}+\mathbf{I}\otimes\Sigma)^t(\mathbf{S}\otimes\mathbf{I}+\mathbf{I}\otimes\Sigma)]\tilde{\mathbf{f}}=(\mathbf{W}^t\Omega\otimes\Lambda)\tilde{\mathbf{g}}, \qquad (4.12)$$

where Λ and Σ are diagonal matrices with diagonal entries given by the eigenvalues of $\mathbf{H}_L$ and $\mathbf{L}$ respectively, $\tilde{\mathbf{f}}=(\mathbf{I}\otimes\mathbf{C}_{M_1}\otimes\mathbf{C}_{M_2})\mathbf{f}$ and $\tilde{\mathbf{g}}=(\mathbf{I}\otimes\mathbf{C}_{M_1}\otimes\mathbf{C}_{M_2})\mathbf{g}$. The system in (4.12) is a block-diagonalized system of M_1M_2 decoupled systems. The vector $\tilde{\mathbf{f}}$ can be computed by solving a set of M_1M_2 decoupled 3-by-3 matrix equations.

4.3 Acquisition of Low-Resolution Movie Frames

In this section, we discuss the case $L=2$, $T_1=T_2=1$ here. We would like to get four low-resolution movie frames with subpixel displacements. First we fix our target object and take movie of the target object by circular motion of the camera, see Figure 1. Next we have to get the four low-resolution movie frames from the recorded movie. The criteria for choosing four movie frames is that they have to be different from each other in terms of 2-norm and the target object should be located in four different positions in the captured movie frames, see Figure 2.

After that, we have to crop the target object out of the movie frames and align them so that they are shifted from each other by approximately one pixel, see Figure 3. Note that the cropped images should all have the same size. These four cropped images serve as the input images for our algorithm.

4.4 Experimental Results

In this section, we illustrate our method by recording a target object in a movie. We tried the degradation matrix in our examples with $w_{rr}=w_{gg}=w_{bb}=0.8$

and $w_{ij} = 0.1$ for $i \neq j$ in (4.8). In the tests, we used the same regularization parameter for each channel, i.e., $\alpha_r = \alpha_g = \alpha_b = \alpha = 70$. Three different target objects are recorded in three examples. The corresponding results are shown in Figures 4–9.

4.5 Concluding Remarks

Since we are using $L = 2$, theoretically, we should align the target object by half a pixel. In our algorithm, we align the target object so that they are shifted from each other by approximately one pixel. In next chapter, we will try to estimate the exact sub-pixel movement of the target object by blind-deconvolution and see whether better images can be reconstructed.

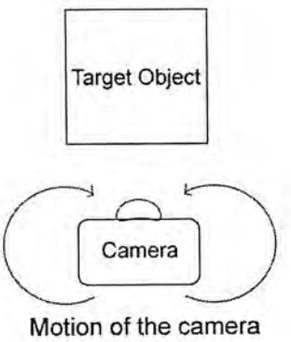

Figure 1: Movement of the camera.

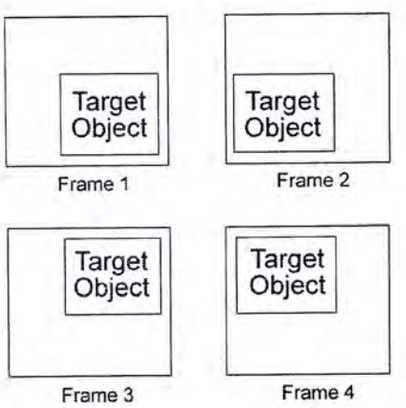

Figure 2: Four different positions of the target object in corresponding frames.

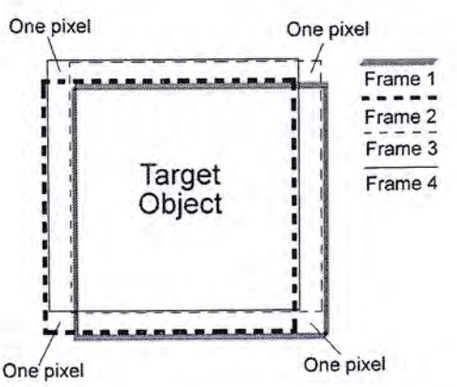

Figure 3: Alignmnet of the target object in corresponding frames.

Figure 4: A low-resolution image (left) and the observed high-resolution image (right).

Figure 5: The reconstructed image using Neunmann boundary condition.

Figure 6:A low-resolution image (left) and the observed high-resolution image (right).

Figure 7: The reconstructed image using Neunmann boundary condition.

Figure 8:A low-resolution image (left) and the observed high-resolution image (right).

