5-2020

Automatic Numerical Methods for Enhancement of Blurred Text-Images via Optimization and Nonlinear Diffusion

Aaditya Kharel

The University of Southern Mississippi

Automatic Numerical Methods for Enhancement of Blurred Text-Images via Optimization and Nonlinear Diffusion

by

Aaditya Kharel

A Thesis
Submitted to the Honors College of
The University of Southern Mississippi
in Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science
in the School of Mathematics and Natural Sciences

May 2020

Approved by

James V. Lambers, Ph.D., Thesis Adviser
Professor of Mathematics

Bernd Schroeder, Ph.D., Director
School of Mathematics and Natural Sciences

Ellen Weinauer, Ph.D., Dean
Honors College

Abstract

In this paper, we propose an automatic numerical method for solving a nonlinear partial-differential-equation (PDE) based image-processing model. The Perona-Malik diffusion equation (PME) accounts for both forward and backward diffusion regimes so as to perform simultaneous denoising and deblurring depending on the value of the gradient. One of the limitations of this equation is that a large value of the gradient for backward diffusion can lead to singularity formation or staircasing. Guidotti-Kim-Lambers (GKL) came up with a bound for backward diffusion to prevent staircasing, where the backward diffusion is only limited to a specific range beyond which backward diffusion is stopped and forward diffusion begins. Our model combines the PME model and GKL model for automatic sharpening of blurred text-images using Nelder-Mead optimization, a derivative free optimization method that uses $n+1$ test points arranged as a simplex for n-dimensional optimization. We solve our model by discretizing the PDE in space using finite difference approximation scheme. Then, we enhance the image in each iteration using Backward Euler time-stepping and Minimum Residual Method (MINRES) in MATLAB. Likewise, we propose a gradient-based sharpness metric for our text-images, which also serves as an objective function for our Nelder-Mead optimizer. Our result shows that our proposed model is accurate in enhancing text images and predicting the unknown value of the blurring kernel for automatic sharpening. Numerical results show that the proposed objective sharpness measure coincide with the subjective sharpness of the enhanced image.

Key Words: Anisotropic diffusion, Denoising, Deblurring, Forward-Backward-Forward Diffusion, Text Images, Nelder-Mead Optimization

Acknowledgments

I would like to extend my sincere thanks to my thesis advisor Dr. James V. Lambers, without whom I would not have received the inspiration and motivation to undertake this research project. I am very grateful for his understanding and considerate nature and the support he provided during my downtime. I feel very proud to be his advisee!

I am thankful to Dr. Zhifu Xie and Dr. Huiqing Zhu for their marvelous support and mentoring during the MAA-NREUP Summer Research 2018, which helped me grow as an undergraduate researcher. I would like to express my sincere gratitude to Wright W. and Annie Rea Cross Endowed Scholarship in Mathematics for funding my honors thesis for two academic years. Likewise, I would like to thank my academic advisor Dr. Bernd Schroeder for always willing to work around my schedule to satisfy the Mathematics degree requirements.

Likewise, I am very grateful to my family for always believing in me and pushing me to achieve more in life. This thesis would not have been possible without my mom, who has guided and directed me at every hurdle in my life no matter where I am. I am tremendously thankful for her love and support.

Last but not the least, I feel very fortunate to have been surrounded by kind, loving and supportive friends during my time at the University of Southern Mississippi. I am grateful to my roommate Shiron Manandhar who has not only been through my joyful moments from the beginning of my undergraduate journey but also during my struggles and tough times. I will forever owe to his kindness and generosity.

Table of Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

Image processing is of wide interest for various applications and image enhancement techniques such as denoising, deblurring, encoding and image compression are widely used for this purpose. Denoising and deblurring are two important techniques for image enhancement as they deal with the noise and focus of images which arise regularly in everyday photography and imagery. Noise in an image is visually characterized as the image appearing “grainy”, where there is a small oscillation in color intensity or pixel value. Denoising is the process of removing this grainy characteristic from the image. Similarly, deblurring is the process of sharpening edges, where an edge is a region of an image characterized by a sudden change in pixel value [3]. In image processing, edges capture most image information. The Perona-Malik equation (PME) performs forward diffusion in time to achieve smoothing (denoising) and backward diffusion in time to achieve sharpening [2]. The PME was the first nonlinear diffusion technique that incorporated a forward-backward diffusion for image enhancement allowing simultaneous denoising and deblurring. The PME is shown below:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1 + k^2 |\nabla u|^2} \right) \nabla u \right] \tag{1.1}$$

where k is the edge detection parameter and u is the image data.

However, one of the limitations of the PME is that a large value of the gradient for backward diffusion can lead to an abrupt change in pixel value causing a “cartoonish” effect known as indiscriminate singularity formation, or “staircasing”. Also, PME is ill-posed or theoretically unsound, meaning that the returned solution may not necessarily be correct. This undesirable phenomenon can be avoided by setting a bound for the gradient to facilitate controlled diffusion [2]. This technique is called forward-backward-forward diffusion because it performs backward diffusion only for a specified range of the gradient and switches to forward diffusion if the gradient is too steep to prevent staircasing for larger gradients. The Guidotti-Kim-Lambers (GKL) model that incorporates state-of-the-art forward-backward-forward diffusion regime is shown below:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1 + |\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] \tag{1.2}$$

where δ and p are the regularization parameters and u represents the image data.

In our proposed model, we integrate the edge detection parameter (k) from Perona-Malik equation and the regularization parameter δ and p from the Guidotti-Kim-Lambers model. The integration of parameters from both equation is believed to provide more control over the diffusion process. The proposed model is shown below:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1+k^2|\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] \tag{1.3}$$

where k is the edge detection parameter from PME and δ is the regularization parameters from GKL model and u represents the image data. The parameter regularization parameter p used originally in GKL model is held constant in our model, where $p = 1.05$.

Among the various experiments performed using the models in (1.1) and (1.2), including the satellite-based image enhancement by Guidotti-Kim-Lambers, the prior methods have focused on general images without text. In our model, we want to particularly look at images containing text as the enhancement process can significantly differ for text-based images. Text-based images also contain distinct and prominent edges that are not typical to scene-based images. Likewise, text-based image processing has wide applications in forensics and law enforcement including license plate detection and hand-writing recognition and hence the necessity to enhance text images is inevitable.

Our primary goal of the project is to detect the value of the blurring kernel for automatic sharpening of blurred text-images. In order to solve our system in (3.12) we discretize the PDE in space to obtain a system of ordinary differential equations. Then, we use the Backward Euler scheme to time-discretize the system of ODEs. The resulting time-discretized system obtained from Backward Euler is solved using the Minimum Residual Method (MINRES). The system solved by MINRES enhances the image in each iteration. We make use of Nelder-Mead optimization, a derivative-free optimization technique, to optimize two regularization parameters and one blurring kernel parameter. The finished version of this project is a MATLAB program that takes an image as input and enhances the input image by sharpening the edges of the image and automatically detecting the unknown initial value of the blurring kernel.

Chapter 2

Literature Review

The accurate and consistent quantification of image sharpness is a very useful measure for various applications in image processing including automatic denoising and deblurring. Devising an accurate and consistent image sharpness measure that works for every image is one of the fundamental challenges in image processing. Various image sharpness algorithms have been used to compute the sharpness of an image, but each of these algorithms seems to be best suited for certain types of images or applications. Image sharpness can broadly be assessed in two ways. One way to assess the image quality is subjectively through human opinion and judgement. However, subjective assessment can lead to biased judgement as it is limited to human vision. Another class of technique is the assessment of image quality objectively through an algorithm. To become consistent in assessment for image quality sharpness, it is necessary that the objective measure also agree with general human judgement. The three types of techniques for objective image sharpness assessment are full-reference, reduced-reference and no-reference image quality assessment [6, 7]. The full-reference image quality assessment compares a known reference image with a blurred image. A survey on full-reference image quality assessment is best described in [9]. Likewise, reduced-reference image quality assessment is a better practical approach to accessing the image quality via partial image information (such as the edges) regarding the reference image. Finally, no-reference image quality assessment provides a way to access image quality without a need for a reference image or any of its extracted features. A no-reference image quality assessment is highly sought after in image processing because in many practical applications, very little to no information about the original image is known.

One of the image sharpness measures as outlined by De and Masilamani in [6] for computing sharpness of blurred images deals with counting the number of frequency components that are above the threshold frequency value, which can be classified as non-reference image quality assessment technique for image quality assessment as discussed above. Some other image assessment measures include the Kurtosis metric [10, 11], derivative-based metric [12], histogram threshold metric [13], frequency threshold metric [13] and variance metric [14]. A full review of these no-reference image quality assessment metric is also best discussed in [8]. Like deblurring, image denoising is fundamental to enhancing images. Methods such as total variation (TV) aim at minimization of the total noise present in an image [15, 3]

Some image sharpness quantifiers make use of edge detection and feature extraction from images. One of the popular edge detection techniques known as Canny Edge Detection [5] was experimented with in our project but did not seem viable given the nature of our project. In fact, the Canny Edge Detection was very good at detecting edges of both blurred and sharp images and hence could not identify subtle differences in edge quantification between a blurry and sharp image.

