

IMAGE SUPER RESOLUTION VIA NON-LOCAL NORMALIZED GRAPH LAPLACIAN REGULARIZATION

by

Cory Falconer

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science

in

The Faculty of Science

Modeling and Computational Science

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

May 2019

© Cory Falconer, 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Cory Falconer**

Master of Science in Modelling and Computational Science

Thesis title: Image Super Resolution via Non-Local Normalized Graph Laplacian Regularization

An oral defense of this thesis took place on March 26, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee Dr. Lenraert van Veen

Research Supervisor Dr. Mehran Ebrahimi

Research Co-supervisor Dr. Sean Bohun

Examining Committee Member Dr. Mark Green

External Examiner Dr. Ken Pu

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Given we live in a digital age where images are regularly being viewed, posted, or utilized, spectators of such images on occasion could prefer a higher resolution perspective. The process of producing a high-resolution image given a single low-resolution noisy measurement is called single-frame image super resolution (SISR). Many interpolation schemes fail to preserve important edge information of images and cannot be used blindly for resolution enhancement. In general, *a priori* constraints can be imposed on the high resolution image approximation. This process is called regularization. We model our SISR problem as an energy minimization procedure which optimally balances data fidelity and the regularization term. The regularization term will incorporate natural image redundancy implicitly via the so called normalized graph Laplacian operator. This operator applies a non-local kernel similarity measure due to choice of a non-local operator for weight assignment. The data fidelity term is modeled as a likelihood estimator that is scaled using a sharpening term composed from the normalized graph Laplacian operator. Finally, a conjugate gradient scheme is used to minimize the objective functional. Promising results on resolution enhancement for a variety of digital images will be presented. Non-local methods can be further enhanced following a “boosting” procedure deemed to enhance a signal by reintroducing a “cleaned“ version of the signal back into the final approximation. This is beneficial for all non-local restoration approaches. We show analytically that successive applications of this boosting operation does not necessarily guarantee a superior final solution.

Acknowledgements

Firstly, I would like to thank my supervisors Dr. C. Sean Bohun and Dr. Mehran Ebrahimi for their guidance and support throughout this project. Secondly, I would like to thank all members of the Modelling and Computational Science program, in particular my fellow companions of the Imaging Lab. I would also like to thank my friends and family for their love and encouragement, particularly my grandmothers. I finally want to send recognition and love to my partner, for with your support and endless tenacity I reached the finish line.

Contents

1	Introduction	1
1.1	Image Processing Prelude	1
1.2	Observation Model	4
1.2.1	Problem Statement	7
1.3	Objectives	8
1.4	Outline	9
1.5	Literature Review	9
1.5.1	Non-Local Means Denoising Algorithm	10
1.5.2	Regularization Using A Graph based Framework	12
2	Non-Local Boosting Of Image Denoising Algorithms	19
2.1	Boosting Of Image Denoising Algorithms	20
2.2	Iterative “Boosting” Of “Boosting” Operators	21
2.3	Experiments And Results Regarding Boosting And Remarks	23
3	Graph Based Regularization For Image Super Resolution	30
3.1	Forward Model And Inverse Problem For Single Image Super Resolution	31
3.2	Regularization With The Normalized Graph Laplacian	34
3.3	Single Image Super Resolution Utilizing A Graph Laplacian Operator As A Penalizer	40

3.4	Computation Of Similarity Block Matrices And Associated Filtering Operator	44
3.5	Implementation Details	47
4	Numerical Experiments And Results	49
4.1	Performance Of Restoration Algorithm For Various Levels Of AWGN	51
4.2	Tuning Parameters	78
4.3	SR Restoration Analysis Regarding Decimation Factor L	79
4.4	Analysis Regarding Smoothing Parameter h For Non-Local Weight Operator	79
4.5	Analysis Regarding Degree of Vertices for Image Graph With Respect To Non-Local Weight Operator	81
5	Conclusion And Future Works	83
Appendices		85
A	Local Averaging Blur Operator	86
B	Train Downsampling Decimation Operator	87
C	Conjugate Gradient Descent	88
D	Block Matching 3D Filtering Denoising Algorithm	90
Bibliography		96

List of Tables

3.1	Dimension of each term composing proposed objective function (3.10) under the assumption that images are square (i.e $N \times N$) and in lexicographical notation with decimation factor denoted by L	43
4.1	Numerical measures of success regarding increasing the resolution of images for no AWGN ($\sigma = 0$).	53
4.2	Numerical measures of success regarding increasing the resolution of images which are corrupted using AWGN with standard deviation $\sigma = 10$. . .	54
4.3	Numerical measures of success regarding increasing the resolution of images which are corrupted using AWGN with standard deviation $\sigma = 50$. .	55
4.4	Resulting PSNR (dB) of super resolved Horse image with $\sigma = 10$ AWGN varying tuning parameters β and η . These results are for when β is positive.	78
4.5	Resulting PSNR (dB) of super resolved Lime image while varying decimation factor L for various amounts of AWGN	79
4.6	Quantitative measures for lime image regarding different search neighbourhood radii for various levels of AWGN.	82

List of Figures

1.1	Visualization of the observation model to represent measurement \underline{y} . We note that the image \underline{y} and restored solution \underline{x} are represented in Lexico-graphical format, and operator \mathbf{H} a sparse matrix. Factor L is set to 1 when the restoration does not require a change in dimension. This implies that for when there is a requirement for change in dimensionality according to the inverse problem to be solved, L will be set to a value that is not 1.	5
1.2	Weight scheme of the Non-Local means denoising algorithm. For similar gray intensity levels vectors regarding pixel neighbourhoods, there will be a large weight score $w(p, q_1)$ and $w(p, q_2)$ while inversely dissimilar pixel neighbourhoods will be scored a much smaller weight such as $w(p, q_3)$. . .	11
1.3	Simple un-directed graph with six vertices $V = \{1, 2, 3, 4, 5, 6\}$ and corresponding neighbouring nodes described by an edge $E = \{\{1, 2\}, \dots, \{1, 6\}\}$ between them. In addition all edges are paired with a numerical weight used to quantify some measure between nodes.	13
1.4	The color shape image is RGB channeled and labeled (a) the ground truth, (b) the noisy input and (c) the denoising of a color image using regularization centered around a non-local normalized graph Laplacian. The solution is computed using an iterative scheme.	17

2.2	Sequential boosting of NLM algorithm under operator $\mathcal{B}^n(\mathbf{f})$ applied on noisy Cameraman (Top) AWGN ($\sigma = 12.5$), (Middle) AWGN ($\sigma = 2.5$), (Bottom) Zoomed AWGN ($\sigma = 2.5$) and corresponding PSNR (dB).	26
2.3	Sequential boosting of NLM algorithm under operator $\mathcal{B}^n(\mathbf{f})$ applied to noisy Saturn (Top) AWGN ($\sigma = 12.5$), (Middle) AWGN ($\sigma = 2.5$), (Bottom) Zoomed AWGN ($\sigma = 2.5$) and corresponding PSNR (dB).	27
2.4	Sequential boosting of NLM algorithm under operator $\mathcal{B}^n(\mathbf{f})$ applied on noisy Cameraman corrupted with (Top) AWGN ($\sigma = 2.5$) and (Bottom) Zoomed AWGN ($\sigma = 2.5$) with corresponding PSNR (dB) when ρ is less than zero (negative values).	28
2.5	Resulting $\mathcal{B}^n(\mathbf{f}) \cdot \underline{y}$ on noisy image \underline{y} with AWGN ($\sigma = 2.5$) for (a) $n = 1$, (b) $n = 10$, (c) $n = 20$, (d) $n = 40$, (e) $n = 50$, (f) $n = 60$, corresponding to the optimal $\rho = 1.3$, (g) $n = 1$, (h) $n = 10$, (i) $n = 20$, (j) $n = 40$, (k) $n = 50$, (l) $n = 60$, corresponding to the optimal $\rho = 1.2$	29
3.1	(a) Low resolution input. (b) Approximation from pixel Nearest Neighbour interpolation. (c) Approximation using BiCubic interpolation. (d) Desired ground truth.	31
3.2	Top Row: The Forward model entails given an Ideal HR image and degradation operator \mathbf{H} followed by corruption with additive noise, a low resolution measurement \underline{y} will be produced. Bottom Row: Inverse problem for approximating a high resolution solution given low resolution measurement \underline{y}	32

3.3	Deblurring examples with blurred noisy Girl image by out-of-focus blur with radius 7 and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 29.40 dB, SSIM = 0.8734), and (d) output of proposed deblurring algorithm (PSNR = 30.58 dB, SSIM = 0.9058).	35
3.4	Non-Local means algorithm neighbour assignment schematic.	38
4.1	Motorcycle data with $\sigma = 0$ AWGN.	60
4.2	Motorcycle data with $\sigma = 10$ AWGN.	61
4.3	Motorcycle data with $\sigma = 50$ AWGN.	62
4.4	Pineapple field data with $\sigma = 0$ AWGN.	63
4.5	Pineapple field data with $\sigma = 10$ AWGN.	64
4.6	Pineapple field data with $\sigma = 50$ AWGN.	65
4.7	Pool data with $\sigma = 0$ AWGN.	66
4.8	Pool data with $\sigma = 10$ AWGN.	67
4.9	Pool data with $\sigma = 50$ AWGN.	68
4.10	Horses data with $\sigma = 0$ AWGN.	69
4.11	Horses data with $\sigma = 10$ AWGN.	70
4.12	Horses data with $\sigma = 50$ AWGN.	71
4.13	Boats data with $\sigma = 0$ AWGN.	72
4.14	Boats data with $\sigma = 10$ AWGN.	73
4.15	Boats data with $\sigma = 50$ AWGN.	74
4.16	Lime data with $\sigma = 0$ AWGN.	75
4.17	Lime data with $\sigma = 10$ AWGN	76
4.18	Lime data with $\sigma = 50$ AWGN	77
4.19	Lime image for various levels of AWGN utilizing different smoothing parameters $h = 10\sigma$	80

D.1 Flowchart for denoising by hard-thresholding in 3D transform domain with block-matching. A similar flowchart can be constructed for the second phase of the algorithm, but instead denoising using a Wiener filter in 3D transform domain.	93
D.2 Denoised house results after applying BM3D denoising algorithm. The images on the left correspond to a noisy ($\sigma = 35$) House and enlarged fragments from it; On the right are the denoised image (PSNR=31.21 dB) and corresponding image fragments.	95

Chapter 1

Introduction

1.1 Image Processing Prelude

Everyday a countless number of digital images are shared, viewed or utilized in some fashion due to the relative ease of common image capturing devices, as well as the explosion of social media outlets. In certain occurrences, one could wish to increase the quality or resolution of an image for a larger and clearer representation of the visual elements that define said image. There are two avenues to solving this problem the first enhancing sensors of capturing devices and second applying an image restoration algorithm. Regarding the first avenue, majority of image capturing devices entail sensors that are built as an array of light detectors to sample a continuous signal and convert it to a discrete representation of pixel intensities. For an accurate image intensity function, sampling is imperative as defining a numerical representation for a given scene requires a function with specific structure and intensity properties that are sampled ideally on the scene itself. If this definition of an image function can be accomplished, we then have sufficient information to represent an image as a mathematical function.

Commonly image functions are defined in matrix form with intensity values belonging to rows and columns to provide a spatially and objectively accurate representation of the chosen scene. In practice regardless of device sophistication and precautions taken during acquisition, signals may result in deviations of a theoretically perfect result if acquired measurement does not satisfy the Nyquist criterion [1]. During acquisition, pixel intensities sampled for a given scene are determined by the amount of light detected by the sensor given some assigned scene [2]. The variety of image distortions that may occur during sampling are numerous including such effects as noise, blurring, and aliasing to name a few. It is not unreasonable to assume one way to remedy the signal acquisition process could be to increase the density of a detector array, though there are natural limitations that adding sensors could raise such as cost or hardware concerns.

The second option entails employing an image restoration algorithm, an inexpensive and powerful alternative to enhancing the visual quality of the image. We loosely divide restoration algorithms into two main categories. The first category consists of algorithms that rely on minimizing some energy function to capture some desired restoration process [3, 4, 5, 6] deemed as reconstruction methods. Particularly, reconstruction based SR methods usually utilize a sophisticated prior knowledge to restrict the solution into a specific solution space, which promotes toward sharp details. The performance of these reconstruction methods are heavily dependent on the scale factor and as the scale factor increases the quality of the reconstruction degrades, in addition to these methods being time consuming. The second category of methods [7, 8] fall under deep learning approaches which are heavily data dependent requiring large amounts of data to both train and test a given network to reach sufficient accuracy and precision requirements. These deep learning methods have proven very successful for several applications, such as image denoising, Super Resolution (SR), colorization and image inpainting to name a few [9]. Given the advancements of software and computer hardware deep learning methods have become particularly popular for solving the problem of SR as they produce high

quality results in a fraction of the time compared to reconstruction based restoration methods. Even though reconstruction based methods are not fast as their deep learning counterparts, reconstruction based methods rely on a statistical framework and an understanding of the physical process when capturing an image. These reconstruction based methods are full of rich and vital mathematics which to advance the field of image processing, must be explored.

We know that all things in nature seek a lower energy state, and researchers have used this fact for a variety of physical and engineering problems. In this work we are interested in restoration methods that solve an objective function through minimization under specific constraints and parameters to impose a desired restored result. As there are many types of image degradation, likewise many algorithms have been developed to combat these image distortions. As an example to denoise an image these methods [10, 11] are adaptations of a variational approach, which restore an image through prescribing an energy to an image signal and reducing the energy. This concept of energy is used to describe a measure of pixel intensity when formulating a minimization problem under a probability framework such as MAP (*Maximum A Priori*). By reducing the energy the quality of the image will gradually be improved. As an example, a high energy image would be considered visually “bad” and inversely low energy a “good” quality image. We note that the energy of an image in general corresponds to the mean squared value of the pixel intensities. As such, in general the energy regarding image \underline{x} as a continuous signal over infinite time is posed as a functional of the form

$$E_\infty = \int_{-\infty}^{\infty} \|\underline{x}(t)\|^2 dt \quad (1.1)$$

where Ω is the image domain. After calculating the first variation of energy, one can find the minimum (restored image) of Functional 1.1 through an optimization approach, entailing minimization using a descent approach.

Loosely speaking, energy minimization methods (or reconstruction methods) can be divided through their use of local or non-local pixel information during restoration. For

local reconstruction algorithms, a restored pixel is usually estimated from intensity information in a small region around the restored pixel [12]. The second class [13, 14, 15] of algorithms are those which restore a given pixel using information from a specified non-local subdomain, and thus in the process collects data from regions which have similar features as target pixel. This process consists of replacing a pixel intensity via processing of a specified set of local patches, denoted as a patch-based method. One of the earliest and fundamental implementations of these non-local restoration methods is the Non-Local Means [16] algorithm (NLM), which denoises an intensity value using a weighted average of pixels that belong to specified neighbouring region. Using these redundant representations for a low resolution images has become a powerful tool for denoising [17]. As such these restoration algorithms centered around a non-local approach have shown to be successful in a variety of image enhancement requirements such as image denoising, deblurring, and super resolution [18] with their usage of redundant representations imposed by a non-local framework.

1.2 Observation Model

We introduce the model for obtaining measurement $\underline{y} \in \mathbb{R}^{(NM) \times 1}$ using a discrete representation where

$$\underline{y} = \mathbf{H}\underline{x} + \underline{n}_\sigma \quad (1.2)$$

with $\underline{x} \in \mathbb{R}^{(NM) \times 1}$ is the realization of ideal image, $\mathbf{H} \in \mathbb{R}^{(NM) \times (NM)}$ a degradation operator and \underline{n}_σ additive white Gaussian noise (AWGN) with σ standard deviation and zero mean. Gaussian white noise is included in the observation model to mimic the effects of noise for a variety of random processes that occur naturally during signal acquisition. As such, the white noise as a discrete signal is regarded as a signal with samples that are deemed a sequence of serially uncorrelated random variables with zero mean and standard deviation σ . Specifically for image processing, the term “white” refers to samples drawn

required to be independent and have identical probability distribution, resulting in a normal distribution of amplitude. In this work the ordering of the image pixels when constrained to vector representation is lexicographical¹.

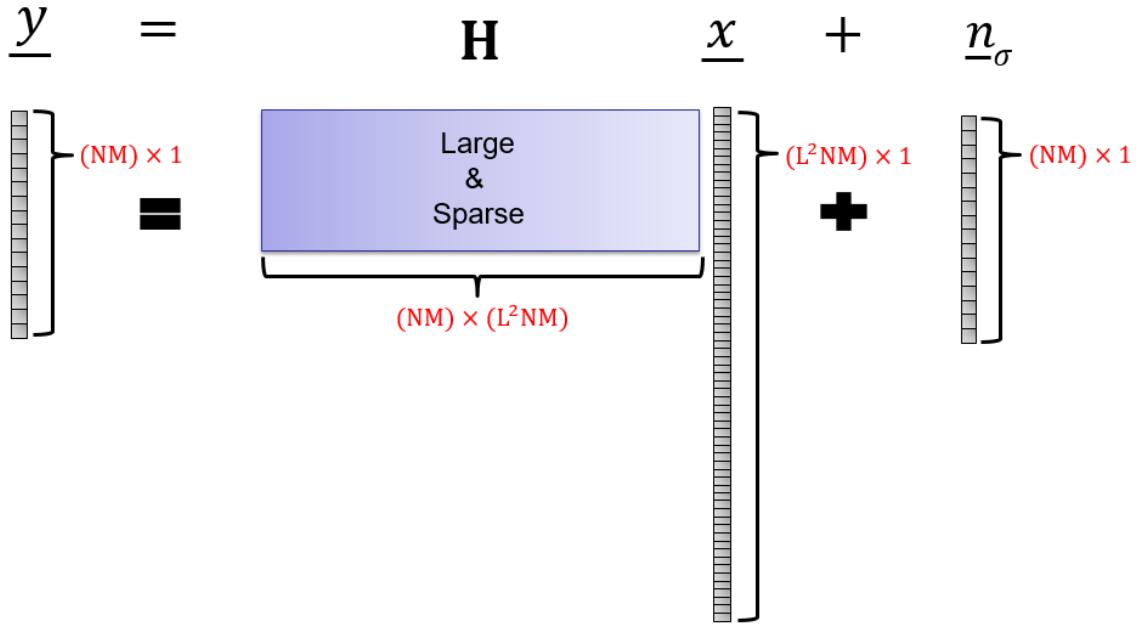


Figure 1.1: Visualization of the observation model to represent measurement \underline{y} . We note that the image \underline{y} and restored solution \underline{x} are represented in Lexicographical format, and operator \mathbf{H} a sparse matrix. Factor L is set to 1 when the restoration does not require a change in dimension. This implies that for when there is a requirement for change in dimensionality according to the inverse problem to be solved, L will be set to a value that is not 1.

One way to produce ideal object \underline{x} from measurement \underline{y} involves the following maximum likelihood estimator (MLE)

$$\underline{x}_{MLE} = \operatorname{argmin}_{\underline{x}} \|\mathbf{H}\underline{x} - \underline{y}\|_2^2.$$

¹The term lexicographical is based on the convention of labeling a pixel in row i and column j of an image as a pixel pair (i, j) and then ordering those pairs treating pixel i as the most significant.

Provided there is enough regularity,

$$\underline{x}_{MLE} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \underline{y} \quad (1.3)$$

which given our matrix formulation of image, is considered a least squares problem. In general a unique solution for an inverse problem using the MLE estimator (1.3) approach depends on if the Gramian matrix $\mathbf{H}_G = \mathbf{H}^T \mathbf{H}$ is square, implying the matrix is invertible. When \mathbf{H}_G is invertible maximum likelihood estimate guarantees a unique solution, implying a well-posed problem. In contrast when matrix \mathbf{H}_G is non invertible the solutions to the respective linear system no longer exist and thus we are left solving an ill-posed inverse problem.

It is evident that degradation operator \mathbf{H} has an influence on the well-possedness of a given inverse problem. Consequently, when an inverse problem is ill-posed we regularize

$$\underline{x}_{REG} = \operatorname{argmin}_{\underline{x}} (\|\mathbf{H}\underline{x} - \underline{y}\|_2^2 + \lambda \underline{x}^T C \underline{x}) = \operatorname{argmin}_{\underline{x}} (\|\mathbf{H}\underline{x} - \underline{y}\|_2^2 + \lambda \mathcal{R}(\underline{x})).$$

which includes a regularizing operator C which is positive-definite and λ , a small positive scalar that controls the amount of regularization [19]. This is equivalent to solving

$$\underline{x}_{REG} = (\mathbf{H}^T \mathbf{H} + \lambda C)^{-1} \mathbf{H}^T \underline{y}.$$

The inclusion of positive-definite matrix C in this format is defined by the term $\mathcal{R}(\underline{x}) = \underline{x}^T C \underline{x}$ based on Tikhonov regularization [19]. In the field of image processing much of the progress in past decades has been an introduction of effective image priors. With regularization one can impose prior conditions, converting solution of energy functional to finding \underline{x} that maximize imposed conditions given measurement \underline{y} . This concept of conditional estimates, where estimating \underline{x} is conditioned on the knowledge of measurement \underline{y} naturally leads to Bayes' law with *aposteriori* probability conditional density defined as

$$\Pr(\underline{x}|\underline{y}) = \frac{\Pr(\underline{y}|\underline{x}) \Pr(\underline{x})}{\Pr(\underline{y})} \quad (1.4)$$

where $\Pr(\underline{y}|\underline{x})$ represents the joint probability and can be viewed as the probability of measurement \underline{y} given any \underline{x} , term $\Pr(\underline{x})$ infers the amount of information we have before being given measurement \underline{y} and $\Pr(\underline{y})$ included for normalization purposes [20]. The use of the posterior density for computing a restored estimate is a well known process [20]. Solutions found from estimators such as Maximum *A Posteriori* (MAP) and MLE are guided according to the principle that noise vector \underline{n}_σ will be modeled under a Gaussian process implying vector \underline{n}_σ and probability priors for images \underline{y} and \underline{x} will follow a normal probability density with zero mean and standard deviation σ [20]. As such solving an inverse imaging problem following a Bayesian framework (provided information about operator \mathbf{H} is available) the solution \underline{x} can be found from the following MAP estimator

$$\underline{x}_{MAP} = \operatorname{argmin}_{\underline{x}} (\|\mathbf{H}\underline{x} - \underline{y}\|^2 + \lambda \mathcal{R}(\underline{x})).$$

in which probability priors are encoded via regularization functional $\mathcal{R}(\underline{x})$. The intent of the regularization term is to push the solution in the direction of the maximum likelihood solution [20]. This formulation is a common approach for solving image restoration problems, specifically super resolution models [21]. When this prior information via composition of probability density function is not included the MAP estimator collapses to the maximum likelihood estimator, which relies only on the measurement \underline{y} , ignoring any contributions from a probability based prior. As expected, when Gramian matrix $\mathbf{H}^T \mathbf{H}$ is non-square the linear system is ill posed, and thus there are infinitely many solutions which naturally leads to the regularization term not only contributing toward uniqueness, but required strictly from an algebraic perspective when increasing the dimension of an image [21].

1.2.1 Problem Statement

Problem *Given single low resolution measurement \underline{y} and degradation operator $\mathbf{H} = \mathbf{DB}$ a composition of blurring \mathbf{B} followed by downsampling \mathbf{D} , we wish find a higher-*

resolution image \underline{x} that minimizes

$$S(\cdot, \underline{y}) = \operatorname{argmin}_{\underline{x}} (\mathcal{D}(\underline{y}, \underline{x}) + \mathcal{R}(\underline{y}, \underline{x})) \quad (1.5)$$

where \mathcal{D} the data fidelity term, attempts to ensure blurred, decimated and filtered solution \underline{x} will be as close as possible to the filtered version of \underline{y} and \mathcal{R} , the regularization term ensures the well-posedness of the problem. That is, we want to find a unique solution $\underline{x} \in \mathbb{R}^{(LN \times LM) \times 1}$ given measurement $\underline{y} \in \mathbb{R}^{(N \times M) \times 1}$ such that ratio L between dimensions is $L > 1$ and restricted to integer values. Typically one take $L = 2$.

We note that in general it is possible to set up-scale factor L as either an integer (odd or even), as well as a fractional scale. In addition, we are not restricted to first up-sampling the image by some factor, then downsampling to the appropriate dimension regarding factor L . Factor L is set to two for easing implementation details, as well the mathematical formulation of the SR method.

1.3 Objectives

For a variety of image inverse problems, non-local based algorithms have shown to be a successful methodology when computing a restored solution. In this work we plan to analyze the effectiveness of using residual enhancement approach, in which information from the cleaned signal is reintroduced into the final denoised approximation using an iterative scheme via a non-local objective approach [22]. We analyze the **Strengthen-Operate-Subtract (SOS)** enhancement algorithm proposed by [22] and in turn derive an analytical expression corresponding to a successive application of this proposed boosting scheme as a new denoising operator. In addition, we analyze the applicability of utilizing a non-local normalized graph Laplacian via a graph based framework to regularize a single image super resolution inverse problem. Single image super resolution is an ill-posed inverse problem which requires efficient regularization to produce a quality high resolution approximation.

1.4 Outline

Our goal of this dissertation is two fold. Firstly in Chapter 2 we derive an analytical expression for successive applications of the SOS boosting algorithm [22] which treats a non-local denoising algorithm as a “black-box” operation. From this SOS “boosting scheme’’ we show successive applications does not always enhance the input image, and in reality is equivalent to a re-parameterization of the original SOS “boosting” algorithm. Given the success of non-local methods for a variety of image restoration problems, we analyze the validity for utilizing a new definition of the non-local normalized graph Laplacian derived for image deblurring by [23] as a regularization term, as well as a “sharpened residual” data fidelity term. Chapter 3 provides the derivation of the new normalized graph Laplacian based on the debluring algorithm proposed by [23], as well as the procedure required to convert the normalized graph Laplacian into a data fidelity and regularization term for composition of an energy functional for debluring. We also provide the derivation for our proposed single image super resolution algorithm including composition of the data fidelity and regularization terms derived according to the new definition of the normalized graph Laplacian. Solving procedure, implementation and software are also included in this chapter. Following this, Chapter 4 contains a variety of restoration experiments for our proposed single image super resolution restoration method. Finally, Chapter 5 includes final remarks of the validity and applicability of utilizing our non-local normalized graph Laplacian for our single image super resolution problem. We also include future works.

1.5 Literature Review

Here we perform a literature review for key ideas regarding this dissertation. First we give a brief review of the widely regarded Non-Local means denoising [16] algorithm which takes advantage of the natural redundancy of images utilizing a non-local ap-

proach of intensity gray level vectors. We then study the normalized graph Laplacian for regularization of image restoration problems, particularly image denoising [24].

1.5.1 Non-Local Means Denoising Algorithm

We begin with the Non-Local Means (NLM) [16] denoising algorithm which exploits the innate redundancy of natural images by replacing every pixel in the image via a weighted average of all the pixels in the discrete image domain Ω . One of the main contributions for success is that the NLM compares both the gray level of a pixel in a single point, but also structural configuration of a given neighbourhood. All weights measurements between pixels i and j are determined from a neighbourhood similarity scheme of respective pixels intensity gray level vector. The scheme follows an approach such that neighbourhood similarity of pixels where “similar” neighbourhoods are assigned high weight and dissimilar neighbourhoods low weight. Following the formulation of the NLM algorithm, the neighbourhood of size k is defined $\forall i \in \Omega$

$$\mathcal{N}_k\{i\} = \{j \mid \|r\|_\infty \leq k\}.$$

where $\|\underline{v}\|_\infty := \max_i |\underline{v}_i|$ the maximum norm. The Non-Local Means operator denoted by $\mathbf{f}(\cdot)$ denoises the intensity value of every pixel $\forall i \in \Omega$ from a weighted average of all pixels in the image domain Ω of a noisy image \underline{y} via

$$\mathbf{f}(\underline{y}(i)) = \sum_{j \in \Omega} w(i, j) \underline{y}(j) \quad (1.6)$$

where weight $w(i, j)$ and normalization $Z(i)$ are defined as

$$w(i, j) = \frac{1}{Z(i)} \left(\exp - \frac{1}{h^2} \|\underline{y}(\mathcal{N}_k\{i\}) - \underline{y}(\mathcal{N}_k\{j\})\|_2^2 \right), \quad Z(i) = \sum_{j \in \Omega} w(i, j)$$

respectively. Measuring similarity of between pixel pair involves a decreasing function of the weighted Euclidean distance of the neighbourhood gray level vector between target

pixel and associated neighbouring pixels, and parameter h is a degree of filtering for noise removal. We note that $\|\cdot\|_2^2$ denotes the Gaussian weighted ℓ^2 norm for any patch \mathbf{P}

$$\|\mathbf{P}\|_2^2 = \|G \star \mathbf{P}\|_2^2$$

where G is a Gaussian kernel with a standard deviation of σ and \star the convolution operator. Normalizing constant $Z(i)$ is computed from a summation of every neighbouring gray level vector of a given pixel $i \in \Omega$. As an image grows in size comparing every pixel

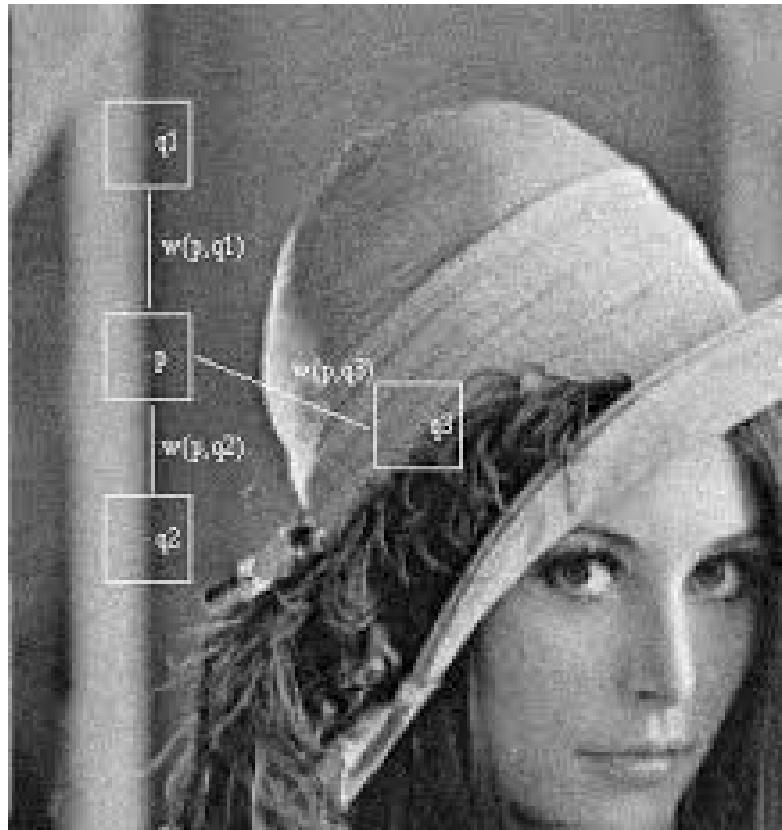


Figure 1.2: Weight scheme of the Non-Local means denoising algorithm. For similar gray intensity levels vectors regarding pixel neighbourhoods, there will be a large weight score $w(p, q_1)$ and $w(p, q_2)$ while inversely dissimilar pixel neighbourhoods will be scored a much smaller weight such as $w(p, q_3)$.