Figure 9: The reconstructed image using Neunmann boundary condition.

Chapter 5

Constrained Total Least-Squares Computations for High-Resolution Image Reconstruction

5.1 Introduction

In the previous chapter, we have introduced how to reconstruct a high-resolution images from movie frames and we assumed the sensors are shifted from each other by $\frac{1}{2} \times \frac{1}{2}$ for $L = 2$ and there are no perturbation errors around the ideal subpixel locations. But this assumption is unrealistic. In this section, we will take into account how to reconstruct a high-resolution images when there is perturbation errors around the ideal subpixel locations. We will use the method proposed in

[5] and apply it to our problem. For simplicity, we will focus on monochrome images.

5.2 The Mathematical Model

We are still using a sensor array with $L_1 \times L_2$ sensors, each sensor has $N_1 \times N_2$ sensing elements, and the size of each sensing element is $T_1 \times T_2$. We would like to reconstruct a high-resolution image of resolution $M_1 \times M_2$ where $M_1 = L_1 \times N_1$ and $M_2 = L_2 \times N_2$. To maintain the aspect ratio of the reconstructed image, we only consider the case where $L_1 = L_2 = L$. For simplicity, we assume L is an even number, $T_1 = T_2 = 1$. If we imposed perturbation errors into the previous model, then the horizontal and vertical displacements $d^x_{l_1 l_2}$ and $d^y_{l_1 l_2}$ are given by

$$d^x_{l_1 l_2} = \frac{T_1}{L}(l_1 + \bar{\epsilon}^x_{l_1 l_2}) \quad \text{and} \quad d^y_{l_1 l_2} = \frac{T_2}{L}(l_2 + \bar{\epsilon}^y_{l_1 l_2}),$$

$$l_1, l_2 = 0, 1, \cdots, L-1 \quad \text{with} \quad (l_1, l_2) \neq (0, 0).$$

where $\bar{\epsilon}^x_{l_1 l_2}$ and $\bar{\epsilon}^y_{l_1 l_2}$ denote the actual normalized horizontal and vertical displacement errors respectively.

It is reasonable to assume that

$$|\bar{\epsilon}^x_{l_1 l_2}| < \frac{1}{2} \quad \text{and} \quad |\bar{\epsilon}^y_{l_1 l_2}| < \frac{1}{2},$$

since if the above conditions are violated, then the low-resolution images acquired from different sensors may have excess overlapping information and the resulting reconstructed high-resolution image is not satisfactory.

Let $\mathbf{g}$ and $\mathbf{f}$ be the vectors formed from the continuous high-resolution observed image $g(x_1, x_2)$ and the original high-resolution continuous scene $f(x_1, x_2)$ respectively using column ordering. Once again, we applied the Neumann boundary condition on the images. The resulting degradation matrices in the x–direction and y–direction have a Toeplitz-plus-Hankel structure.

$$\mathbf{H}^x_{l_1l_2}(\bar{\epsilon}^x_{l_2l_2}) = \frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & \frac{1}{2}-\bar{\epsilon}^x_{l_2l_2} & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 1 & \ddots & \ddots & \ddots & \ddots & \frac{1}{2}-\bar{\epsilon}^x_{l_2l_2} \\ \frac{1}{2}+\bar{\epsilon}^x_{l_2l_2} & \ddots & \ddots & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & & \frac{1}{2}+\bar{\epsilon}^x_{l_2l_2} & 1 & \cdots & 1 \end{pmatrix}$$

$$+\frac{1}{L}\begin{pmatrix} 1 & \cdots & 1 & \frac{1}{2}+\bar{\epsilon}^x_{l_1l_2} & & 0 \\ \vdots & \iddots & \iddots & & & \\ 1 & \iddots & & & & \frac{1}{2}-\bar{\epsilon}^x_{l_1l_2} \\ \frac{1}{2}+\bar{\epsilon}^x_{l_1l_2} & & & & \iddots & 1 \\ & & & \iddots & \iddots & \vdots \\ 0 & & \frac{1}{2}-\bar{\epsilon}^x_{l_1l_2} & 1 & \cdots & 1 \end{pmatrix}$$

The matrix $\mathbf{H}^y_{l_1l_2}(\bar{\epsilon}^y_{l_1l_2})$ is defined similarly. The degradation matrix for the (l_1, l_2) sensor is given by

$$\mathbf{H}_{l_1l_2}(\bar{\epsilon}_{l_1l_2}) = \mathbf{H}^x_{l_1l_2}(\bar{\epsilon}^x_{l_2l_2}) \otimes \mathbf{H}^y_{l_1l_2}(\bar{\epsilon}^y_{l_1l_2}),$$

where $\bar{\epsilon}_{l_1l_2} = (\bar{\epsilon}^x_{l_2l_2}, \bar{\epsilon}^y_{l_2l_2})^t$.

