ĐẠI HỌC THỦ DẦU MỘT
2009 THU DAU MOT UNIVERSITY

Thu Dau Mot University
Journal of Science
ISSN 2615 - 9635
journal homepage: ejs.tdmu.edu.vn

Application of fast Fourier transform method to ill-posed problem in image processing

by ***Nguyen Huu Can*** *(Faculty of Mathematics and Computer Science, University of Science, Ho Chi Minh City, Vietnam; Vietnam National University, Ho Chi Minh City, Vietnam; Applied Analysis Research Group, Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam),*
Nguyen Van Tien *(Faculty of Maths, FPT University HCM, Saigon Hi-tech Park, Thu Duc City, Ho Chi Minh City, Vietnam),*
Mai Quang Vinh *(Division of Applied Mathematics, Thu Dau Mot University, Binh Duong Province, Vietnam)*
Article Info: *Received Feb. 22nd, 2023, Accepted May 15th, 2023, Available online June 15th, 2023*

Corresponding author: nguyenhuucan@tdtu.edu.vn
https://doi.org/10.37550/tdmu.EJS/2023.05.418

ABSTRACT
In this work, we focus on investigating ill-posed problems (according to the definition given by Hadamard) in the topic of its application. Specifically, we present some theories about the properties of the ill-posed problem in image processing. By using discrete Fourier transform and fast Fourier transform methods, we present several results on image processing topic. Finally, some illustrative examples are presented through algorithms running on Python software.
Keywords: *Ill-Posedness; FFT method; Image processing.*

1 Introduction

Recently, the application of mathematical models to solve practical problems is being attracted by many researchers. One of those models is the ill-posed problem in image processing. First, to understand what an ill-posed problem is, we recall the concept of the well-posed problem, introduced by a mathematician named J.S. Hadamard (a prominent French mathematician who lived from 1865 to 1963 and made significant contributions to various fields of mathematics, such as partial differential equations, number theory, differential geometry and complex analysis). He raised the question about ill-posed problems in his book [11]. According to the definition of ill-posed problems is given by Hadamard, which are problems that do not satisfy the conditions of well-posedness, which are existence, uniqueness and stability of the solution. Ill-posed problems are a common occurrence in image processing. These problems arise when the input data is insufficient or noisy, leading to an infinite number of possible solutions. Most of inverse problems are ill-posed, are prevalent in many applications such as medical imaging, astronomy, seismic imaging, nondestructive testing, and signal processing [4,13]. Ill-posed problems in computer vision and imaging are described by linear equations and emphasize in various applied engineering and physics areas such as plasma physics, nuclear physics, geophysics and radiophysics [14].

Regularization is a technique used to solve ill-posed problems in image processing. It involves adding constraints to the problem to reduce the number of possible solutions. Regularization can be achieved by adding a penalty term to the objective function, which helps to control the smoothness of the solution. It can help reduce noise and enhance the quality of images by imposing some constraints or penalties on the image processing model. Regularization techniques can also prevent over fitting, which occurs when the model learns too much from the training data and fails to generalize well to new data. In this topic of regularization, many authors have been concerned about the regularization of the ill-posed problem such as Tuan, Thach, Can [17–19].

Some examples of regularization techniques for image processing are used as follows

- Data augmentation, which creates new images from existing ones by applying transformations such as rotation, scaling, cropping, flipping, etc. This increases the diversity and size of the training data and helps the model learn more robust features [6].

- L_1 and L_2 regularization, which add a term to the loss function that penalizes large weights or coefficients in the model. This reduces the complexity and variance of the model and makes it less sensitive to noise [6, 8].

- Total variation denoising, which minimizes a functional that consists of a data fidelity term and a total variation term. The data fidelity term measures how well the denoised image matches the noisy image, while the total variation term measures how smooth or piecewise constant the denoised image. This technique preserves edges and removes noise in images [8].

- Tikhonov regularization, this is a method used to solve ill-posed problems in image processing. Ill-posed problems are those that do not have a unique solution or whose solution is sensitive to small changes in the input data. Tikhonov regularization is a technique that adds a regularization term to the objective function to control the effect of noise on the solution [15].

