OPTIMISATION IN IMAGE PROCESSING

HON WEN XUAN

UNIVERSITI TUNKU ABDUL RAHMAN

OPTIMISATION IN IMAGE PROCESSING

HON WEN XUAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Hon Wen Xuan

ID No. : 2001171

Date : 16 May 2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"OPTIMISATION IN IMAGE PROCESSING"** was prepared by **HON WEN XUAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ______________________

Supervisor : Dr. Sim Hong Seng

Date : 18/5/2023

Signature : Mohammad Babrdel Bonab

Co-Supervisor : Mohammad Babrdel Bonab

Date : 18/05/2023

ACKNOWLEDGEMENTS

I would like to thank everyone who contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Sim Hong Seng for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents, cousins, my previous academic advisor Dr. Loo Yim Ling, my current academic advisor Dr. Lee Chen Kang, and my friends who had helped and given me encouragement in my final year project.

ABSTRACT

Sometimes when we take a photo using the camera, our hand will shake unexpectedly and cause motion blur on the image, and this is where the deblurring process is needed. Deblurring is the process of turning the blurred image into a clearer image. Some deblurring methods, however, cannot be utilised to correctly deblur images or cannot effectively recover the image to make it appear like the original image to human eyes. This project applies various optimisation methods such as Steepest Descent, Conjugate Gradient, and Barzilai-Borwein method to the blur images and recovers them into clearer images, with the help of the Armijo rule and Lipschitz inequality to find the step size. The whole process is being done using Python. Peak signal-to-noise ratio and structural similarity index measure are used to evaluate the recovered image. Results show that among the proposed optimisation algorithms, the best algorithm to deblur image is the conjugate gradient with the Armijo rule with the peak signal-to-noise ratio of 94.76158623 and structural similarity index measure of 0.999999439, and the worst algorithm is the steepest descent method with the Armijo rule. The drawbacks of the proposed algorithms are also discussed, and suggestions for further research are given. This project, in its entirety, emphasizes the significance of optimisation in image processing and offers a few potential algorithms for enhancing the precision and effectiveness of image deblurring tasks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

A	Blurring Factor
x	Original Image
B	Blurred Image
x_k	Real Numbers
α_k	Step Size
d_k/p_k	Search Direction
c_1	Constant
$\nabla f_k/g_k$	Gradient of Function
r_k	Residual

PSF	Point-spread Function
IQA	Image Quality Assessment
GAN	Generative Adversarial Network
R-SD	Riemannian steepest descent
SD	Steepest Descent
CG	Conjugate Gradient
BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm
BB	Barzilai-Borwein gradient method
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Optimisation has always been in our life for a long time. Optimisation can be applied in mathematical functions or software programming, in real life like determining schedules in airline services and how to best run machinery in manufacturing plants. In the mathematical functions, we are using optimisation to maximise or minimise the output. The purpose of optimisation is to “make something best”, to make use of a situation or resources most effectively. In this project, we will be using various optimisation algorithms in image deblurring problems.

1.2 Background of the Problem

In image processing, we can apply various blurs to an image, for example, average blur, gaussian blur, bilateral blur, etc. To get the blurred image, we simply multiply the original image by the blurring factor. The function is shown below:

$$Ax = B$$

where
A = Blurring factor
x = Original image
B = Blurred image

Deblurring is the process of removing distortions involved with blurring from images by using a mathematical model of a blurring process, in another word, to sharpen the image and make it clearer to human eyes. People deblur images to recover blurred images usually caused by motion or defocusing when taking a photo, or any other kind of blur. To deblur image, we need to know A (blurring factor) and B (blurred image) and apply the optimisation methods.

However, there are also problems when deblurring images. We are not able to recover the blurred image to the same as the original image. This is due to numerous inherent errors in the captured image. Variations in the recording procedure and approximation errors while representing the image with a limited number of digits are the two most significant errors. Developing effective and dependable methods to extract as much information as possible from the provided data is also one of the problems of image deblurring (Hansen et al., 2006, pp.1-2).

