

2017-01-01

Automatic Patch Classification And Diffusion In Adaptive

Luis Alonso Ponce

University of Texas at El Paso, laponce2@miners.utep.edu

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Ponce, Luis Alonso, "Automatic Patch Classification And Diffusion In Adaptive" (2017). *Open Access Theses & Dissertations*. 522.
https://digitalcommons.utep.edu/open_etd/522

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

AUTOMATIC PATCH CLASSIFICATION AND DIFFUSION IN ADAPTIVE
SUPER-RESOLUTION BASED ON OPTIMAL RECOVERY

LUIS ALONSO PONCE LOYA

Master's Program in Electrical Engineering

APPROVED:

Sergio D. Cabrera, Ph.D., Chair

Miguel Velez-Reyes, Ph.D.

Granville Sewell, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Copyright ©

by

Luis A. Ponce

2017

AUTOMATIC PATCH CLASSIFICATION AND DIFFUSION IN ADAPTIVE
SUPER-RESOLUTION BASED ON OPTIMAL RECOVERY

by

LUIS ALONSO PONCE LOYA

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
THE UNIVERSITY OF TEXAS AT EL PASO

May 2017

DEDICATION

Family and Friends for their support.

Dr. Sergio D. Cabrera

Acknowledgements

I would first like to express my profound gratitude to my thesis advisor Dr. Sergio D. Cabrera for his encouragement, support and dedication on this thesis. His guidance helped me to overcome the challenges encountered in this research. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Miguel Velez-Reyes and Dr. Grandville Sewell for their commitment and comments on this thesis.

I am grateful to receive the funding source Texas Instruments Foundation Scholarship that allowed me to pursue my graduate school studies.

Finally, I must express my thankfulness to my loved ones for their unfailing support throughout my Master's Degree. This accomplishment would not have been possible without them.

Abstract

Multi-Frame image restoration is a form of Super-Resolution (SR) which consists of combining multiple Low-Resolution (LR) images in order to reconstruct a single High-Resolution (HR) image. In this thesis we improved a spatially-adaptive SR scheme, based on the framework of Optimal Recovery (OR), whereby the block-by-block processing is done based on the properties of the corresponding local LR image data patches. The bandwidth parameter is adapted based on the local variance of each corresponding block from all LR images that contribute to the HR image block to be reconstructed. Also, an optimal regularization parameter needs to be chosen for the reconstruction of each HR output block. Some improvements achieved over a previous MATLAB implementation include: remove anti-aliasing filter to generate LR images so we can recover HR frequencies, implement a different shift registration for LR images, improve the selection of parameters and block classes according to the input image, bypass the use of Generalized Cross-Validation (GCV) for computation efficiency, implement a variable block size selection, allow different input image sizes, apply code debugging techniques to make the code more efficient in time, and solve image boundaries problems. Furthermore, we extended this approach to obtain estimated gradients as well as to generate edge maps by using OR theory. We also integrated an anisotropic diffusion method in two different ways, as a global post processing step on the whole reconstructed HR image or as a local reconstructed patch post processing step. Final results show significant improvement over previous versions of this approach and over many other SR algorithms of low computational complexity.

Table of Contents

	Page
Acknowledgements	v
Abstract	vi
Table of Contents	vii
List of Tables	ix
List of Figures	x
Chapter	
1 Introduction	1
1.1 Theory of Multi-Frame Super-Resolution.	1
1.2 Motivation	3
1.3 Thesis Overview	5
2 Theory of Optimal Recovery (OR)	6
2.1 Optimal Recovery for One-Dimensional Signals.	6
2.2 Optimal Recovery for Two-Dimensional Signals.	8
3 OR Gradient Estimation in Multi-Frame Super-Resolution	10
3.1 2-D Optimal Recovery gradient estimation from LR images.	10
3.2 Comparisons with Sobel gradient estimation and edge map	14
4 Theory and extensions of Anisotropic Diffusion	17
4.1 Anisotropic vs Isotropic Diffusion	18
4.2 Perona-Malik Anisotropic Diffusion	20
4.3 Extensions by Perona-Malik	22
5 Super-Resolution with Optimal Recovery review and improvements including Anisotropic Diffusion	24
5.1 Improve spatially adaptive SR-OR	24
5.1.1 Shift Registration	25
5.1.2 Flexible block and image sizes	29
5.1.3 Code Debugging	33

5.1.4 Limited bandwidth constant P	34
5.1.5 Global optimal regularization parameter	37
5.1.6 Multi-threshold vs function for bandwidth selection	42
5.1.7 K-means clustering for class representation	44
5.1.8 Boundary problem	55
5.2 Block segmentation with edge map	56
5.4 Incorporation of diffusion to our OR algorithm	61
5.4.1 Post Processing Anisotropic Diffusion	61
5.4.2 Block base post processing Anisotropic Diffusion with improved gradient	64
6 Revisiting Papoulis-Gerchberg (PG)	69
6.1 The PG SR method and code fixes	69
6.2 Maiseli diffusion enhancement SR of PG	70
7 Experiments and Results	73
7.1 Lower bound results with maximum bandwidth ($a=\pi$)	75
7.2 Revisit Overlap vs No Overlap OR Results.	76
7.3 Examples OR with Anisotropic Diffusion	78
7.4 New Comparison with others SR methods	80
8 Conclusion and Future Work	103
References	106
Curriculum Vitae	111

List of Tables

Table 1: Class Bandwidths for the cameraman image.....	55
Table 2: Cameraman image results MSE and SSIM (best result are given in bold font)	95
Table 3 Liftingbody image results MSE and SSIM (best result are given in bold font)	102

List of Figures

Figure 1 a) Super-Resolution (SR) is the process of combining multiple Low-Resolution (LR) images to form one High-Resolution (HR) image, b) Top level block diagram: registration is known in this case, we focus on the SR algorithm.	2
Figure 2 LR generator: to generate our LR images for experiments, we require an original image which passes through different steps, as show.....	2
Figure 3 Estimated Gradients \hat{G}_x , \hat{G}_y , and $\hat{G}_{magnitude}$ from a reconstructed image from chapter 5.	12
Figure 4 Comparison between our estimated magnitude gradient and the ground of truth gradient obtained using Sobel edge detector on the original image and Gradient obtained from the reference LR after simple linear interpolation.	14
Figure 5 Comparison of our edge map and the Sobel segmentation from the original image ...	16
Figure 6 Edges are preserved by using anisotropic diffusion (images courtesy from Dr. Granville Sewell), a) Original data, b) Isotropic diffusion, c) Anisotropic diffusion from Heat Equation.....	20
Figure 7 Difference between Isotropic and Anisotropic Diffusion in one image, a) Original image, b) Isotropic diffused image, c) Anisotropic diffused image [68]	21
Figure 8 Maiseli Super-Resolution algorithm using PG and a modified PM.	23
Figure 9 Block diagram for the generation of the LR images.....	26
Figure 10 LR images after the shift process	28

Figure 11 Different size block from our LR reference image extracted from the same area, a) 4x4 block size, b) 8x8 block size, c) 16x16 block size.	30
Figure 12 Error map from no-overlap approach using different block sizes, a) 4x4 block sizes, b) 8x8 block sizes, c) 16x16 block sizes.....	32
Figure 13 MSE result by varying the P parameter.....	35
Figure 14 Comparison between resulting image using $P=0.083$ and original image.....	35
Figure 15 zoomed version for better comparison between resultant image and original image.....	36
Figure 16 λ values generated by the GCV with $P=0.083$ using 4x4 blocks of the 4 LR images with 50% overlap.	38
Figure 17 log of the Variances of the input data blocks of the cameraman image.....	39
Figure 18 Scattergram from variances and log of the regularization parameter to look for any correlation between them.	39
Figure 19 Relationship between λ and P in terms of MSE results of the recovered image compared to the ground truth original.	40
Figure 20 Comparison between resultant image and original image looks almost identically.	41
Figure 21 zoomed version for comparison between resultant image and original image.....	41

Figure 22 Selection of the five main classes in an input image (Cameraman) by a multi-threshold method. Note that we need to select 4 different values for our multi-threshold to obtain 5 different classes in our result.....	43
Figure 23 K-Means values (Red) represents the representative variance for each block class while Multi-threshold values (Green) represents the limits for each block class that we use.	45
Figure 24 Bandwidth selection using multi-threshold and K-means method in block variances. Using the results from figure 23 and $P=0.1031$, we have $C1= 1.7522$, $C2= 3.3678$, $C3= 5.3923$, $C4=6.8890$, $C5= 8.1521$, $L1= 2.5457$, $L2= 4.2432$, $L3= 6.0852$, $L4= 7.4577$	45
Figure 25 Block Variance Map using block size of 16×16 in each LR image and 50% overlap	46
Figure 26 Location of Block classes in our input image Cameraman	49
Figure 27 Representative blocks for each class with its spectrum in each LR image, a) Class 1 example, b) Class 2 example, c) Class 3 example, d) Class 4 example, e) Class 5 example ...	54
Figure 28 Separation regions using labeling and our edges areas from our edge map inside an HR block, in this case we segment the block into three regions.	56
Figure 29 Input LR data bi is categorized into two different classes based on its variance, smooth block and edge block. Edge blocks are segmented into different regions Ri to reduce region variance and improve the SR process.	57
Figure 30 LR to HR regions mapping with region segmentation.....	58
Figure 31 HR Block reconstruction process and mapping using region segmentation.....	59

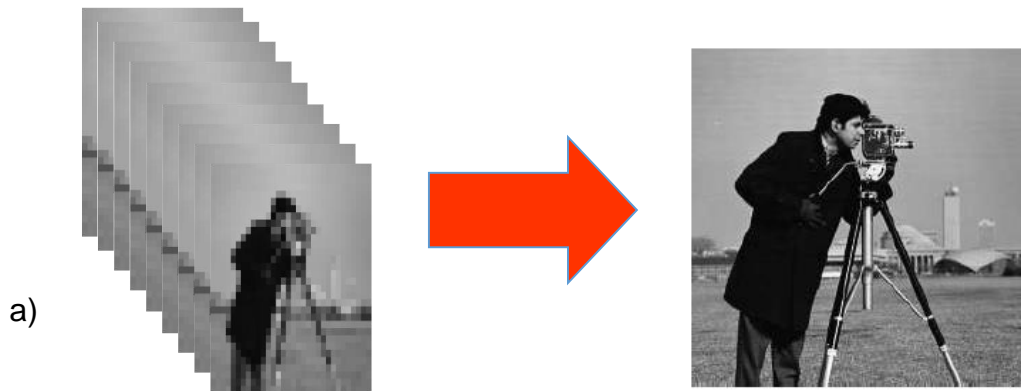
Figure 32 Comparison between resulting image with old approach (top) and with regions segmentation (bottom).	60
Figure 33 Original patches/blocks before extraction of subsets of pixels to become part of the final image.....	61
Figure 34 Comparison of Post-Processing diffusion with estimated gradients (top) and Sobel gradients (bottom)	62
Figure 35 Zoomed version of the comparison of Post-Processing with estimated gradients (top) and Sobel gradients (bottom)	63
Figure 36 Comparison of Block Post-Processing with estimated gradients (top) and Sobel gradients (bottom)	65
Figure 37 Zoomed version of the comparison of Block Post-Processing with estimated gradients (top) and Sobel gradients (bottom)	66
Figure 38 Estimated gradients after being diffused with original PM method	68
Figure 39 Papoulis-Gerchberg Results from MATLAB app after code fix.....	70
Figure 40 Divergence images in first and tenth iteration of Maiseli algorithm and its final result after 11 iterations.	72
Figure 41 Top level block diagram of our various SR-OR experiments and methods	73
Figure 42 More details of the SR-OR Algorithm block diagram	74
Figure 43 More details of estimated gradients SR-OR Algorithm block diagram	74

Figure 44 Results using single full bandwidth $\alpha = \pi$ on our OR Super-Resolution. Spatial and DFT-based spectral domain view using log contrast enhancement.....	76
Figure 45 Result using Overlap approach using all improvements from chapter 5	77
Figure 46 Result using No-Overlap approach using all improvements from chapter 5.....	78
Figure 47 OR Super-Resolution Result using Post-Processing Maiseli diffusion	79
Figure 48 OR Super-Resolution Result using Block Post-Processing Maiseli diffusion.....	80
Figure 49 Original Cameraman and its DFT (spectrum domain) view using log contrast	81
Figure 50 SR Results obtained from MATLAB using cameraman image. Each titled image is accompanied by its contrast spectrum plot with the same contrast enhancement as Fig 44....	94
Figure 51 MSE and SSIM results for each block (MSE is using log10 base for display purposes).	97
Figure 52 Original Liftingbody and its DFT (spectrum domain) view using log	98
Figure 53 Structure-Adaptive Normalized Convolution vs OR Super-Resolution with Overlap and No-Overlap for Liftingbody image	101
Figure 54 bandwidth of vertical and horizontal edges in an image	105

Chapter 1 Introduction

1.1 Theory of Multi-Frame Super-Resolution

In the classical multi-frame Super-Resolution (SR) problem illustrated in figures 1-2, multiple displaced, noisy and blurred Low Resolution (LR) images/frames are used to generate a single High Resolution (HR) image [1], [2]. In order to accomplish this, we need a method to estimate rotation and shifts in LR images using one LR image as a reference, this method is called registration. Then from the obtained registration information and a model for the blur, we can reconstruct a HR image using a combined process of frame-fusion and deconvolution. For our problem, we deal only in improving our SR algorithm, hence we assumed known registration parameters. We also worked only with gray intensity images. For color images, the images are converted into the YUV color space, and only the Y channel is processed by the SR algorithm. The U and V channels are interpolated, and the three components are then converted back to RGB color space to create the final SR result [1].



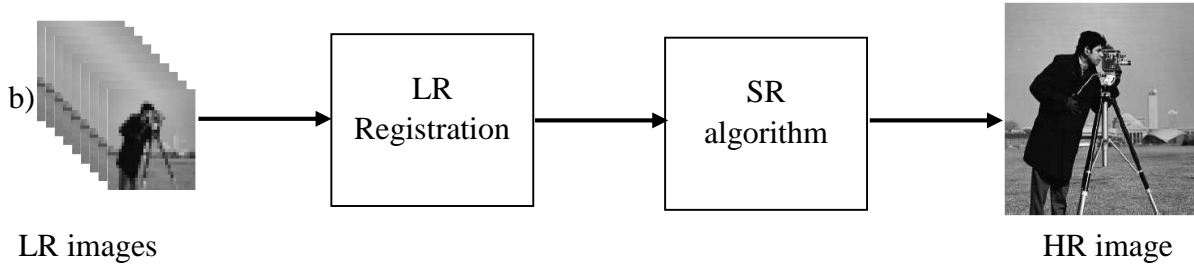


Figure 1 a) Super-Resolution (SR) is the process of combining multiple Low-Resolution (LR) images to form one High-Resolution (HR) image, b) Top level block diagram: registration is known in this case, we focus on the SR algorithm.

To test the quality of our results, a common practice in image reconstruction is to simulate LR images from an original ground truth HR image that is also used to compare with our final HR result. This LR generator, shown in figure 2, requires to include at least one downgrade option: warp, blurring, displacement, downsampling and/or noise. For our algorithm, we will use downsampling with no anti-aliasing and only displacements in the x and y direction with no rotations or blurring.

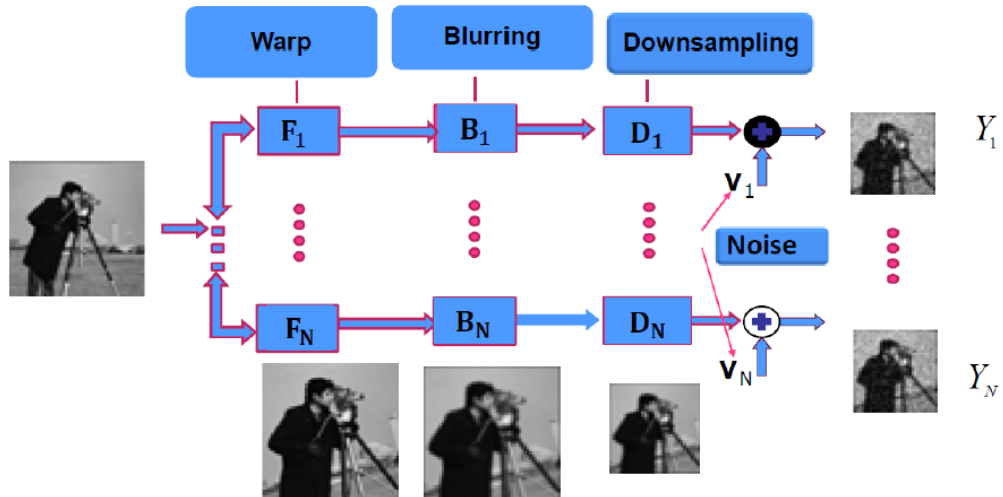


Figure 2 LR generator: to generate our LR images for experiments, we require an original image which passes through different steps, as show.

Additionally, a big problem about whether to apply image processing algorithms globally or locally has been discussed in the community [3]. On the one hand, global processing is done on the whole LR images simultaneously, such approach is often good at handling blur and noise present in LR images but more complex to use since some parts of the image may not be correlated with other parts of the image. On the other hand, local processing is done on patches/blocks of LR images simultaneously to generate an HR patch/block, such an approach is often good at handling edges properly since it focuses on specific parts of the image but it cannot take care of blur and noise present in the LR images. For our SR algorithm, we will use a local approach since we desire to preserve edges in our image and reduce algorithm complexity. Also, in order to increase the accuracy of our results, we will use an overlap approach to add more information to our processing. In this overlap approach, each block will be overlapped with the previous one and the next one in the x and y directions, then we will keep only the center subset of each reconstructed HR block. In general, the more information we use to recover an HR block, the better the quality of the reconstruction in the middle of the block. Overlap allows us to use many input LR pixels to recover a small number of HR pixels at a time, and the recovery of HR pixels is of higher quality since we only keep the pixels in the middle of the block.

1.2 Motivation

Today, the state-of-the-art SR methods being developed are based on mathematical optimization methods for inverse problems [2]. The most challenging applications are those with images taken from a video sequence with complex motion [4], [5]. Potential improvements in image and video resolution are useful in forensic analysis, surveillance, visual quality enhancement for consumer markets, highway traffic monitoring, remote sensing, medical

imaging, etc. We believe it can have a significant impact as a technological improvement for metrology applications that use digital cameras to make accurate measurements. This is called the video metrology problem [6] or optical metrology problem [7].

In addition, when using images to make measurements, control of a worst-case error is an improvement over average error or visual quality, this is the approach followed by the Optimal Recovery (OR) framework. The OR problem uses as input the value of a given information linear operator T (LR image data) to produce as output an optimal, deterministic estimate of the value of a desired linear operator U (HR image data) [10]. Our spatially-adaptive SR approach produces a small HR image patch from a given number of smaller or larger size overlapping LR input data patches. The adaptation can be based on image spectrum bandwidth as in [8]. Other researchers have obtained promising results with the OR framework in other applications [9].

Our long term objective will be to quantify the benefits of using SR (our methods and existing methods) and thus to compare with conventional approaches that do not incorporate super-resolution in video metrology. An important research question is how to use the edge enhancement ability of an anisotropic diffusion method together with OR theory to find a practical, simple new method of SR that is sufficiently adaptive to work on real images from video sequences and improve the quality of the source. A recent promising OR-based method developed by previous students and S. Cabrera [8] is promising. In this thesis, we improve and extend previous OR approaches by incorporating aspects from state-of-the-art image recovery frameworks such as anisotropic diffusion, gradient estimation and segmentation base on edge maps, and keep some degree of regularization optimization [2].

1.3 Thesis Overview

The organization of this thesis is the following: Chapter 2 explains the theory of Optimal Recovery in 1D and 2D. It explains how to obtain an optimal estimation from a set of feasible signals and how to reduce the error using a worst case error criterion. We will focus more in 2D signals to generate a Super-Resolution algorithm. Chapter 3 introduces the analytic expression to obtain estimated gradients using the OR theory of chapter 2. It shows an edge map obtained from estimated gradients and compares these gradients with a ground truth that uses the Sobel edge algorithm from MATLAB in the original image. Chapter 4 briefly introduces diffusion theory as used in image processing. It explains the difference between isotropic and anisotropic diffusion. The Perona-Malik (PM) anisotropic diffusion method and recent improvements used by Maiseli et al [21] are then introduced. They will be used in later chapters for our SR-OR method. Our main reference for this chapter comes from the book by Joachim Weickert [22]. Chapter 5 revisits previous work in adaptive SR-OR. It shows our modifications and improvements to previous algorithms as well as modifications in the code. Chapter 6 explains the Papoulis-Gerchberg SR algorithm and shows its results obtained from a MATLAB application (app) with a few modifications. It shows improvements acquired by Maiseli in combining the PG algorithm with the PM anisotropic diffusion method. Chapter 7 describes experiments performed with our improved SR-OR approach and the new extensions proposed. It also shows comparisons between the no-overlap and the overlap approach, and compares methods with and without diffusion. Likewise, it compares with different SR methods retrieved from the MATLAB app. Finally, chapter 8 explains conclusions and future work.

Chapter 2 Theory of Optimal Recovery (OR)

The Optimal Recovery Method is used for reconstructing signals from non-uniform samples [10] that uses a priori bandwidth or spectral shape information. The key objective is to minimize the worst case error using a norm to measure deviation over all possible feasible signals [11]. In this thesis we will discuss the use of OR for estimation of 1D and 2D signals as well as OR applied to the estimation of gradients of 2D signals for use in chapter 3.

2.1 Optimal Recovery for One-Dimensional Signals

Following the authors in [11], an a -bandlimited signal $f(t)$ can be represented as an orthogonal expansion using its equally spaced samples $f(kT_o)$, for $k=0,\pm1,\pm2,\dots$, for the Nyquist rate sampling period $T_o = \pi/a$ or when it is oversampled using $T = \pi/\sigma$ with any $\sigma > a$ represented with the formula

$$f(t) = \sum_{k=-\infty}^{\infty} f(kT) \frac{\sin a(t-kT)}{\sigma(t-kT)} \quad (1)$$

This equation can be simplified when $f(t)$ uses N uniform samples by

$$f_N(t) = \sum_{k=1}^N f(kT) \frac{\sin a(t-kT)}{\sigma(t-kT)} \quad (2)$$

The OR theory of deterministic estimation of a signal $f(t)$ from N arbitrary samples is applicable in the Hilbert space β_a of a -bandlimited signals. Consider the given measurements as an N -vector \underline{y} with entries $y_k = f(t_k)$, for $k=1,2,3,\dots,N$. Equivalently, the given samples are the values of an information linear transformation $y = I(f)$ that produces the N samples from the signal $f(t)$ [19]

The OR theory states that the signal can be estimated at any arbitrary location t_0 using the OR framework for the specific case of Bounded Linear Functional (BLF) estimation.

We assumed that the a -bandlimited signal $f(t)$ has a known bound on its energy given by

$$\|f\|^2 = \int_{-\infty}^{\infty} |f(\tau)|^2 d\tau \leq \varepsilon^2 \quad (3)$$

So all feasible signals belongs to the set

$$R = \left\{ f \in \beta_a : f(t_k) = y_k ; k = 1, 2, 3, \dots, N ; \|f\|^2 \leq \varepsilon^2 \right\} \quad (4)$$

Therefore, the optimal estimate and the worst case error to be minimized in the estimation of a BLF $f(t_o)$ are:

$$f_{t_0}^o = \arg \min_{f_{t_0}} \{E_{t_0}\} ; E_{t_0} = \sup_{f \in R} |f(t_0) - f_{t_0}| \quad (5)$$

By using the OR theory, the optimal solution at any fixed but arbitrary t_o is always the mid-point of the range of all possible values of $f(t_o)$, that value is $f_{t_0}^o = f_{MN}(t_0)$ where

$$f_{MN}(t) = \sum_{k=1}^N w(k) \frac{\sin a(t - t_k)}{\pi(t - t_k)} \quad (6)$$

Which it is the minimum norm (energy) signal in R . To find the vector \underline{w} for weights $w(k)$, we enforce the interpolation property of any $f \in R$: $f_{MN}(t_m) = y_m$, with $m=1, 2, \dots, N$. In general:

$$\underline{w} = S^{-1} \underline{y} \quad (7)$$

Thus, the minimum worst case error is

$$E_{t_0}^o = \sup_{f \in R} |f(t_0) - f_{MN}(t_0)| \quad (8)$$

A more general problem from equation (8) can be similarly formulated to estimate K values simultaneously. Let $U(f)$ be the set of K desired samples of $f(t)$, a linear transformation from β_a to \Re^K . To solve this version of the OR problem we need

$$\underline{z}^o = \arg \min_{\underline{z} \in \Re^K} \{E_U\}; \quad E_U = \sup_{f \in R} \|U(f) - \underline{z}\|_p \quad (9)$$

The solution is still a function of the minimum norm signal $f_{MN}(t)$, therefore, $z^o = U(f_{MN})$ and the minimum error, using a p -norm, is

$$E_U^o = \sup_{f \in R} \|U(f) - U(f_{MN})\|_p \quad (10)$$

Recent applications of OR include sensor deployment optimization [11]. By selecting additional sensor (sample) locations using the worst case error. Computer simulations of feasible signals help to produce estimates of the theoretical minimal worst-case error bounds.

2.2 Optimal Recovery for Two-Dimensional Signals

Extending the 1D OR theory from the previous section to 2D signals or images, the OR theory modifies $f_{MN}(t)$ to $f_{MN}(x, y)$ where x and y are the two variables that represent the spatial domain. Therefore we modify our equation (6) to

$$f_{MN}(x, y) = \sum_{i=1}^N w(i) h(x - x_i, y - y_i) \quad (11)$$

We can write our S matrix as the expansion:

$$S = \begin{bmatrix} h(x_1 - x_1, y_1 - y_1) & h(x_1 - x_2, y_1 - y_2) & \cdots & h(x_1 - x_{N-1}, y_1 - y_{N-1}) & h(x_1 - x_N, y_1 - y_N) \\ h(x_2 - x_1, y_2 - y_1) & \ddots & \cdots & \ddots & h(x_2 - x_N, y_2 - y_N) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h(x_{N-1} - x_1, y_{N-1} - y_1) & \ddots & \cdots & \ddots & h(x_{N-1} - x_N, y_{N-1} - y_N) \\ h(x_N - x_1, y_N - y_1) & h(x_N - x_2, y_N - y_2) & \cdots & h(x_N - x_{N-1}, y_N - y_{N-1}) & h(x_N - x_N, y_N - y_N) \end{bmatrix} \quad (12)$$

8

with

$$h(x, y) = \frac{\sin(ax)}{\pi x} \frac{\sin(ay)}{\pi y} \quad (13)$$

Thus, S is an $N \times N$ symmetric matrix of separable products of sinc functions evaluated at the differences of the N known pixel locations.

The spectral support is defined by $h(x, y)$ as a square bandwidth where $a \leq \pi$. Its Fourier transform is $H(u, v) = 1; |u| \leq \frac{a}{2\pi}, |v| \leq \frac{a}{2\pi}$, and zero elsewhere. In general, a bandlimited signal can also be expressed as:

$$F(u, v) = H(u, v)F(u, v) \quad (14)$$

Since S is ill-conditioned for $a < \pi$, a regularization strategy is used in solving for the \underline{w} vector.

Then equation (7) changes to

$$\underline{w} = (S + \lambda I)^{-1} \underline{y} \quad (15)$$

Finally, we can substitute (15) into (11) to obtain a closed form expression for our regularized

OR solution. In future chapters we will use \vec{b} to represent our data \underline{y} .

Chapter 3 OR Gradient Estimation in Multi-Frame Super-Resolution

In the area of image reconstruction, edge preservation is a big problem to deal with due to local high variance or high frequency components needed to represent edges. Commonly when these areas are reconstructed, they may present some undesired artifacts like staircasing, blurring, ringing, etc. Many researches have worked to solve this problem [41, 42]. One way to solve this approach is by integrating image gradients. Up to now, our reconstruction ignores edge areas using all LR pixels corresponding to the HR block, causing a mixing of different type of regions in the same block. For this reason, we will attempt to fix these edge areas by using estimated gradients produced by the OR theory to segment regions within the same HR block. Gradient estimates will also allow us to use a diffusion method that incorporates them.

3.1 2-D Optimal Recovery gradient estimation from LR images

In order to obtain our estimated gradients, we will use our OR optimal estimation equation (11) to reconstruct the gradients \widehat{G}_x and \widehat{G}_y of our image. To do that, we take advantage of the linearity property of derivatives and the closed form result of OR theory. We find an analytic expression of the partial derivative of $h(x, y)$ with respect to x for \widehat{G}_x (16) and with respect to y for \widehat{G}_y (17). Finally we can use (16) and (17) to obtain its magnitude (18) that can also be seen in Figure 3.

$$\widehat{G}_x(x, y) = \sum_{i=1}^N w(i) h_x(x_i - x, y_i - y) \quad (16)$$

$$\widehat{G}_y(x, y) = \sum_{i=1}^N w(i) h_y(x_i - x, y_i - y) \quad (17)$$

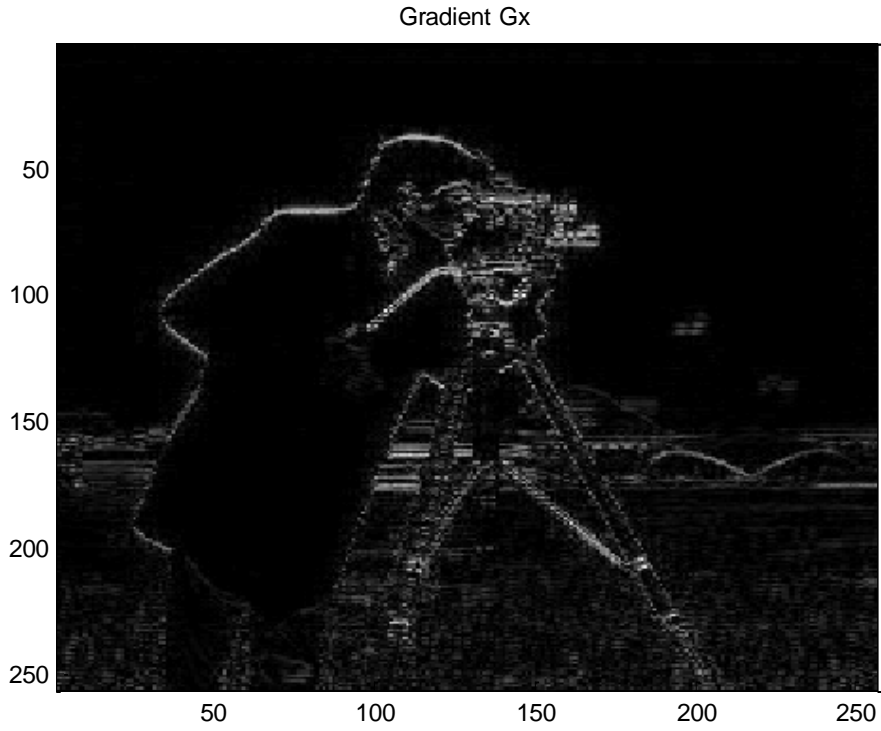
$$G_{mag}(x, y) = \sqrt{\widehat{G}_x^2 + \widehat{G}_y^2} \quad (18)$$

where

$$h_x(x, y) = \frac{\partial}{\partial x} h(x, y) = \frac{ax \cos(ax) - \sin(ax)}{\pi x^2} \cdot \frac{\sin(ay)}{\pi y} \quad (19)$$

$$h_y(x, y) = \frac{\partial}{\partial y} h(x, y) = \frac{\sin(ax)}{\pi x} \cdot \frac{ay \cos(ay) - \sin(ay)}{\pi y^2} \quad (20)$$

and
$$h(x, y) = \frac{\sin(ax)}{\pi x} \cdot \frac{\sin(ay)}{\pi y} \quad (21).$$



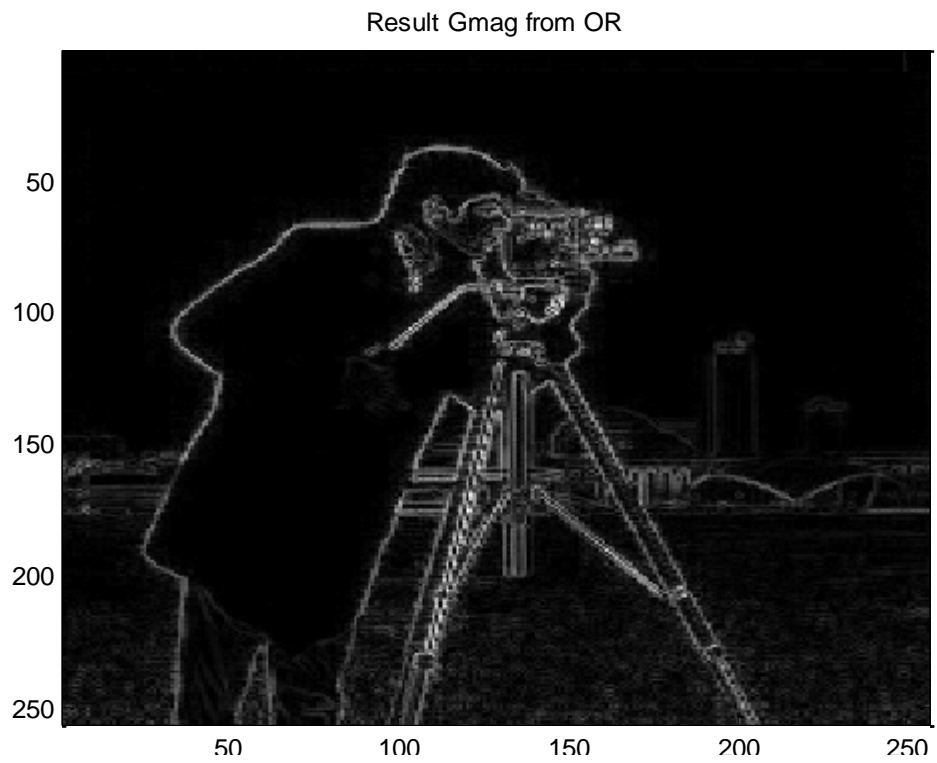
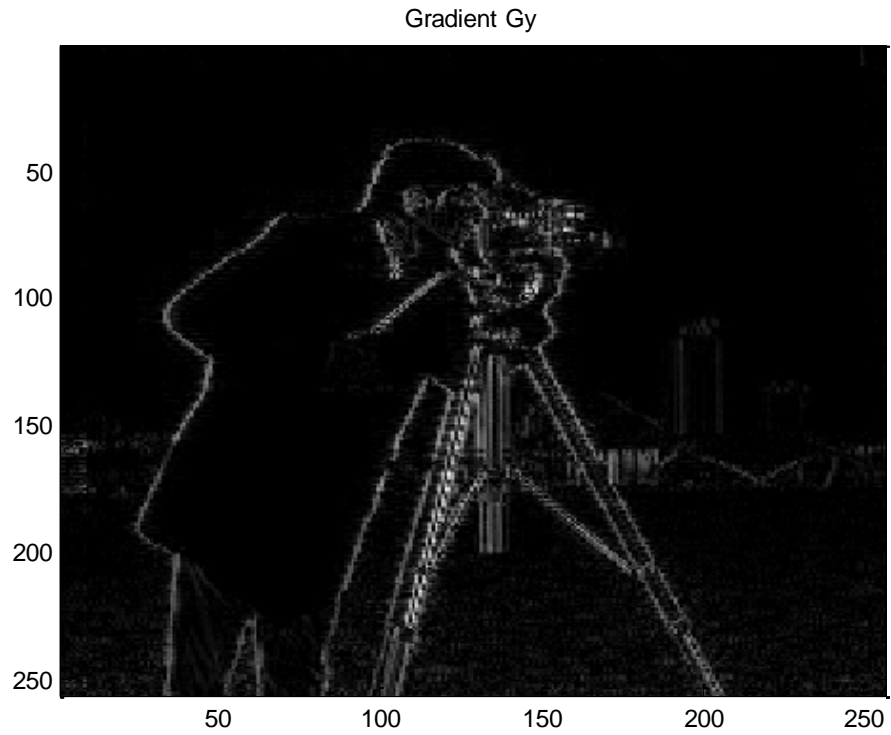
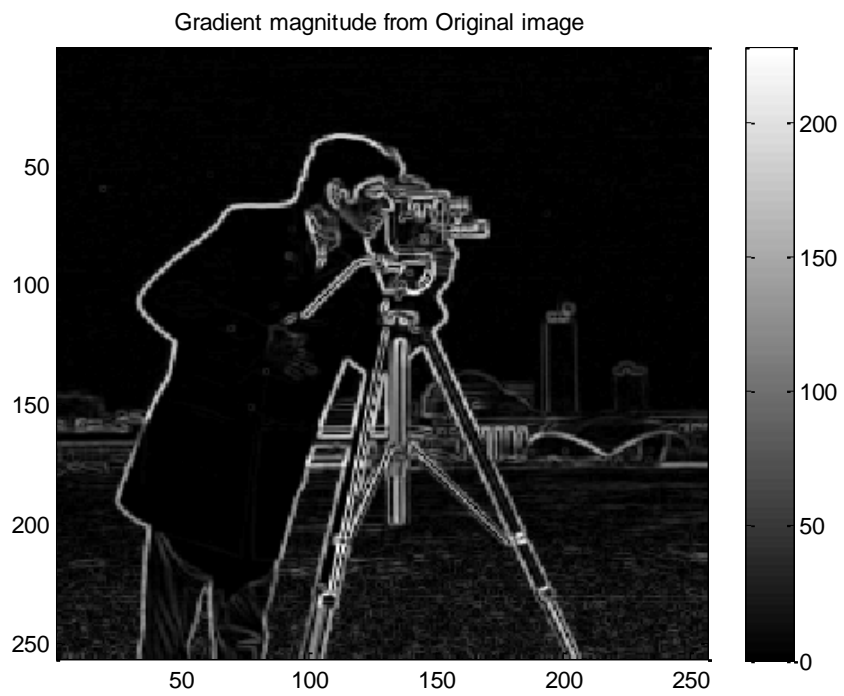
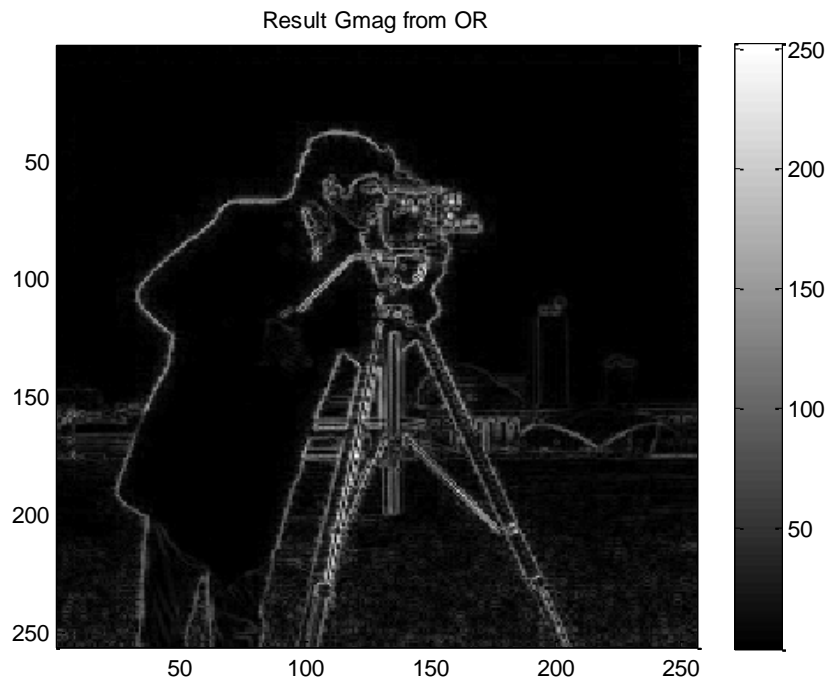


Figure 3 Estimated Gradients \hat{G}_x , \hat{G}_y , and $\hat{G}_{magnitude}$ from a reconstructed image from chapter 5.