There is no general technique for dealing with ill-posed problems. Each situation has to be handled differently depending on the main issue – instability. In this work, we focus on the fast Fourier transform (FFT) method to ill-posed problem in image processing. More specifically, we will apply the FFT algorithm to recover images that are noisy by the Gaussian model.

So, why we choose the FFT method to solve the ill-posed problems in image processing? To answer this question, we reiterate the Fourier transform (FT) method, it is first introduced by Jean Baptiste Joseph Fourier (21 March 1768 - 16 May 1830, was a French mathematician and physicist). As the limiting case of the Fourier series for non-periodic signals, FT is used to transfer the signal to the frequency domain since it offers numerous superior advantages over the classical time domain, particularly for analytical applications. The discrete version of FT, also known as discrete Fourier transform (DFT), has been developed to address a variety of difficulties, particularly those related to digital image processing. Although the advantage of DFT is that it is very easy to program and therefore easy to implement in any coding language, the disadvantage is that it requires a lot of computation time. This increasingly needs to improve in modern computational science. Then FFT method has been introduced to significantly reduce the computational complexity. Particularly when the signals are two dimensional, like picture signals, FFT is a helpful signal encoding technique that may be used for quick processing.

The main purpose of this study is to investigate the following linear observations in additive Gaussian noise model

$$f = Ag + w \tag{1.1}$$

where $f \in \mathbb{R}^n$ is original data (image), $g \in \mathbb{R}^m$ is observed data (image), $A \in \mathbb{R}^{n \times m}$ is a linear operator, $w \in \mathcal{N}(0, \sigma)$ is noise Gaussian smoothing model.

Goal: Recovering data g from original data f (which is given) through the action of operators A and f noisy by the Gaussian model w. In general, ill-posed inverse problem (finding g when f is known) is harder to solve than the direct challenge of finding f when g is known.

2 Preliminaries

In this section, we recall some theories related to the research direction of this work.

2.1 Signal deblurring

In image processing, the deconvolution (or deblurring) problem of recovering an input signal g in time t from an observed signal f is given by

$$f(t) = \int_{\mathbb{R}} a(t - z)g(z)dz + w(t) \tag{2.1}$$

where the function a is the blurring kernel in the reality model such as blur by time, tomography, MRI, etc. When the observed data is not noisy (i.e. $w = 0$) then we have

$$f(t) = \int_{\mathbb{R}} a(t - s)g(z)dz \tag{2.2}$$

using the Fourier transform of (2.2) as follows

$$\widehat{f}(\xi) = \int_{\mathbb{R}} \exp(-i\xi t)f(t)dt. \tag{2.3}$$

Applying the convolution theorem, we have

$$\widehat{f}(\xi) = \widehat{a}(\xi)\widehat{g}(\xi).$$

By using the formula of inverse Fourier transform, we obtain

$$g(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \exp(it\xi) \frac{\widehat{f}(\xi)}{\widehat{a}(\xi)} d\xi.$$

On the other hand, we have $\widehat{f}(\xi) = \widehat{a}(\xi)\widehat{g}(\xi) + \widehat{w}(\xi)$. Applying the convolution theorem again, we have the estimated observe data g_{obs} as follows

$$g_{\text{obs}}(t) = g(t) + \frac{1}{2\pi} \int_{\mathbb{R}} \exp(it\xi) \frac{\widehat{w}(\xi)}{\widehat{a}(\xi)} d\xi, \tag{2.4}$$

2.2 The discrete Fourier transform (DFT)

Let the source data $f(t)$ is the continuous signal and the samples N is given by

$$f[0], f[1], f[2], \ldots, f[k], \ldots, f[N-1]$$

where the DFT has the same meaning as the continuous Fourier transform for signals known only at N distinct moments in time by sample times T (in more detail, this is a finite sequence of input data). Then we have the original signal $f(t)$ is affected by the Fourier transform as follows

$$\widehat{f}(\xi) = \int_{\mathbb{R}} \exp(-i\xi t) f(t) dt.$$

An each sample might be considered $f[k]$ as an impulse with the area $f[k]$. Then, at the sample points, the following integral exists

$$\begin{aligned}\widehat{f}(\xi) &= \int_0^{(N-1)T} \exp(-i\xi t) f(t) dt \\ &= \exp(-i0) f[0] + \exp(-i\xi T) f[1] + \ldots + \exp(-i\xi(N-1)T) f(N-1)\end{aligned}$$

In general, we have

$$\widehat{f}[n] = \sum_{k=0}^{N-1} \exp\left(-i\frac{2\pi}{N} nk\right) f[k].$$

We present the equation above in matrix form as follows

$$\begin{bmatrix} \widehat{f}[0] \\ \widehat{f}[1] \\ \widehat{f}[2] \\ \vdots \\ \widehat{f}[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_N^0 & W_N^{N-1} & W_N^{2(n-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{bmatrix}$$

where $W = \exp(-i2\pi/N)$.