1.3 Problem Statement

Research from Hansen, Nagy and O'Leary (2006, pp.4-5) showed that the naïve solution to deblur image that is based on the image blurring model failed to recover any features of the original image. This is because the model ignored some types of errors. Since the blurry image was captured via a mechanical device, noise and small random errors were expected to be present in the recorded data. But the deblurring model assumes that there will be no error when the system is collecting data from the blurred image.

Besides that, according to Chan, Rajakaruna, Rathnayake and Murray (2014, p.1243), deblurring methods can be divided into Blind deconvolution and Non-blind deconvolution as the two main types, and they used the point-spread function (PSF) as their blurring function. When the PSF is unknown, the blind deconvolution can be used to deblur images and when the PSF is known, the non-blind deconvolution can be used. The PSF can be estimated using embedded inertial sensors like a 3-axis accelerometer and a 3-axis gyroscope thanks to recent advancements in mobile technology. However, using the accelerometer signal to calculate the camera's velocity and displacement is inaccurate since noise from the sensor's drift has accumulated in the signal. Due to the exposure time's short duration, it is challenging to develop a suitable PSF for efficient deblurring.

1.4 Aim and Objectives

The objectives of this project are:

1. To investigate and understand the challenges in deblurring images and the mathematical models that represent the deblurring problems.

2. To compare the effectiveness of various optimization algorithms in image deblurring.
3. To provide the recommendations for improving the best optimization algorithms in image deblurring.

1.5 Proposed Solution

To solve the deblurring problem, we can use the improved steepest descent method using the Armijo line search, conjugate gradient method, and Barzilai-Borwein (BB) gradient method.

The steepest descent method finds the lowest value in the image and after combined with the Armijo line search strategy, it finds the optimal step size that searches along the direction to the lowest value using a loop. The loop will end if it reaches the maximum loop or if it satisfies the Armijo condition as follow:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k$$

where

x_k = real numbers

α_k = step size

d_k = search direction

c_1 = constant

∇f_k = gradient of the function

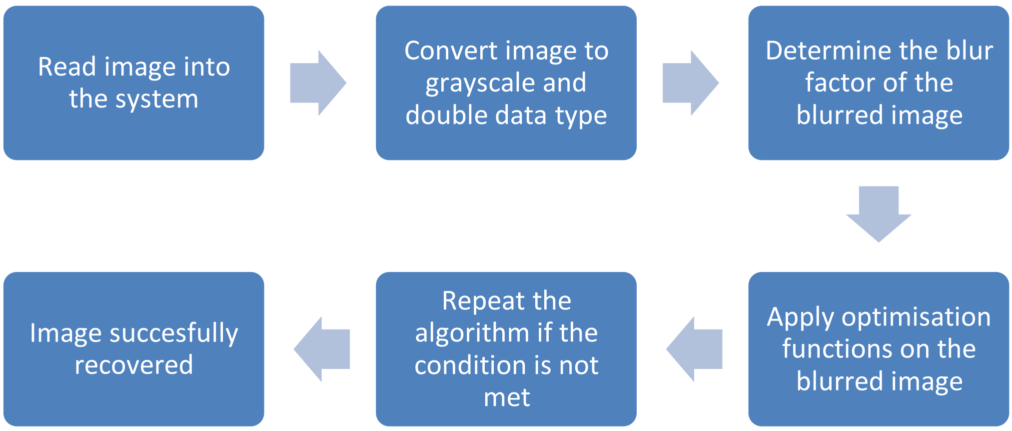
Another algorithm known as the conjugate gradient method is used to numerically solve specific systems of linear equations, provided the matrix is positive-definite. It is often implemented as an iterative algorithm. Energy minimization and other unconstrained optimization issues also can be resolved using the conjugate gradient method. The flow of the conjugate gradient is similar with the steepest descent, but the search direction is calculated differently as below:

$$d_k = -g_k + \beta_k d_{k-1}$$

The detailed implementation of the algorithm will be explained in the methodology section.

The final algorithm that we are employing is the Barzilai-Borwein (BB) gradient method, a family of spectral gradient methods, a well-liked and effective way for resolving sizable unconstrained optimisation issues. Its flow is also similar as the steepest descent method, but its step size rule is different. Its step size can be calculated with Armijo rule or Lipschitz inequality, or by just using the BB methods to calculate. The BB methods is also used to calculate the search direction.