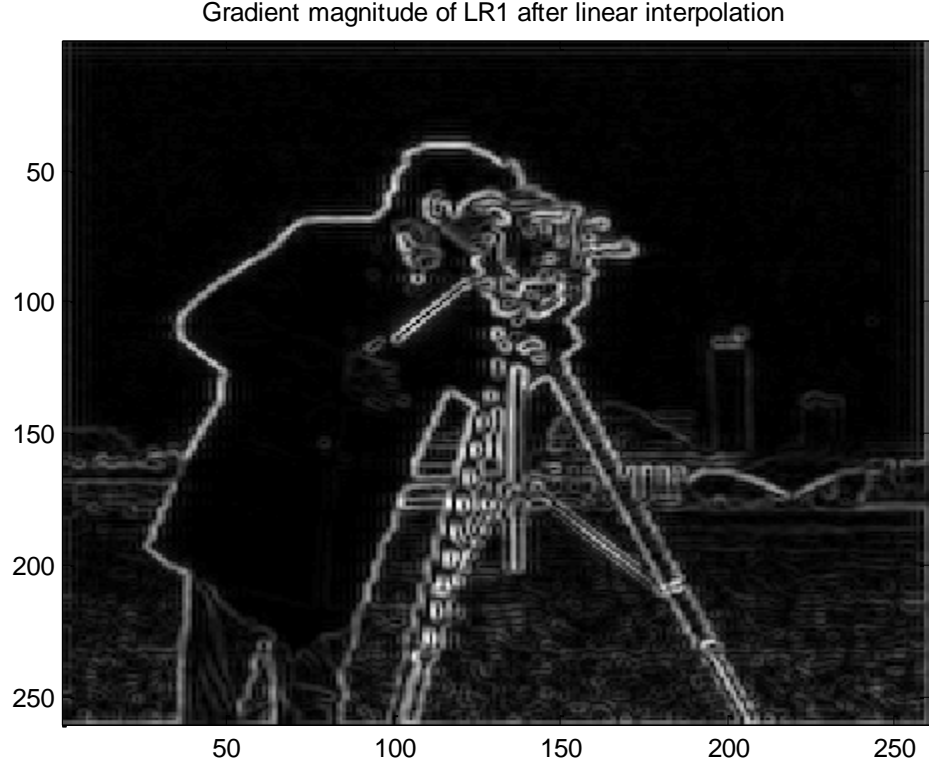


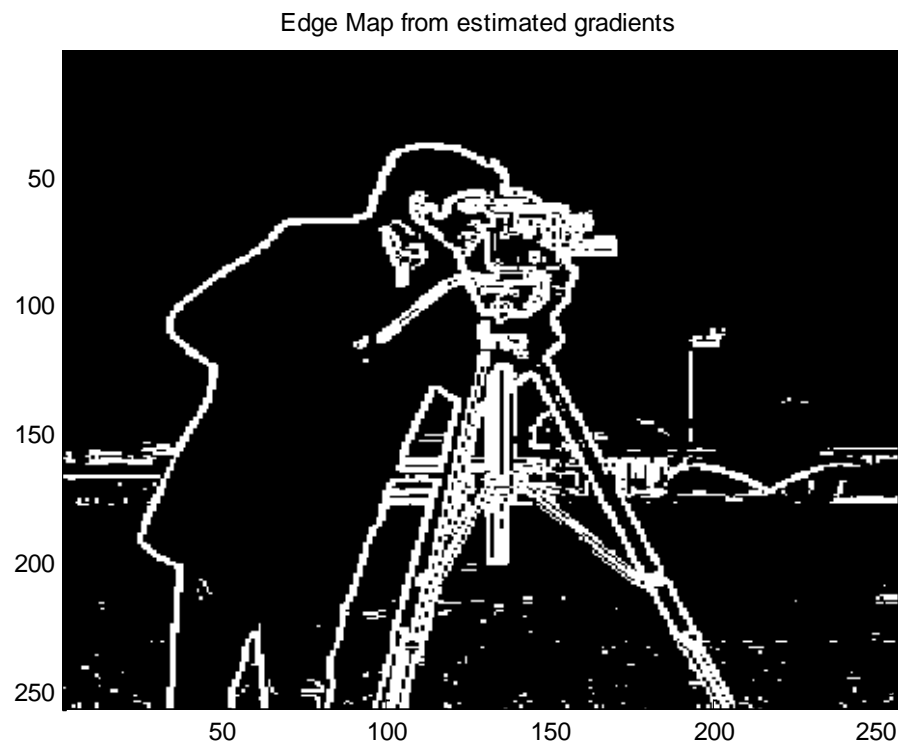
Figure 4: Comparison between our estimated magnitude gradient and the ground truth gradient obtained using Sobel edge detector on the original image and Gradient obtained from the reference LR after simple linear interpolation.

Also, we can obtain a Sobel edge map from the enlarged reference LR image with bilinear interpolation to compare with our estimated gradient magnitude in figure 4. As we can see, comparing our estimated gradients those from with the original image and from the enlarged LR image, we clearly obtain better results compared with those from the reference LR and we get results comparable to the gradient magnitude of the original image.

3.2 Comparisons with Sobel gradient estimation and edge map

By obtaining these estimated gradients, we can use them in methods that require gradients such as the anisotropic diffusion methods in [22, 48, 61]. Moreover, we can also

work with our gradients to generate an edge map. This can be used for different applications that requires binary images [16, 22]. Now that the magnitude of the gradient has been generated, the next step is to generate a binary image by using a threshold with Otsu's method [16] to remove false edges. We can then apply a morphological operation called area opening to get improved location of the edges. Again, a comparison is done between our edge map and the edge map obtained with the Sobel operator on the original Cameraman image in Figure 5.



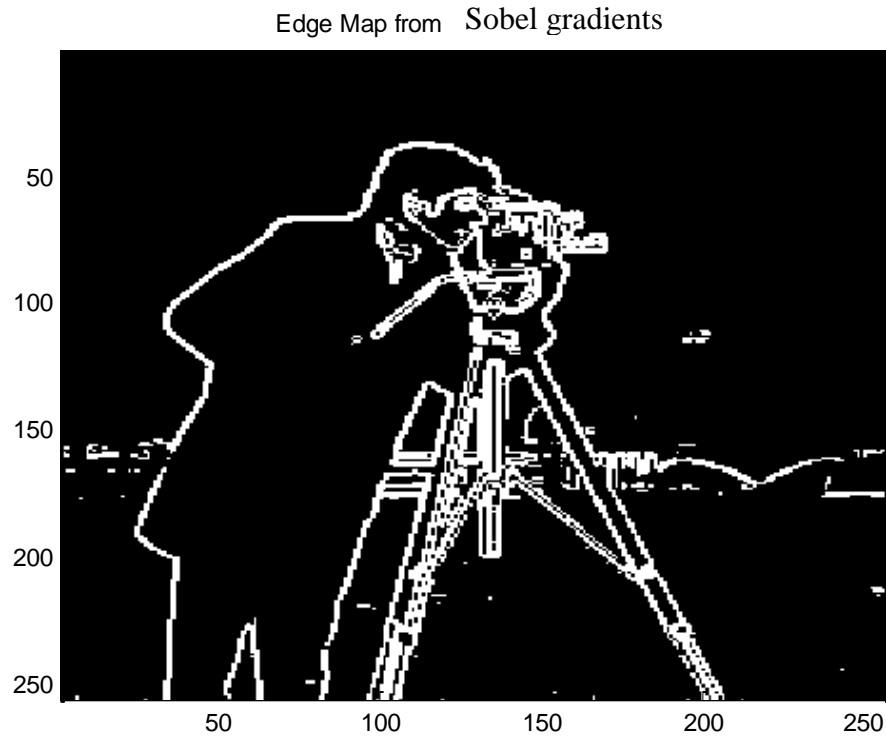


Figure 5 Comparison of our edge map and the Sobel segmentation from the original image

Similarly, our edge map demonstrates good results from OR theory to obtain our estimated gradients followed by thresholding. Clearly better gradients estimates produce better edge maps approaching the ground truth edge maps from the original image. Now that we can obtain good estimated gradients, we can continue to the next step of our SR algorithm which uses these new gradients for image diffusion, as explained in next chapter.

Chapter 4 Theory and Extensions of Anisotropic Diffusion

Diffusion is a typical PDE based solver that is used for image smoothing regarding the original image as the initial state and successive filtered versions as the result of temporal evolution. This whole evolution can be regarded as a so-called scale-space, an embedding of the original image into a simpler global representations of it [22]. In other words, we will use diffusion as a scale-space representation for extracting important information using the structure of the image. To model diffusion mathematically, it is common to express spatial and temporal variables using some physical phenomena. In general, physical ideas of diffusion are borrowed from phenomena such as propagation of heat and mass. Likewise, we can use other phenomena [22], but then again for our application in image processing we will only work with the heat and mass propagation equations. Here, the main idea of diffusion is to consider it as a physical process that equilibrates concentration differences without creating or destroying mass. This is obtained by using the equilibration property expressed by Fick's law:

$$j = -D \cdot \nabla u \quad (22)$$

Where the concentration gradient ∇u causes a flux j to compensate the signal for its gradient. The relation between ∇u and j is described by the diffusivity D , which is a positive definite symmetric matrix. Also, diffusion that only transports mass without destroying it or creating new mass is expressed by the continuity equation:

$$\partial_t u = -\operatorname{div} j \quad (23)$$

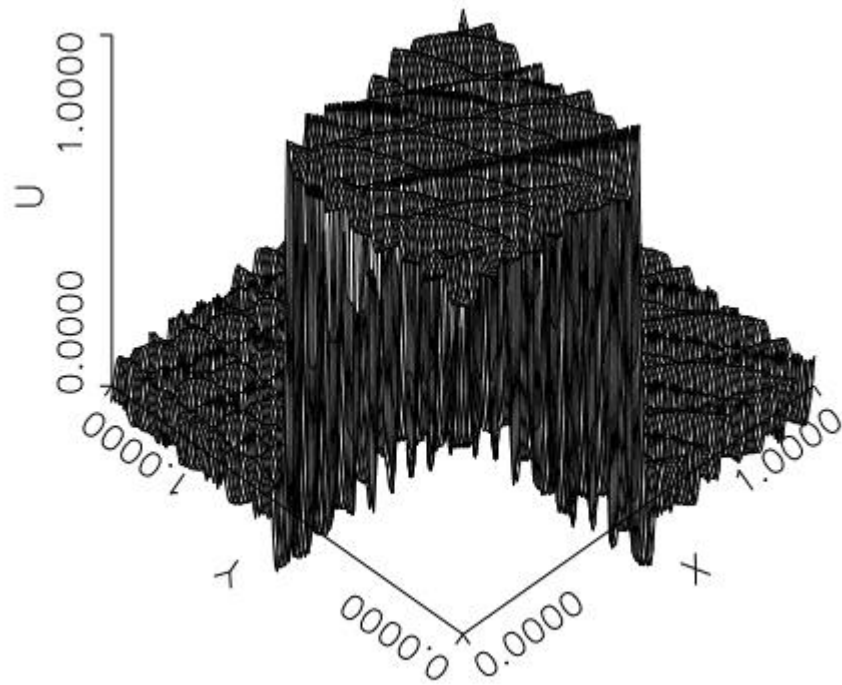
Plugging in Fick's law into the continuity equation we can obtain our diffusion equation:

$$\partial_t u = \text{div}(D \cdot \nabla u) \quad (24)$$

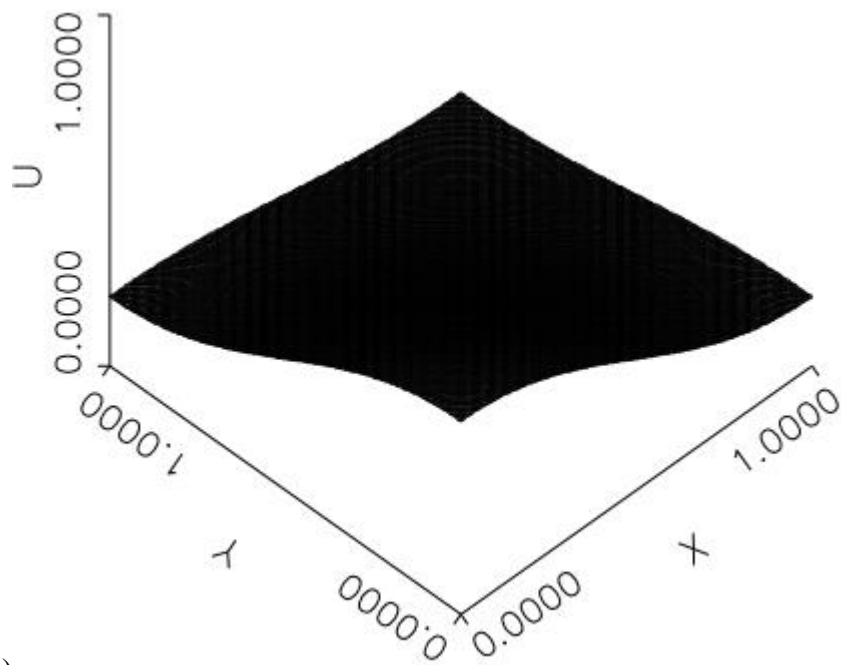
This equation is called the heat equation. In image processing applications, we can associate mass in this equation with the concentration of grey level values at a certain pixel location. Furthermore, we call this nonlinear diffusion when the diffusivity adapts to the image structure and thus apply a process that depends on local properties of the image itself. Otherwise, if diffusion does not depend on the structure of the image, it is called linear diffusion. In our case, we will work with nonlinear diffusion since our diffusion will depend on neighboring pixels and our estimated gradients of the image are used as well.

4.1 Anisotropic vs Isotropic Diffusion

The diffusion operation helps us to smooth areas that have noise or undesired values in portions of the image plane. In the literature, there exist two different methods of diffusion: Isotropic and Anisotropic. Isotropic diffusion diffuses our image in every direction without any restriction and it is characterized by a single diffusion scalar D . In image processing, isotropic diffusion can be seen as an averaging filter or more precisely as image blurring. Anisotropic diffusion also diffuses the image but the effect varies with the direction characterized by the diffusivity matrix D in equation (24). In the image processing area, the divergence operation is equal to the Laplacian operator in an image [67]. Common applications of this type of diffusion are denoising of highly degraded edges or enhancing coherent flow-like images. Here we will use anisotropic diffusion due to its ability to diffuse while preserving edge structure. Differences between isotropic and anisotropic diffusion can be seen in figure 6.



a)



b)

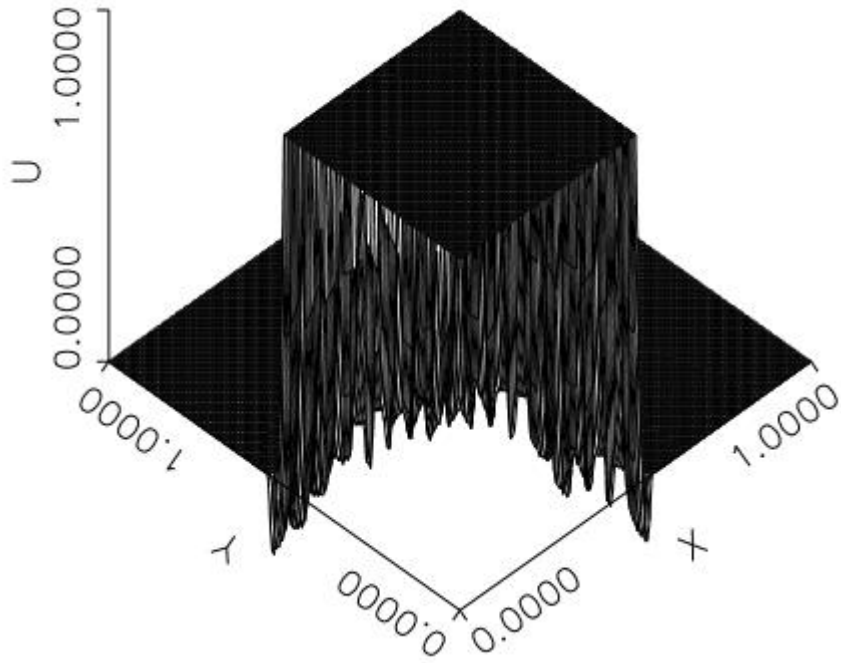


Figure 6 Edges are preserved by using anisotropic diffusion (images courtesy of Dr. Granville Sewell), a) Original data, b) Isotropic diffusion, c) Anisotropic diffusion from Heat Equation.

4.2 Perona-Malik Anisotropic Diffusion

The first and most common method for non-linear anisotropic diffusion is the Perona and Malik (PM) approach. This method smooths an image and simultaneously enhances important features such as edges by incorporating derivatives of the evolving image in the diffusivity. Instead of using a common diffusivity, they apply a function that uses the first difference in a 4-neighborhood of the image pixel to represent the anisotropic part of the diffusion. This allows to avoid mixtures of different regions and help us to enhance edge areas in an image. The new diffusion equation is

$$\partial_t u = \text{div}(g(|\nabla u|^2) \nabla u) \quad (25)$$

where

$$g(s^2) = \frac{1}{1+s^2/\lambda^2} \quad (\lambda > 0) \quad (26)$$

and λ is a contrast parameter, low contrast means forward diffusion while high contrast means backward diffusion. Note that PM is not really an anisotropic algorithm since its diffusivity is a scalar (this makes the algorithm isotropic), however, the community called it anisotropic since this scalar value forces the algorithm to work as anisotropic diffusion by preserving edges. The two conditions that this diffusivity function requires are: maximum value should be a 1 and minimum should be 0, and it must be monotonically decreasing. In figure 7 we can also appreciate their difference by comparing an image with isotropic and PM anisotropic diffusion.

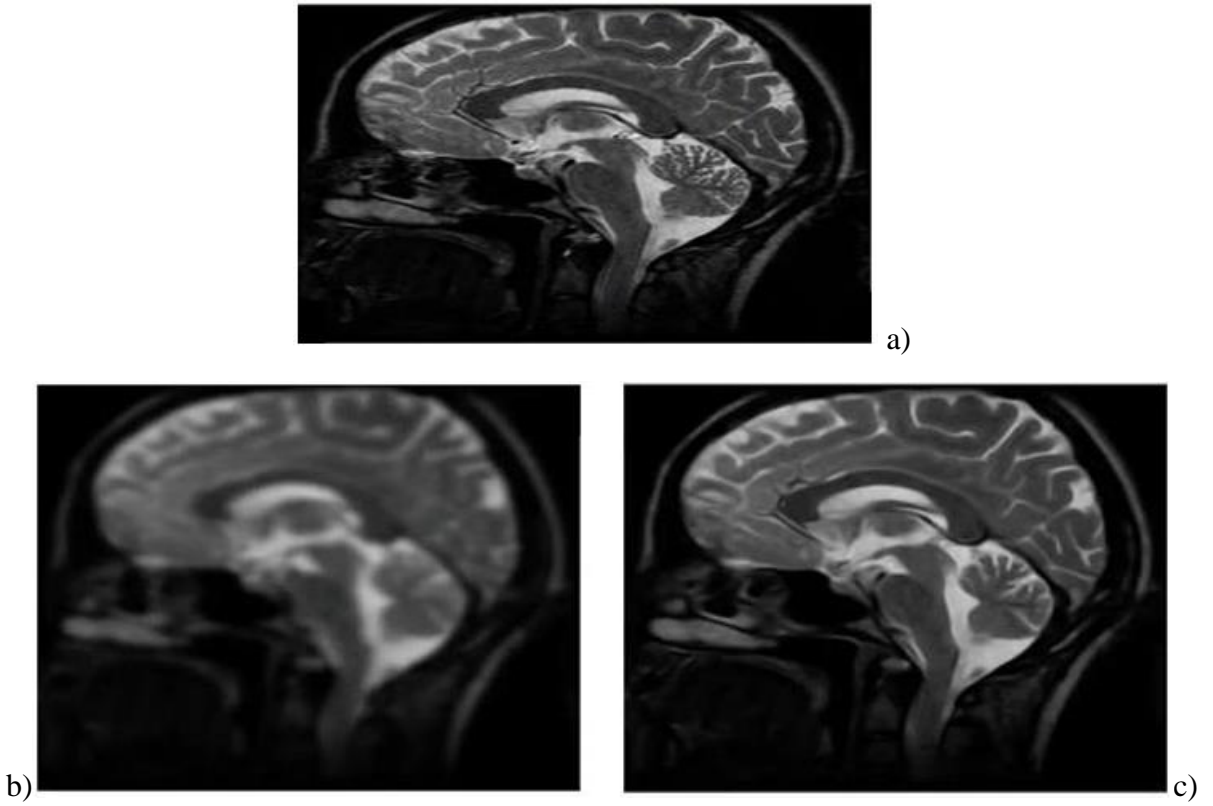


Figure 7 Difference between Isotropic and Anisotropic Diffusion in one image, a) Original image, b) Isotropic diffused image, c) Anisotropic diffused image [68]

4.3 Extensions of Perona-Malik

Recent extensions of the PM method applied to SR is the method used by Maiseli in [20], [21] developed in [53], an adaptive PM model. In [20], he combines the Papoulis-Gerchberg (PG) multi-frame SR method with a modified PM anisotropic diffusion using a new diffusivity function that improves results using a gradient magnitude from the input LR reference image, which is

$$g(s) = \frac{s}{1 + \left(\frac{s}{K}\right)^{p(x)}} \quad (28)$$

where

$$p(|\nabla G_\sigma * u|) = 2 - \frac{2}{1 + k|G_\sigma * u|^2} \quad (29)$$

And G_σ is the Gaussian Kernel.

Results in [20] show visual improvements in the PG method in edge areas which is also our main goal.



Figure 8 Maiseli Super-Resolution algorithm using PG and a modified PM.

Newer methods use this basic idea from PM and proposes different diffusivity functions with the same properties also using gradients [53]. This motivates us to take advantage of the improvements from the adaptive Perona-Malik diffusivity function to integrate it with our new OR based estimated gradient from chapter 3 and propose the use of a Diffusion step inside of our SR-OR algorithm, see chapter 5.

Chapter 5 Super-Resolution with Optimal Recovery review and improvements including Anisotropic Diffusion

5.1 Improved spatially adaptive SR-OR method

The most recent work on Optimal Recovery based Multi-Frame Super-Resolution is described in [8]. This method operates on blocks (patches) of the LR images, using the OR framework to reconstruct blocks of the HR image. Also, the authors use 5 different block classes, depending on block variances to determine which of the 5 fixed bandwidth a_i will be used by the OR process in obtaining an HR. The mapping is defined by the equation (30) that includes a variance to bandwidth mapping parameter P . Our approach is to revive the software that implements this method and use it as a base for improvements, modifications and additions. This is the fastest way to produce new results on extensions of such an approach. For future goals in video processing and video metrology, it is clearly very important to make algorithm formulation and software implementation very computationally efficient for faster processing.

Analyzing the MATLAB code for the SR method in [8] we found potential improvements in: remove anti-aliasing filter to generate LR images so we can recover HR frequencies, implement a new shift registration for LR images, improve selection of parameters and block classes according to the input image, bypass the use of Generalized Cross-Validation (GCV), implement a variable block size choice, allow different input image sizes, apply code debugging techniques to make the code more efficient in time, and solve boundaries problems. Furthermore, we will extend this approach by obtaining estimated gradients as well as generate edge maps by using OR theory, and integrate an anisotropic diffusion method as post processing or block post processing for edge enhancement.

We believe that there is a way to determine the best mapping parameter value P where we can acquire the best possible result, this lead us to propose a way to select the parameter in a better way than before. Also, we propose to bypass the use of GCV in selecting the regularization parameter λ (lambda) in the reconstruction of each patch of the HR image. Using a fixed lambda value we can check for better results in terms of MSE and increased speed in the execution of the code, this will be a Globally optimal regularization parameter selection. Finally, we also improved the bandwidth selection for each class of LR blocks by using a multi-threshold method combined with K-means results for any input image. Multi-threshold values will be used to limit our classes while K-means values will be used to select the representative variance for each class. Besides, we will compare the use of adaptive bandwidth with multiple block classes against the use of a fixed bandwidth value for each block class. This will help us to decide whether it is better to use a fixed, small number of classes or not. For comparison, we will use the Mean Squared Error (MSE) as used in [8] and we will also include the Structural SIMilarity (SSIM) index [63, 64] to make reconstruction quality assessments in addition to visual comparisons. These two metrics are defined as

$$\text{MSE} = \frac{1}{MN} \sum_{y=1}^N \sum_{x=1}^M (f(x, y) - \hat{f}(x, y))^2$$

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where C_1 , and C_2 , are constants to use for stability.

5.1.1 Shift Registration

First of all, shift registration is a main problem for several state of the art method. If we include a good shift registration in our SR algorithm, chances to improve final results are high.

Otherwise, our code may not compete against other methods. One of the improvements from previous work was to avoid quantization of shift parameters that were inevitable in a previous third-party MATLAB function “shift”. Changing to a different function, the official MATLAB function “imtransform”. This way, we can perform arbitrary spatial shifts in the generation of the LR images to have a better match to what the OR method uses. Even if the quantized shifts appear to give us a small difference, if we account for small difference on thousands of pixels, the final error between the reconstructed and the original image would be significant. Also, we have to consider that we are reducing the resolution of our original image by 2 to generate our LR images and then increasing the resolution again by 2 in the generation of the recovered HR image, so, our LR shifts parameters are doubled when applied in the HR spatial domain.. Similar to previous work [8], to produce our LR images, we first interpolate our original image by a factor of 2, and then apply four times larger shift values in the x and y directions (since now the interpolated image would be 4 times bigger than our LR’s) using the “imtransform” function.. Finally, we downsample the shifted images by 4 in the x and y directions to generate our LR images with aliasing. In our examples, shifts for LR images are in the x-direction: 0, 0.12, 0.6, 1 and in the y-direction: 0, 0.2, 0.28, 0.8. We will not include rotation in our algorithm. Figure 9 illustrates the steps required to generate our LR images. Figure 10 shows the 4 LR images for the chosen shifts.

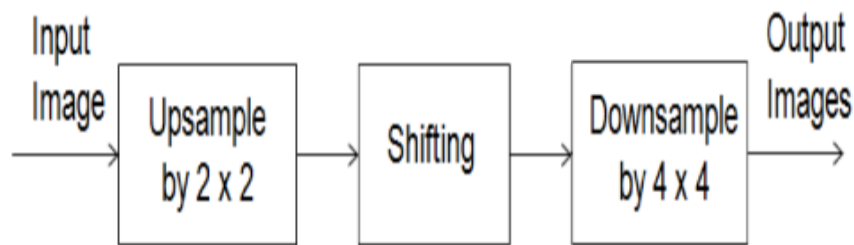
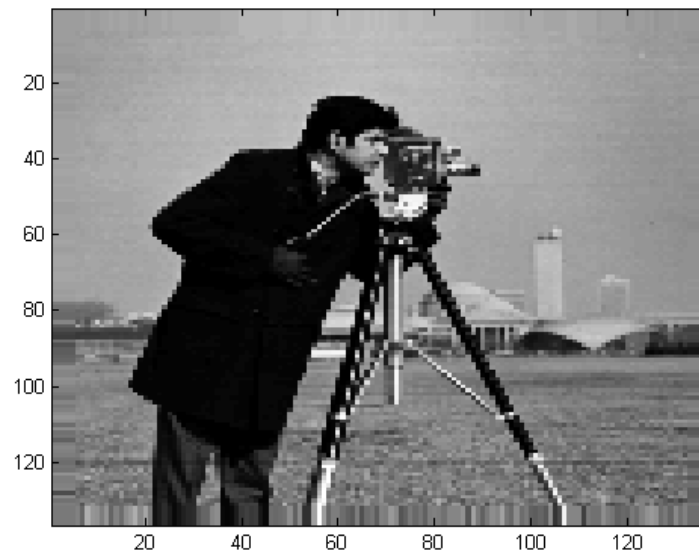


Figure 9 Block diagram for the generation of the LR images

LR1(Reference)



LR2



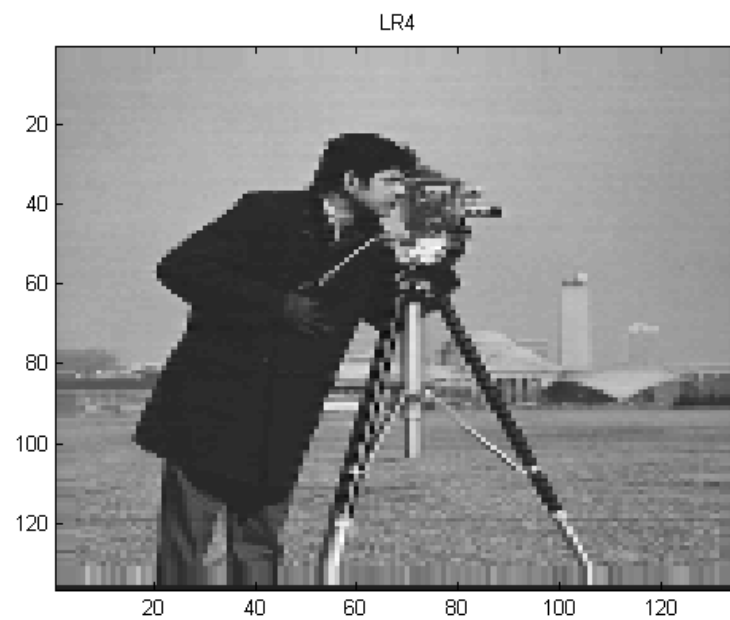
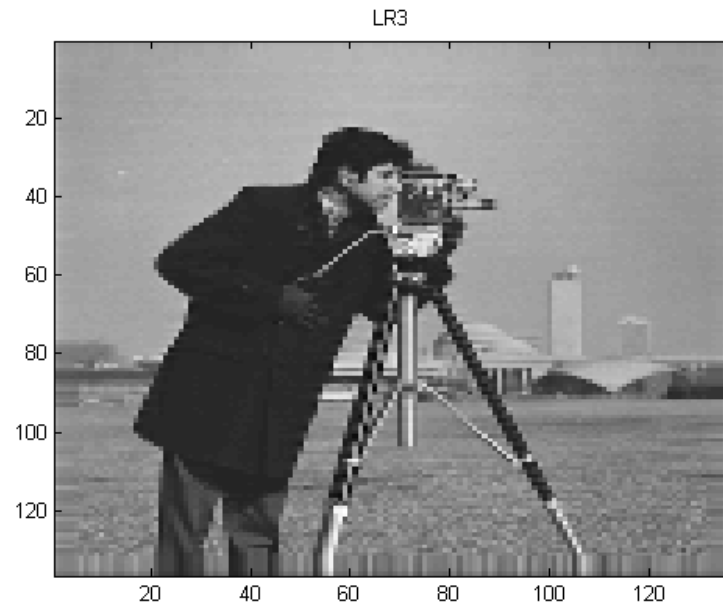
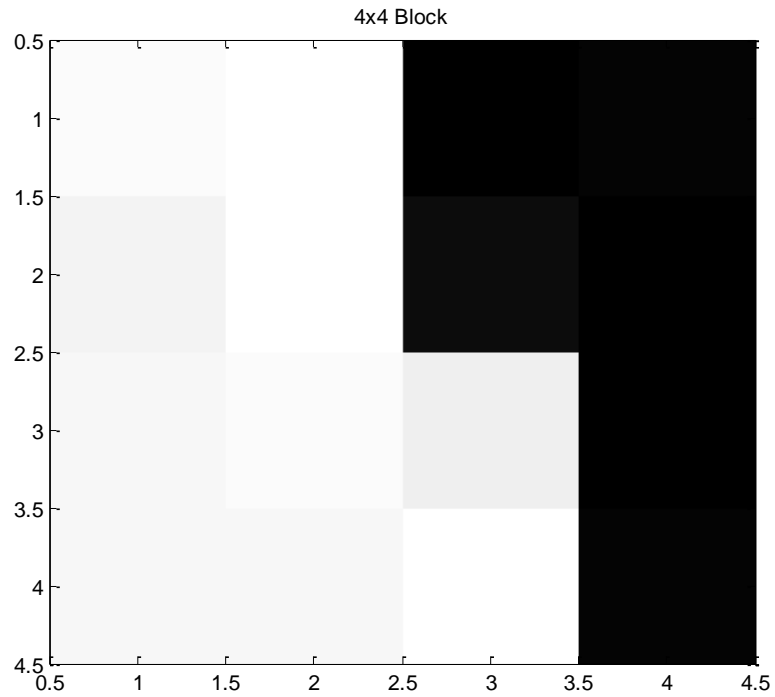


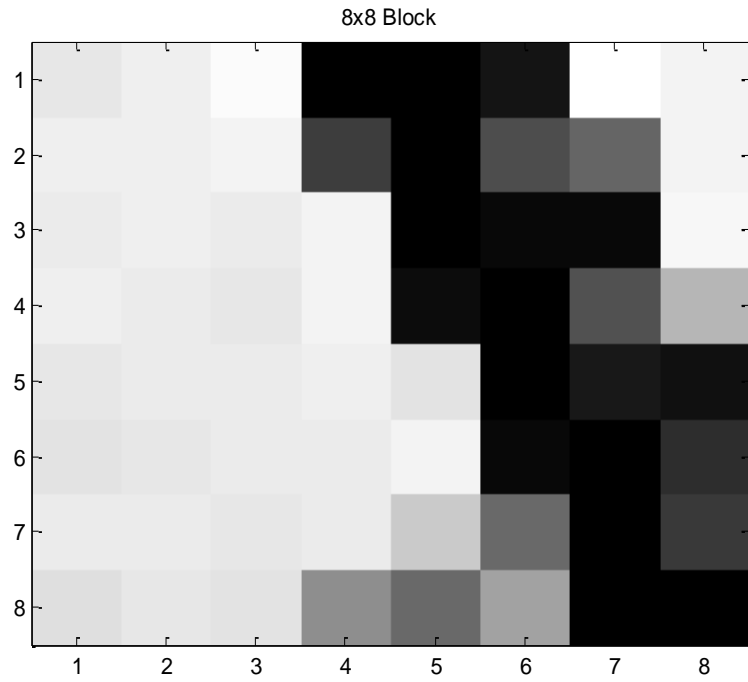
Figure 10 LR images after the shift process. We note the use of row or column replication in the LR images resulting from the use of row or column replication on the original image.

