

MIXED GRAPH SIGNAL ANALYSIS OF JOINT IMAGE DENOISING AND INTERPOLATION

NIRUHAN VISWARUPAN

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

April 2025

©Niruhan Viswarupan, 2025

Abstract

An image signal captured by a photonic sensor is corrupted by noise and other distortions, and requires several operations before rendering a presentable photo, including denoising, demosaicking, rectification, and white balancing. Specifically, denoising reduces acquisition noise without destroying image details, while interpolation generates novel pixels at new 2D locations using acquired image data.

Such image operations are typically applied in sequence—e.g., denoising an image first followed by interpolation—which is sub-optimal in general.

In this thesis, we jointly optimize denoising and interpolation operations from a graph signal processing (GSP) perspective. Specifically, we first develop two theorems that show one-to-one mappings between (pseudo-)linear denoisers / interpolators and undirected / directed graph filters that are solutions to variational optimization problems regularized using different graph smoothness priors. Leveraging these theorems, we investigate scenarios where joint denoising / interpolation operations would outperform separate operations in sequence, and mathematically derive those optimal joint operators. Experiments demonstrate validation results, where optimized joint operators outperformed separate operators in sequence in various practical imaging scenarios.

Thesis Title

The title of the thesis is **Mixed Graph Signal Analysis of Joint Image Denoising and Interpolation**

This is a manuscript thesis consisting of two papers one of which is published and the other in progress.

Supervisory Team

Supervisor Name:

Dr. Gene Cheung

Committee Member Name:

Dr. Michael Brown

Contents

Abstract	ii
List of Figures	vi
1 Introduction	1
2 Related Works	4
2.1 Image Denoising	4
2.1.1 Model-based Local Filters	4
2.1.2 Model-based Non-Local Filters	6
2.1.3 Deep Learning-based Denoisers	7
2.1.4 Graph-based Filters based on GSP	8
2.2 Image Interpolation	9
2.2.1 Demosaicking	10
2.3 Joint Image Denoising / Interpolation	13
2.4 Graph Spectral Image Processing	15
3 Preliminaries	17
3.1 Graph Signal Processing	17
3.1.1 Graph Smoothness Priors	18
4 Mappings between Linear Denoiser / Interpolator and Graph Filters	20
4.1 Mapping between Denoiser and Undirected Graph Filter	20
4.2 Mapping between Interpolator and Directed Graph Filter	22
4.3 Joint Denoising and Interpolation	24
4.3.1 Joint Formulation with Separable Solution	24
4.3.2 Joint Formulation with Non-separable Solution (One Denoiser)	26
4.3.3 Joint Formulation with Non-separable Solution (Two Denoisers)	27
4.4 Generalization to Multiple Interpolators	29
4.4.1 Computation of Joint Denoising / Multi-Interpolation	31
4.4.2 Application of Theorem 3 for Rectangular Θ	31

5	Experiments and Results	33
5.1	Experimental Setup	33
5.2	Experimental Results	36
5.2.1	Experiments for Corollary 2	36
5.2.2	Experiments for Corollary 3	39
5.2.3	Experiments for Theorem 3	43
6	Conclusion and Future Direction	44
	References	45

List of Figures

1	Bayer pattern [1] used in camera sensors with its blue, green, and red components	3
2	Noisy image denoised using total variation (TV) minimization. Denoised image exhibits staircase effect along gradient changes [2].	7
3	Illustrations of mixed graphs for joint interpolation / denoising. Pixels 5,6,7,8 are interpolated using pixels 1,2,3,4. Directed edges for interpolation are shown as red dashed arrows, while undirected edges for denoising are shown as green dashed lines. (a) corresponds to equation 41 where input pixels are denoised before interpolation. (b) corresponds to equation 46 where interpolated pixels are denoised after interpolation.	20
4	Interpolation of blue channel from color filter array (CFA) grid. In the first step we are extracting the blue pixels from an 8×8 bayer grid. In the second step, the extracted pixels are rearranged into 4×4 patch. In the third step final 8×8 blue channel output is interpolated from the 4×4 blue patch. . .	35
5	Output of demosaicking with sequential and joint processes.	36
6	Performance comparison of joint vs sequential processing over a range of noise variances using image rotation and bilateral denoiser on output pixels.	37
7	Performance comparison of joint vs sequential processing over a range of noise variances using image rotation and Non-local Means denoiser on output pixels.	38
8	Performance comparison of joint vs sequential processing over a range of noise variances using image warping and Bilateral denoiser on output pixels.	38
9	Performance comparison of joint vs sequential processing over a range of noise variances using image warping and Gaussian denoiser on output pixels.	39
10	Performance comparison of joint vs sequential processing over a range of noise variances using image rotation and a bilateral denoiser.	41
11	Performance comparison of joint vs sequential processing over a range of noise variances using image rotation and a Gaussian denoiser.	41

12	Performance comparison of joint vs sequential processing over a range of noise variances using image warping and Gaussian denoiser.	42
13	Performance comparison of joint vs sequential processing over a range of noise variances using image warping and bilateral denoiser.	42
14	Performance comparison of joint vs sequential processing over a range of noise variances using demosaicking and a Gaussian denoiser.	43

1 Introduction

An image signal captured by a photonic sensor is often corrupted by noise and other distortions due to a variety of reasons, such as physical sensor defects, lens imperfections, environmental factors (*e.g.*, rain, fog, haze), insufficient or uneven lighting, and camera focus or motion-related effects. There exist statistical noise models aim to capture the resulting adverse effects due to different causes, including Gaussian noise [3], Poisson noise [4], impulse noise (*e.g.*, salt-and-pepper noise) [5], and quantization noise [6].

Given a noise-corrupted image as input, in addition to denoising, there exist many processing steps, such as demosaicking [7], rectification [8], contrast enhancement [9], white balancing [10], gamma correction [11], colour correction [12], before a presentable image can be rendered. In this thesis, we focus on two fundamental operations: denoising and interpolation. Examples of image interpolation abound. *Demosaicking* [7] is the process of interpolating missing colour pixels given only sparse colour observations in a Bayer-patterned 2D grid (with only one of red / green / blue component at each pixel location). Many image transformation tasks, such as rotating, warping and rescaling [8], can also be considered as interpolation. In stereo vision, rectifying a pair of images so that pixels in one row in the left image correspond to pixels of the same row in the right image is important; rectified stereo image pairs are used to estimate a disparity map, which in turn is used for 3D tasks such as 3D object reconstruction and recognition [13, 14].

The need for both image denoising and interpolation naturally leads to the question of how to jointly perform them optimally, so that the output image quality can be maximized. One straightforward approach is to do the two operations sequentially, *i.e.*, either denoise first and then interpolate, or interpolate and then denoise the image. Denoising an image before interpolation reduces noise, ensuring that interpolation operates on a cleaner image.

However, denoising is fundamentally a tradeoff between noise reduction and inadvertent removal of intrinsic image details, and thus an overly-aggressive denoiser would “oversmooth” an image, leaving little details for the interpolator to operate on. On the other hand, an inadequate denoiser would leave nontrivial noise behind in the image, and subsequent interpolator would then spread the noise further to newly interpolated pixels.

Image interpolation before denoising would enable interpolation of new pixels using original image details that would otherwise be damaged by an aggressive denoiser.

However, interpolation first also means spreading acquisition noise in observed pixels to newly interpolated pixels, creating correlated noise patterns that are more difficult to remove subsequently. It would also mean that the noise energy in the signal is amplified through the interpolator.

In this thesis, we investigate the optimal joint denoising / interpolation problem for images from a *graph signal processing* (GSP) perspective [15, 16]. GSP studies mathematical tools like transforms, wavelets, and filters to process discrete signals residing on irregular data kernels described by finite graphs. Assuming signals are smooth with respect to (w.r.t.) the underlying graph kernels, numerous image restoration algorithms have been developed for various applications, including image denoising, interpolation, dequantization, deblurring, contrast enhancement, point cloud denoising, super-resolution, etc [17–21]. Leveraging these works, for joint image denoising / interpolation, we first develop two new theorems that establish one-to-one mappings between a (pseudo-)linear denoiser / interpolator and a undirected / directed graph filter that is a solution to a variational optimization problem regularized by a chosen graph smoothness prior: *graph Laplacian regularizer* (GLR) [22] and *graph shift variation* (GSV) [23], respectively. These one-to-one mappings allow us to combine the denoising and interpolation operations in a unified variational objective for joint optimization. Analyzing these joint optimizations for different combinations of denoisers and interpolators, we identify scenarios where separate denoising / interpolation operations in sequence would be optimal, and scenarios where joint denoising / interpolation operations would outperform separate operations in sequence. For the latter scenarios, we mathematically derive the optimal joint denoising / interpolation operators as functions of initial denoising and interpolation operators. Experiments demonstrate validation results, where optimized joint operators outperformed separate operators in sequence in various practical imaging scenarios.

The outline of the thesis proposal is as follows. Section 2 discusses related works on topics such as image denoising, image interpolation, joint image denoising and interpolation, and graph spectral image processing in detail. Section 3 introduces fundamental concepts and

notations in GSP, laying the foundation for this proposal. In Section 4, we dive deeper into graph interpretations of denoiser and interpolator and our joint optimization formulation.

Penultimate Section 5 describes the experimentations carried out to test the validity of the proposed theorems and the results of the tests. Finally, Section 6 arrives at conclusions based on the theory and experiments and points to future research areas based on the learnings of this undertaking.

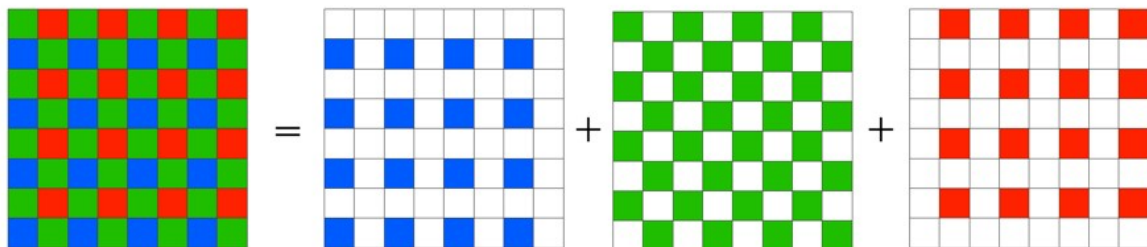


Figure 1: Bayer pattern [1] used in camera sensors with its blue, green, and red components

2 Related Works

We discuss existing works under two broad categories. The first is literature in image interpolation and denoising. The second is graph signal processing (GSP) and its application to imaging.

2.1 Image Denoising

Image denoising is a basic problem in image restoration that aims at removing additive noise from a noise-corrupted image to recover the original pristine image. Noise can be introduced during image acquisition, transmission, or processing; it degrades the visual quality and impairs the performance of subsequent image analysis tasks. Various denoising techniques have been developed over the years, each leveraging different model-based or data-driven signal priors.

2.1.1 Model-based Local Filters

Traditional image denoising methods include spatial domain filters such as mean [24] and median filters [25], which are simple but often lead to loss of image details and blurring. Gaussian filter is a very basic denoising method that employs a Gaussian function to smooth the image, but often blurs fine details as well. Gaussian filter is signal-independent and inherently linear, and is not sufficiently powerful for practical image denoising.

Bilateral filter (BF) [26] combines *domain* and *range* filters to preserve object boundaries while smoothing. Mathematically, BF is expressed as

$$I_f(p) = \frac{1}{W_p} \sum_{q \in \Omega} I(q) \exp\left(-\frac{\|p - q\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|I(p) - I(q)\|^2}{2\sigma_r^2}\right) \quad (1)$$

where $I_f(p)$ is the filtered intensity of pixel at coordinate p , $I(q)$ is the intensity of pixel at coordinate q within the neighborhood Ω , $W_p = \sum_{q \in \Omega} \exp\left(-\frac{\|p - q\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|I(p) - I(q)\|^2}{2\sigma_r^2}\right)$ is a normalization factor, and σ_s and σ_r are parameters for the domain and range filters, respectively.

Unlike Gaussian filter, BF is signal-dependent: the filtered intensity $I_f(p)$ is a function of the difference in intensity values $|I(p) - I(q)|$. This makes BF nonlinear or *pseudo-linear*.

However, when BF is applied iteratively, it can cause undesirable “staircase” effect in the resulting image—artificially created boundaries between adjacent constant pieces that do not exist in the original image.

In a variational optimization approach, a denoising filter can be derived from an optimization problem, with a well-defined fidelity term and a signal regularization term. As an example, restoration of a signal corrupted by *additive white Gaussian noise* (AWGN) can be written as

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu R(\mathbf{x}), \quad (2)$$

where $\|\mathbf{y} - \mathbf{x}\|_2^2$ is the fidelity term, and $R(\cdot)$ is a chosen prior for regularization. The parameter μ controls the trade-off between the fidelity term and the regularization term, also known as the *bias-variance tradeoff* in the machine learning literature [27].

Sparse Representation: *Sparse coding* is another variational approach to image denoising [28]. In this method, an image (or image patch) $\mathbf{x} \in \mathbb{R}^d$ is represented as a linear combination $\mathbf{D}\boldsymbol{\alpha}$ of atoms from an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{d \times n}$ such that $n > d$, and the code vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ is assumed to be sparse. The optimization problem can be formulated as follows:

$$\boldsymbol{\alpha}^* = \arg \min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (3)$$

The overcomplete dictionary $\mathbf{D}$ can be trained offline using data and learning methods such as K-SVD [29], or constructed from multiple known orthogonal transforms, such as the *discrete cosine transform* (DCT). Given that the ℓ_0 -norm is combinatorial and minimizing it is NP-hard, greedy algorithms such as matching pursuit or orthonormal matching pursuit [30] are employed to approximately solve (3).

Alternatively, promoting sparsity in $\boldsymbol{\alpha}$ can be done by convexifying the non-convex ℓ_0 -norm to a convex ℓ_1 -norm (thus relaxing an NP-hard problem to a convex problem). Then, it can be solved using basis pursuit [31], iterative shrinkage-thresholding algorithm (ISTA) [32], or fast iterative shrinkage-thresholding algorithm (FISTA) [33]. However, the ℓ_1 -norm is non-differentiable, necessitating iterative algorithms like ISTA and FISTA, which are computationally intensive.

Although sparse coding is a general and intuitive framework, it faces two main challenges:

(i) acquiring a suitable overcomplete dictionary essential for these methods can be demanding and complex, and (ii) solving the sparse optimization objective, as formulated in (3), poses considerable computational challenges.

Total Variation: A widely used prior in variational denoising is *total variation* (TV)-based regularization [2, 3], based on the principle that natural images are mostly smooth, with occasional large discontinuities. Mathematically, the (discrete) total variation $R_{TV}(\mathbf{x})$ is defined as:

$$R_{TV}(X) = \sum_{j \in \Omega_i} \|x_i - x_j\|_q, \quad (4)$$

where Ω_i represents the set of four horizontal and vertical neighboring pixels of x_i , and $\|\cdot\|_q$ denotes the ℓ_q -norm—typically $q = 1$. A concise notation for $R_{TV}(\mathbf{x})$ is $\|\nabla \mathbf{x}\|_q$, where $\nabla \mathbf{x}$ is the gradient of $\mathbf{x}$, calculated by the difference between consecutive samples. For an image, two pixels are consecutive if they are either horizontally or vertically adjacent. Hence, the dimensionality of $\|\nabla \mathbf{x}\|_q$ is $2N(N - 1)$. Generally, the ℓ_1 -norm is applied to promote sparsity in the optimal solution. In this scenario, R_{TV} can be solved using ISTA [34] or FISTA [33]. However, as with other sparse representations, solving TV is computationally intensive, mainly because of the non-differentiable ℓ_1 -norm term, which lacks closed-form solutions [35]. Further, the ℓ_1 -norm induces a sparse gradient that leads to piecewise constant signals, resulting in an undesirable staircase effect in denoised images (Figure 2).