Another technique that was experimented with for this project for image quality assessment was the Fast Fourier Transform (FFT) in which the signal in the spatial-domain was converted into the frequency domain. However, when comparing the FFT signals of blur and sharp image, no significant changes in the frequency component signals were seen, making it difficult to quantify image sharpness. Accurate image sharpness quantification is an integral part of our project in order to optimize sharpness of an image with a set of constraints. As a result, a gradient-based technique served best for this purpose which has been discussed in detail in Chapter 3.

Chapter 3

Methodology

3.1. Software Model

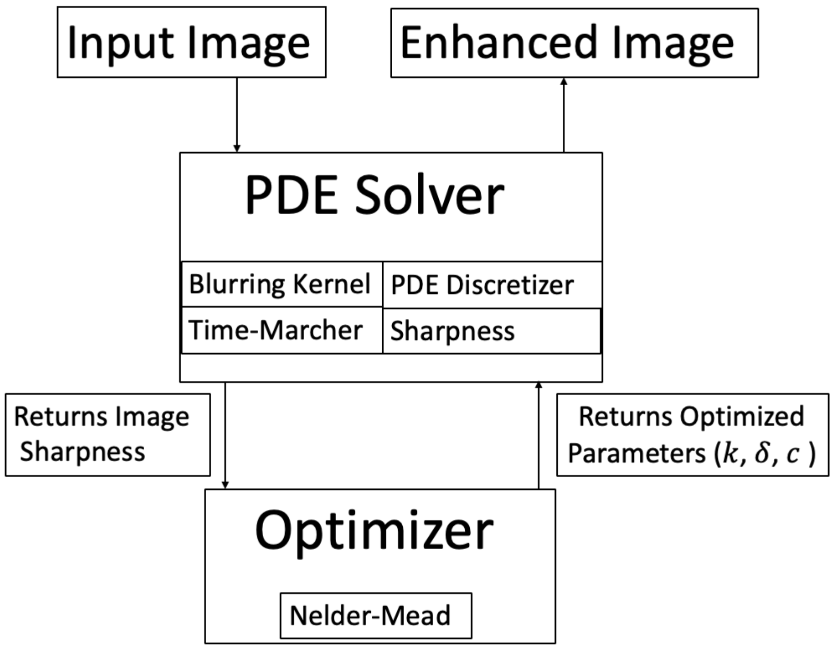

Figure 3.1: Software Design Model

Our software model takes a text image as input with square dimensions, which is fed into the PDE solver. The PDE solver consists of four major components–namely, Blurring Kernel, PDE Discretizer, Time-Marcher and Sharpness. The Blurring Kernel module applies Gaussian Blur to the image with a chosen value of c that controls the amount of blur to be applied to the image. The lower the value of c, the greater is the blurring effect. The PDE Discretizer module is responsible for spatial discretization of the PDE using a finite-difference approximation scheme resulting in a system of ODEs. In the Time-Marcher module, the system of equations arising from Backward Euler is solved iteratively using MINRES. The sharpness module computes the sharpness of the image using a gradient-based method. The image sharpness that is returned from the PDE solver module is then fed into the Nelder-Mead optimization which first optimizes over the blurring kernel parameter c and then performs a two-dimensional search for the optimized values of the k and δ variables for maximized sharpness.

3.2. Reading the Image

Our method reads a square text image in MATLAB, which is stored in a square matrix as double-precision floating-point numbers. Each entry of the square matrix corresponds to the pixel value of the image that ranges over a 0 to 255 color scale, where 0 indicates black color and 255 indicates white color. Since the image is a color image and contains Red-Green-Blue (RGB) channels, we take a slice of the red channel to process our image data. The user can choose to look at any segment of the image by specifying a range of rows and columns within the image dimensions. However, the only constraint we impose is that user must choose an even square dimension matrix which makes it easier to solve our system. It should be noted that requiring a square segment is not an essential feature of the method, but a convenience for implementation and exposition.

Once the 1-channel slicing of the image data is obtained, we reshape our matrix into a vector by converting columns of data into one single column. This vector serves as our initial data u_0 at time $t = 0$.

3.3. Discretization of PDE

As discussed earlier, the Discretizer module is responsible for the spatial discretization of the PDE. The first step towards discretization begins by forming a 1-D differentiation matrix with periodic boundary conditions. The entries of the 1-D differenciation matrix come from the finite-difference approximation scheme shown below:

$$\frac{\partial u}{\partial x} = \frac{u_{i+1} - u_i}{\Delta x} \tag{3.1}$$

where $i = 1, 2, 3, ..., N$ represents the i^{th} grid point. Using our finite-difference approximation scheme for our first-order differential operator above, we create an $N \times N$ forward difference Toeplitz matrix with -1 on the main diagonal and 1 on the super diagonal. Additionally, the $(N, 1)$ entry of the matrix is also set to 1 to account for the periodic boundary condition. The 1-D forward difference matrix is shown below:

$$D^{+} = \frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & & & \ddots & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 & -1 \end{pmatrix}$$

In a similar way, we construct the 1-D backward-differentiation matrix and 1-D centered differentiation matrix using the following relations:

$$D^{-} = -\left(D^{+}\right)^{T} \tag{3.2}$$

$$D^{C} = \frac{1}{2}\left(D^{+} + D^{-}\right) \tag{3.3}$$

where D^{-} is the backward differentiation matrix, D^{+} is the forward differentiation matrix, and D^{C} is the centered differentiation matrix. These 1-D differentiation matrices can be used to construct 2-D differentiation matrices using the following relations:

$$\begin{aligned} {D_x}^{+} &= D^{+} \otimes I \\ {D_x}^{-} &= D^{-} \otimes I \\ {D_y}^{+} &= I \otimes D^{+} \\ {D_y}^{-} &= I \otimes D^{-} \end{aligned} \tag{3.4}$$

where ${D_x}^{+}$ is the forward differentiation matrix with respect to x, ${D_x}^{-}$ is the backward differentiation matrix with respect to x, ${D_y}^{+}$ is the forward differentiation matrix with respect to y, ${D_y}^{-}$ is the backward differentiation matrix with respect to y, and I is the identity matrix. The operation $\otimes$ symbolizes Kronecker product between two matrices.

The 2-D differentiation matrices are useful in the discretization of the PDE model. Let us begin by reiterating our PDE model:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1+k^2|\nabla u|^2} + \delta|\nabla u|^{p-2}\right)\nabla u\right] \tag{3.5}$$

The PDE above can be re-written in the form below:

$$u_t = \nabla \cdot \left(g\left(u\right)\nabla u\right) \tag{3.6}$$

where $g(u)$ is the nonlinear coefficient of the PDE given by:

$$g(u) = \frac{1}{1+k^2(u_x^2+u_y^2)} + \delta(u_x^2+u_y^2)^{\frac{p-2}{2}} \tag{3.7}$$

where u_x and u_y are given by the centered differentiation matrix (D_x^C) with respect to x and centered differentiation matrix (D_y^C) with respect to y respectively as shown below:

$$u_x = D_x^C u \tag{3.8}$$

$$u_y = D_y^C u \tag{3.9}$$

Our goal is to convert the PDE, which is a function of three variables in x, y and t into a function of one variable in t using the finite-difference approximation scheme. This yields a system of ordinary differentiation equations (ODEs) as a result of the discretization at each grid point. The resulting system of ODEs can be expressed in the form below:

$$\vec{u}' = [A(\vec{u})]\,\vec{u} \tag{3.10}$$

where A is the symmetric positive semi-definite matrix given by:

$$A = D_x{}^{+}\left(G(\vec{u})D_x{}^{-}\right) + D_y{}^{+}\left(G(\vec{u})D_y{}^{-}\right) \tag{3.11}$$

and G is the diagonal matrix containing the coefficients $g(u)$ evaluated at each grid point.

3.4. Backward Euler and Minimum Residual (MINRES) Method

After we discretize our PDE in space and obtain the system of ODEs, we use an implicit time-stepping method such as Backward Euler to solve this system. This is because explicit methods are usually not practical for solving the diffusion equation because of the limitations that they impose on the time-step. A general Backward Euler scheme used in time discretization of the ODE is shown below:

$$\vec{u}^{n+1} = \vec{u}^{n} + \Delta t A(\vec{u}^{n})\,\vec{u}^{n+1} \tag{3.12}$$

$$(I - \Delta t A)\vec{u}^{n+1} = \vec{u}^{n} \tag{3.13}$$

where u^n represents the solution at the current time-step and u^{n+1} represents the solution at the next time-step.

In order to aid us in detecting the unknown value of the blurring kernel, we use a fidelity term in our model that includes the effect of our blurring kernel. This model with added fidelity term is shown below:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1+k^2|\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] + \lambda K'(u_o - Ku) \tag{3.14}$$

where K is the Gaussian blurring operator, K' is the adjoint operator, λ is the scaling parameter and u_o is the original blurred image. Since Gaussian blur is symmetric, $K = K'$. This can be proven below:

Theorem 1. Let $g(x)$ be a function of x, $f(y)$ be a function of y, $k(x,y)$ be a blurring function of x and y and $k'(x,y)$ be the adjoint of $k(x,y)$. If $k(x,y) = e^{c\|x-y\|^2}$, then $k(x,y) = k'(x,y)$.