5.1.2 Flexible block and image sizes

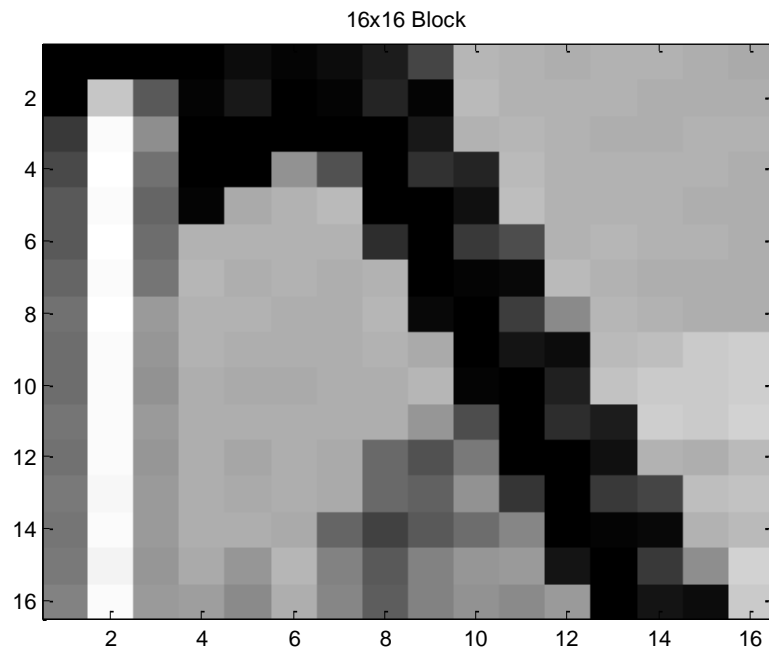
Next, we improved our code by implementing a variable that allows us to use different block sizes in selecting the input LR image data to reconstruct each patch of the HR image (multiple of 2 but preferably powers of 2) in addition to modifying all parameters needed according to the input image size. Figure 11 shows different block sizes from the same part of the cameraman image.



a)



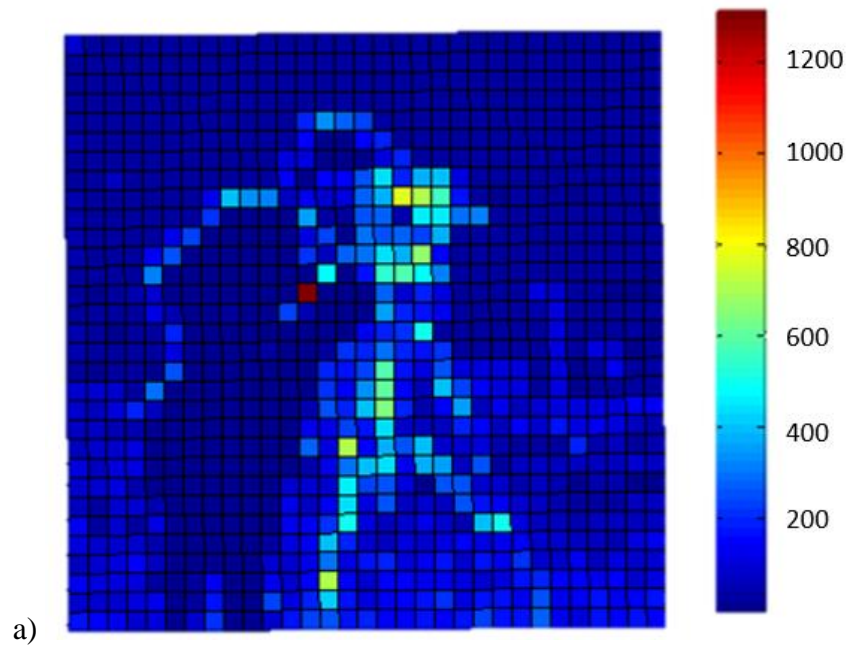
b)



c)

Figure 11 Different size blocks from our LR reference image extracted from the same area, a) 4x4 block size, b) 8x8 block size, c) 16x16 block size.

The use of different block sizes makes us realize the tradeoff between using a bigger block size vs using a small one. The advantage of using a small block size against bigger block size are: the number of data we use for one computation decreases and number of regions in one block also decreases which causes a reduction in the number of pixels from mixed spatial regions. Disadvantages of using small blocks are: the total number of blocks in one image will increase forcing the code to require more block operations for one image, also using less information may result in obtaining undesired intensity values and discontinuities between one block and another block next to it. Another benefit of being able to select different block sizes is that we can locate where our final error is concentrated. Choosing different block sizes in our No-Overlap OR approach, we can display a block-based MSE map comparing our resulting image with our original image, see figure 12.



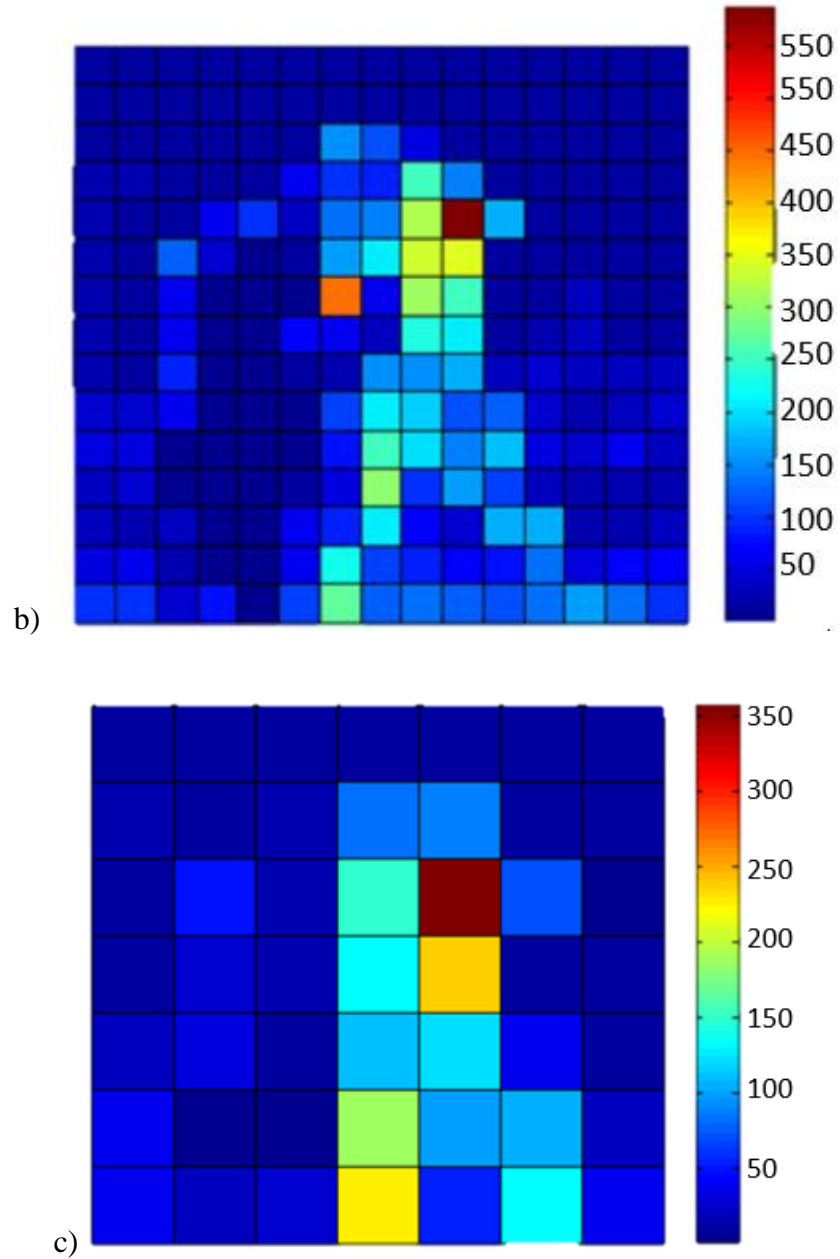


Figure 12 Error map from no-overlap approach using different block sizes, a) 4x4 block sizes, b) 8x8 block sizes, c) 16x16 block sizes.

As seen in this figure, edge areas are highly correlated with large MSE. This is an extra motivation to work with diffusion to improve the reconstruction of edge areas as discussed in chapter 4.

In addition to allowing the code to select different block sizes, we also make the MATLAB code more flexible by allowing any input image size. Previously we could only use images of sizes 256x256.

5.1.3 Code Debugging

A sophisticated and complex SR method [14] was recently evaluated and found to require a high computational effort to reproduce results found in the literature. For a black and white video of size 296x352x30 (30 frames of 296 high and 352 width) the method executed each frame in approximately 10 minutes with no option to parallelize due to dependency from one block to another. For this reason, not only reconstruction error measurements should be considered but also code performance should be considered in image and video processing.

There are several ways to optimize code in MATLAB [15], variable pre-allocation, use of MATLAB Executables (MEX) files, reduce function calls, and parallel computing. Loops in code are inefficient in program optimization due the fact that if a variable increase its size every iteration, this will give us an allocation and deallocation problem. One way to solve this problem is by doing a pre-allocation of variables by initializing all variables to their final size setting them to zero so that we can overwrite them as the code is executed. Then we have MEX files that contain C++ code (a compiled language), these files allow us to be much faster than normal MATLAB code. In practice, it is desirable to use MEX files for functions that are executed several times due its time of execution could be reduced up to 1000 times. Finally, only for codes with independent blocks of data inside loops, we can implement a parallel computation code in MATLAB. Parallel operations cannot access values from previous blocks but they can help to improve our code speed efficiency. Other recommendations that can help

us to improve speed are minimize I/O, avoid calling functions several times, and avoid printing values or displaying results with graphs or images. Aside from previous recommendations for code improvement, we reduced our run time by precomputing our main matrices S and H for each variance class instead of computing them for each block in the image. Furthermore, by limiting the number of block classes using K-means clustering, we reduced the number of computations for each class. For our algorithm we leave the use of MEX files and parallel computing as future work. However, using other recommendations we were able to reduce previous execution time from three minutes to run one SR image example to 1 or 2 seconds per example.

5.1.4 Limited bandwidth constant P

The variance to bandwidth mapping parameter P has been used in previous work [8] as a fixed constant value with minor justification. Furthermore, different images may require a different P value and thus, further analysis was done. By testing several values for P and comparing with the original image, we can check if P has a huge impact or not. In order to test the sensitivity of MSE, we chose 101 different P values from 0.03 to 0.13 to find the best choice. This gives us the possibility for improvements by carefully selecting the P parameter. In figure 13 we plot the resulting MSE of our reconstructed image vs the P value selected. We found a minimum $MSE=66.2918$ at $P=0.083$ which give us better results than the previously reported $MSE=79.42$, see figure 14 and the zoomed version in figure 15. These results show us that selecting the correct P value for each image has an impact on our results so we need to consider a formulation to determine a P value that achieves the best results for any input image.

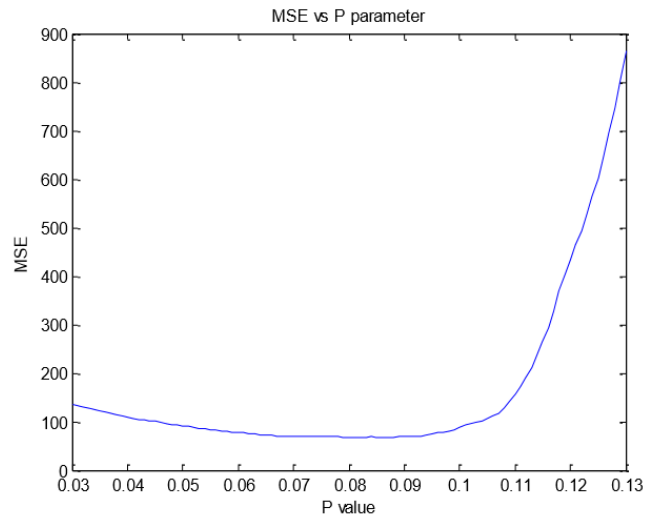


Figure 13 MSE result by varying the P parameter

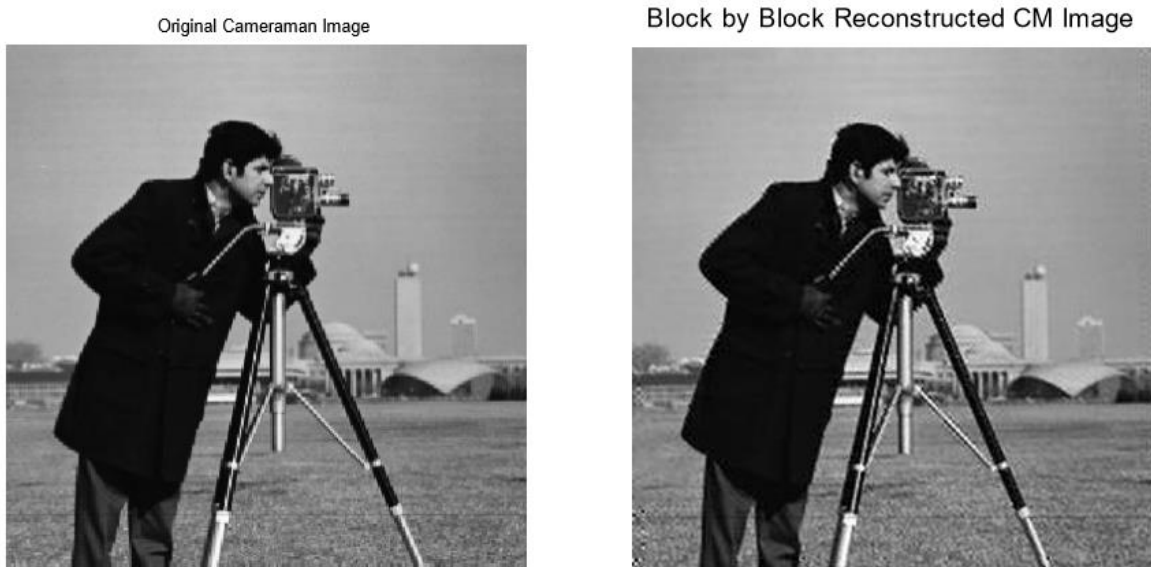


Figure 14 Comparison between resulting image using $P=0.083$ and original image

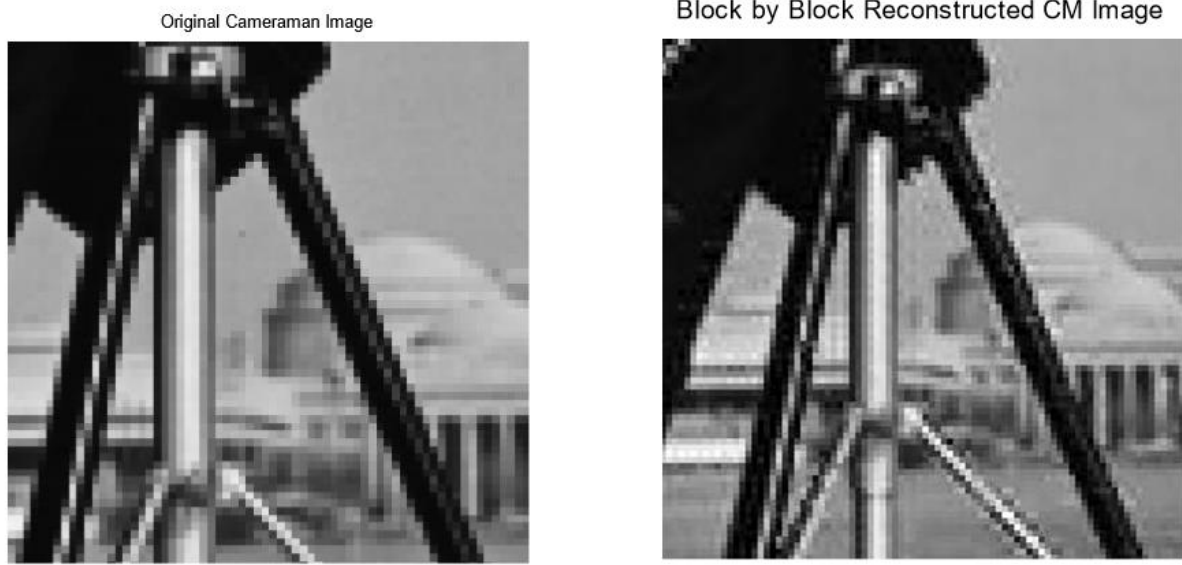


Figure 15 zoomed version for better comparison between resultant image and original image

To start formulating our problem for a justified selection of P , as seen in equation (30), the P parameter is being used to select our bandwidth used in OR for each block class. As stated in [17] our limits for our bandwidth are from $-\pi$ to π assuming a sampling period of $T_x = T_y = 1$ in the HR image, therefore we can limit our max value of (30) to π by calculating a likely highest value possible for our variance in one block.. To calculate a worst case block scenario for our formula when we have the highest variance possible, we can assume that one block can have 100% salt and pepper noise with 50% salt (max value of one pixel = 255) and 50% pepper (min value of one pixel = 0). For this block, we use the variance equation:

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N} \quad (29)$$

Where mean value μ is 127.5 since half values in the block are 255 and the other half are 0. Therefore, in our worst case block, we obtained a variance σ^2 equal to $127.5^2 = 16256$ and its $\log(\sigma^2)$ is equal to 9.6962. Note that in our whole algorithm we use log base e , it can be

changed to log base 10 but we keep this choice for compatibility with previous work. Finally, we can use this value as the highest log of the variance possible in one block, and since highest bandwidth possible is π , then $a_{max} = \pi * P * \log(16256) = \pi$, and $P = 1/9.6962 = 0.1031$ is the value to be chosen for any input image.

5.1.5 Global Optimization regularization parameter

As mention earlier, we propose to bypass GCV regularization for this algorithm to increase the efficiency of the code for future video processing applications, therefore, we modified the algorithm to use only one regularization parameter lambda (λ) for all blocks. Similarly as previous experiments for P parameter, now we reduce our problem to determine our best constant lambda that achieve us the minimum MSE results. Here we also considered a possible correlation with P parameter, so as we run previous experiment, we also give a set of experiment values where we modified our regularization constant. In figure 16, we note that lambda value must be a small value to make the algorithm work appropriate, so for our experiment we set 500 different lambdas with a minimum lambda $\lambda = 0.00001$, and a maximum $\lambda = 0.1$, also we use 51 values of P from 0.03 to 0.13 to find the best values for our two parameters at the same time and find an improvement in either speed and/or final MSE in our results, see figure 19. Furthermore, to check if there is a correlation between regularization parameter and variances, we display in figure 17 block variances plot to compare with figure 17 where we still use GCV and also a scattergram in figure 18 between log of lambdas and log of variances.

In figures 16-18, the LR image data is taken as 4x4 blocks from all 4 128x128 LR images. With an overlap of 50%, the boundary extension by replication, the total and number of blocks of input data are $4 \times 32 \times 32 = 4096$.

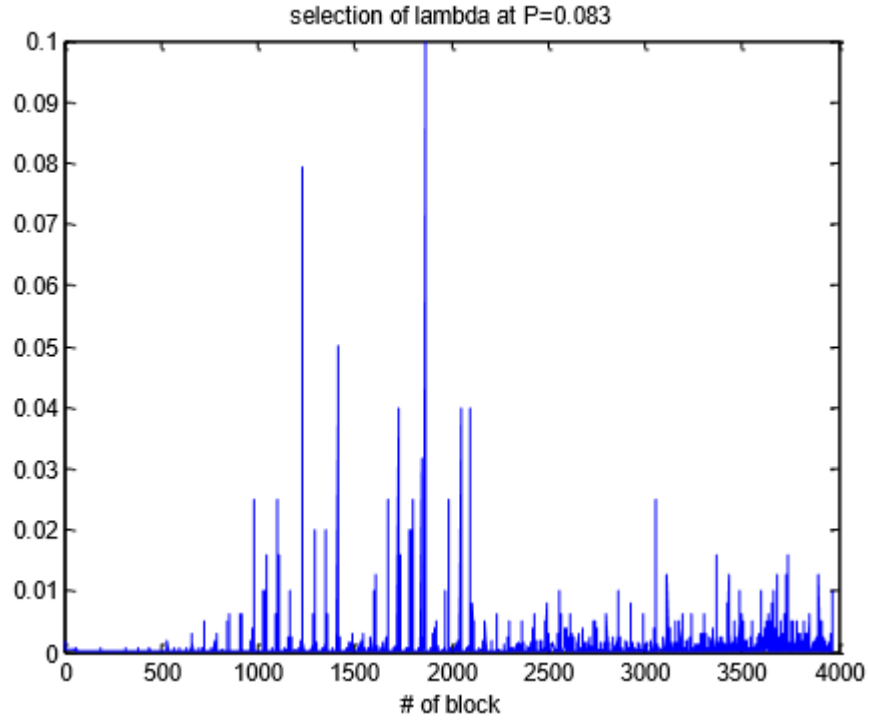


Figure 16: λ values generated by the GCV with $P=0.083$ using 4x4 blocks of the 4 LR images with 50% overlap.

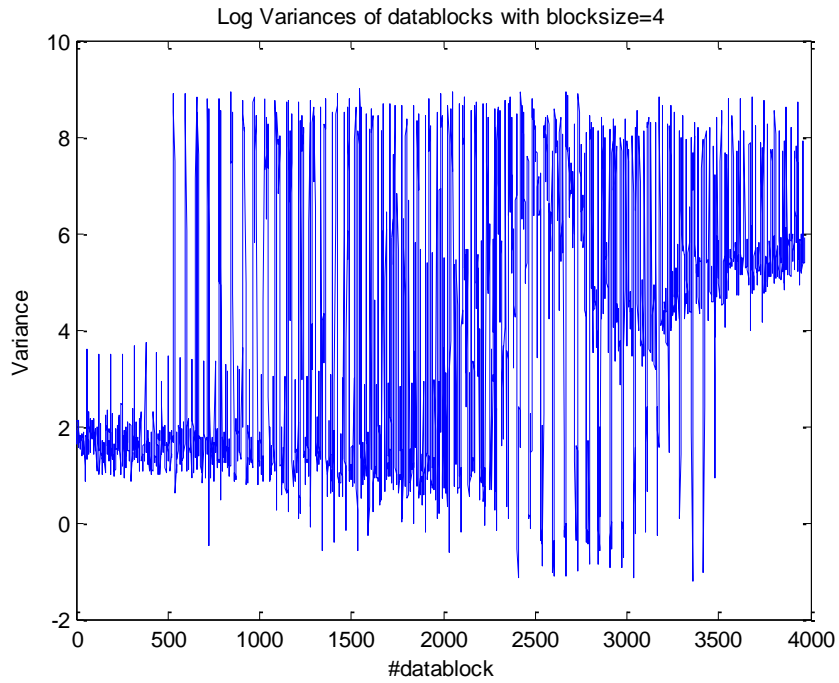


Figure 17 log of the Variances of the input data blocks of the cameraman image

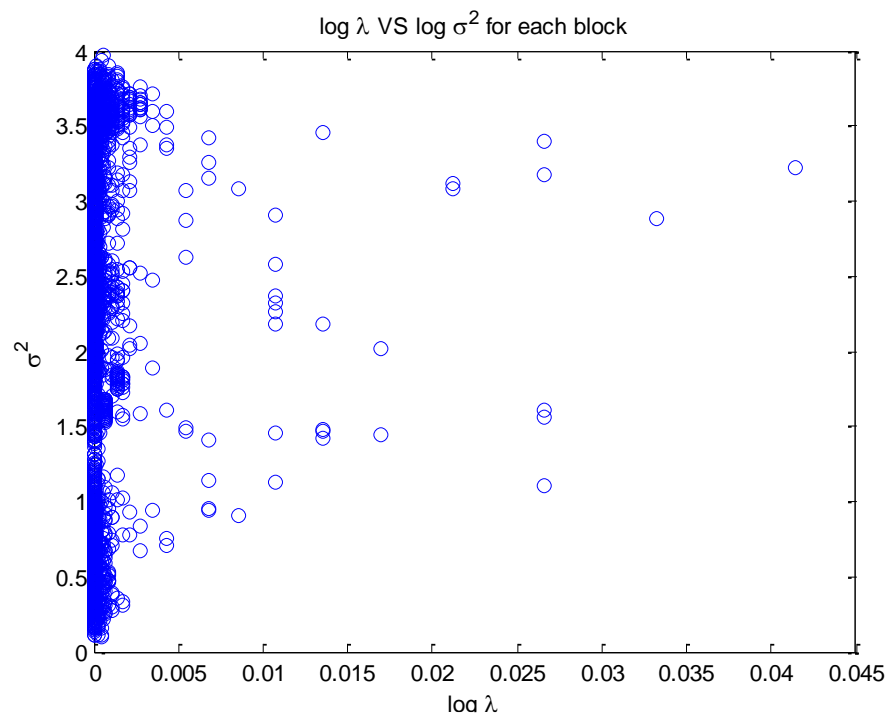


Figure 18 Scattergram from variances and log of the regularization parameter to look for any correlation between them.

As we see in figure 18 lambda is always low independent from our variance for each block.

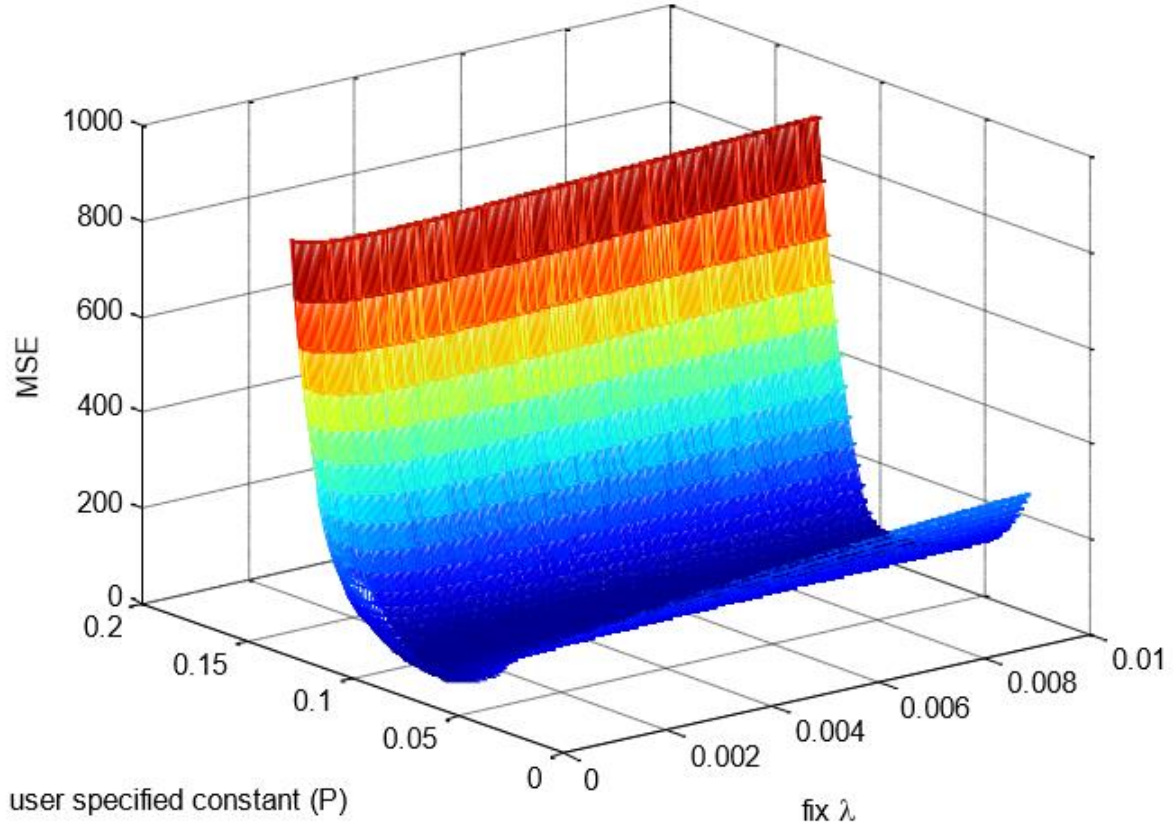


Figure 19 Relationship between λ and P in terms of MSE results of the recovered image compared to the ground truth original.

The next experiment involved varying both P and λ . We found that the best result was at $P=0.09$ and $\lambda=0.0028$ with a $MSE=48.9360$ using the previous students' code. This MSE is much better than all previous results. From these parameter, we can display our new resultant image in figure 8. From these results in figure 19, we note that lambda value does not have a huge impact on our final MSE results. Using a fixed lambda reduces our run time of the SR-OR algorithm, therefore we proposed to bypass the GCV regularization parameter computation method and use a fixed small lambda (the one we obtained as the best in figure 19, $\lambda=0.0028$).

This will help us to obtain similar results as before under approximately 1 second run instead 1 minute or more when we use the GCV method.

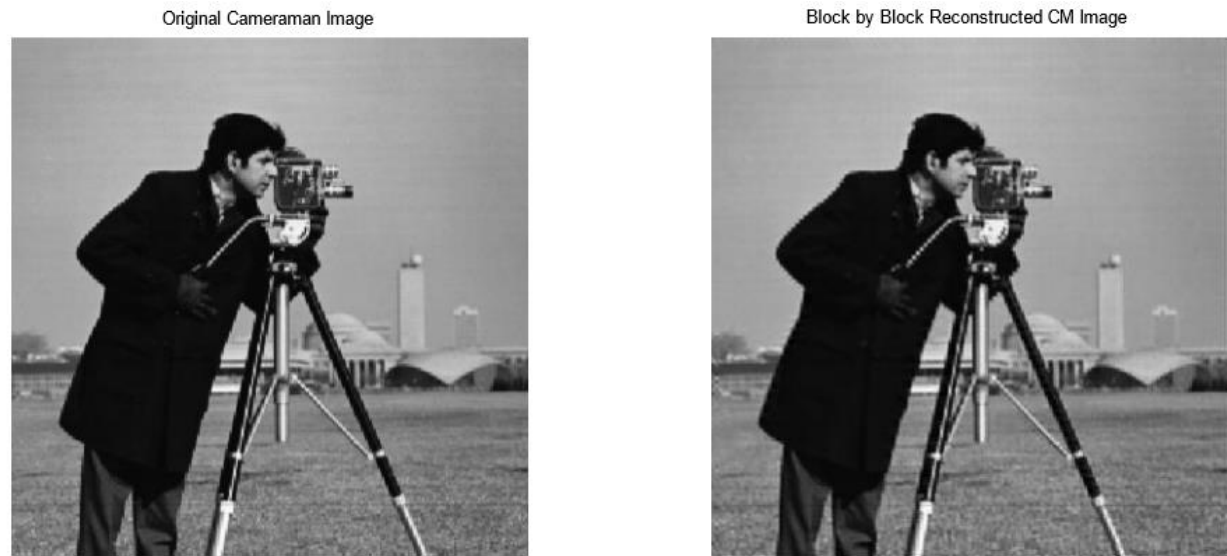


Figure 20 Comparison between resultant image and original image looks almost identically

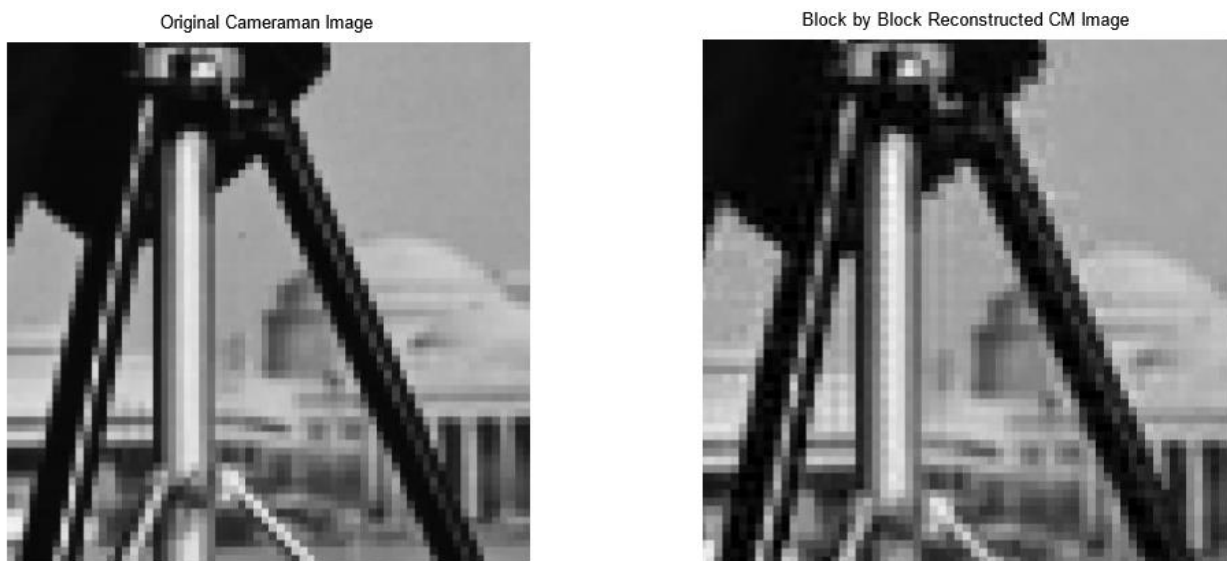


Figure 21 zoomed version for comparison between original and resultant image.

In Fig 21 we can see the improvement from Fig 15, we can note that more areas were improved in this new case but then again edges are still a problem.

5.1.6 Multi-threshold vs function for bandwidth selection

In order to select our block classes in any input image, we will continue with the previous approach of using LR input block or patch variances as references for our choice of bandwidths to reconstruct each HR block using the OR framework [8]. To obtain the bandwidth, we have two different options: determine the variance for each block and apply a multi-threshold method to separate blocks into different classes in the image or apply a general function to get a unique bandwidth for each block. For the multi-threshold method, we will choose 5 different classes to separate our blocks where classes can vary from smooth blocks to high variance blocks including edge blocks. By doing this, we just need to determine five different bandwidths using (30) so we can reduce our computational effort in our algorithm. Also, because variances are distributed over a large dynamic range, we will express them using log scale. Figure 22 shows how we implement the thresholding of the logarithm of the variance using an optimal/automatic multi-thresholding algorithm used in Image Processing, a generalized Otsu method [16]. On the left side of Figure 22 we have low variance blocks which are related to smooth areas, and at the right side we have high variance blocks that corresponds to high frequency texture or edges areas.

For the second approach, we could use a function to determine a unique bandwidth for each block as stated earlier.

$$a = \pi * P * \log (\sigma_b^2). \quad (30)$$

where P is our mapping parameter and σ_b^2 is the variance of the block from the LRs. By using this method we can obtain the exact solution for our problem, however, it requires a high computation effort since we need to calculate a different S matrix (and its inverse) for each block of the HR image but final results show no huge difference. For this reason, the discrete choice of classes and bandwidths is preferred.

