Steering Kernel Regression via Laplacian for Image Denoising in Spatial Domain

Reza Salami

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
February 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

Prof. Dr. Aykut Hocanın
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion, it is fully adequate, in scope and quality, as a thesis of the degree of Master of Science in Electrical and Electronic Engineering.

Prof. Dr. Hüseyin Özkaramanli
Supervisor

Examining Committee

1. Prof. Dr. Hüseyin Özkaramanli
2. Prof. Dr. Şener Uysal
3. Assoc. Prof. Dr. Hasan Demirel

ABSTRACT

Recently digital imaging devices are used in many applications and they often suffer from some degradation, such as noise, blurring, aliasing effects, and more due to environment limitations. Captured images are mostly not of favorable quality and need to be enhanced by software. One of the significant reasons of the performance degradations for most methods is the presence of noise. Noise removal, therefore, is one of the most important tools for many applications. In this thesis, we focus on this issue as one of the main important problem of image processing.

We discuss about the various sources of noise corrupting image and illustrate the statistical behavior of noise and discuss about how to eliminate the effects of the noise from our images.

The classic kernel regression (KR) is a statistical framework that enables us to regard a variety of image restoration problems as regression, and it has a few beneficial properties instead of other regression methods. We have modified the classic kernel regression (KR) with steering matrices which are estimated by the singular value decomposition of the second derivatives of pixels that we apply, which makes our method relying not only the spatial properties (the sample location and density), but also the photometric properties of these samples (i.e., pixel value). Thus, the effective size and shape of the regression kernel are adapted locally to the underlying image structure.

Steering kernel regression (SKR) method has been shown to provide excellent denoising result. Steering kernels adapt to the local pixel intensity statistics and geometry. SKR employs the gradient for finding the structure of local region by applying the first order structure tensor, but in this work we propose to use an adaptive kernel, which is based on the second derivative of pixels and find the structure tensor of hessian matrix (which related to the second derivative) for each pixel to make the structure tensor more robust in the face of noise to maintain the image details. Since edges in an image have significant profile, we apply second derivative because gradient produces thick edges while second order derivative (Laplacian) produces finer edges also magnitude of gradient can be used to detect presence of edge at point, but sign of second derivative can be used to determine whether edge pixel itself lies on the dark or bright side of edges. The motivation behind structure tensor is a fact that image contains directional structure such as edges and the motivation behind second derivative is to obtain finer edges.

Quantitative and perceptual evaluations from simulations have been shown that proposed framework indicates an average PSNR improvement compared to other framework, and compared with the conventional SKR method an average 0.3 dB PSNR increase is obtained. Comparisons show the superiority of this method over other descriptors.

Keywords: Denoising, kernel function, kernel regression, Steering matrix, Taylor series, structure tensor.

ÖZ

Görüntü alma aygıtları birçok uygulamada sıkça kullanılmaktadır. Bu aygıtlar aracılığı ile elde edilen gürültü, görüntüler, bulandırma, frekans örtüşmesi ve daha birçok etken tarafından olumsuz yönde etkilenmektedirler. Bu nedenle elde edilen görüntülerin tasarlanacak algoritmalar tarafından iyileştirilmesi gerekmektedir. En önemli olumsuz etken ise, gürültüdür.

Gürültünün temizlenmesi çoğu uygulama için büyük önem arzetmektedir. Bu tezde görüntülerde gürültü temizleme konusu incelenecektir. Görüntüleri kötüleştiren değişik gürültü çeşitleri tartışıldıktan sonra klasik temizleme yöntemleri sunulacaktır.

Istatistiksel bir çerçeve olan klasik çekirdek regresyon birçok görüntü işleme uygulaması için önemli bir enstrümandır. Bu tezde klasik çekirdek regresyon yöntemini yönlü çekirdek regresyon için uyarlanmıştır. Yön matrisleri ikinci türev kullanılarak tekil değer çözümlemesi ile hesaplanmıştır. Önerilen yöntem hem piksellerin hem koordinatlarını hemde şiddetlerini dikkate almaktadır. Bu şekilde regresyon çekirdeği yerel olarak görüntünün yapısına uyarlanmaktadır. Yönlü çekirdek regresyon yönteminin gürültü temizlemede çok başarılı sonuçlar verdiği gösterilmiştir.