1.6 Proposed Approach



1.7 Scope and Limitation of the Study

This project applies various optimisation methods on the blur images and recovers them into clearer images. The whole process is being done by OpenCV-Python. The source code editor that is used to develop the code in this project is Visual Studio Code. The code will first read an image into the system and apply the blur effect if it is not blurred. Then, the system will try to recover the blurred image using the mathematics formula that is written with python code. In this project, we are using the steepest descent method, conjugate gradient method, and Barzilai-Borwein gradient method to deblur images.

On the other hand, the limitation of the study is that there aren't many resources and journals on the internet or library that focus on optimisation on deblurring methods, which makes this study have some uncertainty as there are fewer resources to refer to. The limitation of the human eyes is also one of the limitations in this study as it is hard to tell the differences between the recovered image by just looking at the images, so the help of the PSNR and SSIM is needed to evaluate the recovered image.

1.8 Contribution of the Study

This study offers two contributions. First, it introduces a few brand-new methods for improving image processing algorithms. When compared to conventional optimisation algorithms, these methods enable the effective optimisation of multi-parameter complex image processing algorithms, improving performance and reducing computation times.

Second, a case study on image deblurring is used in this study to show how effective the proposed optimisation algorithms are. The outcomes demonstrate that, in terms of both objective metrics and subjective visual quality, the optimised image deblurring algorithms outperform several state-of-the-art deblurring methods. The proposed methods are demonstrated to be robust to various types and degrees of blur, making them attractive strategies for a variety of image processing applications.

1.9 Outline of the Report

This report first summarizes the key findings and debates in the field of image processing research, including methods to deblur the images and methods to solve the optimisation problems, in the literature review section. Next, the research design and methodology used in this study will be explained. The results of each of the algorithms used to recover images and the discussion of it will be included in this report after the methodology section. Finally, the study will be concluded and recommendations for future work will be provided.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The deblurring issues in image processing have been the subject of numerous hypotheses aimed at improving them. Although the literature discusses a wide range of these beliefs, this analysis will concentrate on three key points that recur often in the literature under consideration. These points are optimization in image processing, methods to deblur the image in image processing, and methods to solve optimization problems. Although these points are presented in several situations in the literature, the main focus of this project will be on the strategies employed to solve the deblurring problems.

2.2 Literature Review

Recent proposals and research are made related to optimization in image processing. Ikarashi, Ragen-Kelly, Fukusato, Kato, and Igarashi (2021) proposed a programming support method called “guided optimization” that enables programmers to comprehend and efficiently optimize image processing code without the time-consuming trial-and-error process of traditional text editors. Studies have been done on theoretical underpinnings, features, and applications of the algorithm (Wang et al., 2022) and in image quality assessment (IQA) (Ding et al., 2021) to see if their models will contribute to image processing optimization technology. However, none of these studies related to the methods of deblurring images. Knowing the methods to deblur image is important because it will help a lot in this project.

Methods to deblur images are also proposed in several studies. Sada and Mahesh (2018) showed that blind deconvolution techniques gave better results in deblurring images in comparison with non-blind deconvolution techniques. Zhang et al. (2019) proposed an image deblurring method based on Generative Adversarial Network (GAN) architecture using dual path connection, while a study by Khongkraphan, Phonon, and Nuiphom (2021) described an effective method for deblurring images that relies on convolutions and an iterative notion. Throughout the literature, we can know that most of the deblurring techniques will be using convolution and deconvolution.

In order to apply the optimization techniques to the problems, mathematical algorithms must be used. To resolve non-smooth optimization issues with nonlinear inequality and linear equality requirements, a new one-layer recurrent neural network was suggested by Ebadi et al. (2017) that simultaneously integrates the steepest descent and gradient projection approaches. Similarly, Sato (2021) incorporated the Armijo backtracking line search with the Riemannian steepest descent (R-SD) approach to suggest a novel modification. However, in order to solve an unconstrained optimization problem, Abubakar et al. (2021) proposed a hybrid conjugate gradient (CG) scheme that, without the use of a line search, has sufficient descent direction and trust region features. On the other hand, Berahas, Byrd, and Nocedal (2019) suggested the quasi-Newton method for the optimization of noisy functions, which takes advantage of the scalability and power of BFGS updating.