The degradation matrix for the whole sensor array is composed of degradation matrices from each sensor and is given by

$$\mathbf{H}_L(\bar{\epsilon}) = \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \mathbf{D}_{l_1l_2}\mathbf{H}_{l_1l_2}(\bar{\epsilon}_{l_1l_2}), \tag{5.1}$$

where $\bar{\epsilon} = [\bar{\epsilon}^x_{00}\bar{\epsilon}^y_{00}\bar{\epsilon}^x_{01}\bar{\epsilon}^y_{01}\cdots\bar{\epsilon}^x_{L-1\,L-2}\bar{\epsilon}^y_{L-1\,L-2}\bar{\epsilon}^x_{L-1\,L-1}\bar{\epsilon}^y_{L-1\,L-1}]^t$ and of size $2L^2 \times 1$. $\mathbf{D}_{l_1l_2}$ are diagonal matrices with diagonal elements equal to 1 if the corresponding element of $\mathbf{g}$ comes from the (l_1, l_2)-th sensor and zero otherwise and are called the sampling matrices. For instance, see [3].

The reconstruction of high-resolution images can be modeled as

$$\mathbf{g} = H(\bar{\epsilon})\mathbf{f} + \mathbf{n}.$$

where $H(\bar{\epsilon})$ is the blurring matrix, $\mathbf{f}$ is the high-resolution image we would like to find, $\mathbf{g}$ is the observed high-resolution image and $\mathbf{n}$ is the noise.

Let $\mathcal{L}(\bar{\epsilon}, \mathbf{f})$ be the operator $\mathcal{L}(\bar{\epsilon}, \mathbf{f}) = H(\bar{\epsilon})\mathbf{f} - \mathbf{g}$. This operator is linear in $\mathbf{f}$ and nonlinear in $\bar{\epsilon}$. Then the reconstruction problem can be formulated as follows:

$$\min_{\bar{\epsilon},\mathbf{f}} \left\{ \|\mathcal{L}(\bar{\epsilon}, \mathbf{f})\|_2^2 \right\}. \tag{5.2}$$

That is, we would like to find the best $\bar{\epsilon}$ and $\mathbf{f}$ such that the residual $H(\bar{\epsilon})\mathbf{f} - \mathbf{g}$ is minimized.

However, image reconstruction problem is generally ill-conditioned, therefore regularization is imposed. Instead of solving (5.2), we regularize the system and solve

$$\min_{\bar{\epsilon},\mathbf{f}} \left\{ \|\mathcal{L}(\bar{\epsilon}, \mathbf{f})\|_2^2 + \mu\|\mathbf{Pf}\|_\mathbf{2}^\mathbf{2} + \|\bar{\epsilon}\|_\mathbf{2}^\mathbf{2} \right\}.$$

Since the above problem is linear with respect to $\mathbf{f}$, so we first take an initial guess of $\bar{\epsilon}$, then we solve the following linear least-squares problem

$$\min_{\mathbf{f}} \left\{ \|\mathcal{L}(\bar{\epsilon}, \mathbf{f})\|_2^2 + \mu\|\mathbf{Pf}\|_\mathbf{2}^\mathbf{2} + \|\bar{\epsilon}\|_\mathbf{2}^\mathbf{2} \right\}.$$

Then we can solve the following nonlinear least-squares problem

$$\min_{\bar{\epsilon}} \left\{ \|\mathcal{L}(\bar{\epsilon}, \mathbf{f})\|_2^2 + \mu\|\mathbf{Pf}\|_\mathbf{2}^\mathbf{2} + \|\bar{\epsilon}\|_\mathbf{2}^\mathbf{2} \right\}.$$

The above subroutine can be solved by the iterative methods (Newton's method, Gauss Newton method, etc) introduced in Chapter 3. But it usually involves more than one iteration to get the best $\bar{\epsilon}$. Here we apply the idea of [5] and just iteratively improve ϵ by linearization and then iterate the equation once.