Remark 2.1. *If the waveform was periodic, the continuous Fourier transform may be calculated across a defined range (often the fundamental period T_0) as opposed to from $-\infty$ to $+\infty$. The DFT similarly considers the data as if it were periodic since there are only a limited number of input data points (i.e., $f(N)$ to $f(2N-1)$ are equivalent from $f(0)$ to $f(N-1)$).*

In addition, the inverse discrete Fourier transform of

$$\widehat{f}[n] = \sum_{k=0}^{N-1} \exp\left(-i\frac{2\pi}{N} nk\right) f[k]$$

is shown by

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} \exp\left(i\frac{2\pi}{N} nk\right) \widehat{f}[n].$$

2.3 Noise Gaussian smoothing model

Applying the Riesz theorem, we use a linear radial kernel to make image isotropic linear filtering boils down to a convolution of the image. The positivity of the kernel is often how the smoothing requirement is represented. Naturally, the Gaussian is the model for these kernels

$$\mathbf{y} \to w_{G_\alpha}(\mathbf{y}) := \frac{\exp\left(-\frac{|\mathbf{y}|^2}{4\alpha^2}\right)}{4\pi\alpha^2}.$$

As a result, w_{G_α} has standard deviation α and it is clear to observe that.

Theorem 2.2 (by Gabor 1960, see [16]). *The convolution with a Gaussian kernel's image method noise w_{G_α} is given by*

$$h - w_{G_\alpha} * h = -\alpha^2 \Delta h + o\left(\alpha^2\right).$$

As this conclusion holds true for any positive radial kernel with limited variance, the Gaussian example may be retained without losing its generality. If α is small enough, the preceding estimate will be accepted. On the other hand, the size of the neighborhood involved in the smoothing affects the noise reduction qualities, so that the noise gets reduced by averaging. So in the following we assume that $\alpha = p\varepsilon$, where p stands for the number of samples of the function u and noise n in an interval of length α. To assure a decrease in noise, the spatial ratio k must be significantly greater than 1. The effect of a Gaussian smoothing on the noise can be estimated at a reference pixel $j = 0$. Considering this pixel, we have

$$\begin{aligned} w_{G_\alpha} * n(0) &= \sum_{j \in I} \int_{Q_j} n(\mathbf{x}) w_{G_\alpha}(\mathbf{y}) d\mathbf{y} \\ &= \sum_{j \in I} n_j w_{G_\alpha}(j) \varepsilon^2 \end{aligned}$$

where the Q_j square pixels centered in j have size ε^2, $n(\mathbf{y})$ is being interpolated as a piecewise constant function and $w_{G_\alpha}(j)$ denotes the mean value of the function w_{G_α} on the pixel j.

The additive of variances of independent centered random variables, where $\mathrm{Var}(\mathbf{y})$ denotes the variance of a random variable with value X as follows

$$\begin{aligned} \mathrm{Var}\left(w_{G_\alpha} * n(0)\right) &= \sum_i \varepsilon^4 w_{G_\alpha}(i)^2 \sigma^2 \\ &\simeq \sigma^2 \varepsilon^2 \int (w_{G_\alpha}(\mathbf{y}))^2 d\mathbf{y} = \frac{\varepsilon^2 \sigma^2}{8\pi\alpha^2}. \end{aligned}$$

Remark 2.3. *Let $n(\mathbf{y})$ be a piecewise constant white noise, with $n(\mathbf{y}) = y_j$ on each square pixel j. Assume that the n_j are independent and identically distributed (i.i.d.) with zero mean and variance σ^2. After a Gaussian convolution of n by w_{G_α}, the "noise residue" obtains*

$$\mathrm{Var}\left(w_{G_\alpha} * n(0)\right) \simeq \frac{\varepsilon^2 \sigma^2}{8\pi\alpha^2}.$$

In other words, the noise's standard deviation, which may be seen as the noise amplitude, is increased by $\frac{\varepsilon}{\alpha\sqrt{8\pi}}$.