2.3 Discussion

The study by Ikarashi et al., Wang et al., and Ding et al. focus on optimizing image processing systems but differ in their specific objectives and methodologies. Ikarashi et al. introduces a proof-of-concept system, Roly-poly, with two scheduling phases, Wang et al. explores the integration of deep learning and big data, while Ding et al. evaluates and compares IQA models for optimization purposes. Collectively, these papers offer insights into different approaches and tools for improving image processing systems, ranging from programming support methods to architectural implementations and quality assessment models.

Zhang et al., Sada et al., and Khongkraphan et al. focus on the topic of image deblurring but approach it from different perspectives. Zhang et al. proposes a modified approach that combines GANs with traditional deblurring algorithms, achieving improved restoration quality. Sada and Mahesh provides a comprehensive analysis of various deblurring techniques, examining their benefits and drawbacks. Khongkraphan et al. introduces an efficient method that employs a smoothing function to enhance the deblurring process. They contribute to the advancement of image deblurring by providing a detailed review, introducing novel approaches, and presenting efficient methods for enhancing image restoration quality.

On the other hand, Ebadi et al. and Sato both propose techniques for optimizing different types of problems. The former offers a tailored solution for handling nonsmooth optimization problems, while the latter addresses optimization problems on Riemannian manifolds and presents a modification to the Riemannian steepest descent method by incorporating the Armijo condition. While the second study provides a unique improvement to the Riemannian steepest descent approach, the first study makes use of neural networks. These studies demonstrate creative methods for obtaining more efficient and effective optimisation, which helps to optimise many problem domains.

While the studies mentioned above propose innovative approaches for optimization, the steepest descent method, conjugate gradient method, and Barzilai-Borwein method have been widely studied and have a strong theoretical foundation. While the steepest descent method may have slower convergence compared to more sophisticated methods, it is computationally efficient and can be particularly effective in problems with relatively smooth and convex objective functions. Conjugate gradient method is particularly beneficial for problems with large-scale or ill-conditioned systems, as it can effectively handle such scenarios and converge more rapidly compared to the steepest descent method. Barzilai-Borwein method can provide competitive performance for both smooth and nonsmooth optimization problems and has been shown to converge faster than traditional gradient-based methods in certain scenarios.

2.4 Summary

From the various methods that the authors used for optimization, the common step they use to implement the algorithms is to apply the Armijo condition to find the optimal step size for the calculation. After reviewing the literature, it is known that people all over the world have benefited a lot from image processing optimization technology, and the optimization does not converge to the reference image and can generate severe distortions.

The steepest descent method, conjugate gradient method, and Barzilai-Borwein method have been successfully applied to various optimization problems and have demonstrated their effectiveness and efficiency over the years. However, the choice of the most suitable optimization algorithm depends on the specific problem characteristics, and it is always recommended to experiment and compare different methods to identify the most appropriate one for a given scenario.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The steepest descent method, conjugate gradient method, and Barzilai-Borwein (BB) gradient method will be used in this project to solve the optimization problem in image processing. Each of these optimisation algorithms will be implemented with two different approaches, which is by using the Armijo condition and Lipschitz inequality to find the step size, with the exception of BB method, which includes an additional method to find the step size – by using the BB method itself.

3.2 Methodology

Before we implement the optimisation algorithms, we will first develop our blurring matrix. The blurring matrix that is used in this project is the Toeplitz matrix dragging effect. The method is shown as below:

$$k = n$$

$$\begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} & \frac{1}{n} & 0 \\ 0 & \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\ 0 & 0 & \frac{1}{n} & \dots & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\ \vdots & \vdots & \vdots & \ddots & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\ 0 & 0 & 0 & 0 & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{n} & \frac{1}{n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{n} \end{pmatrix}$$

where

$k =$ Kernel size

Matrix size = the size of the image

Figure 3.1 shows the original Lena image in grayscale and figure 3.2 shows the image after applying the Toeplitz matrix dragging effect with kernel size set to 20.