To address this, *total generalized variation* (TGV) [36] was proposed as an extension of TV, which includes higher-order derivatives alongside the first derivative to mitigate the staircase effect. However, TGV still involves the non-differentiable ℓ_1 -norm, making it challenging to solve efficiently.

2.1.2 Model-based Non-Local Filters

Non-local means (NLM) [37] denoises an image by averaging pixels in a target patch with similar patches, preserving texture and fine details better than local methods. BM3D (Block-Matching and 3D Filtering) [38] is a highly effective image denoising algorithm that combines non-local means and collaborative filtering. The algorithm works by dividing the image into overlapping blocks, searching for similar blocks within a predefined window, and grouping



Figure 2: Noisy image denoised using total variation (TV) minimization. Denoised image exhibits staircase effect along gradient changes [2].

these blocks to form a 3D array. This array undergoes a 3D transform, such as the Discrete Cosine Transform, followed by Wiener filtering to attenuate noise. The inverse transform then reconstructs the denoised blocks, which are aggregated back into their original positions to form the final image.

2.1.3 Deep Learning-based Denoisers

Recently, deep learning-based methods has become dominant in the image denoising problem domain. DnCNN [39] leverages residual learning to efficiently remove Gaussian noise from images. Denoising autoencoder [40] is a type of neural network that learns to reconstruct a clean input from a corrupted version of it. The model consists of an encoder that maps the input to a latent space, and a decoder that reconstructs the input from the latent representation. Lately, transformer networks such as Swin Transformer [41] are employed for image denoising. Swin Transformer is a hierarchical vision transformer that processes visual data by using shifted windows for self-attention computation. This enables the capturing of both local and global dependencies. In SwinIR [42], the network is adapted for image restoration. The architecture consists of three modules: shallow feature extraction, deep

feature extraction using residual Swin Transformer blocks (RSTBs), and a reconstruction module. SwinIR reconstructs a high-quality image by focusing on recovering lost high-frequency details while maintaining low-frequency stability through skip connections.

2.1.4 Graph-based Filters based on GSP

GSP-based methods have shown great promise in image denoising in recent years. In [43], a regularization termed called *optimal graph Laplacian regularizer* (OGLR), which assumes that the target pixel is smooth with respect to a graph, is derived for denoising in the discrete domain. An iterative algorithm with the same name OGLR is developed based on the derived Laplacian. Experimental results show that this method is competitive with state-of-the-art denoising schemes such as BM3D [38]. [44] combines model-based approach and deep learning by integrating graph Laplacian regularization as a trainable module in a deep learning network. This is shown to be less susceptible to overfitting than traditional CNNs and achieve higher performance with smaller datasets.

In [45], unrolling of *graph total variation* (GTV) [19, 46] as layers of a neural network is studied. GTV is defined as,

$$\text{GTV}(\mathbf{x}) = \sum_{(i,j) \in \mathcal{E}} w_{i,j} |x_i - x_j| \quad (5)$$

where $\mathbf{x}$ is the image signal represented as a vector of pixel intensities. $w_{i,j}$ of edge (i, j) is the non-negative weight encoding the similarity between nodes (pixels) i and j . These weights can be computed based on pairwise spatial or intensity similarity, as done in bilateral filter (BF) [26]. The denoising problem then can be formulated as

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \text{GTV}(\mathbf{x}), \quad (6)$$

where μ is a tradeoff parameter balancing the fidelity term and the signal smoothness term.

In [19], An ℓ_1 -Laplacian matrix $\mathbf{L}_\Gamma$ is introduced as

$$\mathbf{L}_\Gamma = \text{diag}(\mathbf{\Gamma}\mathbf{1}) - \mathbf{\Gamma}, \quad (7)$$

where $\mathbf{\Gamma}$ is the adjacency matrix with edge weights Γ_{ij} , given a signal estimate $\mathbf{x}^\circ$, as

$$\Gamma_{ij} = \frac{w_{i,j}}{\max\{|x_i^o - x_j^o|, \rho\}}, \quad (8)$$

where $\rho > 0$ is a chosen parameter to circumvent numerical instability when $|x_i^o - x_j^o| \approx 0$.

Using these definitions, (6) is reformulated as

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \mathbf{x}^\top \mathbf{L}_\Gamma \mathbf{x}. \quad (9)$$

Then, the optimization with only ℓ_2 -norm terms has a closed-form solution,

$$\mathbf{x}^* = (\mathbf{I} + \mu \mathbf{L}_\Gamma)^{-1} \mathbf{y}. \quad (10)$$

(10) is computed multiple times to solve the original denoising problem. This iteration is unrolled into several GTV layers in a neural network called DeepGTV, where each layer uses the signal reconstructed by the previous layer to generate an even better approximation of the signal as output.

2.2 Image Interpolation

Image interpolation is another fundamental operation in image processing, used for tasks such as image rescaling, translating, shrinking, rotating, warping, etc. Several techniques have been proposed for image interpolation, ranging from classical methods to modern deep learning approaches. Nearest neighbor interpolation is the simplest method, where the value of the nearest pixel is assigned to the interpolated pixel. It is computationally efficient but can produce blocky images with jagged edges. Bilinear interpolation [47] is another classical method, which calculates the interpolated pixel value as a weighted average of the four nearest pixel values. It provides smoother results than nearest neighbor interpolation but can introduce blurring. Bicubic interpolation [48] computes the weighted average of the 16 nearest pixels, providing even smoother results and better edge preservation compared to bilinear interpolation. However, it is more computationally intensive.

Spline interpolation (including B-splines [49], [50] and cubic splines [48]) is a more advanced interpolation technique modelling the image as a piecewise polynomial function. This method offers high-quality results, especially for smooth regions and gradual transitions.

Lanczos resampling image interpolation [51], which is based on sinc function, is effective in reducing aliasing artifacts and preserving image details. It is widely used in high-quality image scaling applications.

Recently, deep learning based image interpolation methods have been proposed. Super-Resolution Convolutional Neural Network (SRCNN) [52] employs a deep convolutional neural network (CNN) to perform image super-resolution. The network learns to map low-resolution images to high-resolution counterparts, effectively interpolating missing pixel values. Deep Image Prior (DIP) [53] proposed a randomly initialized CNN as a prior to reconstruct images. Without any training data, the network is optimized to fit the corrupted image, which can be applied to tasks like super-resolution and inpainting.

2.2.1 Demosaicking

Demosaicking is a critical interpolation task in digital imaging, aimed at reconstructing a full-color image from the incomplete color samples captured by a camera sensor equipped with a color filter array (CFA), such as the widely used Bayer pattern [1] (see Figure 1). In a Bayer-patterned grid, each pixel records only one of the three primary color components—red, green, or blue—leaving the other two components to be estimated. This process is inherently an interpolation problem, as it involves generating missing pixel values based on sparse observations, and is often complicated by noise in the raw sensor data. We study demosaicking as an practical example in our joint interpolation / denoising framework in this thesis.

Traditional demosaicking methods rely on simple interpolation techniques. For instance, bilinear interpolation [47] estimates missing color values by averaging neighboring pixel intensities of the same color channel. For a pixel at position (i, j) in a Bayer pattern, the missing green value $G(i, j)$ at a red pixel location can be computed as:

$$G(i, j) = \frac{G(i-1, j) + G(i+1, j) + G(i, j-1) + G(i, j+1)}{4}, \quad (11)$$

where $G(i-1, j)$, $G(i+1, j)$, etc., are the green values at neighboring positions. While computationally efficient, this method often introduces artifacts like color aliasing and blurring near edges due to its inability to exploit inter-channel correlations or local structures [7].

Bicubic interpolation [54], a more advanced technique than bilinear interpolation, estimates missing pixel values using a cubic polynomial over a 4×4 neighborhood of known pixels [48]. For a pixel value $f(x, y)$ at fractional coordinates (x, y) , it can be expressed as:

$$f(x, y) = \sum_{i=-1}^2 \sum_{j=-1}^2 f(m+i, n+j) w(x - (m+i)) w(y - (n+j)), \quad (12)$$

where $m = \lfloor x \rfloor$, $n = \lfloor y \rfloor$, $f(m+i, n+j)$ are the known pixel values in the 4×4 grid, and

$$w(t) = \begin{cases} \frac{3}{2}|t|^3 - \frac{5}{2}|t|^2 + 1 & \text{if } |t| \leq 1, \\ -\frac{1}{2}|t|^3 + \frac{5}{2}|t|^2 - 4|t| + 2 & \text{if } 1 < |t| \leq 2, \\ 0 & \text{if } |t| > 2 \end{cases} \quad (13)$$

is the cubic weighting function [48]. While offering smoother results than bilinear interpolation, its application in demosaicking can lead to color artifacts and smoothing in textured or edge regions [55].

Spline interpolation [56], particularly cubic splines, provides a smooth reconstruction of missing pixel values by fitting piecewise cubic polynomials between known data points [57]. For a 1D signal with known values $f(x_i)$ at points x_i , the cubic spline $s(x)$ between x_i and x_{i+1} is given by:

$$s(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x_i \leq x < x_{i+1}, \quad (14)$$

where coefficients a_i , b_i , c_i , and d_i are determined by ensuring continuity of the function and its first and second derivatives across knots [57]. In demosaicking, this can be extended to 2D by applying splines separably along rows and columns of the CFA data. While effective for smooth transitions, spline interpolation may oversmooth sharp edges and increase computational complexity [7].

To address some of these issues such as color artifacts and aliasing, *adaptive homogeneity-directed demosaicking* [58] leverages local edge directionality. It selects interpolation direction (horizontal or vertical) based on homogeneity measures, such as intensity differences, to preserve edges, though it may falter under noisy conditions.

Model-based approaches enhance demosaicking by casting it as an optimization problem. A notable example is *total variation (TV)-based demosaicking* [59], which jointly reconstructs

the full-color image while suppressing noise. In [59], Condat proposes a variational framework for joint demosaicking and denoising using total variation (TV) minimization. The approach formulates demosaicking as a constrained optimization problem, minimizing a TV norm that separates luminance and chrominance components:

$$\|\mathbf{a}\|_{\text{TV}} = \mu \|\nabla \mathbf{a}^L\|_{\ell_1} + \|\nabla \mathbf{a}^C\|_{\ell_1}, \quad (15)$$

for some parameter $\mu > 0$, where $\mathbf{a}^L$ and $\mathbf{a}^C$ are the luminance and chrominance components of the vector-valued image $\mathbf{a}$, and $\|\mathbf{a}\|_{\ell_1} = \sum_{\mathbf{k} \in \mathbb{Z}^2} \sqrt{\mathbf{a}[\mathbf{k}]^H \mathbf{a}[\mathbf{k}]}$ with H indicating the complex conjugate transpose [59]. The optimization problem is then:

$$\mathbf{d} = \arg \min_{\mathbf{a}} \|\mathbf{a}\|_{\text{TV}} \quad \text{s.t.} \quad \mathbf{a}^X[\mathbf{k}] = v[\mathbf{k}], \quad \forall \mathbf{k} \in \mathbb{Z}^2, \quad (16)$$

where $v[\mathbf{k}]$ contains the noisy CFA measurements at pixel positions $\mathbf{k}$, and $\mathbf{a}^X[\mathbf{k}]$ denotes the color channel $X \in \{R, G, B\}$ sampled at $\mathbf{k}$. While effective for edge preservation and noise reduction, the TV prior can introduce staircase artifacts.

Another model-based method, *gradient-based demosaicking* [60], interpolates using directional gradients, which improves edge fidelity. For instance, the green value at a red pixel is estimated as:

$$G(i, j) = \frac{G(i-1, j) + G(i+1, j) + G(i, j-1) + G(i, j+1)}{4} + \alpha \Delta_R(i, j), \quad (17)$$

where $\Delta_R(i, j)$ is the gradient of the red channel at that location, computed by:

$$\Delta_R(i, j) = r(i, j) - \frac{1}{4} \sum_{(m,n) \in \{(0,-2), (0,2), (-2,0), (2,0)\}} r(i+m, j+n), \quad (18)$$

where $r(i, j)$ denotes red pixel.

Deep learning-based methods have also been applied to demosaicking in recent years. [61] proposes a cascade of convolutional residual denoising networks (CRDNs) for demosaicking, modeling it as an inverse problem solved iteratively via proximal gradient descent. The CRDNs are trained to approximate the proximal operator by minimizing the following loss function:

$$\hat{Q}(\mathbf{x}, \mathbf{x}_0) = \frac{1}{2(\sigma/\sqrt{a})^2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \phi(\mathbf{x}) + c, \quad \text{where} \quad \mathbf{z} = \mathbf{y} + (\mathbf{I} - \mathbf{M})\mathbf{x}_0, \quad (19)$$

with σ as the noise level, a is a scaling factor for noise, $\mathbf{M}$ as the CFA sampling mask, $\phi(\mathbf{x})$ as the learned regularizer, c is a constant, $\mathbf{y}$ is the observed mosaicked image, and $\mathbf{x}_0$ is the initial estimate of the image.

[62] introduces a deep residual learning approach for demosaicking, focusing on recovering fine details in noise-free Bayer CFA data. They formulate demosaicking as an optimization problem inspired by MAP estimation with a Markov random fields prior:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{M} \odot \mathbf{x}\|_2^2 + \lambda \sum_{k=1}^K \sum_{p=1}^N \rho_k((\mathbf{f}_k * \mathbf{x})_p), \quad (20)$$

where $\mathbf{y}$ is the CFA image, $\mathbf{x}$ is the full-color image, $\mathbf{M}$ is the CFA sampling mask, λ is the regularization parameter, and the second term is an MRF prior with filters $\mathbf{f}_k$ and penalty functions $\rho_k(\cdot)$. The optimization is solved iteratively via gradient descent:

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \lambda \sum_{k=1}^K \left(\tilde{\mathbf{f}}_k * \phi_k(\mathbf{f}_k * \mathbf{x}_{t-1}) \right), \quad (21)$$

where $\tilde{\mathbf{f}}_k$ is the adjoint filter, and $\phi_k(\cdot) = \rho'_k(\cdot)$. The CNN learns to approximate these updates to refine the demosaicked image.

While effective, these methods depend on extensive training data, limiting adaptability to diverse noise profiles—a challenge our operator-agnostic GSP framework addresses analytically.

2.3 Joint Image Denoising / Interpolation

Joint demosaicking and denoising (JDD) is a well-studied problem in the imaging literature. Bayer-patterned grid pixels captured by cameras need to be demosaicked (interpolated) to produce missing RGB values on the 2D image grid. Doing this with every color channel will produce a full colour image. As with all other raw images, captured pixels on the Bayer-patterned grid are noise-corrupted, and thus require denoising.