Figure courtesy of Buades.A et. al., A Non-Local Algorithm for Image Denoising, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005, VOL 2, pg. 60-65.

in the image, extracting associated neighbour intensity vector and computing similarity becomes computationally demanding. Thus rather than comparing each pixel with every other pixel in the image, only a specified region of pixels belonging to Ω will be considered a neighbour of target pixel we wish to denoise. Restricting the NLM algorithm by only computing similarities between pixel neighbourhoods in a sample region greatly reduces the computational burden of the algorithm.

1.5.2 Regularization Using A Graph based Framework

There have been a variety of successful image restoration algorithms centered around restoring a given measurement utilizing a graph based approach [25]. In regards to the energy based optimization, one option for a regularizer is the normalized graph Laplacian which is centered around defining an image as a weighted graph and computing associated Laplacian difference operators for composition. In terms of an image the high frequency components of the image corresponds to the vital edge information creating the structure of the image. We expect that regions which contain edge information, there will be a large change of intensity. As the Laplacian is a second order difference operator, it is especially good at edge preservation (finding pixel intensities in which have high change in intensity) during restoration. Specifically some successful restoration algorithms centered around the composition of a Graph Laplacian include [26, 27, 28, 29] which Laplacian operators are computed according to non-local “interactions” of pixels. One of the earliest works regarding a non-local Graph Laplacian as a regularizer is Bougleux et. al. [24] for image denoising. From this work we begin with the definition of an image as an intensity function $\underline{x} = (x(1), \dots, x(NM))^T$ on vertices V of a weighted k -neighbourhood graph $G_k = G(V, E, w)$ consisting of a triple denoted by

1. A set of vertices V (image pixels);
2. Collection of edges $E \subset V \times V$;

3. Set of weights $w(u, v)$ to describe similarity between vertices.

Given a graph G_k the k -neighbourhood of a vertex $u \in V$ is the set of vertices located at a distance lower than k such that

$$\mathcal{N}_k(u) = \{v \in V \setminus \{u\} : \mu(u, v) \leq k, k \geq 0\} \quad (1.7)$$

where $\mu : V \times V \rightarrow \mathbb{R}^+$ quantifies the proximity between two vertices, usually based on the chosen weight function. Thus an edge (u, v) only exists in k -neighbourhood of graph G_k when $v \in \mathcal{N}_k(u)$ and vice-versa. In the context of image processing, the composition of the graph Laplacian requires a weight function K which describes the underlying structure of the image.

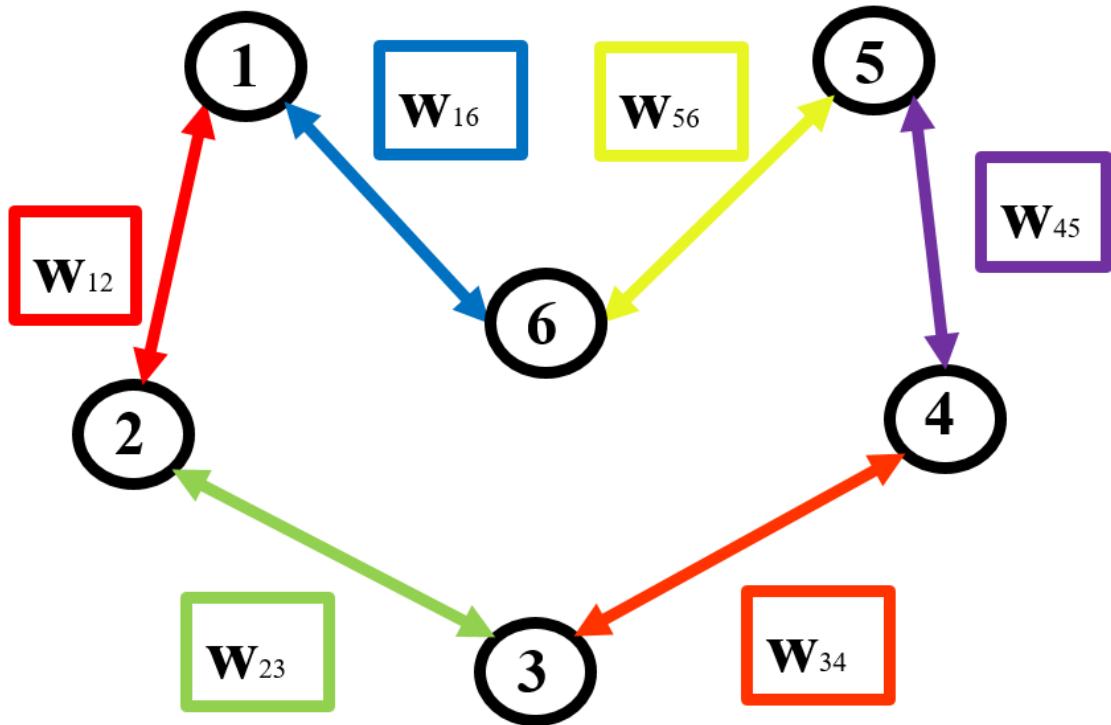


Figure 1.3: Simple un-directed graph with six vertices $V = \{1, 2, 3, 4, 5, 6\}$ and corresponding neighbouring nodes described by an edge $E = \{\{1, 2\}, \dots, \{1, 6\}\}$ between them. In addition all edges are paired with a numerical weight used to quantify some measure between nodes.

In this thesis, the weighted image graphs are connected and undirected $\{u, v\} \in E \Leftrightarrow \{v, u\} \in E$ such that $w(u, v) = w(v, u)$. The weight of function $w : V \times V \rightarrow \mathbb{R}^+$ between

vertices $\{u, v\} \in E$ denoted as w_{uv} which are non-negative by convention is given by ²

$$w(u, v) = \begin{cases} w_{uv} & \text{if } \{u, v\} \in E \\ 0 & \text{otherwise.} \end{cases} \quad (1.8)$$

Depending on the choice of the weight function, it is possible to compose either a non-local or local similarity [24] block weight matrix. We note that vertices u and v are considered similar if $w_{uv} = 1$ and dissimilar if $w_{uv} = 0$. For unweighted graphs, the weight function would be $w = 1$ implying all edges $\{u, v\} \in E$ would be equally weighted.

Another important property of a graph is the degree of a vertex V denoted by $\deg(V) : V \rightarrow \mathbb{R}^+$ which is the sum of weights in the neighbourhood of that vertex

$$\deg(u) = \sum_{v \sim u} w_{uv}, \quad \forall u \in V \quad (1.9)$$

with $v \sim u$ representing the vertices v which connected to vertex u induced according to edge (u, v) so that $v \in \mathcal{N}_k(u)$. Furthermore, we define D as a sparse diagonal degree matrix with

$$D(u, v) = \begin{cases} \deg(u) & u = v \\ 0 & u \neq v. \end{cases} \quad (1.10)$$

The weighted difference of a function $\underline{x} : V \rightarrow \mathbb{R}$ on an edge $\{u, v\}$ of graph G_k is defined as

$$(dx)(u, v) = \sqrt{w(u, v)}(\underline{x}(v) - \underline{x}(u)). \quad (1.11)$$

For more information regarding the inclusion of the root term regarding similarity matrix $w(u, v)$ please refer to the work of F. Chung [30]. As such from the edge derivative 1.11 the weighted gradient vector of a function \underline{x} at vertex $u \in V$ is given by

$$\nabla \underline{x}(u) = [dx(u, 1), \dots, dx(u, M)]^T, \quad \forall \{u, v\} \in E, \quad j \in \{1, 2, \dots, M\} \quad (1.12)$$

²In this thesis, the weighted image graphs are connected and undirected $\{u, v\} \in E \leftrightarrow \{v, u\} \in E$ such that $w(u, v) = w(v, u)$.

whose magnitude represents the maximum rate of change of function \underline{x} at the u -th vertex. The direction of the gradient follows the maximum rate of change of the respective function, and total flow into vertex u is given by taking the divergence operator of gradient (1.12) which produces the Laplace operator for vertex $u \in V$ according to

$$\Delta \underline{x}(u) = \sum_{v \sim u} w(u, v) (\underline{x}(v) - \underline{x}(u)). \quad (1.13)$$

Given the gradient operator 1.12, the regularity of an image \underline{x} at vertex $u \in V$ according to its p -Dirichlet energy [24] based on the local variation is given by functional

$$\mathcal{R}(\underline{x}) = \frac{1}{p} \left(\sum_{u \in V} (\nabla \underline{x}(u))^p \right)^{\frac{1}{p}}, \quad p \in (0, +\infty).$$

Choosing $p = 2$ the above regularization functional corresponds to Tikhonov regularization [24] and is used to compose two versions of the graph Laplacian, the non-normalized and normalized versions respectively

$$\mathcal{R}_u(\underline{x}) = \frac{1}{2} \sum_{u \in V} \sum_{v \sim u} w(u, v) (\underline{x}(v) - \underline{x}(u))^2 = \underline{x}^T (D - W) \underline{x}, \quad (1.14)$$

$$\mathcal{R}_n(\underline{x}) = \frac{1}{2} \sum_{u \in V} \sum_{v \sim u} w(u, v) \left(\frac{\underline{x}(v)}{\sqrt{D(v, v)}} - \frac{\underline{x}(v)}{\sqrt{D(u, u)}} \right)^2 = \underline{x}^T (I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \underline{x}. \quad (1.15)$$

As seen from equation (1.14) the non-normalized graph Laplacian \mathcal{R}_u is composed of a difference between sparse similarity weight matrix $W(u, v) = w(u, v) \quad \forall u, v \in V$ and matrix D . When comparing the non-normalized graph Laplacian to the normalized counterpart \mathcal{R}_n (1.15) this operator includes a normalization of each pixel intensity value according to the respective degree of that vertex. Observing regularization term (1.14) one notices that the Graph Laplacian functional (1.14) can be equivalently expressed as the following term $\underline{x}^T (D - W) \underline{x}$ which is constructed according to the following assumptions.

1. As the image graph is undirected an edge (u, v) has an identical weight to corresponding reverse edge (v, u) , i.e $w_{uv} = w_{vu}$.

2. For a given vertex $u \in V$ the only vertices v according to edge set $\{u, v\} \in E$ will be assigned a similarity measure. This implies that the similarity between a given vertex and all vertices not belonging to edges E are set to 0.

Using the assumptions above we illustrate the equivalence statement for the Graph Laplacian regularization functional when constructed as $\underline{x}^T(D - W)\underline{x}$. For the following equivalence proposition, we set the nomenclature for intensity value of a given vertex $u \in V$ from $\underline{x}(u)$ to \underline{x}_u . As such the graph Laplacian as a regularizer is formulated as

$$\begin{aligned}\mathcal{R}(\underline{x}) &= \frac{1}{2} \sum_{u \in V} \sum_{v \in V} w(u, v) (\underline{x}_v - \underline{x}_u)^2 \\ &= \frac{1}{2} \left(\sum_{u \in V} \sum_{v \in V} w(u, v) (\underline{x}_v^2 - 2\underline{x}_v \underline{x}_u + \underline{x}_u^2) \right) \\ &= \frac{1}{2} \left(\sum_{u \in V} \sum_{v \in V} w(u, v) \underline{x}_v^2 - 2 \sum_{u \in V} \sum_{v \in V} w(u, v) \underline{x}_v \underline{x}_u + \sum_{v \in V} \sum_{v \sim u} w(u, v) \underline{x}_u^2 \right). \quad (1.16)\end{aligned}$$

From the assumption that edges $\{u, v\} \in E$ are undirected and corresponding similarity measures are equivalent $w_{uv} = w_{vu}$, then equation (1.16) is written

$$\begin{aligned}\mathcal{R}(\underline{x}) &= \frac{1}{2} \left(2 \sum_{u \in V} \sum_{v \in V} w(u, v) \underline{x}_u^2 - 2 \sum_{u \in V} \sum_{v \in V} w(u, v) \underline{x}_v \underline{x}_u \right) \\ &= \sum_{u \in V} \left(\sum_{v \in V} w(u, v) \underline{x}_u - \sum_{v \in V} w(u, v) \underline{x}_v \right) \underline{x}_u. \quad (1.17)\end{aligned}$$

Restricting similarity measure between set of edges $\{u, v\}$ equation (1.17) can be written as

$$\begin{aligned}\mathcal{R}(\underline{x}) &= \sum_{u \in V} \underline{x}_u \left(\sum_{v \sim u} w(u, v) - \sum_{v \sim u} w(u, v) \underline{x}_v \right) \underline{x}_u \\ &= \left(\underline{x}_1, \dots, \underline{x}_N \right)^T \left(\begin{pmatrix} \sum_{v \sim 1} w_{1v} & & \\ & \ddots & \\ & & \sum_{v \sim N} w_{Nv} \end{pmatrix} - \begin{pmatrix} \vdots \\ W_{uv} \\ \vdots \end{pmatrix} \right) \begin{pmatrix} \underline{x}_1 \\ \vdots \\ \underline{x}_N \end{pmatrix} \\ &= \underline{x}^T (D - W) \underline{x}.\end{aligned}$$

where diagonal matrix D has entries along the main diagonal, and are zero everywhere else. From the result above, one notices that equation (1.14) is found by a normalization with respect to the square root of sparse diagonal matrix D where $D^{-\frac{1}{2}} = \frac{1}{\sqrt{D}}$. The normalization of equation (1.14) leads to the normalized graph Laplacian defined as

$$\begin{aligned} &= D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} = D^{-\frac{1}{2}}DD^{-\frac{1}{2}} - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \\ &= I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}. \end{aligned}$$

In the context of image restoration according to [24], the energy functions which include the normalized graph Laplacian chosen as a regularizer take the form

$$E(\cdot, \underline{y}) = \underset{\underline{x}}{\operatorname{argmin}} \left(\|\underline{y} - \mathbf{H}\underline{x}\|^2 + \eta \mathcal{R}(\underline{x}) \right) = \underset{\underline{x}}{\operatorname{argmin}} \left(\|\underline{y} - \mathbf{H}\underline{x}\|^2 + \eta \underline{x}^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})\underline{x} \right).$$

The first term is a data fidelity term which is designed to force a solution \underline{x} to remain similar to input \underline{y} . The second term is a regularizer derived using the definition of the normalized graph Laplacian from equation (1.15). For example in the introductory paper [24] the normalized graph Laplacian is a regularization operator which will smooth noise of a specific target pixel using contributions from a local neighbourhood centered around vertex $u \in V$.

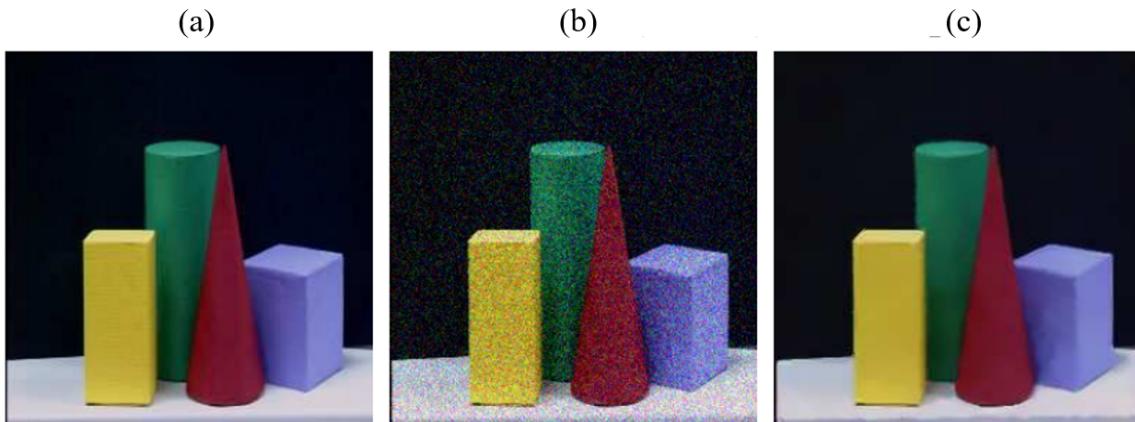


Figure 1.4: The color shape image is RGB channeled and labeled (a) the ground truth, (b) the noisy input and (c) the denoising of a color image using regularization centered around a non-local normalized graph Laplacian. The solution is computed using an iterative scheme.

As seen in Figure 1.4, the normalized non-local graph Laplacian performs well as a regularization functional. The results from the denoising experiment above imply applying non-local schemes utilizing a normalized graph Laplacian via a graph based approach shows promise for other types of image restoration problems.

Figure courtesy of Bougleux S. et. al., Discrete Regularization on Weighted Graphs for Image and Mesh Filtering, Scale Space and Variational Methods in Computer Vision, pp. 128-139.

Chapter 2

Non-Local Boosting Of Image Denoising Algorithms

In past decades non-local example based algorithms have proven to be an efficient method to solve the classical problem of image denoising, though these methods may also remove relevant image content. With a non-local approach there have been a multitude of methods proposed that enhance a restoration algorithm through strengthening a signal with a “post-processing” phase [31, 32, 33]. The post-processing phase can be split into two categories, the first a class of methods which recycle noisy residuals deemed to import any “stolen” image content back into the denoised approximation. The second class of methods deem to strengthen an approximation with a “cleaned” signal. Using a cleaned signal rather than residuals is intended to bypass any unnecessary reintroduction of noise back into our image [22].

Here we focus on work which utilizes a sequential approach such as “Boosting of Non-Local denoising methods” procedure [22] proposed by Romano et al. The “boosting” algorithm proposed by these authors successfully apply the second class of post-processing methods, in which after using their algorithm there is a clear improvement regarding PSNR and visual quality regardless of the image or noise. Here we analyze if it possible

to apply this boosting concept sequentially, with the intent to iteratively improving the quality of the denoised image. As such our choice of non-local method is of course the Non-Local means algorithm [16] which as we saw denoises a given pixel intensity from a weighted average of all pixels in a specified image domain. If the reader needs a refresher regarding the Non-Local means algorithm they are guided to the Literature review earlier in Section 1.5.1 of this thesis.

2.1 Boosting Of Image Denoising Algorithms

Similar to before, we deem a noisy image \underline{y} to be represented as

$$\underline{y} = \mathbf{H}\underline{x} + \underline{n}_\sigma = \underline{x} + \underline{n}_\sigma$$

where \underline{x} is the ideal object, degradation operator \mathbf{H} set as the identity matrix and \underline{n}_σ additive white gaussian noise (AWGN) with σ standard deviation and zero mean. Thus as proposed from the authors of [22] the “**SOS** boosting algorithm” recovers a superior approximation of ideal objective \underline{x} given measurement \underline{y} with the following procedure.

1. Strengthen the signal by adding the previously denoised image and original noisy input.
2. Operate an image denoising algorithm $\mathbf{f}(\cdot)$ on new enhanced signal.
3. Subtract the previously denoised image from the result of 2.

The above series of operations is a single iteration of the **SOS** procedure. Treating the denoising operator \mathbf{f} a “black-box” function, the general boosting method as proposed from [22] is

$$\hat{\underline{x}}^{k+1} = \tau \mathbf{f}(\underline{y} + \rho \hat{\underline{x}}^k) - (\tau\rho + \tau - 1) \hat{\underline{x}}^k$$

where $\hat{\underline{x}}^k$ denotes the approximate solution at iteration k , and $\hat{\underline{x}}^0$ is an initial approximation of \underline{x} . The tuning parameters are ρ which determines the steady state solution and

τ the rate of convergence. Since the boosting operations are sequential and fixed for all iterations, taking $\hat{\underline{x}}^{k+1} = \hat{\underline{x}}^k = \hat{\underline{x}}^*$ we find steady state solution satisfying

$$\hat{\mathbf{x}}^* = ((\rho + 1)I - \rho \mathbf{f})^{-1} \mathbf{f} \cdot \underline{y} \quad (2.1)$$

considering a linearization of the discretized denoising operator \mathbf{f} as $\mathbf{f}(\underline{y}) = \mathbf{f} \cdot \underline{y}$ as in [22]. Of course matrix I is the identity. Thus according to the authors of [22] the solution given from equation (2.1) yields a superior approximation compared to only denoising the signal \underline{y} using denoising operator \mathbf{f} as such $\mathbf{f} \cdot \underline{y}$. From the structure of equation (2.1) we define $\mathcal{B}(\mathbf{f})$ as

$$\mathcal{B}(\mathbf{f}) = ((\rho + 1)I - \rho \mathbf{f})^{-1} \mathbf{f}$$

which is deemed as the “boosted” form of operator \mathbf{f} and can be envisioned as a “new” denoising algorithm compared to only denoising with \mathbf{f} . Now, the question is raised on the applicability of further enhancing the boosting by considering it as a new operator and successively applying the boosting process.

2.2 Iterative “Boosting” Of “Boosting” Operators

We can think of our problem as applying the boosting operation onto itself. As such we begin with deriving this operation by a substitution of $\mathcal{B}(\mathbf{f})$ for linearized denoising filter \mathbf{f} with a fixed steady state parameter ρ for all iterations. Thus we proposed the following theorem which summarizes the result of n consecutive boosting operations.

Theorem 2.2.1. *Given some denoising operator \mathbf{f} , and some fixed parameter ρ with $\mathcal{B}(\mathbf{f}) = ((\rho + 1)I - \rho \mathbf{f})^{-1} \mathbf{f}$ then $\forall n \in \mathbb{N}$*

$$\mathcal{B}^n(\mathbf{f}) = \mathcal{B}(\mathcal{B}(\dots(\mathbf{f}))\dots) = ((\rho + 1)^n(I - \mathbf{f}) + \mathbf{f})^{-1} \mathbf{f}. \quad (2.2)$$

Proof. By induction, taking $n = 1$,

$$\begin{aligned}\mathcal{B}^1(\mathbf{f}) &= ((\rho + 1)^1(I - \mathbf{f}) + \mathbf{f})^{-1} \mathbf{f} = ((\rho + 1)I - \rho\mathbf{f} - \mathbf{f} + \mathbf{f})^{-1} \mathbf{f} \\ &= ((\rho + 1)I - \rho\mathbf{f})^{-1} \mathbf{f}.\end{aligned}$$

Hence, $\mathcal{B}^1(\mathbf{f}) = \mathcal{B}(\mathbf{f}) = ((\rho + 1)I - \rho\mathbf{f})^{-1} \mathbf{f}$. Now, assuming equation (2.2) holds for some positive integer $n = k$, that is

$$\mathcal{B}^k(\mathbf{f}) = ((\rho + 1)^k(I - \mathbf{f}) + \mathbf{f})^{-1} \mathbf{f}, \quad (2.3)$$

we find that

$$\begin{aligned}\mathcal{B}^{k+1}(\mathbf{f}) &= \mathcal{B}(\mathcal{B}^k(\mathbf{f})) \\ &= ((1 + \rho)I - \rho\mathcal{B}^k(\mathbf{f}))^{-1} \mathcal{B}^k(\mathbf{f}) \\ &= \left((1 + \rho)I - \rho \left((1 + \rho)^k(I - \mathbf{f}) + \mathbf{f} \right)^{-1} \mathbf{f} \right)^{-1} \left((1 + \rho)^k(I - \mathbf{f}) + \mathbf{f} \right)^{-1} \mathbf{f} \\ &= \left(\left((1 + \rho)^k(I - \mathbf{f}) + \mathbf{f} \right) \left((1 + \rho)I - \rho \left((1 + \rho)^k(I - \mathbf{f}) + \mathbf{f} \right)^{-1} \mathbf{f} \right) \right)^{-1} \mathbf{f} \\ &= ((1 + \rho)^{k+1}(I - \mathbf{f}) + (1 + \rho)\mathbf{f} - \rho\mathbf{f})^{-1} \mathbf{f}.\end{aligned}$$

This means that the theorem holds for $n = k + 1$. By the principle of arithmetic induction, equation (2.2) is true $\forall n \in \mathbb{N}$. \square

As indicated by Theorem 2.2.1, a sequential application of operator \mathcal{B} with respect to denoising filter \mathbf{f} is possible in one step. As such, successively applying the boosting operator is nothing but a re-parametrization of the expression. That is $(\rho + 1) \mapsto (\rho + 1)^n$, or equivalently $\rho \mapsto (\rho + 1)^n - 1$. We can also observe that if $I - \mathbf{f}$ is invertible and $\rho > 0$, then

$$\lim_{n \rightarrow \infty} \mathcal{B}^n(\mathbf{f}) = \lim_{n \rightarrow \infty} ((\rho + 1)^n(I - \mathbf{f}) + \mathbf{f})^{-1} \mathbf{f} = \lim_{n \rightarrow \infty} ((\rho + 1)^n(I - \mathbf{f}))^{-1} \mathbf{f} = \underline{0}$$

implying that the image would eventually go to black.

2.3 Experiments And Results Regarding Boosting And Remarks

We now investigate the effects of applying the effect of boosting the NLM operator, as well as its consecutive applications on test images. To do so we implemented a sparse representation of the NLM algorithm as denoising operator \mathbf{f} with search window of size 5×5 and smoothing parameter $h = 10\sigma$, where σ is the standard deviation of the additive white noise corrupting ideal object \underline{x} .

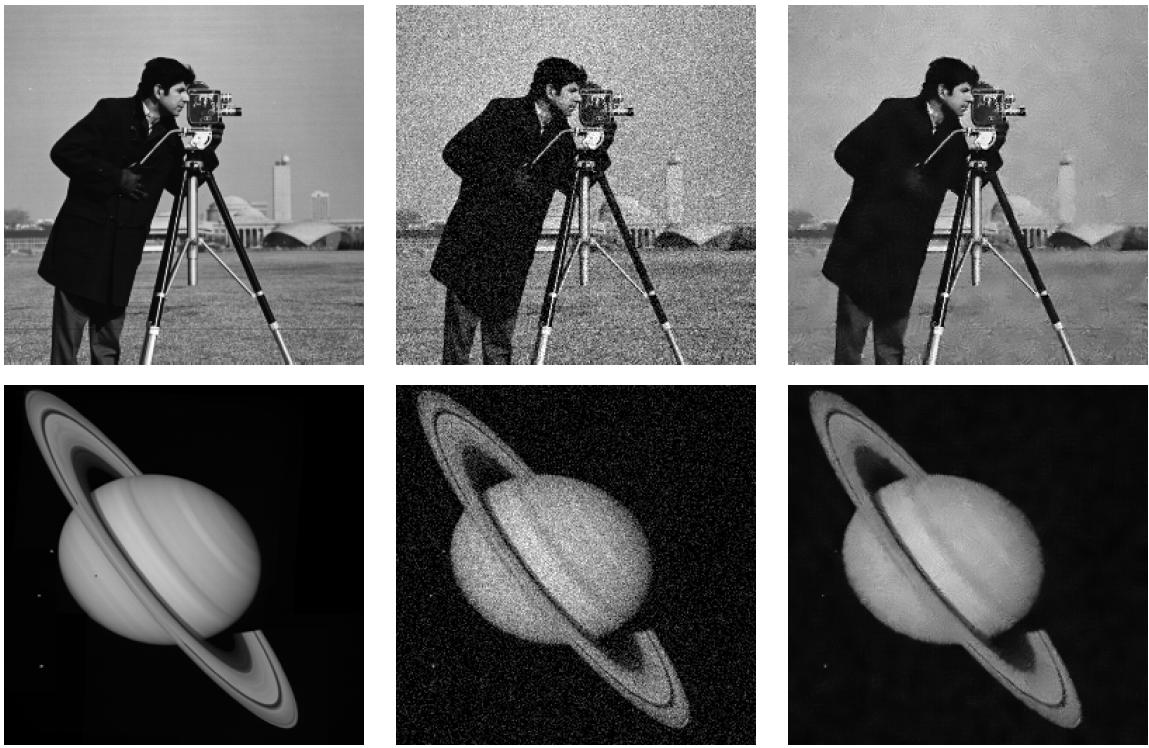


Figure 2.1: (a) Ground truth \underline{x} , (b) Corrupted with AWGN ($\sigma = 2.5$), \underline{y} , (c) NLM restored $\mathbf{f} \cdot \underline{y}$, (d) Ground truth \underline{x} , (e) Corrupted with AWGN ($\sigma = 2.5$) \underline{y} , (f) NLM restored $\mathbf{f} \cdot \underline{y}$.

Observing Figure 2.1 after applying the NLM on a noisy Cameraman and Saturn images corrupted with Additive White Gaussian Noise (AWGN) of $\sigma = 2.5$ there is an increase of overall image quality. We quantitatively measure the performance of each application of boosting $\mathcal{B}^n(\mathbf{f})$ with the PSNR as a function of boosting iteration counter

n . We also define $\mathcal{B}^0(\mathbf{f}) = \mathbf{f}$, i.e the NLM denoising operator.