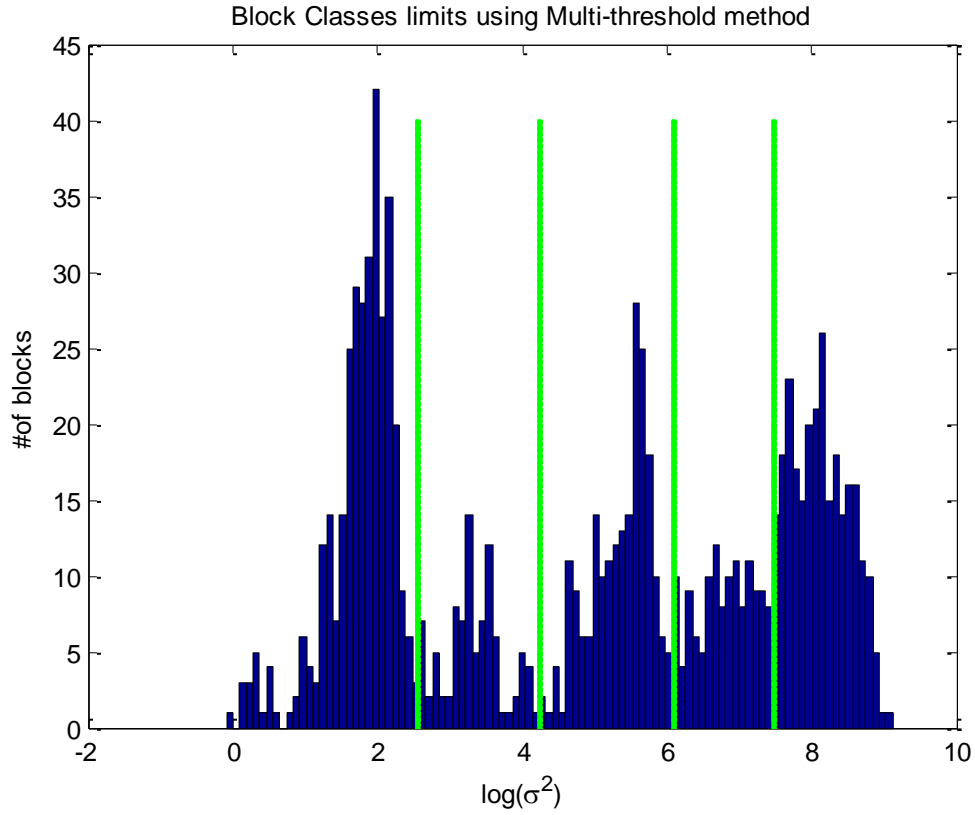


Figure 22 Selection of the five main classes in an input image (Cameraman) by a multi-threshold method. Note that we need to select 4 different values for our multi-threshold to obtain 5 different classes in our result

With this in mind, even though multi-threshold automates our block class selection, we still have to determine a representative variance to use for selecting each bandwidth for that class. K-means clustering will help us with this problem.

5.1.7 K-means clustering for class representation

Similar to the multi-threshold approach, K-means clustering will help us to determine our bandwidths for each block class [16]. The only difference between them is that K-means approach will give us the representative variance for each block class while the multi-threshold approach gave us the block variance ranges for each class as seen in figure 23, here we use 8x8 block size. This will help us to remove the fixed choices of variances for each class from previous work [8] and allow us to automatically get our 5 classes for any input image. Given the number of clusters “K” (for our work we fix it to 5 classes), at each iteration, the K-means algorithm will classify the given data into K different classes by obtaining the mean of the data that are closest to the Kth cluster center until the difference from one iteration to another is less than a tolerance value selected. After that, we obtain the center for each class that we will use to be our representative bandwidth for each block class. Figure 24 shows a complete block diagram for our bandwidth selection.

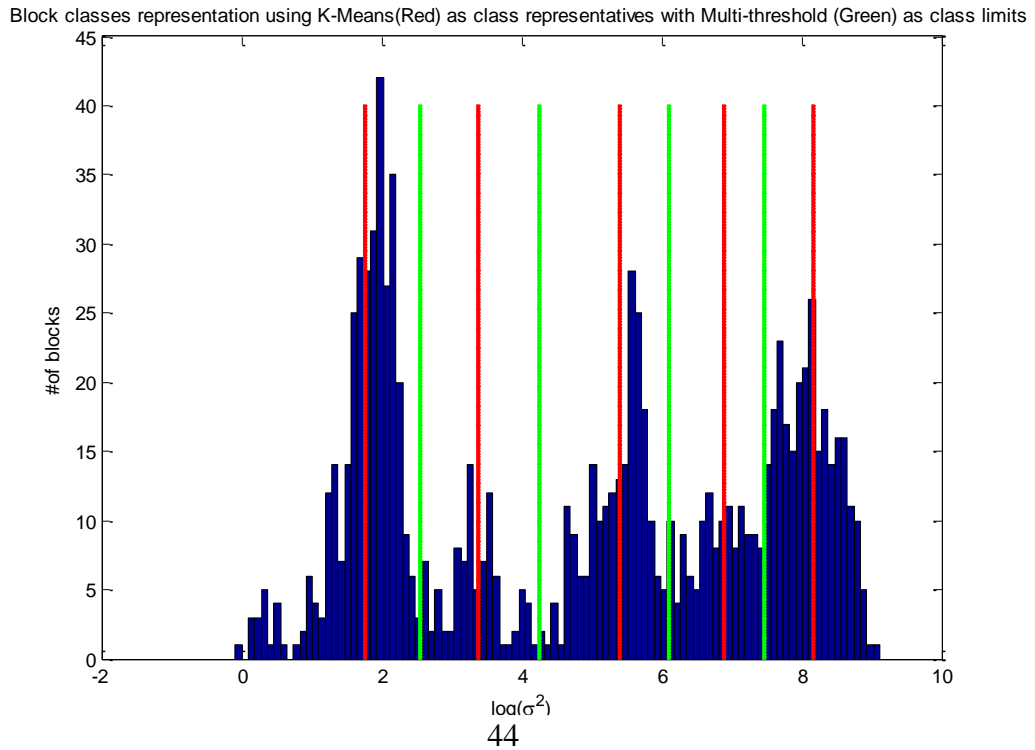


Figure 23 K-Means values (Red) are the representative variances for each block class while Multi-threshold values (Green) representing the class boundaries for each block class.

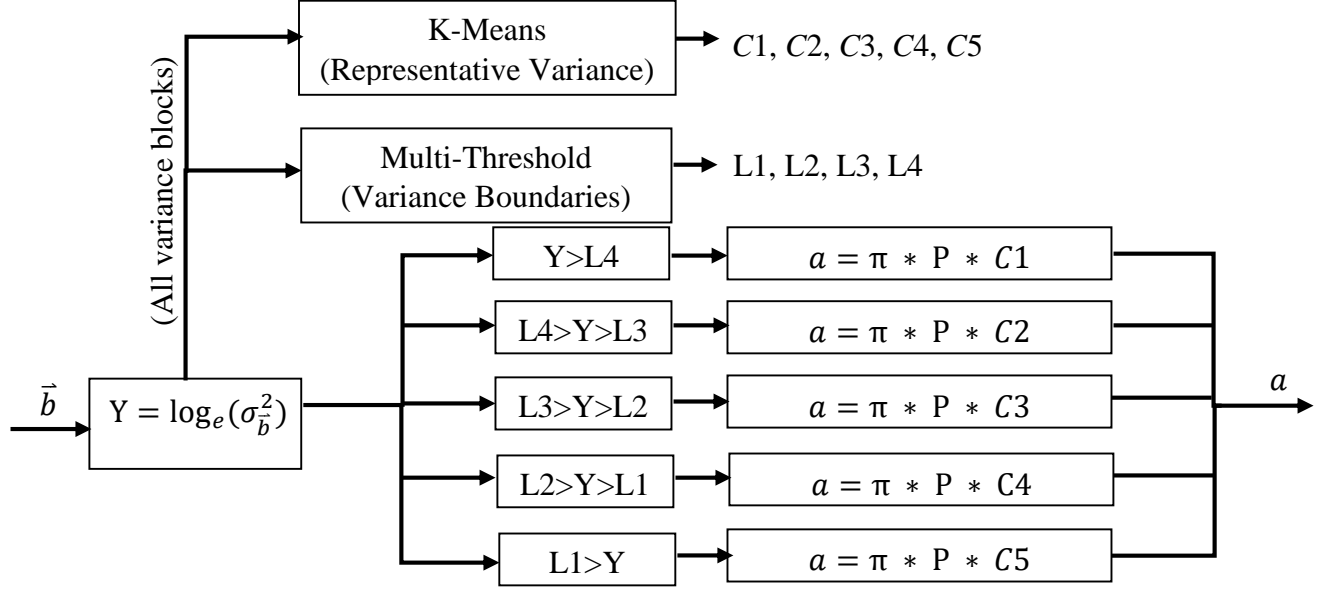


Figure 24 Bandwidth selection using multi-threshold and K-means method in block variances.

Using the results from figure 23 and $P=0.1031$, we have $C1= 1.7522$, $C2= 3.3678$, $C3= 5.3923$, $C4=6.8890$, $C5= 8.1521$, $L1= 2.5457$, $L2= 4.2432$, $L3= 6.0852$, $L4= 7.4577$.

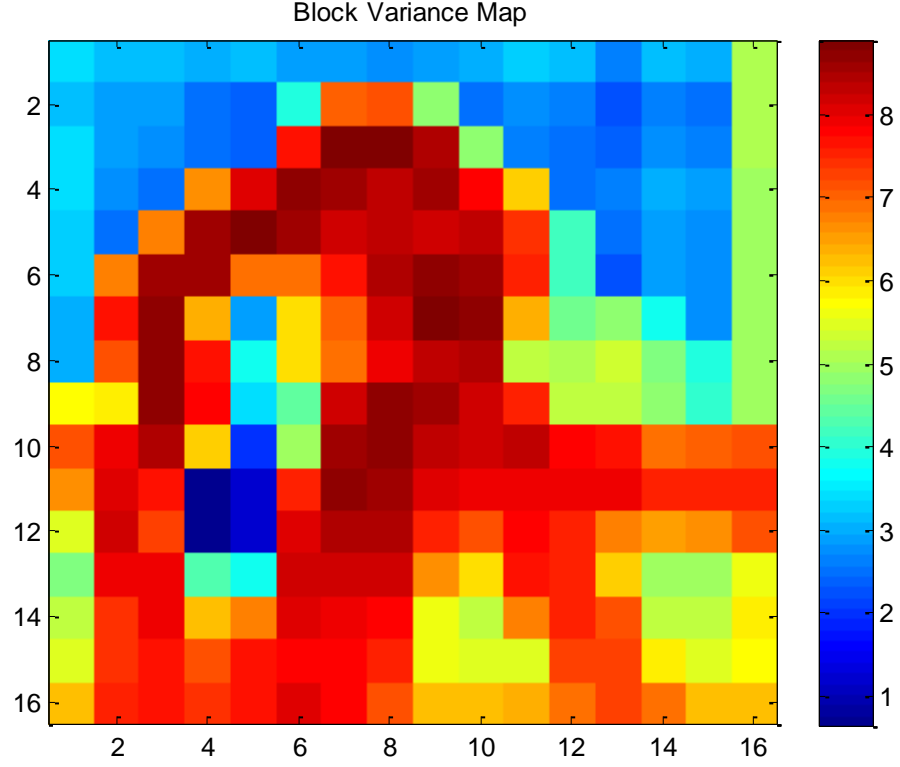
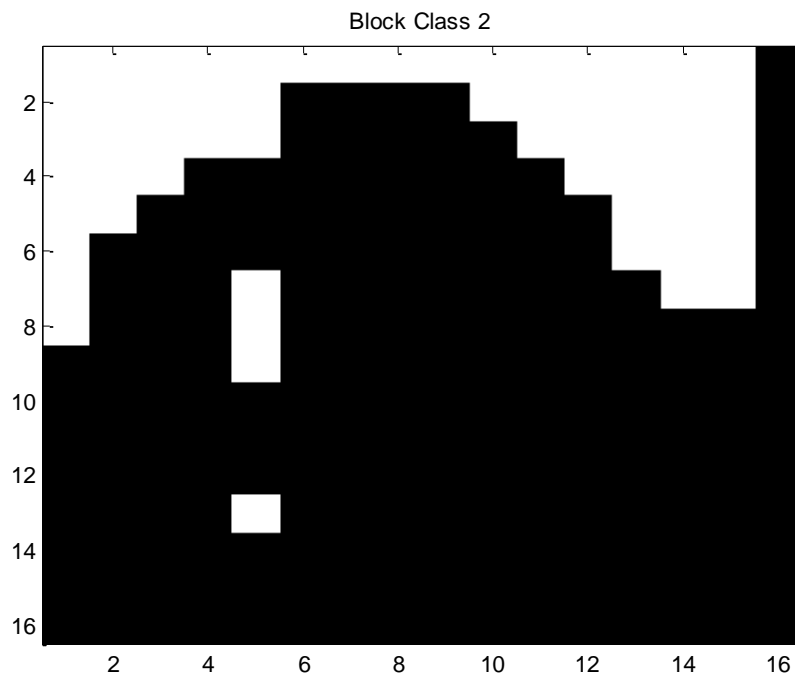
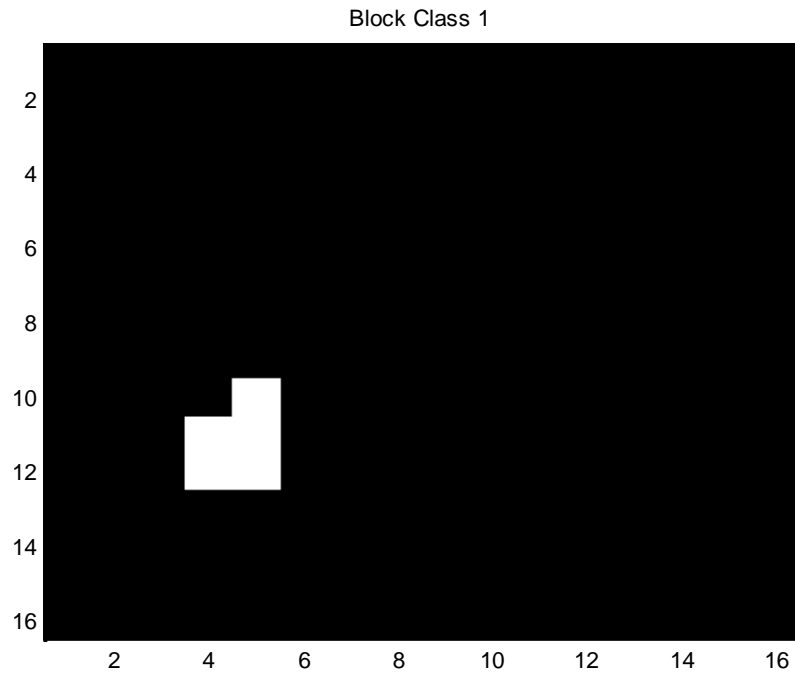
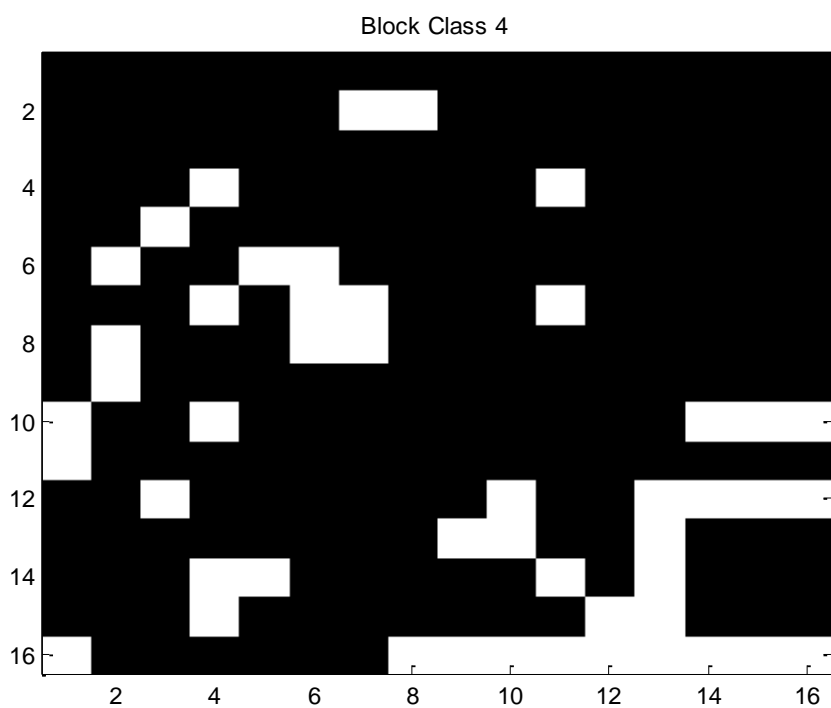
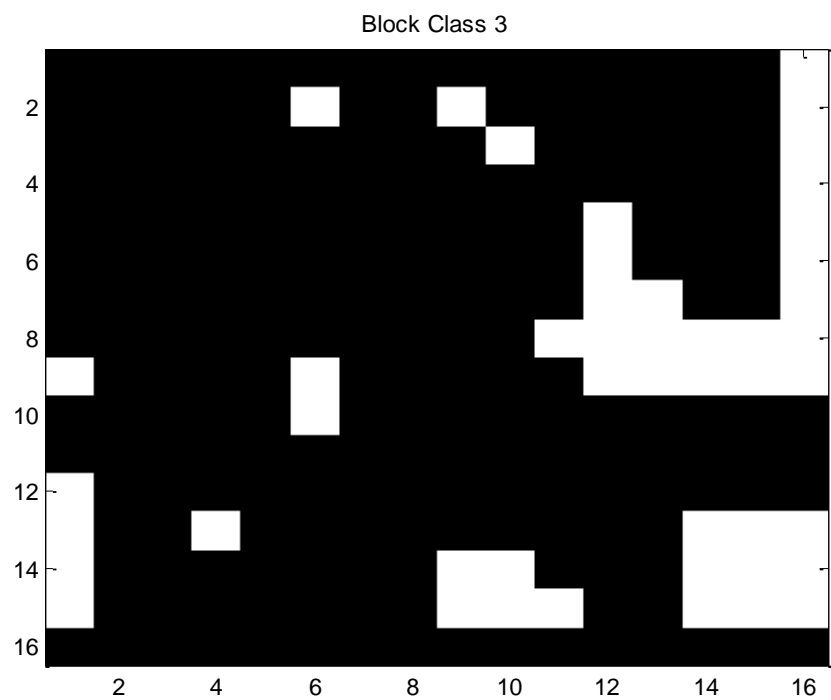


Figure 25 Block Variance Map using block size of 16x16 in each LR image and 50% overlap.

To determine the class membership of each block, we classify each block depending on its variance. In Fig. 25 we can see the log base e of the variance for each block to give us an idea which blocks belongs to each class. Figure 26 shows us in detail the exact location for each class and figure 27 shows us one example of each block class from each of our LR images also with its frequency domain 16x16 DFT log magnitude. This last figure of the frequency domain show us the reason why we select different bandwidths depending on block variance.





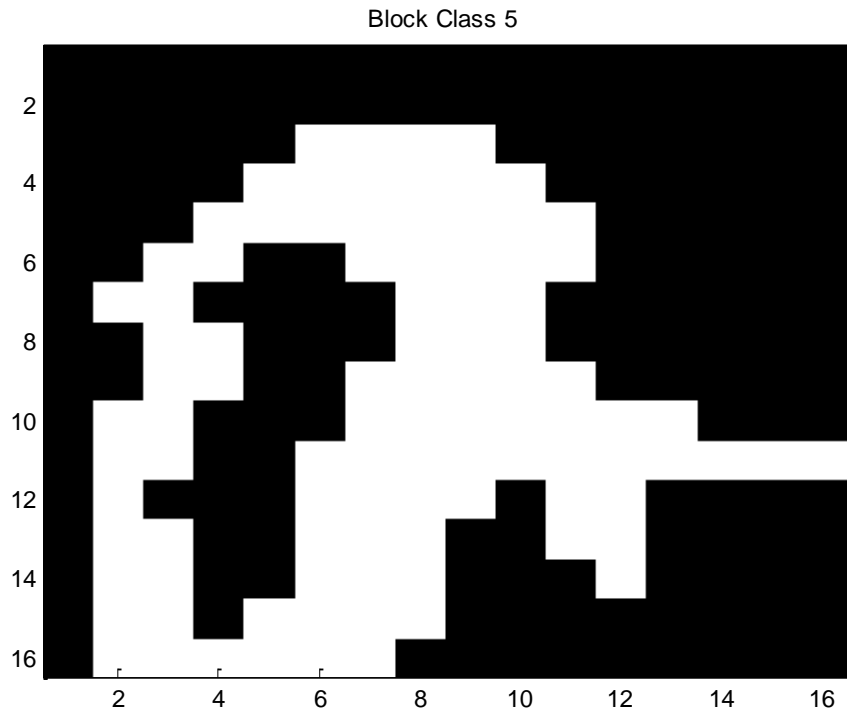
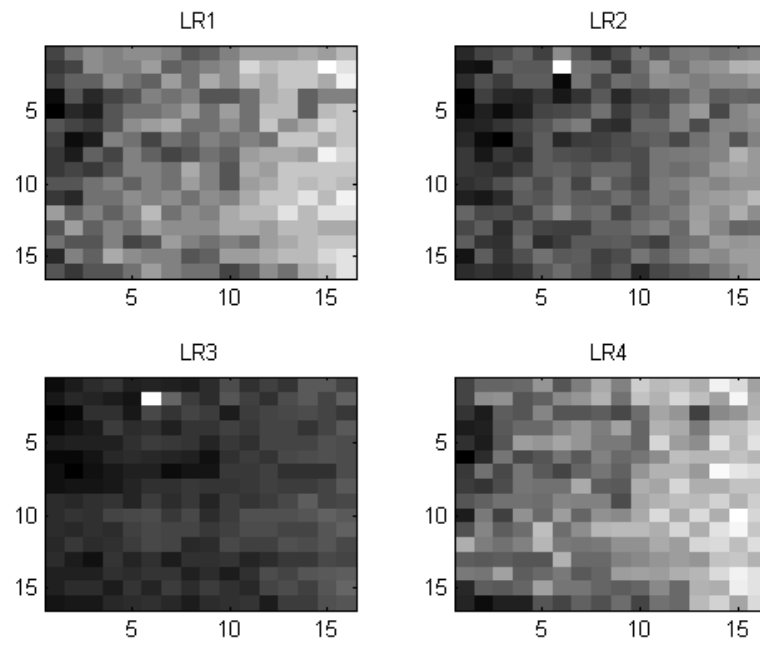
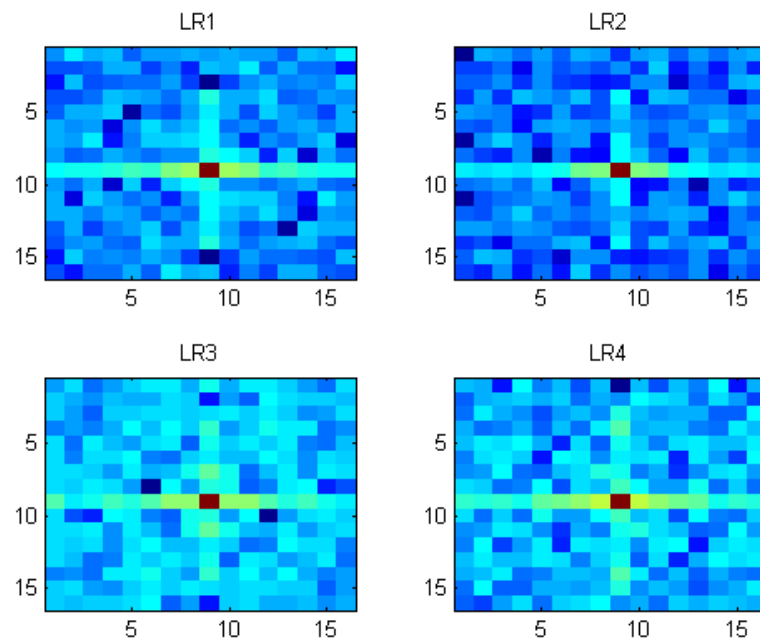
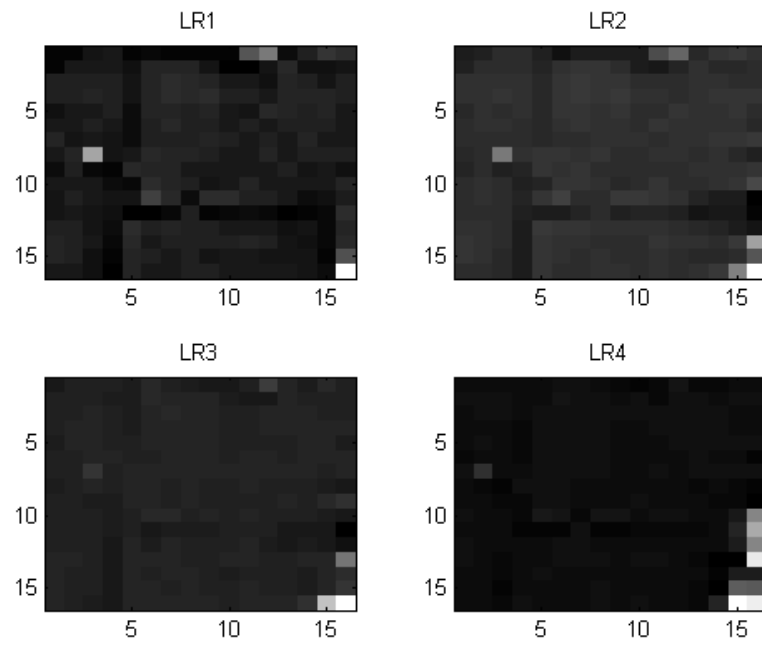


Figure 26 Location of Block classes in our input image Cameraman

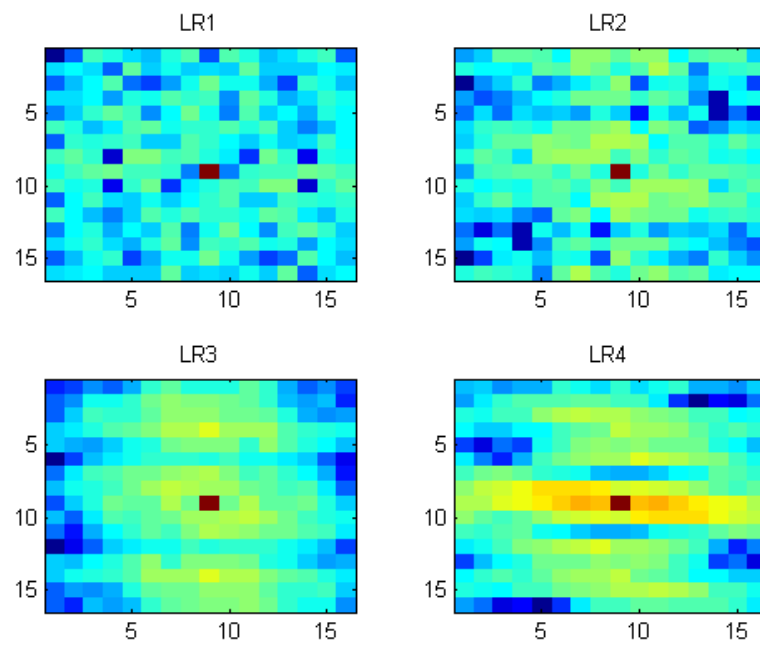


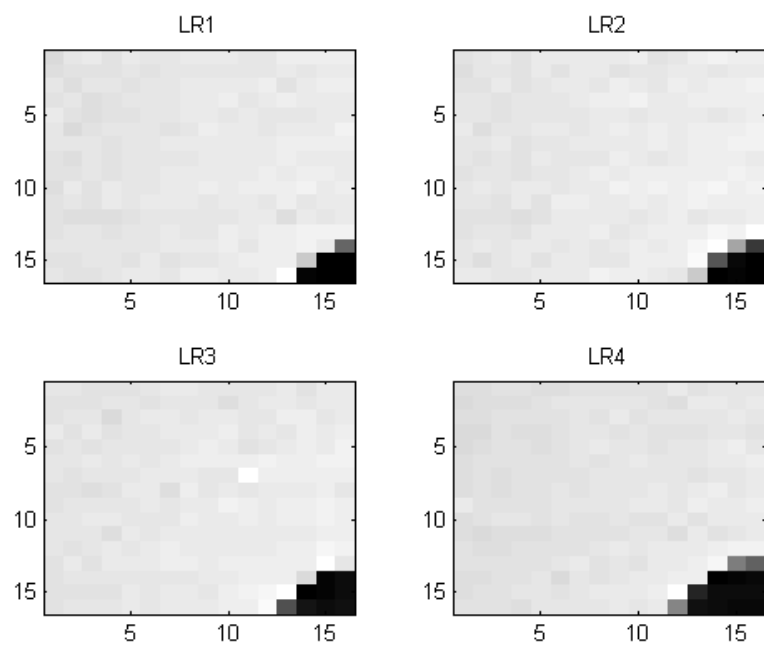
a)



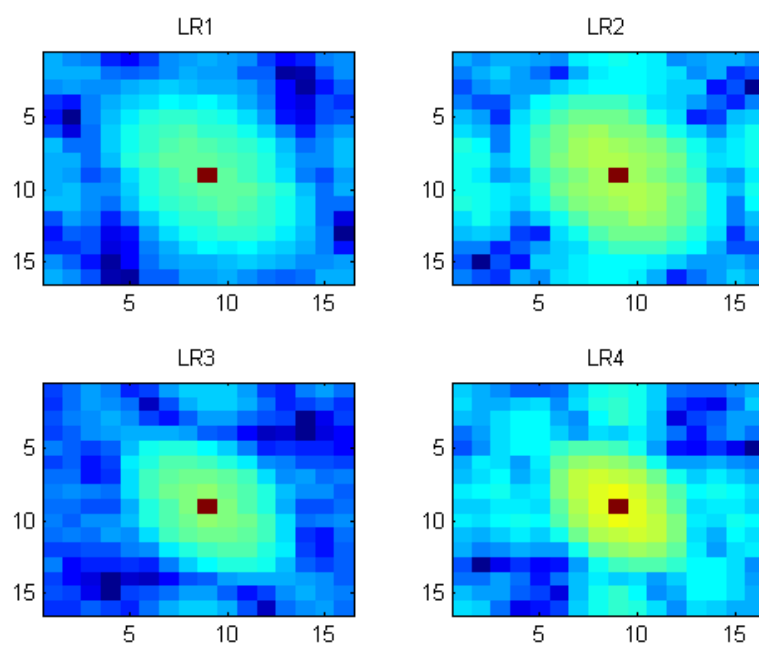


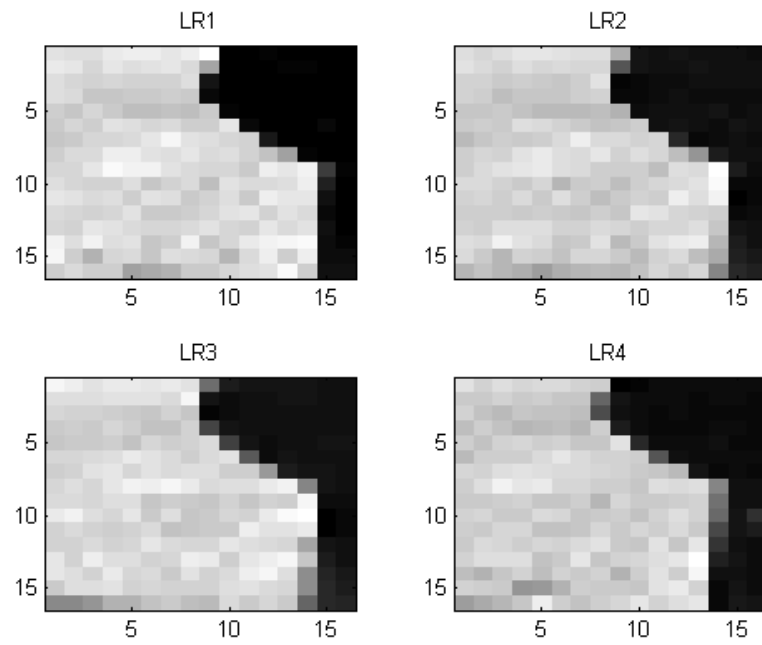
b)



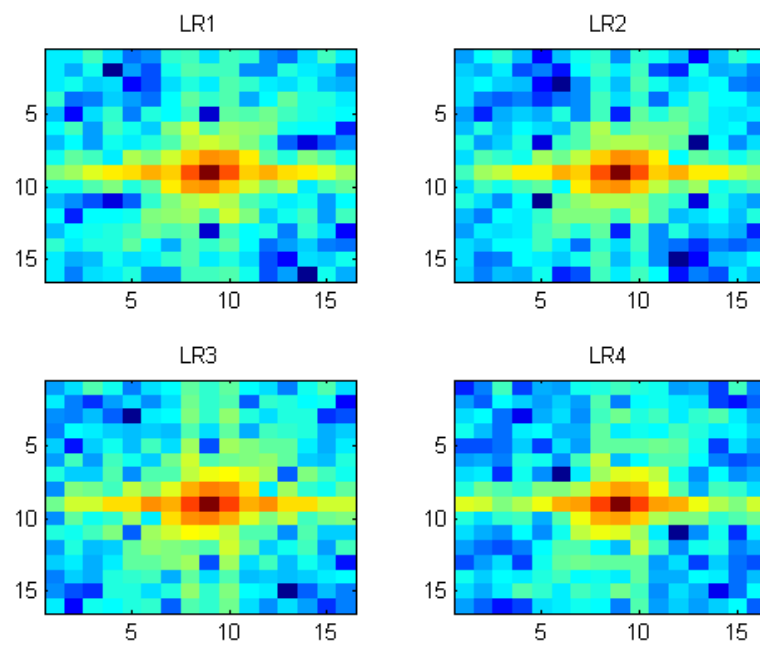


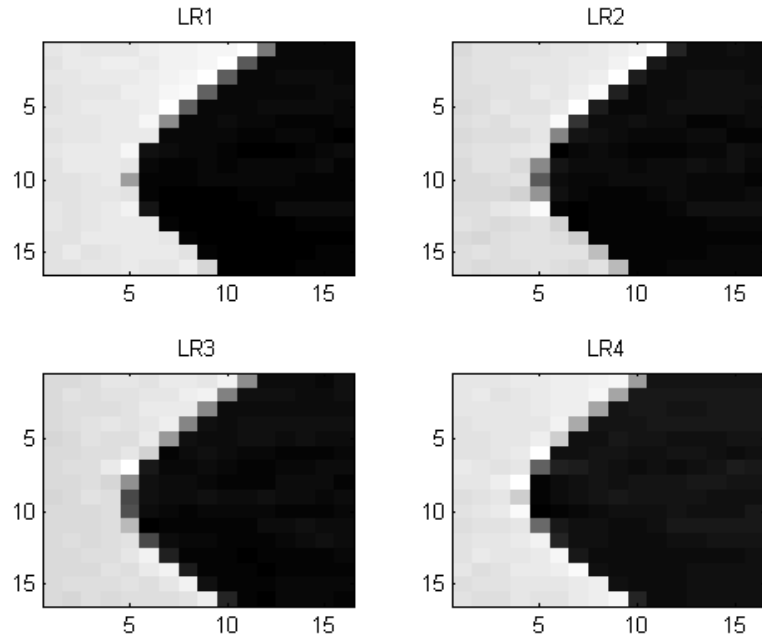
c)





d)





e)

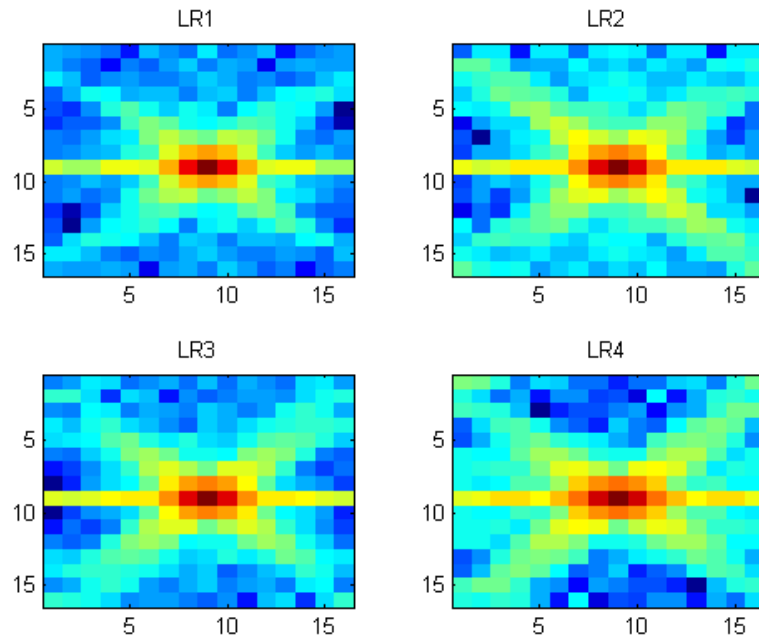


Figure 27 Representative blocks for each class with its spectrum in each LR image, a) Class 1 example, b) Class 2 example, c) Class 3 example, d) Class 4 example, e) Class 5 example

As we can see in the spectrum for each class, we are justified to use different bandwidths for different classes. Class 1 represents blocks inside smooth regions or with low variance, therefore we use a small bandwidth because only low frequencies are important for these blocks while their high frequencies are not relevant. Class 5 represents edges or high frequency texture areas where the variances are much higher, thus high frequencies are important to reconstruct and high bandwidths must be used in the OR reconstruction of these blocks. Table 1 show us the bandwidth we found for each block class in the cameraman input image.