2.4 Algorithm and tools

Firstly, we consider the spectral theory $Av_k = \lambda_k v_k$. Then the eigenvectors of a circulation matrix as follows

$$V = [v_0 \quad v_1 \quad \dots \quad v_{N-1}] \tag{2.5}$$

where

$$v_k = \frac{1}{\sqrt{N}} \left[e^{i\frac{2\pi}{N}kn} \text{ for } n = 0, \dots, N-1 \right]^T.$$

And the eigenvalues are given by

$$L = \operatorname{diag}(\lambda_0, \dots, \lambda_{N-1})$$

where $\lambda_k = \sum_{n=0}^{N-1} h[n] e^{-j\frac{2\pi}{N}kn} = \sqrt{N}\,(v_k^* h)$ is the DFT of $h[n]$ at frequency $\frac{2\pi}{N}k$.

Secondly, the illustration is supported by Python software (version 3.10) on a laptop run Windows 10 (64 bit), i7-11370H, 16 GB RAM, GPU NVIDIA RTX 3050Ti. To count the running the CPU times, we use the Python calculate runtime in library `import time` with the code

`start_time = time.clock()`,

and `print(time.clock() - start_time, "seconds")`. Some libraries are used in Python software which are

`scipy`, `scipy.fftpack`, `scipy.ndimage`, `numpy`, `matplotlib`, `matplotlib.colors` and `LogNorm`. The algorithm image processing in Python is shown in Algorithm 1.

3 Illustration

In this section, we present some example to check the theory part presented in the previous section. First example, we use the input data is an image of the logo of Thu Dau Mot University (TDMU) in Figures 1, the results are shown in Figures 3 and 3. Second example, a photo of a campus at TDMU is used for this example in Figures 2 (source of pictures: `https://tdmu.edu.vn`). To convert these pictures to data in Python software, we divide it into size 400x256 pixels for each picture.

Acknowledgements. This work is supported by the conference "Mathematics and Applications" at Thu Dau Mot University.

References

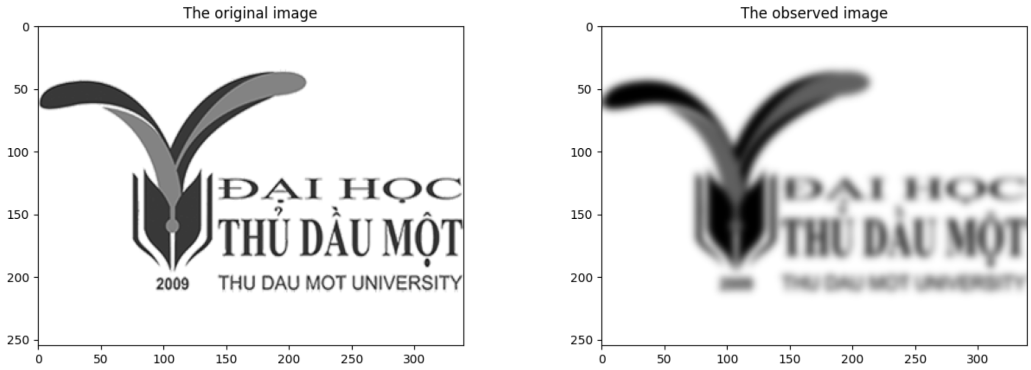
[1] Stephanie Glen. *Well Posed and Ill Posed problems & Tikhonov Regularization*, https://www.statisticshowto.com/well-posed-ill

[2] Kabanikhin, Sergey I. *Inverse and ill-posed problems.* In Inverse and Ill-posed Problems. de Gruyter, 2011.

[3] Kabanikhin, Sergei Igorevich. *Definitions and examples of inverse and ill-posed problems.* (2008): 317-357.