Figure 3.1: Original grayscale Lena image

Figure 3.2: Blurred Lena image

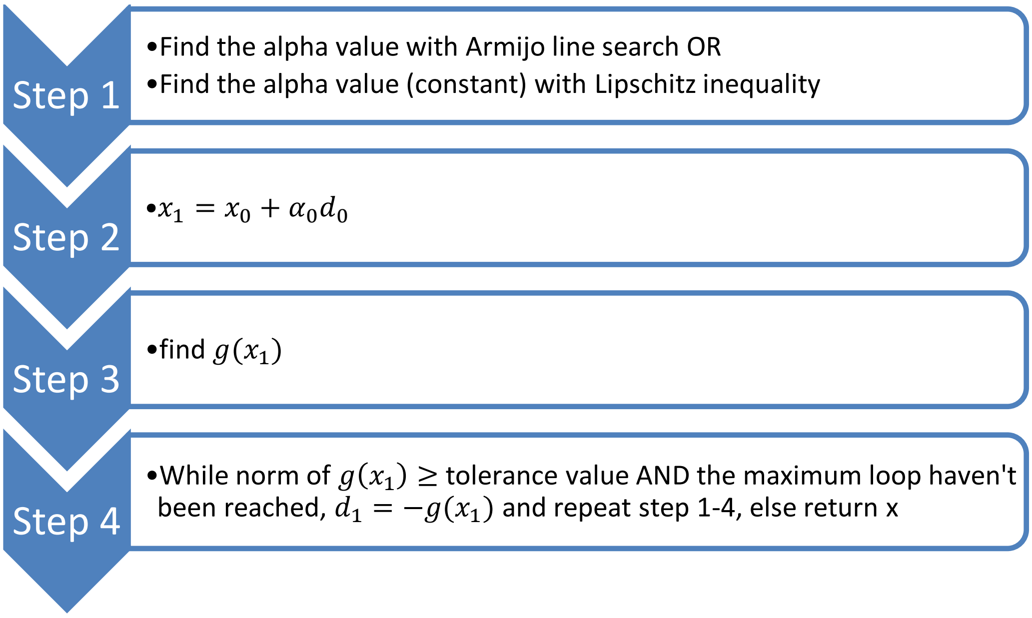
Now we set the blurring factor to A (we use the Toeplitz matrix dragging effect in this case) and the blurred image to B. Assume that the original image is unknown, we set it to x and assign the initial guessing value of all ones to the matrix x use which we will be passing it to our functions to recover the image later.

After identifying the blurring factor and the blurred image, first we will define a function $f(x)$ that returns $\|Ax - B\|_2^2$ which is the square of the Frobenius norm. Then, we will calculate a function $g(x)$ that returns $2A^T Ax - 2A^T B$ which is the gradient of the function $f(x)$. Next, we will get the matrix g by passing the matrix x (the matrix with initial guessing value, denoted as

x_0) to the function $g(x)$ and we will also get the matrix d which is the negative of matrix g. Finally, we will pass the matrix x, g, and d (the initial values, denoted as x_0, g_0, d_0) to the proposed methods and set the desired tolerance value and maximum loop, and the image will be recovered.

3.2.1 Steepest Descent

In order to approximate an integral using the steepest descent approach, one deforms a contour integral in the complex plane to pass relatively close to a stationary point (saddle point), generally in the direction of steepest descent or stationary phase. Few steps will be implemented in the steepest descent method:



When finding the alpha value, we use the Armijo line search:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k$$

which will accept the matrix x, g, and d that we have defined before as its parameters, or the Lipschitz inequality:

$$\alpha = \frac{1}{L}$$

where

L = 2 * max eigenvalue of $A^T A$

which the alpha value will be constant in the whole process.

3.2.2 Conjugate Gradient

Another optimisation technique known as the conjugate gradient method is used to numerically solve specific systems of linear equations, namely those whose matrix is positive-definite. Various nonlinear conjugate gradient methods seek minima of nonlinear optimization problems. Conjugate gradient, assuming exact arithmetic, converges in at most n steps, where n is the size of the matrix of the system.