Proof.

$$\begin{aligned} < f, kg > &= \int f(y) \int k(x,y) g(x)\, dx\, dy \\ &= \int \int f(y) k(x,y) g(x)\, dx\, dy \\ &= \int g(x) \int f(y) k(x,y)\, dy\, dx \\ &= \int g(x) \int f(y) e^{c\|x-y\|^2}\, dy\, dx \\ &= \int g(x) \int f(y) e^{c\|y-x\|^2}\, dy\, dx \\ &= \int g(x) \int f(y) k(y,x)\, dy\, dx \\ &= < g, kf > \end{aligned} \tag{3.15}$$

But, $< f, kg > = < g, k'f >$ by definition of k'. So, $k = k'$ □

As a result of the added fidelity term in our model, we derive a variant of the generalized implicit backward Euler scheme that caters to our model:

$$\begin{aligned} \vec{u}_t &= A\vec{u} + \lambda k'(u_o - ku) \\ \frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} &= A\vec{u}^{n+1} + \lambda k'(u_o - k\vec{u}^n) \\ \vec{u}^{n+1} - \Delta t A \vec{u}^{n+1} &= \vec{u}^n + \lambda \Delta t k'(u_o - k\vec{u}^n) \\ (I - \Delta t A)\vec{u}^{n+1} &= \vec{u}^n + \lambda \Delta t k'(u_o - k\vec{u}^n) \end{aligned} \tag{3.16}$$

We then use the Minimum Residual Method (MINRES) in MATLAB to solve our time discretized system in (3.16). MINRES solves a least squares problem in each iteration to find a minimum norm residual solution for the system. The description on how MINRES solves this problem is shown below:

$$\vec{x}^{(k)} = \vec{x}^{(0)} + \vec{y}^{(k)} \tag{3.17}$$

Here, $\vec{y}^{(k)} \in \mathcal{K}(\vec{r}^{(0)}, A, k)$=span{ $\vec{r}^{(0)}$, $A\ \vec{r}^{(0)}$, $A^{k-1}\ \vec{r}^{(0)}$ } is the Krylov Subspace such that $\vec{r}^{(0)}$=$\vec{b}$ $-A\ \vec{x}^{(0)}$ is the initial residual and k is the dimension. In MINRES, $\vec{y}^{(k)}$ is chosen in such a way that:

$$\left\|\vec{r}^{(k)}\right\|_2 = \left\|\vec{b} - A\vec{x}^{(k)}\right\|_2 \tag{3.18}$$

is minimized over all $\vec{x}^{(r)}$ of the form (**??**), thereby solving a least squares problem in each iteration. The choice for an iterative method such as MINRES is made in our solution because methods such as Gaussian elimination with pivoting can destroy the sparse structure of the matrix. Also, an iterative method has the advantage about the knowledge of the solution from the previous time-step. These methods are not aiming for super high accuracy and hence has much faster convergence. MINRES is also a better choice compared to slower converging stationary iterative methods such as Jacobi, Gauss-Seidel and Successive Over-Relaxation (SOR). Likewise, the choice for Backward Euler is made for time discretization because it is unconditionally stable. The proof for its unconditional stability is shown below. A general system of ODEs can be expressed as:

$$\vec{y'} = A\vec{y} \tag{3.19}$$

where $A = Q\Lambda Q^T$ is a symmetric positive semi-definite matrix and $Q^T Q = I = QQ^T$ because Q is an orthogonal matrix. Let $\lambda_1, \lambda_2, ..., \lambda_N$ be the eigenvalues. Then, multiplying both sides of equation (3.19) by Q^T we get

$$\begin{aligned}
Q^T \vec{y'} &= Q^T A\vec{y} \\
(Q^T \vec{y})' &= Q^T AQ(Q^T \vec{y}) \\
\vec{w'} &= \Lambda\vec{w} \quad [\vec{w} = (Q^T \vec{y})] \\
w_i' &= \lambda_i w_i
\end{aligned} \tag{3.20}$$

Then, multiplying the Backward Euler scheme in (3.12) by Q^T, we get

$$\begin{aligned}
Q^T (\vec{y}^{n+1} &= \vec{y}^n + \Delta t A\vec{y}^{n+1}) \\
\vec{w}^{n+1} &= \vec{w}^n + \Delta t \Lambda\vec{w}^{n+1} \\
\vec{w}_i^{n+1} &= \vec{w}_i^{\,n} + \Delta t \lambda_i w_i^{n+1} \\
w_i^{n+1} &= \frac{1}{1-\ \Delta t\lambda_i} w_i^n
\end{aligned} \tag{3.21}$$

Since $\lambda_i < 0$ it follows that $\left|\frac{1}{1-\ \Delta t\lambda_i}\right| \leqslant 1$. Hence, the Backward Euler scheme is unconditionally stable.

3.5. Computing Image Sharpness

Our approach for computing sharpness is to use a gradient-based method. We compute the gradient matrix of our image along each x and y direction separately. Then, we square each corresponding entries of the gradient matrix, sum them, and take the square root. The gradient-based approach is shown below:

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{3.22}$$

where G is our gradient matrix, and G_x and G_y are the gradient matrix along the x and y direction of the reshaped image matrix. Once our gradient matrix of the processed image is defined, we are ready to compute the sharpness of the image as follows:

$$sharpness = \frac{(-0.1\sum_{i=1}^{N}\sum_{j=1}^{N} G_{ij})}{N} + error \tag{3.23}$$

where

$$error = \frac{\|u_o - ku\|}{\|u_o\|} \tag{3.24}$$

such that u_0 is the originally blurred image, and k is the blurring kernel applied to the enhanced vectorized image u.

3.6. Nelder-Mead Optimization

The module that is responsible for automatic detection of the unknown value of the blurring kernel including optimization of k and δ variables from the PDE model is Nelder-Mead optimization. Nelder-Mead is a derivative-free optimization method and comes with the advantage of not having to deal with computing the derivative of the objective function. We attempt to maximize the sharpness of our image by minimizing the $-sharpness$ function shown in (3.23). Nelder-Mead optimization is based on the idea of a simplex, where Nelder-Mead uses $n+1$ test points arranged as a simplex for an n-dimensional optimization space. That is, to optimize two variables in 2-D space, Nelder-Mead uses three test points arranged as a simplex (triangle) to extrapolate the behavior of the objective function measured at each of the $n+1$ test points in order to find a new test-point and replace the oldest of these points with the new test-point.

Chapter 4

Results

Having discussed the methods that were used in automatic detection of the parameter value of the blurring kernel in the previous section, we are ready to discuss the parameter choices used to run our numerical experiments and tabulate results obtained from several image runs. Let us begin by reiterating our model containing the fidelity term:

$$u_t = \nabla \cdot \left[\left(\frac{1}{1+k^2|\nabla u|^2} + \delta |\nabla u|^{p-2} \right) \nabla u \right] + \lambda K(c)'(u_o - K(c)u) \tag{4.1}$$

where $K(c)$ is a function of blurring parameter c defined as a Gaussian blur:

$$\begin{aligned} K(i,j) &= e^{\frac{\left(\frac{-128 w_{ij}}{N}\right)^2}{|c|}} \\ W_{ij} &= \sqrt{(W_{2x})^2 + (W_{2y})^2} \\ [W_{2x}, W_{2y}] &= meshgrid(w) \end{aligned} \tag{4.2}$$

where, *meshgrid* returns matrices consisting of the x and y coordinates of all possible ordered pairs of values in w, which contains the range of frequencies in the order that MATLAB uses in its *fft* and *ifft* functions. As we already know the lower the value of the blurring parameter c, the greater is the blurring effect and vice-versa. Since the blurring parameter c is one of the most important variables that distorts (blurs) any given image in our experiment, its automatic detection is crucial to automatic enhancement (sharpening). As a result, we perform 1-D Nelder-Mead optimization on the blurring parameter c to guess the correct value of the blurring parameter. The algorithm and constraints used to settle on its optimal value is shown below:

```
$c1 \leftarrow nelderMead(f(x, tstep, fileName, rows, cols, blur\_value), 150)$
$c2 \leftarrow nelderMead(f(x, tstep, fileName, rows, cols, blur\_value), c1)$
**while** $relative\_error(c1, c2) > 0.05$ **do**
    $c1 \leftarrow c2$
    $tstep1 \leftarrow tstep2$
    $c2 \leftarrow nelderMead(f, c1)$
**end while**
```

Once the optimal value of the blurring parameter c is chosen by Nelder-Mead, our goal is to focus on 2-D optimization of the edge detection parameter (k) and the regularization parameter (δ) in (4.1). The value of another regularization parameter (p) in (4.1) is held constant at 1.05. The rationale for using this value of p comes from the fact that $p \gtrapprox 1$ is believed to be best at simultaneous denoising and deblurring as discussed in [2] and [4]. Likewise, the scaling parameter $\lambda = 100$ throughout our numerical experiments. Guidotti-Kim-Lambers in their experiments for satellite image enhancement have used $\lambda = 30$. In order to blur the original text image, two values of the blurring parameter $c = 10$ and $c = 50$ are used depending on the image and the amount of blur we want to apply. $c = 10$ was chosen for most experiments because it applied a decent amount of Gaussian blur. Any other values other than $c = 10$ and $c = 50$ can also be experimented with. However, it should be noted that too much blurring can wreck the image and make the image recovery next to impossible. Hence, the blurring value should be chosen cautiously.