Curves relating to the computed PSNR of $\mathcal{B}^n(\mathbf{f}) \cdot \underline{y}$ are given in Figures 2.2 and 2.3 respectively for Cameraman and Saturn. For each figure two different values of noise standard deviation, namely $\sigma = 2.5$ and $\sigma = 12.5$ are considered. The parameter ρ is varied within the range of 0.1 to 1.5 for both of these curves. It is interesting to note that for both images, when the noise standard deviation is $\sigma = 12.5$, no boosting can outperform the original NLM algorithm. This fact can be verified by considering the family of curves given in Figures 2.2(a) and 2.3(a) are all decreasing. For smaller noise standard deviation $\sigma = 2.5$, the peak of the curves can be observed in Figures 2.2(b) and 2.3(b). A zoomed in version of both images for the case of $\sigma = 2.5$ can be seen on Figures 2.2(c) and 2.3(c). It is evident that when $\rho = 1.3$ and $\rho = 1.2$ for the first application $n = 1$ corresponds to the peak of the PSNR relating to the two images Cameraman and Saturn for the noise standard deviation $\sigma = 2.5$. Next, Figure 2.5 illustrates the resulting images $\mathcal{B}^n(\mathbf{f}) \cdot \underline{y}$ for a range of n values while fixing the corresponding peak ρ values of $\rho = 1.3$ and $\rho = 1.2$. Finally, Figure 2.4 illustrates the case for which when ρ is less than zero (negative). Observing the trends of the Figure, one notices that negative values of ρ have a chaotic behaviour when compared to the positive ρ counterpart. In addition, when $\rho = -1$ one notes that the boosting operator from Equation 2.3 collapses toward the identity matrix, implying no “boosting” operation on the noisy Cameraman image. We also note that as ρ moves from negative to more positive values, the PSNR increases implying that the choice for $\rho > 0$ for enhancement is a good choice.

Based on these results choosing an optimal value of parameter ρ is a challenging task for the SOS boosting algorithm. In addition we can conclude that the SOS boosting algorithm does not improve the performance of the NLM algorithm at high levels of noise, regardless of parameter ρ . We also show that iterative applications of the SOS boosting algorithm is equivalent to a re-parameterization of parameter ρ . In addition iterative applications of the SOS boosting operator for positive values of ρ will converge to a zero

image (black), implying the SOS boosting as a general remedy for denoising schemes is incorrect. One other interesting note is that when $\rho > 0$, the PSNR will approach toward to a value of zero as counter n increases. This implies that if the operator is normalized (i.e., if it preserves the energy of the image) it will converge to an image with the same energy of the original one (or the image without boosting).

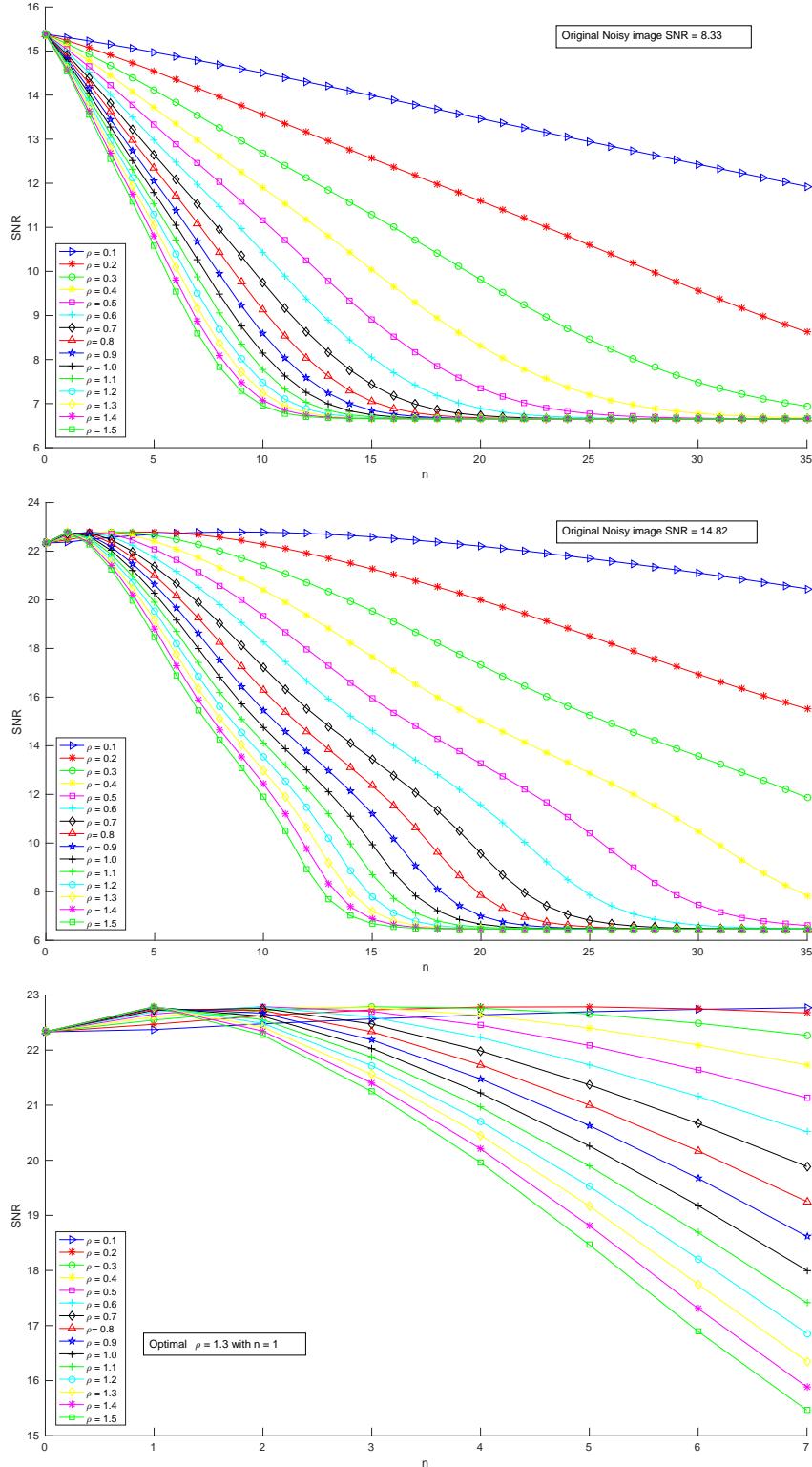


Figure 2.2: Sequential boosting of NLM algorithm under operator $B^n(\mathbf{f})$ applied on noisy Cameraman (Top) AWGN ($\sigma = 12.5$), (Middle) AWGN ($\sigma = 2.5$), (Bottom) Zoomed AWGN ($\sigma = 2.5$) and corresponding PSNR (dB).

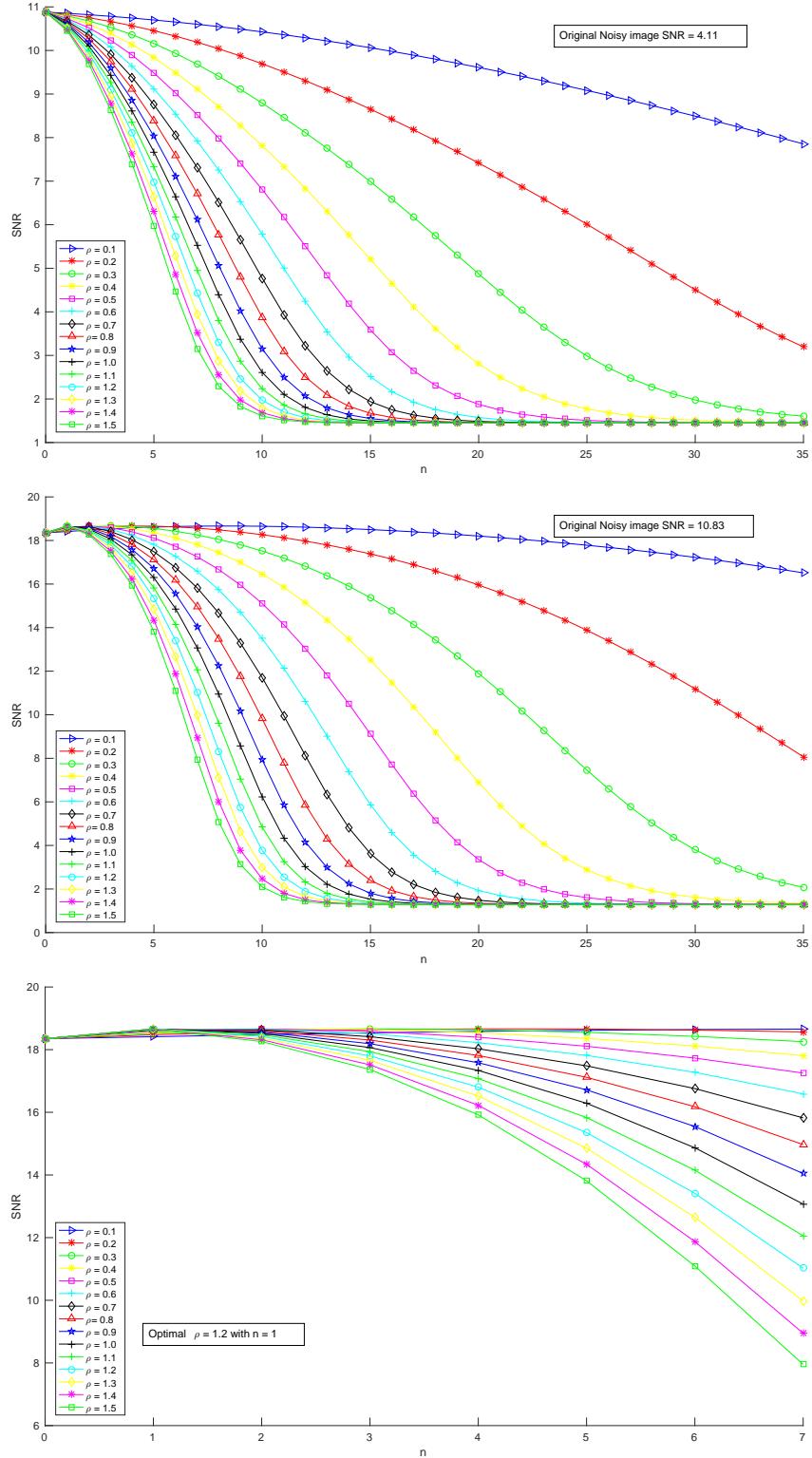


Figure 2.3: Sequential boosting of NLM algorithm under operator $B^n(\mathbf{f})$ applied to noisy Saturn (Top) AWGN ($\sigma = 12.5$), (Middle) AWGN ($\sigma = 2.5$), (Bottom) Zoomed AWGN ($\sigma = 2.5$) and corresponding PSNR (dB).

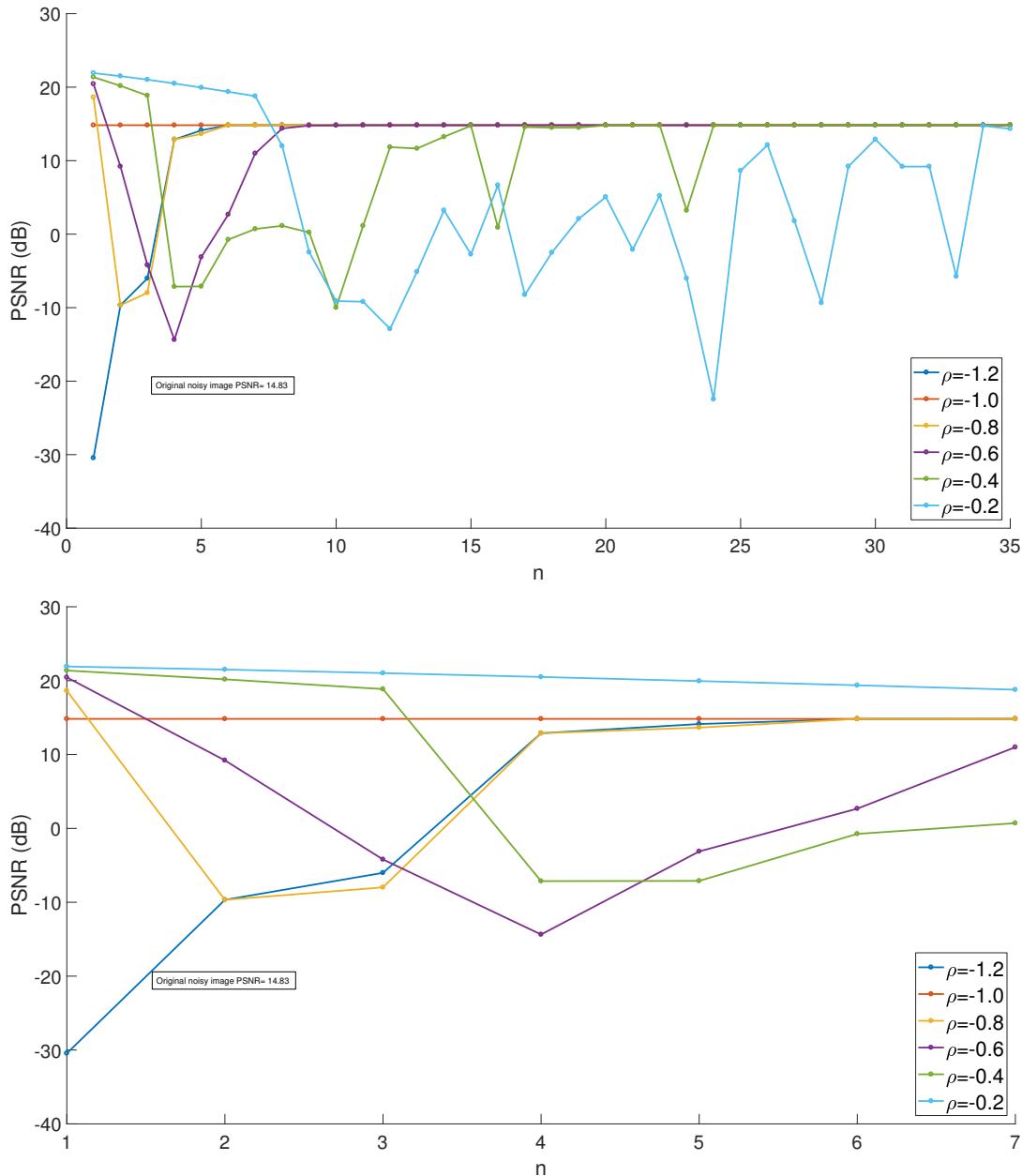


Figure 2.4: Sequential boosting of NLM algorithm under operator $\mathcal{B}^n(\mathbf{f})$ applied on noisy Cameraman corrupted with (Top) AWGN ($\sigma = 2.5$) and (Bottom) Zoomed AWGN ($\sigma = 2.5$) with corresponding PSNR (dB) when ρ is less than zero (negative values).

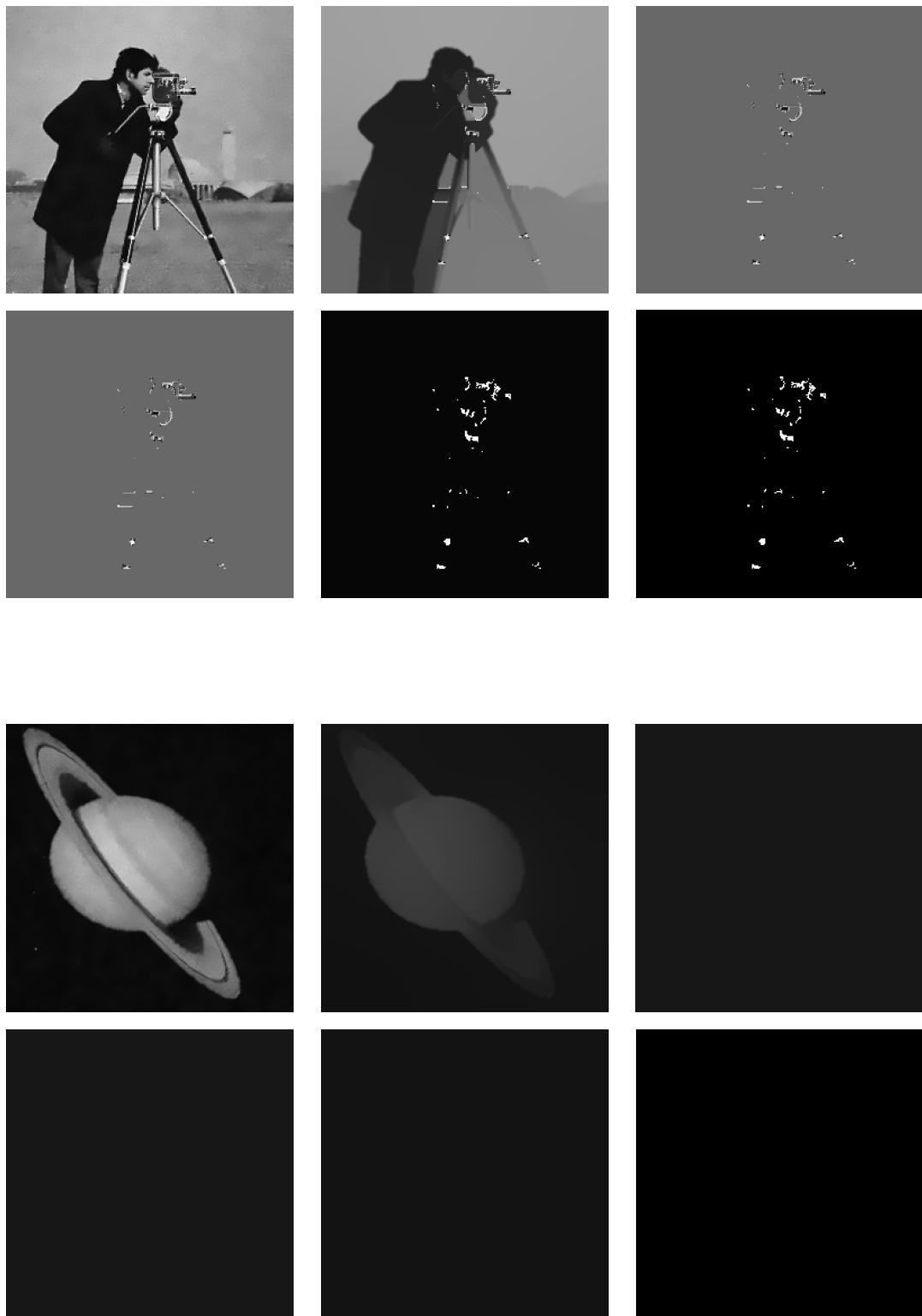


Figure 2.5: Resulting $\mathcal{B}^n(\mathbf{f}) \cdot \underline{y}$ on noisy image \underline{y} with AWGN ($\sigma = 2.5$) for (a) $n = 1$, (b) $n = 10$, (c) $n = 20$, (d) $n = 40$, (e) $n = 50$, (f) $n = 60$, corresponding to the optimal $\rho = 1.3$, (g) $n = 1$, (h) $n = 10$, (i) $n = 20$, (j) $n = 40$, (k) $n = 50$, (l) $n = 60$, corresponding to the optimal $\rho = 1.2$.

Chapter 3

Graph Based Regularization For Image Super Resolution

Interpolation methods are commonly used for resolution enhancement as they are computationally inexpensive and perform well at solving the problem regarding increasing the resolution size of the image [34, 35]. Though interpolation based methods, when used to increase the dimension of an image can result in a degradation of visual clarity. To increase the dimension (resolution) polynomial interpolation is used to bootstrap the Low Resolution (LR) measurement to an appropriate dimension for the corresponding High Resolution (HR) plane. Following interpolation, denoising and deblurring filters were applied to a bootstrapped image to produce a higher resolution image [36]. Since interpolation methods use local pixel information to sample for a new HR grid point, it is highly plausible that if a certain region contains blur, aliasing or edge halos those degradation, will be inherited into the restored image we are attempting to capture. Generally interpolation methods have difficulty extracting high frequency information from LR measurements, since interpolation methods do not utilize any supplementary information when approximating a high resolution solution and thus high frequency information is difficult to recover [21].

A common technique to increase the dimension of low resolution input, as shown in Figure 3.1, are polynomial interpolation methods such as nearest neighbour pixel replication, bicubic interpolation and cubic spline interpolation [37, 38]. A close observation of “super resolution” solutions using methods Nearest Neighbour and BiCubic interpolation illustrate they poorly retain important edge information in the approximation, compared to the required high resolution ground truth displayed on the far right of Figure 3.1. The unreliability of increasing the resolution of low resolution measurement using interpolation techniques is the inability to compute high frequency details due to the lack of supplementary information [39].

3.1 Forward Model And Inverse Problem For Single Image Super Resolution

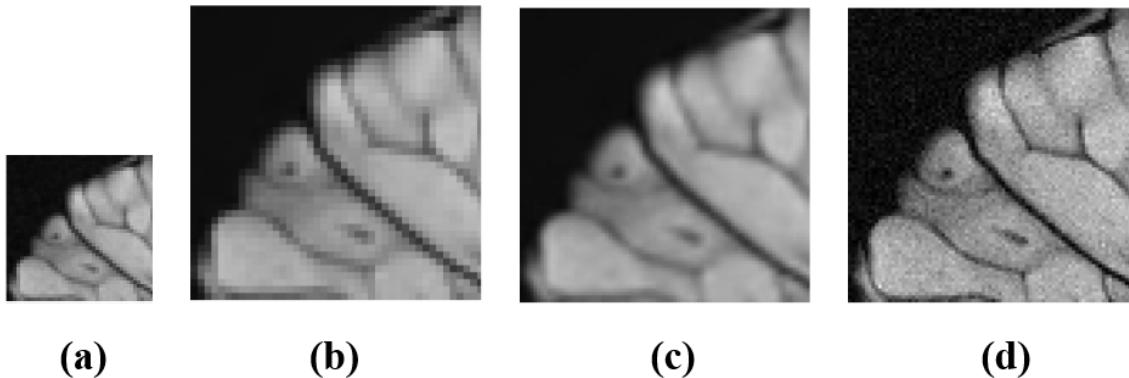


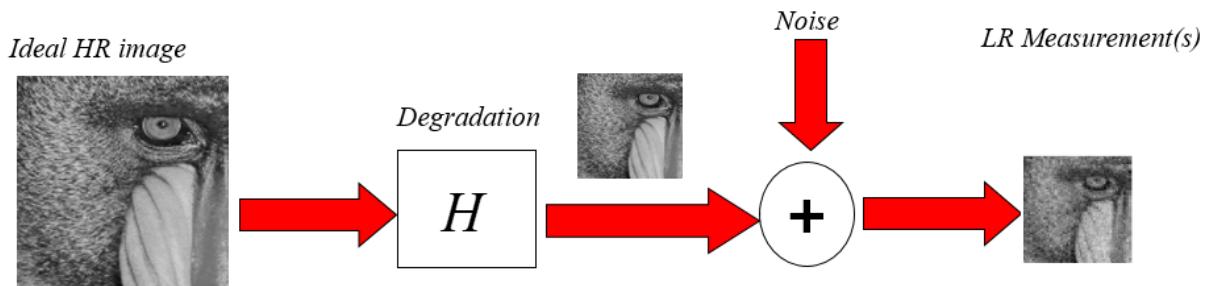
Figure 3.1: (a) Low resolution input. (b) Approximation from pixel Nearest Neighbour interpolation. (c) Approximation using BiCubic interpolation. (d) Desired ground truth.

There have been a variety of approaches to solving the single image super resolution problem, as the usage of standard interpolation methods do not attain the desired high resolution image of a given scene.

Our restoration problem can be thought of creating something from nothing, as we are

essentially creating pixel information for the denser HR plane for which pixel information does not necessarily exist in LR plane. The intent of super resolution methods is to The forward model for our restoration problem is described on the top row of Figure 3.2, where a given low resolution measurement \underline{y} is a product of corrupting an ideal object \underline{x} with degradation operator $\mathbf{H} = \mathbf{DB}$ [40] a composition of blurring \mathbf{B} followed by down sampling \mathbf{D} . Following this, Additive White Gaussian noise (AWGN) denoted as n_σ is applied to corrupted $\mathbf{H}\underline{x}$, where σ represents the standard deviation of the Gaussian white noise vector.

Model:



Problem:

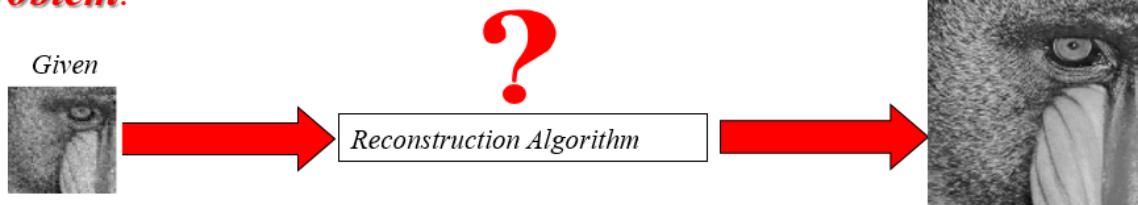


Figure 3.2: Top Row: The Forward model entails given an Ideal HR image and degradation operator \mathbf{H} followed by corruption with additive noise, a low resolution measurement \underline{y} will be produced. Bottom Row: Inverse problem for approximating a high resolution solution given low resolution measurement \underline{y} .

In this work our restoration process can be thought of solving an inverse problem involving a single measurement \underline{y} , what reconstruction algorithms are at our disposal to create a high resolution approximation of ideal object (or scene we wish to capture) \underline{x} . For visual sake, the forward model for our restoration problem is given on the top row of Figure 3.2 and equivalent inverse problem can be seen on the bottom row of Figure 3.2.

As a note the forward model from Figure 3.2 is the synthetic case where both blurring and down sampling operators are known or can be accurately estimated. An important distinction is that the data we are attempting to restore is not blurry upon acquisition, but rather corrupted with chosen degradation operators. An alternative is the interesting case for which blurring and down sampling operators are not known *a priori*.

One can think of solving an inverse problem as the process of replicating the causes of corruption when sampling measurement \underline{y} with respect to ideal object \underline{x} . That is, we would like to find a cause and effect relationship between our low resolution and high resolution images. Since our solution depends on degradation operator \mathbf{H} , we show the steps for how one could replicate said operator for the case of single image super resolution (SISR). For simplicity we can assume that both LR measurement \underline{y} and HR approximation \underline{x} are square images, i.e $N = M$ (number of rows are equal to the number of columns). We denote degradation operator \mathbf{H} implemented as a sparse matrix to be a composition of blurring \mathbf{B} followed by down sampling \mathbf{D} , which are both represented in a discrete implementation. Then, in our case degradation is represented as

$$\mathbf{H} = \mathbf{DB} \in \mathbb{R}^{(N^2) \times (L^2N^2)}$$

where $\mathbf{D} \in \mathbb{R}^{(N^2) \times (L^2N^2)}$, and $\mathbf{B} \in \mathbb{R}^{(L^2N^2) \times (L^2N^2)}$. Operator \mathbf{D} is responsible for dimension reduction, where after applying \mathbf{D} to blurring \mathbf{B} the resulting degradation operator \mathbf{H} is no longer square. With this result, matrix \mathbf{H} is non-invertible and our linear system is no longer guaranteed a unique solution, implying an ill-posed inverse problem.

We know that from the composition of degradation \mathbf{H} , our inverse problem will require regularization. In general regularization, for image restoration falls into two classes. The first are stochastic methods which utilize information from the covariance matrices of the ground truth image (ideal object), as well as the image noise[41]. The second class of regularization methods fall under a deterministic approach(s) where natural images are deemed relatively featureless with minimal high frequency activity in the data set [42, 43]. Restricting our *a priori* information to a single measurement naturally does not

aid toward the uniqueness of our high resolution and only further complicates things, as is the reality of our inverse problem. Currently one of the most successful approaches to solving the single image super resolution inverse problems entails an example based approach, which small image patches i.e, (5×5) exist in an image domain Ω of scale equivalent to, or larger than a given patch. We can utilize this redundancy to establish a “database of weights” is then used to restore the low-resolution measurement in such a way that following the restoration, a high-resolution version will be produced [43]. Given the non-local composition of the normalized graph Laplacian [24] and patched based “dictionary” from composition of similarity matrices, this operator is a strong candidate for a powerful regularization operator for our single image resolution inverse problem.

3.2 Regularization With The Normalized Graph Laplacian

In the literature review we saw that the normalized graph Laplacian performed well to regularize image restoration functions designed to balance data fidelity and regularization functionals that produce a denoised version of AWGN noisy low resolution input [24]. In general to compute the restored solution \underline{x} (with the normalized graph Laplacian chosen as a regularizer derived accordingly according to Section 1.3.1) the following objective is minimized

$$\begin{aligned} E(\cdot, \underline{y}) &= \underset{\underline{x}}{\operatorname{argmin}} \left((\underline{y} - \mathbf{H}\underline{x})^T (\underline{y} - \mathbf{H}\underline{x}) + \eta \mathcal{R}_n(\underline{x}) \right) \\ &= \underset{\underline{x}}{\operatorname{argmin}} \left((\underline{y} - \mathbf{H}\underline{x})^T (\underline{y} - \mathbf{H}\underline{x}) + \eta \sum_{u \in V} \sum_{v \sim u} w(u, v) \left(\frac{\underline{x}(u)}{\sqrt{D(u, u)}} - \frac{\underline{x}(v)}{\sqrt{D(v, v)}} \right)^2 \right) \\ &= \underset{\underline{x}}{\operatorname{argmin}} \left((\underline{y} - \mathbf{H}\underline{x})^T (\underline{y} - \mathbf{H}\underline{x}) + \eta \left(\underline{x}^T \left(I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \right) \underline{x} \right) \right) \end{aligned} \quad (3.1)$$

with η a tuning parameter and degradation operator \mathbf{H} dependent on the posed restoration problem (denoising, deblurring, etc.). In this section we review the derivation of a

new normalized graph Laplacian for the problem of image deblurring opposed to normalized graph Laplacian $I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ used for denoising in [24]. More related to our proposed inverse problem the authors of [23] adapted the definition of the normalized graph Laplacian to regularize an energy functional for image deblurring and sharpening. Their energy functions includes a variation of sparse block similarity matrix w denoted as sparse similarity matrix K derived according to the Non-Local means algorithm [16]. Once similarity matrix K is computed, normalization follows by diagonal scaling matrix which creates a doubly symmetric filtering matrix denoted as W . There is also an inclusion of a scaling term on data fidelity term, which is deemed to sharpen the residuals of the data fidelity term. As seen in Figure 3.3 using their proposed algorithm on a blurry girl image computes a superior PSNR and SSIM compared to utilizing traditional normalized graph Laplacian as a regularizer. Due to the superior performance of their proposed method we will adapt this regularizer from [23] for our super resolution inverse problem. We will address more regarding the properties, benefits and full derivation of proposed deblurring algorithm later in this dissertation.