To implement the conjugate gradient method, we will do the similar steps as steepest descent method, but with the different formula for the search direction after the first iteration instead of just $d = -g$. The formula of d after the first iteration is shown as below:

$$d_k = -g_k + \beta_k d_{k-1}$$

where

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$$

The steps to implement the conjugate gradient method are:

Step 1
- Find the alpha value with Armijo line search OR
- Find the alpha value (constant) with Lipschitz inequality

Step 2
- $x_1 = x_0 + \alpha_0 d_0$

Step 3
- $g_1 = g(x_1)$

Step 4
- While norm of $g_1 \geq$ tolerance value AND the maximum loop haven't been reached, $d_1 = -g_1 + \beta_1 d_0$ and repeat step 1-4, else return x

3.2.3 Barzilai-Borwein Gradient

The Barzilai-Borwein (or BB) gradient method was introduced by Barzilai and Borwein and uses a two-point step size gradient. The step size in the method is determined by a two-point approximation of the secant equation that underlies quasi-Newton methods. The BB method accelerates convergence of the gradient method significantly while requiring less computer work than the gradient method itself.

Also similarly with the steepest descent method, the BB method will find the step size value, and use it to compute the value of x. But what is special about the BB method is it can find the step size value without the help of the Armijo line search strategy and the Lipschitz inequality. It is because the BB methods are already including the function to find the step size on its own. There are 2 different formulas for the BB method, and for the sake of simplicity, we will call them BB1 and BB2. The BB1 is calculated as:

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$$

And BB2 will be calculated as:

$$\gamma_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}$$

where

$$s_{k-1} = x_k - x_{k-1}$$

$$y_{k-1} = g_k - g_{k-1}$$

The BB method is also used to find the step size differently from the steepest descent. The calculation of the step size in BB method is:

$$d_k = -\gamma_k g_k$$

The steps to implement the conjugate gradient method are:

Step 1
- Find the alpha value with Armijo line search OR
- Find the alpha value (constant) with Lipschitz inequality OR
- FInd the alpha value using BB1 / BB2

Step 2
- $x_1 = x_0 + \alpha_0 d_0$

Step 3
- $g_1 = g(x_1)$

Step 4
- While norm of $g_1 \geq$ tolerance value AND the maximum loop haven't been reached, $d_1 = -\gamma_1 g_1$ and repeat step 1-4, else return x

3.3 Summary

It is believed that the three optimization algorithms proposed above will recover the blurred image with high similarity to the original image, although they are not able to recover the blurred image with 100% similarity to the original image (meaning the recovered image will have a bit different than the original image). The goal is just to make the blurred image clearer to human perception with the best possible algorithms.

Throughout the process of proposing the optimisation algorithm, a lot of research is done on various mathematical models that represent the deblurring problem, and from the research, the objective to understand the challenges to deblur image on each of the optimisation algorithm is achieved. In the later section, the effectiveness of each proposed method in image deblurring will be compared and the recommendation for improving the optimisation methods will be provided.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

To solve the optimization problem in image processing, in this final year project, the steepest descent method, conjugate gradient method, and the Barzilai-Borwein gradient method is used with the combination of Armijo rule and Lipschitz inequality, and the recovered image of each scenario is being compared to see which method and combination is better at recovering images. The OpenCV's default method filter2D is also used in this project to see if it can recover the image better than the proposed method.

Each of the recovered image has been calculated with peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). PSNR is a measure of the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is calculated by comparing the original image with the blurred image and measuring the peak error (the maximum difference between the pixel values of the two images) between them. On the other hand, SSIM measures the quality of a blurred image by comparing it with the original image and evaluating the structural similarity between them. Unlike PSNR, SSIM takes into account the structural information of the images, such as the contrast, luminance, and texture. The higher the PSNR the better the result, and the closer the SSIM to 1 the better the result.

4.2 Required libraries in Python

To use the required mathematical functions, we must import cv2, math, NumPy, and linalg from both scipy and NumPy.

4.3 Results

Table 4.1 shows the result of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) of different algorithms, from the best to the worst.