Below, five different text-images that were used in the experimentation of our model is shown. The images are arranged in the following order: original image, blurred image and enhanced image. Sharpness values for each of these images have also been computed and their relative sharpness error given by the sharpness error metric in (3.24) is reported in Table 4.2

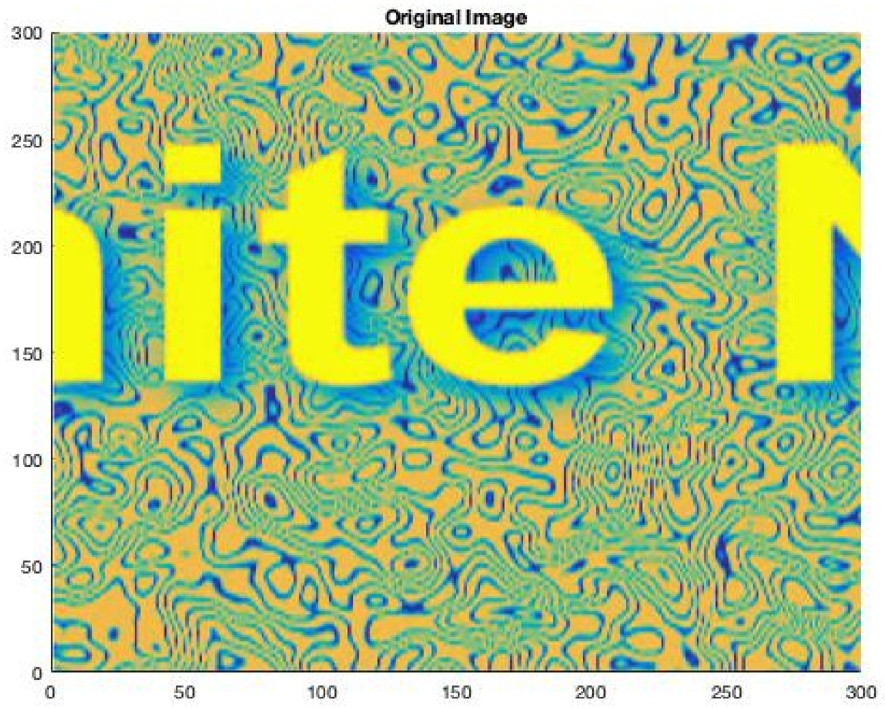

Figure 4.1: Original Reference Image with Sharpness Value=32.9512

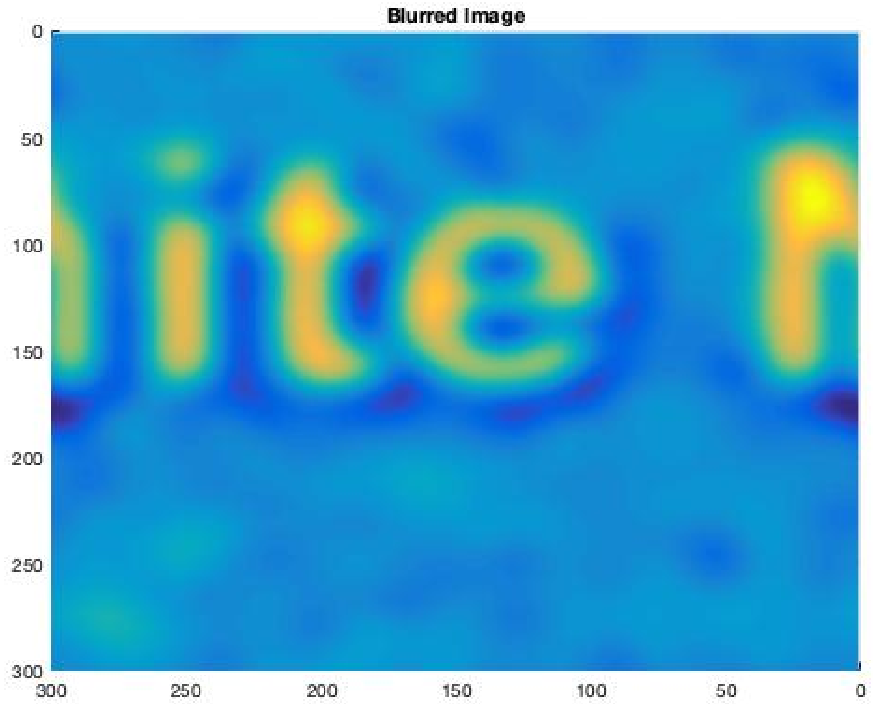

Figure 4.2: Blurred Reference Image with Sharpness Value=1.2305

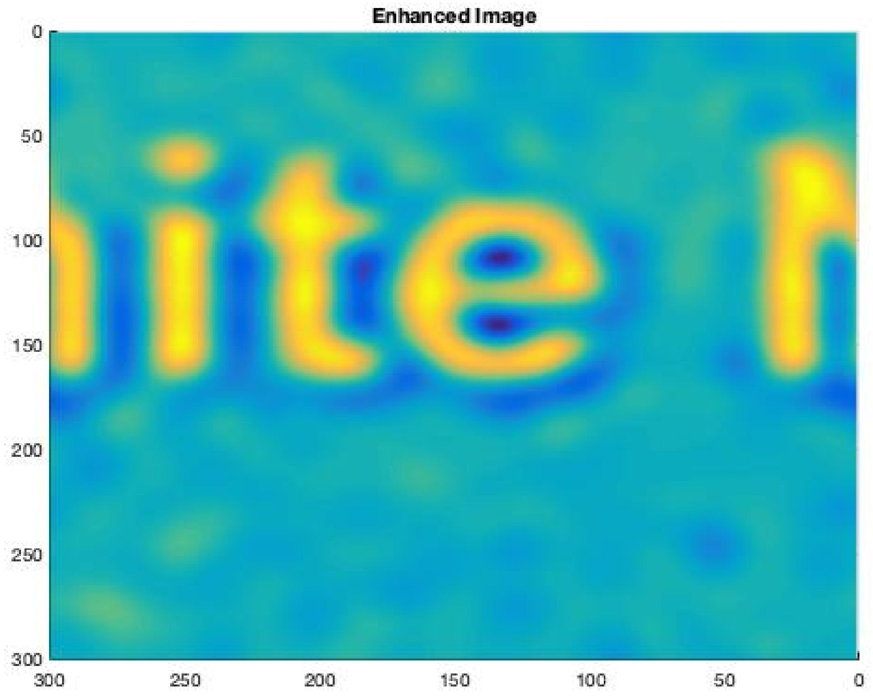

Figure 4.3: Enhanced Image with Sharpness Value=3.4076

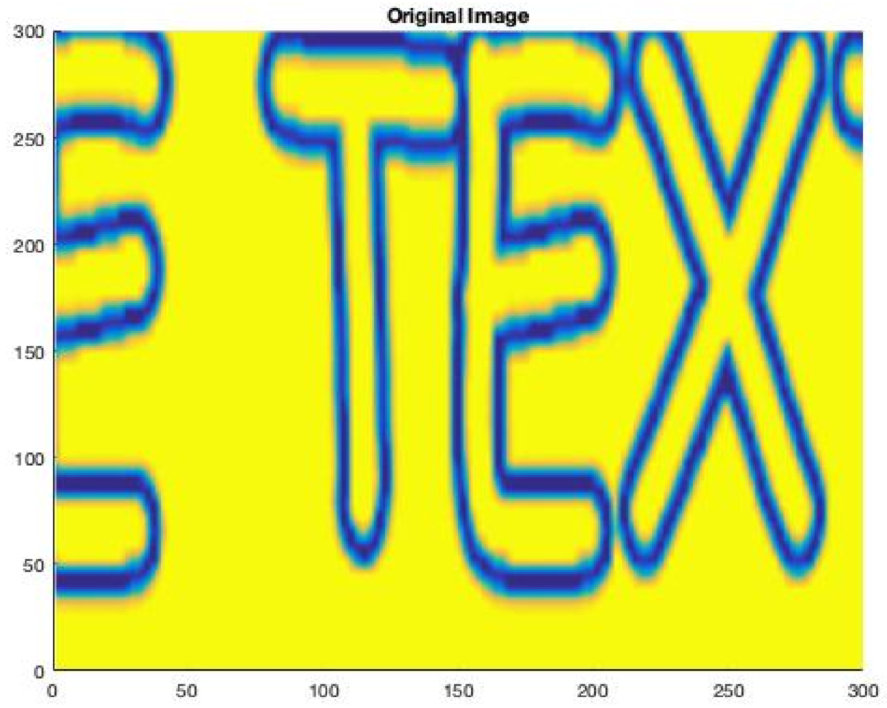

Figure 4.4: Original Reference Image with Sharpness Value=13.5801

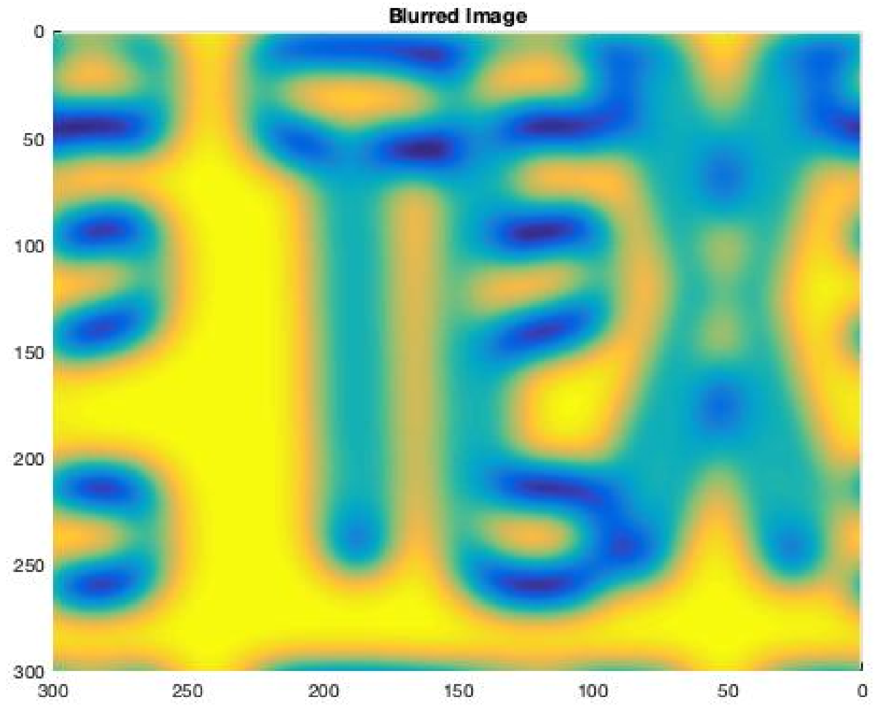

Figure 4.5: Blurred Reference Image with Sharpness Value=2.8709

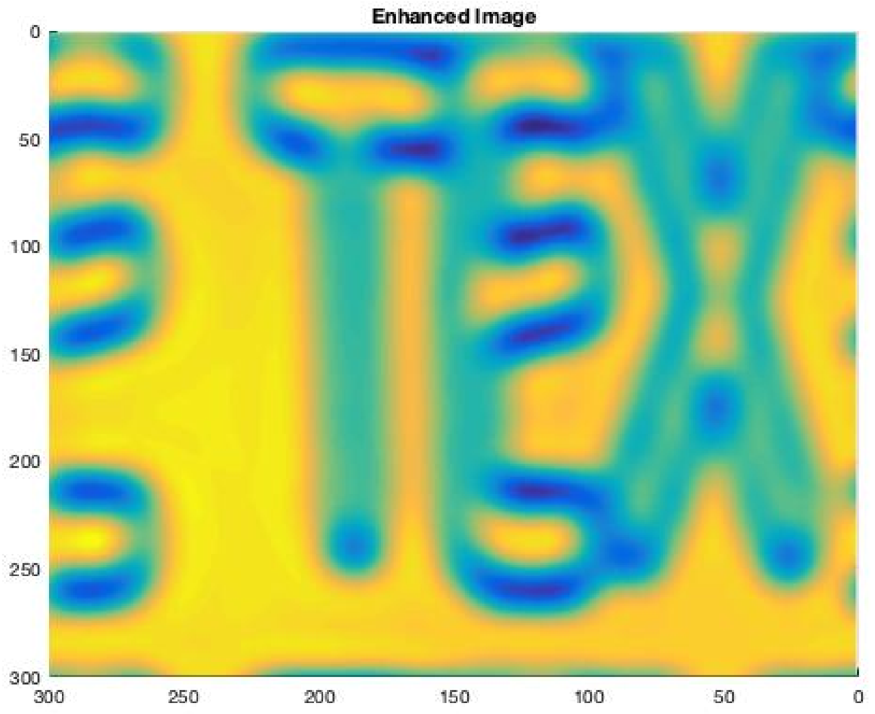

Figure 4.6: Enhanced Image with Sharpness Value=5.3651

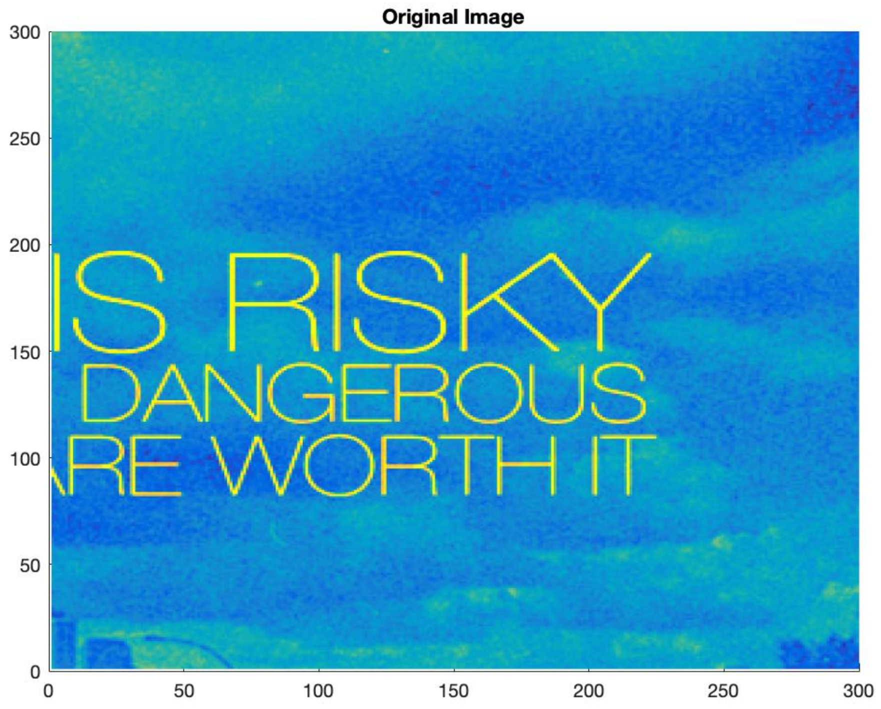

Figure 4.7: Original Reference Image with Sharpness Value=12.6909

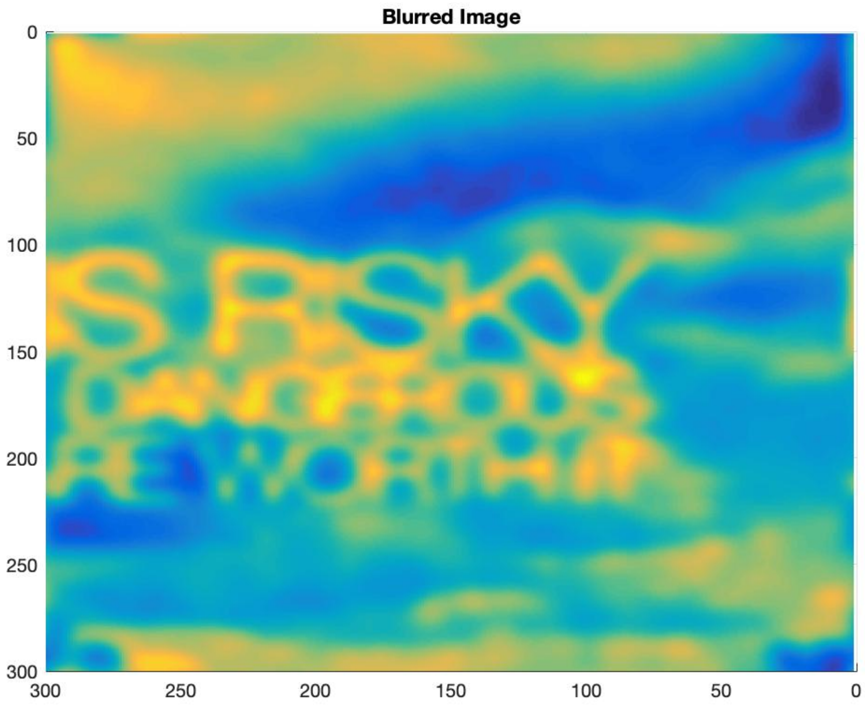

Figure 4.8: Blurred Reference Image with Sharpness Value=1.6172

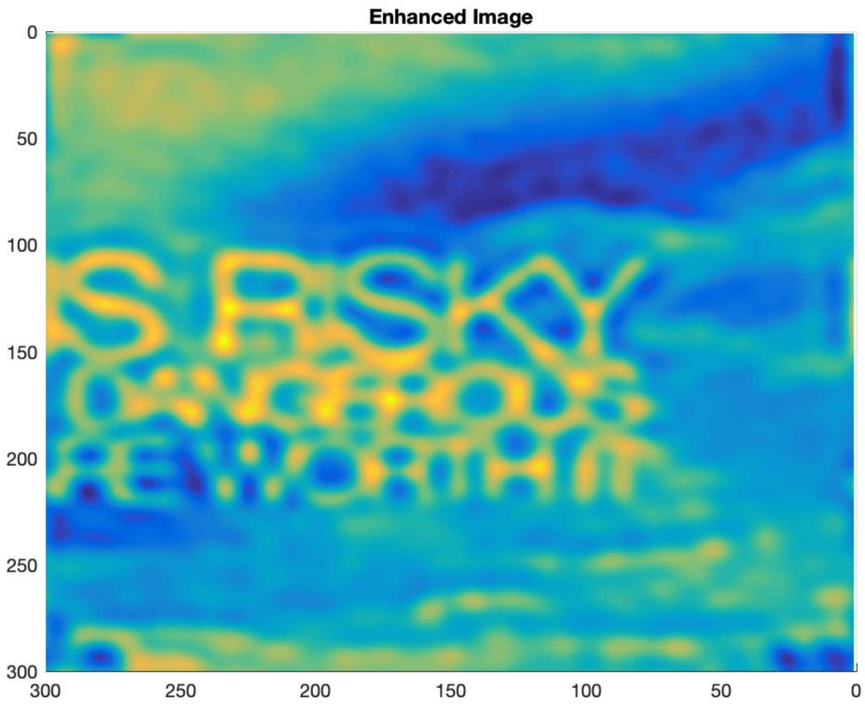

Figure 4.9: Enhanced Image with Sharpness Value=5.8053

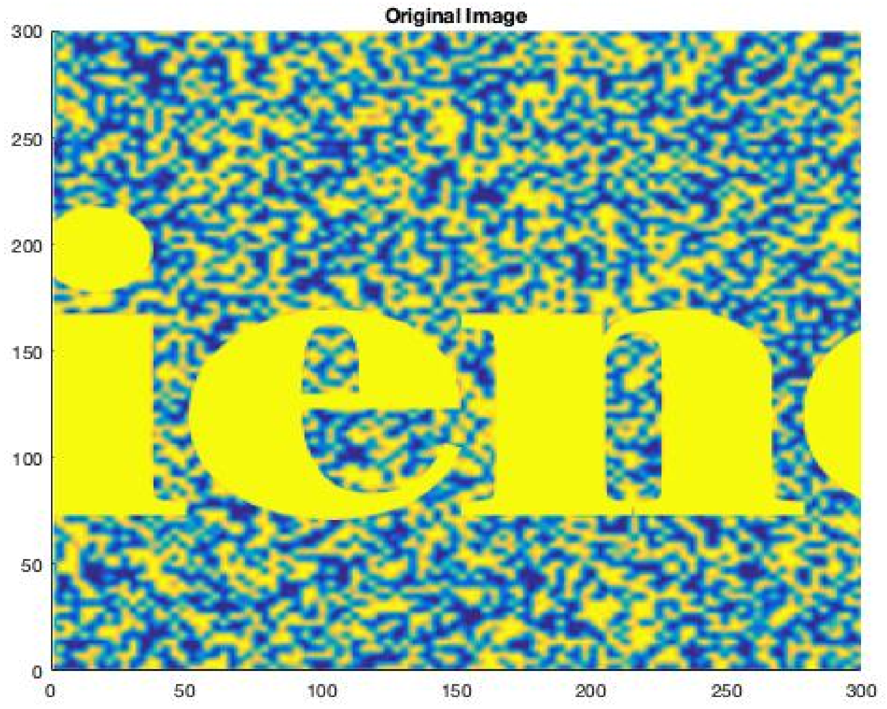

Figure 4.10: Original Reference Image with Sharpness Value=32.9244

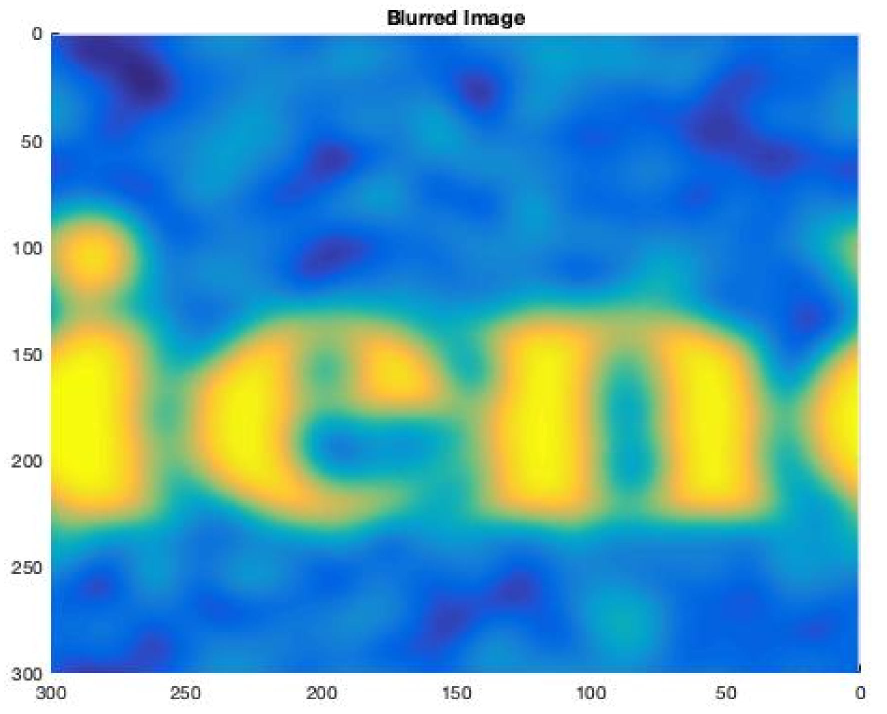

Figure 4.11: Blurred Reference Image with Sharpness Value=1.9340

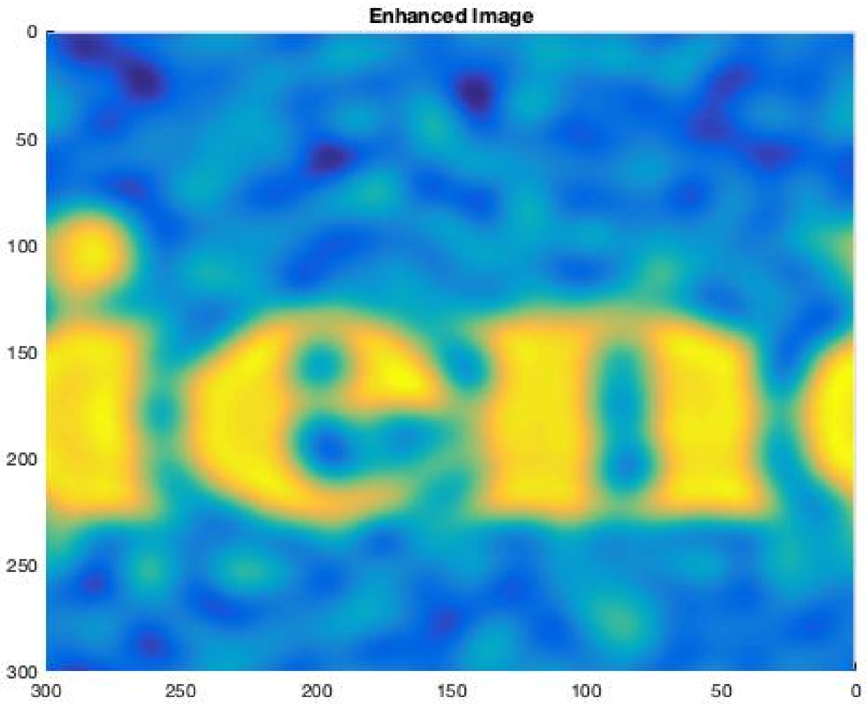