In the following, we are going to explain the idea in more detail. We assumed $\bar{\epsilon}$ and the observed high-resolution image $\bar{\mathbf{g}}$ are subject to errors and they can be rewritten as

$$\bar{\epsilon} = \epsilon + \delta\epsilon \quad \text{and} \quad \bar{\mathbf{g}} = \mathbf{g} + \delta\mathbf{g}, \tag{5.3}$$

where

$$\epsilon = [\epsilon^x_{00}\epsilon^y_{00}\epsilon^x_{01}\epsilon^y_{01}\ldots\epsilon^x_{L-1\,L-2}\epsilon^y_{L-1\,L-2}\epsilon^x_{L-1\,L-1}\epsilon^y_{L-1\,L-1}]^t,$$

$$\delta\epsilon = \Delta = [\delta\epsilon^x_{00}\delta\epsilon^y_{00}\delta\epsilon^x_{01}\delta\epsilon^y_{01}\ldots\delta\epsilon^x_{L-1\,L-2}\delta\epsilon^y_{L-1\,L-2}\delta\epsilon^x_{L-1\,L-1}\delta\epsilon^y_{L-1\,L-1}]^t,$$

and

$$\delta\mathbf{g} = [\delta g_1 \delta g_2 \ldots \delta g_{M_1 M_2}]^t.$$

By using (5.1) and (5.3)

$$\mathbf{H}_L(\bar{\epsilon})\mathbf{f} = \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\mathbf{D}_{l_1 l_2}\mathbf{H}_{l_1 l_2}(\bar{\epsilon}_{l_1 l_2})\mathbf{f} = \bar{\mathbf{g}} = \mathbf{g} + \delta\mathbf{g}$$

and this equation can be reformulated as

$$\begin{aligned}
&\left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\mathbf{D}_{l_1 l_2}(\mathbf{H}^x_{l_1 l_2}(\epsilon_{l_1 l_2}) \otimes \mathbf{H}^y_{l_1 l_2}(\epsilon_{l_1 l_2}))\right]\mathbf{f} - \mathbf{g} \\
&+ \left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^x_{l_1 l_2}(\mathbf{D}_{l_1 l_2}\mathbf{E} \otimes \mathbf{H}^y_{l_1 l_2}(\epsilon_{l_1 l_2}))\right]\mathbf{f} \\
&+ \left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^y_{l_1 l_2}(\mathbf{D}_{l_1 l_2}\mathbf{H}^x_{l_1 l_2}(\epsilon_{l_1 l_2}) \otimes \mathbf{E})\right]\mathbf{f} \\
&+ \left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^x_{l_1 l_2}\delta\epsilon^y_{l_1 l_2}(\mathbf{D}_{l_1 l_2}\mathbf{E} \otimes \mathbf{E})\right]\mathbf{f} - \delta\mathbf{g} = 0
\end{aligned}$$

or

$$\left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\mathbf{D}_{l_1l_2}(\mathbf{H}^x_{l_1l_2}(\epsilon_{l_1l_2})\otimes\mathbf{H}^y_{l_1l_2}(\epsilon_{l_1l_2}))\right]\mathbf{f}-\mathbf{g}$$
$$+\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^x_{l_1l_2}\mathbf{f}^y_{l_1l_2}+\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^y_{l_1l_2}\mathbf{f}^x_{l_1l_2}$$
$$+\left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^x_{l_1l_2}\delta\epsilon^y_{l_1l_2}(\mathbf{D}_{l_1l_2}\mathbf{E}\otimes\mathbf{E})\right]\mathbf{f}-\delta\mathbf{g}=0, \tag{5.4}$$

where

$$\mathbf{E}=\frac{1}{L}\begin{pmatrix}0&\dots&0&-1&&0\\ \vdots&\ddots&\ddots&\ddots&\ddots&\\ 0&\ddots&\ddots&\ddots&\ddots&-1\\ 1&\ddots&\ddots&\ddots&\ddots&0\\ &\ddots&\ddots&\ddots&\ddots&\vdots\\ 0&&1&0&\dots&0\end{pmatrix}+\frac{1}{L}\begin{pmatrix}0&\dots&0&1&&0\\ \vdots&\iddots&\iddots&\iddots&\iddots&\\ 0&\iddots&\iddots&\iddots&\iddots&-1\\ 1&\iddots&\iddots&\iddots&\iddots&0\\ &\iddots&\iddots&\iddots&\iddots&\vdots\\ 0&&-1&0&\dots&0\end{pmatrix}$$

and

$$\mathbf{f}^x_{l_1l_2}=\mathbf{D}_{l_1l_2}(\mathbf{H}^x_{l_1l_2}(\epsilon_{l_1l_2})\otimes\mathbf{E})\mathbf{f},\quad \mathbf{f}^y_{l_1l_2}=\mathbf{D}_{l_1l_2}(\mathbf{E}\otimes\mathbf{H}^y_{l_1l_2}(\epsilon_{l_1l_2}))\mathbf{f}.$$