Table 4.1: The results of the algorithms

Algorithm to deblur image	**PSNR**	**SSIM**
Conjugate Gradient with Armijo	94.76158623	0.999999439
Barzilai and Borwein Gradient Method 2 with Armijo	93.59736804	0.999999208
Barzilai and Borwein Gradient Method 2 without Lipschitz or Armijo	93.59736804	0.999999208
Conjugate Gradient with Lipschitz	93.10939909	0.999999087
Barzilai and Borwein Gradient Method 1 with Armijo	92.0153788	0.999998565
Barzilai and Borwein Gradient Method 1 without Lipschitz or Armijo	92.0153788	0.999998565
Barzilai and Borwein Gradient Method 2 with Lipschitz	88.7204286	0.999995556
Barzilai and Borwein Gradient Method 1 with Lipschitz	88.15029473	0.999994752
Steepest Descent with Lipschitz	81.97649634	0.999978227
OpenCv-Python filter2D() function	67.53823251	0.998949032
Steepest Descent with Armijo	65.51771248	0.997308954

Figure 4.1 and Figure 4.2 shows the best result and the worst result respectively among the recovered images.

Figure 4.1: Image recovered using conjugate gradient method with Armijo rule.

Figure 4.2: Image recovered using steepest descent method with Armijo rule.

4.4 Discussion

From the test above, we can find out that the conjugate gradient method using Armijo rule to find the step size is the best method to recover image among all optimisation algorithms, with the PSNR value of 94.76158623 and the SSIM value of 0.999999439. The steepest descent method with Armijo rule not only is the worst method to recover the image (with the PSNR value of 65.51771248 and the SSIM value of 0.997308954), it also has the longest executing time among other optimisation algorithms. From the worst result, we can see from our own eyes that the image is obviously different than other recovered image, as the image is darker and the stripes in the image is more obvious.

We can also found out that using the Armijo rule to find the step size will always have much longer execute time than using the Lipschitz inequality, because the Armijo function will keep looping until it satisfy the Armijo condition or reach the maximum loop only it will return the step size, while the Lipschitz inequality will calculate the step size and use the returned value constantly throughout the whole execution.

On the other hand, the OpenCV's filter2D method cannot deblur the image properly as it can only sharpen the image. From table 4.1, we found out that the PSNR and SSIM result of the BB method with Armijo rule and BB method without Armijo rule or Lipschitz inequality are the same, but the one with Armijo rule has longer executing time. Although the filter2D method cannot deblur the image properly, but the PSNR and SSIM result of it still better than steepest descent with Armijo rule, which the recovered image looks quite similar to the original image. It is believe that because the recovered image of the steepest descent with Armijo rule has darker pixel intensity, or on other words, the darkness of the image is darker than the original image, which makes the system thinks that the recover image is much different than the original image, and thus the PSNR and the SSIM showed is lower than filter2D. Other than the 2 worst result, the PSNR and the SSIM result of other optimisation algorithms is pretty close to the original image, with the PSNR value of above 80 and and SSIM value of above 0.9999.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, it is better to use the conjugate gradient method with the Armijo rule to recover the blurred image, and the use of steepest descent method with Armijo rule to recover image should be avoided, as it doesn't give a good result and yet the execution time is higher. It is also better to avoid using Armijo rule to find the step size for the sake of time efficiency, with the exception of conjugate gradient method, because using the Armijo function really take up a lot of time and the difference of the recovered image isn't that big for the human eyes.

From this final year project, I have learned a lot in the field of image processing, and also the mathematical functions and formula that I never learn in my academic journey. I have also get more familiar with python as the whole project is focus on using python to write the codes to recover the images. I believe the knowledge I learned from this project can help me in my future work.

5.2 Recommendations for future work

In the future, the possibilities of the proposed optimisation algorithms for further image processing applications can be investigated. The proposed optimisation algorithms may be applicable to various image processing tasks like image segmentation or feature detection, even though this project used image deblurring as a case study. Future research could look into how well the proposed algorithms perform on these other tasks.

Besides, the generalization of the proposed algorithms to different image types can be further investigated. This project focused on grayscale image with Toeplitz matrix dragging effect as the blur factor. Future research could examine the performance of the proposed optimisation methods for various types of blurs or other kinds of images, such as colour image or medical image.

Other than that, the proposed algorithms can be evaluated on larger datasets. While the proposed algorithms showed promising results for image deblurring this study, the images used are just the same image which is the Lena image. Future work could evaluate the effectiveness of the proposed algorithms on more different images to further demonstrate its practical utility.