Figure 4.12: Enhanced Image with Sharpness Value=3.9185

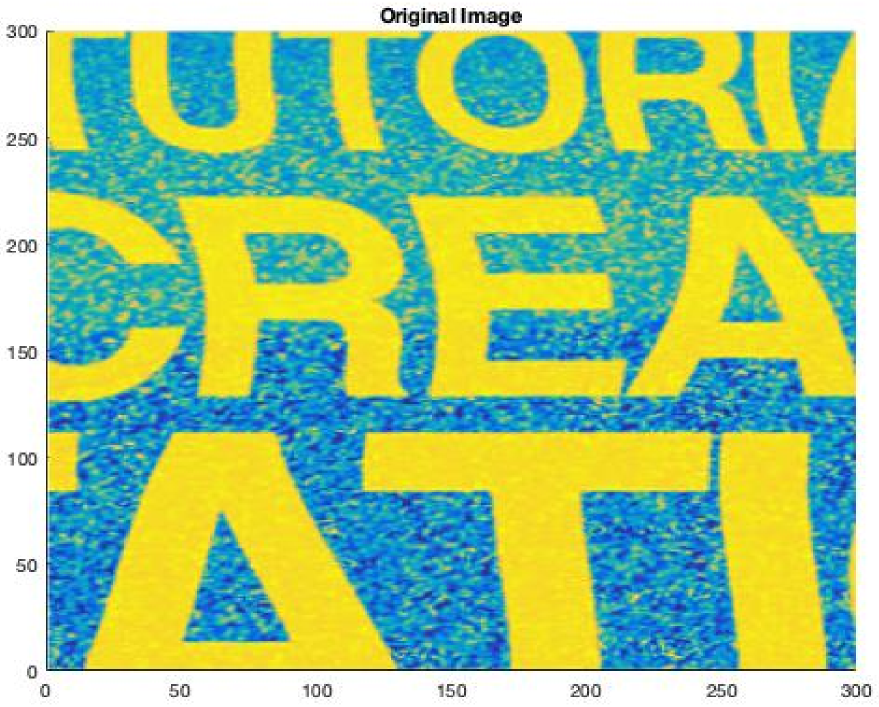

Figure 4.13: Original Reference Image with Sharpness Value=22.5562

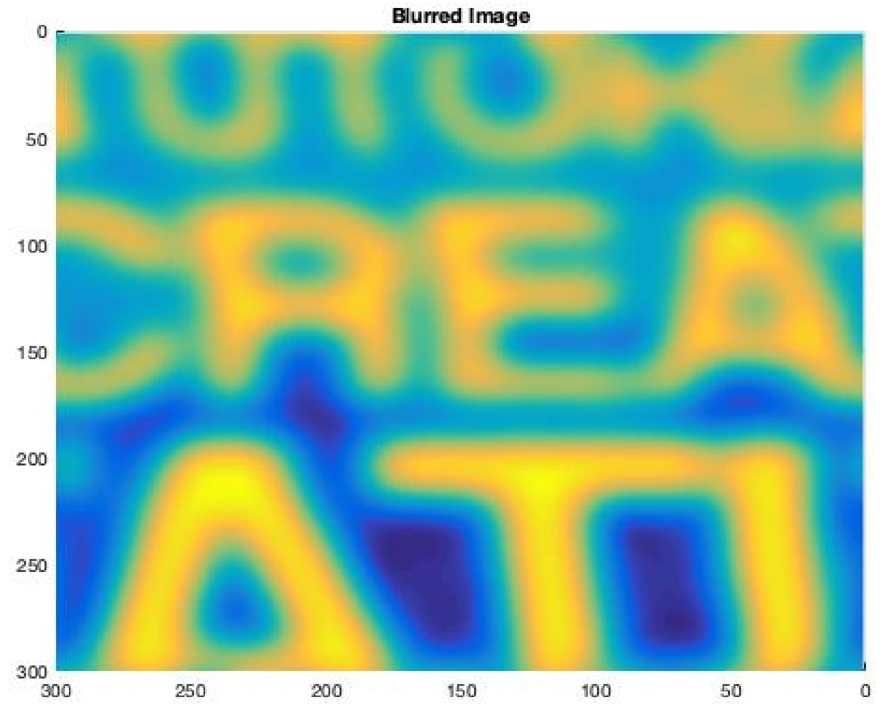

Figure 4.14: Blurred Reference Image with Sharpness Value=3.0452

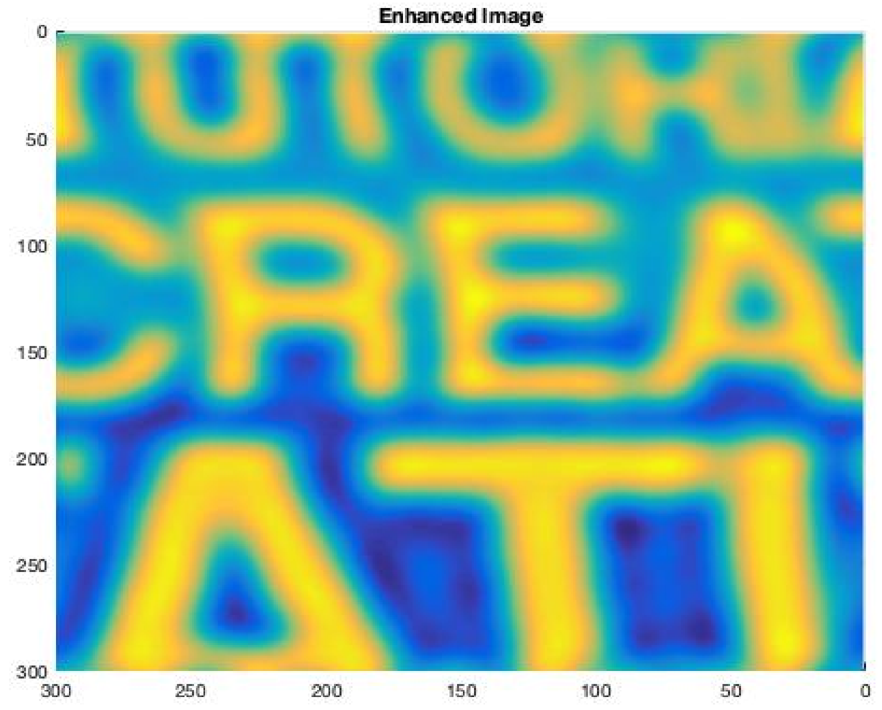

Figure 4.15: Enhanced Image with Sharpness Value=6.7778

For each enhanced image above, their edges are more distinct than their blurred version. Although it is not possible to recover the original image as it is, the model is very accurate in identifying edges and enhancing image to some degree.

The table below summarizes the initial parametric value choices for k, δ, and c along with the optimizer-guessed automatic parametric value choices.

Figure	Initial k	Optimal k	Initial d	Optimal d	Actual c	Predicted c
4.3	0.001	-0.2026	0.1	-1.9759	10	11.0764
4.6	0.001	0.3297	0.1	0.0978	10	19.2430
4.9	0.001	0.3145	0.1	-0.0002	50	31.4934
4.12	0.001	1.1745	0.1	0.9189	10	-12.3092
4.15	0.001	1.4106	0.1	0.4139	10	13.8149

Table 4.1: Table summarizing the initial guess for k and δ provided to the optimizer (initial k and δ), including optimizer-guessed optimal values of k and δ (optimal k and δ) for best possible enhancement. The last two columns consists of actual values of the blurring kernel c as compared to the optimizer-predicted values.

Likewise, the table below contains information related to solving the system during each time-step including the total number of time-steps required to achieve the accuracy.