Because $\delta\epsilon^x_{l_1l_2},\delta\epsilon^y_{l_1l_2}(l_1,l_2=0,1,\cdots,L-1)$ should be very small, the term associated with $\delta\epsilon^x_{l_1l_2}\delta\epsilon^y_{l_1l_2}$ can be neglected, therefore,

$$\left[\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta^x_{l_1l_2}\delta^y_{l_1l_2}(\mathbf{D}_{l_1l_2}\mathbf{E}\otimes\mathbf{E})\right]\mathbf{f} \tag{5.5}$$

can be ignored. The constrained total least squares formulation can be expressed as

$$\min_{\mathbf{f}}\left\|\begin{bmatrix}\Delta\\ \delta\mathbf{g}\end{bmatrix}\right\|_2^2$$

subject to

$$\mathbf{H}_L(\epsilon)\mathbf{f}-\mathbf{g}+\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^x_{l_1l_2}\cdot\mathbf{f}^y_{l_1l_2}\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1}\delta\epsilon^y_{l_1l_2}\cdot\mathbf{f}^x_{l_1l_2}-\delta\mathbf{g}=0. \tag{5.6}$$

The above image reconstruction problem is generally ill-conditioned and is extremely sensitive to noise. In order to achieve stability, regularization should be used. We will use Tikhonov regularization in our problem and the regularized solution $\mathbf{f}$ is computed as the solution to

$$\min_{\mathbf{f}} \left\| \begin{bmatrix} \Delta \\ \delta \mathbf{g} \end{bmatrix} \right\|_2^2 + \mu \|\mathbf{P}\mathbf{f}\|_2^2 \tag{5.7}$$

subject to Eq. (5.5). The regularization parameter μ controls the degree of bias of the solution and $\|\mathbf{P}\mathbf{f}\|_2$ is chosen to be $\|\mathbf{f}\|_2$ or $\|\mathbf{R}\mathbf{f}\|_2$, where $\mathbf{R}$ is a first-order difference operator matrix. We called the above problem the Regularized Constrained Total Least Squares (RCTLS) problem.

5.3 Numerical Algorithm

In this section, a numerical algorithm to minimize Eq. (5.6) is introduced. By Eqs. (5.5) and (5.6), we have

$$\min_{\mathbf{f},\Delta} J(\mathbf{f},\Delta) \equiv \min_{\mathbf{f},\Delta} \left\{ \left\| \mathbf{H}_L(\epsilon)\mathbf{f} - \mathbf{g} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^x_{l_1 l_2} \cdot \mathbf{f}^y_{l_1 l_2} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^y_{l_1 l_2} \cdot \mathbf{f}^x_{l_1 l_2} \right\|_2^2 + \left\| \Delta \right\|_2^2 + \mu \left\| \mathbf{P}\mathbf{f} \right\|_2^2 \right\} \tag{5.8}$$

For a given $\mathbf{f}$, the function $J(\mathbf{f},\cdot)$ is convex with respect to Δ, and for a given Δ, the function $J(\cdot,\Delta)$ is also convex with respect to $\mathbf{f}$. Therefore, with an initial guess Δ_0, one can minimize Eq. (5.8) by solving

$$J(\mathbf{f}_1,\Delta_0) = \min_{\mathbf{f}} J(\cdot,\Delta_0)$$

and then

$$J(\mathbf{f}_1,\Delta_1) = \min_{\Delta} J(\mathbf{f}_1,\cdot)$$

in a cyclic fashion. Precisely, the algorithm is stated as follows:

Assume that Δ_{n-1} is available:

Determine $\mathbf{f}_n$ by solving the following least-squares problem

$$\min_{\mathbf{f_n}} \left\{ \left\| \mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^x_{n-1\,l_1l_2}\mathbf{f}^y_{n\,l_1l_2} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^y_{n-1\,l_1l_2}\mathbf{f}^x_{n\,l_1l_2} \right\|_2^2 + \mu\|\mathbf{P}\mathbf{f}_n\|_2^2 \right\}, \tag{5.9}$$

where

$$\mathbf{f}^x_{n\,l_1l_2} = \mathbf{D}_{l_1l_2}(\mathbf{H}^x_{l_1l_2}(\epsilon_{l_1l_2}) \otimes \mathbf{E})\mathbf{f}_n \quad \text{and} \quad \mathbf{f}^y_{n\,l_1l_2} = \mathbf{D}_{l_1l_2}(\mathbf{E} \otimes \mathbf{H}^y_{l_1l_2}(\epsilon_{l_1l_2}))\mathbf{f}_n \tag{5.10}$$