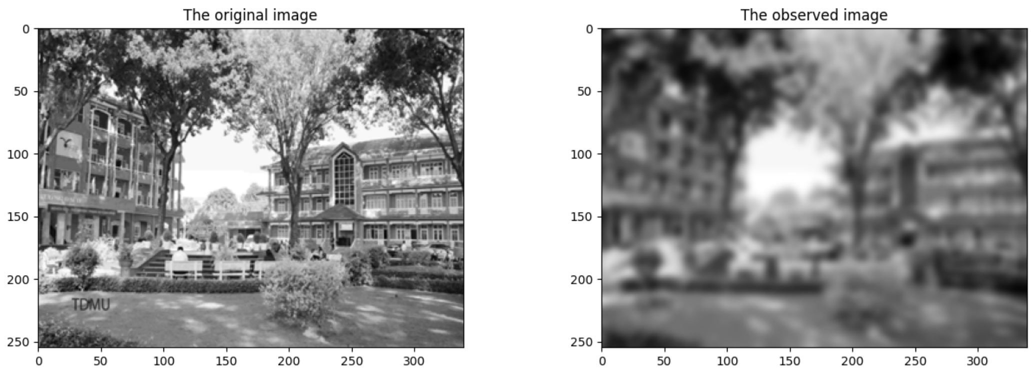
Algorithm 1 Algorithm image processing in Python

```
### The input data in format
im = plt.imread('The position of input image.tif').astype(float)
matplotlib.pyplot.plt.figure()
matplotlib.pyplot.plt.imshow(im, plt.cm.gray)
### Calculating the FFT of input data
im_fft = fftpack.fft2(im)
### Gaussian model
im_blur = ndimage.gaussian_filter(im, 3)
matplotlib.pyplot.plt.figure()
matplotlib.pyplot.plt.imshow(im_blur, plt.cm.gray)
### The spectral theory
def plot_spectrum(im_fft):
    from matplotlib.colors import LogNorm
    ### Make a logarithmic colormap
    matplotlib.pyplot.plt.imshow(np.abs(im_fft), norm=LogNorm(vmin=5))
    matplotlib.pyplot.plt.colorbar()
matplotlib.pyplot.plt.figure()
matplotlib.pyplot.plot_spectrum(im_fft)
### Definition of the fraction of coefficients
keep_fraction = 0.25
im_fft2 = im_fft.copy()
### Set c and r are the number of columns and rows
r,c = im_fft2.shape
im_fft2[:,int(c*keep_fraction):int(c*(1-keep_fraction))] = 0
im_fft2[int(r*keep_fraction):int(r*(1-keep_fraction))] = 0
### Recovering denoised image from filtered spectrum
matplotlib.pyplot.plt.figure()
matplotlib.pyplot.plot_spectrum(im_fft2)
```



(a) The input data f (b) f_G is noise by Gaussian model

Figure 1: The original and observed images of TDMU's logo



(a) The input data f (b) f_G is noise by Gaussian model

Figure 2: The original and observed images of a picture in campus at TDMU

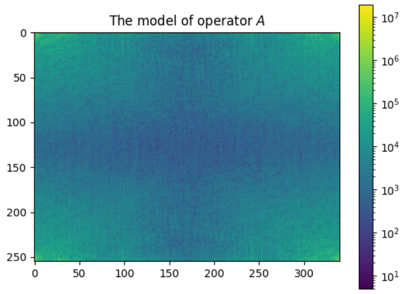
[4] Bakushinsky, Anatoly, and A. Goncharsky. *Ill-posed problems: theory and applications*. Vol. *301*. Springer Science & Business Media, 2012.

[5] Yagola, Anatoly. *Introduction to the theory, numerical methods and applications of ill-posed problems*. Lecture Notes, Chalmers University of Technology, Sweden (2013).

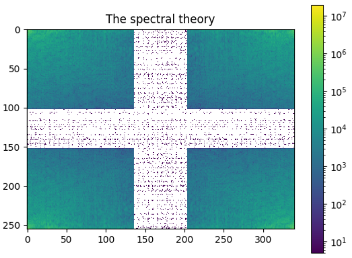
[6] Santos, Claudio Filipi Gonçalves Dos, and Joao Paulo Papa. *Avoiding overfitting: A survey on regularization methods for convolutional neural networks*. ACM Computing Surveys (CSUR) *54* (10s) (2022): 1-25.