Table 1 Class Bandwidths for the cameraman image

Class	Bandwidth with block size of 8
1	0.1807π
2	0.3473π
3	0.5561π
4	0.7105π
5	0.8408π

5.1.8 Boundary problem

One of the problems we have using SR methods is the possible missing information in our images boundaries. As seen in figure 10, due the shift process, LR 2, 3, and 4 have missing parts of their boundaries. Therefore, our reconstruction near boundaries may differ and affect our algorithm results for the reason that less input information is present in those areas. To solve

our boundaries problem, we decided to modify our LR images before using them as input in any algorithm we tested for this thesis. Because our OR algorithm depends on the size of LR data blocks and special processing must be done near boundaries, we decided to do a pixel replication on our boundaries with the size of one extra block. This will help any algorithm affected by this and the image shifting problem since it will generate extra information in those areas. Also, since we increase the size of our LR images with this method, we also increase the size of our final HR result. Therefore, final results must focus on the center of the image and cut out its boundaries to restore the correct HR image size.

5.2 Block segmentation with edge map

Alternatively, another different way to handle blocks in the image for higher variance classes (classes in edge areas) can be by separating blocks into smaller regions and process each set with the OR theory separately. Once we build our edge map as in chapter 3, we can use it to separate/segment the regions in each block by using the classic labeling method in [18] that returns the 8-connected regions found in our edge map.

For example, in figure 28 we have a small HR block with two edges within, using the labeling method, we can differentiate the three regions in the block and work with each one independently.



Figure 28 Separation regions using labeling and our edges areas from our edge map inside an HR block, in this case we segment the block into three regions.

Now that regions are separated, each region can be reconstructed independently and reintegrated into the same block. By using this modification, only pixels with the same denomination are selected to apply OR reconstruction, reducing the probabilities of selecting “wrong” pixel value. Next we elaborate on this approach that we can use but only discuss it in this section of the chapter.

As we have mentioned earlier, OR theory selects an optimal bandwidth based on the variance information in the collection of LR blocks to be used to reconstruct one HR patch. This means, pixels on both sides of edges may be mixed and generate a wrong reconstruction on those areas. In order to address the problem of high variances in edge areas, we use our estimated edge map to divide blocks into different regions “R” and to apply our OR method to each region separately, see Fig. 29. This method helps us to increase the smoothness on each side of an edge within each block.

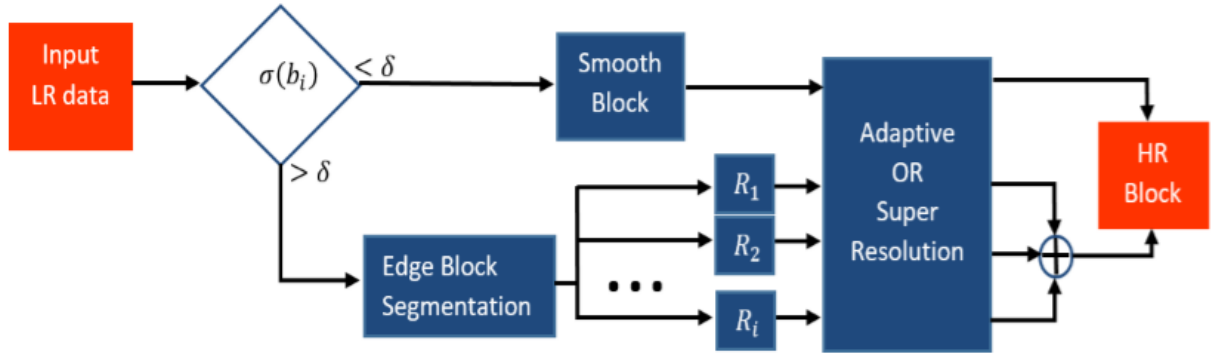


Figure 29 Input LR data b_i is categorized into two different classes based on its variance, smooth block and edge block. Edge blocks are segmented into different regions R_i to reduce region variance and improve the SR process.

Figure 30 show us how regions are being mapped from our LR to our HR in the edge blocks.

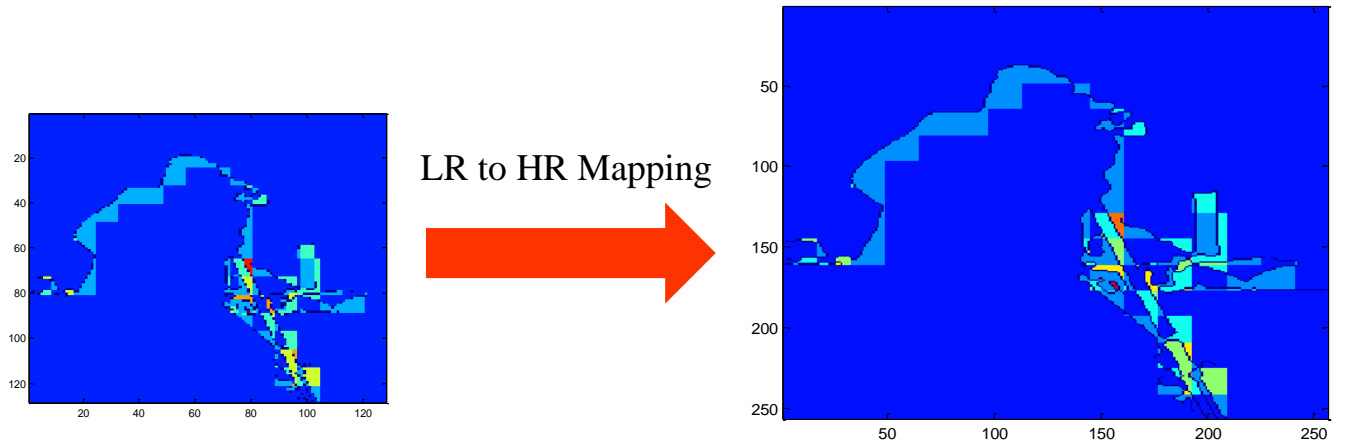


Figure 30 LR to HR regions mapping with region segmentation

Due its complexity, we were only able to implement this idea in our no overlap approach obtaining an enhancement in edges areas at the cost of increasing the amount of time it takes for the final result which makes the code less efficient. It should be noted that the edge map pixels need to be merged with one of the neighboring region by using an HR mapping seen in figure 30. Also the pixel locations in region R in the HR block are downsampled to select corresponding LR input data sample locations. Figure 31 show us how one block is being processed and mapped to achieve a final result and figure 32 show us a comparison with actual results without using block segmentation.

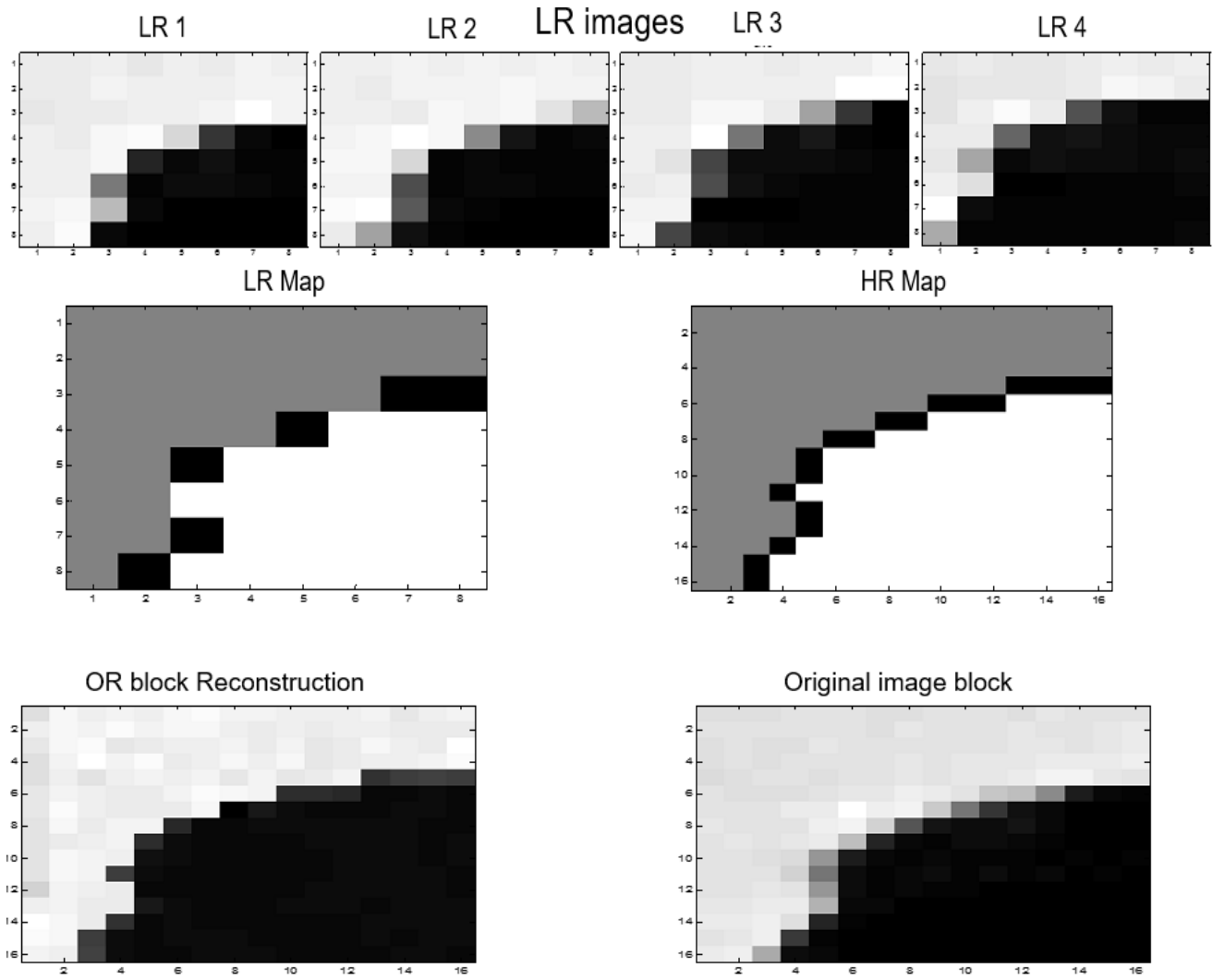


Figure 31 HR Block reconstruction process and mapping using region segmentation

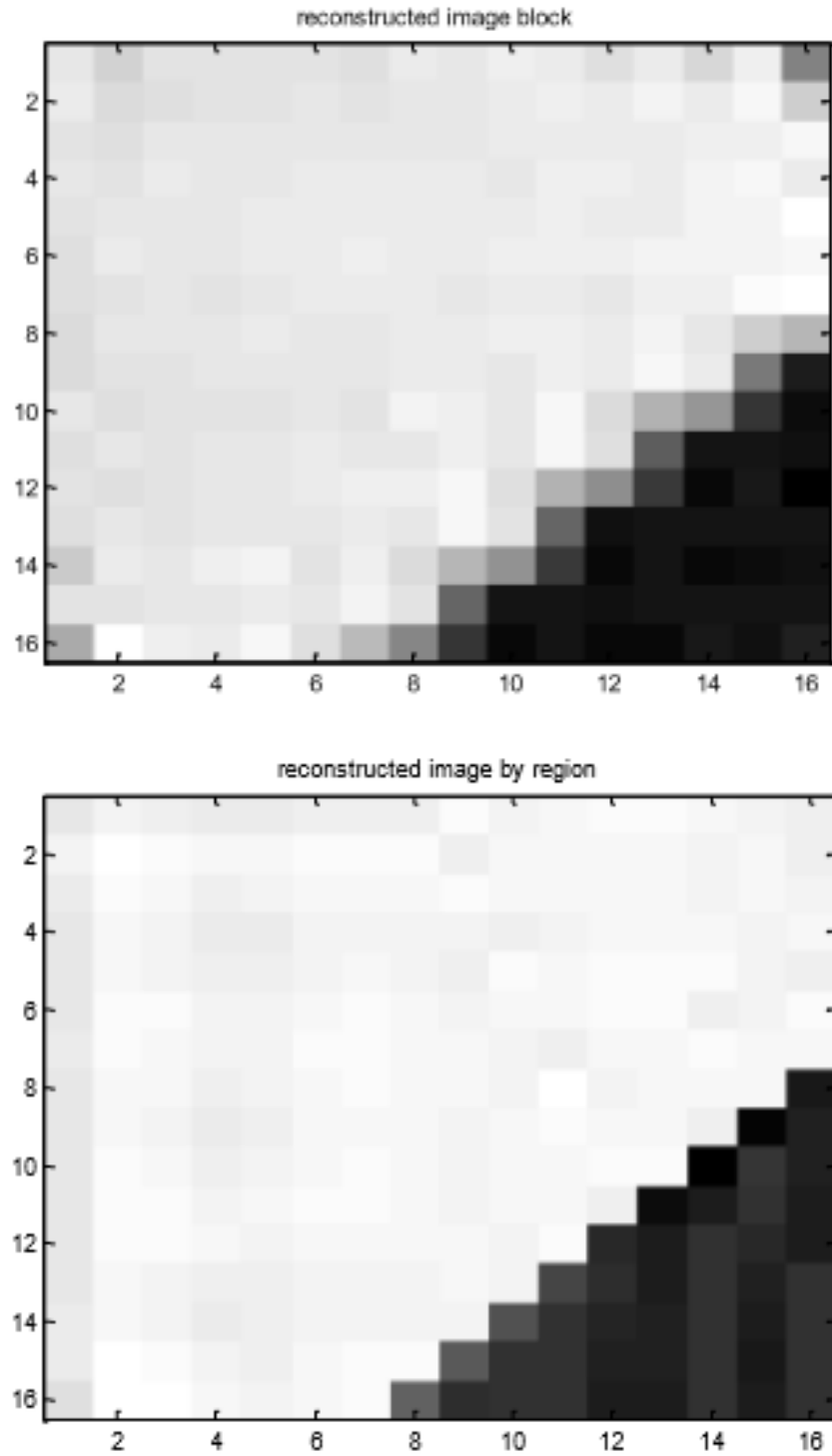


Figure 32 Comparison between resulting image with old approach (top) and with regions segmentation (bottom).

5.4 Incorporation of diffusion to our SR-OR algorithm

As we have discussed, Perona-Malik (PM) Anisotropic Diffusion can help us to improve our reconstructed edge areas. To incorporate this method, we have the option to use it as a post-processing operation on the full-size HR image (global processing), or use it locally as post processing of each entire reconstructed HR block right before selecting a subset to become part of the full-size HR image. Figure 33 shows us how our reconstructed HR blocks look in one row of blocks of the image before apply subset selection to assemble part of the full size image.



Figure 33 Original patches/blocks before extraction of subsets of pixels to become part of the final image.

5.4.1 Post Processing Anisotropic Diffusion

For the post processing, first we must obtain the final result from our OR Super-Resolution method then apply a version of PM diffusion. This approach will use the whole image as an input, which means we will operate as global processing. In this case, we will use the modified PM version used by Maiseli, discussed in chapter 4, since we desire to improve our edge areas in our image by incorporating OR estimated gradients. Also, for the gradients we will compare results using our estimated gradients vs gradients from the original image. Results using this approach are displayed in Figures 34 and 35.

OR Result with Post-Processing diffusion using Estimated Gradients



OR Result with Post-Processing diffusion using Sobel Gradients



Figure 34 Comparison of Post-Processing diffusion with estimated gradients (top) and Sobel gradients (bottom)

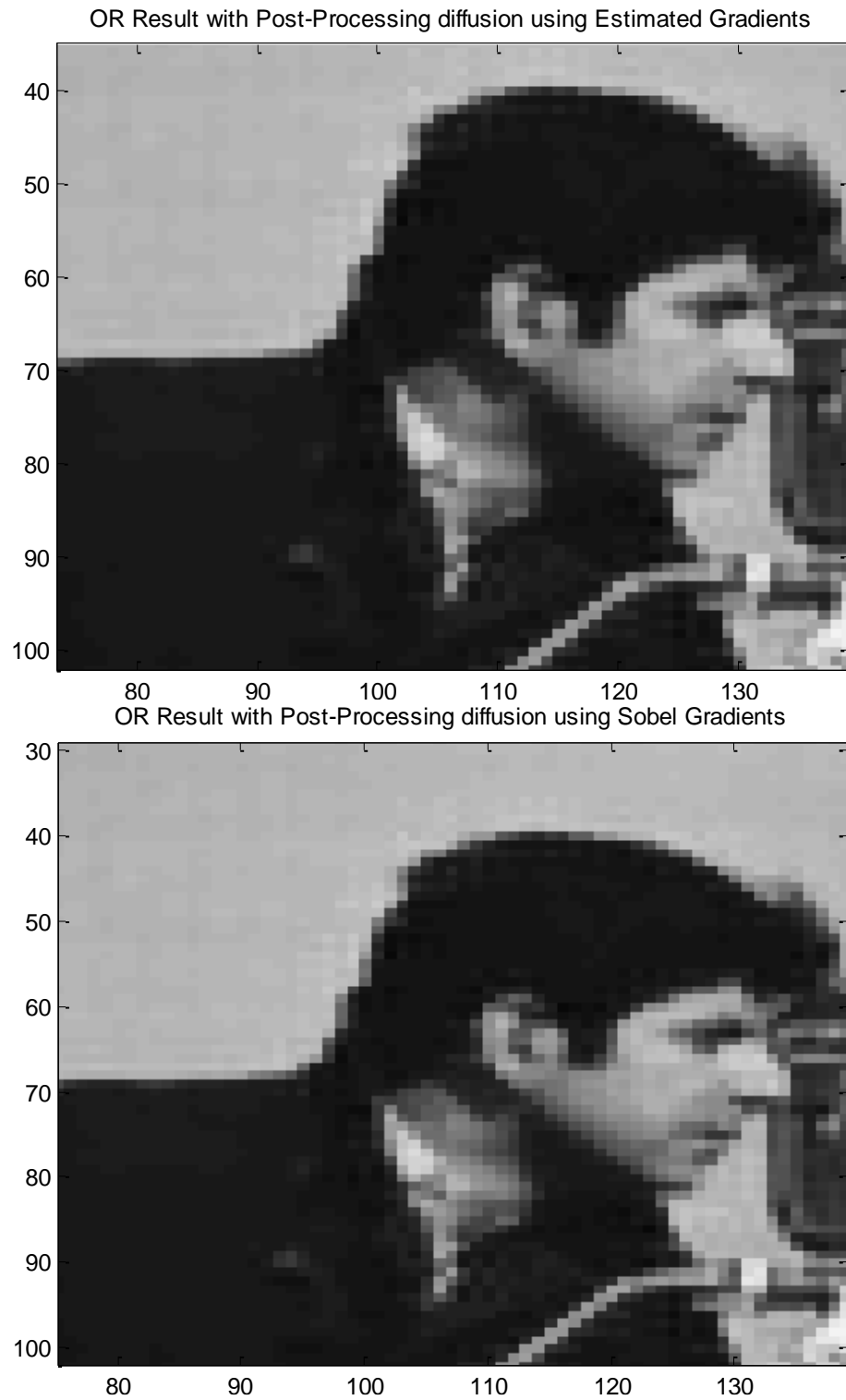


Figure 35 Zoomed version of the comparison of Post-Processing with estimated gradients (top) and Sobel gradients (bottom)

5.4.2 Block based post processing Anisotropic Diffusion with improved gradient

Alternatively, we can also use Diffusion locally in our OR method on each of our reconstructed HR blocks. Again, using the modified version of PM diffusion with our estimated gradients, we diffuse each block separately before sub-block selection. Final results can be seen in Figure 36 and 37. We also include a comparison with the global post processing result.



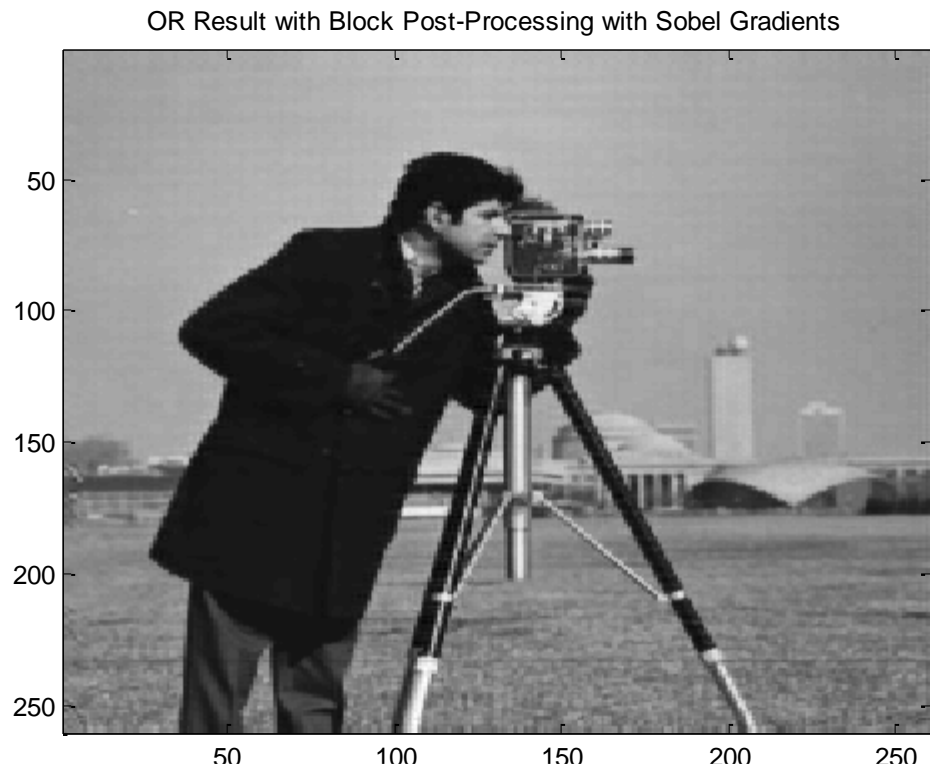
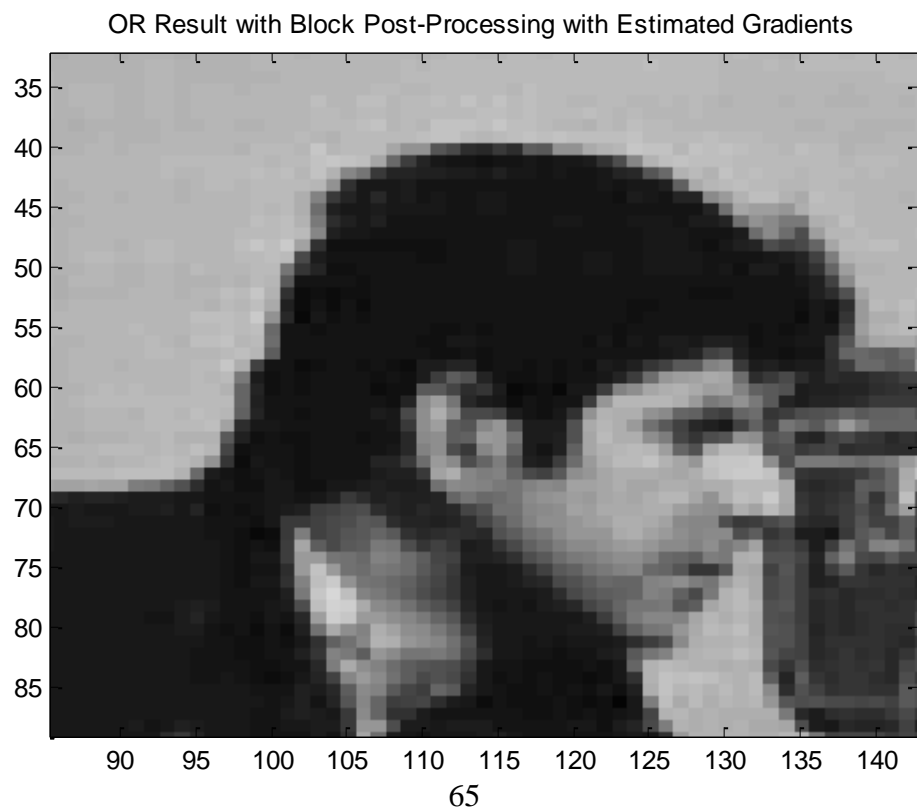


Figure 36 Comparison of Block Post-Processing with estimated gradients (top) and Sobel gradients (bottom)



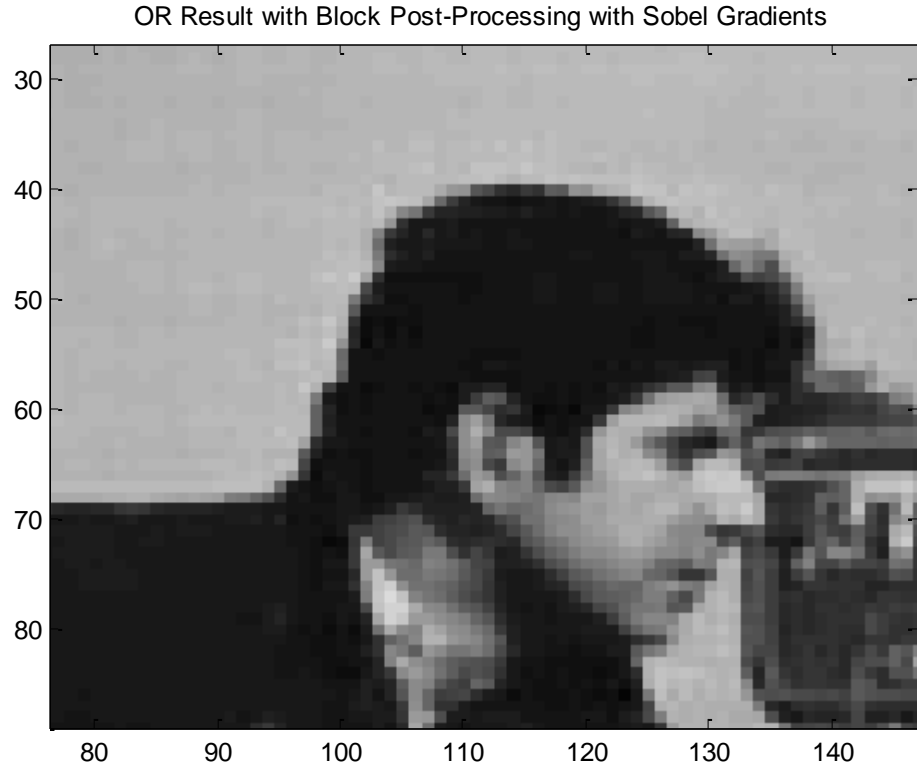
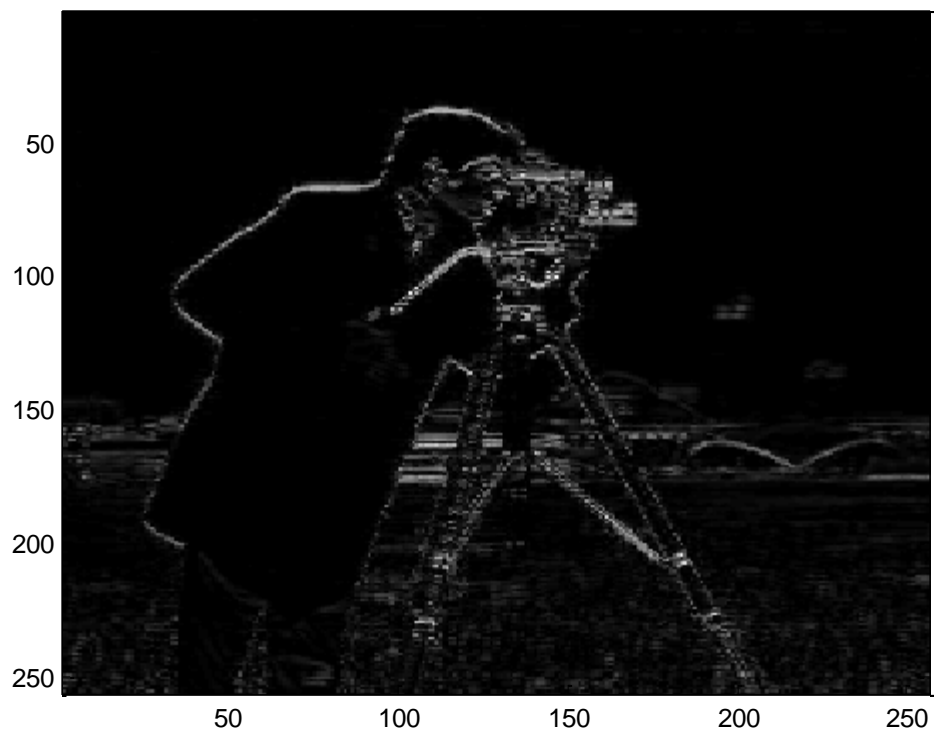


Figure 37 Zoomed version of the comparison of Block Post-Processing with estimated gradients (top) and Sobel gradients (bottom)

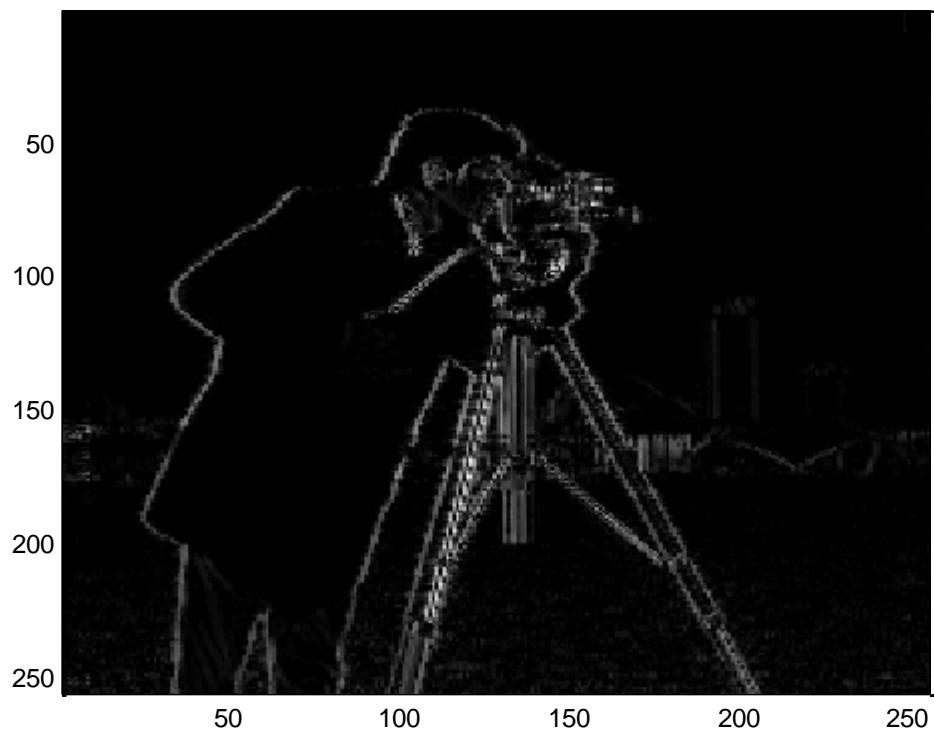
Disadvantages using the global approach is that diffusion will not have the extra information that we destroyed (subset selection) at the moment we overlap to form the assembled HR image. Using diffusion in the local approach will take more time to operate since we require to call the diffusion function for each reconstructed patch.

For curiosity purposes, we can also apply PM diffusion to our estimated gradients to obtain results in figure 38.

Gradient Gx



Gradient Gy



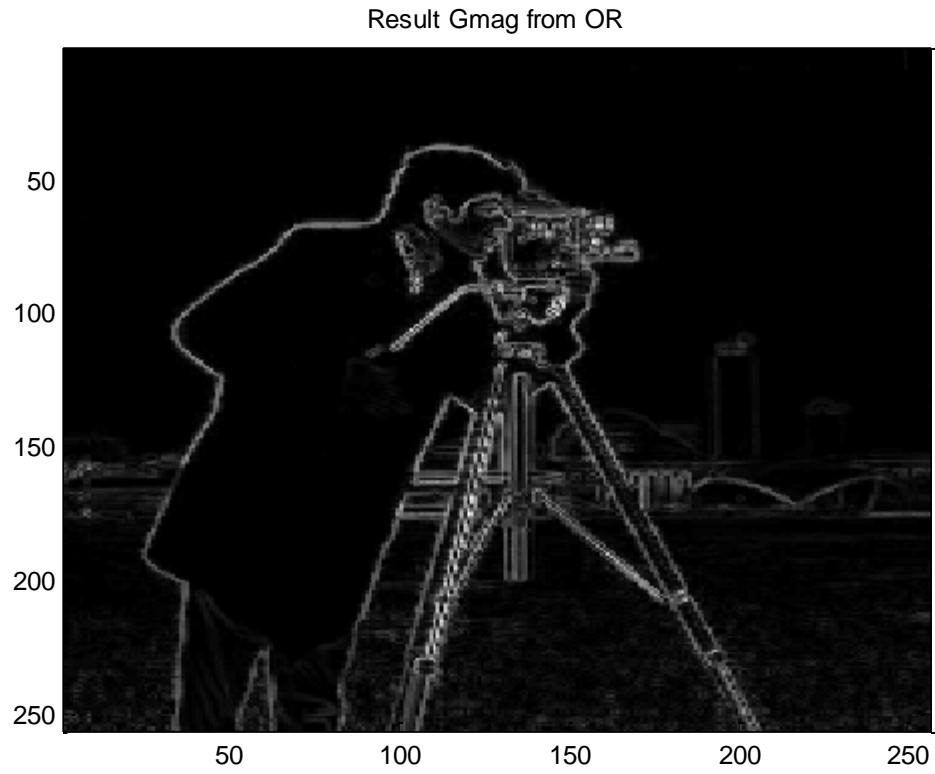


Figure 38 Estimated gradients after being diffused with original PM method

Chapter 6 Revisiting Papoulis-Gerchberg (PG)

The reason why we select PG method as our main competitor is because as in OR theory, this SR algorithm uses bandwidth information to generate its HR image. We will discuss some errors we found in this code from a MATLAB app and how this method has evolved by integrating modified PM anisotropic diffusion by Maiseli [21].

6.1 The PG SR method and code fixes

The Papoulis-Gerchberg algorithm is a method that performs signal reconstruction by limiting the spectrum to a known bandwidth and extrapolating/interpolating missing information (HR image pixels) by using the constraint imposed by the known LR information. Using the Vandewalle MATLAB application for super-resolution [12] we were able to use the PG method that the research community has been using for their papers including [8]. However, after working with this code for a while we found some errors. By running the PG method, we noted that in the app it only worked for magnification factors that are multiples of 4, for other cases (2,6, etc) the code showed undesirable results due to the automatic selection of the bandwidth parameter. Also, we discovered that registration is limited to the use of integer shift values not allowing fractions or arbitrary numbers. Figure 39 shows a SR magnification of 2 image result from the MATLAB app after we fixed it by properly selecting the bandwidth parameter to less than $\pi/2$.