To solve this problem, various approaches have been proposed to combat noise contamination during interpolation. A total least squares based denoising method is studied in [63]. In [64], the authors address the JDD problem by learning a series of energy minimizations. In [65], a convolutional neural network (CNN) is trained to map noisy Bayer

inputs M_i to clean RGB outputs O_i , minimizing an L_2 loss function over a training dataset $\mathcal{D} = \{(\sigma_i, M_i, I_i)\}_i$, where σ_i is the noise variance, and I_i is the ground-truth patch. The loss function is defined as:

$$\mathcal{L}(\{w^{(d)}, b^{(d)}\}_d) = \frac{1}{p^2|\mathcal{D}|} \sum_i \|O_i - I_i\|^2, \quad (22)$$

where p is the patch size, and the weights $w^{(d)}$ and biases $b^{(d)}$ are optimized using the ADAM algorithm [66]. The network learns to simultaneously interpolate missing colors and denoise, achieving high PSNR and detail preservation.

[67] employs a *Generative Adversarial Network (GAN)*, optimizing a combined loss function that includes pixel-wise, perceptual, and adversarial terms. The pixel-wise loss is defined as:

$$L_{\text{MSE}} = \frac{1}{3WH} \sum_{c=1}^3 \sum_{w=1}^W \sum_{h=1}^H \|(G_{\Theta_G}(I^{\text{input}}))_{c,w,h} - I_{w,h}^c\|, \quad (23)$$

where I^{input} is the noisy CFA input, I^c is the ground-truth image for channel c , and $G_{\Theta_G}(I^{\text{input}})$ is the network's output. The perceptual loss, based on features from the 19th layer of a pre-trained VGG19 network, is

$$L_p = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^c)_{x,y} - \phi_{i,j}(G_{\Theta_G}(I^{\text{input}}))_{x,y})^2, \quad (24)$$

where ϕ_j denotes the feature map at layer j . The adversarial loss for the generator is

$$L_A = -\frac{1}{N} \sum_{i=1}^N \log D_{\Theta_D}(I_i^{\text{output}}), \quad (25)$$

where $I_i^{\text{output}} = G_{\Theta_G}(I_i^{\text{input}})$, and D_{Θ_D} is the discriminator. The total loss combines these terms:

$$L = L_{\text{MSE}} + \lambda_p L_p + \lambda_A L_A, \quad (26)$$

where λ_p and λ_A are Lagrangian parameters to control tradeoff among the regularization terms.

Another well-studied joint denoising / interpolation problem is joint denoising and super-resolution. The goal of super-resolution problem is to reconstruct a high-resolution image from one or more low-resolution images. During this process, noise in a low-resolution

image will be amplified and affect the quality of the reconstructed high-resolution image. A dictionary-based solution for solving super-resolution of depth images is presented in [68].

Existing literature often deals with specific denoisers and interpolators. In our work, however, we develop theorems that are *operator-agnostic*, as long as they satisfy certain specified conditions (to be discussed in detail). This results in broad applicability to a wide range of image transformation tasks. Note also that we are not proposing any new denoiser or interpolator; rather, we investigate how best to combine a denoiser and an interpolator, so that the joint task can be optimally performed.

2.4 Graph Spectral Image Processing

Graph signal processing (GSP) is a relatively new field that studies computation tools like transforms and wavelets for discrete signals on irregular data kernels described by finite graphs. An image can be interpreted as a graph signal, where each pixel is represented by a node in the graph. GSP tools have been used for many imaging problems [16], such as image compression [17], [69], denoising [43], [44], [45], soft decoding of JPEG images [70], deblurring [19], contrast enhancement [20], and interpolation [35]. The general approach in GSP-based image processing literature is to construct an undirected similarity graph that encodes inter-pixel relations based on predefined metrics, such as pixel intensity similarity. This similarity graph serves as an underlying data kernel to derive optimal low-pass graph filters via convex optimization, which are then used to perform specific image processing tasks. Specifically, given a similarity graph, a suitable graph filter is derived by solving a *maximum a posteriori* (MAP) problem regularized by a chosen graph smoothness prior. Popular graph smoothness priors include graph Laplacian regularizer (GLR) [43] and graph total variation (GTV) [19, 46]. Each smoothness prior requires a well-defined graph, which can be constructed either statistically or using machine learning techniques. There are model-based statistical graph learning approaches [71–74], as well as deep learning based methods [44, 45], which can be used to construct the underlying graphs.

In this proposal, we study the optimization of a joint denoising / interpolation operation from a GSP perspective. To derive an optimal graph-based denoiser, GLR [43] is used as the signal prior in our formulation, where an *undirected* graph is used to model inter-pixel

similarities. On the other hand, to derive an optimal graph-based interpolator, we study the mapping from original pixels to interpolated pixels using a *directed* graph; this leads to a directed graph regularization term called *graph shift variation* (GSV) [23] for optimization. Using a directed graph here makes sense for image interpolation, as original pixels should influence newly interpolated pixels but not vice versa. To our knowledge, we are the first in the literature to employ a directed graph for image interpolation.

3 Preliminaries

3.1 Graph Signal Processing

We review basic definitions in GSP to facilitate subsequent discussions on algorithm development. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of a set of N nodes $\mathcal{V} = \{1, \dots, N\}$ and a set of edges $\mathcal{E}$. The edge set can be specified as $\{(i, j, w_{i,j})\}$, where $i, j \in \mathcal{V}$, and $w_{i,j} \in \mathbb{R}$ is a real-valued scalar weight (which can be positive or negative) of an edge $(i, j) \in \mathcal{E}$ that connects nodes i and j . This weight reflects the degree of (dis)similarity between signal values at nodes i and j —the larger the weight $w_{i,j}$, the more similar the signal values x_i and x_j are expected to be.

An *adjacency matrix* can be defined as $\mathbf{W} \in \mathbb{R}^{N \times N}$, where $W_{i,j} = w_{i,j}$ if $(i, j) \in \mathcal{E}$, and $W_{i,j} = 0$ otherwise. In our analysis, we consider both undirected and directed graphs. For an undirected graph, $\mathbf{W}$ is symmetric, i.e., $W_{i,j} = W_{j,i}, \forall i, j \in \mathcal{V}$.

A diagonal *degree matrix* can be defined as $\mathbf{D} \in \mathbb{R}^{N \times N}$, where $D_{i,i} = \sum_j W_{i,j}$. Given defined $\mathbf{W}$ and $\mathbf{D}$, we can define a *combinatorial graph Laplacian matrix* as $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$. To account for cases where the graph includes self-loops (i.e., $w_{i,i} \neq 0$ for some i) the *generalized graph Laplacian matrix* $\mathbf{L}_g$ is defined as $\mathbf{L}_g \triangleq \mathbf{D} - \mathbf{W} + \text{diag}(\mathbf{W})$. In both cases, we assume that the Laplacian matrix is *positive semi-definite* (PSD), which means that it satisfies the condition $\mathbf{x}^\top \mathbf{L} \mathbf{x} \geq 0, \forall \mathbf{x}$. If the constructed graph is positive, i.e., $w_{i,j} \geq 0, \forall i, j \in \mathcal{V}$, then $\mathbf{L}$ and $\mathbf{L}_g$ are provably PSD [75]. If a matrix $\mathbf{L}$ is real and symmetric, then by the Spectral Theorem [76], it can be eigen-decomposed into $\mathbf{L} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top$, where $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_N]$ is the matrix of orthonormal eigenvectors as columns, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ is the diagonal matrix of real-valued eigenvalues. $\mathbf{L}$ being PSD means $\lambda_k \geq 0, \forall k$.

If we assign a signal value $x_i \in \mathbb{R}$ to each node i , then the ensemble $\mathbf{x} = [x_1 \dots x_N]^\top \in \mathbb{R}^N$ is called a *graph signal* on graph $\mathcal{G}$. The *graph spectrum* of an undirected graph $\mathcal{G}$ consists of the eigen-pairs $(\lambda_k, \mathbf{v}_k)$, for $k \in \{1, \dots, N\}$, of the Laplacian matrix $\mathbf{L}$. In this context, the eigenvalue λ_k represents the k -th frequency, and the corresponding eigenvector $\mathbf{v}_k$ is the k -th graph Fourier mode. The GFT coefficients of a signal $\mathbf{x}$ are given by $\tilde{\mathbf{x}} = \mathbf{V}^\top \mathbf{x}$.

3.1.1 Graph Smoothness Priors

There are various graph smoothness priors proposed in the GSP literature, each assuming that the signal $\mathbf{x}$ is smooth with respect to the underlying graph $\mathcal{G}$, but expressed using different mathematical formulations. The most common prior is the *graph Laplacian regularizer* (GLR) [22]:

$$\mathbf{x}^\top \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} (x_i - x_j)^2 = \sum_k \lambda_k \tilde{x}_k^2. \quad (27)$$

Here, $\mathbf{L}$ is the combinatorial graph Laplacian for graph $\mathcal{G}$, λ_k is the k -th eigenvalue of $\mathbf{L}$, and $\tilde{x}_k$ is the k -th GFT coefficient for signal $\mathbf{x}$, obtained by computing $\tilde{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$. From (27), in the nodal domain, a small GLR implies that connected node pairs (i, j) with large edge weights $w_{i,j}$ should have similar signal values x_i and x_j . In the graph frequency domain, a small GLR indicates that the signal energies $\tilde{x}_k^2$ should primarily be concentrated in low frequencies λ_k , implying that the signal $\mathbf{x}$ should exhibit low-pass characteristics.

Another frequently used graph smoothness prior is the *graph shift variation* (GSV) [23]. To define it, we first introduce the row-stochastic adjacency matrix $\mathbf{A}_r \triangleq \mathbf{D}^{-1} \mathbf{A}$. The GSV is then expressed as

$$\begin{aligned} \|\mathbf{x} - \mathbf{A}_r \mathbf{x}\|_2^2 &= \|(\mathbf{I} - \mathbf{A}_r) \mathbf{x}\|_2^2 = \|\mathbf{L}_r \mathbf{x}\|_2^2 \\ &= \sum_k \tilde{\lambda}_k^2 \tilde{x}_k^2. \end{aligned} \quad (28)$$

Here, $\tilde{\lambda}_k$ is the k -th eigenvalue of the random walk graph Laplacian $\mathbf{L}_r \triangleq \mathbf{D}^{-1} \mathbf{L}$, and $\tilde{x}_k$ is the k -th coefficient obtained from the transform $\tilde{\mathbf{V}}^\top \mathbf{x}$, where $\tilde{\mathbf{V}}$ contains the right eigenvectors of $\mathbf{L}_r$ as its columns, *i.e.*, $\mathbf{L}_r = \tilde{\mathbf{U}} \text{diag}(\tilde{\lambda}_k) \tilde{\mathbf{V}}^\top$.

GSV (28) represents the ℓ_2 -norm difference between the signal $\mathbf{x}$ and its shifted version $\mathbf{A}_r \mathbf{x}$, where the row-stochastic adjacency matrix $\mathbf{A}_r$ serves as the *graph shift operator*. Similar to GLR in (27), a small GSV indicates that signal $\mathbf{x}$ has its energy predominantly in the low frequencies $\tilde{\lambda}_k$.

The GSV can also be expressed as $\mathbf{x}^\top \mathbf{L}_r^\top \mathbf{L}_r \mathbf{x}$, which was referred to as the *left eigenvectors of the random walk graph Laplacian* (LERaG) in [77]. This formulation was proposed for JPEG image dequantization as a normalized signal prior that evaluates to zero for a constant

signal. Unlike GLR (27), a notable feature of GSV (28) is that it remains well-defined even when the graph $\mathcal{G}$ is directed.

There are other graph smoothness priors discussed in the literature, such as *graph total variation* (GTV) [78, 79] and *gradient graph Laplacian regularizer* (GGLR) [80]. However, in this thesis, we focus on GLR and GSV not only because they are widely used, but also due to their convenient quadratic form that is amenable to fast optimization.

4 Mappings between Linear Denoiser / Interpolator and Graph Filters

We first describe two theorems that connect a linear denoiser / interpolator to a corresponding undirected / directed graph filter.

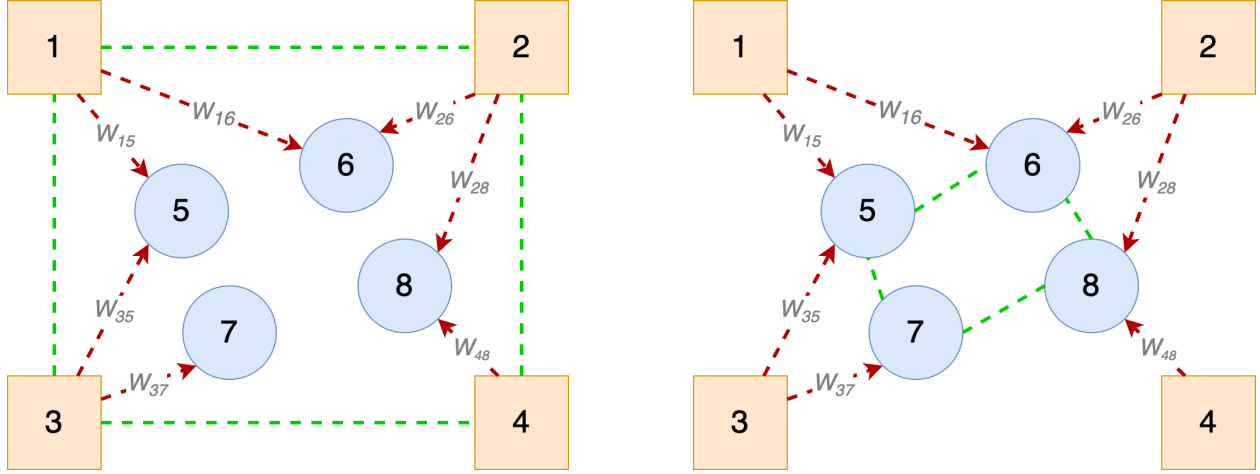


Figure 3: Illustrations of mixed graphs for joint interpolation / denoising. Pixels 5,6,7,8 are interpolated using pixels 1,2,3,4. Directed edges for interpolation are shown as red dashed arrows, while undirected edges for denoising are shown as green dashed lines. (a) corresponds to equation 41 where input pixels are denoised before interpolation. (b) corresponds to equation 46 where interpolated pixels are denoised after interpolation.

4.1 Mapping between Denoiser and Undirected Graph Filter

We begin by stating a theorem, similar to one in [81], that connects a denoising matrix $\Psi \in \mathbb{R}^{N \times N}$ to the solution filter of a graph-regularized *maximum a posteriori* (MAP) optimization problem. Unlike [81], which relied on a proximal map property to demonstrate the relationship, our derivation is based purely on linear algebra, providing a simpler and more direct interpretation.

Consider a linear denoising operation described by

$$\mathbf{x} = \Psi \mathbf{y} \tag{29}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ are the output and input signals of the denoiser, respectively. Theorem 1 in [81] assumes that Ψ is symmetric, doubly stochastic¹, and invertible. The doubly stochastic property implies *non-expansiveness*, meaning $\|\Psi\mathbf{x}\|^2 \leq \|\mathbf{x}\|^2$ for all $\mathbf{x}$, which in turn means that the eigenvalues λ_k 's of Ψ satisfy $|\lambda_k| \leq 1$ for all k .

In contrast, we assume that Ψ is non-expansive, symmetric, and *positive definite* (PD), *i.e.*, $0 < \lambda_k \leq 1$ for all k . Thus, unlike [81], we do not require the largest eigenvalue of Ψ to be exactly one (corresponding to eigenvector $\mathbf{1}$), *i.e.*, $\Psi\mathbf{1} = \mathbf{1}$. On the other hand, we require eigenvalues to be *strictly* positive.