Yön çekirdekleri pixel geometri ve değerlerine uyarlanmıştır. Klasik yönlü çekirdek regresyonun birinci türevi kullanmaktadır. Bu çalışmada ise piksellerin ikinci türevi kullanılmış ve Hession matrisi aracılığı ile pikselin yapısal niteliği gürültüye karşı daha sağlam hale getirilmiştir.Bu yöntemle detay bilgiler, kenarlar daha iyi ifade edilmiştir. Kenar bölgelerde birinci türev daha kalın bir profil yaratmakta, ikinci türev ise daha ince bir profil vermektedir. Birinci türevinbüyüklüğü kenar bölgesi olup olmadığını ifade etmektedir. İkinci türevin işareti kenar pikselinin parlak veya koyu bölgede olup olmadığını da vermektedir.

Sayısal kıyaslamalar önerilen yöntemin klasik yönlü çekirdek regresyona kıyasla ortalama 0.3 dB iyileştirme sağladığı yönündedir. Görsel kıyaslamalar da iyileştirmeyi doğrular niteliktedir.

Anahtar kelimeler: Gürültünün temizlenmesi, çekirdek regresyon, gergi yapısı, taylor serısi, yönledırme matrısi.

Dedicated to my loving family.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my supervisor Prof. Dr. Huseyin Özkaramanli for his help and guidance. It was a pleasure and honor for me to work with him. I needed his support, advice and encouragement during the thesis period. I am grateful to him for his moral support through all the stages during this research work.

I am thankful to Prof. Dr. Aykut Hocanın, Head, Department of Electrical and Electronic engineering who provided all the official facilities to me. I am also thankful to other committee members.

I take this opportunity to express my regards and respect to my parents whose support and encouragement I can never forget in my life.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

$\mathrm{c_i}$	Covariance Matrix
dB	Decibel
$\mathcal{H}$	Hessian Operators
$\mathcal{K}_{\mathrm{adapt}}$	Adaptive Kernel
vech	The Half-Vectorization Operator
δ	Noise Variance
μ	Mean
ε_{i}	IID Noise
∇	Gradient Operators
H	Smoothing Parameter
L	Laplacian Operator
Y	Input Noisy Image
Z	Output Denoised Estimate
AWGN	Additive White Gaussian Noise
GLF	Gaussian Low Pass Filter
IID	Independent and Identically Distributed
KR	Kernel Regression
LARK	Locally Adaptive Regression Kernel
LMSE	Laplacian Mean Square Error
MD	Maximum Difference
MSR	Mean-to-Standard-Deviation Ratio
MR	Magnetic Resonance
MSE	Mean Square Error

NEW	Nadaraya-Watson Estimator
PSNR	Peak Signal to Noise Ratio
RMSE	Root Mean Squared Error
SAR	Synthetic Aperture Radar
SKR	Steering kernel Regression
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity Index
SVD	Singular Value Decomposition

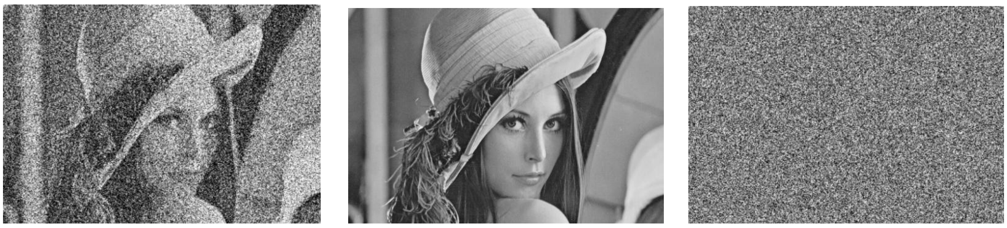
Chapter 1

INTRODUCTION

1.1 Introduction

Image denoising is the algorithm to find a clean image, given a noisy one. Usually, it is supposed that the noisy image is the sum of an underlying clean image and a noise component shown in Figure 1.1. Hence image denoising is a decomposition problem: The task is to decompose a noisy image into a clean image and a noise component.

There are several ways of such decompositions. One is interested in finding an acceptable clean image, given a noisy one. The notion of acceptable image is not clearly described, but the idea is that the denoised (cleaned) image must be similar to the original clean image. The notion of acceptable image therefore, involves prior knowledge, without prior information, image denoising would be impossible.



(a) Noisy = (b) Clean + (c) Noise

Figure 1.1: A noisy image is the summation of noise and clean image.
In this figure (a) is the noisy image, (b) is the clean image and (c) is the noise that added to the image.

Typical images are corrupted with noise modeled with either a Gaussian, exponential or salt and pepper distributions. The other noise is a speckle noise, which is multiplicative in nature. The presence of noise in an image is either in an additive or multiplicative form.