Figure 3.3: Deblurring examples with blurred noisy Girl image by out-of-focus blur with radius 7 and additive white Gaussian noise with standard deviation $\sigma = 1$: (a) clean image, (b) blurred noisy image, (c) output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 29.40 dB, SSIM = 0.8734), and (d) output of proposed deblurring algorithm (PSNR = 30.58 dB, SSIM = 0.9058).

According to the algorithm proposed by the authors of [23] to deblur a measurement \underline{y} the follow objective is minimized

$$E(\cdot, \underline{y}) = \underset{\underline{x}}{\operatorname{argmin}} \left((\underline{y} - \mathbf{H}\underline{x})^T (I + \beta(I - W)) (\underline{y} - \mathbf{H}\underline{x}) + \eta \underline{x}^T (I - C^{-\frac{1}{2}} K C^{-\frac{1}{2}}) \underline{x} \right) \quad (3.2)$$

with respect to unknown vector \underline{x} where matrix $K = K(u, v)$ measures similarity between edges $\{u, v\}$, \mathbf{H} a symmetric blur operator and C a diagonal scaling matrix. In the case for deblurring degradation operator \mathbf{H} and its transpose \mathbf{H}^T are symmetric blurring matrices. We proceed to analyze the doubly stochastic filtering matrix W composed from similarity weight matrix K with normalization by diagonal scaling matrix C . In addition one should note the inclusion of a “sharpening” term which scales the data fidelity term of the deblurring energy function. Tuning parameters η and β are set depending on the amount of blur and noise respectively of low resolution \underline{y} . To compute similarity matrix K the weight function chosen is derived from the Non-Local means denoising algorithm [16]. Thus given the (u, v) th element of the edges $E \subset V \times V$ the NLM based weight matrix K is given by

$$K(u, v) = \exp \left(-\frac{\|\tilde{\underline{y}}_u - \tilde{\underline{y}}_v\|^2}{h^2} \right) \quad (3.3)$$

where $\tilde{\underline{y}}_u$ is the patch surrounding current target pixel $u = (i, j)$ of measurement \underline{y} and $\tilde{\underline{y}}_v$ the patch surrounding neighbouring vertex v of target pixel u according to the set of vertices defined from equation (1.7). Finally h is a smoothing parameter. The NLM algorithm is utilized not only to measure similarity between pixels but is also responsible for establishing the edges for graph G_k .

To visualize the assignment of neighbouring vertices given a target vertex u , we refer to the illustration of the NLM weight schematic in Figure 3.4. Given a target vertex $u \in V$ illustrated as a red dot and corresponding intensity gray level vector (orange block surrounding target vertex) only nodes $v \in V$ with pixel locations belonging to search window (green region) will be considered a neighbouring vertex of target pixel u and have an established edge $\{u, v\} \in E$ for graph G_k . As such the size of green

patch from Figure 3.4 is set according to search window radius of the NLM algorithm. Of course one could vary the radius of search window to include more pixels of image domain Ω , though is usually set to ensure only a small neighbourhood (eg. 11×11) around target vertex u will establish an edge $\{u, v\} \in E$. Finally the yellow border surrounding search window represent assigned boundary conditions usually symmetric. Following the composition of matrix K the next step is to normalize with C a diagonal scaling matrix composed from applying the Sinkhorn-Knopp balancing algorithm [44] to positive definite symmetric matrix K . The Sinkhorn-Knopp (**SK**) algorithm is an iterative method which will alternately rescales all rows and columns of matrix K to sum to a value of 1 until convergence. We now state the Sinkhorn theorem, the premise which the Sinkhorn-Knopp algorithm is composed from.

Theorem 3.2.1. *Sinkhorn's Theorem If matrix A is an $N \times N$ matrix with strictly positive elements, then there exists diagonal matrices D_1 and D_2 with strictly diagonal elements such that D_1AD_2 is a doubly stochastic matrices. The matrices D_1 and D_2 unique up to a multiplicative factor in the sense that multiplying the first matrix by some constant and dividing the latter by the same constant.*

As seen diagonal scaling matrix C replaces degree matrix D when compared to traditional normalized Laplacian seen in the denoising example from [24]. After scaling similarity matrix K with diagonal matrix C the resulting filtering matrix is given by

$$W = C^{-\frac{1}{2}}KC^{-\frac{1}{2}} \quad (3.4)$$

which is doubly stochastic, symmetric and positive definite. This definition of the graph Laplacian has superior spectral properties than its degree normalized and un-normalized counterpart [30]. Decomposing symmetric filtering matrix as $W = \Gamma J \Gamma^T$ where Γ is the orthonormal matrix with columns representing the eigenvectors of filtering matrix W and $J = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{N^2}\}$ their corresponding eigenvalues. From the authors of the deblurring algorithm [23] given that filtering matrix W is doubly stochastic associated

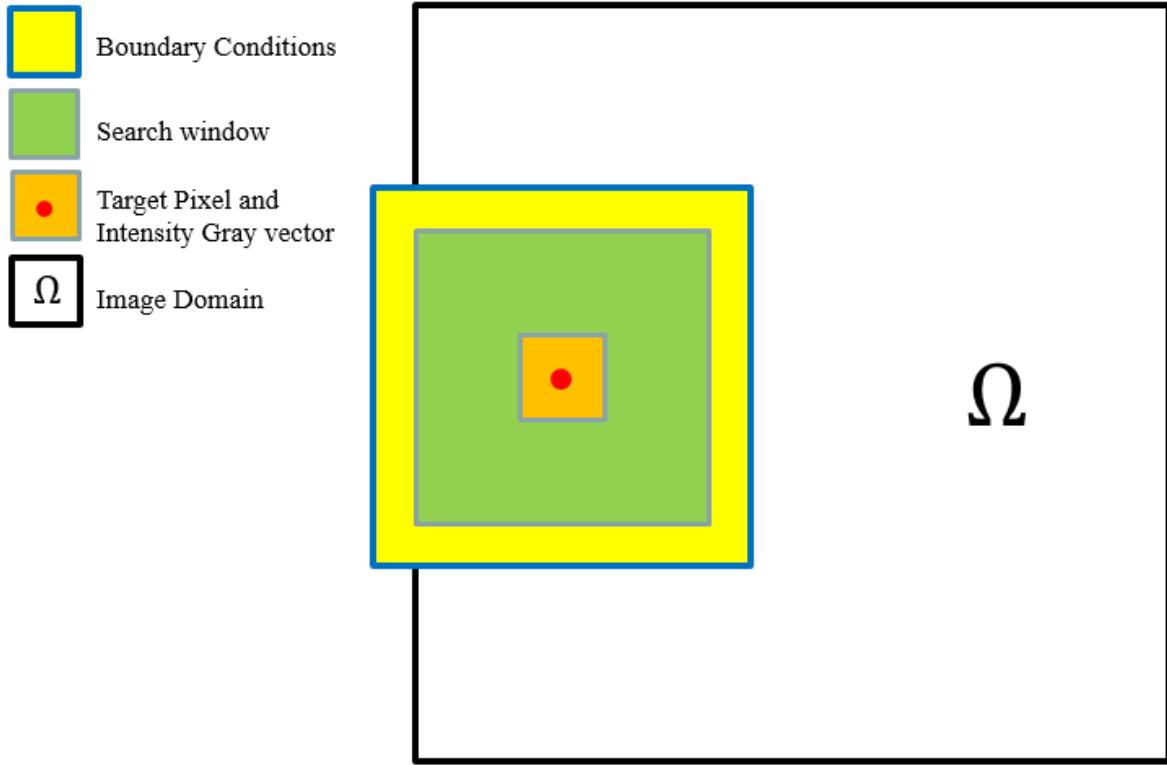


Figure 3.4: Non-Local means algorithm neighbour assignment schematic.

eigenvalues follow $\lambda_1 = 1 > \lambda_2 \geq \dots \geq \lambda_{N^2} \geq 0$ implying that largest eigenvalue is exactly equal to one with corresponding eigenvector $\mathbf{v}_1 = (\frac{1}{\sqrt{N^2}})(\mathbf{1}_{N^2})^T$. This is a favorable property since it ensures the preservation of DC component of signal after applying filtering matrix W [45].

Following the composition of filtering matrix W it is now plausible to define the new normalized graph Laplacian

$$I - C^{-\frac{1}{2}} K C^{-\frac{1}{2}} = I - W \quad (3.5)$$

which is sparse, symmetric, positive semi-definite and when applied to a constant returns the zero vector $\mathbf{0}_{N^2}$. This is the true definition of the normalized graph Laplacian for image restoration purposes according to [23]. The Laplacian operator $I - W$ is a high pass filter, and furthermore when applied to an image is equivalent to a data-adaptive Laplacian filter. The Laplacian operator is an important ingredient for the composition

of the sharpening filter applied to the data fidelity term, which will be explored shortly.

Now that we have the normalized Laplacian, we will demonstrate to the reader the difference operators associated with converting this new definition of the Graph Laplacian into a regularization functional. Similar to before we begin with the difference operator

$$dx(u, v) = \sqrt{K(u, v)} \left(\frac{\underline{x}(u)}{\sqrt{C(u, u)}} - \frac{\underline{x}(v)}{\sqrt{C(v, v)}} \right) \quad (3.6)$$

with $K(u, v)$ the similarity matrix composed according to the weight measure from the NLM algorithm and $C(u, u)$ and $C(v, v)$ elements of diagonal scaling matrix C for vertices u and v respectively. As before since we are working with a multidimensional function the gradient is given as

$$\nabla(x(u)) = [dx(u, v_1), \dots, dx(u, v_m)]^T, \quad \forall(u, v) \in E.$$

Thus with the definition of the divergence operator and the normalized graph Laplacian $I - C^{-\frac{1}{2}}KC^{-\frac{1}{2}}$ the proposed regularization functional is

$$R(\underline{x}) = \eta \sum_{u \in V} \sum_{v \sim u} K(u, v) \left(\frac{\underline{x}(u)}{\sqrt{C(u, u)}} - \frac{\underline{x}(v)}{\sqrt{C(v, v)}} \right)^2 = \eta \underline{x}^T (I - W) \underline{x} \quad (3.7)$$

with $v \sim u$ indicates vertices v considered a neighbour of target node u according to edges $\{u, v\} \in E$. Given the deblurring objective from [23] the regularization functional given in equation (3.2) will penalize high frequencies of the low resolution measurement.

From here we guide the reader on the composition of regularization term for deblurring energy functional (3.2) and also review inclusion of a sharpening operator applied to data fidelity term

$$\mathcal{D}(\underline{y}, \underline{x}) = (\underline{y} - \mathbf{H}\underline{x})^T (I + \beta (I - W)) (\underline{y} - \mathbf{H}\underline{x}) \quad (3.8)$$

opposed to standard ℓ^2 -norm $\|\underline{y} - \mathbf{H}\underline{x}\|^2$ corresponding to parameter $\beta = 0$. We reiterate that the purpose of data fidelity operator is to ensure the blurred and filtered solution \underline{x} is close as possible to filtered input \underline{y} . Ensuring tuning parameter $\beta \geq 1$ yields symmetric

positive definite matrix

$$I + \beta(I - W)$$

which acts as a sharpening filter on residuals of $\underline{y} - \mathbf{H}\underline{x}$ since matrix $I - W$ behaves similar to a high pass filter. With this inclusion of sharpening filter the restored solution \underline{x} should maintain fine details in the restored image while limiting the overall visual degradation from single low resolution measurement \underline{y} . Shortly we will address the adaptive behavior of the sharpening filter once we arrive at the algorithm breakdown of the deblurring method.

Now that both fidelity and regularization terms are defined we guide the reader on steps to solve the deblurring energy function (3.2). Setting gradient $\nabla E(\cdot, \underline{y}) = \underline{0}$ yields the following set of symmetric positive definite linear system of equations

$$-2\mathbf{H}^T (I + \beta(I - W)) (\underline{y} - \mathbf{H}\underline{x}) + 2\eta(I - W)\underline{x} = \underline{0}$$

or written as an expression for \underline{x} as

$$\left(\mathbf{H}^T (I + \beta(I - W)) \mathbf{H} + \eta(I - W) \right) \underline{x} = \mathbf{H}^T (I + \beta(I - W)) \underline{y}. \quad (3.9)$$

Since the linear system (3.9) is symmetric and positive definite, it can be solved optimally (avoiding computing any matrix inverses) using the conjugate gradient descent algorithm.

3.3 Single Image Super Resolution Utilizing A Graph Laplacian Operator As A Penalizer

We now introduce our energy function we wish to solve for our single image super resolution inverse problem

$$S(\cdot, \underline{y}) = \operatorname{argmin}_{\underline{x}} (\underline{y} - \mathbf{H}\underline{x})^T (I_S + \beta(I_S - W_S)) (\underline{y} - \mathbf{H}\underline{x}) + \eta \underline{x}^T (I_L - W_L) \underline{x} \quad (3.10)$$

which once minimized will produce \underline{x} the high resolution solution we aim to capture from our single low resolution measurement \underline{y} . As before terms β and η are tuning parameters dependent on amount of noise and blur of input measurement \underline{y} , matrix \mathbf{H} a degradation operator and finally filtering matrix W . Before proceeding to the breakdown of associated normalized graph Laplacian operators used for composition, our proposed objective (3.10) has a few concerns that must be addressed.

1. Operator \mathbf{H} is now a composition of down sampling and deblurring which converts our linear equations to an underdetermined system.
2. The filtering matrix W is now unique across the energy function and can no longer be used to compose both regularizer and data fidelity terms.

This is a natural complication of the inverse problem we wish to solve. We remind the reader the degradation operator for our inverse problem is

$$\mathbf{H} = \mathbf{DB} \in \mathbb{R}^{(N^2) \times (L^2 N^2)}$$

with decimation operator $\mathbf{D} \in \mathbb{R}^{(N^2) \times (L^2 N^2)}$ and local averaging blurring operator $\mathbf{B} \in \mathbb{R}^{(L^2 N^2) \times (L^2 N^2)}$.

We clarify the dimension and role of each term regarding our energy function (3.10) with the table above 3.1. According to Table 3.1 the dimensionality and role of the degradation operator \mathbf{H} after applied to a vector $\underline{v} \in \mathbb{R}^{N^2 \times 1}$ the number of rows according to decimation factor L will be less than the number of columns, implying we must solve an underdetermined linear system. Our goal is to create a single unique solution which will be reminiscent of measurement \underline{y} . Naturally an inclusion of decimation in degradation operator \mathbf{H} creates a dimensionality concern for filtering matrix W compared to the earlier deblurring energy functional (3.2) proposed by Kheradmand et.al[23]. Given this result we now find a way to simulate filtering matrix W_L before composing our normalized graph Laplacian regularizer. The filtering matrix W_L should be composed

from an approximation of high resolution solution \underline{x} according to initial measurement \underline{y} . As such we state an algorithm that will initiate the construction of doubly stochastic filtering matrix W_L . The algorithm steps include

1. Estimate noise standard deviation σ from \underline{y} .
2. Denoise input \underline{y} using BM3D denoising algorithm.
3. Using denoised \underline{y} upsample using B-spline interpolation to simulate higher dimension \underline{y}_L which coincides with decimation factor L (full composition of decimation operator \mathbf{D} can be found in Appendix A)
4. Compute filtering matrix W_L using simulated high resolution image \underline{y}_L .

We choose to interpolate using splines rather than polynomials, as polynomials are often undesirably global. This implies observations at one range of the pixels will have a strong influence on what the polynomial does at a different range, which could lead to unbounded error given the LR measurement. After composing filtering matrix W_L for the regularizer of our objective, we now turn our attention to W_S calculated for the data fidelity term. Filtering matrix W_S is composed after denoising measurement \underline{y} and extracting similarity matrix K_S . This matrix unlike K_L is not symmetric and thus does not guarantee the preservation of DC component of signal when converted to filtering matrix W_S . Now that we have addressed some preliminary concerns to implementing the normalized graph Laplacian for our inverse problem, we begin with the derivation of proposed similarity block matrices used to compose both regularization and data fidelity functionals.

Term	Role	Dimension
Associated with input		
\underline{y}	Corrupted LR input	$N^2 \times 1$
\mathbf{D}	Decimation operator	$N^2 \times L^2 N^2$
\mathbf{B}	Blurring operator	$L^2 N^2 \times L^2 N^2$
$\mathbf{H} = \mathbf{DB}$	Degradation operator	$N^2 \times L^2 N^2$
Associated with data fidelity \mathcal{D}		
β	Tuning parameter	1×1
I_S	Identity matrix	$N^2 \times N^2$
W_S	Filtering matrix for \mathcal{D}	$N^2 \times N^2$
Associated with regularization \mathcal{R}		
η	Tuning parameter	1×1
I_L	Identity matrix	$L^2 N^2 \times L^2 N^2$
W_L	Filtering matrix for \mathcal{R}	$L^2 N^2 \times L^2 N^2$
Associated with output		
\underline{x}	Reconstructed HR estimate	$L^2 N^2 \times 1$

Table 3.1: Dimension of each term composing proposed objective function (3.10) under the assumption that images are square (i.e $N \times N$) and in lexicographical notation with decimation factor denoted by L .

3.4 Computation Of Similarity Block Matrices And Associated Filtering Operator

After pre-processing low resolution input \underline{y} with the BM3D denoising algorithm we expect the image noise will be minimized and thus result in a better representation of important edge information when computing the similarity weight matrix K . Thus after denoising the noisy signal \underline{y} we compute two block similarity matrices for every edge $\{u, v\}$ using the similarity measure derived from the [16] Non-Local means denoising algorithm. Thus (u, v) -th element of similarity matrix for the data fidelity term is then defined as

$$K_S(u, v) = \exp \left(-\frac{\|\tilde{\underline{y}}_u - \tilde{\underline{y}}_v\|^2}{h^2} \right) \quad (3.11)$$

where $\tilde{\underline{y}}_u$ is the intensity patch surrounding current target vertex (or pixel) $u = (i, j)$ of BM3D filtered measurement \underline{y} and $\tilde{\underline{y}}_v$ the patch surrounding neighbouring vertex v of target pixel u according to the set of vertices defined from equation (1.7). Regarding our second similarity matrix K_L used to compose the regularization functional

$$K_L(u, v) = \exp \left(-\frac{\|\tilde{\underline{z}}_u - \tilde{\underline{z}}_v\|^2}{h^2} \right) \quad (3.12)$$

where $\tilde{\underline{z}}_u$ is the intensity patch surrounding target vertex (or pixel) $u = (i, j)$ of BM3D filtered bootstrapped measurement \underline{y}_L and $\tilde{\underline{z}}_v$ the patch surrounding neighbouring vertex v of target pixel u according to the set of vertices defined from equation (1.7).

Due to the positive nature of intensity gray neighbourhoods all similarity weights are positive inferring similarity matrices are positive definite. As before calculating similarity measures between edges $\{u, v\}$ are restricted to a finite set of neighbouring pixels around it according to the search window radius of NLM algorithm, thus K_L and K_S are sparse by composition. Similar to before after calculating similarity matrices K_L and K_S we apply the Sinkhorn-Knopp (**SK**) balancing algorithm [44] (derived according to Theorem 3.2.1) which returns matrices

$$C_S = \mathbf{SK}(K_S), \quad C_L = \mathbf{SK}(K_L).$$

where C_S and C_L are diagonal scaling matrices composed from K_S and K_L respectively. Given diagonal scaling matrices above we can compute required doubly stochastic filtering matrices as

$$W_L = I_L - C_L^{-\frac{1}{2}} K_L C_L^{-\frac{1}{2}}, \quad W_S = I_S - C_S^{-\frac{1}{2}} K_S C_S^{-\frac{1}{2}}.$$

According to the authors of [23] this is the correct definition of the normalized Laplacian for image processing purposes, as compared to the traditional version $I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$. This is due to the fact that the traditional normalized Laplacian does not have property of the zero eigenvalue having associated constant eigenvector $\frac{1}{\sqrt{N^2}}(1_{N^2})$ which is a beneficial property for filtering purposes. This property implies that when normalized Laplacian filter is applied to a constant function \underline{x} the zero vector is returned. Adapted to fit our inverse problem the difference operator of a function $\underline{x} : V \rightarrow \mathbb{R}^+$ on an edge $\{u, v\}$ of G_k is given by

$$(d\underline{x})(u, v) = \sqrt{K_L(u, v)} \left(\frac{\underline{x}(u)}{\sqrt{C_L(u, u)}} - \frac{\underline{x}(v)}{\sqrt{C_L(v, v)}} \right)$$

with $C_L(u, u)$ and $C_L(v, v)$ associated vertices u and v -th diagonal scaling normalization values. It follows then the weighted gradient vector of a function \underline{x} at vertex $u \in V$ is

$$\nabla \underline{x}(u) \approx (dx(u, v_1), \dots, dx(u, v_m))^T, \quad \forall (u, v) \in E. \quad (3.13)$$

Taking the divergence operator of equation (3.13) yields the Laplace for a single vertex $u \in V$ with respect to normalized graph Laplacian $I_L - C_L^{-\frac{1}{2}} K_L C_L^{-\frac{1}{2}}$ as

$$\Delta \underline{x}(u) = \frac{1}{\sqrt{C_L(u, u)}} \sum_{v \sim u} K_L(u, v) \left(\frac{\underline{x}(u)}{\sqrt{C_L(u, u)}} - \frac{\underline{x}(v)}{\sqrt{C_L(v, v)}} \right). \quad (3.14)$$

Thus putting all ingredients previously defined in this section together we can write our regularization functional as

$$R(\underline{x}) = \eta \sum_{u \in V} \sum_{v \sim u} K_L(u, v) \left(\frac{\underline{x}(u)}{\sqrt{C_L(u, u)}} - \frac{\underline{x}(v)}{\sqrt{C_L(v, v)}} \right)^2 = \eta \underline{x}^T (I_L - W_L) \underline{x} \quad (3.15)$$

with η a tuning parameter. In our method we use filtering matrix W_S to compose the data fidelity term

$$D(\underline{x}, \underline{y}) = (\underline{y} - \mathbf{H}\underline{x})^T (I_S + \beta(I_S - W_S)) (\underline{y} - \mathbf{H}\underline{x})$$

which includes a scale of residuals by sharpening operator $I_S + \beta(I_S - W_S)$ and β a tuning parameter. We will now describe how to implement both regularization and data fidelity functionals to solve our inverse problem, and in the process develop a new single image super resolution restoration algorithm. Recall that the energy function we wish to solve is composed from above defined regularization and fidelity functionals

$$S(\cdot, \underline{y}) = \underset{\underline{x}}{\operatorname{argmin}} \left((\underline{y} - \mathbf{H}\underline{x})^T (I_S + \beta(I_S - W_S)) (\underline{y} - \mathbf{H}\underline{x}) + \eta \underline{x}^T (I_L - W_L) \underline{x} \right) \quad (3.16)$$

where W_S and W_L are filtering matrices for composition of data fidelity and regularization term respectively, \underline{y} low resolution measurement \underline{x} high resolution solution, $\mathbf{H} = \mathbf{DB}$ the degradation operator for our specific inverse problem. Again parameters β and η tuned for amount of noise and blur of input \underline{y} . We can simplify our energy function using following change of variables

$$\mathbf{P} = I_S + \beta(I_S - W_S), \quad \mathbf{Q} = \eta(I_L - W_L).$$

Obviously since matrix \mathbf{P} is composed according to a sum between identity matrix and parameter scaled difference of identity with filtering matrix W_S the dimension of sharpening \mathbf{P} remain consistent regarding dimensionality of filtering matrix $W_S \in (\frac{N}{k})^2 \times (\frac{N}{k})^2$ with k the factor regarding decimation operator \mathbf{D} . The same idea can be applied regarding dimensions to regularizer term \mathbf{Q} . Filling in our change of variables from above for equation (3.16) is written as

$$\begin{aligned} S(\underline{x}, \underline{y}) &= (\underline{y} - \mathbf{H}\underline{x})^T \mathbf{P} (\underline{y} - \mathbf{H}\underline{x}) + \underline{x}^T \mathbf{Q} \underline{x} \\ &= \underline{y}^T \mathbf{P} \underline{y} - (\underline{y}^T \mathbf{P} \mathbf{H} \underline{x})^T - \underline{x}^T \mathbf{H}^T \mathbf{P}^T \underline{y} + \underline{x}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \underline{x} + \underline{x}^T \mathbf{Q} \underline{x} \\ &= \underline{y}^T \mathbf{P} \underline{y} - \underline{x}^T \mathbf{H}^T \mathbf{P}^T \underline{y} - \underline{x}^T \mathbf{H}^T \mathbf{P}^T \underline{y} + \underline{x}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \underline{x} + \underline{x}^T \mathbf{Q} \underline{x} \\ &= \underline{y}^T \mathbf{P} \underline{y} - 2\underline{x}^T \mathbf{H}^T \mathbf{P}^T \underline{y} + \underline{x}^T \mathbf{H}^T \mathbf{P} \mathbf{H} \underline{x} + \underline{x}^T \mathbf{Q} \underline{x} \end{aligned} \quad (3.17)$$

where $\underline{y}^T \mathbf{P} \mathbf{H} \underline{x}$ is a 1×1 matrix, and therefore it's own transpose.

This is our new objective function we wish to minimize thus taking the gradient of (3.17) and setting $\nabla S = \underline{0}$ yields

$$-2\mathbf{H}^T \mathbf{P}^T \underline{y} + 2\mathbf{H}^T \mathbf{P} \mathbf{H} \underline{x} + 2\mathbf{Q} \underline{x} = \underline{0}$$

which results in the following set of positive definite system $(\mathbf{H}^T \mathbf{P} \mathbf{H} + \mathbf{Q}) \underline{x} = \mathbf{H}^T \mathbf{P}^T \underline{y}$ or solving for \underline{x} ,

$$\underline{x} = (\mathbf{H}^T \mathbf{P} \mathbf{H} + \mathbf{Q})^{-1} \mathbf{H}^T \mathbf{P}^T \underline{y}, \quad \mathbf{P} = I_S + \beta(I_S - W_S) \quad (3.18)$$

with restriction on parameter β (see composition of matrix \mathbf{P} on pg.50) to ensure linear system of equations is positive definite. We solve above system of linear equations with an iterative scheme

$$\underline{x}_{k+1} = \underline{x}_k - \gamma_k \underline{d}_k(\underline{x}_k), \quad k \geq 0$$

where γ_k and $\underline{d}_k(\underline{x}_k)$ are based upon conjugate gradient descent (full method provided in Appendix C) and initial input $\underline{x}_0 \in \mathbb{R}^{L^2 N^2 \times 1}$, a denoised version of low resolution input \underline{y} . At this point, it is important to understand that the ill-posedness of our inverse problem has multiple contributing factors. As an example, the decimation operator \mathbf{D} implies a non-unique solution, and moreover, we wish to create a high resolution approximation from only a single low resolution measurement.

3.5 Implementation Details

Here we describe the implementation details of our single image super resolution restoration algorithm. The first step is to denoise blurred noisy low resolution measurement \underline{y} with BM3D denoising algorithm. Following this similarity matrices K_S and K_L are computed from denoised measurement \underline{y} the simulated high resolution image \underline{y}_L and

respectively. In these experiments, we choose to bootstrap filtered low resolution measurement to desired dimension using bicubic interpolation thus simulating \underline{y}_L . With defined similarity matrices we proceed to compute our filtering matrices W_L and W_S which are a product of applying the Sinkhorn-Knopp matrix balancing algorithm to respective similarity matrices K_L and K_S . Calculating similarity matrices K_L and K_S is one of the most computationally demanding parts of our algorithm as every pixel and associated pixel neighbours require a Gaussian weighted euclidean norm of respective gray scale intensity vectors. As such since we are using the NLM weight scheme which has a special form due to construction using a neighbouring schematic we calculate similarity weights using an integral image [46] approach. Following the composition of the normalized graph Laplacian associated linear system of equations are given according to our proposed super resolution energy functional. The linear system of equations is then solved using conjugate gradient descent which retrieves high resolution solution \underline{x} .

Tuning parameters β and η are data dependent which implies to find optimal tuning values we have two options, the first a guess and check and second using some parameter finding algorithm. In general when parameter η is larger than β the amount of blur is greater than the amount of noise. In the reverse case when the amount of noise is much higher than amount of blur, parameter β is much larger than η . We compare performance of our super resolution method opposed to bootstrapping image using an interpolating polynomial. For this thesis all work is implemented in MATLAB and ran on a Lenovo Yoga Thinkpad with access to 8GB RAM, four Cores and an Intel i7 processor. The major bottle neck of the algorithm is the computation of simialrity matrix K and solving linear system utilizing conjugate gradient descent algorithm. On average for when decimation factor $L = 2$ these bottleneck steps (specifically extraction of matrix K) require 1 – 2GB of RAM, which is greatly reduced thanks to the special structure of the NLM algorithm and integral image approach. General time to solve our energy function was under 10 minutes, depending on image size.

Chapter 4

Numerical Experiments And Results

The performance of our algorithm is tested on a series of real and synthetic images, both RGB and gray scale. For the case of coloured images, each channel (RGB) is solved independently and once all channels have been restored are compiled to form restored final estimate \underline{x} . We plan to investigate the performance of our method at both extrema of degradation, the first more practical cases of minimal noise and blur and inversely large noise and image blur. We model noise as Additive White Gaussian noise (AWGN) with σ standard deviation.¹ In addition for these experiments blur operator **B** represents local averaging blur according to a factor z (more information can be found in Appendix A) and train down sampling operator **D** (found in Appendix B) according to factor L . We quantify the performance of our restoration algorithm with two measurements, the Peak Signal-to-Noise Ratio (PSNR)

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right)$$

with $\left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right)$ where MAX_I the maximum intensity value between 0 – 255 (or 0 – 1 if image is normalized) of an image vector, followed by normalization by the square root of

¹Additive refers to the noise being added to the original signal; white indicates a uniform frequency domain; Gaussian refers to the time domain having a normal distribution with standard deviation σ and zero mean.

the respective mean square error (MSE) defined in usual form

$$\text{MSE} = \frac{1}{NM} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - G(i, j))^2$$

with N and M the number of rows and columns respectively (dimension) of desired high resolution solution \underline{x} , $I(i, j)$ the intensity level at pixel (i, j) for \underline{x} and $G(i, j)$ the respective intensity level for the ground truth image at identical pixel pair (i, j) .