[7] Chauhan, Kalpana, and Rajeev Kumar Chauhan. *An improved regularization and fitting-based segmentation method for echocardiographic images*. In Image Processing for Automated Diagnosis of Cardiac Diseases, pp. 157-172. Academic Press, 2021.

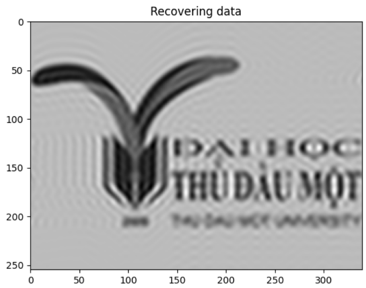
[8] Chambolle, Antonin. *An algorithm for total variation minimization and applications*. Journal of Mathematical imaging and vision, *20* (2004): 89-97.



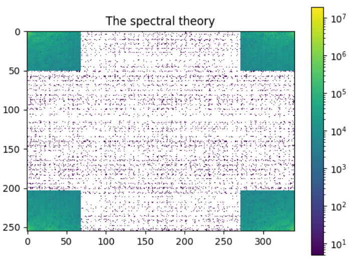
(a) The spectrum of the operator A



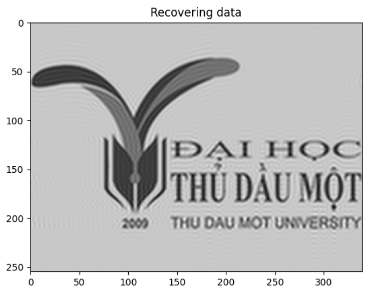
(b) The spectrum of FFT for $\alpha = 10^{-1}$



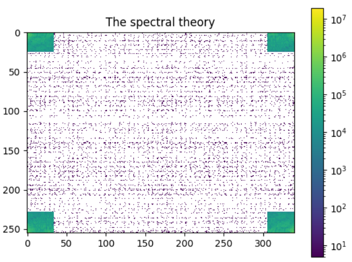
(c) Regularized data $\widehat{g}_\alpha$ for $\alpha = 10^{-1}$, CPU time: 8.428 seconds



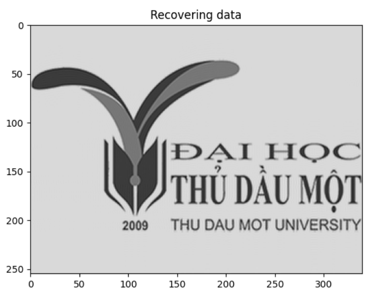
(d) The spectrum of FFT for $\alpha = 2 * 10^{-1}$



(e) Regularized data $\widehat{g}_\alpha$ for $\alpha = 2 * 10^{-1}$, , CPU time: 9.023 seconds

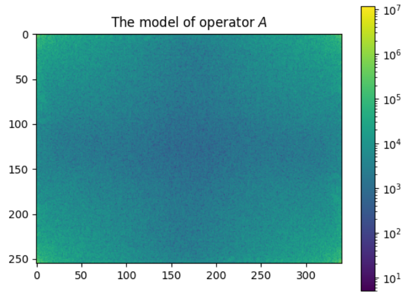


(f) The spectrum of FFT $\alpha = 4 * 10^{-1}$

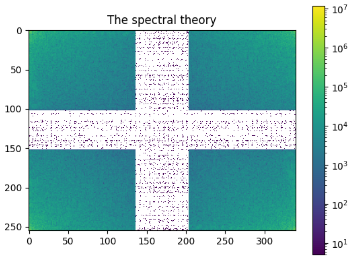


(g) Regularized data $\widehat{g}_\alpha$ for $\alpha = 4 * 10^{-1}$, CPU time: 11.642 seconds

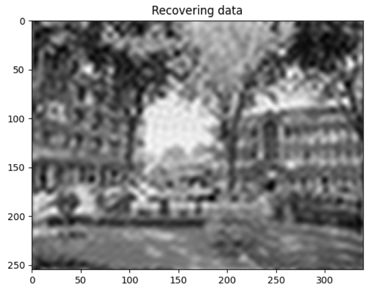
Figure 3: Recovering image of TDMU's logo



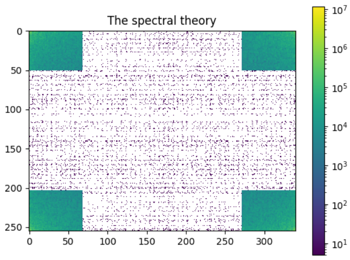
(a) The spectral theory of the operator A