The filter weights of a *signal-dependent* denoiser $\Psi(\mathbf{x})$, such as the well-known *bilateral filter* (BF) [26], depend on the target signal $\mathbf{x}$, and thus $\Psi(\mathbf{x})$ is non-linear. However, it can be considered *pseudo-linear* [81]: in each iteration t of an iterative filtering process, we use a fixed linear filter $\Psi(\mathbf{x}_{t-1})$ calculated from the previous estimated solution $\mathbf{x}_{t-1}$, as is commonly done [82].

Next, consider the following standard MAP optimization problem for denoising the input $\mathbf{y}$, using GLR (27) [22] as the signal prior:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu \mathbf{x}^\top \mathbf{L} \mathbf{x}. \quad (30)$$

Here, $\mu > 0$ is a strictly positive weight parameter, and $\mathbf{L}$ is a graph Laplacian matrix for the underlying graph $\mathcal{G}$. Assuming that $\mathbf{L}$ is PSD, (30) is an unconstrained and convex *quadratic programming* (QP) problem with solution

$$\mathbf{x}^* = (\mathbf{I}_N + \mu \mathbf{L})^{-1} \mathbf{y} \quad (31)$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. Note that the coefficient matrix $\mathbf{I}_N + \mu \mathbf{L}$ is provably *positive definite* (PD) via Weyl's Inequality [83] and thus invertible. We now connect denoiser Ψ (29) and the solution to the MAP problem (30):

Theorem 1. *Denoiser Ψ (29) is the solution filter for the MAP problem (30) if $\mathbf{L} = \mu^{-1}(\Psi^{-1} - \mathbf{I}_N)$, assuming matrix Ψ is non-expansive, symmetric, and PD [84].*

¹Entries of each row and column of a doubly stochastic matrix sum to one. [82] also assumed that the filtering matrix is doubly stochastic.

Proof. Symmetry means Ψ has real eigenvalues. Non-expansiveness and PDness mean Ψ is invertible with positive eigenvalues $0 < \lambda_k \leq 1, \forall k$. Thus, Ψ^{-1} exists and has positive eigenvalues $\{\lambda_k^{-1}\}$. Condition $\lambda_k \leq 1, \forall k$ means that $1 \leq \lambda_k^{-1}, \forall k$. Thus, the eigenvalues of $\Psi^{-1} - \mathbf{I}_N$ are $\lambda_k^{-1} - 1 \geq 0, \forall k$. This implies $\mathbf{L} = \mu^{-1}(\Psi^{-1} - \mathbf{I}_N)$ is PSD, and thus quadratic objective (30) is convex. Taking derivative w.r.t. $\mathbf{x}$ and setting it to zero, optimization (30) has (31) as solution. Inserting $\mathbf{L} = \mu^{-1}(\Psi^{-1} - \mathbf{I}_N)$ into (31), we get $\mathbf{x}^* = \Psi \mathbf{y}$, and thus Ψ is the resulting solution filter. $\square$

4.2 Mapping between Interpolator and Directed Graph Filter

We now describe an analogous theorem for a linear interpolator. Consider a linear interpolator $\Theta \in \mathbb{R}^{N \times M}$ that generates N new pixels from M original pixels $\mathbf{y} \in \mathbb{R}^M$. Mathematically, we express this as

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{I}_M \\ \Theta \end{bmatrix} \mathbf{y}. \quad (32)$$

Here, $\mathbf{x} = [\mathbf{y}^\top \quad \tilde{\mathbf{x}}^\top]^\top \in \mathbb{R}^{M+N}$ is the length- $(M+N)$ target signal that includes the original M pixels.

Next, we state a MAP optimization objective for interpolation, similar to the previous objective (30) for denoising. Let $\mathbf{H} = [\mathbf{I}_M \quad \mathbf{0}_{M,N}]$ be a $M \times (M+N)$ *sampling matrix* that selects the original M pixels from signal $\mathbf{x}$, where $\mathbf{0}_{M,N}$ is a $M \times N$ matrix of zeros. Denote by $\mathbf{A}$ an *asymmetric* adjacency matrix that specifies *directional* edges in a directed graph $\mathcal{G}^d$ for signal $\mathbf{x}$. Specifically, $\mathbf{A}$ describes edges only from the M original pixels to the N new pixels:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{M,M} & \mathbf{A}_{M,N} \\ \mathbf{0}_{N,M} & \mathbf{0}_{N,N} \end{bmatrix}. \quad (33)$$

We now formulate a MAP optimization objective using GSV (28) as the signal prior:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \gamma \|\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x})\|_2^2, \quad (34)$$

where $\gamma > 0$ is a strictly positive weight parameter. The term $\|\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x})\|_2^2$ in GSV states that a smooth graph signal $\mathbf{x}$ should be similar to its shifted version $\mathbf{A}\mathbf{x}$, but it considers

only the M original pixels in the objective. We note that (34) is convex for *any* definition of $\mathbf{A}$. We state formally a theorem to connect interpolator $[\mathbf{I}_M; \boldsymbol{\Theta}]$ in (32) and the solution to the MAP problem (34).

Theorem 2. *The interpolator $[\mathbf{I}_M; \boldsymbol{\Theta}]$ (32) is the solution filter to the MAP problem (34) if $M = N$, $\boldsymbol{\Theta}$ is invertible, and $\mathbf{A}_{M,N} = \boldsymbol{\Theta}^{-1}$ [84].*

Proof. we rewrite $\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x}) = \mathbf{H}(\mathbf{I} - \mathbf{A})\mathbf{x}$. Given (34) is convex for any $\mathbf{A}$, we take the derivative w.r.t. $\mathbf{x}$ and set it to 0, resulting in

$$\left(\mathbf{H}^\top \mathbf{H} + \gamma(\mathbf{I} - \mathbf{A})^\top \mathbf{H}^\top \mathbf{H}(\mathbf{I} - \mathbf{A}) \right) \mathbf{x} = \mathbf{H}^\top \mathbf{y}. \quad (35)$$

Given the definitions of $\mathbf{H}$ and $\mathbf{A}$, we can rewrite (35) as

$$\underbrace{\begin{bmatrix} (1 + \gamma)\mathbf{I}_M & -\gamma\mathbf{A}_{M,N} \\ -\gamma\mathbf{A}_{M,N}^\top & \gamma\mathbf{A}_{N,N}^2 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} = \mathbf{H}^\top \mathbf{y}, \quad (36)$$

where $\mathbf{A}_{N,N}^2 = \mathbf{A}_{M,N}^\top \mathbf{A}_{M,N}$. (We write $\mathbf{A}_{M,N}^\top$ instead of $\mathbf{A}_{N,M}$ to remind ourselves that though $M = N$ is a required condition in Theorem 2, $\mathbf{A}_{M,N}$ is asymmetric in general.)

Using a matrix inversion formula [85],

$$\begin{bmatrix} \dot{\mathbf{A}} & \dot{\mathbf{B}} \\ \dot{\mathbf{C}} & \dot{\mathbf{D}} \end{bmatrix}^{-1} = \begin{bmatrix} \dot{\mathbf{P}} & -\dot{\mathbf{P}}\dot{\mathbf{B}}\dot{\mathbf{D}}^{-1} \\ -\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}}\dot{\mathbf{P}} & \dot{\mathbf{D}}^{-1} + \dot{\mathbf{D}}^{-1}\dot{\mathbf{C}}\dot{\mathbf{P}}\dot{\mathbf{B}}\dot{\mathbf{D}}^{-1} \end{bmatrix} \quad (37)$$

where $\dot{\mathbf{P}} = (\dot{\mathbf{A}} - \dot{\mathbf{B}}\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}})^{-1}$, we first compute $\dot{\mathbf{P}}$ for coefficient matrix $\mathbf{C}$ in (36) as

$$\begin{aligned} \dot{\mathbf{P}} &= ((1 + \gamma)\mathbf{I}_M - (-\gamma\mathbf{A}_{M,N})(\gamma\mathbf{A}_{N,N}^2)^{-1}(-\gamma\mathbf{A}_{M,N}^\top))^{-1} \\ &= ((1 + \gamma)\mathbf{I}_M - \gamma\mathbf{A}_{M,N}(\mathbf{A}_{M,N}^\top \mathbf{A}_{M,N})^{-1}\mathbf{A}_{M,N}^\top)^{-1} \\ &\stackrel{(a)}{=} ((1 + \gamma)\mathbf{I}_M - \gamma\mathbf{A}_{M,N}\mathbf{A}_{M,N}^{-1}(\mathbf{A}_{M,N}^\top)^{-1}\mathbf{A}_{M,N}^\top)^{-1} \\ &= ((1 + \gamma)\mathbf{I}_M - \gamma\mathbf{I}_M)^{-1} = \mathbf{I}_M, \end{aligned} \quad (38)$$

where in (a) we apply the assumptions that $N = M$ and $\mathbf{A}_{M,N} = \boldsymbol{\Theta}^{-1}$.

We can now write $\mathbf{x}$ from (36) as

$$\begin{aligned}
\mathbf{x} &= \mathbf{C}^{-1} \mathbf{H}^\top \mathbf{y} = \begin{bmatrix} \mathbf{I}_M \\ -(\gamma \mathbf{A}_{N,N}^2)^{-1} (-\gamma \mathbf{A}_{M,N}^\top) \mathbf{I}_M \end{bmatrix} \mathbf{y} \\
&= \begin{bmatrix} \mathbf{I}_M \\ \mathbf{A}_{M,N}^{-1} (\mathbf{A}_{M,N}^\top)^{-1} \mathbf{A}_{M,N}^\top \end{bmatrix} \mathbf{y} = \begin{bmatrix} \mathbf{I}_M \\ \mathbf{A}_{M,N}^{-1} \end{bmatrix} \mathbf{y}.
\end{aligned} \tag{39}$$

Given $\mathbf{A}_{M,N}^{-1} = \mathbf{\Theta}$ by assumption, we conclude that

$$\mathbf{x} = \begin{bmatrix} \mathbf{I}_M \\ \mathbf{\Theta} \end{bmatrix} \mathbf{y}. \tag{40}$$

Hence, $[\mathbf{I}_M; \mathbf{\Theta}]$ is the solution filter to (34). $\square$

4.3 Joint Denoising and Interpolation

Having described the mappings between linear denoisers / interpolators to their corresponding graph filters, we next investigate joint denoising and interpolation from a graph perspective. Let $\mathbf{A} \in \mathbb{R}^{(M+N) \times (M+N)}$, in the form (33), be the adjacency matrix of a *directed* graph $\mathcal{G}^d$ that connects M original pixels to N new pixels, corresponding to an interpolator $\mathbf{\Theta}$. Additionally, let $\mathbf{L} \in \mathbb{R}^{M \times M}$ be the graph Laplacian matrix for an *undirected* graph $\mathcal{G}^u$ that interconnects the M noisy original pixels, corresponding to a denoiser $\mathbf{\Psi}$.

4.3.1 Joint Formulation with Separable Solution

A direct approach to formulate a joint denoising / interpolation scheme is to combine the MAP problems (30) and (34) as

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \gamma \|\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x})\|_2^2 + \mu (\mathbf{H}\mathbf{x})^\top \mathbf{L}(\mathbf{H}\mathbf{x}). \tag{41}$$

In words, (41) states that the desired signal $\mathbf{x}$ should be smooth with respect to *two* graphs, $\mathcal{G}^d$ and $\mathcal{G}^u$. Specifically, $\mathbf{x}$ should resemble its shifted version $\mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is the adjacency matrix of a *directed* graph $\mathcal{G}^d$, and the original pixels $\mathbf{H}\mathbf{x}$ should be smooth with respect to an *undirected* graph $\mathcal{G}^u$ defined by $\mathbf{L}$. This approach aligns with our intuition from Section 4.1 and 4.2: using *directional* edges to connect original and interpolated pixels and *bidirectional* edges to connect noisy pixels.

We show that the optimal solution to the posed MAP problem (41) takes a particular form.

Corollary 1. (41) *has an optimal separable solution.*

Proof. Optimization (41) is an unconstrained convex QP problem. Taking the derivative w.r.t. $\mathbf{x}$ and setting it to 0, we get

$$(\mathbf{H}^\top \mathbf{H} + \gamma ((\mathbf{I} - \mathbf{A})^\top \mathbf{H}^\top \mathbf{H} (\mathbf{I} - \mathbf{A})) + \mu \mathbf{H}^\top \mathbf{L} \mathbf{H}) \mathbf{x}^* = \mathbf{H}^\top \mathbf{y}. \quad (42)$$

Denote by $\mathbf{C}$ the coefficient matrix on the left-hand side. Given $\mathbf{H} = [\mathbf{I}_M \ \mathbf{0}_{M,N}]$, $\mathbf{H}^\top \mathbf{L} \mathbf{H}$ has nonzero term $\mathbf{L}$ only in the upper-left sub-matrix. Given (36), $\mathbf{C}$ differs only in the upper-left block, *i.e.*,

$$\mathbf{C} = \begin{bmatrix} (\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} & -\gamma\mathbf{A}_{M,N} \\ -\gamma\mathbf{A}_{M,N}^\top & \gamma\mathbf{A}_{N,N}^2 \end{bmatrix}. \quad (43)$$

Using the matrix inverse formula (37), we can first write block $\dot{\mathbf{P}} = (\dot{\mathbf{A}} - \dot{\mathbf{B}}\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}})^{-1}$ as:

$$\begin{aligned} \dot{\mathbf{P}} &= \left((\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} - (-\gamma\mathbf{A}_{M,N})(\gamma\mathbf{A}_{N,N}^2)^{-1}(-\gamma\mathbf{A}_{M,N}^\top) \right)^{-1} \\ &= ((\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} - \gamma\mathbf{I}_M)^{-1} = (\mathbf{I}_M + \mu\mathbf{L})^{-1}. \end{aligned} \quad (44)$$

Thus, the solution $\mathbf{x}^*$ for optimization (42) is

$$\begin{aligned} \mathbf{x}^* &= \mathbf{C}^{-1} \mathbf{H}^\top \mathbf{y} \\ &= \begin{bmatrix} (\mathbf{I}_M + \mu\mathbf{L})^{-1} \\ -(\gamma\mathbf{A}_{N,N}^2)^{-1}(-\gamma\mathbf{A}_{M,N}^\top)(\mathbf{I}_M + \mu\mathbf{L})^{-1} \end{bmatrix} \mathbf{y} \\ &= \begin{bmatrix} (\mathbf{I}_M + \mu\mathbf{L})^{-1} \\ \mathbf{A}_{M,N}^{-1}(\mathbf{I}_M + \mu\mathbf{L})^{-1} \end{bmatrix} \mathbf{y} = \begin{bmatrix} \boldsymbol{\Psi} \\ \boldsymbol{\Theta} \boldsymbol{\Psi} \end{bmatrix} \mathbf{y}, \end{aligned} \quad (45)$$

where $\boldsymbol{\Psi} = (\mathbf{I}_M + \mu\mathbf{L})^{-1}$ is the denoiser, and $\boldsymbol{\Theta} = \mathbf{A}_{M,N}^{-1}$ is the interpolator. We see that solution (45) to optimization (41) is *entirely separable*: input $\mathbf{y}$ first undergoes denoising via original denoiser $\boldsymbol{\Psi}$, then subsequently interpolation via original interpolator $\boldsymbol{\Theta}$. $\square$

4.3.2 Joint Formulation with Non-separable Solution (One Denoiser)

Next, we consider a scenario where we introduce a GLR smoothness prior [22] for the N interpolated pixels instead of a smoothness prior for the M original pixels in (41), resulting in

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \gamma \|\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x})\|_2^2 + \kappa(\mathbf{G}\mathbf{x})^\top \bar{\mathbf{L}}(\mathbf{G}\mathbf{x}), \quad (46)$$

where $\mathbf{G} = [\mathbf{0}_{N,M} \ \mathbf{I}_N]$, a complementary matrix to $\mathbf{H}$, selects only the N new pixels from signal $\mathbf{x}$, and $\bar{\mathbf{L}} \in \mathbb{R}^{N \times N}$ denotes a graph Laplacian matrix for an undirected graph $\bar{\mathcal{G}}^u$ connecting the interpolated pixels.