Our second measurement of performance is the Structural Similarity Index Measure (SSIM). This quality assessment measure not only takes into consideration the luminance (pixel intensity), but also contrast and structure when computing the SSIM measurement [47]. The three components are combined to yield the following similarity measure denoted as a function of the three components

$$\text{SSIM}(\underline{x}, \underline{y}) = f(l(\underline{x}, \underline{y}), c(\underline{x}, \underline{y}), s(\underline{x}, \underline{y})).$$

We also note that the three components are considered relatively independent of each other. The similarity function SSIM is expected to also satisfy the following conditions.

1. Symmetric : $\text{SSIM}(\underline{x}, \underline{y}) = \text{SSIM}(\underline{y}, \underline{x})$.
2. Boundless : $\text{SSIM}(\underline{x}, \underline{y}) \leq 1$.
3. Unique maximum: $\text{SSIM}(\underline{x}, \underline{y}) = 1$ if and only if $\underline{x} = \underline{y}$

The luminance comparison is defined as

$$l(\underline{x}, \underline{y}) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

where μ_x and μ_y the mean values of intensity level vectors \underline{x} and \underline{y} respectively, and c_1 is included to avoid instability when $\mu_x^2 + \mu_y^2$ is close to zero. The contrast comparison function resembles a similar form

$$c(\underline{x}, \underline{y}) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

where $\sigma_{\underline{x}}$ and $\sigma_{\underline{y}}$ the standard deviation of intensity level vectors \underline{x} and \underline{y} respectively, and c_1 is included to avoid instability when $\sigma_{\underline{x}}^2 + \sigma_{\underline{y}}^2$ is close to zero. Following the definitions of the luminance and contrast functions, the structural comparison can now be defined. The structural comparison is found after luminance subtraction and normalization by the standard deviation. The unit vectors for images \underline{x} and \underline{y} are defined as $(\underline{x} - \mu_{\underline{x}}) / \sigma_{\underline{x}}$ and $(\underline{y} - \mu_{\underline{y}}) / \sigma_{\underline{y}}$. The inner product (correlation) between these unit vectors is an effective and simple measure to quantify the structural similarity of the images we wish to compare. An important point to note is that the correlation between $(\underline{x} - \mu_{\underline{x}}) / \sigma_{\underline{x}}$ and $(\underline{y} - \mu_{\underline{y}}) / \sigma_{\underline{y}}$ is equivalent to the correlation between \underline{x} and \underline{y} [47]. Following this, the structural comparison function is defined as

$$s(\underline{x}, \underline{y}) = \frac{\sigma_{xy} + c_3}{\sigma_{\underline{x}}\sigma_{\underline{y}} + c_3}$$

where c_3 included to avoid instability. In the discrete form, the covariance σ_{xy} between images \underline{x} and \underline{y} is approximated with

$$\sigma_{xy} = \frac{1}{N^2 - 1} \sum_{i=1}^{N^2-1} (\underline{x}_i - \mu_{\underline{x}})(\underline{y}_i - \mu_{\underline{y}})$$

The three comparison function defined above as luminance, contrast and structure are then combined to yield the following similarity measure between images \underline{x} and \underline{y}

$$\text{SSIM}(\underline{x}, \underline{y}) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Of course since our problem requires a visually appealing solution \underline{x} we also investigate the effects of the algorithm qualitatively with visual inspection of given restored images.

4.1 Performance Of Restoration Algorithm For Various Levels Of AWGN

In our first set of numerical experiments, we analyze the effects of varying the amount of additive white Gaussian noise to our synthetic low resolution measurement $\underline{y} = \mathbf{H}\underline{x} + \underline{n}_{\sigma}$,

such that the measurement will be corrupted with either no noise ($\sigma = 0$), small amount of noise ($\sigma = 10$), or a very large amount of noise ($\sigma = 50$). The goal of this experiment is to explore the maximum amount of noise our algorithm can handle while still performing successfully. In the following set of experiments, unless otherwise stated all images are square and decimation factor k is set to a value of two. Also, for local averaging blur operator \mathbf{B} , the zooming factor is also set to two. Tuning parameters β and η are varied, ensuring that β is chosen such that filtering matrix W is doubly stochastic and $\eta > 0$ to ensure penalization of the energy functional. The search window size for pixels considered a neighbouring pixel of some target pixel is set to 11×11 . In addition the intensity gray level vector patch size is set to be 5×5 . The boundary conditions imposed surrounding the search window are symmetric. To test the effectiveness of our restoration algorithm, we test our method on a variety of low resolution images. We compute the PSNR and SSIM for each phase of the algorithm, as well as provide a visual analysis.

As a note in the following data tables the column labeled as NN corresponds to high resolution approximation via nearest neighbour interpolation, column BC via BiCubic interpolation, and of course initial quantity measures corresponding to measurement \underline{y} under column (LR). Results belonging to the SR column correspond to proposed algorithm for super-resolution. The following tables analyze quantitatively the results from increasing the resolution using the aforementioned methods for three different levels of image noise. For each of the images of the data set the PSNR and SSIM are given, the first Table 4.1 the case of zero additive noise ($\sigma = 0$). Each data cell contains two rows where the top row is PSNR in dB and bottom row SSIM. Observing the results it is clear that for the case of no additive noise ($\sigma = 0$), our super resolution algorithm outperforms interpolation methods by a substantial margin in terms of both SSIM and PSNR. We hope that this increase in PSNR and SSIM will correspond to an increase in visual clarity for each of the images. The second Table 4.2 contains results as a moderate ($\sigma = 10$) amount of additive white noise is added to the low resolution image. Finally, Table 4.3

results for high amount ($\sigma = 50$) of additive white noise. These results indicate that as we begin to introduce noise even though there is an increase in quality measure, the effectiveness of our method begins to diminish as the amount of image noise increases. As before each data cell contains two rows, where the top row is PSNR in dB and bottom row SSIM.

Image		LR	NN	BC	SR	Figure
Motorcycle	PSNR (dB) SSIM	22.51 0.77	22.39 0.77	23.0 0.79	26.21 0.88	4.1
Pineapple	PSNR (dB) SSIM	20.31 0.63	20.23 0.62	20.58 0.63	22.70 0.75	4.4
Pool	PSNR (dB) SSIM	24.31 0.87	24.28 0.87	24.72 0.88	28.20 0.93	4.7
Horses	PSNR (dB) SSIM	26.98 0.87	26.96 0.87	27.51 0.88	30.50 0.93	4.10
Boat	PSNR (dB) SSIM	24.79 0.84	24.74 0.84	25.20 0.85	28.04 0.91	4.13
Lime	PSNR (dB) SSIM	24.66 0.63	24.59 0.62	25.32 0.63	28.31 0.68	4.16

Table 4.1: Numerical measures of success regarding increasing the resolution of images for no AWGN ($\sigma = 0$).

As seen from Figure 4.1 the PSNR of low resolution measurement y is 22.51 dB, and one denoised with BM3D and bootstrapped by a factor of two for desired dimension using Nearest Neighbour (NN) and Bicubic Interpolation (BI), the measures of the solution using interpolation methods NN yields a PSNR of 22.39 dB and BC yielding PSNR of 23.0 dB respectively. Using our algorithm, the super resolved solution attains a PSNR value of 26.21 dB, a substantial increase compared to PSNR of low resolution measurement with respect to both interpolated results. Creating a high resolution approximation of a given low resolution measurement using interpolation methods does not seem to indicate

Image		LR	NN	BC	SR	Figure
Motorcycle	PSNR (dB)	21.47	21.85	22.31	24.06	4.2
	SSIM	0.68	0.72	0.74	0.80	
Pineapple	PSNR (dB)	19.67	19.85	20.14	21.52	4.5
	SSIM	0.55	0.57	0.58	0.68	
Pool	PSNR (dB)	22.85	22.86	24.19	26.51	4.8
	SSIM	0.74	0.84	0.85	0.90	
Horses	PSNR (dB)	24.53	26.34	26.75	28.51	4.11
	SSIM	0.67	0.83	0.84	0.87	
Boat	PSNR (dB)	23.15	24.14	24.48	26.04	4.14
	SSIM	0.69	0.80	0.81	0.85	
Lime	PSNR (dB)	23.20	24.18	24.78	26.74	4.17
	SSIM	0.47	0.58	0.59	0.63	

Table 4.2: Numerical measures of success regarding increasing the resolution of images which are corrupted using AWGN with standard deviation $\sigma = 10$.

a large increase in PSNR and SSIM, though this data set does not have any image noise thus the effects of BM3D are minimal.

We extract a small patch of the full images for Figure 4.1 of size 100×100 from the motorcycle data set to closely observe fine edge details for each methods solution compared to ground truth image. As seen from the low resolution measurement compared to BM3D denoised and interpolated versions, the edge information is very blurry specifically around detailed components of the image such as the wings above the text, or the text itself. Using our algorithm the text and wing components of the image become much sharper, and we can begin to notice separations of the feathers regarding the wings. We also note that using our algorithm the number 250 becomes much more visible below the CR label, while the yellow and green plastic has more evident shadows and depth.

In this experiment we add noise to motorcycle data set setting the AWGN to have a standard deviation of $\sigma = 10$. This is the more practical case for real life scenarios, as

Image		LR	NN	BC	SR	Figure
Motorcycle	PSNR (dB)	14.31	19.64	19.76	19.96	4.3
	SSIM	0.27	0.55	0.56	0.58	
Pineapple	PSNR (dB)	14.03	18.14	18.19	18.36	4.6
	SSIM	0.22	0.38	0.38	0.40	
Pool	PSNR (dB)	14.37	22.16	22.25	22.58	4.9
	SSIM	0.36	0.76	0.77	0.78	
Horses	PSNR (dB)	14.55	24.63	24.76	24.89	4.12
	SSIM	0.16	0.74	0.74	0.75	
Boat	PSNR (dB)	14.38	22.34	22.44	22.56	4.15
	SSIM	0.20	0.71	0.71	0.73	
Lime	PSNR (dB)	15.15	22.02	22.19	22.52	4.18
	SSIM	0.15	0.48	0.49	0.50	

Table 4.3: Numerical measures of success regarding increasing the resolution of images which are corrupted using AWGN with standard deviation $\sigma = 50$.

capturing images using sensor devices without introducing image noise is not legitimate. Observing Figure 4.2 we notice that as noise is introduced to measurement y , the effects of the BM3D algorithm for noise removal become apparent (large increase in PSNR) when bootstrapping to desired resolution using NN and BC interpolation. Comparing the solution from our super resolution algorithm with respective interpolation methods there is an evident increase in overall PSNR. In addition we note that there is an evident increase in PSNR for NN interpolation (PSNR of 21.85 dB) and BC interpolation (PSNR of 22.31 dB) solutions compared to LR measurement, as now the effects of the BM3D denoising method become apparent. For the case of no noise $\sigma = 0$ the proposed SR algorithm yields a PSNR of 26.21 dB compared to moderate amount of AWGN with standard deviation $\sigma = 10$ which yields a PSNR of 24.06 dB which is evidently lower. Thus as illustrated from the numerical results in Table 4.2, it's evident as we begin to introduce noise into the low resolution measurement the success of our methods become

affected.

We also extract a small patch of the images in Figure 4.2 for the case of AWGN $\sigma = 10$. Comparing the low resolution measurement (PSNR of 21.47 dB) to the results from denoising followed by interpolating (NN and BC), the SR methods yields an increase in edge detail while suppressing pixelation of the image. We now compare the two results from interpolation our algorithm, starting with the overall increase of edge information and sharpness. When noise is added it seems fine details such as the separation of feathers on the wing become more difficult to extract, regardless of the method utilized. Observing the text and numerical components of the image, we notice that for larger text the sharpness substantially increases. Lastly, the yellow and green plastic components of the image especially the shadowing components become more sharp.

We now corrupt the Motorcycle data set with a high amount of AWGN ($\sigma = 50$). As illustrated from the results of Figure 4.3 the largest overall jump in PSNR and visual clarity is the application of the BM3D algorithm to remove image noise from low resolution measurement \underline{y} . After using interpolation methods to increase the resolution, it is evident that neither NN nor BC performs successfully as the image is still extremely blurry. In addition the result from our SR algorithm even though there is an increase in clarity it is marginally better than interpolation, implying that very noisy images may not be a good candidate for our proposed method.

Extracting a small patch in Figure 4.3 for each of the images, it is clear that all three results barely reproduce a visually appealing solution. For all methods all components of the image are blurry, and increasingly difficult to distinguish between each other. The text and numerical information of the image have very little to no structure, and the logo is barely recognizable compared to the cases of moderate and no noise solutions. In addition the smoother regions of the image such as the green and yellow plastic are very blurry, and the fine edge information regarding the shadows become less evident. It seems obvious then that in the case where images are severely corrupted with noise, our

super resolution algorithm will not perform well.

We perform a similar visual analysis regarding the pineapple data set for three levels of AWGN, identical to the corruption for the motorcycle data set. Firstly for the case of no additive Gaussian noise $\sigma = 0$ there is a minimal increase in PSNR Figure 4.4 when comparing the denoised interpolated solutions and the low resolution measurement. Again this is expected since the effects of the BM3D algorithm are negligible and the only increase in PSNR is contributed from the interpolation methods. Though when creating the high resolution image using proposed super resolution algorithm there is a substantial increase in PSNR 22.70 dB compared to the low resolution measurement with PSNR of 20.31 dB.

Visually, the effects of the algorithm can be seen more clearly on the bottom row of Figure 4.4. As expected from the performance measures and visual observation of the full image when interpolating the denoised image using respective techniques the result is similar to the input, the BC interpolation has superior results than the NN replication. Comparing the result using our method we note that the structure of the pineapple is much more sharp, as well as the edges for the plants becoming more distinct compared to the low resolution measurement \underline{y} . In addition the ridge information along the pineapple becomes more sharp, though the coloring information seems to be mostly lost.

We now introduce additive white Gaussian noise with standard deviation $\sigma = 10$ to the pineapple data set illustrated in Figure 4.5 and perform an analysis similar to the previous segment. We note that the effects of the BM3D denoising operator become apparent, as after denoising data and bootstrapping using interpolation there is a moderate increase in performance measure attaining a PSNR of 19.85 dB and PSNR of 20.14 dB for NN and BC respectively compared to the low resolution measurement with PSNR of 19.67 dB. The greatest increase in visual clarity and performance measures is after applying our super resolution algorithm, which results in a PSNR of 21.52 dB. Though as expected and similar to the case of the motorcycle data set as noise is introduced into

the image, the effectiveness of our super resolution algorithm begins to abate. We also notice the general increase of sharpness and edge detail across the entire image.

Again we extract a small region of the pineapple images for $\sigma = 10$ for various stages of the inverse problem. Beginning with the low resolution input there is a moderate increase in resolution after denoising using BM3D and bootstrapping using NN and BC interpolation. Observing Figure 4.5 one notices that comparing the LR measurement and the interpolated solutions, the pineapple as well as surrounding foliage is still very blurry and is missing substantial edge information that separates object in the image. The super resolved image using our restoration algorithm outputs a visually appealing image, though the sharpness of the pineapple and foliage is not the same quality compared to the case of no noise ($\sigma = 0$). This evidence follows suit with our earlier result for the motorcycle experiment that at moderate amounts of noise the algorithm still performs well, though is not as effective compared to the no noise examples.

The last case we analyze is Figure 4.6 which corresponds to the pineapple data corrupted with large amount of additive white Gaussian noise ($\sigma = 50$). As expected the largest increase in PSNR and visual clarity is after applying the BM3D denoising algorithm to the severely noisy measurement (PSNR of 14.03 dB), as bootstrapping the image using interpolation methods NN yielding PSNR of 18.14 dB and BC yielding PSNR of 18.19 dB. We also observe the result from applying proposed super resolution algorithm the PSNR is 18.36 dB which is an increase in measure compared to interpolation, though quite minimal. Also observing the entire image domain of the super resolved image, it is evident that there is almost no obvious increase in visual clarity.

We extract small regions for each of the images in Figure 4.6 from the severely noisy pineapple experiment. We notice that in the LR measurement the image is severely noisy and important edge information is difficult to see. After applying the denoising algorithm and interpolating it is evident that the images are severely blurry and lack the sharpness required for a quality image. Comparing these results to our super resolved solution

we observe that the results visually are no better than interpolation, as the image lacks sharpness and is overall blurry. Specifically we notice that the pineapple information is otherwise lost due to over smoothing and the leaves of the plants are portrayed as one large lump. This result is consistent with our earliest experiment regarding the severely noisy motorcycle data, in which severely noisy images are difficult to super resolve if not impossible using this restoration method.

For the rest of the images, the reader can perform a similar analysis for the performance metrics as well as visually inspecting the quality of the restoration algorithm.

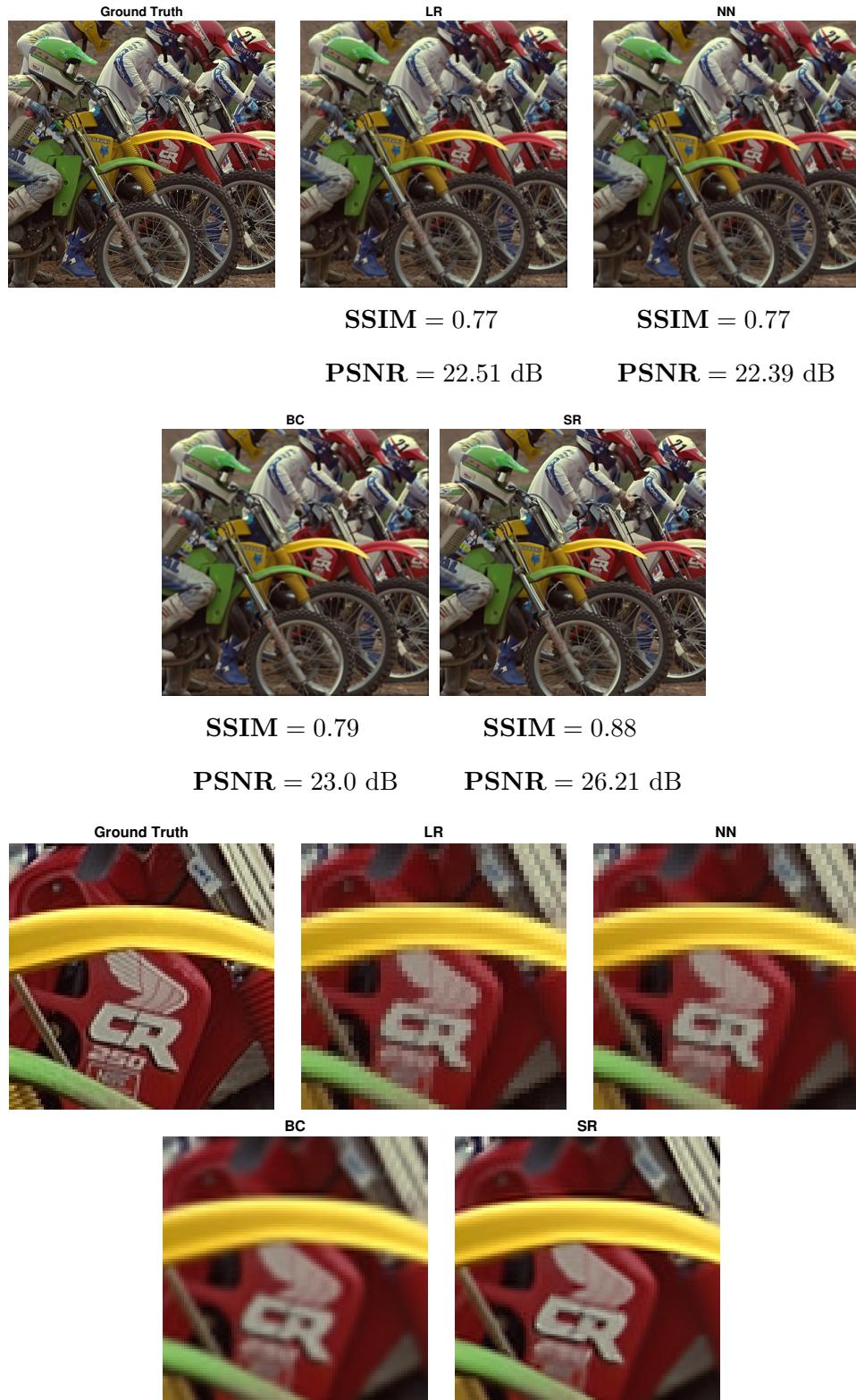
Figure 4.1: Motorcycle data with $\sigma = 0$ AWGN.

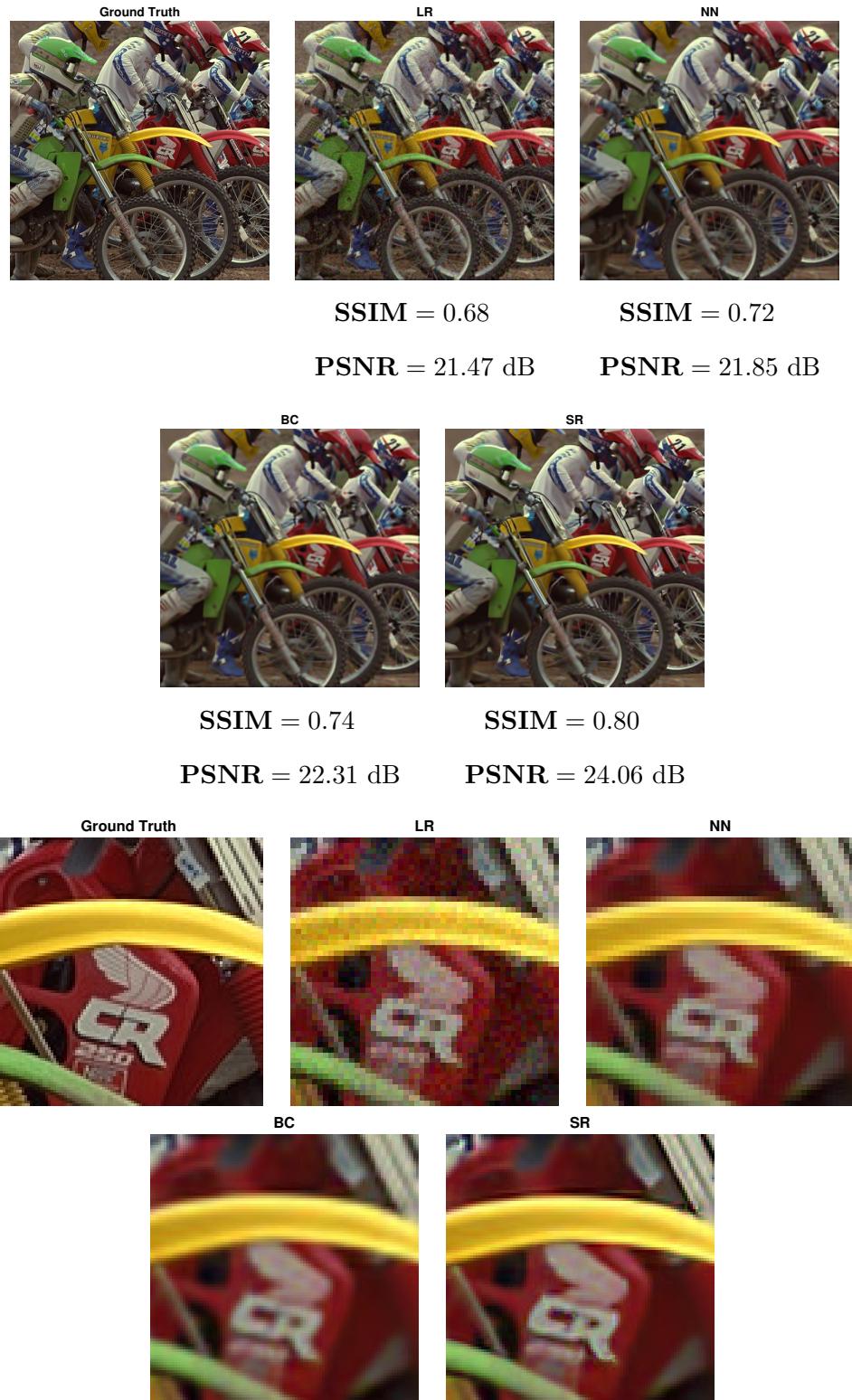
Figure 4.2: Motorcycle data with $\sigma = 10$ AWGN.

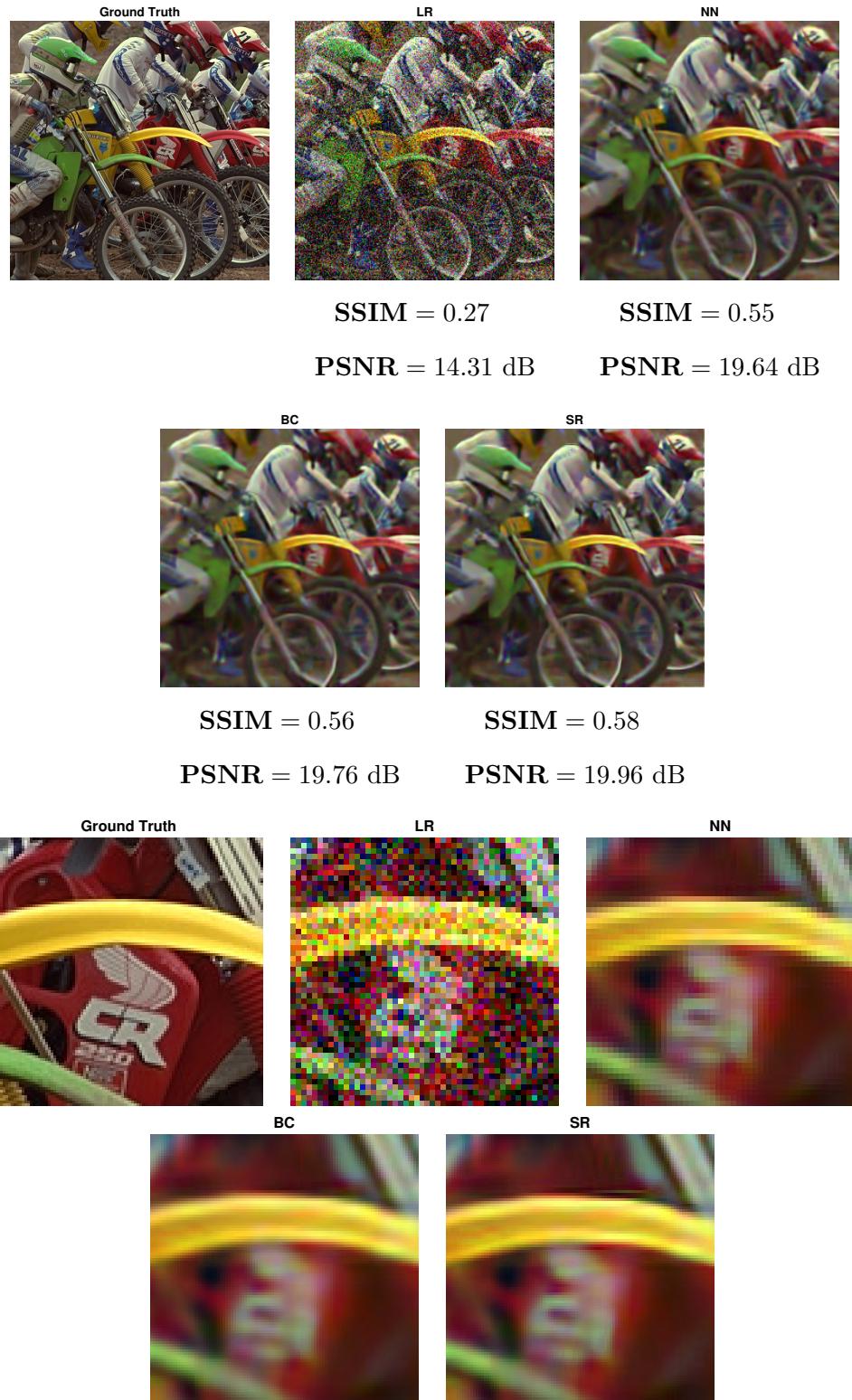
Figure 4.3: Motorcycle data with $\sigma = 50$ AWGN.