Figure	Sharpness Value	MINRES Iterations	MINRES Tolerance	Time-Steps	Sharpness Error
4.3	3.4076	3	10^{-6}	10	0.4813
4.6	5.2509	21	10^{-6}	10	0.2096
4.9	5.8053	2	10^{-6}	10	0.3892
4.12	3.9185	24	10^{-6}	10	0.3663
4.15	6.7778	17	10^{-6}	10	0.1457

Table 4.2: Table summarizing the sharpness value of the enhanced image, MINRES iterations required to solve the system, MINRES tolerance used for solving the system, total number of time steps before arriving at maximum sharpness and relative sharpness error of the enhanced image.

Although the number of time-steps appear to be fixed in Table 4.2, it is important to note that the constraint actually forces the iteration to run for at least 10 time-steps until a diminished sharpness value is encountered. The reason for choosing a minimum value for the time-step is to avoid the algorithm to prematurely land at compromised image sharpness. The constraint that is imposed for solving the system is shown below:

```
while ($number\_of\_timesteps > 10$ or $new\_sharpness < old\_sharpness$) do
    PDE Solver
```

```
end while
```

It may seem that the *while* loop runs indefinitely if the sharpness keeps increasing. However, it is important to note that our model prevents arbitrary growth of sharpness with the guaranteed prevention of staircasing. This advantage comes from the regularization term that accounts for the forward-backward-forward diffusion. That is, when image becomes arbitrarily sharp, the model switches from deblurring (backward diffusion) to denoising (forward diffusion). However, one of the limitations to this constraint is that the sharpness metrics computed at smaller time-steps (less than 10) are discarded because the algorithm is allowed to run for a minimum of 10 time-steps.

Chapter 5

Conclusion

Through numerical experiments, we have seen that our proposed sharpness metric agrees with the enhanced images. The result shows that the sharpness metric is highest for the original reference image and least for the blurred image. The sharpness value for the enhanced image is between the original reference image and the blurred image. This means that the proposed objective metric agrees with the subjective assessment. Although the initial blurring value and the optimizer-guessed value do not perfectly coincide, the prediction of c is still produced closer to the true blur value. The automatic method certainly comes with the cost of having to run the Nelder-Mead optimizer until the convergence of the parameters k, δ and c. Likewise, the proposed model is effective in enhancing (sharpening) text-based images, thereby agreeing to our desired goal. However, one of the challenges during the 2-D optimization of k and δ was that the 2-D Nelder-Mead method failed to converge. This problem was solved by changing the simultaneous 2-D Nelder-Mead optimization into multi-stage 1-D Nelder-Mead optimization.

Future research direction for this project could deal with experimentation with other optimization techniques to compare if they are faster than Nelder-Mead. Likewsie, it might also be worth investigating if fewer time steps are possible to obtain an enhanced image of similar sharpness.