Corollary 2. (46) has an optimal non-separable solution.

Proof. (46) is convex, quadratic and differentiable. The optimal solution $\mathbf{x}^*$ can be computed via a system of linear equations,

$$(\mathbf{H}^\top \mathbf{H} + \gamma ((\mathbf{I} - \mathbf{A})^\top \mathbf{H}^\top \mathbf{H} (\mathbf{I} - \mathbf{A})) + \mathcal{L}) \mathbf{x}^* = \mathbf{H}^\top \mathbf{y}, \quad (47)$$

where $\mathcal{L} = \kappa \mathbf{G}^\top \bar{\mathbf{L}} \mathbf{G}$ and is block-diagonal, *i.e.*,

$$\mathcal{L} = \begin{bmatrix} \mathbf{0}_M & \mathbf{0}_{M,N} \\ \mathbf{0}_{N,M} & \kappa \bar{\mathbf{L}} \end{bmatrix}. \quad (48)$$

A similar derivation shows that coefficient matrix $\mathbf{C}$ changes from (43) to

$$\mathbf{C} = \begin{bmatrix} (\gamma + 1)\mathbf{I}_M & -\gamma \mathbf{A}_{M,N} \\ -\gamma \mathbf{A}_{M,N}^\top & \gamma \mathbf{A}_{N,N}^2 + \kappa \bar{\mathbf{L}} \end{bmatrix}. \quad (49)$$

Using again the matrix inverse formula (37), we first write block $\dot{\mathbf{P}} = (\dot{\mathbf{A}} - \dot{\mathbf{B}}\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}})^{-1}$ as

$$\begin{aligned} \dot{\mathbf{P}} &= ((\gamma + 1)\mathbf{I}_M - (-\gamma \mathbf{A}_{M,N})(\gamma \mathbf{A}_{N,N}^2 + \kappa \bar{\mathbf{L}})^{-1}(-\gamma \mathbf{A}_{M,N}^\top))^{-1} \\ &= ((\gamma + 1)\mathbf{I}_M - \gamma^2 \mathbf{A}_{M,N}(\gamma \mathbf{A}_{N,N}^2 + \kappa \bar{\mathbf{L}})^{-1} \mathbf{A}_{M,N}^\top)^{-1}. \end{aligned} \quad (50)$$

The solution for $\mathbf{x}^*$ in (47) is

$$\begin{aligned} \mathbf{x}^* &= \mathbf{C}^{-1} \mathbf{H}^\top \mathbf{y} = \begin{bmatrix} \dot{\mathbf{P}} \\ -(\gamma \mathbf{A}_{N,N}^2 + \kappa \bar{\mathbf{L}})^{-1}(-\gamma \mathbf{A}_{M,N}^\top) \dot{\mathbf{P}} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\mathbf{P}} \\ \gamma(\gamma \mathbf{A}_{N,N}^2 + \kappa \bar{\mathbf{L}})^{-1} \mathbf{A}_{M,N}^\top \dot{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} \Psi^* \\ \Theta^* \Psi^* \end{bmatrix} \mathbf{y}, \end{aligned} \quad (51)$$

where $\Psi^* = \dot{\mathbf{P}}$ is a derived denoiser for original pixels, and $\Theta^* = \gamma(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}\mathbf{A}_{M,N}^\top$ is an augmented interpolator. Since $\Theta^*\Psi^* \neq \bar{\Psi}\Theta$, the solution to (46) is not separable². $\square$

In Section 5, we will evaluate the derived joint interpolation / denoising operators in (51) for joint denoising and interpolation and compare them to the conventional sequential approach in non-separable scenarios. Next, we extend the joint formulation (46) to a more general case with *two* denoisers: one for the M original pixels, and one for the N interpolated pixels.

4.3.3 Joint Formulation with Non-separable Solution (Two Denoisers)

We consider a scenario where we introduce an additional GLR smoothness prior [22] for the N interpolated pixels to the formulation in (41), resulting in

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \gamma\|\mathbf{H}(\mathbf{x} - \mathbf{A}\mathbf{x})\|_2^2 + \mu(\mathbf{H}\mathbf{x})^\top \mathbf{L}(\mathbf{H}\mathbf{x}) + \kappa(\mathbf{G}\mathbf{x})^\top \bar{\mathbf{L}}(\mathbf{G}\mathbf{x}) \quad (52)$$

where $\mathbf{G} = [\mathbf{0}_{N,M} \ \mathbf{I}_N]$ again selects only the N interpolated pixels from signal $\mathbf{x}$, and $\bar{\mathbf{L}} \in \mathbb{R}^{N \times N}$ denotes a graph Laplacian matrix for an undirected graph $\bar{\mathcal{G}}^u$ connecting the interpolated pixels.

Corollary 3. (52) *has an optimal non-separable solution.*

Proof. Like (41) and (46), (52) remains convex, quadratic and differentiable. The optimal solution $\mathbf{x}^*$ can be computed via a system of linear equations,

$$(\mathbf{H}^\top \mathbf{H} + \gamma((\mathbf{I} - \mathbf{A})^\top \mathbf{H}^\top \mathbf{H}(\mathbf{I} - \mathbf{A})) + \mathcal{L}) \mathbf{x}^* = \mathbf{H}^\top \mathbf{y}, \quad (53)$$

where $\mathcal{L} = \mu\mathbf{H}^\top \mathbf{L} \mathbf{H} + \kappa\mathbf{G}^\top \bar{\mathbf{L}} \mathbf{G}$ combines the two GLR priors corresponding to undirected graphs $\mathcal{G}^u$ and $\bar{\mathcal{G}}^u$ and is block-diagonal, *i.e.*,

$$\mathcal{L} = \begin{bmatrix} \mu\mathbf{L} & \mathbf{0}_{M,N} \\ \mathbf{0}_{N,M} & \kappa\bar{\mathbf{L}} \end{bmatrix}. \quad (54)$$

²Though the solution $\Theta^*\Psi^*$ to (46) is a sequence of two separate matrix operations, each matrix is a non-separable function of input matrices $\{\mathbf{A}, \bar{\mathbf{L}}\}$ or $\{\Theta, \bar{\Psi}\}$.

A similar derivation shows that coefficient matrix $\mathbf{C}$ changes from (43) to

$$\mathbf{C} = \begin{bmatrix} (\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} & -\gamma\mathbf{A}_{M,N} \\ -\gamma\mathbf{A}_{M,N}^\top & \gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}} \end{bmatrix}. \quad (55)$$

Using again the matrix inverse formula (37), we first write block $\dot{\mathbf{P}} = (\dot{\mathbf{A}} - \dot{\mathbf{B}}\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}})^{-1}$ as

$$\begin{aligned} \dot{\mathbf{P}} &= ((\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} - (-\gamma\mathbf{A}_{M,N})(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}(-\gamma\mathbf{A}_{M,N}^\top))^{-1} \\ &= ((\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} - \gamma^2\mathbf{A}_{M,N}(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}\mathbf{A}_{M,N}^\top)^{-1}. \end{aligned} \quad (56)$$

The solution for $\mathbf{x}^*$ in (47) is

$$\begin{aligned} \mathbf{x}^* &= \mathbf{C}^{-1}\mathbf{H}^\top \mathbf{y} \\ &= \begin{bmatrix} \dot{\mathbf{P}} \\ -(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}(-\gamma\mathbf{A}_{M,N}^\top)\dot{\mathbf{P}} \end{bmatrix} \\ &= \begin{bmatrix} \dot{\mathbf{P}} \\ \gamma(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}\mathbf{A}_{M,N}^\top\dot{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}^* \\ \boldsymbol{\Theta}^* \boldsymbol{\Psi}^* \end{bmatrix} \mathbf{y} \end{aligned} \quad (57)$$

where $\boldsymbol{\Psi}^* = \dot{\mathbf{P}}$ is a modified denoiser, and $\boldsymbol{\Theta}^* = \gamma(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}\mathbf{A}_{M,N}^\top$ is a modified interpolator. Since $\boldsymbol{\Theta}^*\boldsymbol{\Psi}^* \neq \bar{\boldsymbol{\Psi}}\boldsymbol{\Theta}\boldsymbol{\Psi}$, the optimal solution to (46) is not separable³. $\square$

Remarks:

If $\kappa = 0$ (*i.e.*, no second denoiser $\bar{\boldsymbol{\Psi}}$ for N interpolated pixels), then $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}$, $\boldsymbol{\Theta}^* = \boldsymbol{\Theta}$, (57) defaults to (45), and the solution becomes separable.

On the other hand, if $\mu = 0$ (*i.e.*, no first denoiser $\boldsymbol{\Psi}$ for M original pixels⁴), then

$$\boldsymbol{\Psi}^* = \left((\gamma + 1)\mathbf{I}_M - \gamma^2\mathbf{A}_{M,N}(\gamma\mathbf{A}_{N,N}^2 + \kappa\bar{\mathbf{L}})^{-1}\mathbf{A}_{M,N}^\top \right)^{-1}. \quad (58)$$

In words, $\boldsymbol{\Psi}^*$ is a function of interpolator $\boldsymbol{\Theta}$ and second denoiser $\bar{\boldsymbol{\Psi}}$; *i.e.*, a new denoiser $\boldsymbol{\Psi}^*$ for M original pixels is created due to the joint optimization (52). The solution remains non-separable.

³Though the solution $\boldsymbol{\Theta}^*\boldsymbol{\Psi}^*$ to (46) is a sequence of two separate matrix operations, each matrix is a non-separable function of input matrices $\{\mathbf{A}, \mathbf{L}, \bar{\mathbf{L}}\}$ or $\{\boldsymbol{\Theta}, \boldsymbol{\Psi}, \bar{\boldsymbol{\Psi}}\}$.

⁴This case was discussed in the conference version of this paper [?].

4.4 Generalization to Multiple Interpolators

One difficulty in using Theorem 2 in practice is that interpolator Θ must be square and full rank. This means that the number of new interpolated pixels is the same as the number of original pixels, and the interpolated pixels must be *linearly independent*.

We generalize (32) to the case where there are multiple full-rank square interpolators for a given signal $\mathbf{x}$.

Consider P interpolators $\{\Theta_p\}_{p=1}^P$, each of size $\mathbb{R}^{M \times M}$, that interpolates M linearly independent pixels from M original ones, where $PM = N$. We can write the general linear interpolation operation as

$$\mathbf{x} = \begin{bmatrix} \mathbf{I}_M \\ \Theta_1 \\ \vdots \\ \Theta_P \end{bmatrix} \mathbf{y}. \quad (59)$$

Next, denote by $\{\mathbf{A}_p\}_{p=1}^P$ the set of adjacency matrices corresponding to interpolators $\{\Theta_p\}_{p=1}^P$, where each $\mathbf{A}_p \in \mathbb{R}^{(N+M) \times (N+M)}$ is defined as

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{0}_{M,M} & \mathbf{0}_{M,N_p^-} & \mathbf{A}_{p,M,N} & \mathbf{0}_{M,N_p^+} \\ \mathbf{0}_{N,M} & \mathbf{0}_{N,N_p^-} & \mathbf{0}_{N,M} & \mathbf{0}_{N,N_p^+} \end{bmatrix}. \quad (60)$$

$N_p^- \triangleq (p-1)M$ and $N_p^+ \triangleq (P-p)M$ are the number of interpolated pixels before and after the p -th interpolator, respectively. Each (typically asymmetric) sub-matrix $\mathbf{A}_{p,M,N} \in \mathbb{R}^{M \times M}$ in $\mathbf{A}_p$ connects directional edges from M original pixels to M new interpolated pixels.

We generalize MAP objective (34) using GSV as prior to

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \gamma \sum_{p=1}^P \|\mathbf{H}(\mathbf{x} - \mathbf{A}_p\mathbf{x})\|_2^2. \quad (61)$$

We now state formally the extension of Theorem 2 to multiple linear interpolators as follows.

Theorem 3. *Interpolator $[\mathbf{I}_M; \Theta_1; \dots; \Theta_P]$ is the solution filter to the MAP problem (61) if $\mathbf{A}_{p,M,N} = \Theta_p^{-1}, \forall p$, given all P interpolators $\{\Theta_p\}_{p=1}^P$ are square and full rank.*

Proof. Taking the derivative of the objective in (61) w.r.t. $\mathbf{x}$ and setting it to 0, we get

$$\left[\mathbf{H}^\top \mathbf{H} + \gamma \sum_{p=1}^P (\mathbf{I}_M - \mathbf{A}_p)^\top \mathbf{H}^\top \mathbf{H} (\mathbf{I}_M - \mathbf{A}_p) \right] \mathbf{x}^* = \mathbf{H}^\top \mathbf{y}. \quad (62)$$

Using the definitions of $\mathbf{H}$ and $\mathbf{A}_p$, the linear system simplifies to $\mathbf{C}\mathbf{x}^* = \mathbf{H}^\top \mathbf{y}$, where coefficient matrix $\mathbf{C}$ is

$$\mathbf{C} = \begin{bmatrix} (1 + \gamma)\mathbf{I}_M & -\gamma\mathbf{A}_{1,M,N} & -\gamma\mathbf{A}_{2,M,N} \dots & -\gamma\mathbf{A}_{P,M,N} \\ -\gamma\mathbf{A}_{1,M,N}^\top & \gamma\mathbf{A}_{1,M,M}^2 & \mathbf{0}_{M,M} \dots & \mathbf{0}_{M,M} \\ \vdots & & \ddots & \vdots \\ -\gamma\mathbf{A}_{P,M,N}^\top & \mathbf{0}_{M,M} & \dots \mathbf{0}_{M,M} & \gamma\mathbf{A}_{P,M,M}^2 \end{bmatrix}. \quad (63)$$

Using again matrix inverse formula (37), and letting sub-block $\dot{\mathbf{D}}$ be the lower-right sub-matrix $\text{diag}(\gamma\mathbf{A}_{1,M,M}^2, \dots, \gamma\mathbf{A}_{P,M,M}^2)$ above, we first compute block $\dot{\mathbf{P}} = (\dot{\mathbf{A}} - \dot{\mathbf{B}}\dot{\mathbf{D}}^{-1}\dot{\mathbf{C}})^{-1}$:

$$\begin{aligned} \dot{\mathbf{P}} &= ((1 + \gamma)\mathbf{I}_M - \text{vec}^\top(\gamma\mathbf{A}_{p,M,N}\}) \text{diag}^{-1}(\{\gamma\mathbf{A}_{p,M,M}^2\}) \\ &\quad \text{vec}(\{-\gamma\mathbf{A}_{p,M,N}^\top\})^{-1} \\ &= ((1 + \gamma)\mathbf{I}_M - \gamma \text{diag}(\{\mathbf{A}_{p,M,N}(\mathbf{A}_{p,M,M}^2)^{-1}\mathbf{A}_{p,M,N}^\top\}))^{-1} \\ &\stackrel{(a)}{=} ((1 + \gamma)\mathbf{I}_M - \gamma\mathbf{I}_M)^{-1} = \mathbf{I}_M, \end{aligned} \quad (64)$$

where $\text{vec}(\{v_i\})$ denotes a column vector with $\{v_i\}$ as entries. In (a), we apply the assumption that $\mathbf{A}_{p,M,N} = (\boldsymbol{\Theta}_p)^{-1}, \forall p$.