The least-squares solution $\mathbf{f}_n$ can be found by solving

$$[\mathbf{H}_L(\epsilon + \Delta_{n-1})^t\mathbf{H}_L(\epsilon + \Delta_{n-1}) + \mu\mathbf{P}^t\mathbf{P}]\mathbf{f}_n = \mathbf{H}_L(\epsilon + \Delta_{n-1})^t\mathbf{g}. \tag{5.11}$$

Determine Δ_n by solving the following least-squares problem

$$\min_{\Delta_n} \left\{ \|\Delta_n\|_2^2 + \left\| \mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^x_{l_1l_2}\mathbf{f}^y_{n\,l_1l_2} + \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} \delta\epsilon^y_{l_1l_2}\mathbf{f}^x_{n\,l_1l_2} \right\|_2^2 \right\}. \tag{5.12}$$

By using Eq. (5.10), the above equation can be rewritten as

$$\min_{\Delta_n}\{\|\Delta_n\|_2^2 + \|\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g} + \mathbf{Q}(\mathbf{f}_n)\Delta_n\|_2^2\} \tag{5.13}$$

where

$$\mathbf{Q}(\mathbf{f}) = [\mathbf{f}^y_{00}|\mathbf{f}^x_{00}|\mathbf{f}^y_{01}|\mathbf{f}^x_{01}|\cdots|\mathbf{f}^y_{L-1L-2}|\mathbf{f}^x_{L-1L-2}|\mathbf{f}^y_{L-1L-1}|\mathbf{f}^x_{L-1L-1}].$$

The cost function in Eq. (5.13) becomes

$$\begin{aligned} J(\mathbf{f}_n, \Delta_n) =& \Delta_n^t\Delta_n + \mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g} + \mathbf{Q}(\mathbf{f}_n\Delta_n)^t(\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g} + \mathbf{Q}(\mathbf{f}_n\Delta_n) \\ =& \Delta_n^t\Delta_n + (\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g})^t(\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g}) + (\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g})^t\mathbf{Q}(\mathbf{f}_n)\Delta_n \\ &+ (\mathbf{Q}(\mathbf{f}_n)\Delta_n)^t \times (\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g}) + (\mathbf{Q}(\mathbf{f}_n)\Delta_n)^t(\mathbf{Q}(\mathbf{f}_n)\Delta_n). \end{aligned}$$

The gradient of the above cost function with respect to Δ_n is

$$2\Delta_n + 2\mathbf{Q}(\mathbf{f}_n)^t(\mathbf{H}_L(\epsilon)\mathbf{f}_n - \mathbf{g}) + 2\mathbf{Q}(\mathbf{f}_n)^t\mathbf{Q}(\mathbf{f}_n)\Delta_n.$$

By setting the above function equals to zero, we get

$$\Delta_n = [\mathbf{I}_{2L^2} + \mathbf{Q}(\mathbf{f}_n)^t\mathbf{Q}(\mathbf{f}_n)]^{-1}\mathbf{Q}(\mathbf{f}_n)^t(\mathbf{g} - \mathbf{H}_L(\epsilon)\mathbf{f}_n). \tag{5.14}$$

Eq. (5.14) can be solved in just one iteration, because the size of $\mathbf{I}_{2L^2}+\mathbf{Q}(\mathbf{f}_n)^t\mathbf{Q}(\mathbf{f}_n)$ is small ($2L^2 \times 2L^2$), so it is very easy to find its inverse.

The above method is obtained by linearizing $\bar{\epsilon} = \epsilon + \delta\epsilon$ and then solving the corresponding linear least-squares problem.

5.4 Numerical Results

First, we will use the above algorithm to reconstruct a high-resolution image from several degraded and shifted images obtained from camera with calibration errors, see Figures 5.1, 5.2 and 5.3. Then we will use the low-resolution images obtained from a movie to reconstruct a high-resolution image. See Figures 5.4, 5.5 and 5.6. Both of the results are using $\mu = 0.1$ and the initial guess of $\bar{\epsilon} = \vec{0}$.

5.5 Concluding Remarks

From the numerical results, we can see that the algorithm works well in the reconstruction of high-resolution image by using the low-resolution source images from camera. But it fails to reconstruct a good quality high-resolution image from the source images obtained in a movie. Reasons that lead to failure maybe inaccurate modelling of the situation, the calibration errors that the algorithm calculated are incorrect when using the source images from movie frames. One

Figure 5.1: One of the low-resolution image (left) and the observed high-resolution image (right).