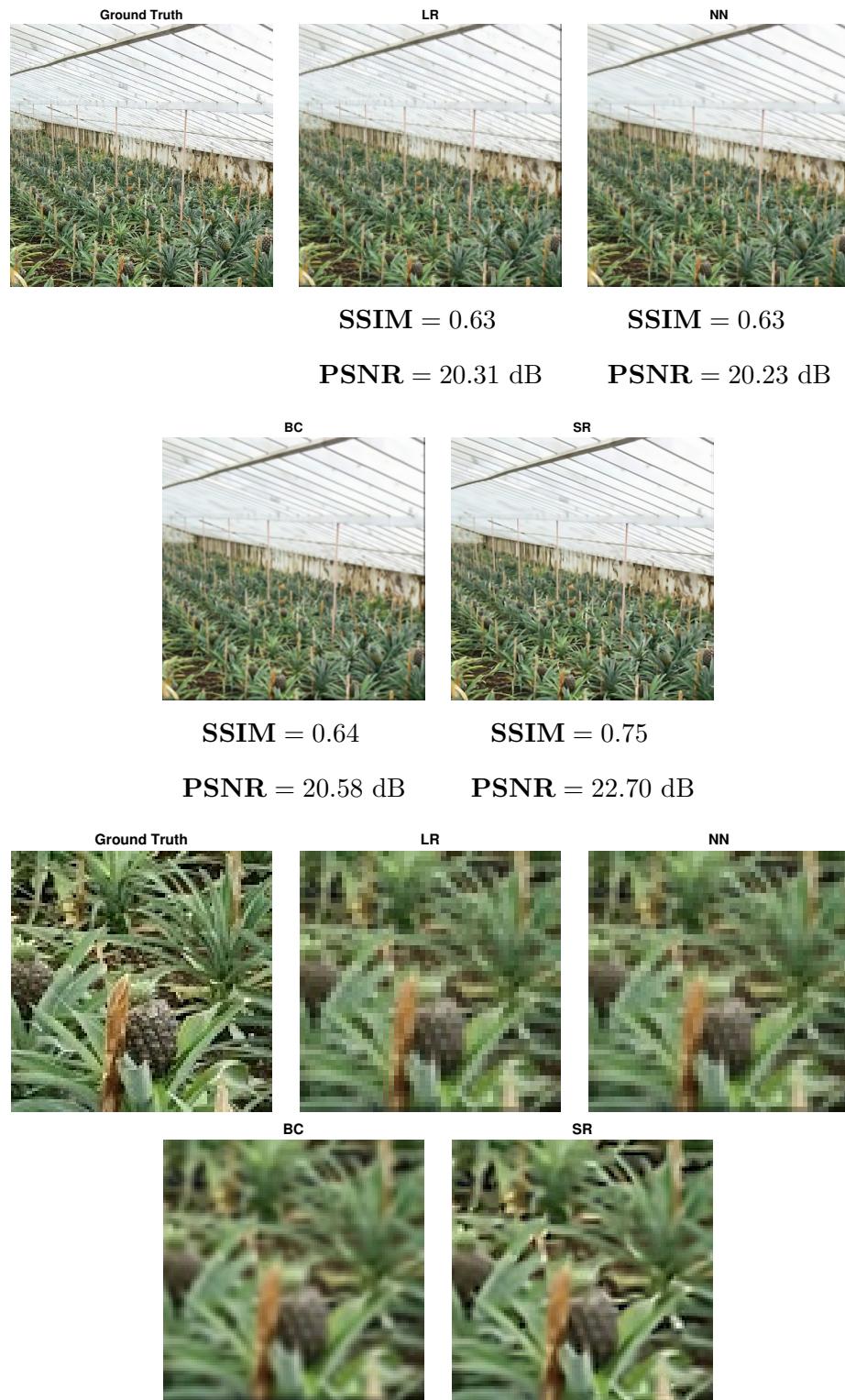
Figure 4.4: Pineapple field data with $\sigma = 0$ AWGN.

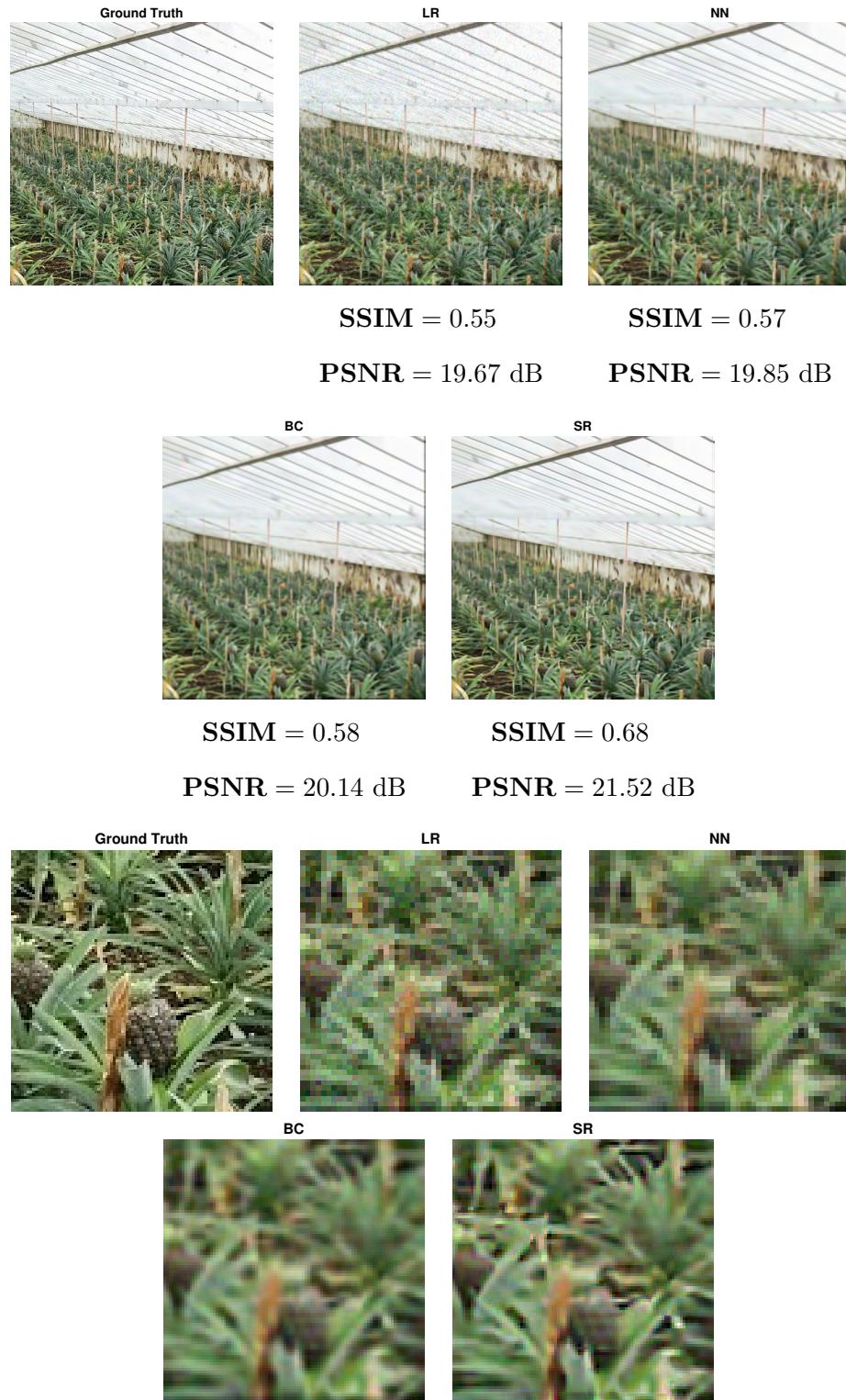
Figure 4.5: Pineapple field data with $\sigma = 10$ AWGN.

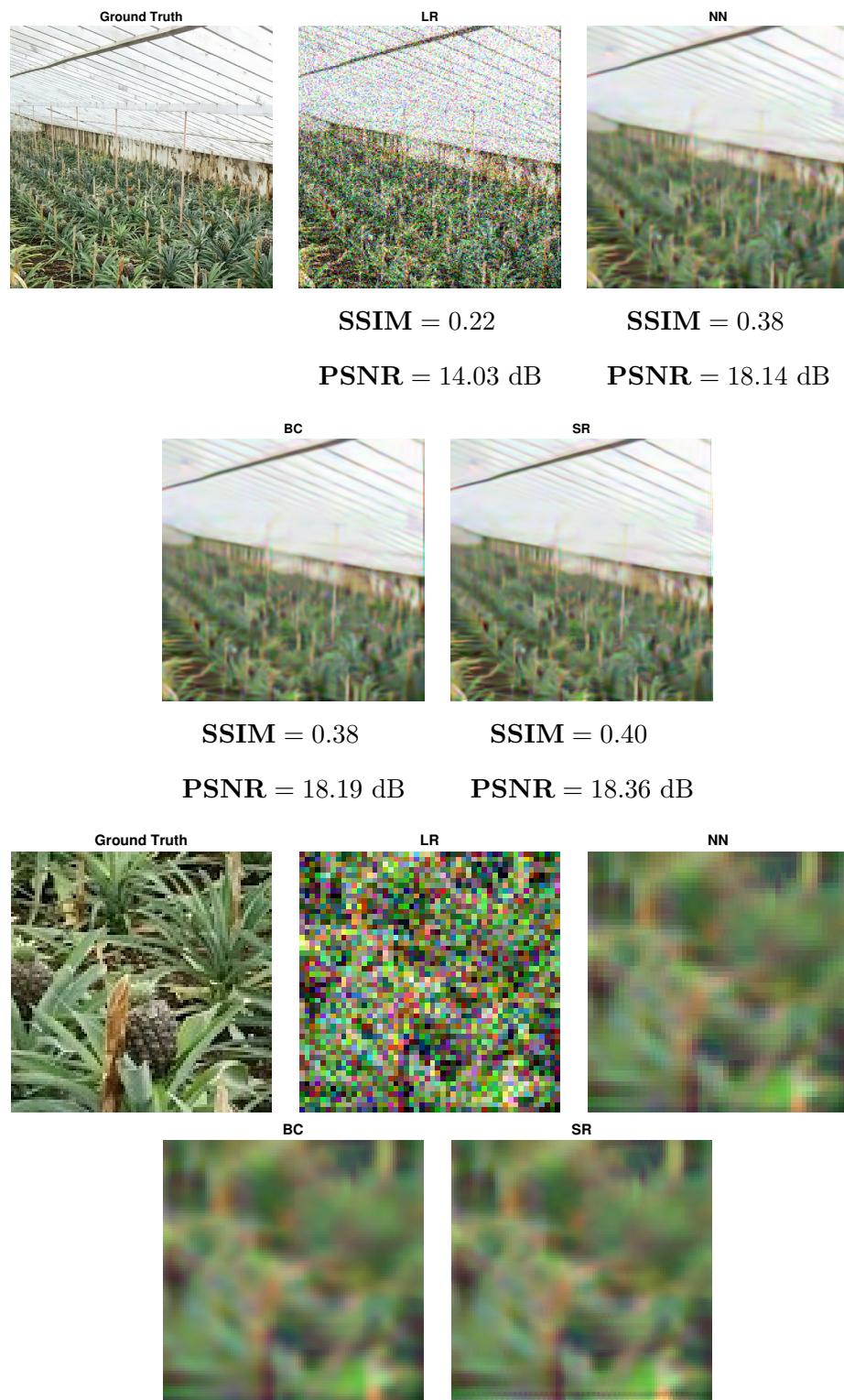
Figure 4.6: Pineapple field data with $\sigma = 50$ AWGN.

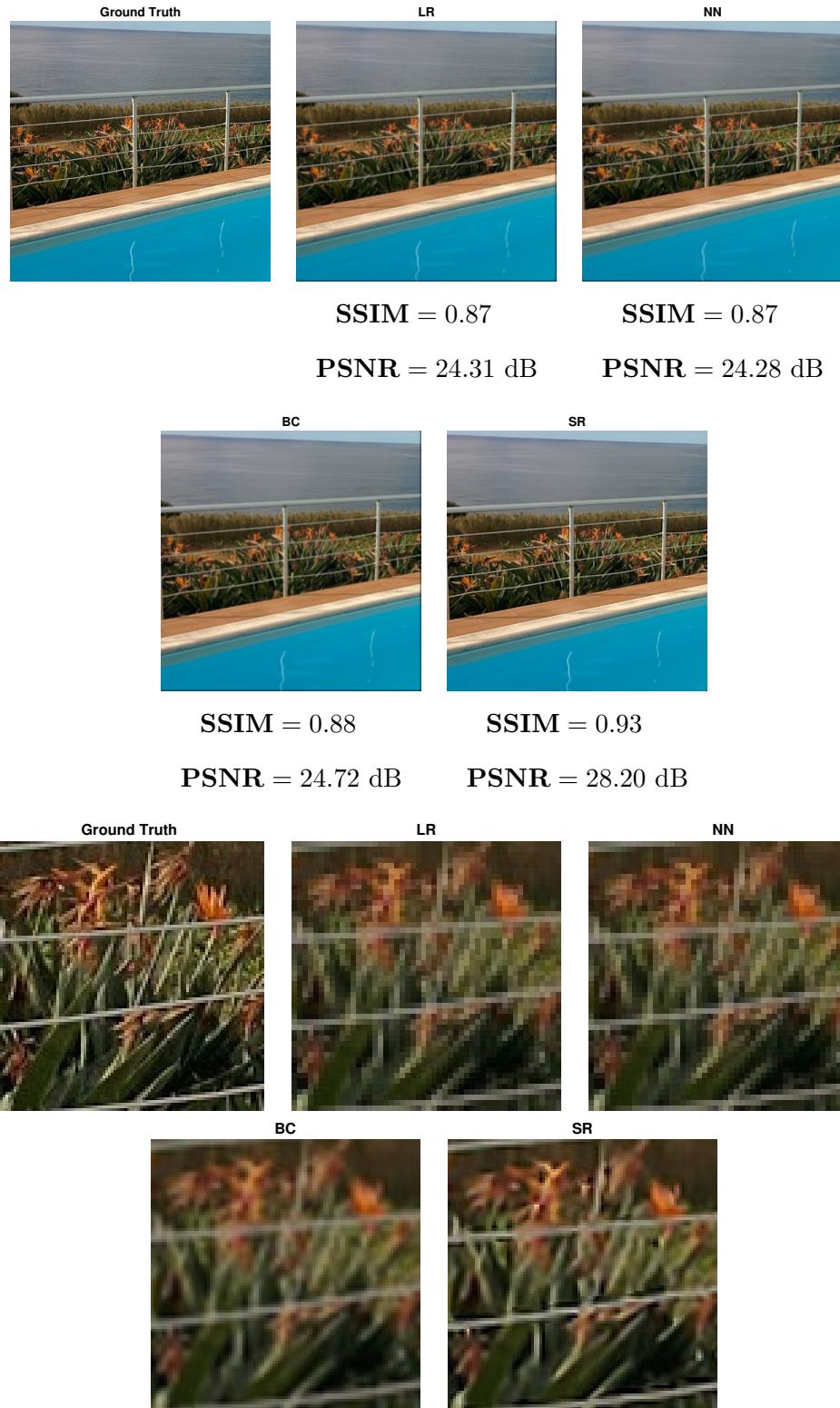
Figure 4.7: Pool data with $\sigma = 0$ AWGN.

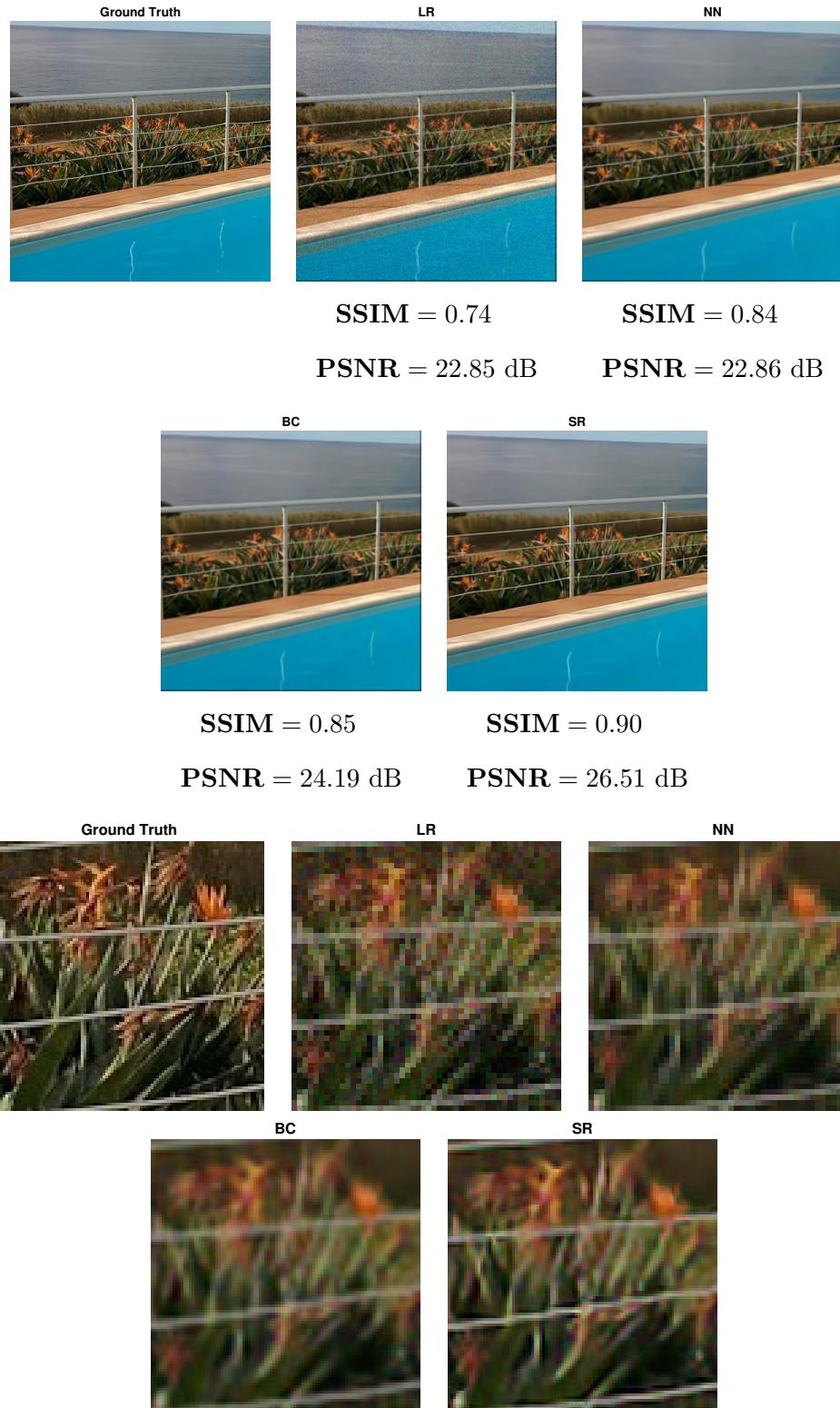
Figure 4.8: Pool data with $\sigma = 10$ AWGN.

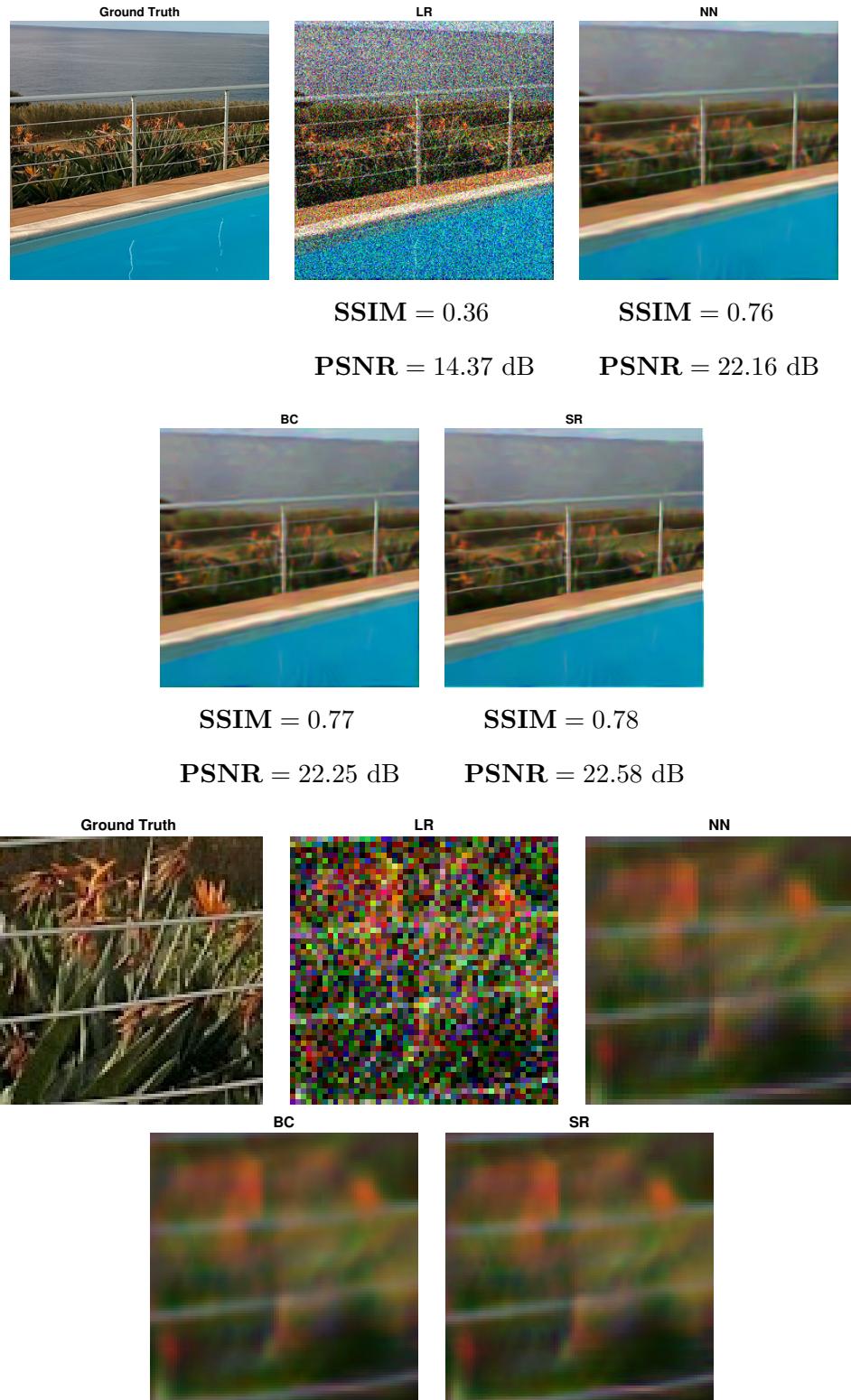
Figure 4.9: Pool data with $\sigma = 50$ AWGN.

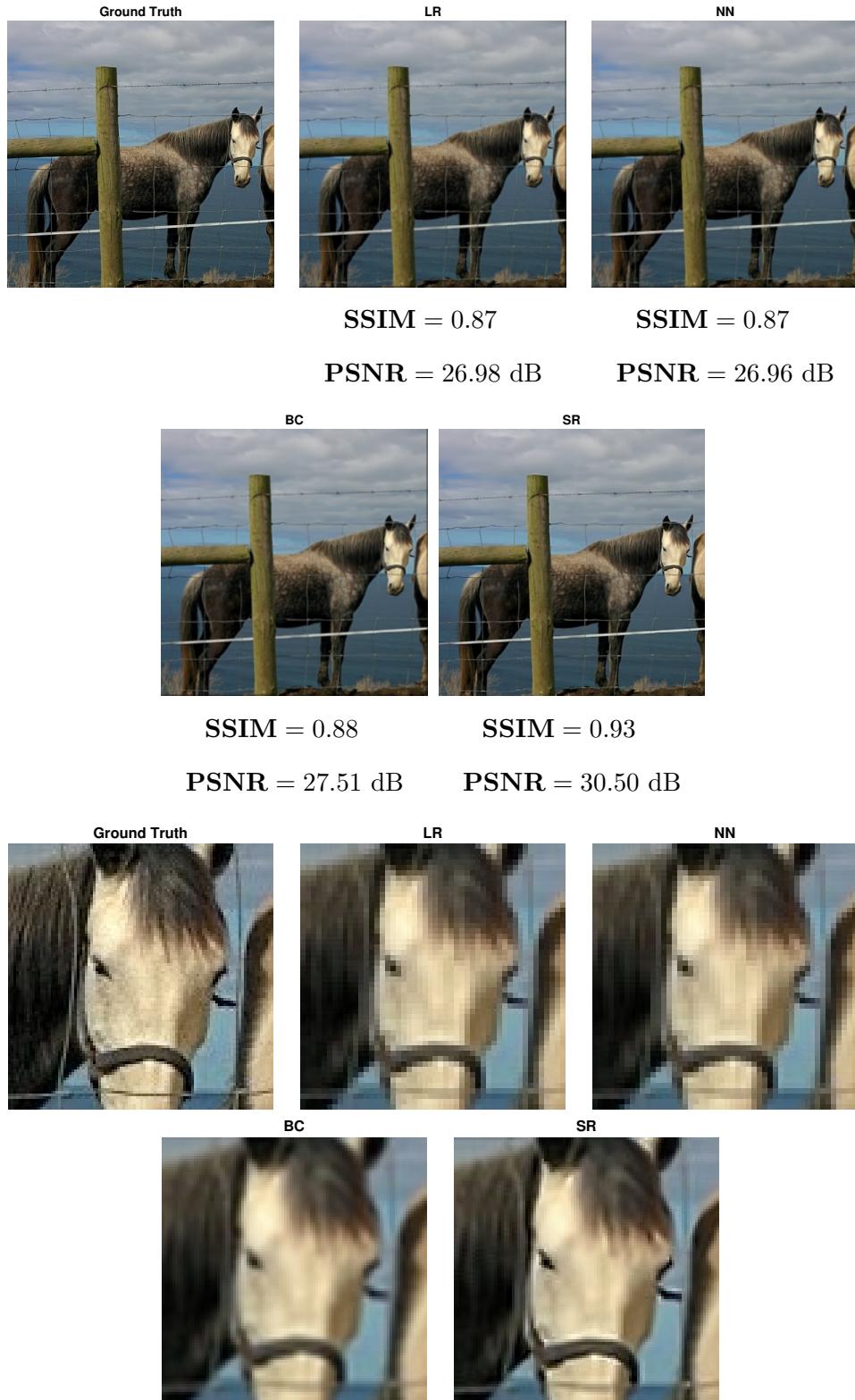
Figure 4.10: Horses data with $\sigma = 0$ AWGN.

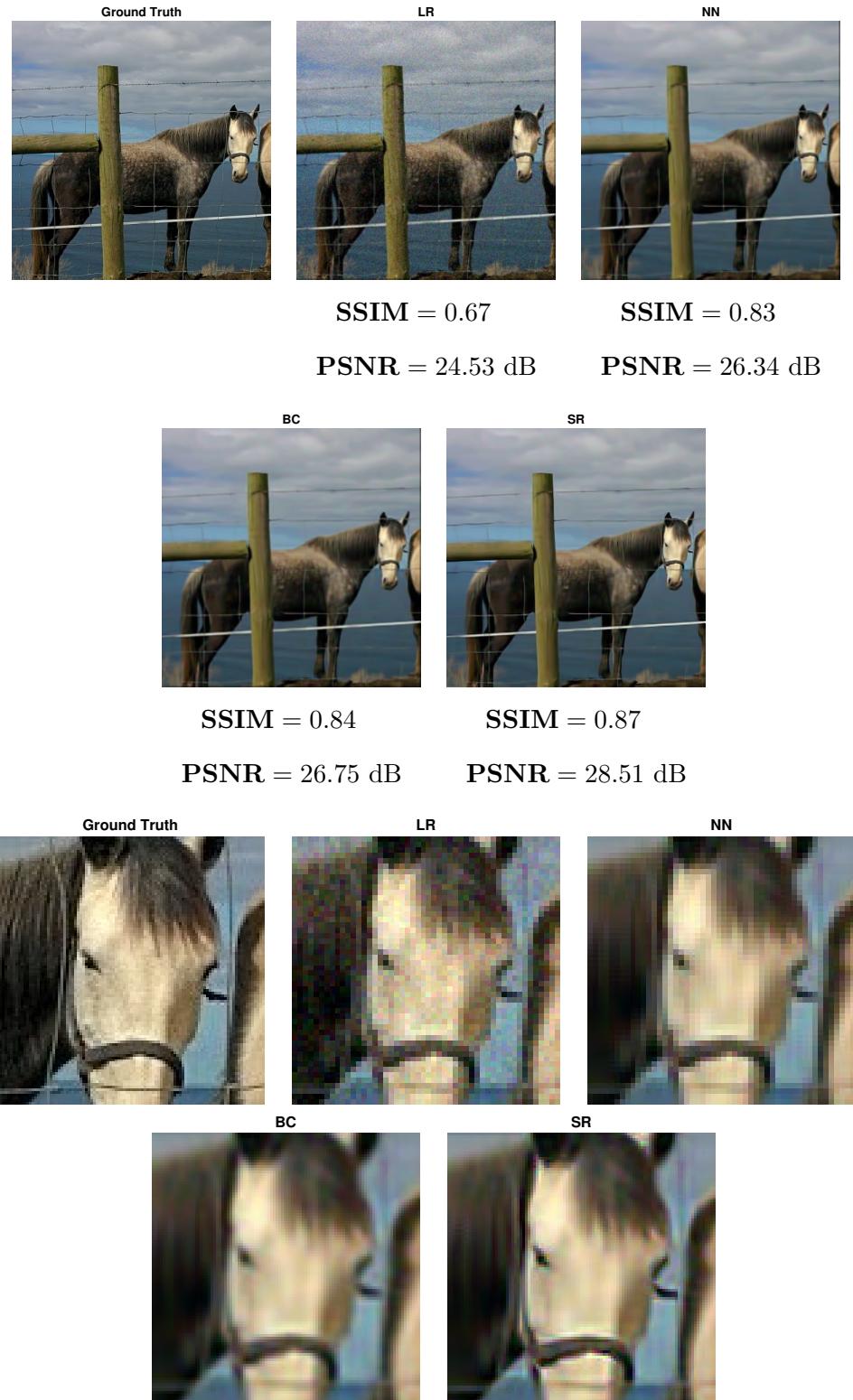
Figure 4.11: Horses data with $\sigma = 10$ AWGN.