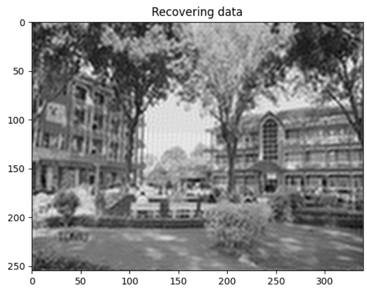
(b) The spectral theory for $\alpha = 10^{-1}$



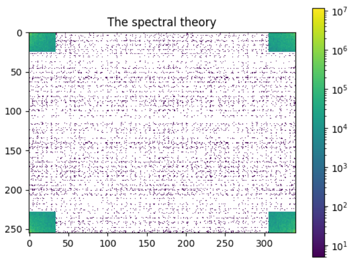
(c) Regularized data $\widehat{g}_\alpha$ for $\alpha = 10^{-1}$, CPU time: 9.452 seconds



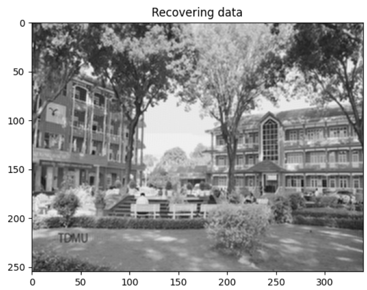
(d) The spectral theory for $\alpha = 2 * 10^{-1}$



(e) Regularized data $\widehat{g}_\alpha$ for $\alpha = 2 * 10^{-1}$, CPU time: 8.421 seconds



(f) The spectral theory for $\alpha = 4 * 10^{-1}$



(g) Regularized data $\widehat{g}_\alpha$ for $\alpha = 4 * 10^{-1}$, CPU time: 13.532 seconds

Figure 4: Recovering image for a picture of campus at TDMU

[9] Hansen, Per Christian. *Regularization tools version 4.0 for Matlab 7.3.* Numerical algorithms *46* (2007): 189-194.

[10] Bracewell, R. N., & Bracewell, R. N. (1986). *The Fourier transform and its applications* (Vol. *31999*, pp. 267-272). New York: McGraw-Hill.

[11] Hadamard, J. (1964). *La théorie des équations aux dérivées partielles.* Éditions scientifiques.

[12] Hadamard, J. (1902). *Sur les problèmes aux dérivées partielles et leur signification physique*. Princeton university bulletin, 49-52.

[13] Alaifari, R. (2021). *Ill-Posed Problems: From Linear to Nonlinear and Beyond*. Harmonic and Applied Analysis: From Radon Transforms to Machine Learning, 101-148.

[14] Abidi, M. A., Gribok, A. V., Paik, J., Abidi, M. A., Gribok, A. V., & Paik, J. (2016). *Ill-Posed Problems in Imaging and Computer Vision*. Optimization Techniques in Computer Vision: Ill-Posed Problems and Regularization, 3-27.

[15] Bouhamidi, A., & Jbilou, K. (2007). *Sylvester Tikhonov-regularization methods in image restoration*. Journal of Computational and Applied Mathematics, *206*(1), 86-98.

[16] Authors: A. Buades, B. Coll, & J. M. Morel (2005). *A Review of Image Denoising Algorithms, with a New One*. SIAM Journal on Multiscale Modeling and Simulation, *4*(2), 460-489.

[17] Tuan, N. H., Hai, N. M., Thach, T. N., & Can, N. H. (2022). *On stochastic elliptic equations driven by Wiener process with non-local condition*. Discrete and Continuous Dynamical Systems-S, 0-0.

[18] Can, N. H., Zhou, Y., Tuan, N. H., & Thach, T. N. (2020). *Regularized solution approximation of a fractional pseudo-parabolic problem with a nonlinear source term and random data*. Chaos, Solitons & Fractals, 136, 109847.

[19] Tuan, N. H., Baleanu, D., Thach, T. N., O'Regan, D., & Can, N. H. (2020). *Final value problem for nonlinear time fractional reaction–diffusion equation with discrete data*. Journal of Computational and Applied Mathematics, 376, 112883.