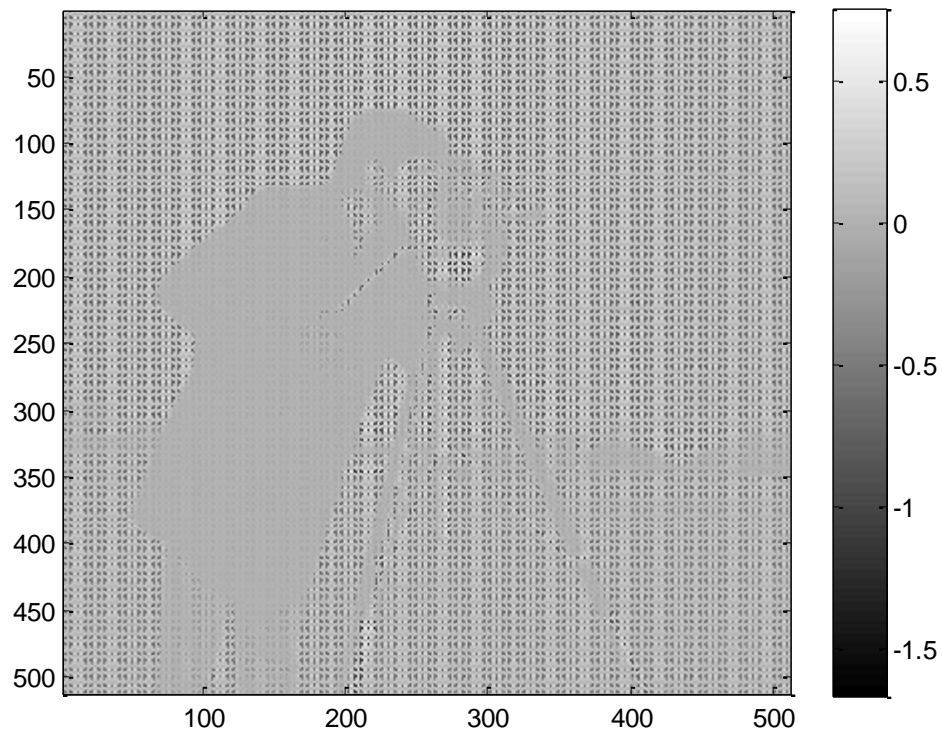


Figure 39 Papoulis-Gerchberg Results from MATLAB app after code fix

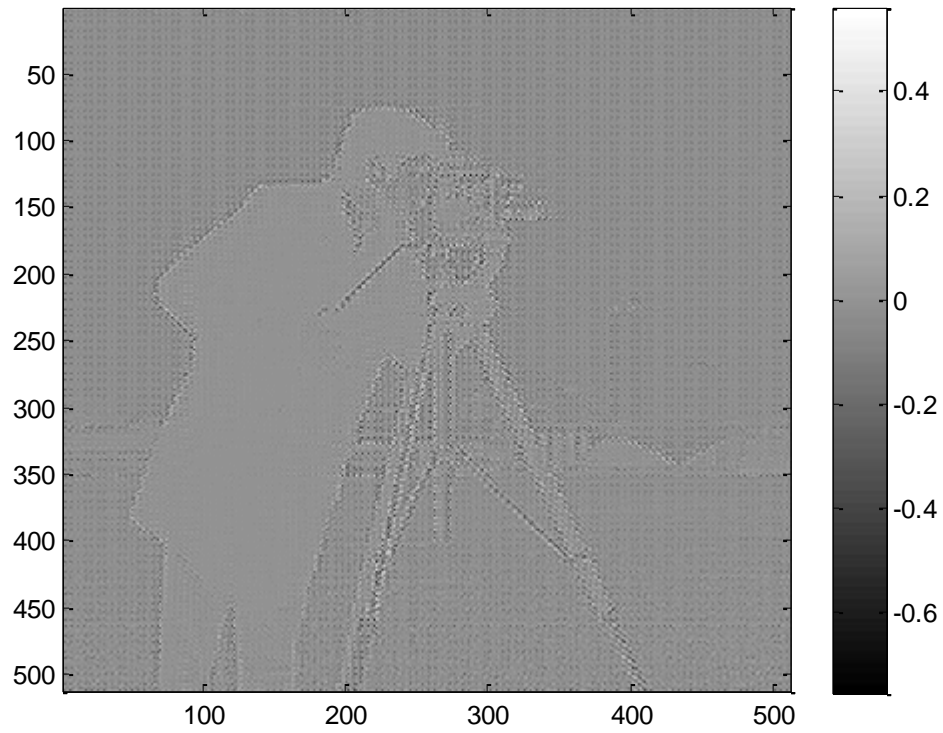
6.2 Maiseli diffusion enhancement of PG for SR

Maiseli proposed a PG method combined with the modified Perona-Malik Anisotropic Diffusion method using gradients. This is done by inserting a diffusion step at each iteration right after the PG method restores the know LR pixels on the bandlimited version of the previous iteration's HR reconstruction. As we discussed in chapter 4, Maiseli used a modified Diffusion method that incorporates a diffusivity that makes use of the gradient of the image. He states that using this diffusivity, the SR method would take advantage of the edge enhancement that the diffusion process provides and the final results in PG method would get improved. In figure 40 we display the diffusion divergence image in the first and tenth iterations and the final results from the algorithm after 11 iterations with magnification of 4

Divergence image in first iteration



Divergence image in tenth iteration



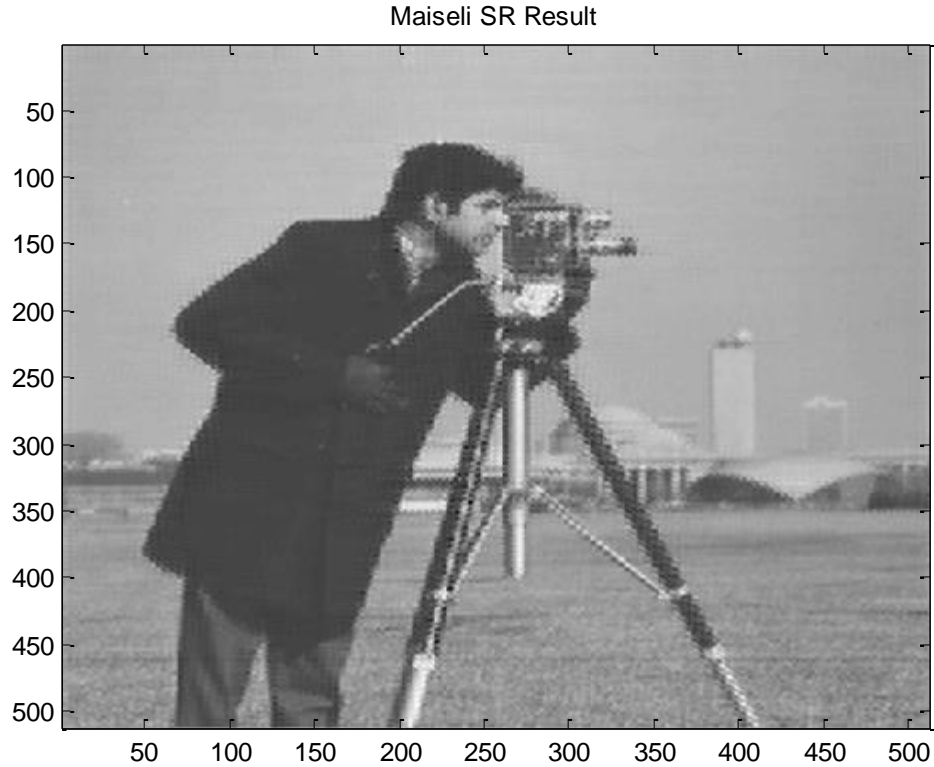


Figure 40 Divergence images in first and tenth iteration of Maiseli algorithm and its final result after 11 iterations.

We made the same correction we did on the plain PG code in the MATLAB app. Also, we changed the *gradient* MATLAB function call and replaced it with the *imgradient* which is the 2D version. We discovered that these two functions have some differences and if we use images we should use the *imgradient* function to avoid some issues. Also, we discovered that the *imgradient* function does not rescale by 4 or 8 when obtaining the gradient using the Sobel mask, so we also divided by 4 in order to rescale each gradient in x and y directions.

Chapter 7 Experiments and Results

For the experimental results, we used the same LR shifts as the previous student [8] to get a fair comparison against previous results. LR shifts in the x (row) direction are 0, 0.12, 0.6, and 1 while in y (column) direction they are 0, 0.2, 0.28, and 0.8, no rotation is assumed and all SR algorithms used here use known registration parameters (shifts and no rotation). The LR images generated follow the process of figure 9. In this section we show results using a single bandwidth over all blocks in the image, revisit results with overlap vs no overlap SR-OR results, discuss results with anisotropic diffusion with the modified PM method in the global and local approach, and finally we compare these results with other state-of-the-art SR algorithms with low complexity that are available from the MATLAB app. Also, we discuss how large shifts affect our SR algorithm. Not all results are displayed images, some are only given in the final table of quality metrics. Block diagrams for full implementation can be seen in figures 41, 42, and 43.

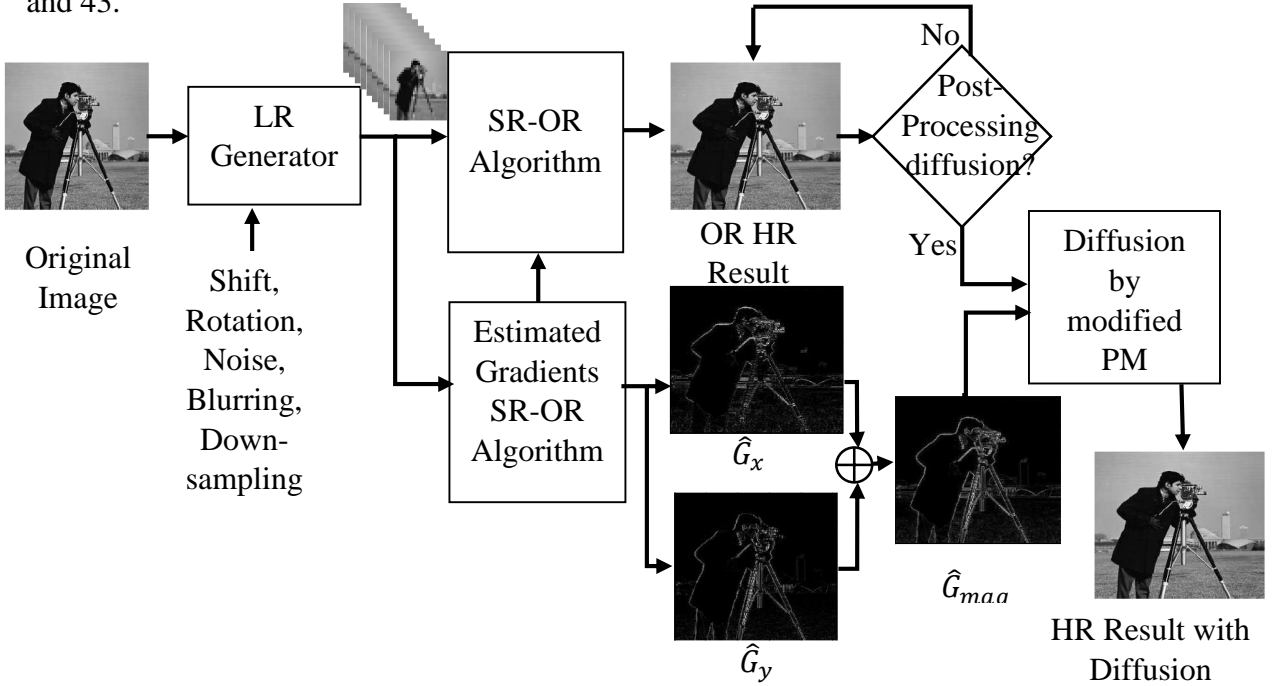


Figure 41 Top level block diagram of our various SR-OR experiments and methods

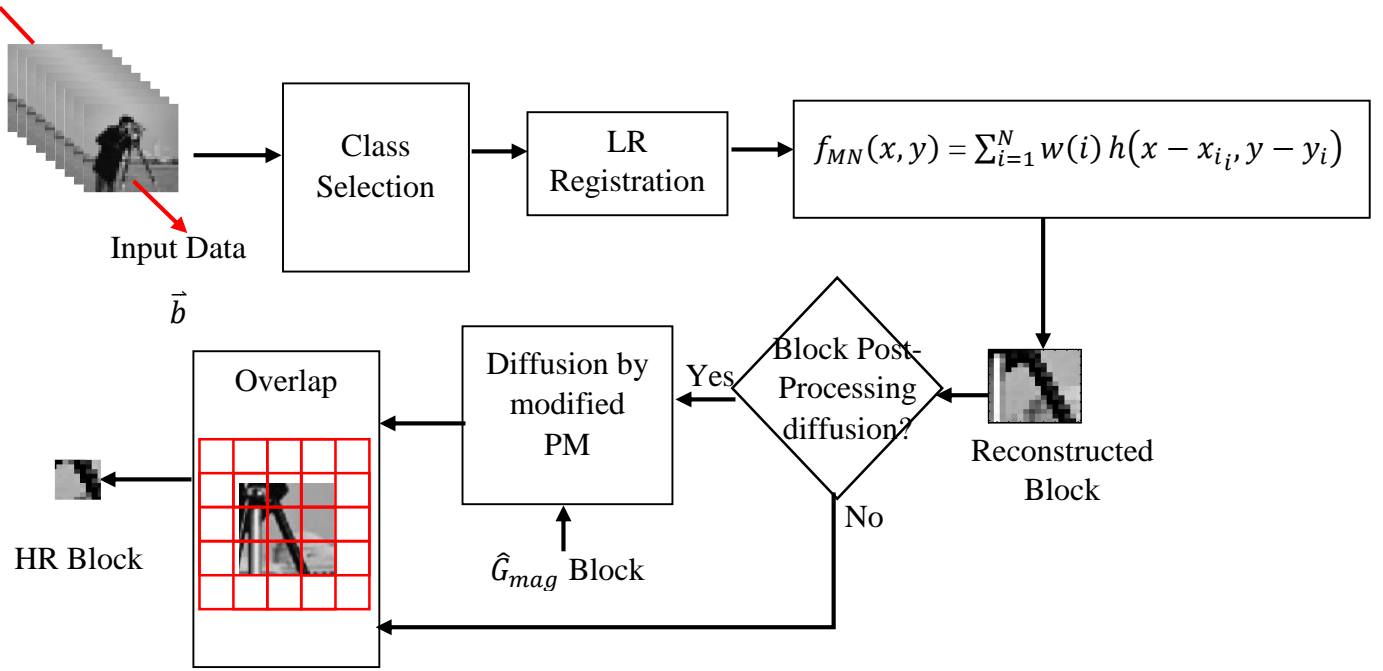


Figure 42 More details of the SR-OR Algorithm block diagram

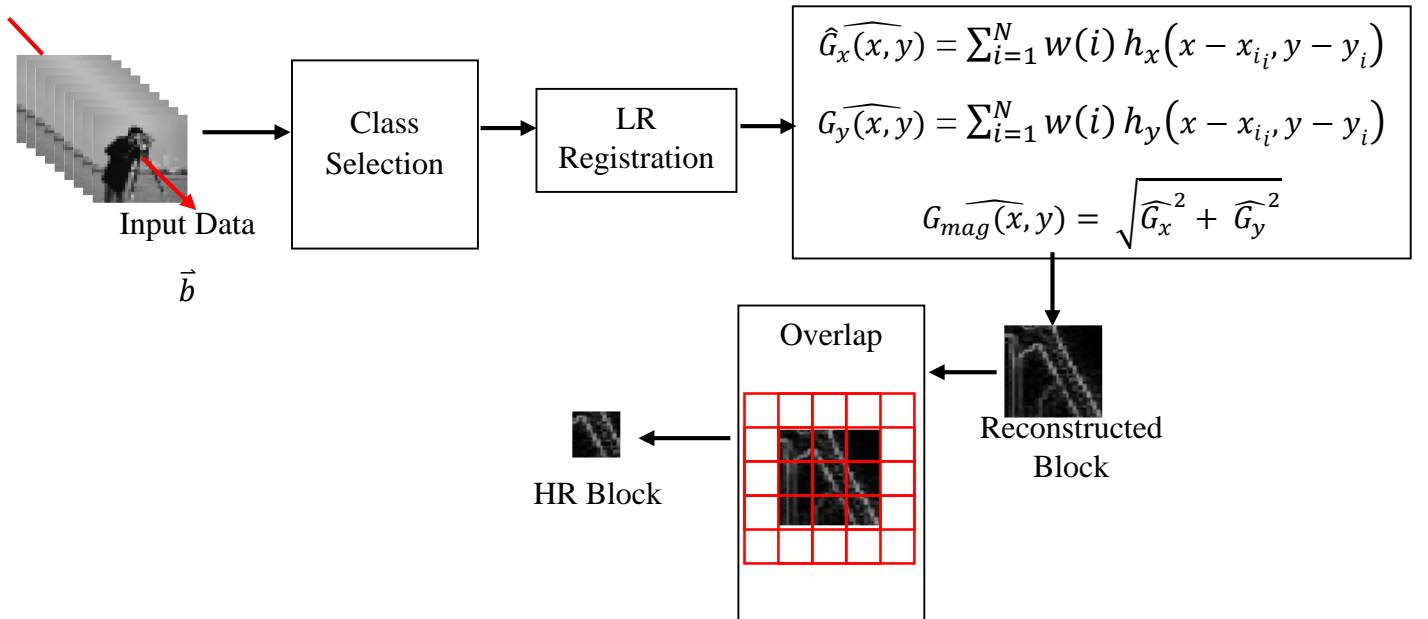
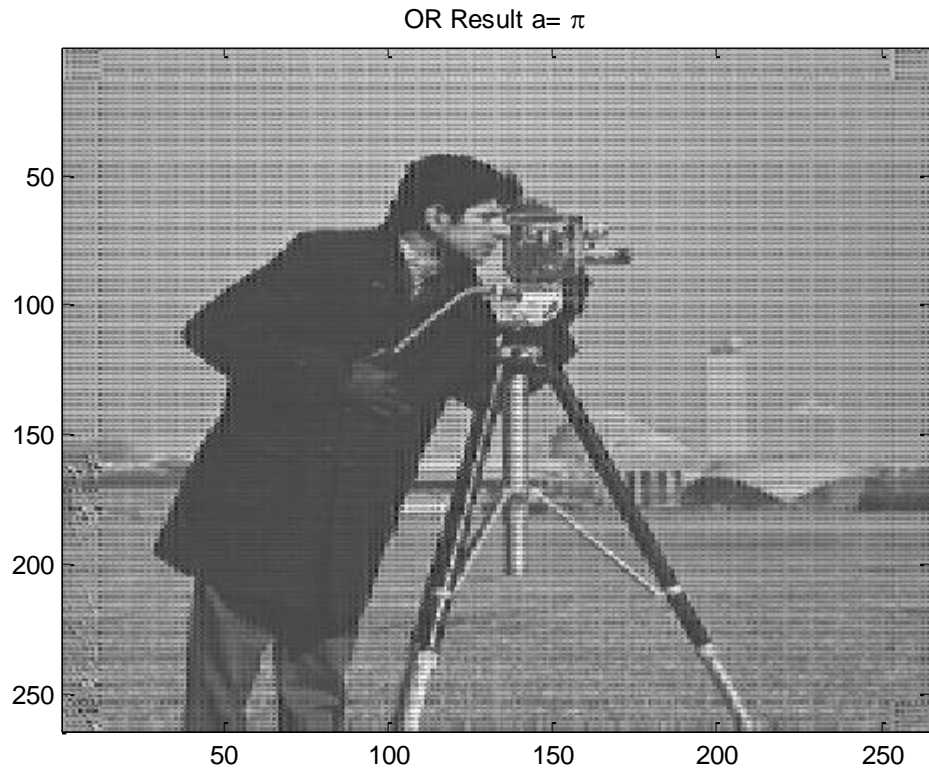


Figure 43 More details of estimated gradients SR-OR Algorithm block diagram

7.1 Lower bound results with maximum bandwidth ($a = \pi$)

Lower bound results for our OR Super-Resolution algorithm will assume single full bandwidth $a = \pi$ parameter for all blocks, ignoring the classification of blocks in the image, see Fig. 44 which shows the reconstructed image and its spectrum using $\log_e(1 + \text{magnitude})$ contrast enhancement.



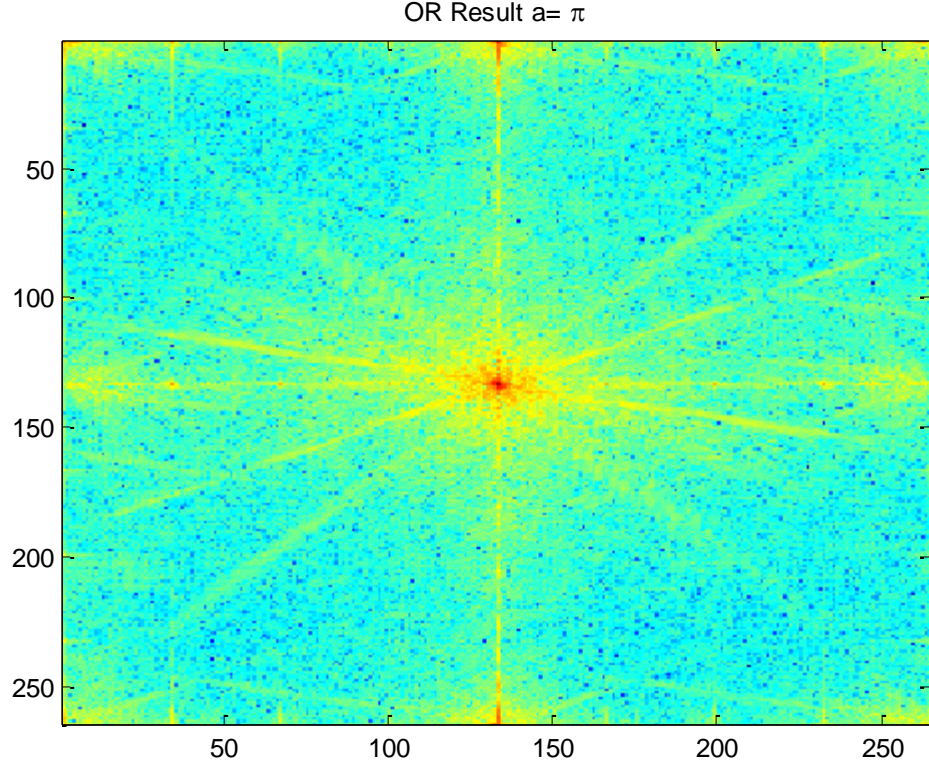


Figure 44 Results using single full bandwidth $a = \pi$ on our OR Super-Resolution. Spatial and DFT-based spectral domain view using log contrast enhancement.

For our lower bound error we got an $MSE = 2.1291e+03$, and a $SSIM = 0.2305$. If we clip the resulting image to 8-bit integer values we get $MSE = 146.8217$ and $SSIM = 0.3434$. This tells us that our results cannot be worse than these values since our maximum bandwidth possible is π in the reconstruction of each HR block using the OR approach.

7.2 Revisit Overlap vs No Overlap OR Results

Using all improvements we mentioned in chapter 5 we revisit results from previous work [8] on OR based Super-Resolution with the overlap and the no overlap approaches. For no overlap, the block size used to take data from the 4 LR images is 4 by 4 and HR blocks are the smallest possible of size 8 by 8. For the other case, we use 50% overlap in the subset

selection so we keep only the middle 4 by 4 pixels from each 8x8 reconstructed HR block and discard the rest. Figures 45 and 46 can be compared to Figs. 2 b) and 2 c) in [8]



Figure 45 Result using Overlap approach using all improvements from chapter 5

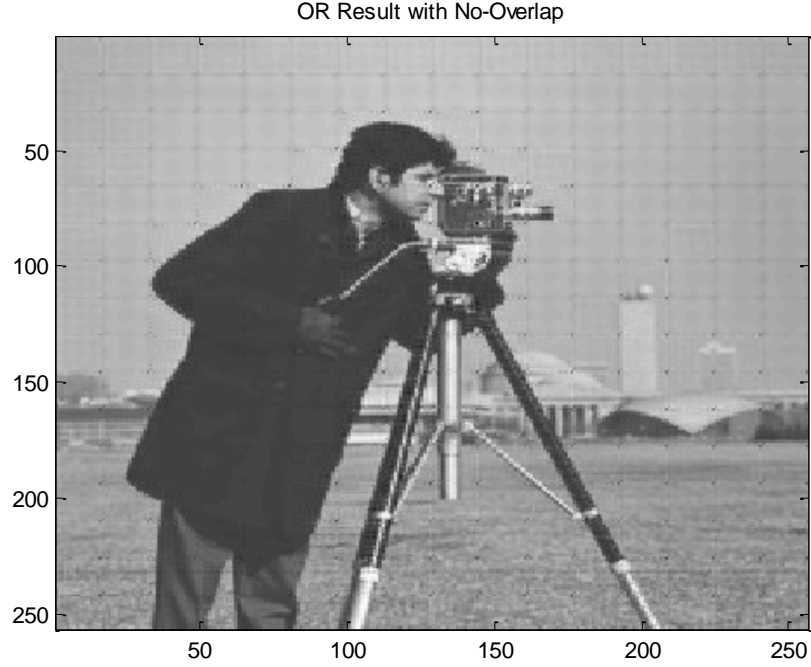


Figure 46 Result using No-Overlap approach using all improvements from chapter 5.

7.3 Examples OR with Anisotropic Diffusion

Here we show our (post-processing of Fig. 45) global diffusion result we obtained when we used our estimated gradients in the modified PM anisotropic diffusion used by Maiseli [21], see Fig 47. Also, we show the result when we diffuse each HR block separately right before it is cut in size to assemble the HR image by mosaicking, see Fig 48. Finally we generate an example using our ground truth Sobel gradients from the original image using global diffusion.

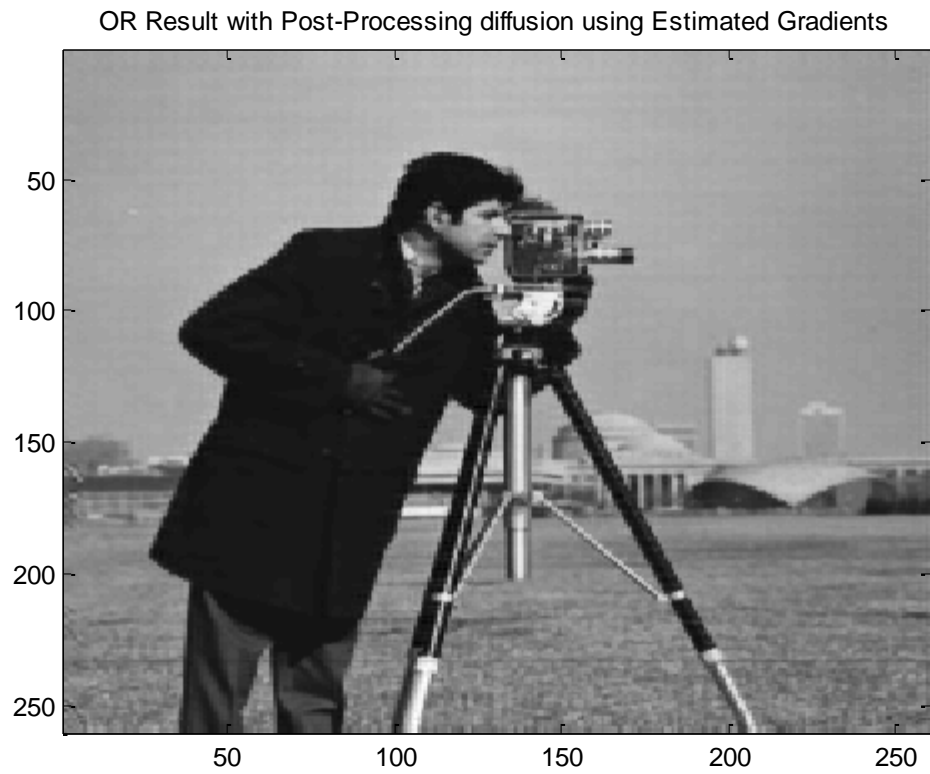


Figure 47 OR Super-Resolution Result using Post-Processing Maiseli diffusion

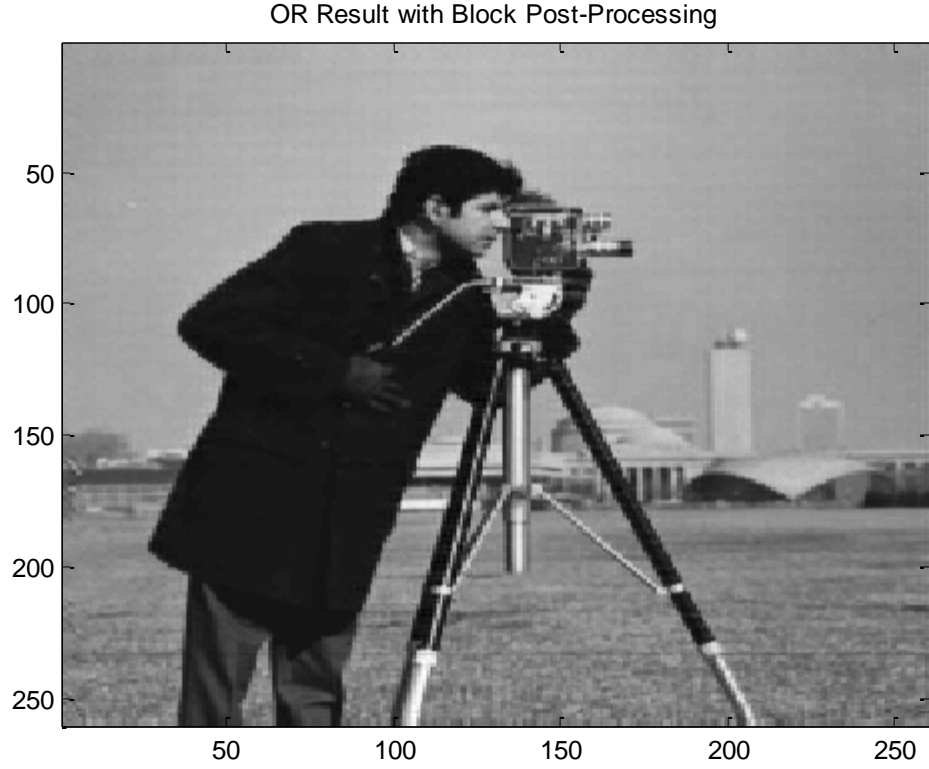


Figure 48 OR Super-Resolution Result using Block Post-Processing Maiseli diffusion

7.4 New Comparison with others SR methods

Here we compared our OR Super-Resolution results with algorithms available from the MATLAB app [12] using the same LR images with the same known registration parameters in every method. We use the celebrated Cameraman image of size 256x256 (see figure 49) and the Liftingbody image of sizes 512x512 (see figure 52) as references. Comparisons are made using the MSE and SSIM image quality metrics.

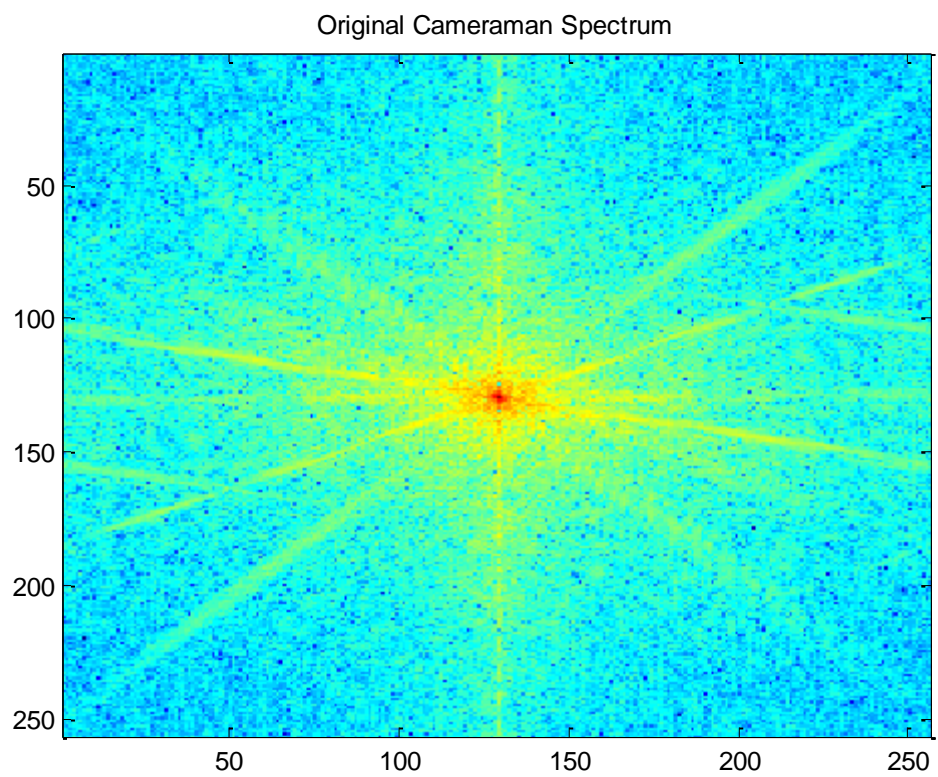
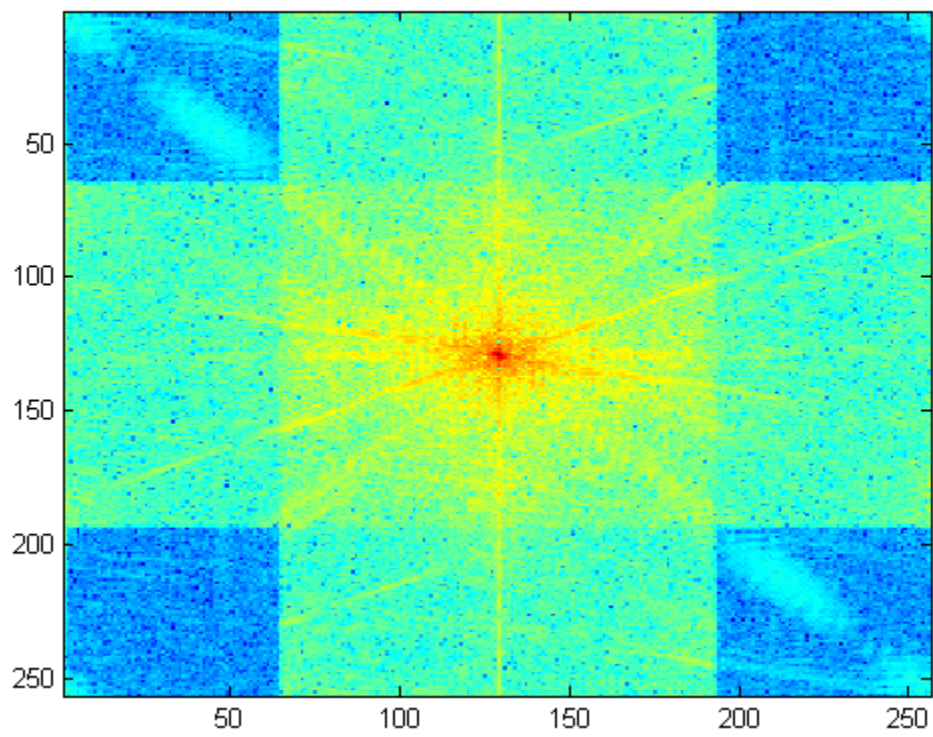


Figure 49 Original Cameraman and its DFT (spectrum domain) view using log contrast.