Using (37) and $\dot{\mathbf{P}} = \mathbf{I}_M$, we can now write $\mathbf{x}^*$ as

$$\begin{aligned} \mathbf{x}^* &= \mathbf{C}^{-1}\mathbf{H}^\top \mathbf{y} \\ &= \begin{bmatrix} \mathbf{I}_M \\ -\text{diag}^{-1}(\gamma\mathbf{A}_{p,M,M}^2) \text{vec}(\{-\gamma\mathbf{A}_{p,M,N}^\top\}) \end{bmatrix} \mathbf{y} \\ &= \begin{bmatrix} \mathbf{I}_M \\ (\mathbf{A}_{1,M,M}^2)^{-1}\mathbf{A}_{1,M,N}^\top \\ \vdots \\ (\mathbf{A}_{P,M,M}^2)^{-1}\mathbf{A}_{P,M,N}^\top \end{bmatrix} \mathbf{y} \stackrel{(a)}{=} \begin{bmatrix} \mathbf{I}_M \\ \boldsymbol{\Theta}_1 \\ \vdots \\ \boldsymbol{\Theta}_P \end{bmatrix} \mathbf{y}, \end{aligned} \quad (65)$$

where in (a) we apply the assumption that $\mathbf{A}_{p,M,N} = \boldsymbol{\Theta}_p^{-1}, \forall p$. □

4.4.1 Computation of Joint Denoising / Multi-Interpolation

The joint denoising / interpolation formulation (46) generalizes for the multiple-interpolator case to

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{y}^* - \mathbf{H}\mathbf{x}\|_2^2 + \gamma \sum_{p=1}^P \|\mathbf{H}(\mathbf{x} - \mathbf{A}_p\mathbf{x})\|_2^2 + \mu(\mathbf{H}\mathbf{x})^\top \mathbf{L}(\mathbf{H}\mathbf{x}) \\ + \kappa(\mathbf{G}\mathbf{x})^\top \bar{\mathbf{L}}(\mathbf{G}\mathbf{x}). \end{aligned} \quad (66)$$

We show how (66) can also be solved efficiently. Generalizing from (49), solution $\mathbf{x}^*$ to (66) can be obtained by solving a linear system $\mathbf{C}\mathbf{x}^* = \mathbf{H}^\top \mathbf{y}$, where the coefficient matrix $\mathbf{C}$ is

$$\mathbf{C} = \left[\begin{array}{c|c} (\gamma + 1)\mathbf{I}_M + \mu\mathbf{L} & -\gamma\mathbf{A}_{1,M,N} \dots - \gamma\mathbf{A}_{P,M,N} \\ \hline -\gamma\mathbf{A}_{1,M,N}^\top & \\ \vdots & \\ -\gamma\mathbf{A}_{P,M,N}^\top & \mathbf{D} + \kappa\bar{\mathbf{L}} \end{array} \right]$$

$$\mathbf{D} = \gamma \text{diag}(\mathbf{A}_{1,N,N}^2, \dots, \mathbf{A}_{P,N,N}^2). \quad (67)$$

As done previously, linear system $\mathbf{C}\mathbf{x}^* = \mathbf{H}^\top \mathbf{y}$ can be solved efficiently via CG.

4.4.2 Application of Theorem 3 for Rectangular Θ

When given an interpolator matrix Θ that is rectangular, we use Theorem 3 as follows. If $\Theta \in \mathbb{R}^{H \times N}$ is a *long* matrix (assuming the rows are independent) for $H < N$, then we simply add $N - H$ linearly independent *dummy* row to Θ so that the resulting interpolator is square.

On the other hand, if $\Theta \in \mathbb{R}^{H \times N}$ is a *tall* matrix (assuming the columns are independent) for $H > N$, then we first divide Θ into $P = \lfloor \frac{H}{N} \rfloor$ square interpolators Θ_p , each of size $N \times N$. Then, to the remaining $H - PN < N$ rows in Θ , we add $N - H - PN$ dummy rows to compose Θ_{P+1} ; *i.e.*,

$$\Theta' = \begin{bmatrix} \Theta_1 \\ \vdots \\ \Theta_P \\ \Theta_{P+1} \end{bmatrix} \quad (68)$$

We can now treat modified Θ' as a set of $P + 1$ interpolators $\{\Theta_i\}_{i=1}^{P+1}$, each of size $N \times N$.

5 Experiments and Results

5.1 Experimental Setup

Experiments were conducted to test our derived operators in (51) for joint denoising / interpolation (denoted by `joint`) compared to original sequential operations (denoted by `sequential`) in the non-separable case.

For denoisers, we employed Gaussian filter [82], bilateral filter (BF) [26], and nonlocal means (NLM) [37,86]. For interpolators, we used linear operators for image rotation, warping using homography transform, and demosaicking operator with bilinear interpolation. The output images were evaluated using signal-to-noise ratio (PSNR).

First, we discuss the experiments involving interpolators for rotation and warping. Popular 512×512 grayscale images, `Lena` and `peppers`, were used for rotation and warping, respectively. The experiments were run in Matlab R2022a⁵. Gaussian noise of different variances were added to the images.

The joint denoising / interpolation operation was performed on 10×10 output patches. The size and location of the input patch from which the output patch is generated depend on the interpolation operation and the output location.

To ensure that the number of input pixels is equal to the output pixels (*i.e.*, $M = N$), we interpolated dummy pixels by adding linearly independent rows to Θ . It is important to ensure that Θ is full rank (thus invertible) when adding new rows.

To ensure denoiser Ψ is non-expansive, symmetric and PD according to Theorem 1, we ran the Sinkhorn-Knopp procedure [87] for an input linear denoiser with non-negative entries and independent rows, so that matrix Ψ was *double stochastic*, and thus its eigenvalues satisfied $|\lambda_i| \leq 1, \forall i$. Note that for a chosen interpolation operation, Θ changed from patch to patch, and so the denoiser needed to adapt to the input and output dimensions when using the Gaussian filter. For BF, and NLM, the denoiser itself changed from patch to patch. To solve (47) in each iteration, `pcg` function in MATLAB implementing a version of conjugate gradient [88] was used.

⁵The code developed for these experiments are made available at <https://github.com/sybernix/tip-joint-experiments>

For experiments in Section 5.2.1 to test Corollary 2, the following settings were used. Image rotation was performed at 20 degrees anti-clockwise, and for image warping a homography matrix of $[1, 0.2, 0; 0.1, 1, 0; 0, 0, 1]$ was used. For the experiments with Gaussian filter and BF, the hyperparameters were selected as $\mu = 0.3$, $\gamma = 0.5$, $\kappa = 0.3$. For Gaussian denoiser, a variance of 0.3 was used, and for BF, variance of 0.3 was used for both spatial and range kernels. For experiments with NLM, $\gamma = 0.6$, $\kappa = 0.2$, while μ was kept the same. The patch size for NLM was 3×3 and the search window size was 9×9 .

For experiments in Section 5.2.2 to test Corollary 3, the following settings were used. Image rotation was performed at 20 degrees anti-clockwise, and for image warping a homography matrix of $[1, 0.2, 0; 0.1, 1, 0; 0, 0, 1]$ was used. For the experiments with Gaussian filter and BF, the hyperparameters were selected as $\mu = 0.3$, $\gamma = 0.5$, $\kappa = 0.3$. For Gaussian denoiser, σ was set to 0.3, and for Bilateral filter, both spatial and range kernel parameters σ_s , σ_r were set to 0.3. For experiments with NLM, $\gamma = 0.6$, $\kappa = 0.2$, while μ was kept the same.

For image demosaicking experiment in Section 5.2.3 to test Theorem 3, the McMaster dataset [89] was used. Code was developed in Python and shared in the same Github repository. The formulation in the equation (66) was used to combine multiple interpolators and denoisers to achieve demosaicking with denoising of both CFA grid and the output channel. For the purpose of testing, the blue channel was interpolated from the CFA image. For red and green channels, the procedure is similar with minor modifications to the interpolator.

The hyperparameters used were $\gamma = 0.2$, $\kappa = 0.1$, $\mu = 0.3$. Another parameter $\epsilon = 0.035$ was used to regularize Θ_p to ensure non-singular matrices as follows:

$$\Theta_p = \Theta_p + \epsilon \mathbf{I}. \quad (69)$$

Patches of size 8×8 were extracted from the CFA. One out of four pixels in CFA is a blue pixel; see Figure 4 for an illustration. This means that 16 pixels out of the 64 pixels in a 8×8 patch correspond to the blue channel. From the 8×8 CFA patch, we can select the 16 blue pixels, using which we need to interpolate the remaining $64 - 16 = 48$ blue pixels. Together with the original 16 blue pixels, the output image will be of same size as the input CFA, *i.e.*, 64. Hence, the interpolator Θ is of size 64×16 .

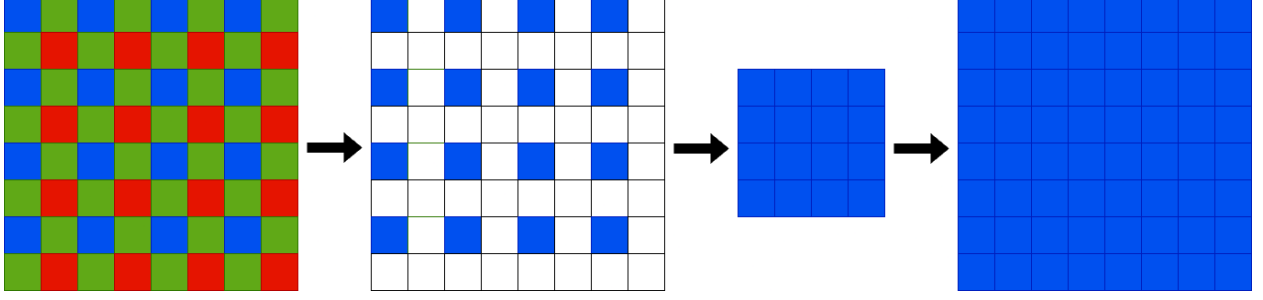


Figure 4: Interpolation of blue channel from color filter array (CFA) grid. In the first step we are extracting the blue pixels from an 8×8 bayer grid. In the second step, the extracted pixels are rearranged into 4×4 patch. In the third step final 8×8 blue channel output is interpolated from the 4×4 blue patch.

We slice the tall matrix into four 16×16 Θ_p . Consider the following equation which shows the interpolation of 8×8 blue channel (flattened as $\mathbf{x}_{blue\ channel}^{64 \times 1}$) where $\mathbf{y}_{blue\ patch}^{16 \times 1}$ is the flattened 4×4 blue patch from CFA

$$\mathbf{x}_{blue\ channel}^{64 \times 1} = \Theta^{64 \times 16} \cdot \mathbf{y}_{blue\ patch}^{16 \times 1}. \quad (70)$$

To slice $\Theta^{64 \times 16}$ into four submatrices Θ_p (for $p = 1, 2, 3, 4$), each of size 16×16 , we first partition the 64 rows of Θ into four groups of 16 rows each. The matrix Θ can be written as

$$\Theta^{64 \times 16} = \begin{bmatrix} \Theta_1^{16 \times 16} \\ \Theta_2^{16 \times 16} \\ \Theta_3^{16 \times 16} \\ \Theta_4^{16 \times 16} \end{bmatrix} \quad (71)$$

where each Θ_p (for $p = 1, 2, 3, 4$) is a submatrix of size 16×16 . In the context of the original equation (70), each Θ_p can be used to compute a portion of $\mathbf{x}_{64 \times 1}$. Since $\mathbf{x}_{64 \times 1}$ is the result of $\Theta_{64 \times 16} \cdot \mathbf{y}_{16 \times 1}$, we can write:

$$\mathbf{x}^{64 \times 1} = \begin{bmatrix} \mathbf{x}_1^{16 \times 1} \\ \mathbf{x}_2^{16 \times 1} \\ \mathbf{x}_3^{16 \times 1} \\ \mathbf{x}_4^{16 \times 1} \end{bmatrix}. \quad (72)$$

where each $\mathbf{x}_p$ (for $p = 1, 2, 3, 4$) is a 16×1 vector, computed as:

$$\mathbf{x}_p^{16 \times 1} = \Theta_p^{16 \times 16} \cdot \mathbf{y}^{16 \times 1} \quad (73)$$

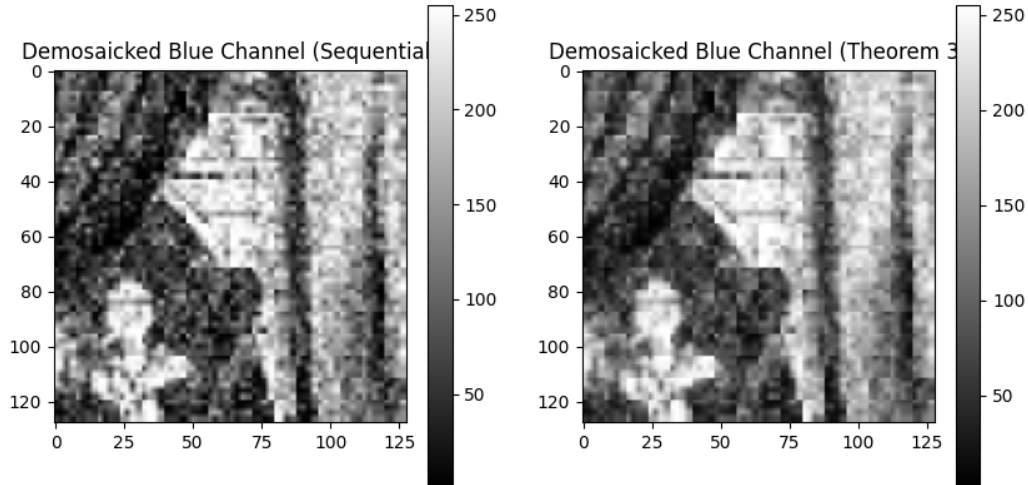


Figure 5: Output of demosaicking with `sequential` and `joint` processes.

The solution to system $\mathbf{C}\mathbf{x}^* = \mathbf{H}^\top \mathbf{y}$, where $\mathbf{C}$ is as defined in (67), is calculated using conjugate gradient with tolerance of 10^{-3} and a maximum of 3000 iterations.

5.2 Experimental Results

5.2.1 Experiments for Corollary 2

Figure 6 show the performance of joint and separate processing of image rotation (interpolation) and bilateral denoiser (denoising) applied to output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. We observe a maximum PSNR gain of 1.35dB for the derived joint operator in (51) over separate operators in this setting.

Figure 7 show the performance of joint and separate processing of image rotation (interpolation) and non-local means denoiser (denoising) applied to output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. We observe a maximum PSNR gain of 0.77dB for the derived joint operator in (51) over separate operators in this setting.

Figure 8 show the performance of joint and separate processing of image warping (inter-

polution) and bilateral denoiser (denoising) applied to output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. We observe a maximum PSNR gain of 1.05dB for the derived joint operator in (51) over separate operators in this setting.

Figure 9 show the performance of joint and separate processing of image warping (interpolation) and Gaussian denoiser (denoising) applied to output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. We observe a maximum PSNR gain of 1.14dB for the derived joint operator in (51) over separate operators in this setting.