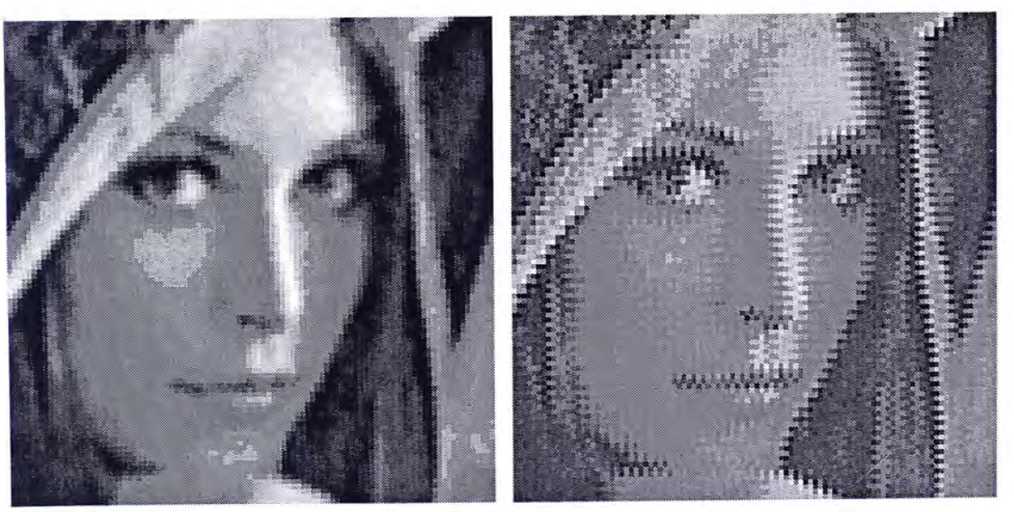

Figure 5.2: The high-resolution image obtained from the first iteration (left) and from the second iteration (right).

Figure 5.3: The high-resolution image obtained from the third iteration (left) and from the forth iteration (right).

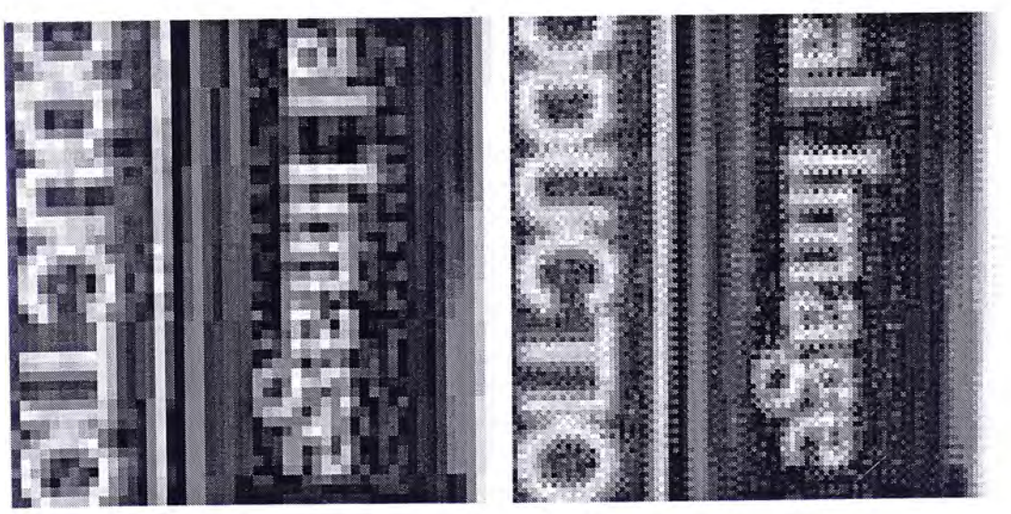

Figure 5.4: One of the low-resolution image (left) and the observed high-resolution image (right).