Appendix A

MATLAB Code

A.0.1 Driver Function

```
function [c]=multi_iterNelMead

%Image Specifications
fileName='test4.jpg';
rows=1:300;
cols=201:500;
blur_value=50;

%Reading the Image
A=double(imread(fileName));
A2=A(rows,cols,1);
[p,q]=size(A2);
B2=blurimage(A2,size(A2),blur_value);

%Nelder-Mead Optimization for c
tstep1=4;
tstep2=10;
c1=nelderMead(@(x)(r0(x,tstep1,fileName,rows,cols,blur_value)),150)
c2=nelderMead(@(x)(r0(x,tstep2,fileName,rows,cols,blur_value)),c1)
while abs((c1-c2)/c2)>0.05
    c1=c2;
    tstep1=tstep2;
    tstep2=2*tstep2;
    c2=nelderMead(@(x)(r0(x,tstep2,fileName,rows,cols,blur_value)),c1)
end

% Nelder-Mead Optimization for k and d
init_k=0.001;
init_Delta=0.1;
k=nelderMead(@(x)(r2(x,10,fileName,rows,cols,blur_value,c2)),init_k);
Delta=nelderMead(@(x)(r4(x,10,fileName,rows,cols,blur_value,c2,k)),
    init_Delta);
c2
optimal_x=[k,Delta]
r3(optimal_x,c2,fileName,rows,cols,blur_value)

%Compute and Display the Sharpness of Original Image
[Gx, Gy]=gradient(A2);
S=sqrt(Gx.*Gx+Gy.*Gy);
ori_sharp=sum(sum(S))./(numel(Gx))


```

```
%Compute and Display the Sharpness of Blurred Image
[Gx, Gy]=gradient(B2);
S=sqrt(Gx.*Gx+Gy.*Gy);
blur_sharp=sum(sum(S))./(numel(Gx))
disp('done')
```

A.0.2 PDE Solver

```
function [old_sharpness]=solvePDE(x,c,fileName,rows,cols,blur_value)

k=x(1);
Delta=x(2);
p0=1.05;
lambda=100;
if (lambda>1000)
    sharpness=0;
    return
end

%Reading the Image
A=double(imread(fileName));
A2=A(rows,cols,1);
[p,q]=size(A2);
N=p;
B2=blurimage(A2,size(A2),blur_value);

%Display Original Image
figure(1);
surf(A2(end:-1:1,:));
view(0,90);
title('Original Image')
shading flat;

%Convert matrix columns into a vector
Avector = reshape(A2',[],1);
Bvector = reshape(B2',[], 1);
u0=Bvector; % (u0 is solution at t=0)

%Forward Difference matrix (one-dimensional)
e=ones(N,1);
Dplus=spdiags([-e,e], [0,1], N, N);
Dplus(N,1)=1;
%Backward Difference matrix (one-dimensional)
Dminus=-(Dplus)';
%Centered Difference matrix (one-dimensional)
Dcenter=(Dplus+Dminus)/2;

sparseI=speye(N,N);

% Create n^2xn^2 Forward Difference wrt x,
Dxplus=kron(Dplus, sparseI);
% Create n^2xn^2 Backward Difference wrt x,
Dxminus=kron(Dminus, sparseI);

% Forward Difference wrt y,
Dyplus=kron(sparseI, Dplus);
% Backward differentiation wrt y
Dyminus=kron(sparseI, Dminus);

% Centered Difference wrt x,
Dxcenter=kron(Dcenter, sparseI);
% Centered Difference wrt y
Dycenter=kron(sparseI, Dcenter);

%Display Blurred Image
u=u0;
mu=min(u);
MU=max(u);
u=(u-mu)*255/(MU-mu);
finalA=(finalA-mu)*255/(MU-mu);
figure(2);
surf(finalA(end:-1:1,:)');
title('Blurred Image');
view(180,90);
shading flat;

m_ind=0:p-1;
n_ind=0:q-1;
[m2,n2]=meshgrid(m_ind, n_ind);
lambd=-4*sin(m2*pi/p).^2-4*sin(n2*pi/p).^2;
lambda_reshape=reshape(lambd, numel(lambd), 1);
n=0;
sharpness=0;

%Backward Euler and MINRES
while (n<10 || sharpness>old_sharpness)%atleast 10 time-steps

    ux=Dxcenter*u;
    uy=Dycenter*u;
    uxy=ux.^2+uy.^2;
    uxy_updated=ux.^2+uy.^2;
    for j=1:length(uxy_updated)
    if (uxy_updated(j)==0)
        uxy_updated(j)=0.0001; %if the value is zero, small non-zero
    end
    end

    %PDE coefficients
    gu=(1./(1+k^2*uxy)+abs(Delta)*(uxy_updated).^(p0-2)/2);
    g0=mean(gu);
    %construct a sparse diagonal matrix from the coeffients
    G=spdiags(gu, 0, N^2, N^2);

    %symmetric positive semi-definite matrix A
    A=Dxplus*G*Dxminus+Dyplus*G*Dyminus;


    I=speye(N^2, N^2);
    %var comes from Backward Euler Time-Stepping
    var=I-dt*A;
    %Error Metric
    error=norm(u0-blurimage(u,[N,N],c))/norm(u0);
    %Fidelity Term
    rhs=(u+dt*lambda*blurimage(u0-blurimage(u,[N,N],c),[N,N],c));
    %MINRES with preconditioner
    [u_new,¬,¬,¬,RESVEC]=minres(var,rhs, 1e-6, 100, ...
       @(b)mfun(b,g0,dt,lambda_reshape));
    %Update the Image
    u=u_new;
    %Reshape image vector into matrix
    finalA=reshape(u, N, N);

    %Scaling
    mu=min(u);
    MU=max(u);
    u_scaled=(u-mu)*255/(MU-mu);
    finalA=reshape(u_scaled, N, N);

    %Update Image Sharpness
    [Gx, Gy]=gradient(finalA);
    S=sqrt(Gx.*Gx+Gy.*Gy);
    old_sharpness=sharpness;
    sharpness=0.1*sum(sum(S))./(numel(Gx))
    if (isnan(sharpness))
        sharpness=0;
    end
    n=n+1;
end
iterations=n

%Display Enhanced Image
figure(3)
surf(finalA(end:-1:1,:)');
view(180,90);
title('Enhanced Image');
shading flat;
pause(0.1)

%Compute and Display Sharpness of Enhanced Image
[Gx, Gy]=gradient(finalA);
S=sqrt(Gx.*Gx+Gy.*Gy);
enh_sharpness=sum(sum(S))./(numel(Gx))
error
```

Bibliography

[1] J. V. Lambers, *“Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration",* Electronic Transactions on Numerical Analysis 31 (2008), p. 86-109.

[2] P. Guidotti, Y. Kim, and J. V. Lambers, *“Image restoration with a new class of forward-backward-forward diffusion equations of Perona-Malik type with Applications to Satellite Image Enhancement",* SIAM Journal on Imaging Sciences 6(3) (2013), p. 1416-1444

[3] P. Guidotti, and J. V. Lambers, *“Two New Nonlinear Nonlocal Diffusions for Noise Reduction",* Journal of Mathematical Imaging and Vision 33 (2009), p. 27-35.

[4] P. Perona, and J. Malik, *“A Computational Approach to Edge Detection",* IEEE Transactions on pattern analysis and machine intelligence 12.7 (1990): 629-639.

[5] J. Canny *“Scale-space and edge detection using anisotropic diffusion",* IEEE Transactions on Pattern Analysis and Machine Intelligence vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986

[6] K. De, and V. Masilamani, *“Image Sharpness Measure for Blurred Images in Frequency Domain",* Procedia Engineering. 64.10.1016/j.proeng.2013.09.086.

[7] Z. Wang, and A. C Bovik., *“Modern Image Quality Assessment, San Rafael, C.A.: Morgan and Claypool",* Academic Pres, ch. 1, pp. 1-15. 2006

[8] R. Ferzli, and L. J. Karam, *“A No-reference Objective Image Sharpness Metric based on the notion of Just Noticeable Blur (JNB)",* IEEE Transactions on Image Processing, vol. 18, no. 4, pp. 717-728. 2009

[9] G. Alphy, and S. J. Livingston, *“A survey on full reference image quality assessment algorithms.",* International Journal of Research in Engineering and Technology 2.12 (2013): 303-307.

[10] N. Zhang, A. Vladar, M. Postek, and B. Larrabee, *“A kurtosis-based statistitcal measure for two-dimensional processes and its application to image sharpness",* Proc. Section of Physical and Engineering Sci- ences of American Statistical Society, 2003, pp. 4730-4736.

[11] J. Caviedes, and F. Oberti, *“A new sharpness metric based on local kurtosis, edge and energy information",* Signal Processing: Image Communication 19.2 (2004): 147-161.

[12] C. F. Batten, *“Autofocusing and Astigmatism Correction in the Scanning Electron Microscope",* M.Phil. thesis, Univ. Cambridge, Cam- bridge, U.K., 2000.

[13] L. Firestone, K. Cook, N. Talsania, and K. Preston, *“Comparison of autofocus methods for automated microscopy",* Cytometry, vol. 12, pp. 195-206, 1991.

[14] S. Erasmus, and K. Smith, *“An automatic focusing and astigmatism correction system for the SEM and CTEM,",* J. Microscopy, vol. 127, pp. 185-199, 1982.

[15] A. Buades, C. Bartomeu, and M. Jean-Michel *“A review of image denoising algorithms, with a new one",* Multiscale Modeling & Simulation 4.2 (2005): 490-530.