Overall, the plots show that **joint** (derived joint operator in (51)) performed better than **sequential** (separate operators) in general. While the performance of both schemes degraded as noise variance increased, the performance of **sequential** degraded faster than **joint**.

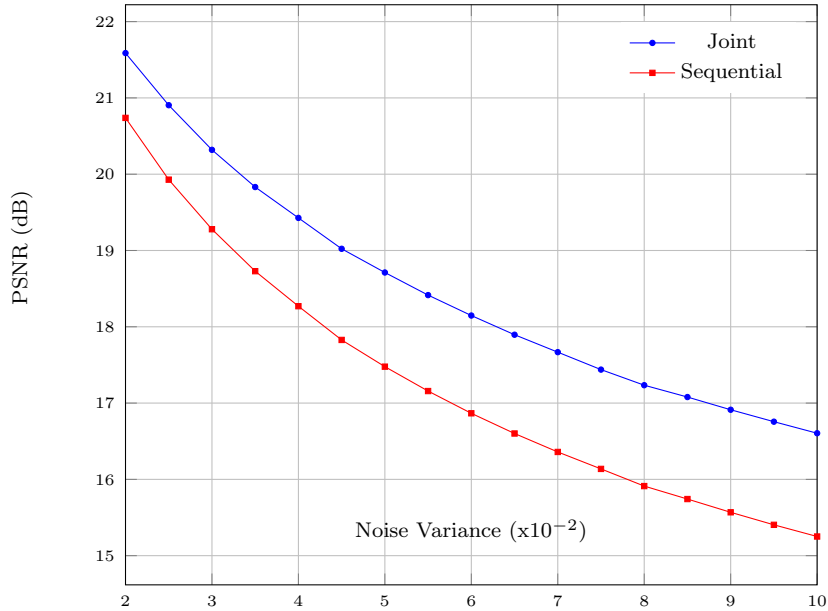


Figure 6: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image rotation and bilateral denoiser on output pixels.

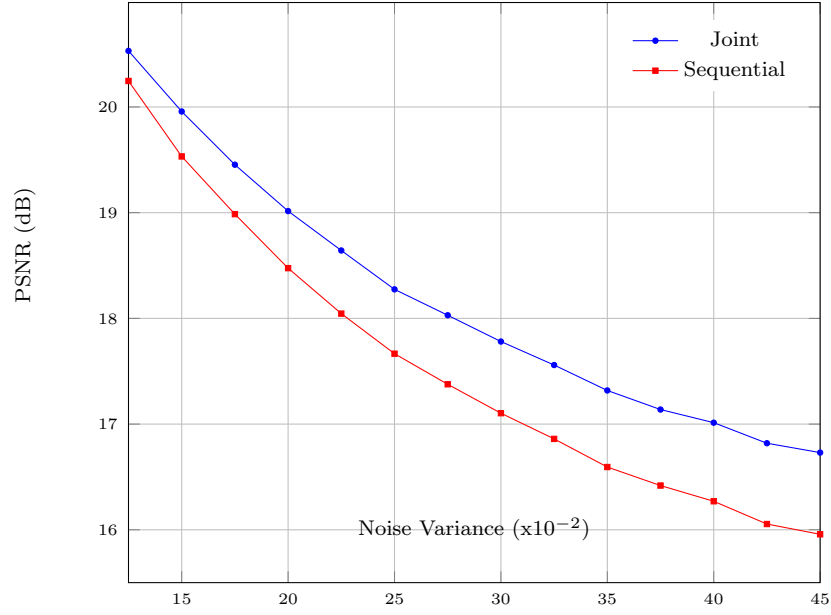


Figure 7: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image rotation and Non-local Means denoiser on output pixels.

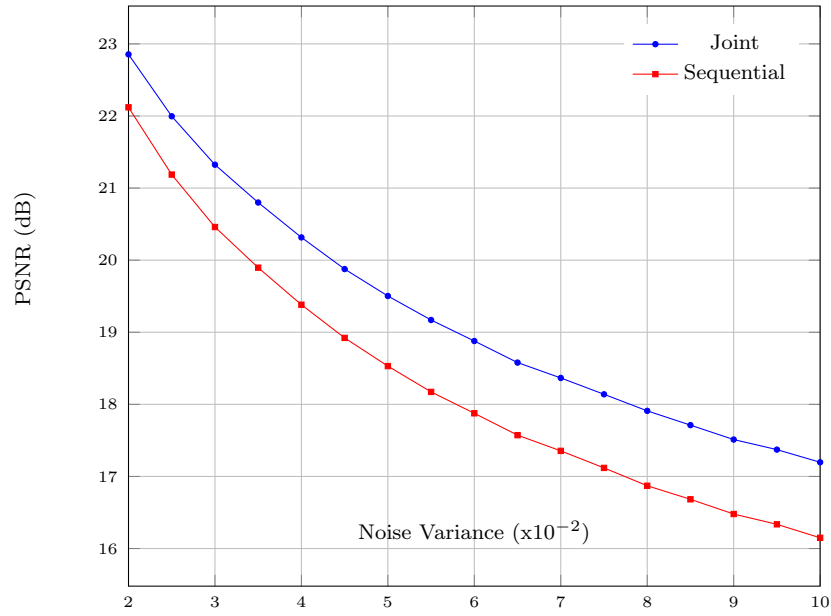


Figure 8: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image warping and Bilateral denoiser on output pixels.

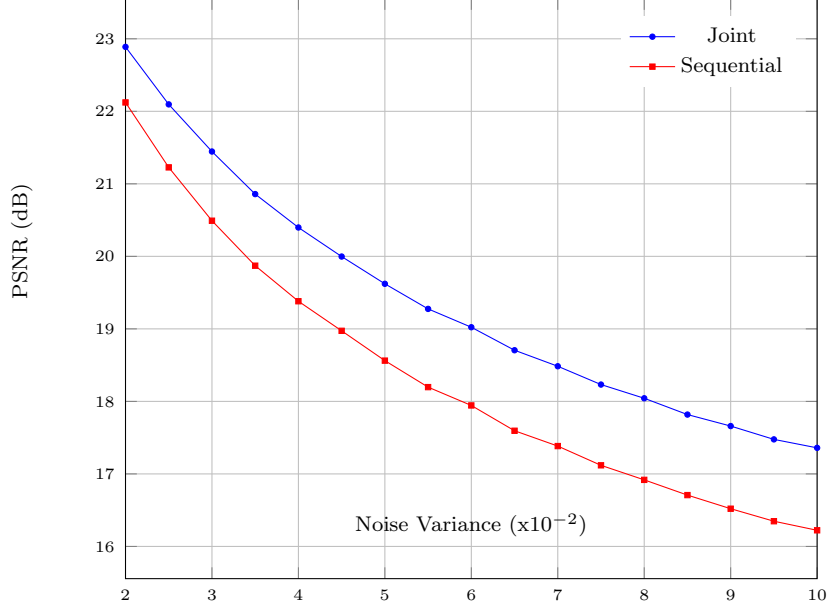


Figure 9: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image warping and Gaussian denoiser on output pixels.

5.2.2 Experiments for Corollary 3

Figure 10 shows the performance of the experiment where interpolator was image rotation and denoiser was bilateral filter. The denoiser was applied twice—both to the input and and the output pixels using the formulation in (53). PSNR of **joint** and **sequential** were evaluated for iid Gaussian noise variance from 0.01 to 0.05. The maximum PSNR gain for **joint** (joint operator in (57)) over **sequential** was at the maximum noise variance tested; gain of 1.32 dB at 0.05 noise variance. We also noted structural similarity index measure (SSIM) gain when using **joint** over **sequential**. SSIM gain of 0.02 was observed at noise level of 0.04.

Figure 11 show the performance of joint and separate processing of image rotation (interpolation) and Gaussian denoiser (denoising) applied to both input and output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. PSNR gain of **joint** (joint operator in (57)) increases over **sequential** as noise variance increases from 0.01 to 0.05 reaching maximum gain of 1.82 dB at 0.05. SSIM gain of 0.06 was observed at noise level of 0.015.

Similar to the experiments with image rotation, image warping was tested with Gaussian

filter and bilateral filter as the two denoisers. Figure 13 show the performance of joint and separate processing of image warping (interpolation) and Gaussian denoiser (denoising) applied to both input and output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. The maximum PSNR gain 1.46 dB for `joint` (joint operator in (57)) was observed at noise variance of 0.01 which. However, the gain at maximum noise was 1.38 dB, which is still substantial compared to the gain at lowest noise setting. SSIM gain of 0.017 was observed at noise level of 0.04.

Figure 12 show the performance of joint and separate processing of image warping (interpolation) and Bilateral filter (denoising) applied to both input and output pixels, given that the input image patch was corrupted by iid Gaussian noise of different variances. The performance is shown in PSNR. The plots show trends similar to that observed with image rotation experiments. PSNR gain for `joint` (joint operator in (57)) increases with image noise to a maximum of 1.07 dB at 0.05.

Overall, the plots show that similar to the experiments in Section 5.2.1, `joint` schemes performed better than `sequential` in general, and while the performance of both schemes degraded as noise variance increased, the performance of `sequential` degraded faster than `joint`.

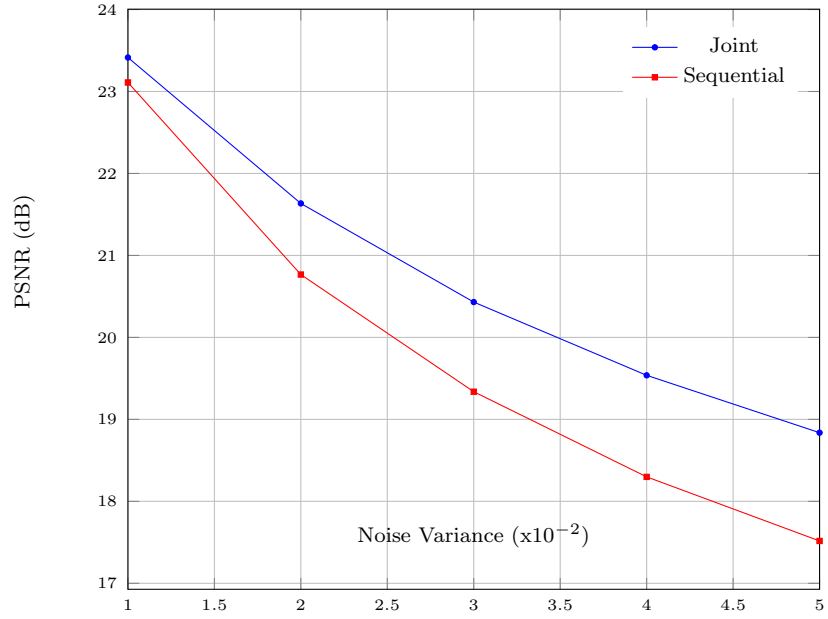


Figure 10: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image rotation and a bilateral denoiser.

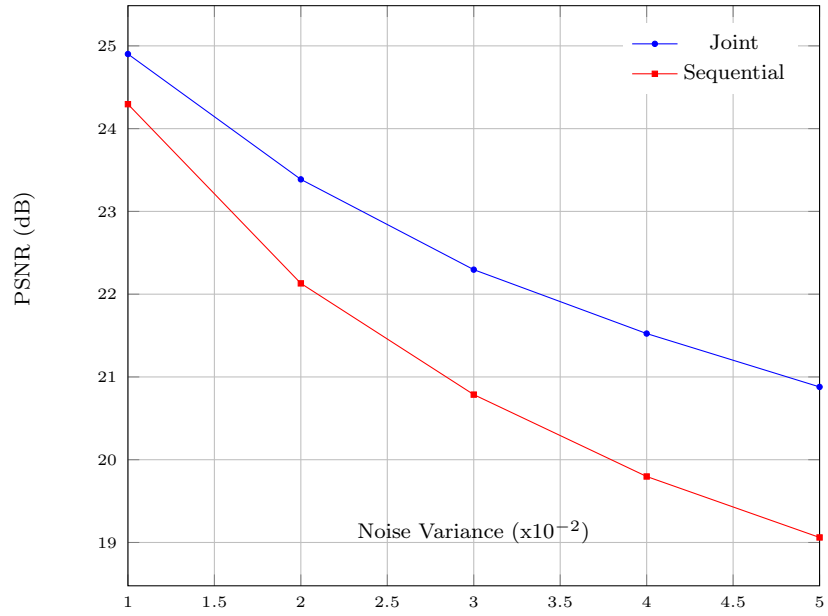


Figure 11: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image rotation and a Gaussian denoiser.

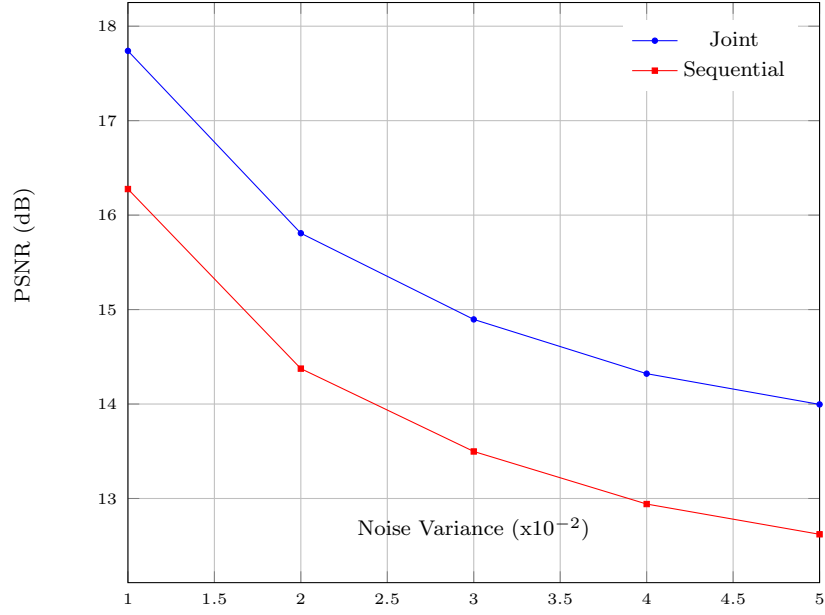


Figure 12: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image warping and Gaussian denoiser.

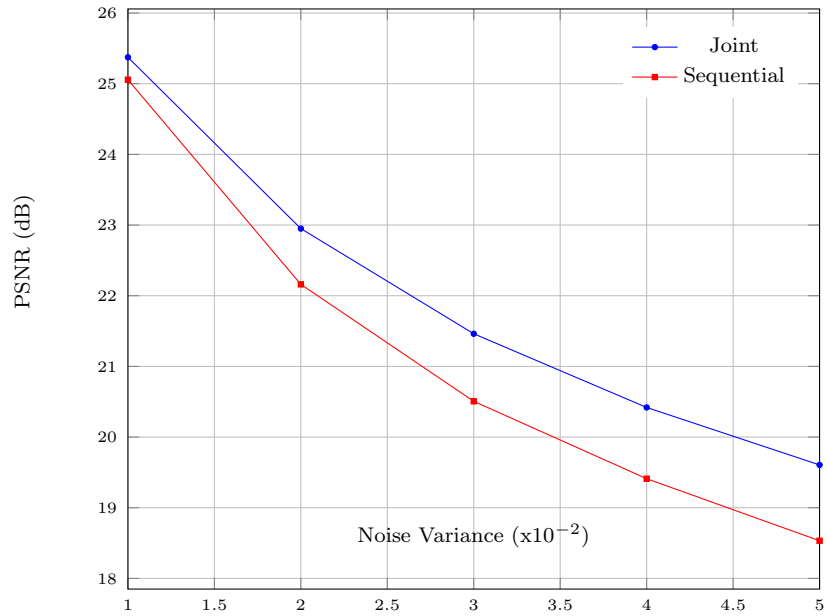


Figure 13: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using image warping and bilateral denoiser.