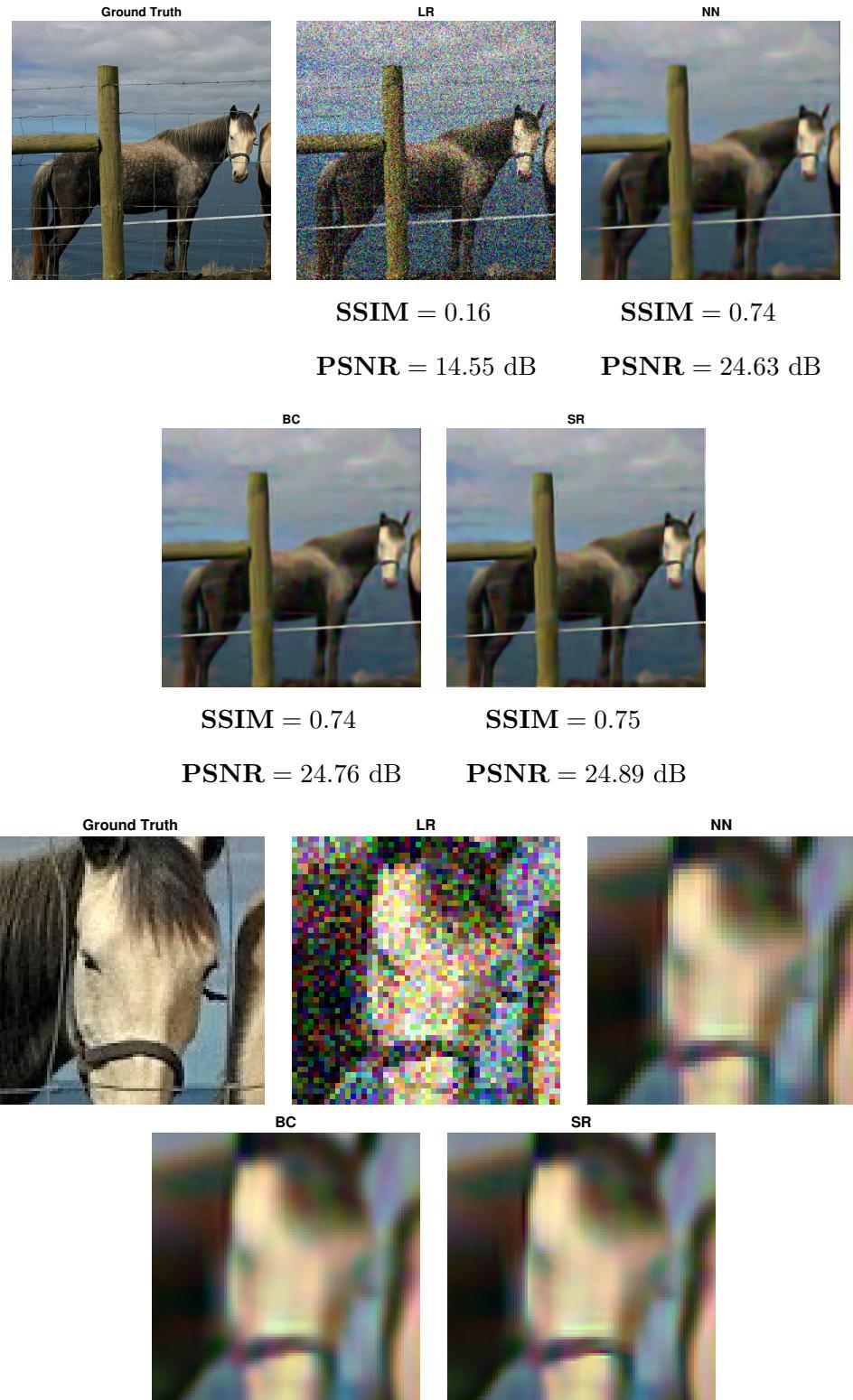
Figure 4.12: Horses data with $\sigma = 50$ AWGN.

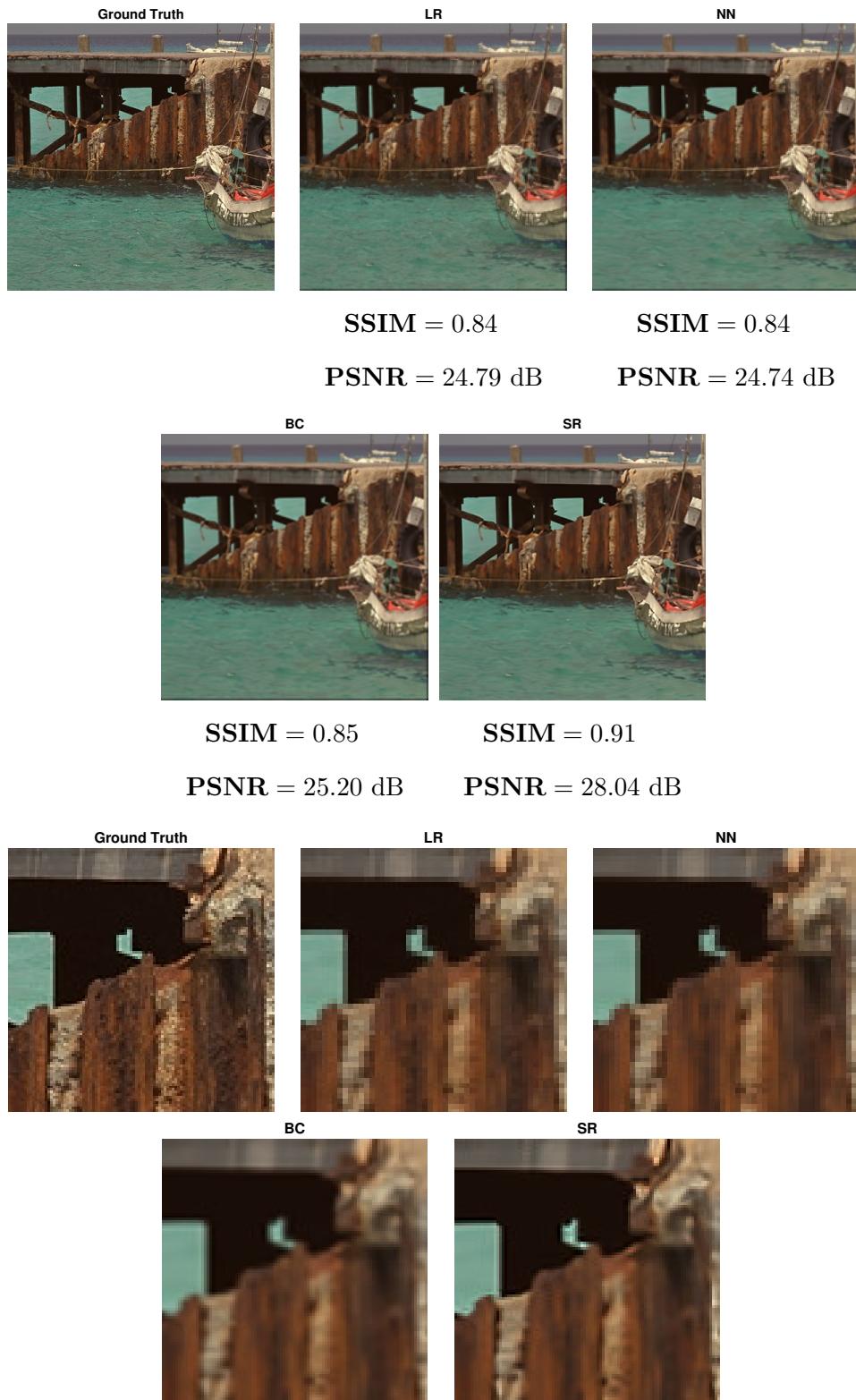
Figure 4.13: Boats data with $\sigma = 0$ AWGN.

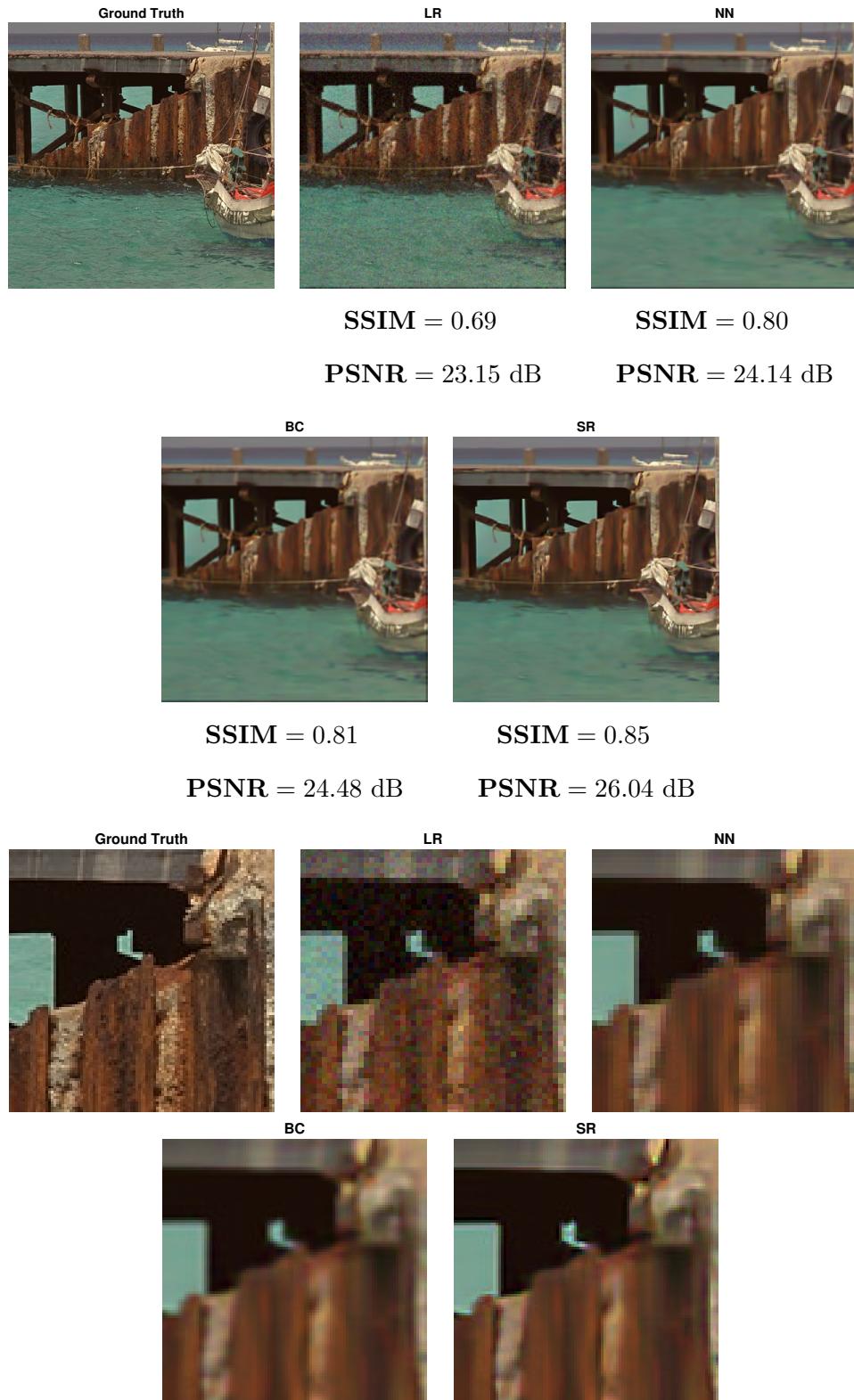
Figure 4.14: Boats data with $\sigma = 10$ AWGN.

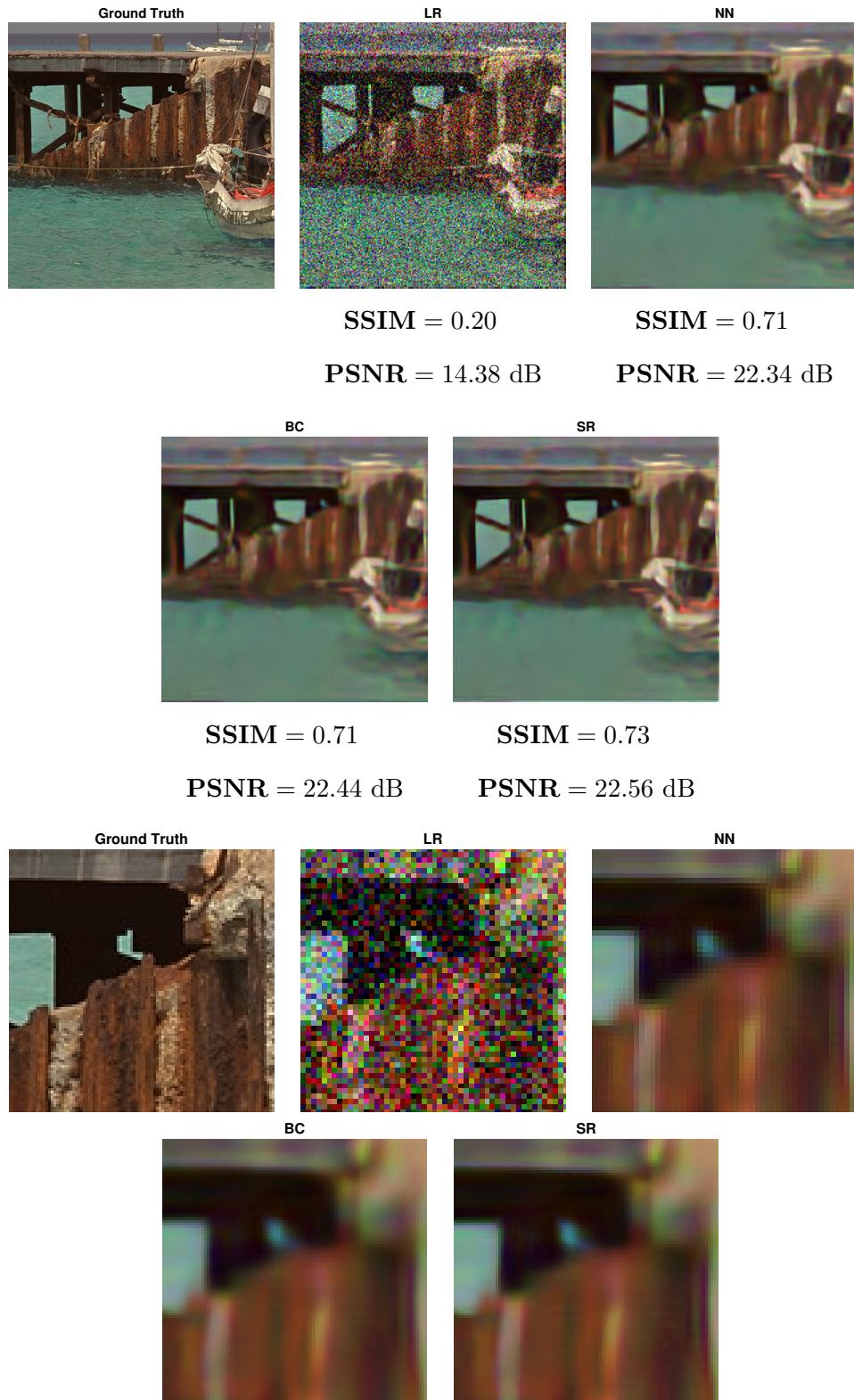
Figure 4.15: Boats data with $\sigma = 50$ AWGN.

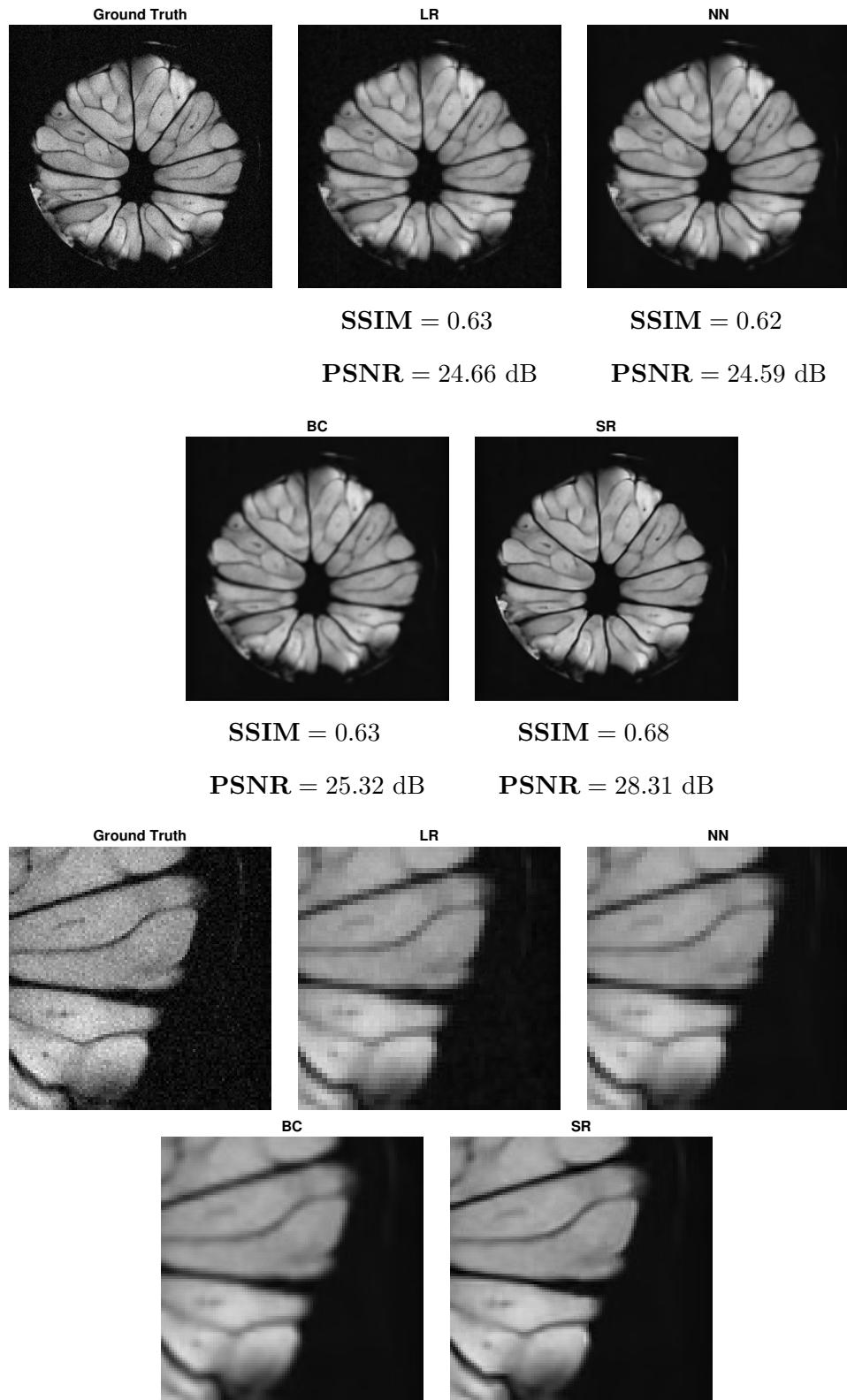
Figure 4.16: Lime data with $\sigma = 0$ AWGN.

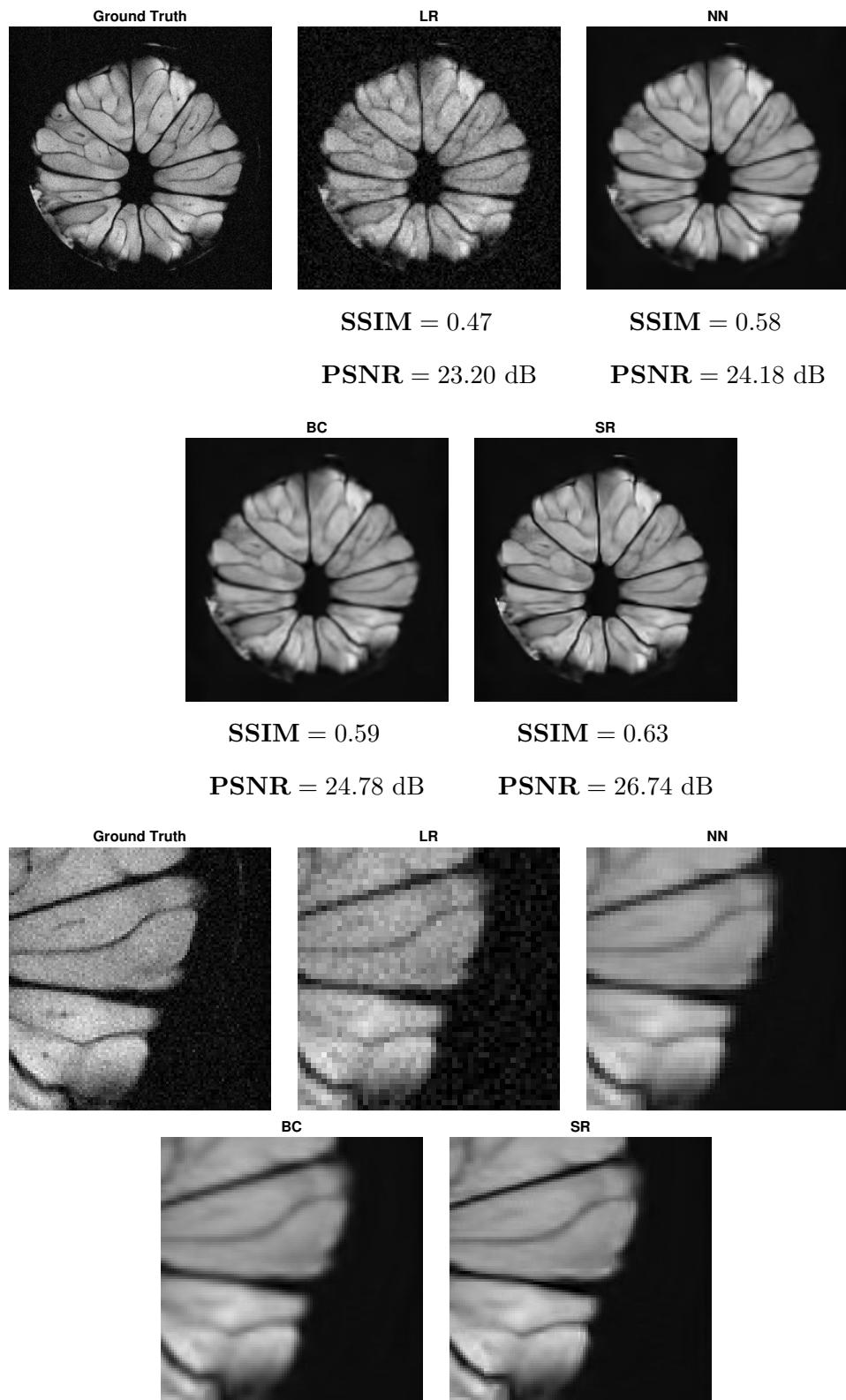
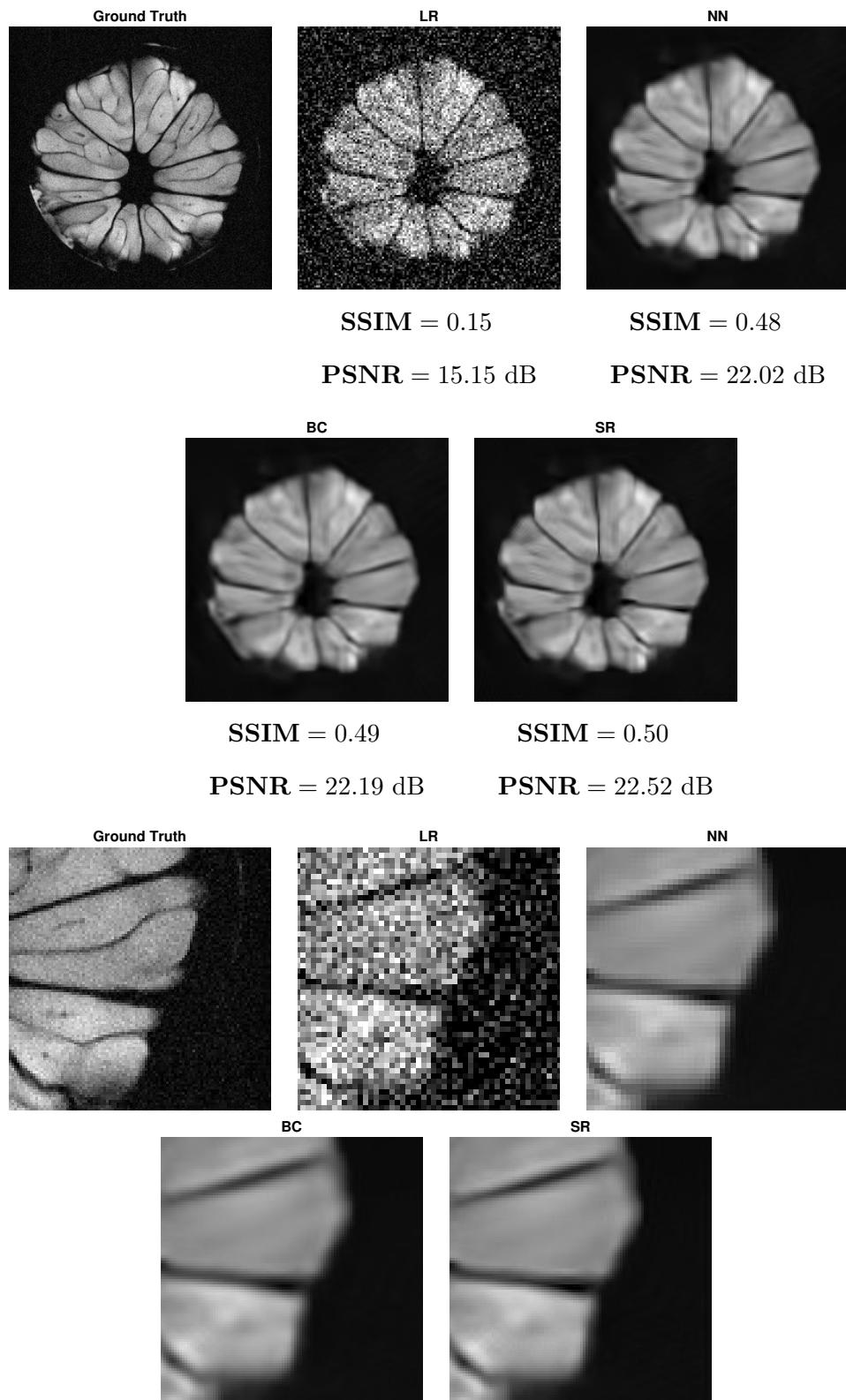
Figure 4.17: Lime data with $\sigma = 10$ AWGN

Figure 4.18: Lime data with $\sigma = 50$ AWGN

4.2 Tuning Parameters

Here we analyze the resulting PSNR of using our proposed algorithm on a noisy blurred Horses image with AWGN $\sigma = 10$ while varying tuning parameters η and β . As expected from the results, tuning parameters are data dependent and optimal values must be found either through trial and error, or some parameter finding algorithm.

(η, β)	0.001	0.1	0.2	0.5	1	2	5	10	15
0.001	27.42	27.42	27.42	27.42	27.42	27.43	28.06	28.07	28.10
0.03	27.21	27.23	27.25	27.30	27.33	27.37	27.40	27.41	27.41
0.06	26.94	26.99	27.03	27.12	27.21	27.30	27.37	27.40	27.40
0.08	26.76	26.83	26.88	27.00	27.12	27.24	27.34	27.38	27.40
0.1	26.59	26.67	26.73	26.88	27.03	27.18	27.32	27.37	27.39
0.5	24.67	24.79	24.91	25.20	25.57	26.05	26.70	27.05	27.19
1	23.68	23.79	23.90	24.17	24.55	25.11	26.00	26.61	26.88
2	22.71	22.79	22.87	23.11	23.44	23.98	24.99	25.84	26.27
5	21.48	21.53	21.59	21.73	21.97	22.38	23.31	24.29	24.91
10	20.60	20.63	20.67	20.76	20.92	21.21	21.94	22.84	23.50

Table 4.4: Resulting PSNR (dB) of super resolved Horse image with $\sigma = 10$ AWGN varying tuning parameters β and η . These results are for when β is positive.

The only restriction is on tuning parameter β which ensures that linear system of equations when solving objective function will be positive definite, to ensure the doubly stochastic property of filtering matrix W . In general since implementation of this algorithm requires tuning which is inherently data dependent, we suggest that any user implement a parameter finding algorithm. This will ensure that the resulting image will be optimal when using our super resolution algorithm.

4.3 SR Restoration Analysis Regarding Decimation Factor L

. Here we analyze the restoration results for various levels of decimation factor L . The role of decimation factor L is to control the ratio we wish to up sample our LR measurement by, and naturally as L increases we are forced to approximate an increasing number of pixel intensity information. Observing Table 4.5, one will note that as decimation factor L increases, the PSNR of the restored solution will begin to decrease. As expected, since we are forced to synthesize pixel information for a larger HR plane the quality of the solution will begin to degrade.

(L, σ)		0	10	50
2	PSNR (dB)	27.34	25.43	21.86
	SSIM	0.88	0.79	0.65
4	PSNR (dB)	23.10	21.78	19.35
	SSIM	0.73	0.66	0.57
8	PSNR (dB)	20.21	19.38	17.41
	SSIM	0.61	0.56	0.51

Table 4.5: Resulting PSNR (dB) of super resolved Lime image while varying decimation factor L for various amounts of AWGN

4.4 Analysis Regarding Smoothing Parameter h For Non-Local Weight Operator

We recall that composition of similarity matrices K_S and K_L are of the form $K = \exp(-\|\mathbf{G}\|_2^2/h^2)$ which require an assignment of parameter h (used to control the smoothing of image noise) during the patch based weight assignment scheme from the NLM algorithm. We note that $\|\mathbf{G}\|_2^2$ denotes the Gaussian weighted ℓ^2 norm for any patch \mathbf{P}

and standard deviation of σ . As such, the following experiment will illustrate the relationship between the amount of additive white Gaussian noise and smoothing parameter h for a blurred boat image with various amount of additive noise. In this experiment in addition to degradation operator \mathbf{H} , we apply various levels of additive white Gaussian noise, ranging from $\sigma \in \{0, 10, 20, 30, 40, 50\}$ to the motorcycle image. We set smoothing parameter to be $h = 10\sigma$.