Papoulis Gerchberg



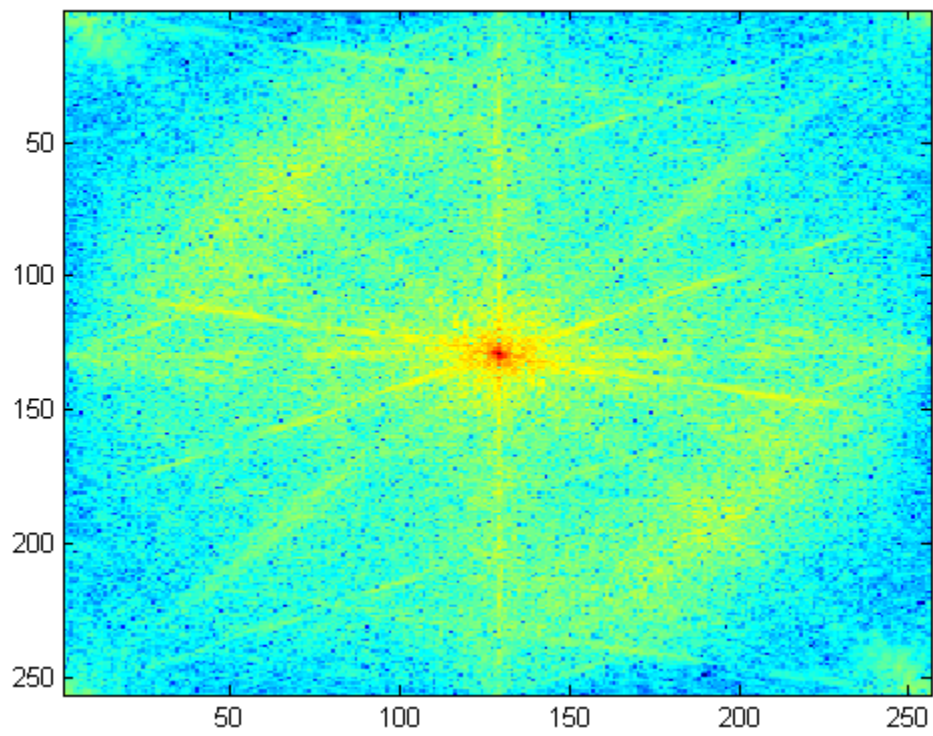
PG Spectrum



Iterated Back Projection



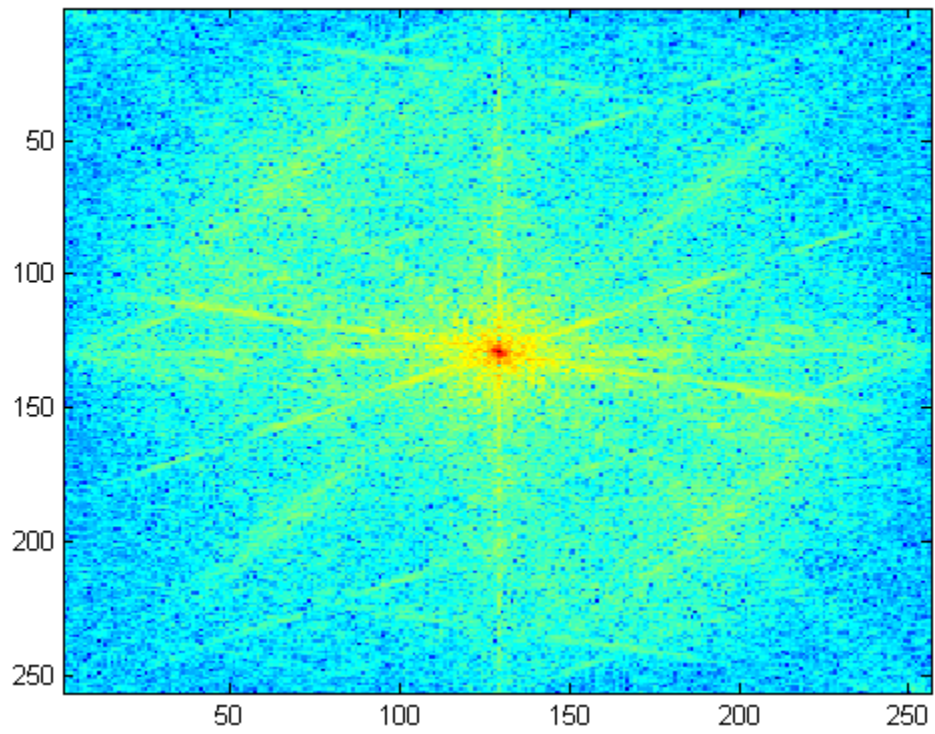
BP Spectrum



Robust Super-resolution



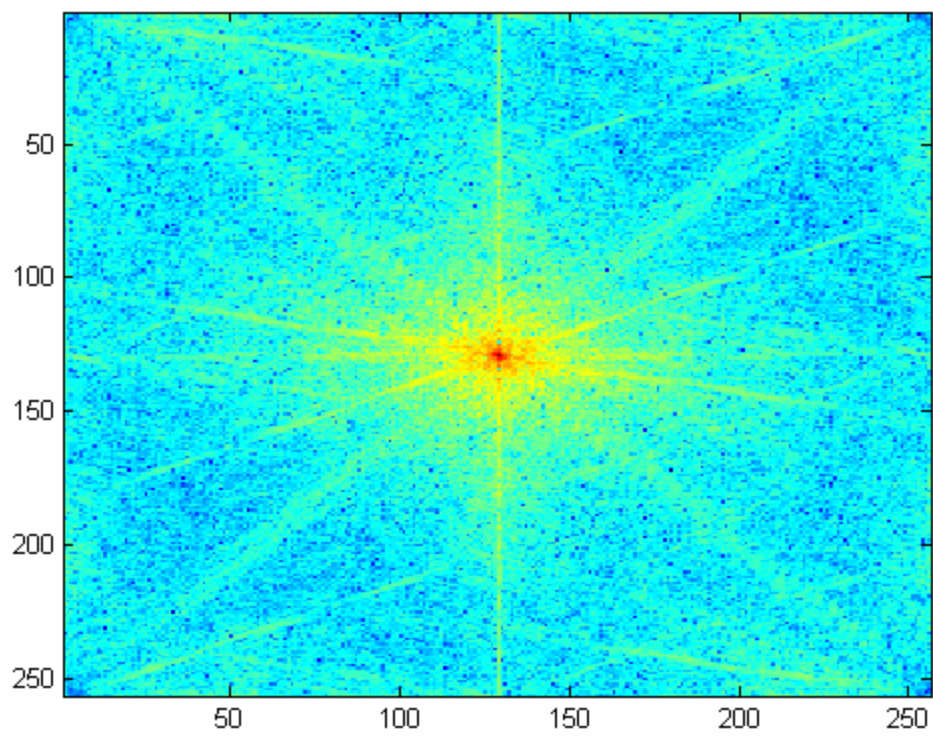
RS Spectrum



POCS



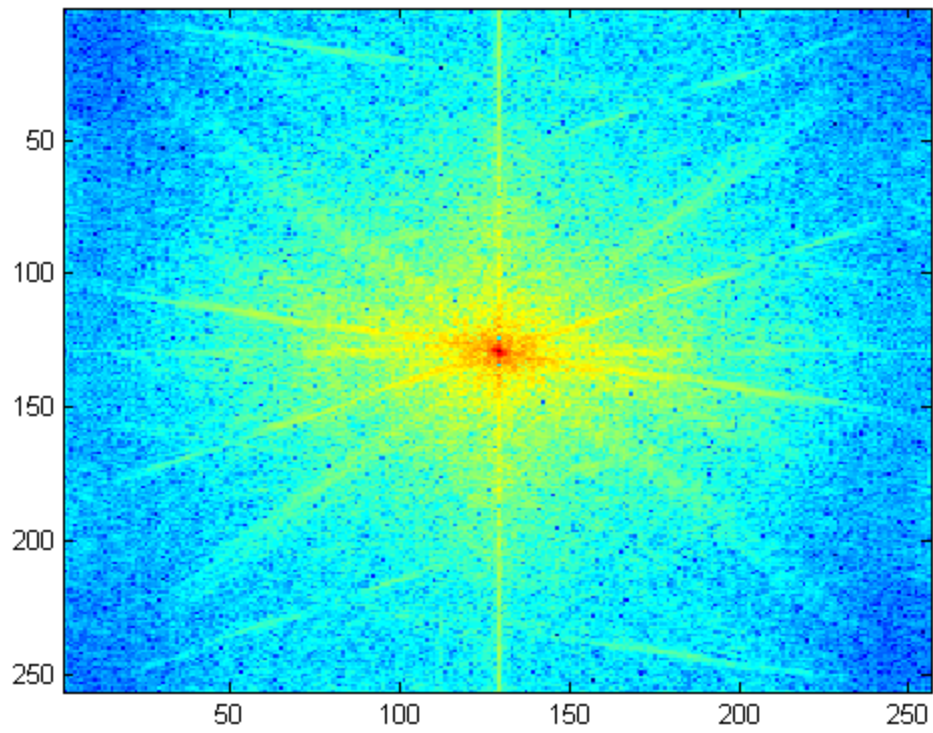
POCS Spectrum



Structure-Adaptive Normalized Convolution(0)



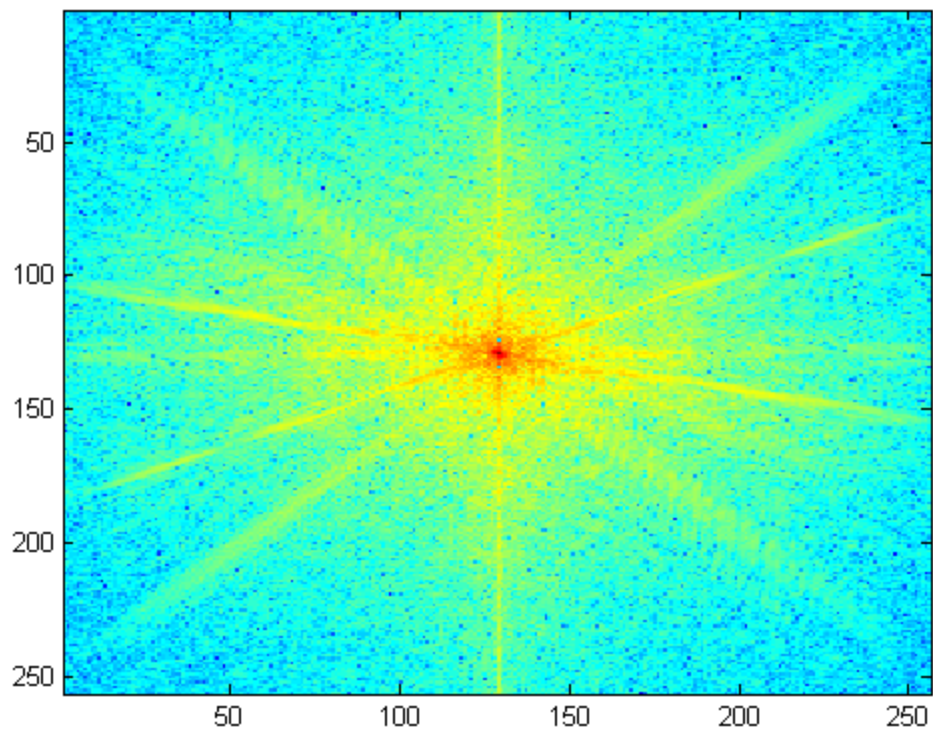
SANC Spectrum



Structure-Adaptive Normalized Convolution(1)



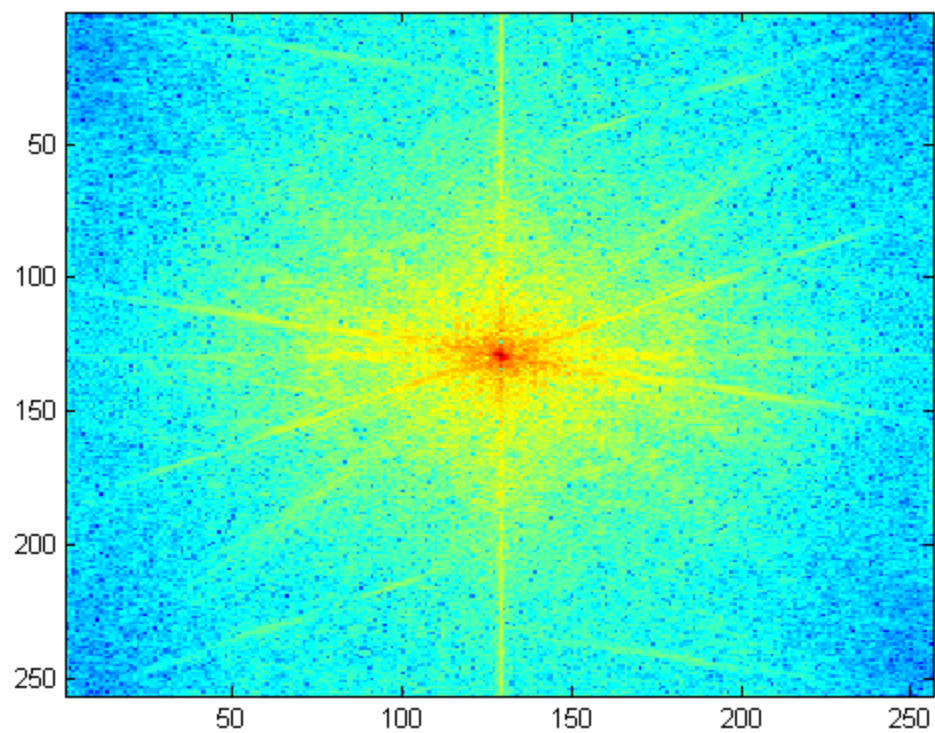
SANC Spectrum



Structure-Adaptive Normalized Convolution(2)



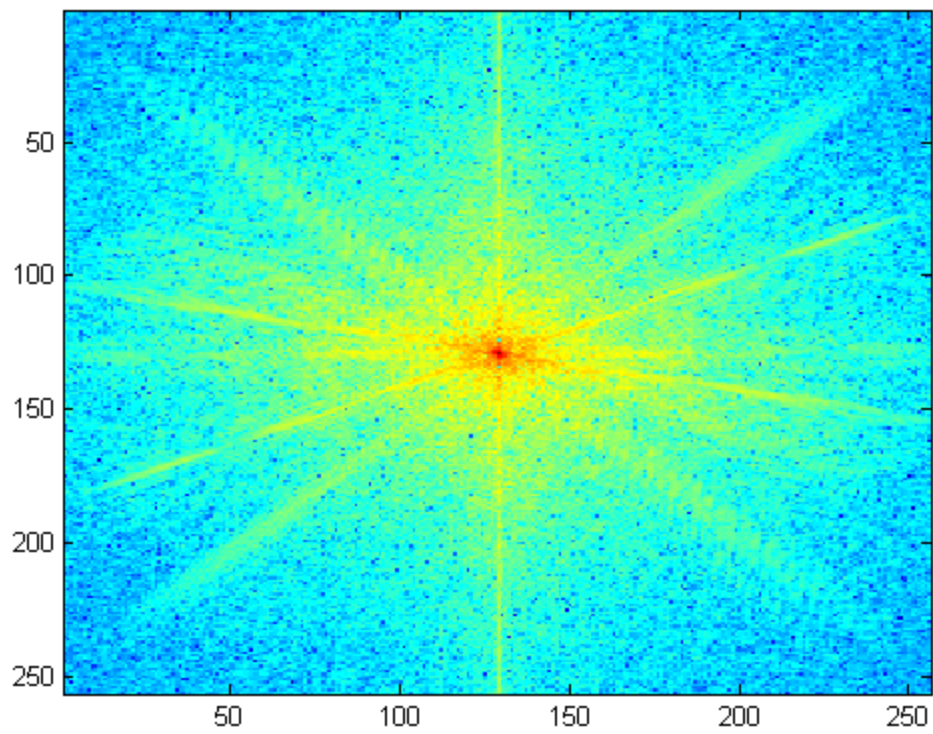
SANC Spectrum



Structure-Adaptive Normalized Convolution(3)



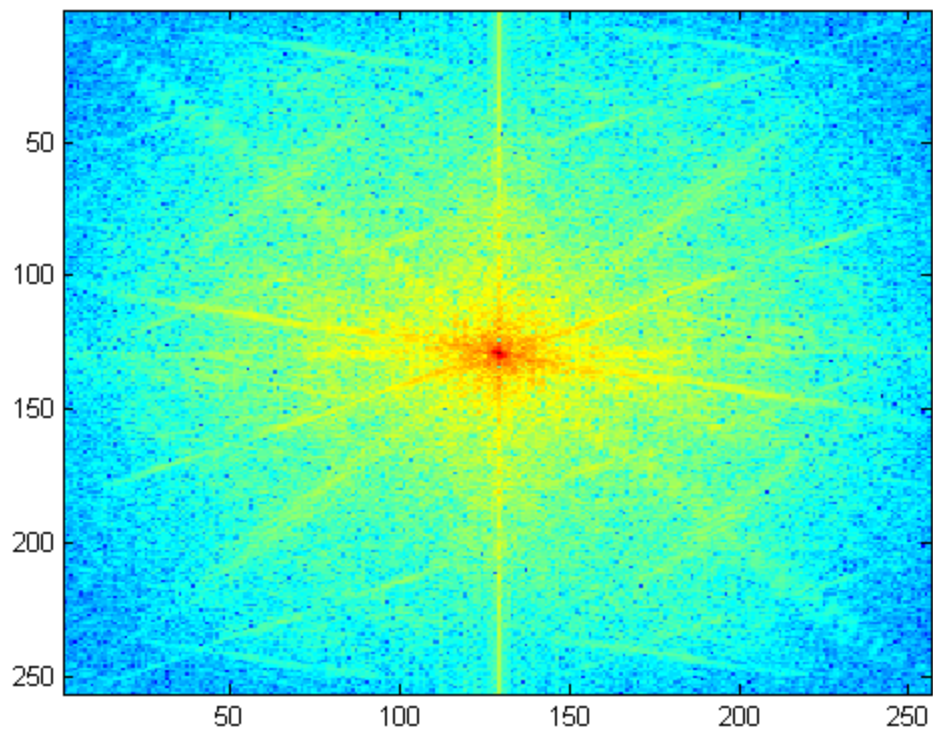
SANC Spectrum



Anisotropic Diffusion Super-resolution



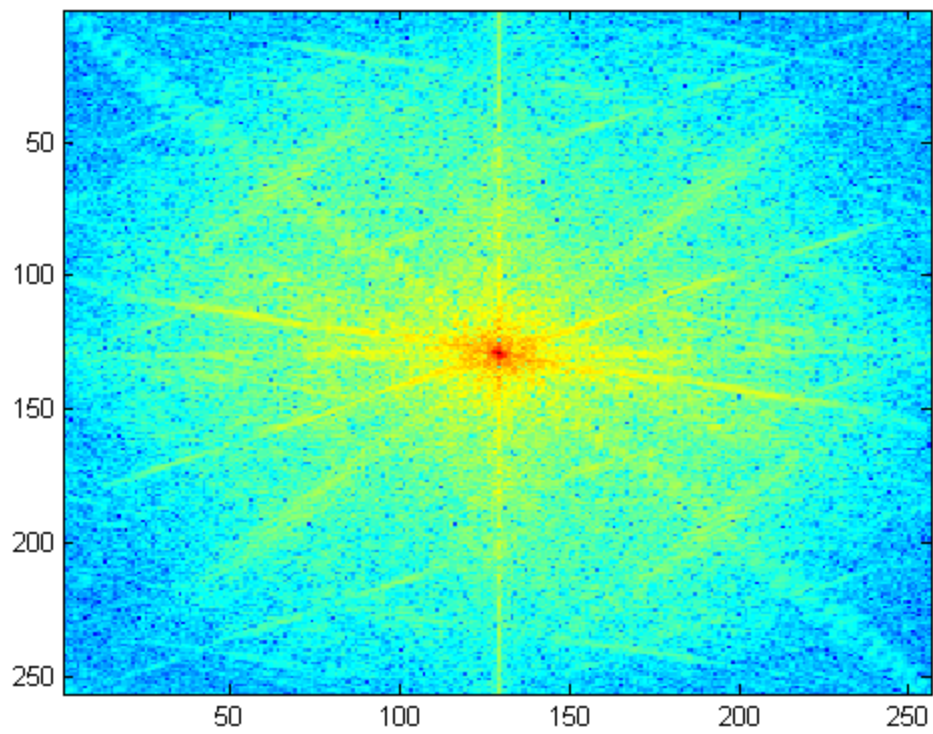
ADSR Spectrum



Variable Exponent SR1



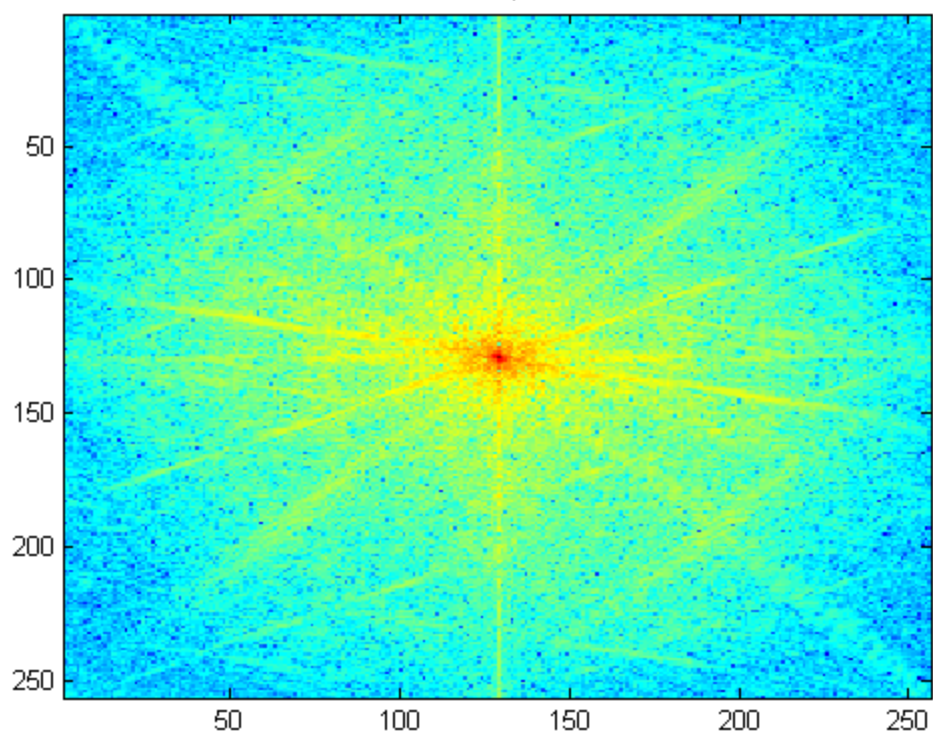
VAREX1 Spectrum



Variable Exponent SR2



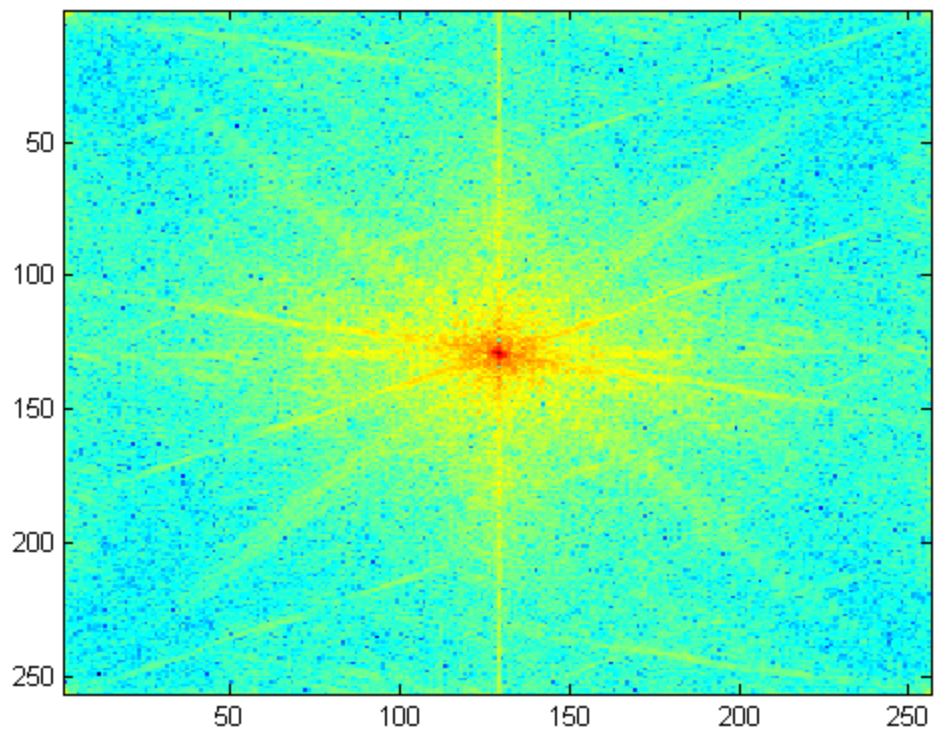
VAREX2 Spectrum



Super-resolution with Details Refinement Capabilities



SR_H FCE Spectrum



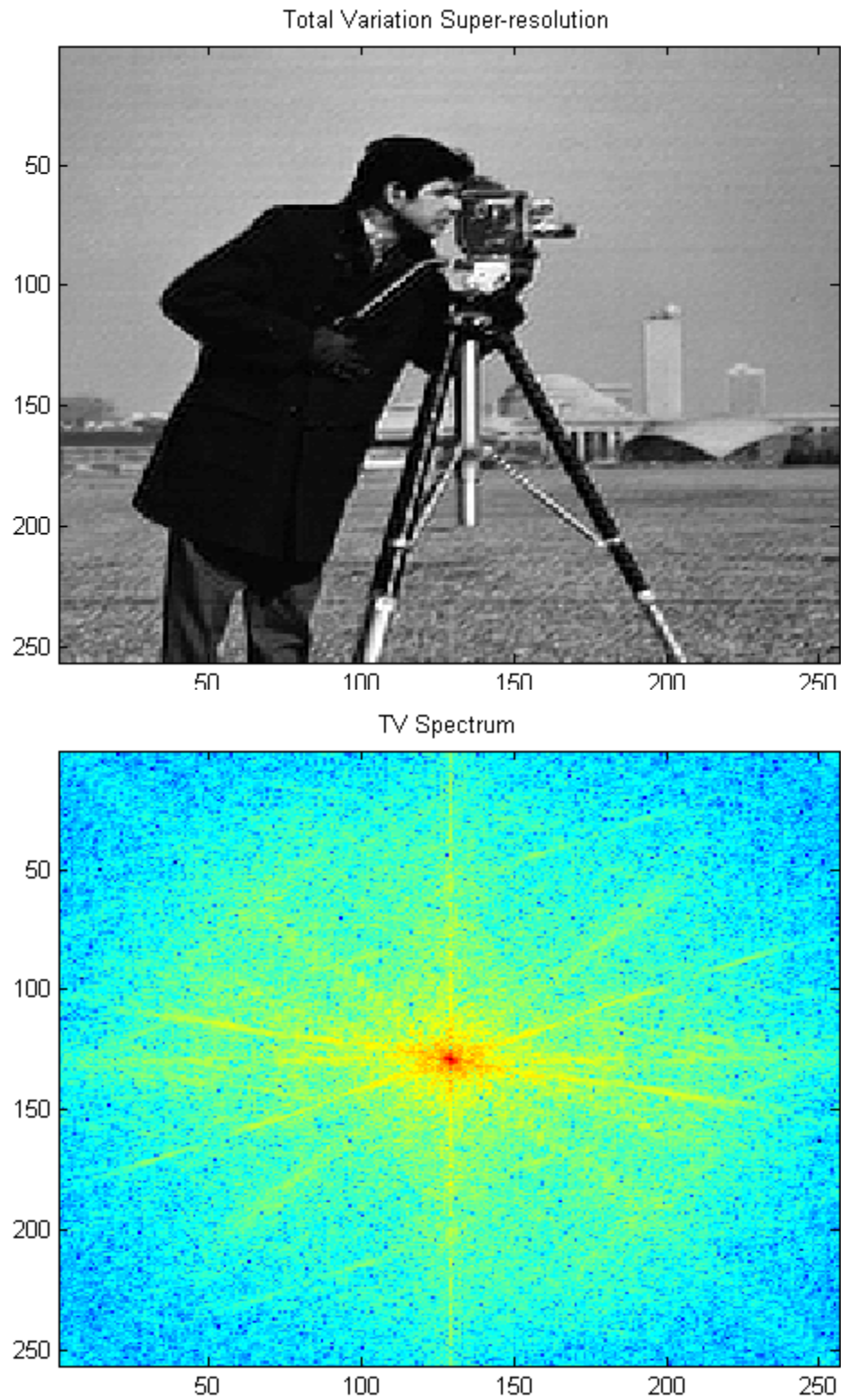


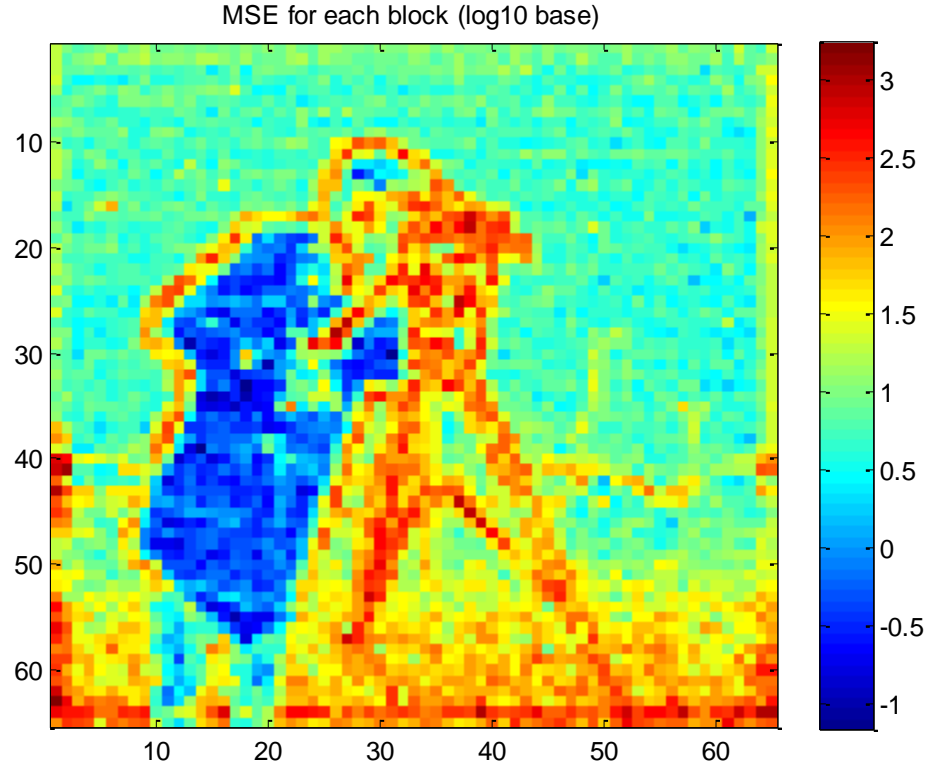
Figure 50 SR Results obtained from MATLAB app using cameraman image. Each titled image is accompanied by its DFT-based spectrum plot with the same log contrast enhancement.

Methods used are: Papoulis-Gerchberg, Iterated Back Projection, Robust Super-Resolution, POCS, Structure-Adaptive Normalized Convolution, Anisotropic Diffusion Super-Resolution, Variable Exponent SR1, Variable Exponent SR2, Super-resolution with Details Refinement Capabilities, and Total Variation Super-resolution. Table 2 show all image quality metrics for all methods applied to the cameraman image.

Table 2 Cameraman image results MSE and SSIM (best result are given in bold font)

SR method	MSE	SSIM
OR Overlap / uint8 OR Overlap (Block size = 4)	41.6237/ 13.4608	.6810/ .9278
OR Overlap with Post- processing Diffusion (Block size = 4)	41.8295/ 13.5718	.6230/ .9319
OR Overlap with Block Post- processing Diffusion (Block size = 4)	44.4169/ 14.4151	.6166/ .9299
OR No Overlap (Block size = 8)	72.5064/ 21.4141	.5406/ .8533
Spatially Adaptive OR Method with overlapping [8]	79.42	NA
Papoulis-Gerchberg	163.8124	.6511
Iterated Back Projection	566.1951	.4423
Robust Super-Resolution	456.7875	.4975
POCS	158.3437	.7139
Structure-Adaptive Normalized Convolution (best result=NC1)	60.2159	.7803
Anisotropic Diffusion Super-Resolution	173.8660	.6239
Variable Exponent SR1	195.5214	.6389
Variable Exponent SR2	208.8404	.6361
Super-resolution with Details Refinement Capabilities	129.0024	.6115
Total Variation Super-resolution	246.8689	.5607

Next, since we are working with blocks or patches, we can also display MSE and SSIM for the 4 by 4 blocks of the HR image of our best result in Table 2, see Fig 51. Largest MSE errors are on the edges and large SSIM errors are visible also in the high-frequency texture regions. Our next set of results correspond to the image Liftingbody, see Fig 53 and Table 3.



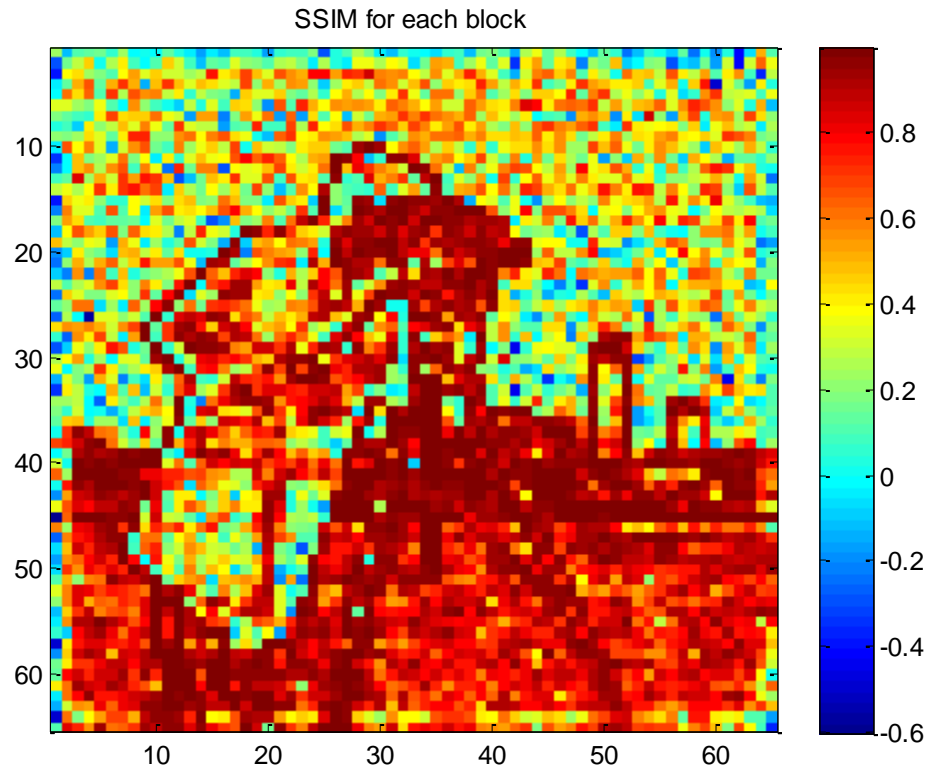


Figure 51 MSE and SSIM results for each block (MSE is using log10 base for display purposes).