5.2.3 Experiments for Theorem 3

Figure 14 shows the performance for the experiments designed to test Theorem 3 with multiple interpolators. Demosaicking was the interpolation operation and Gaussian denoiser was applied on input and output pixels. General trend of **joint** showing increasing PSNR gain over **sequential** was exhibited over a noise range 0.05 to 0.45. Maximum PSNR gain 0.79 dB was observed at variance 0.45.

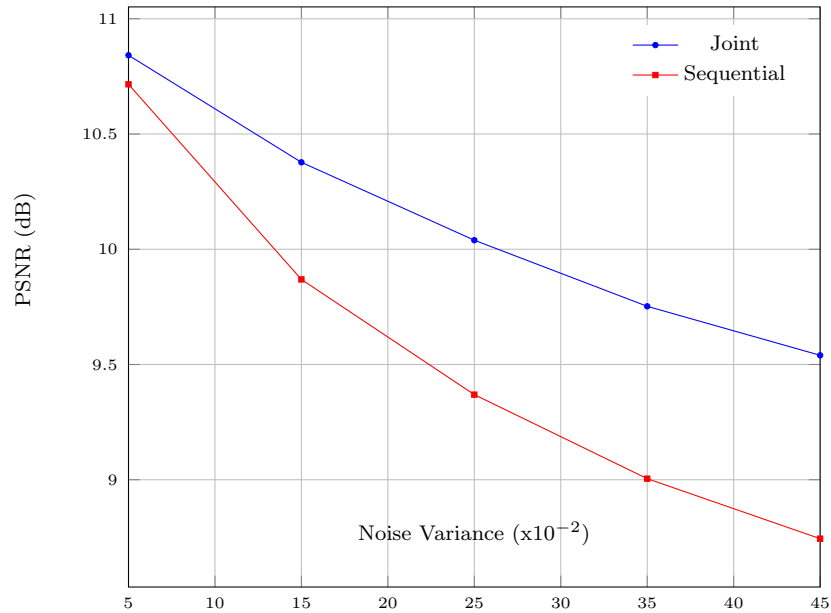


Figure 14: Performance comparison of **joint** vs **sequential** processing over a range of noise variances using demosaicking and a Gaussian denoiser.

6 Conclusion and Future Direction

We presented two theorems, under mild conditions, connecting any linear denoiser / interpolator to optimal graph filter using undirected / directed graph smoothness priors, respectively. Specifically, Theorem 1 connects any denoiser to a solution filter to GLR-regularized MAP optimization problem, given that the denoiser is non-expansive, symmetric, and PD. Theorem 2 connects any interpolator to a solution filter to a GSV-regularized MAP optimization problem, given that the interpolator is square and invertible.

Corollary 1 utilizes results from Theorems 1 and 2 to show that the joint denoising and interpolation problem has a separable optimal solution, if the denoiser is applied exclusively on input pixels; *i.e.*, the denoiser and interpolator can be sequentially applied for an optimal solution. Corollary 2 shows that the joint optimization is non-separable if the denoiser is exclusively applied to interpolated pixels; *i.e.*, for optimal output we cannot apply interpolator and denoiser one after the other. Corollary 3 shows that when denoising is applied to both input pixels as well as output pixels, the optimal solution is again non-separable.

Finally, we generalize Theorem 2 to Theorem 3 to handle multiple interpolators. This is useful in scenarios where the number of output pixels are more than the input pixels, *i.e.*, the interpolator matrix Θ is a tall matrix. Experimental results confirm our hypothesis, and the joint optimization framework proposed in this thesis outperforms sequential processing considerably.

Further studies can be conducted on how pseudo-linear filters learned from data [90] can be used in the proposed joint framework. We can also study how to reduce the computational complexity of certain steps in the algorithms used in our work. Specifically, to compute the corresponding Laplacian matrix $\mathbf{L}$ and adjacency matrix $\mathbf{A}$, we need to invert denoiser matrix Ψ and interpolator matrix Θ . Inverting a general matrix is an operation of order $\mathcal{O}(N^3)$ [91]. By utilizing Taylor series expansion [91, 92] or CLIME [93] we can potentially bring down the complexity of these operations.

References

- [1] B. E. Bayer, “Color imaging array,” July 20 1976. US Patent 3,971,065.
- [2] A. Chambolle and P.-L. Lions, “Image recovery via total variation minimization and related problems,” *Numerische Mathematik*, vol. 76, pp. 167–188, 1997.
- [3] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [4] F. Luisier, T. Blu, and M. Unser, “Image denoising in mixed poisson–gaussian noise,” *IEEE Transactions on image processing*, vol. 20, no. 3, pp. 696–708, 2010.
- [5] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” *IEEE Transactions on image processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [6] C. Boncelet, “Image noise models,” in *The essential guide to image processing*, pp. 143–167, Elsevier, 2009.
- [7] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: A systematic survey,” in *Visual Communications and Image Processing 2008*, vol. 6822, pp. 489–503, SPIE, 2008.
- [8] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.
- [9] J. A. Stark, “Adaptive image contrast enhancement using generalizations of histogram equalization,” *IEEE Transactions on image processing*, vol. 9, no. 5, pp. 889–896, 2000.
- [10] Y.-C. Liu, W.-H. Chan, and Y.-Q. Chen, “Automatic white balance for digital still camera,” *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 460–466, 1995.
- [11] C. Poynton, *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.
- [12] G. D. Finlayson, M. S. Drew, and B. V. Funt, “Color constancy: generalized diagonal transforms suffice,” *JOSA A*, vol. 11, no. 11, pp. 3011–3019, 1994.

- [13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [14] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International journal of computer vision*, vol. 47, pp. 7–42, 2002.
- [15] A. Ortega, P. Frossard, J. Kovavcević, J. M. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [16] G. Cheung and E. Magli, *Graph spectral image processing*. John Wiley & Sons, 2021.
- [17] W. Hu, G. Cheung, A. Ortega, and O. C. Au, “Multiresolution graph fourier transform for compression of piecewise smooth images,” *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, 2014.
- [18] J. Pang, G. Cheung, A. Ortega, and O. C. Au, “Optimal graph laplacian regularization for natural image denoising,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2294–2298, IEEE, 2015.
- [19] Y. Bai, G. Cheung, X. Liu, and W. Gao, “Graph-based blind image deblurring from a single photograph,” *IEEE transactions on image processing*, vol. 28, no. 3, pp. 1404–1418, 2018.
- [20] X. Liu, G. Cheung, X. Ji, D. Zhao, and W. Gao, “Graph-based joint dequantization and contrast enhancement of poorly lit jpeg images,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1205–1219, 2018.
- [21] C. Dinesh, G. Cheung, and I. V. Bajić, “Point cloud denoising via feature graph laplacian regularization,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4143–4158, 2020.
- [22] J. Pang and G. Cheung, “Graph Laplacian regularization for inverse imaging: Analysis in the continuous domain,” in *IEEE Transactions on Image Processing*, vol. 26, no.4, pp. 1770–1785, April 2017.

- [23] Y. Romano, M. Elad, and P. Milanfar, “The little engine that could: Regularization by denoising (red),” in *SIAM Journal on Imaging Sciences*, vol. 10, no.4, pp. 1804–1844, 2017.
- [24] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [25] T. Huang, G. Yang, and G. Tang, “A fast two-dimensional median filtering algorithm,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 27, no. 1, pp. 13–18, 1979.
- [26] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pp. 839–846, IEEE, 1998.
- [27] S. Geman, E. Bienenstock, and R. Doursat, “Neural networks and the bias/variance dilemma,” *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.
- [28] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [29] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [30] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44, IEEE, 1993.
- [31] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.
- [32] K. Bredies and D. A. Lorenz, “Linear convergence of iterative soft-thresholding,” *Journal of Fourier Analysis and Applications*, vol. 14, pp. 813–837, 2008.

- [33] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [34] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [35] F. Chen, G. Cheung, and X. Zhang, “Fast & robust image interpolation using gradient graph laplacian regularizer,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1964–1968, IEEE, 2021.
- [36] K. Bredies, K. Kunisch, and T. Pock, “Total generalized variation,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.
- [37] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 2, pp. 60–65, Ieee, 2005.
- [38] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [39] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [40] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.
- [41] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

- [42] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.
- [43] J. Pang and G. Cheung, “Graph laplacian regularization for image denoising: Analysis in the continuous domain,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1770–1785, 2017.
- [44] J. Zeng, J. Pang, W. Sun, and G. Cheung, “Deep graph laplacian regularization for robust denoising of real images,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.
- [45] H. Vu, G. Cheung, and Y. C. Eldar, “Unrolling of deep graph total variation for image denoising,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2050–2054, IEEE, 2021.
- [46] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot, “Dual constrained tv-based regularization on graphs,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1246–1273, 2013.
- [47] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery, *Numerical recipes in C++ the art of scientific computing*. Cambridge university press, 2001.
- [48] R. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.
- [49] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing. i. theory,” *IEEE transactions on signal processing*, vol. 41, no. 2, pp. 821–833, 1993.
- [50] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing. ii. efficiency design and applications,” *IEEE transactions on signal processing*, vol. 41, no. 2, pp. 834–848, 1993.
- [51] C. E. Duchon, “Lanczos filtering in one and two dimensions,” *Journal of Applied Meteorology and Climatology*, vol. 18, no. 8, pp. 1016–1022, 1979.

- [52] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.
- [53] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.
- [54] W. Yu, “Colour demosaicking method using adaptive cubic convolution interpolation with sequential averaging,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 153, no. 5, pp. 666–676, 2006.
- [55] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: color filter array interpolation,” *IEEE Signal processing magazine*, vol. 22, no. 1, pp. 44–54, 2005.
- [56] R. C. Gonzales, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Pearson Prentice Hall, 2004.
- [57] M. Unser, “Splines: A perfect fit for signal and image processing,” *IEEE Signal processing magazine*, vol. 16, no. 6, pp. 22–38, 1999.
- [58] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” *Ieee transactions on image processing*, vol. 14, no. 3, pp. 360–369, 2005.
- [59] L. Condat and S. Mosaddegh, “Joint demosaicking and denoising by total variation minimization,” in *2012 19th IEEE International Conference on Image Processing*, pp. 2781–2784, IEEE, 2012.
- [60] H. S. Malvar, L.-w. He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *2004 IEEE international conference on acoustics, speech, and signal processing*, vol. 3, pp. iii–485, IEEE, 2004.
- [61] F. Kokkinos and S. Lefkimmiatis, “Deep image demosaicking using a cascade of convolutional residual denoising networks,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 303–319, 2018.

- [62] R. Tan, K. Zhang, W. Zuo, and L. Zhang, “Color image demosaicking via deep residual learning,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, vol. 2, p. 6, 2017.
- [63] K. Hirakawa and T. W. Parks, “Joint demosaicing and denoising,” *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2146–2157, 2006.
- [64] T. Klatzer, K. Hammernik, P. Knobelreiter, and T. Pock, “Learning joint demosaicing and denoising based on sequential energy minimization,” in *2016 IEEE International Conference on Computational Photography (ICCP)*, pp. 1–11, IEEE, 2016.
- [65] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, “Deep joint demosaicking and denoising,” *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.
- [66] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [67] W. Dong, M. Yuan, X. Li, and G. Shi, “Joint demosaicing and denoising with perceptual optimization on a generative adversarial network,” *arXiv preprint arXiv:1802.04723*, 2018.
- [68] J. Xie, R. S. Feris, S.-S. Yu, and M.-T. Sun, “Joint super resolution and denoising from a single depth image,” *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1525–1537, 2015.
- [69] W. Hu, G. Cheung, and A. Ortega, “Intra-prediction and generalized graph fourier transform for image coding,” *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1913–1917, 2015.
- [70] X. Liu, G. Cheung, X. Wu, and D. Zhao, “Random walk graph laplacian-based smoothness prior for soft decoding of jpeg images,” *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 509–524, 2016.
- [71] H. E. Egilmez, E. Pavez, and A. Ortega, “Graph learning from data under laplacian and structural constraints,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, 2017.

- [72] S. Bagheri, G. Cheung, A. Ortega, and F. Wang, “Learning sparse graph laplacian with k eigenvector prior via iterative glasso and projection,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5365–5369, IEEE, 2021.
- [73] W. Hu, X. Gao, G. Cheung, and Z. Guo, “Feature graph learning for 3d point cloud denoising,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2841–2856, 2020.
- [74] C. Yang, G. Cheung, and W. Hu, “Signed graph metric learning via gershgorin disc perfect alignment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7219–7234, 2021.
- [75] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, “Graph spectral image processing,” in *Proceedings of the IEEE*, vol. 106, no.5, pp. 907–930, May 2018.
- [76] S. Axler, *Linear algebra done right*. Springer Nature, 2024.
- [77] X. Liu, G. Cheung, X. Wu, and D. Zhao, “Random walk graph laplacian based smoothness prior for soft decoding of JPEG images,” in *IEEE Transactions on Image Processing*, vol. 26, no.2, pp. 509–524, February 2017.
- [78] A. Elmoataz, O. Lezoray, and S. Bougleux, “Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing,” *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.
- [79] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot, “Dual constrained TV-based regularization on graphs,” in *SIAM Journal on Imaging Sciences*, vol. 6, no.4, p. 1246–1273, 2013.
- [80] F. Chen, G. Cheung, and X. Zhang, “Fast and robust image interpolation using gradient graph laplacian regularizer,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1964–1968, 2021.
- [81] S. H. Chan, “Performance analysis of plug-and-play ADMM: A graph signal processing perspective,” *IEEE Transactions on Computational Imaging*, vol. 5, no. 2, pp. 274–286, 2019.

- [82] P. Milanfar, “A tour of modern image filtering: New insights and methods, both practical and theoretical,” *IEEE signal processing magazine*, vol. 30, no. 1, pp. 106–128, 2012.
- [83] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [84] N. Viswarupan, G. Cheung, F. Lan, and M. S. Brown, “Mixed graph signal analysis of joint image denoising/interpolation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9431–9435, IEEE, 2024.
- [85] D. S. Bernstein, *Matrix mathematics: theory, facts, and formulas*. Princeton University Press, 2009.
- [86] A. Buades, B. Coll, and J.-M. Morel, “Non-local means denoising,” *Image Processing On Line*, vol. 1, pp. 208–212, 2011.
- [87] P. A. Knight, “The sinkhorn–knopp algorithm: convergence and applications,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.
- [88] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [89] L. Zhang, X. Wu, A. Buades, and X. Li, “Color demosaicking by local directional interpolation and nonlocal adaptive thresholding,” *Journal of Electronic imaging*, vol. 20, no. 2, pp. 023016–023016, 2011.
- [90] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, “Burst denoising with kernel prediction networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2502–2510, 2018.
- [91] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
- [92] J. Stewart, *Calculus: early transcendentals*. Cengage learning, 2012.

- [93] T. Cai, W. Liu, and X. Luo, “A constrained ℓ_1 minimization approach to sparse precision matrix estimation,” *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 594–607, 2011.