REFERENCES

Abubakar, A.B., Kumam, P., Malik, M. and Ibrahim, A.H., 2021. A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems. *Mathematics and Computers in Simulation*, 201, pp.640-657.

Berahas, A.S., Byrd, R.H. and Nocedal, J., 2019. Derivative-free optimization of noisy functions via quasi-Newton methods. *SIAM Journal on Optimization*, *29*(2), pp.965-993.

Burdakov, O., Dai, Y.H. and Huang, N., 2019. Stabilized Barzilai-Borwein method. *arXiv preprint arXiv:1907.06409*.

Cericola, V. (2015). *Quasi-Newton methods*. [online] optimization.mccormick.northwestern.edu. Available at: https://optimization.mccormick.northwestern.edu/index.php/Quasi-Newton_methods#:~:text=Quasi%2DNewton%20Methods%20(QNMs),)%20that%20is%20twice%2Ddifferentiable. [Accessed 30 Aug. 2022].

Chan, K.Y., Rajakaruna, N., Rathnayake, C. and Murray, I., 2014. Image deblurring using a hybrid optimization algorithm. *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp.1243-1249. doi:10.1109/cec.2014.6900266.

Ding, K., Ma, K., Wang, S., & Simoncelli, E. P., 2021. Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems. *International Journal of Computer Vision*, *129*(4), pp.1258–1281. https://doi.org/10.1007/s11263-020-01419-7

Ebadi, M.J., Hosseini, A. and Hosseini, M.M., 2017. A projection type steepest descent neural network for solving a class of nonsmooth optimization problems. *Neurocomputing*, *235*, pp.164-181.

Hansen, P.C., Nagy J.G., O'Leary D.P., 2006. The Image Deblurring Problem. *Deblurring Images*, pp.1-12. https://doi.org/10.1137/1.9780898718874.ch1

Ikarashi, Y., Ragan-Kelley, J., Fukusato, T., Kato, J. and Igarashi, T., 2021, October. Guided Optimization for Image Processing Pipelines. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 1-5). IEEE.

Khongkraphan, K., Phonon, A., & Nuiphom, S., 2021. An Efficient Blind Image Deblurring Using a Smoothing Function. *Applied Computational Intelligence and Soft Computing*, *2021*, pp.1–10. https://doi.org/10.1155/2021/6684345

Sada, M.M. and Mahesh, M.G., 2018. Image deblurring techniques—a detail review. *Int. J. Sci. Res. Sci. Eng. Technol*, 4(2), p.15.

Sato, H., 2021, December. Simple Acceleration of the Riemannian Steepest Descent Method with the Armijo Condition. In *2021 60th IEEE Conference on Decision and Control (CDC)* (pp. 3836-3843). IEEE.

Sun, W. and Yuan, Y.X., 2006. *Optimization theory and methods: nonlinear programming* (Vol. 1). Springer Science & Business Media. pp.126–128.

Uzila, A. (2021). *Complete Step-by-step Conjugate Gradient Algorithm from Scratch.* [online] towardsdatascience.com. Available at: https://towardsdatascience.com/complete-step-by-step-conjugate-gradient-algorithm-from-scratch-202c07fb52a8 [Accessed 30 Aug. 2022].

Wang, W., Wang, J., Zhang, Z. and Shi, D., 2022, April. OpenCV Implementation of Image Processing Optimization Architecture of Deep Learning Algorithm based on Big Data Processing Technology. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 65-68). IEEE.

Zhang, X., Lv, Y., Li, Y., Liu, Y. and Luo, P., 2019, July. A modified image processing method for deblurring based on GAN networks. In *2019 5th International Conference on Big Data and Information Analytics (BigDIA)* (pp. 29-34). IEEE.

APPENDICES

Appendix A: Lists of Recovered Images

Steepest Descent with Armijo

Steepest descent with Lipschitz

Conjugate gradient with Armijo

Conjugate gradient with Lipschitz

BB1 with Lipschitz

BB2 with Lipschitz

BB1 without Armijo and Lipschitz

BB2 without Armijo and Lipschitz

BB1 with Armijo

BB2 with Armijo

OpenCV filter2D() function