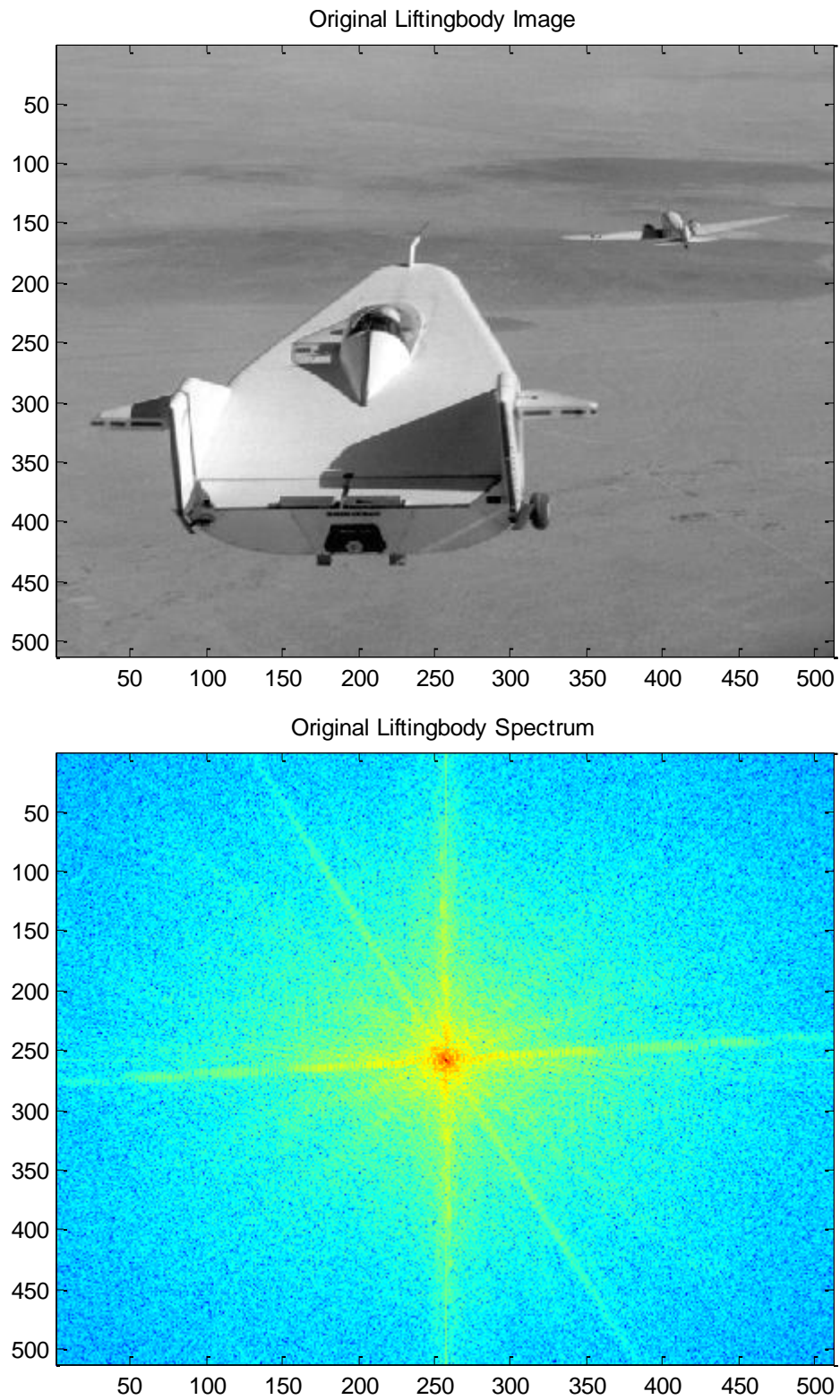
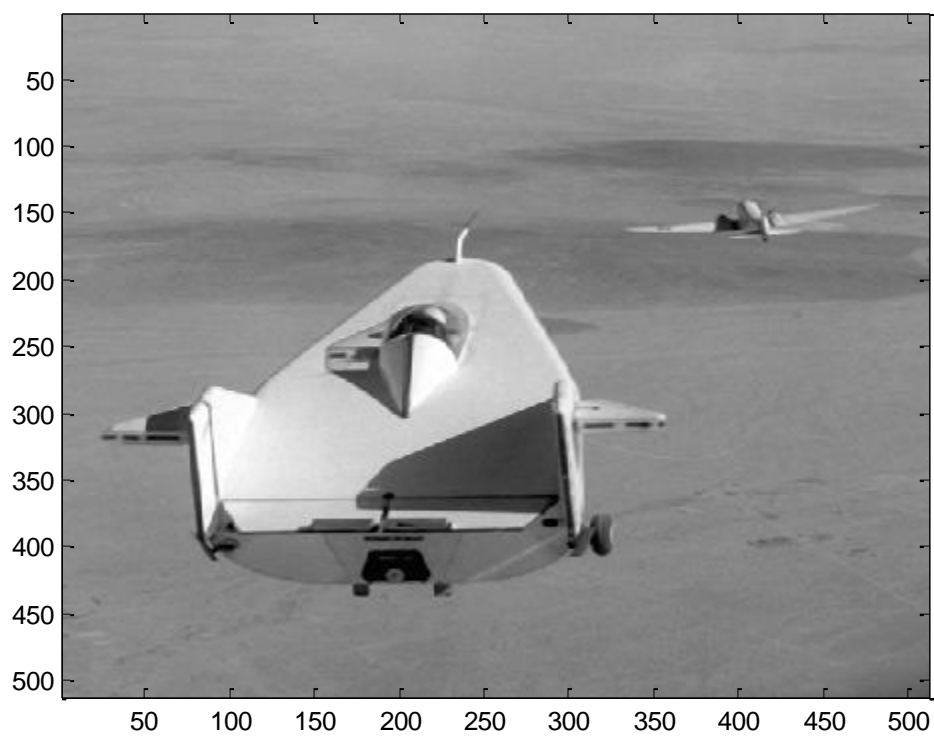
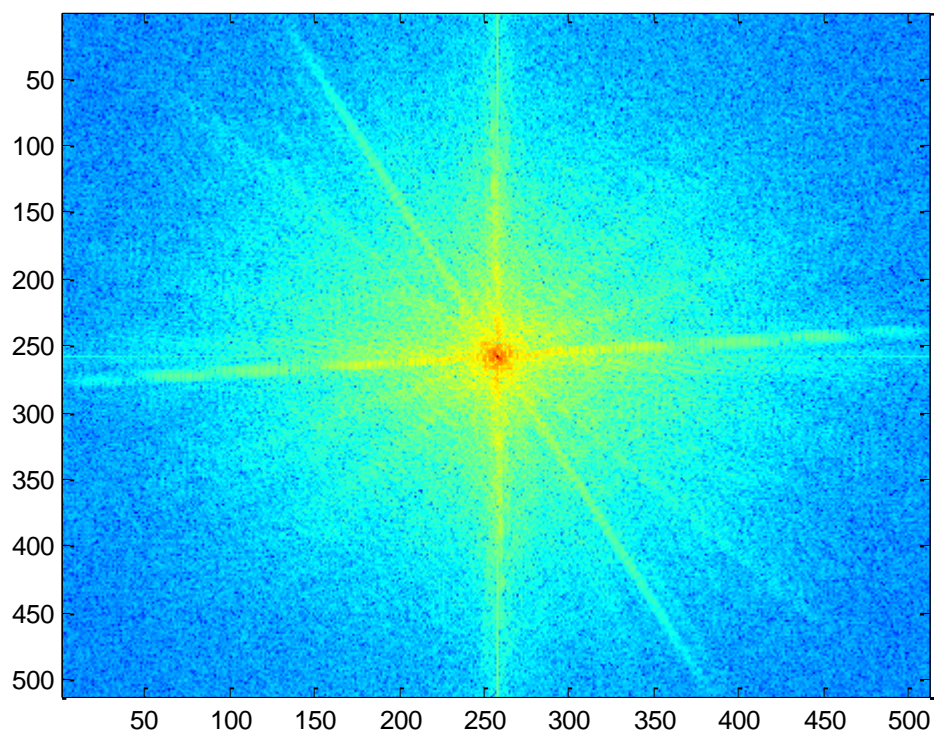


Figure 52 Original Liftingbody and its DFT-based spectrum domain view using log contrast.

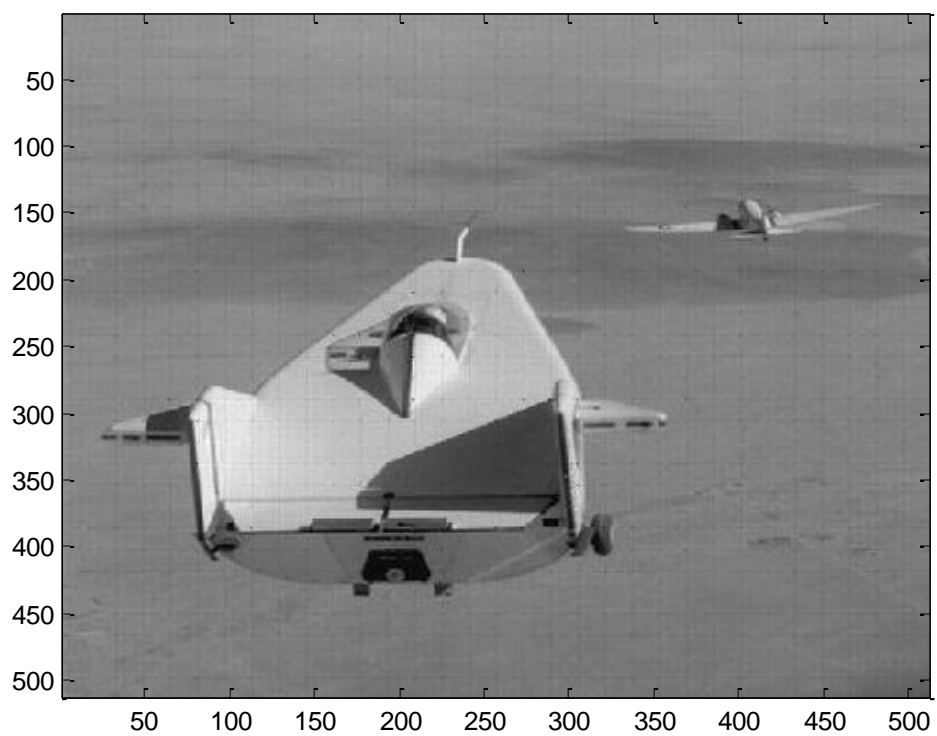
Structure-Adaptive Normalized Convolution(1)



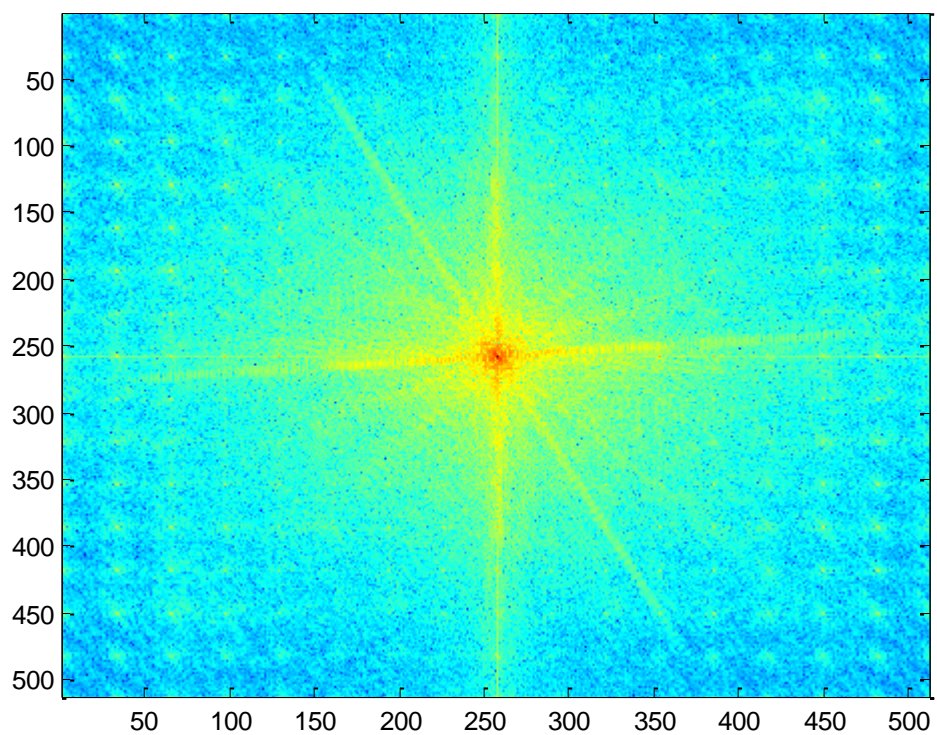
SANC Spectrum



OR Result with No-Overlap



OR Result with No-Overlap



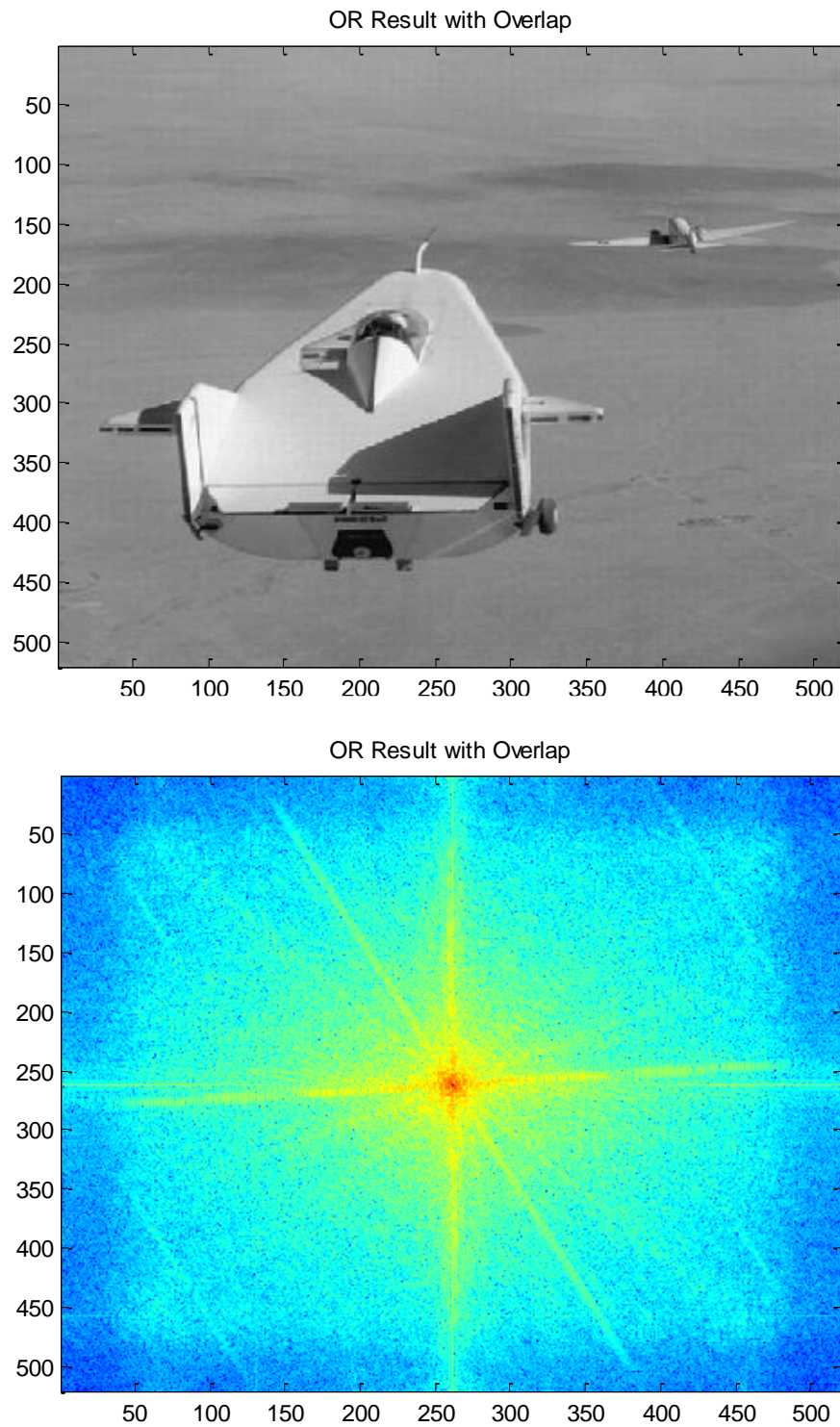


Figure 53 Structure-Adaptive Normalized Convolution vs OR Super-Resolution with Overlap and No-Overlap for Liftingbody image

Table 3 Liftingbody image results MSE and SSIM (best results are shown in bold font)

SR method	MSE	SSIM
OR Overlap / uint8 OR Overlap (Block size = 8)	6.0755/ 4.4692	.4918/ .9365
OR Overlap with Post- processing Diffusion (Block size = 8)	5.9045 / 4.9623	.4618/ .9279
OR Overlap with Block Post- processing Diffusion (Block size = 8)	6.0054/ 5.0973	.4587/ .9274
OR No Overlap (Block size = 8)	19.3400/ 11.5481	.3860/ .8528
Papoulis-Gerchberg	14.5476	.7200
Iterated Back Projection	51.3525	.4955
Robust Super-Resolution	45.0273	.5390
POCS	17.7919	.7350
Structure-Adaptive Normalized Convolution (best result=NC1 and NC3)	4.9798	.7934
Anisotropic Diffusion Super-Resolution	16.0670	.6456
Variable Exponent SR1	19.1141	.6465
Variable Exponent SR2	20.6605	.6428
Super-resolution with Details Refinement Capabilities	28.7643	.5558
Total Variation Super-resolution	35.2118	.5522

Note: MSE and SSIM results obtained in the SR MATLAB app were obtained from reconstructed images converted to unsigned 8-bit integers by clipping values outside of the range [0,255]. To be fairer with previous work [8], we also displayed the double precision result before clipping the images to demonstrate improvements over previous work. Also, as we see the differences in MSE in Table 2 and Table 3, smoother images are easier to recover from LR images using any SR algorithm.

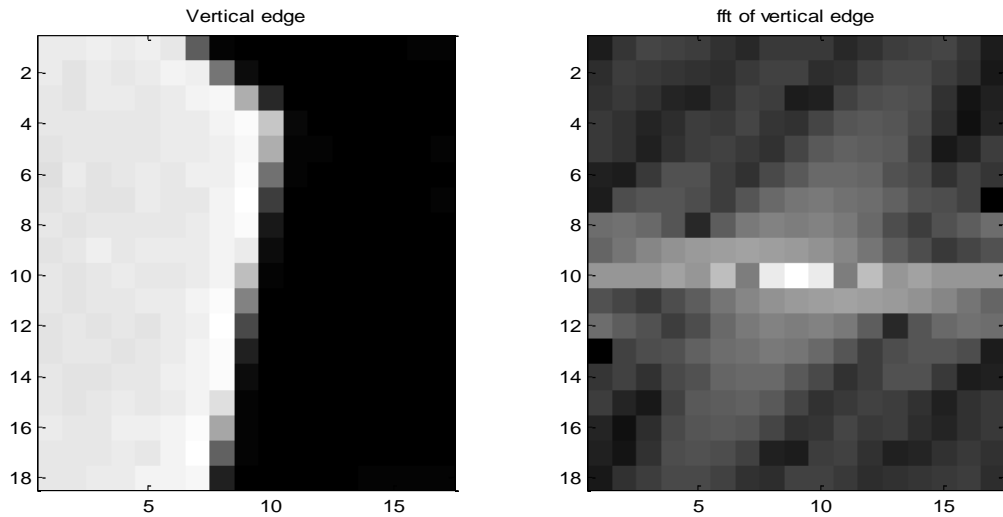
Chapter 8 Conclusion and Future Work

In conclusion, improvements over previous OR based SR methods [8] were: remove anti-aliasing filter to generate LR images so we can recover HR frequencies, implement a new shift registration for LR images, improve selection of parameters and block classes according to the input image, bypass the use of GCV for computational efficiency, implement a variable block size selection, allow different input image sizes, apply code debugging techniques to make the code more efficient in time, and solve image boundaries problems. Furthermore, we extend this approach by obtaining estimated gradients and binary edge maps by using the OR theory framework itself, and integrate an anisotropic diffusion method as global post-processing or local, patch-based post-processing for edge enhancement. Here anisotropic diffusion was not able to improve our results considerably in terms of our metrics, however, by visual comparison, the results improved. Finally, we were able to reduce the MSE from 79.42 to 41.6237 and achieved an SSIM of .6810 before clipping to 8-bit integers. Also, we compare our algorithm against others state-of-the-art SR methods with low complexity available in a MATLAB app [12]. The main issue with this MATLAB app is that the output images are unsigned 8-bit integers. To get a fair comparisons against these algorithms we had to convert our double precision results to unsigned integer of 8 bits as well by clipping values outside of the range [0,255]. This way, our best results were an MSE of 13.4608 and a SSIM of 0.9319 for our cameraman.

As future work we suggest to automate the selection of the number of block classes for each image. Right now we are using only 5 fixed different classes for the K-means and 4 values for multi-thresholding, however some images may need more or less classes depending on their image details (texture, smoothness, objects structure, etc) so the OR super-resolution can

maximizes its efficiency. Also, due the use of only 5 fixed classes, we can only implement the class approach on 8x8 LR block size for the cameraman image. We found that different block sizes will show wrong class selection when we display a histogram similar to figure 23. Changing block sizes will increase or decrease their variance such that they may require more or less classes to avoid this issue.

In addition, in this work we only worked with square bandwidth 2D regions using two different sinc functions with the same bandwidth parameter a . Considering the Fourier Transform of an image, vertical edges need higher bandwidth in the horizontal orientation while horizontal edges needs higher bandwidth in the vertical orientation, see Fig 54; diagonals edges require a rotated spectral support. Therefore we can modify our sinc functions using different bandwidth parameter for each sinc to obtain a rectangular bandwidth regions or use other shapes. This will improve how we process our edge blocks and potentially reduce our final MSE results at the same time.



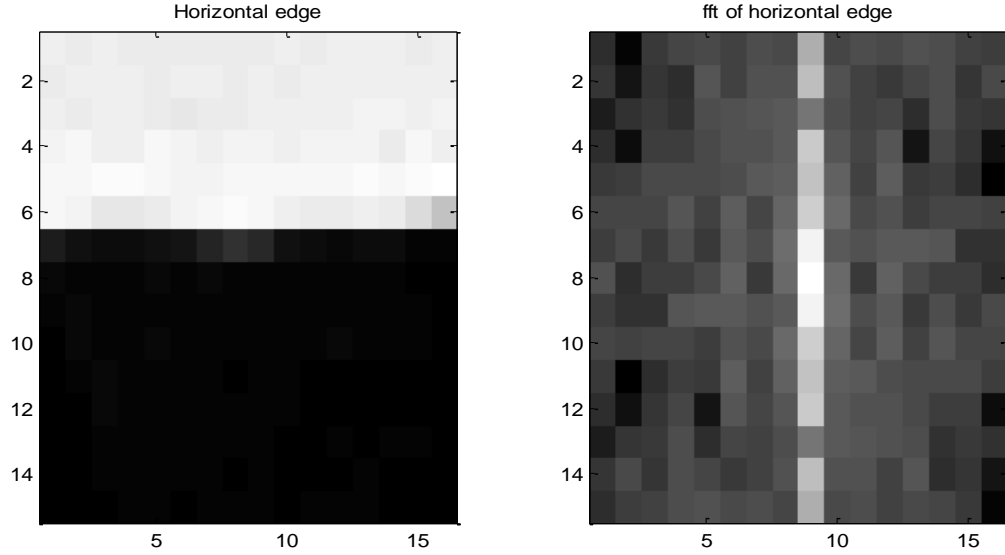


Figure 54 bandwidth of vertical and horizontal edges in an image

Moreover as a future work, we propose to revive the GCV method to obtain one regularization parameter for each class and not for each block as done before. This could help us to select the best lambda possible for each bandwidth independently of the input image data. Alternatively, the block that is representative for each class could be used to generate the lambda needed for that specific class, such that all blocks in the same class would use the same lambda.

Also, due to its processing speed, we have interest in implementing our OR super-resolution method in video metrology applications. Motion estimation in video processing may be considered for those cases to obtain registration parameters. We also need to consider noise in our SR algorithm as well as work with real images where aliasing is not present on the LR images, a more difficult case since the high-frequency information is lost rather than aliased. Finally, the SR-OR should be evaluated on color images using standard approached to process color images using methods developed for gray scale images.

References

- [1] P. Milanfar Ed., Super-Resolution Imaging, CRC Press, Boca Raton, FL, 2011.
- [2] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey." *Machine vision and applications* 25.6 (2014): 1423-1468.
- [3] Jiji, C.V., Chaudhuri, S., Chatterjee, P.: Single frame image super-resolution: should we process locally or globally? *Multidimens. Syst. Signal Process.* 18, 123–152 (2007)
- [4] Lee, I-Hsien, Nirmal K. Bose, and Chih-Wei Lin. "Locally adaptive regularized superresolution on video with arbitrary motion." *Image Processing (ICIP), 2010 17th IEEE International Conference on.* IEEE, 2010.
- [5] Liang, Zifei, et al. "Combining coarse and fine registration for video frame superresolution reconstruction." *Journal of Electronic Imaging* 23.6 (2014): 063018-063018.
- [6] Guo, Feng, and Rama Chellappa. "Video metrology using a single camera." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.7 (2010): 1329-1335.
- [7] Mora-González, Miguel, et al. *Image Processing for Optical Metrology Chapter 25 in MATLAB—A Ubiquitous Tool for the Practical Engineer*, INTECH Open Access, 2011.
- [8] A. J. Shaik and S. D. Cabrera, "Spatially Adaptive Superresolution using the Optimal Recovery Framework," in *Proc. of IEEE Southwest Symposium on Image Analysis and Interpretation*, Santa Fe, NM., April 2012.
- [9] D. Muresan and T. W. Parks, "Demosaiicing Using Optimal Recovery," *IEEE Transactions on Image Processing*, Vol. 14, No. 2, pp 267-278, 2005.
- [10] C. A. Micchelli and T. J. Rivlin, "A Survey of Optimal Recovery," in *Optimal Estimation in Approximation Theory*, Plenum Press, pp.1-54, 1977.
- [11] S. D. Cabrera, V. Moram and J. G. Rosiles, "Sensor deployment optimization based on optimal recovery interpolation," in *Proc. Of Wireless Sensing, Localization, and Processing VI*, SPIE Vol. 8061, PP. 806107-1 to 12 Orlando FL, April 2011
- [12] SuperresolutionDemo - File Exchange - MATLAB Central, Baraka Maiseli Baraka Maiseli (view profile) 11 files 63 downloads 4.5 - <https://www.mathworks.com/matlabcentral/fileexchange/49538-superresolutiondemo>
- [13] Peyman Milanfar <https://users.soe.ucsc.edu/~milanfar/software/superresolution.html>
- [14] H. Takeda and P. Milanfar, "Locally Adaptive Kernel Regression for Space-Time Super-Resolution", Chapter in *Super-resolution Imaging* Edited by P. Milanfar, September 27, 2010.
- [15] Optimizing and Accelerating your MATLAB Code https://www.mathworks.com/videos/optimizing-and-accelerating-your-matlab-code-107711.html?s_v1=7489&elq_cid=9301019&form_seq=conf714&elqsid=1442323477367&potential_use=Commercial&country_code=US&elq=3c6715f17d6b4f5283d33e925a92c9b7&

lqCampaignId=2372&elqaid=7489&elqat=1&elqTrackId=522b6d2d5803421088bba5893f5c1f10

- [16] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing 3rd edition." (2008).
- [17] Lim, J. S. (1990). Two-dimensional signal and image processing. Englewood Cliffs, NJ, Prentice Hall, 1990.
- [18] Haralick, Robert M., and Linda G. Shapiro, Computer and Robot Vision, Volume I, Addison-Wesley, 1992, pp. 28-48.
- [19] A. A. Melkman and C. A. Michelli, "Optimal Estimation of Linear Operators in Hilbert Spaces from Inaccurate Data," SIAM J. on Numerical Analysis, Vol. 16, pp. 87-105, Feb. 1979.
- [20] Maiseli, Baraka. "Diffusion-steered super-resolution method based on the Papoulis–Gerchberg algorithm." IET Image Processing 10.10 (2016): 683-692.
- [21] Maiseli, B., Wu, C., Mei, J., Liu, Q., & Gao, H. (2014). A robust super-resolution method with improved high-frequency components estimation and aliasing correction capabilities. Journal of the Franklin Institute, 351(1), 513-527.
- [22] Weickert, Joachim. Anisotropic diffusion in image processing. Vol. 1. Stuttgart: Teubner, 1998.
- [23] S. Villena, M. Vega, S.D. Babacan, R. Molina, A.K. Katsaggelos Bayesian combination of sparse and non-sparse priors in image super-resolution, Digit. Signal Process., 23 (2) (2013), pp. 530–541
- [24] S. Villena, M. Vega, R. Molina, A.K. Katsaggelos, Bayesian super-resolution image reconstruction using an ℓ_1 prior, in: 6th ISPA Conference, Salzburg, Austria, 2009, pp. 152–157.
- [25] S. Farsiu, M. D. Robinson, M. Elad, P. Milanfar, "Fast and robust multiframe super-resolution", IEEE Trans. Image Process., vol. 13, no. 10, pp. 1327-1344, Oct. 2004.
- [26] S. Villena, M. Vega, D. Babacan, R. Molina, A. Katsaggelos, Using the Kullback–Leibler divergence to combine image priors in super-resolution image reconstruction, in: IEEE International Conference on Image Processing, Hong Kong, China, 2010, pp. 893–896.
- [27] S.D. Babacan, R. Molina, A. Katsaggelos, Variational Bayesian super-resolution, IEEE Trans. Image Process., 20 (4) (2011), pp. 984-999
- [28] A. Zomet, A. Rav-Acha, S. Peleg, Robust super-resolution, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), 2001, pp. 645–650.
- [29] L. G. Brown. A survey of image registration techniques. Computing Surveys, 24(4):325-376, 1992.
- [30] S. Chaudhuri, Super-Resolution Imaging, MA, Norwell:Kluwer, 2001.

- [31] Y. J. Kim, J. H. Park, G. S. Shin, H.-S. Lee, D.-H. Kim, S. H. Park, and J. Kim, "Evaluating super-resolution algorithms", in Proc. of the SPIE-IS&T Electronic Imaging, Image Quality and System Performance VIII, vol. 7867, pp. 78670D-78670D-7, 2011.
- [32] A. J. Shah, S. B. Gupta, "Image super-resolution—A survey", Proc. 1st Int. Conf. Emerg. Technol. Trends Electron. Commun. Netw., pp. 1-6, Dec. 2012.
- [33] Tuan Q. Pham et al.; Super-resolution Fusion using Adaptive Normalized Averaging; 10th Annual Conference of the Advanced School for Computing and Imaging (ASCI); Jun. 2004, XP 002539619; pp. 107-114. [14] Gandam, A., Sidhu, J.S.: A Post-Processing Algorithm for Detection & Removal of Corner Outlier. International Journal of Computer Applications 4(2) (2010) ISSN:09758887
- [34] Farsiu, S., Robinson, D., Elad, M., & Milanfar, P. (2004). Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2), 47-57.
- [35] Huang, Y., & Long, Y. (2006, September). Super-resolution using neural networks based on the optimal recovery theory. In Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on (pp. 465-470). IEEE.
- [36] Dong, W., Shi, G., Zhang, L., & Wu, X. (2010, July). Super-resolution with nonlocal regularized sparse representation. In Visual Communications and Image Processing 2010 (pp. 77440H-77440H). International Society for Optics and Photonics.
- [37] Vandewalle, P., Susstrunk, S. E., & Vetterli, M. (2003, June). Superresolution images reconstructed from aliased images. In *Visual Communications and Image Processing 2003* (pp. 1398-1405). International Society for Optics and Photonics.
- [38] Salari, E., & Bao, G. (2012). Super-resolution using an enhanced Papoulis–Gerchberg algorithm. *IET image processing*, 6(7), 959-965.
- [39] Maiseli, B. J., Elisha, O. A., & Gao, H. (2015). A multi-frame super-resolution method based on the variable-exponent nonlinear diffusion regularizer. *EURASIP Journal on Image and Video Processing*, 2015(1), 22.
- [40] Chatterjee, P., Mukherjee, S., Chaudhuri, S., & Seetharaman, G. (2009). Application of Papoulis–Gerchberg method in image super-resolution and inpainting. *The computer journal*, 52(1), 80-89.
- [41] Pal, C., Chakrabarti, A., & Ghosh, R. (2015). A brief survey of recent edge-preserving smoothing algorithms on digital images. *arXiv preprint arXiv:1503.07297*.
- [42] Zhou, Q., Chen, S., Liu, J., & Tang, X. (2011, November). Edge-preserving single image super-resolution. In *Proceedings of the 19th ACM international conference on Multimedia* (pp. 1037-1040). ACM.
- [43] Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on* (pp. 839-846). IEEE.

- [44] Maiseli, B. (2016). Diffusion-steered super-resolution method based on the Papoulis–Gerchberg algorithm. *IET Image Processing*, 10(10), 683-692.
- [45] Chaudhury, K. N., & Dabhade, S. D. (2016). Fast and provably accurate bilateral filtering. *IEEE Transactions on Image Processing*, 25(6), 2519-2528.
- [46] Sanchez, D., Baquero, M., Gonzalez, D., & Alfonso, E. (2007). Improvement of resolution in equivalent currents reconstruction using Papoulis-Gerchberg algorithm and replicas of the spectrum. *Electronics Letters*, 43(19), 1010-1012.
- [47] Li, X., Hu, Y., Gao, X., Tao, D., & Ning, B. (2010). A multi-frame image super-resolution method. *Signal Processing*, 90(2), 405-414.
- [48] Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7), 629-639.
- [49] Park, S. C., Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3), 21-36.
- [50] Nasrollahi, K., & Moeslund, T. B. (2014). Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6), 1423-1468.
- [51] Barash, D. (2001, July). Bilateral filtering and anisotropic diffusion: Towards a unified viewpoint. In *International Conference on Scale-Space Theories in Computer Vision* (pp. 273-280). Springer Berlin Heidelberg.
- [52] Carreon, J. A., & Cabrera, S. D. (2013, May). Towards the use of learned dictionaries and compressive sensing in wideband signal detection. In *SPIE Defense, Security, and Sensing* (pp. 87170N-87170N). International Society for Optics and Photonics.
- [53] Guo, Z., Sun, J., Zhang, D., & Wu, B. (2012). Adaptive Perona–Malik model based on the variable exponent for image denoising. *IEEE Transactions on Image Processing*, 21(3), 958-967.
- [54] Zhu, H., Lu, Y., & Wu, Q. (2007, September). Super-resolution image restoration by maximum likelihood method and edge-oriented diffusion. In *International Symposium on Photoelectronic Detection and Imaging: Technology and Applications 2007* (pp. 66250Y-66250Y). International Society for Optics and Photonics.
- [55] Laghrib, A., Ghazdali, A., Hakim, A., & Raghay, S. (2016). A multi-frame super-resolution using diffusion registration and a nonlocal variational image restoration. *Computers & Mathematics with Applications*, 72(9), 2535-2548.
- [56] Sekuboyina, A. K., & Seelamantula, C. S. (2016, March). An efficient formulation and parameter selection for multiple image super-resolution. In *Communication (NCC), 2016 Twenty Second National Conference on* (pp. 1-6). IEEE.
- [57] Muresan, D. D., & Parks, T. W. (2005). Demosaicing using optimal recovery. *IEEE Transactions on Image Processing*, 14(2), 267-278.

- [58] Mora-González, M., Muñoz-Maciel, J., Casillas, F. J., Peña-Lecona, F. G., Chiu-Zarate, R., & de Guevara, H. P. L. (2011). Image Processing for Optical Metrology. *Chap, 25*, 523-546.
- [59] Muresan, D. D. (2002). Review of optimal recovery. *Rapport Technique TR-2002-10, DMMD*.
- [60] Moram, V., & Cabrera, S. D. (2011, January). Superresolution using the optimal recovery framework with automatic Generalized Cross-Validation. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE* (pp. 344-349). IEEE.
- [61] Li, J., Wang, L., & Bao, P. (2009, August). An industrial CT image adaptive filtering method based on anisotropic diffusion. In *Mechatronics and Automation, 2009. ICMA 2009. International Conference on* (pp. 1009-1014). IEEE.
- [62] Mansfield, L. E. (1972). Optimal approximation and error bounds in spaces of bivariate functions. *Journal of Approximation Theory*, 5(1), 77-96.
- [63] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.
- [64] Hore, A., & Ziou, D. (2010, August). Image quality metrics: PSNR vs. SSIM. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (pp. 2366-2369). IEEE.
- [65] Sapiro, G. (2006). Geometric partial differential equations and image analysis. Cambridge university press.
- [66] Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., & Lenzen, F. (2009). Variational methods in imaging (Vol. 320). New York: Springer.
- [67] Aubert, G., & Kornprobst, P. (2006). Mathematical problems in image processing: partial differential equations and the calculus of variations (Vol. 147). Springer Science & Business Media.
- [68] Digital image processing: p054 - Anisotropic Diffusion Alfred936 - https://www.youtube.com/watch?v=B_TiVX7zN8U&index=55&list=PLZ9qNFMHZ-A79y1StvUUqgyL-O0fZh2rs
- [69] Zhu, Wenjing, et al. "Image deblocking using convex optimization with modified total variation method." Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE. IEEE, 2013.
- [70] Sun, J., Sun, J., Xu, Z., & Shum, H. Y. (2008, June). Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1-8). IEEE.

Curriculum Vitae

Luis Ponce was born in April 2nd, 1993 in Chihuahua, Mexico. He finished his bachelor degree in Electrical and Computer Engineer in May 2016 from University of Texas at El Paso (UTEP). His area concentration include Digital Signal Processing and Image Processing. While at UTEP, he worked as math tutor at MARCS, research assistant at COURI under the program MERITUS with Dr. Cabrera, and Teacher Assistant for the class of microprocessors systems 1 with Dr. Romero. He graduate with Masters in May 2017. He has presented research posters at the conference of SIAM Imaging Science in Albuquerque May 2016 with topic of Image Super-Resolution, and UTEP symposium for COURI research in May 2015 and 2016 with topics of Image Inpainting and Image Super-Resolution respectively. His future plans consist on write a research paper for this thesis, get a job in the industry in the field of image processing and apply for PhD program in Electrical Engineering.

Permanent address: Arboleda de los Cipresses #8951-63

Chihuahua, Chihuahua, C.P. 32543

This thesis/dissertation was typed by Luis A. Ponce