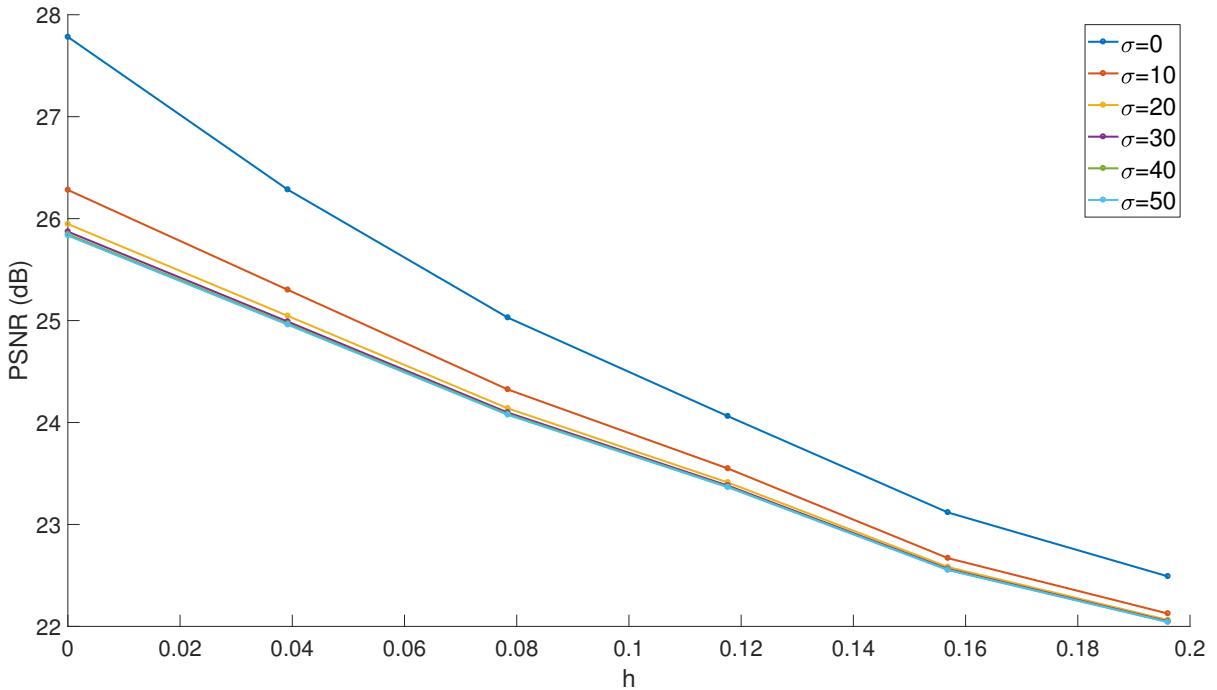


Figure 4.19: Lime image for various levels of AWGN utilizing different smoothing parameters $h = 10\sigma$.

The Figure 4.19 indicates that parameter h has an influence on the quality of the results, since the computed PSNR of restored images decreases as the smoothing parameter increases. It is also interesting to note that for the case where there is no noise ($\sigma = 0$), the PSNR is the largest, further promoting our results from earlier. Thus, smoothing parameter h seems to have an influence on the quality of results when constructing the similarity matrices K_L and K_S , implying parameter h must be chosen wisely to avoid over smoothing.

4.5 Analysis Regarding Degree of Vertices for Image Graph With Respect To Non-Local Weight Operator

We recall that the degree for each vertex $u \in V$ is assigned according to the search window radius of the NLM restoration algorithm, visualized in the neighbour assignment schematic from Figure 3.4. As the search window radius grows larger, the number of neighbours will increase inherently increasing the number of edges for each vertex (or increasing the degree of each vertex). As the number of edges for each vertex $u \in V$ increases the size of sparse similarity matrix K will also become more dense, implying that now each restored vertex will incorporate information from neighbouring pixels of a larger domain. In this experiment, we wish to investigate the relationship between the size of the search window radius and amount of additive noise corrupting low measurement \underline{y} . We vary the amount of additive white Gaussian noise to be of standard deviation $\sigma \in \{0, 10, 50\}$ to the lime image data set. The corresponding PSNR and SSIM are shown in Table 4.6 for various search neighbourhood radii.

The results from this experiment indicate that increasing the degree of each vertex for the image graph will not have an effect on the resulting PSNR, implying that the effectiveness of similarity matrix K capturing accurate weights is described well for small search window radii. This is a beneficial result from the computational demand perspective of the problem. We conclude that increasing the search window radius has invariant results on the overall quality of the image.

(Search radius, σ)		0	10	50
7	PSNR (dB)	27.74	26.22	22.45
	SSIM	0.68	0.62	0.50
9	PSNR (dB)	27.72	26.24	22.45
	SSIM	0.68	0.62	0.49
11	PSNR (dB)	27.71	26.23	22.47
	SSIM	0.68	0.62	0.49
13	PSNR (dB)	27.73	26.27	22.46
	SSIM	0.68	0.62	0.49
15	PSNR (dB)	27.72	26.24	22.47
	SSIM	0.68	0.62	0.49

Table 4.6: Quantitative measures for lime image regarding different search neighbourhood radii for various levels of AWGN.

Chapter 5

Conclusion And Future Works

In this work introduced the concept of creating an iterative boosting algorithm based on the SOS post-processing method which utilizes a non-local restoration approach. We show that in general treating the sequential boosting operator as it's own denoising algorithm is not efficient, and can actually result in a decrease in visual quality and performance measure. In addition we analyzed the iterative SOS boosting algorithm for image denoising, and proved that sequential application of boosting operator is equivalent to a re-parametrization of the original energy function. Through numerical experiments we illustrate that an iterative application of the boosting operator results in an overall decrease in visual clarity and PSNR, after initial application of boosting operation.

Following the theme of non-local restoration, we present the derivation to compute a non-local regularization functional for our SISR problem which is constructed according to the normalized graph Laplacian operator proposed by [23]. The normalized graph Laplacian functional is paired with a residual enhanced likelihood estimator as a data fidelity term, leading to our proposed objective functional to solve our SISR inverse problem. The resulting linear system of equations is then solved using an iterative method, such as Conjugate Gradient Descent. We test the proposed SR method on a series of images with varying level of image noise and find that the SR method performs well for

images with no noise ($\sigma = 0$) and moderate amount of noise ($\sigma = 10$), though as the image becomes severely noisy ($\sigma = 50$) the SR method is not as effective. Following this, we analyze the resulting PSNR of high resolution approximations when varying parameters regarding composition of similarity matrices. Results from these experiments imply that tuning parameters regarding similarity matrices has an invariant effect on resulting PSNR of high resolution approximation.

One question we raise is the applicability of applying the SOS boosting procedure to our proposed single image super resolution algorithm. Given the non-local composition of the regularization functional, the normalized graph Laplacian is a prime candidate for a post enhancement procedure such as the SOS boosting method. Furthermore, according to [22] the SOS procedure reduces the local-global gap between neighboring patches when the chosen regularizer is the normalized graph Laplacian, which would be a beneficial property for our enhancement procedure.

Appendices

Appendix A

Local Averaging Blur Operator

Corresponding local averaging blur filter is defined [40] As such we can state that the local averaging bluring matrix $\mathbf{B} \in \mathbb{R}^{(N-z+1)^d \times (N)^d}$ with zooming factor $z \in \mathbb{N}$ to be

$$\mathbf{B} := \underbrace{\mathbf{B}_1 \otimes \cdots \otimes \mathbf{B}_1}_{d \text{ times}}$$

where $\mathbf{B}_1 \in \mathbb{R}^{(N-z+1)^d \times (N)^d}$ is

$$\mathbf{B}_1 = \frac{1}{z} \begin{bmatrix} \underbrace{1 \cdots 1}_{z \text{ times}} & 0 & \underbrace{0 \cdots \cdots 0}_{N-z-1 \text{ times}} \\ 0 & \underbrace{1 \cdots \cdots 1}_{z \text{ times}} & \underbrace{0 \cdots \cdots 0}_{N-z-2 \text{ times}} \\ \vdots & \vdots & \vdots \\ \underbrace{0 \cdots \cdots 0}_{N-z \text{ times}} & 1 & \underbrace{1 \cdots \cdots 1}_{z-1 \text{ times}} \end{bmatrix}$$

where \otimes is the kronecker delta product.

Appendix B

Train Downsampling Decimation Operator

Corresponding train downsampling operator is defined in [40]. As such we can state that the discrete train downsampling operator = $\mathbf{D} \in \mathbb{R}^{[\frac{N}{L}]^d \times (N-L+1)^d}$ with decimation factor $L \in \mathbb{N}$ as

$$\mathbf{D} := \underbrace{\mathbf{D}_1 \otimes \cdots \otimes \mathbf{D}_1}_{d \text{ times}}$$

where $\mathbf{D}_1 \in \mathbb{R}^{[\frac{N}{L}] \times (N - L + 1)}$ is

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & \underbrace{0 \dots \dots \dots 0}_{N-L-1 \text{ times}} \\ \underbrace{0 \dots \dots 0}_{L \text{ times}} & 1 & \underbrace{0 \dots \dots \dots 0}_{N-2L-1 \text{ times}} \\ \underbrace{0 \dots \dots 0}_{2L \text{ times}} & 1 & \underbrace{0 \dots \dots \dots 0}_{N-3L-1 \text{ times}} \\ \vdots & \vdots & \vdots \\ \underbrace{0 \dots \dots \dots 0}_{([\frac{N}{L}-1])L \text{ times}} & 1 & \underbrace{0 \dots \dots \dots 0}_{N - ([\frac{N}{L}])L - 1 \text{ times}} \end{bmatrix}$$

where \otimes is the kronecker delta product.

Appendix C

Conjugate Gradient Descent

The conjugate gradient descent theorem follows as such

Theorem C.0.1. *Using symmetric and positive definite matrix Q , let $\{\underline{d}_0, \underline{d}_1, \dots, \underline{d}_{n-1}\}$ vectors be Q conjugate if*

$$\underline{d}_i^T Q \underline{d}_j, \quad \forall i \neq j$$

Since vectors $\{\underline{d}_0, \underline{d}_1, \dots, \underline{d}_{n-1}\}$ are Q -conjugate they are linearly independent, so x^ becomes*

$$x^* = \alpha_0 \underline{d}_0 + \dots + \alpha_{n-1} \underline{d}_{n-1}$$

After a multiplication of a common factor and rearranging, we get

$$\alpha_i = \frac{\underline{d}_i^T b}{\underline{d}_i^T Q \underline{d}_i}$$

The Q conjugancy allows us to build solution x^ as*

$$x^* = \sum_{i=0}^{N-1} \alpha_i \underline{d}_i = \sum_{i=0}^{N-1} \frac{\underline{d}_i^T b \underline{d}_i}{\underline{d}_i^T Q \underline{d}_i}$$

The solution to our objective no longer requires a computation of any inverses, but rather only requires a series of inner product calculations. Using arbitrary starting point $\underline{x}_0 \in \mathbb{R}^n$, we step toward minimum point of our objective using the following update rule

$$\underline{x}^{k+1} = \underline{x}^k + \alpha_k \underline{d}_k$$

Using initial direction and gradient vectors as

$$d_0 = -g_0 = b - Q\underline{x}_0$$

where g_0 is our initial gradient direction. Then, to update gradient at step k

$$g_k = Q\underline{x}_k - b$$

along with Q conjugate direction vectors and stepsize β_k

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

$$\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$$

Appendix D

Block Matching 3D Filtering Denoising Algorithm

The Block Matching 3D filtering algorithm (BM3D) [3] is an image noise removal algorithm that has shown high performance, and is considered a current state of the art method. The algorithm denoises a measurement \underline{y} using a two step method that differ with respect to the filtering operator for noise attenuation in the transform domain. We begin with an explanation of the first phase of the BM3D algorithm, which entails creating an initial approximation of denoised image with BM3D’s “block matching” methodology and applying hard thresholding to remove noise in a 3D transform domain. Once computing an initial approximation of the true signal is complete the second phase consists of developing a Wiener filter to further attenuate noise from the hard-threshold denoised signal created during phase one of the algorithm.

We follow the nomenclature of the block matching denoising algorithm BM3D proposed by Dabov et. al. [3]. Similar to before, noisy observations are of the form $\underline{z}(x) = \underline{y} + \underline{n}_{\sigma^2}$ where \underline{y} the true image and \underline{n}_{σ^2} white Gaussian noise with σ^2 variance. The variable Z_x denotes a block of fixed size $\mathcal{N}_1 \times \mathcal{N}_1$ extracted from noisy image \underline{x} located at pixel x of currently processed noisy measurement. The first phase of the

BM3D algorithm can be visualized in Figure D.1 where BM3D extracts similar intensity blocks of a noisy image y across different pixel locations given fixed some reference block Z_{x_R} with $x_R \in \Omega$ in Ω the image domain. Given currently processed reference block Z_{x_R} similar blocks are found from applying Block matching where neighboring blocks which exhibit a high level of correlation to currently processed reference block are found from a sliding operation. The sliding operation given reference block Z_{x_R} constitutes successively shifting in a sequential manner (usually left to right) to pixels x along Ω and extracting intensity blocks of fixed size $\mathcal{N}_1 \times \mathcal{N}_1$ which exhibit high correlation to reference block Z_{x_R} . Any extracted blocks that exhibit high correlation to reference block will be deemed as the image “reference fragments”. A few assumptions regarding the BM3D include the following.

1. Some of the blocks of fixed size $\mathcal{N}_1 \times \mathcal{N}_1$ of the true denoised image exhibit mutual correlation.
2. Selected 3D transform unitary operator converts specified blocks into a sparse representation.
3. Standard deviation of additive white Gaussian noise can be estimated *a priori*.

Thus following the formulation of the BM3D algorithm to compute correlation between blocks the d -distance measure

$$d(Z_{x_1}, Z_{x_2}) = N_1^{-1} \left\| \Upsilon \left(\mathcal{T}_{2D}(Z_{x_1}), \lambda_{thr2D\sigma} \sqrt{2 \log(N_1^2)} \right) - \Upsilon \left(\mathcal{T}_{2D}(Z_{x_2}), \lambda_{thr2D\sigma} \sqrt{2 \log(N_1^2)} \right) \right\|_2 \quad (\text{D.1})$$

with $x_1, x_2 \in \Omega$, \mathcal{T}_{2D} is a 2D linear unitary transform¹ operator such as Discrete Fourier Transform (DFT), Υ is a hard-threshold operator, $\lambda_{thr2D\sigma}$ the fixed threshold parameter, and $\|\cdot\|_2$ denotes the ℓ^2 -norm. The hard threshold operator Υ is defined as

¹A unitary transform $U : A \mapsto B$ preserves inner products so that $(x, y)_A = (Ux, Uy)_B$ where A and B are Hilbert spaces. Examples include but are not limited to rotations, shifts, and fourier transform.

$$\Upsilon(\lambda, \lambda_{thr2D}) = \begin{cases} \lambda & \text{if } |\lambda| > \lambda_{thr} \\ 0 & \text{otherwise.} \end{cases} \quad (\text{D.2})$$

After applying the “block matching” procedure the set of blocks $\mathbf{S}_{x_R} \subseteq \Omega$ with similar coordinates to Z_{x_R} are found according to distance measure (D.1); the set \mathbf{S}_{x_R} is defined as

$$\mathbf{S}_{x_R} = \{x \in \Omega \mid d(Z_{x_R}, Z_x) < \tau_{match}\}$$

where τ_{match} is the maximum d-distance which two blocks are considered to be similar. Of course $d(Z_{x_R}, Z_{x_R}) = 0$. Once all noisy similar blocks $Z_{x \in \mathbf{S}_{x_R}}$ are matched, they are stacked in a 3D array of size $\mathcal{N}_l \times \mathcal{N}_l \times |\mathbf{S}_{x_R}|$ denoted as matrix $\mathbf{Z}_{S_{x_R}}$. Following this a unitary 3D transform \mathcal{T}_{3D} on $\mathbf{Z}_{S_{x_R}}$ is applied to compute a sparse representation of the true denoised signal. Image noise is attenuated by hard-thresholding transform coefficients of sparse representation of transformed signal $\mathbf{Z}_{S_{x_R}}$. After noise attenuation of stack $\mathbf{Z}_{S_{x_R}}$ inverse transform operator \mathcal{T}_{3D}^{-1} yields a 3D array of denoised estimates according to hard-thresholding piecewise function

$$\hat{\mathbf{Y}}_{x_R} = \mathcal{T}_{3D}^{-1}(\Upsilon(\mathcal{T}_{3D}(\mathbf{Z}_{S_{x_R}}), \lambda_{thr3D\sigma} \sqrt{2 \log(N_l^2)})) \quad (\text{D.3})$$

with $\lambda_{thr3D\sigma}$ a fixed threshold parameter, and the number of image fragments of array $\hat{\mathbf{Y}}_{x_R}$ equal to $|\mathbf{S}_{x_R}|$ since it is composed from local block estimates $\hat{\mathbf{Z}}_{x \in \mathbf{S}_{x_R}}^{x_R}$.

Once all fragments of set \mathbf{S}_{x_R} are denoised, each fragments similarity with respect to reference block x_R is measured by weight function

$$w_{x_R} = \begin{cases} \frac{1}{N_{har}} & \text{if } N_{har} \geq 1 \\ 1 & \text{otherwise.} \end{cases}$$

with N_{har} the number of non-zero transform coefficients after applying hard-thresholding operator. After looping through all pixels in the image and processing corresponding

reference blocks with (D.3), a set of local block estimates $\hat{\mathbf{Z}}_{x \in \mathbf{S}_{x_R}}^{x_R}$ and their associated weights $w_{x_R}, \forall x_R \in \Omega$ are available to create a denoised estimate

$$e(\underline{x}) = \frac{\sum_{x_R \in \Omega} \sum_{x \in \mathbf{S}_{x_R}} w_{x_R} \hat{Y}_{x_m}^{x_R}(x)}{\sum_{x_R \in \Omega} \sum_{x \in \mathbf{S}_{x_R}} w_{x_R} \chi_{x_m}(x)}, \quad \forall x \in \Omega \quad (\text{D.4})$$

with $\chi_{x_m} : \Omega \rightarrow \{0, 1\}$ the characteristic function of the square support of a block located at $x_m \in \Omega$. The term $\hat{Y}_{x_m}^{x_R}(x)$ is an estimate of denoised image with $x, x_R \in \Omega$ and slices of given pixel $x_m \in \mathbf{S}_{x_R}$. This step concludes the first phase of the algorithm, intended to create an initial approximate of initial denoised image.

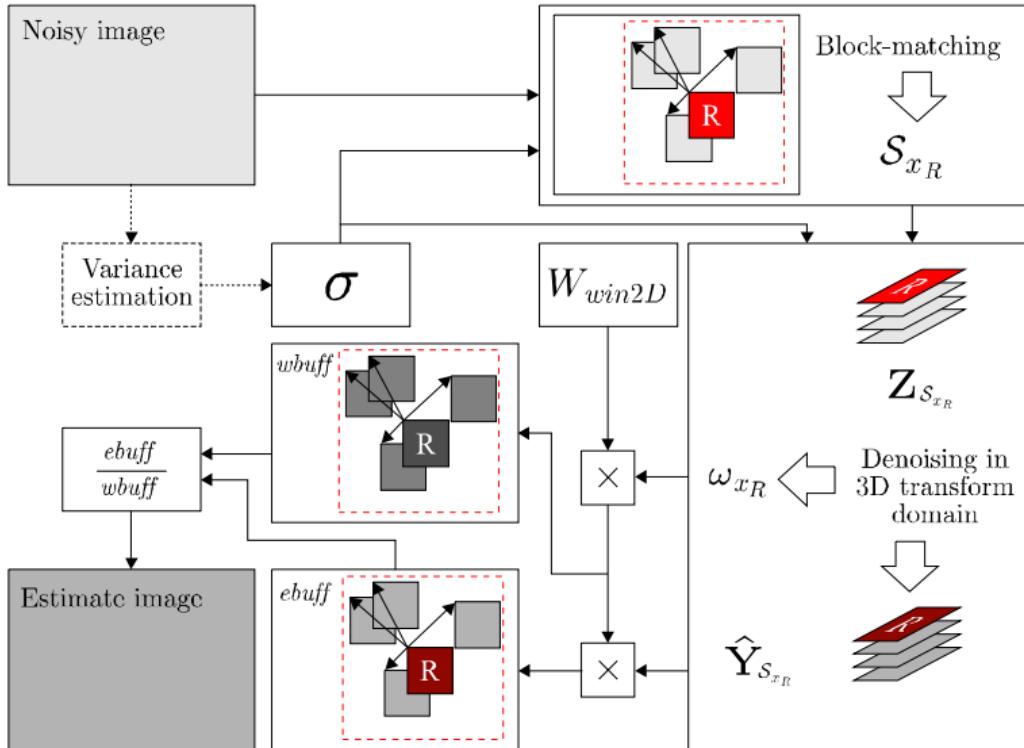


Figure D.1: Flowchart for denoising by hard-thresholding in 3D transform domain with block-matching. A similar flowchart can be constructed for the second phase of the algorithm, but instead denoising using a Wiener filter in 3D transform domain.

Figure 1.4 courtesy of Dabov A. et. al, Image denoising with blockmatching and 3D filtering, VOL 6064, pg. 606414, 2006.

According to [3] given an initial estimate $e : \Omega \rightarrow \mathbb{R}$ of the denoised image, an empirical Wiener filter can then be computed to replace the hard-thresholding filter from phase one of the algorithm. In the second phase of the BM3D algorithm, the “block matching” procedure is repeated but rather on the estimate of the denoised image computed from phase one. Thus, the new d -distance measure which constitutes the set of neighbors for each reference pixel $x \in \Omega$ is

$$\mathbf{S}_{x_R} = \{x \in \Omega | N_1^{-1} \|(E_{x_r} - \bar{E}_{x_r}) - (E_x - \bar{E}_x)\|_2 < \tau_{match}\}, \quad (\text{D.5})$$

with \bar{E}_{x_R} and \bar{E}_x are the mean values of the estimated image blocks E_{x_R} and E_x respectively. The intention of mean subtraction improves matching of blocks with similar structure but different mean values. As stated, in this phase of the algorithm the Wiener filter replaces the hard-thresholding operator (D.2). In this application, the Wiener filter restores a degraded signal through an inverse filtering process analogous to a minimization of the overall mean square error. Following this filtering process a denoised version of the noisy measurement will be approximated. Though, coefficients for the Wiener filter must be found from an initial approximation $e(\underline{x})$ of a low resolution measurement. As such the initial estimate $e(x)$ is brought into the frequency domain after applying 3D transform operator \mathcal{T}_{3D} . Secondly, coefficients for the Wiener filter are computed from the following

$$\mathbf{W}_{S_{x_R}} = \frac{|\mathcal{T}_{3D}(\mathbf{E}_{S_{x_R}})|^2}{|\mathcal{T}_{3D}(\mathbf{E}_{S_{x_R}})|^2 + \sigma^2}$$

where $\mathbf{E}_{S_{x_R}}$ represents a 3D stacked array built from the matched blocks $E_{x \in S_{x_R}}$ (similar to the construction of stacked matrix $\mathbf{Z}_{S_{x_R}}$ from $Z_{x \in \mathbf{S}_{x_R}}$). The stacked noisy observations $\mathbf{Z}_{S_{x_R}}$ (after applying transform operator \mathcal{T}_{3D} to convert to 3D domain) are filtered by an element wise multiplication with Wiener filter $\mathbf{W}_{S_{x_R}}$. Following this then an inverse 3D transform \mathcal{T}_{3D}^{-1} is applied to convert the signal back to the spatial domain. The formulation for this process is given by

$$\hat{\mathbf{Y}}_{S_{x_R}} = \mathcal{T}_{3D}^{-1} \left(\mathbf{W}_{S_{x_R}} \mathcal{T}_{3D}(\mathbf{Z}_{S_{x_R}}) \right), \quad (\text{D.6})$$

with $\hat{\mathbf{Y}}_{S_{x_R}}$ the stacked denoised local block estimates $\hat{Y}_{S_{x_R}}^{x_R}$ at matched locations $x \in S_{x_R}$.

The weights for each slice of $\hat{\mathbf{Y}}_{S_{x_R}}$ are assigned via

$$w_{x_R} = \left(\sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \sum_{t=1}^{|S_{x_R}|} |\mathbf{W}_{S_{x_R}}(i, j, t)|^2 \right)^{-1}. \quad (\text{D.7})$$

The final denoised image is an aggregation similar to (D.4), but rather the weights to scale each reconstructed block estimate $\hat{Y}_{x \in S_{x_R}}^{x_R}$ are calculated using (D.7).



Figure D.2: Denoised house results after applying BM3D denoising algorithm. The images on the left correspond to a noisy ($\sigma = 35$) House and enlarged fragments from it; On the right are the denoised image (PSNR=31.21 dB) and corresponding image fragments.

As illustrated from the house data in Figure D.2 the BM3D denoising algorithm does a fantastic job at noise removal, and in terms of objective criterion, outperforms all currently known denoising methods [3].

Bibliography

- [1] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.
- [2] J. C. Whitaker, *The Electronics Handbook*. 2005.
- [3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3D filtering,” vol. 6064, p. 606414, 2006.
- [4] C. C. Kang and W. J. Wang, “A novel edge detection method based on the maximizing objective function,” *Pattern Recognition*, vol. 40, no. 2, pp. 609–618, 2007.
- [5] R. M. Yan, Y. F. Zhang, Y. S. Li, and C. K. Wu, “Super-resolution image reconstruction based on guided cost function,” *Proceedings - International Conference on Image Processing, ICIP*, no. 60702058, pp. 1985–1988, 2010.
- [6] J. Yang, J. Wright, S. T. Huang, and M. Yi, “Image super-resolution via sparse representation,” *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [7] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2Noise: Learning Image Restoration without Clean Data,” no. 3, 2018.
- [8] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, “Supervised learning of image restoration

- with convolutional networks,” *Proceedings of the IEEE International Conference on Computer Vision*, 2007.
- [9] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-h. Xue, and Q. Liao, “Deep Learning for Single Image Super-Resolution :,” pp. 1–17.
- [10] J. M. Bioucas-Dias, a. T. Figueiredo, P. Oliveira, T. Norte, and A. R. Pais, “Total Variation-Based Image Deconvolution: A Majorization- Minimization approach,” no. 1, pp. 861–864, 2006.
- [11] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, pp. 259–268, 1992.
- [12] V. Katkovnik, “Anisotropic local likelihood approximations: theory, algorithms, applications,” *Proceedings of SPIE*, pp. 181–192, 2005.
- [13] I. Selesnick, “Total variation denoising (an MM algorithm),” *NYU Polytechnic School of Engineering Lecture Notes*, 2012.
- [14] M. Zhang and B. K. Gunturk, “Multiresolution bilateral filtering for image denoising,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, 2008.
- [15] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.
- [16] A. Buades, B. Coll, and J.-M. Morel, “A Non-local Algorithm for Image Denoising,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, no. 0, pp. 60–65, 2005.
- [17] M. Elad and M. Aharon, “Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries,” vol. 15, no. 12, pp. 3736–3745, 2006.

- [18] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision*, vol. 86, no. 1, pp. 1–32, 2010.
- [19] V. Tikhonov, A.N. and Arsenin, "Solutions of Ill-Posed Problems," no. Winston and Sons, 1977.
- [20] H. K., "Introduction to Bayesian image analysis," *Medical Imaging: Image Processing*, vol. 836, pp. 716–731, 1993.
- [21] J. Yang and T. Huang, "Image super-resolution: Historical overview and future challenges," *Super-resolution imaging*, pp. 3–25, 2010.
- [22] Y. Romano and M. Elad, "Boosting of Image Denoising Algorithms," vol. 8, no. 2, pp. 1187–1219, 2015.
- [23] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5136–5151, 2014.
- [24] S. Bougleux, A. Elmoataz, and M. Melkemi, "Discrete Regularization on Weighted Graphs for Image and Mesh Filtering," *Scale Space and Variational Methods in Computer Vision*, pp. 128–139.
- [25] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [26] 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.

- [27] F. G. Meyer and X. Shen, “Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising,” *Applied and Computational Harmonic Analysis*, vol. 36, no. 2, pp. 326–334, 2014.
- [28] G. Gilboa and S. Osher, “Nonlocal Operators with Applications to Image Processing,” *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008.
- [29] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-b. Sibarita, and J. Salamero, “Patch-Based Nonlocal Functional for Denoising Fluorescence Microscopy Image Sequences,” vol. 29, no. 2, pp. 442–454, 2010.
- [30] F. Chung, *Spectral Graph Theory*. AMS, vol 92 ed., 1997.
- [31] M. R. Charest, M. Elad, and P. Milanfar, “A general iterative regularization framework for image denoising,” *2006 IEEE Conference on Information Sciences and Systems, CISS 2006 - Proceedings*, vol. 95064, pp. 452–457, 2007.
- [32] Y. Romano and M. Elad, “Improving K-SVD denoising by post-processing its Method-Noise,” *Department of Computer Science Technion, Haifa 32000, Israel*, pp. 435–439, 2013.
- [33] H. Talebi, X. Zhu, and P. Milanfar, “How to SAIF-ly boost denoising performance,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1470–1485, 2013.
- [34] M. C. Chiang and T. E. Boult, “Efficient super-resolution via image warping,” *Image and Vision Computing*, vol. 18, no. 10, pp. 761–771, 2000.
- [35] M. Elad and Y. Hel-or, “A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space-invariant Blur,” *Image (Rochester, N.Y.)*, vol. 10, no. 8, pp. 1187–1193, 2001.
- [36] M. S. Alam, J. G. Bognar, R. C. Hardie, and B. J. Yasuda, “Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased

- video frames,” *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 915–923, 2000.
- [37] H. S. Hsieh and H. C. Andrews, “Cubic Splines for image Interpolation and Filtering,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-26, no. 6, pp. 508–517, 1978.
- [38] R. G. Keys, “Cubic Convolution Interpolation for Digital Image Processing,” no. I, pp. 1153–1160, 1981.
- [39] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang, “Image super-resolution: The techniques, applications, and future,” *Signal Processing*, vol. 128, pp. 389–408, 2016.
- [40] M. Ebrahimi, E. R. Vrscay, and A. L. Martel, “Coupled Multiframe Super-Resolution with Diffusive Motion Model And Total Variation Regularization,” pp. 62–69, 2009.
- [41] P. Samarasinghe and R. Kennedy, “Regularized Image Restoration,” *Image Restoration - Recent Advances and Applications*, vol. 6, pp. 111–133, 2012.
- [42] M. R. Banham and A. K. Katsaggelos, “Digital image restoration,” *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.
- [43] D. Glasner, S. Bagon, and M. Irani, “Super Resolution from A Single Image.”.
- [44] P. A. Knight and D. Ruiz, “A fast algorithm for matrix balancing,” *IMA Journal of Numerical Analysis*, vol. 33, no. 3, pp. 1029–1047, 2013.
- [45] P. Milanfar, “Symmetrizing Smoothing Filters,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 263–284, 2013.
- [46] M. Hussein, F. Porikli, and L. Davis, “Kernel integral images: A framework for fast non-uniform filtering,” *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, no. Figure 1, 2008.

- [47] Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, E. P. Simoncelli, and S. Member, “Image Quality Assessment : From Error Visibility to Structural Similarity,” vol. 13, no. 4, pp. 600–612, 2004.