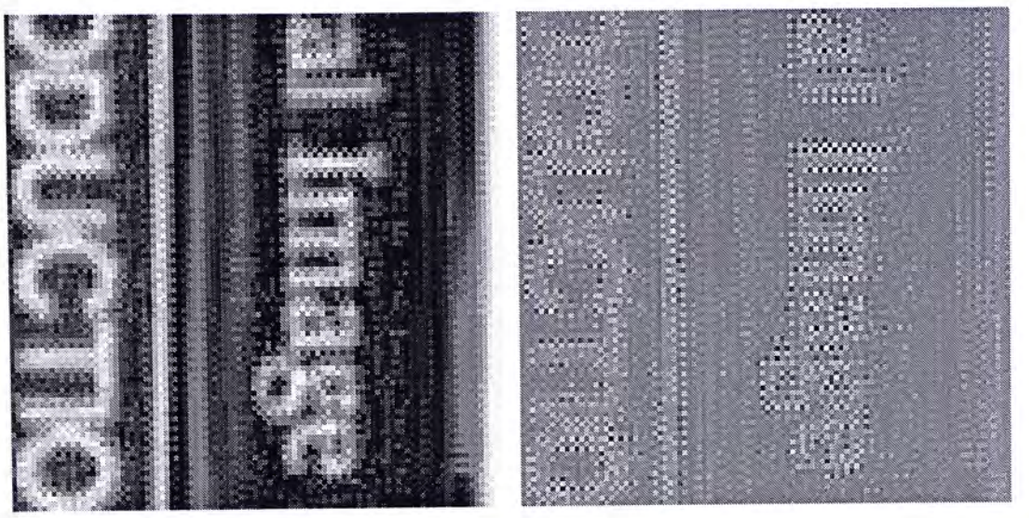

Figure 5.5: The high-resolution image obtained from the first iteration (left) and from the second iteration (right).

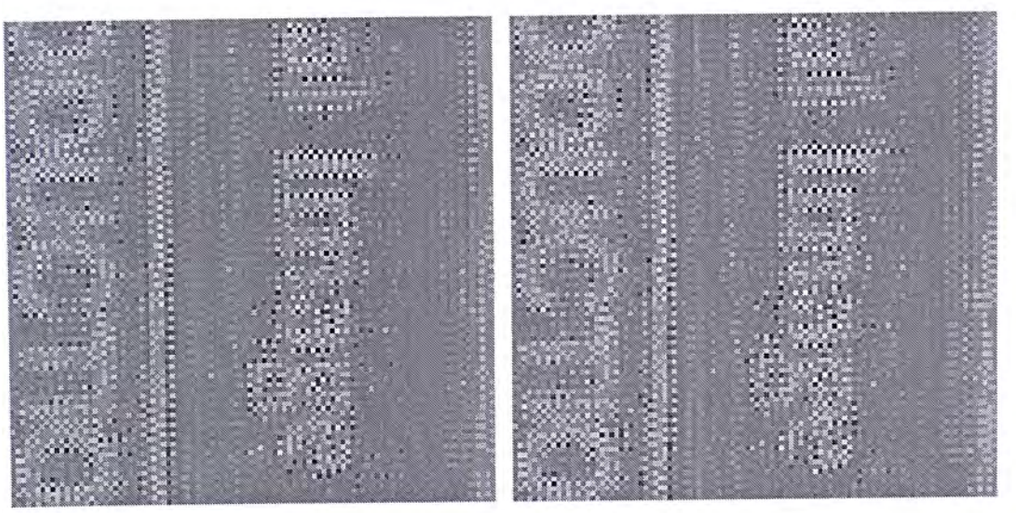

Figure 5.6: The high-resolution image obtained from the third iteration (left) and from the forth iteration (right).

may try to use a better model or explore other methods (for example, by using the motion vectors, see [9]) to reconstruct the high-resolution image.

Bibliography

[1] John C. Russ, *The image processing handbook–2nd ed.*, CRC Press, 32 – 46.

[2] M. Ng, W. Kwan, and R. Chan, *A Fast Algorithm for High-Resolution Color Image Reconstruction with Multisensors*, Lecture Notes in Computer Science, 1988 (2001), 615–627.

[3] N. Bose and K. Boo, *High-Resolution Image Reconstruction with Multisensors*, International Journal of Imaging Systems and Technology, 9 (1988), 294–304.

[4] N. Galatsanos, A. Katsaggelos, R. Chin, and A. Hillery, *Least Squares Restoration of Multichannel Images*, IEEE Trans. Signal Processing, 39 (1991), 2237–2252.

[5] M. Ng, J. Koo and N. Bose, *Constrained Total Least-Squares Computations for High-Resolution Image Reconstruction with Multisensors*, International Journal of Imaging Systems and Technology, 12 (2002), 35–42.

[6] A. Ruhe, *Accelerated Gauss-Newton algorithms for nonlinear least squares problems*, BIT, 19 (1979), 356–367.

[7] J. Dennis and R. Schnable, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[8] Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[9] C. Segall, R. Molina, A. Katsaggelos, *High-Resolution Images from Low-Resolution Compressed Video*, IEEE Signal Processing Magazine, 20 (2003), 37–48.

[10] M. Kang, S. Chaudhuri, *Super-resolution image reconstruction* , IEEE Signal Processing Magazine, 20 (2003), 19–20.

[11] S.Park, M. Park, M. Kang, *Super-resolution image reconstruction: a technical overview*, IEEE Signal Processing Magazine, 20 (2003), 21–36.