An additive noise follows the rule:

$$w(x,y) = s(x,y) + n(x,y) \tag{1.1}$$

When the multiplicative noise satisfies:

$$w(x,y) = s(x,y) \times n(x,y) \tag{1.2}$$

where $\mathrm{s(x,y)}$ is the original image, $\mathrm{n(x,y)}$ denotes the noise introduced into the image to produce the corrupted image $\mathrm{w(x,y)}$, and $(\mathrm{x,y})$ represents the pixel location. The above image algebra is done at pixel level.

The sources and types of noise are related to the physical measurement. Noise usually comes from a source that is different from the one to be measured such as read-out noise in digital cameras, but sometimes it is because of the measurement process itself like photon shot noise [1].

Sometimes, noise might be because of the mathematical manipulation of an image, as is the case in image deconvolution or image compression. Often, a measurement is corrupted by several sources of noise, and it is usually difficult to determine fully all of them. Always, noise is the undesirable part of the image. Ideally, one seeks to decrease noise by manipulating the signal acquisition process, but when such a modification is impossible, denoising algorithms are required.

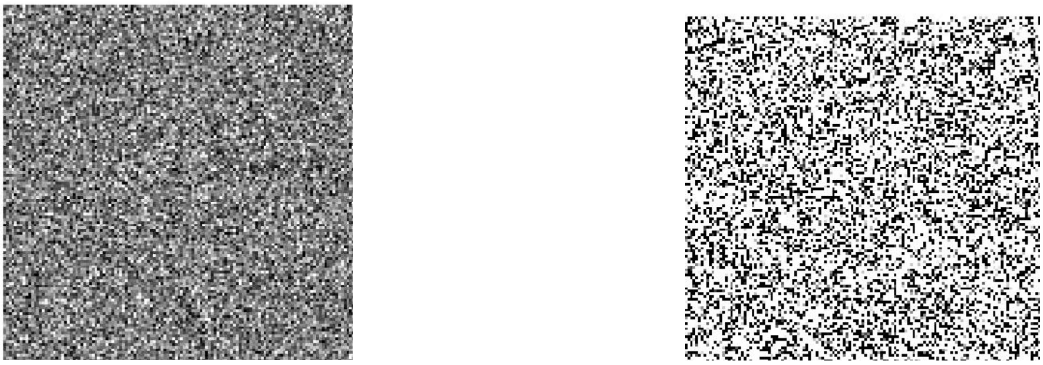
1.2 Various Types of Noises

1.2.1 Gaussian Noise

Gaussian noise is coherently distributed over the signal [1]. This describes that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value. As the name demonstrates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution, Figure 1.2 exhibit the Gaussian noise with different variances. As the name indicates, this type of noise has a Gaussian distribution, which has a bell shaped probability distribution function given by:

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2/2\sigma^2} \tag{1.3}$$

where g represents the gray level, m is the mean or average of the function, and σ is the standard deviation of the noise.



(a) (Mean=0, variance 0.05)

(b) (Mean=1.5, variance 10)

Figure 1.2: (a) and (b) are two different Gaussian noises

1.2.2 Speckle Noise

This noise called as a multiplicative noise. This type of noise passes in almost all coherent imaging tactics such as laser, acoustics and SAR (Synthetic Aperture Radar) imagery. The source of this noise is attributed to random interference among the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise [2]. Speckle noise follows a gamma distribution and is given as:

$$f(g) = (\frac{g^{\alpha-1}}{(\alpha-1)!\,a^{\alpha}}) \times e^{-\frac{g}{a}} \tag{1.4}$$

where variance is $\mathrm{a}^{2\alpha}$ and g is the gray level.

We show an image with speckle noise (variance 0.05) in Figure 1.3.

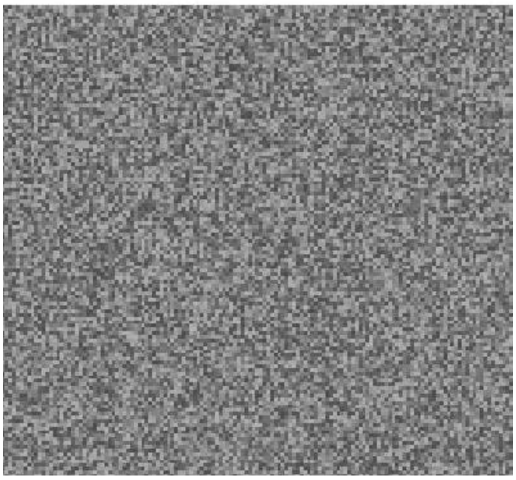


Figure 1.3: Speckle noise

1.2.3 Rician Noise

Magnetic resonance images are usually contorted by Rician noise [3]. In MRI data, each pixel consists of a complex number. For viewing MRI data, the absolute value of each complex number is appropriated. If the real and imaginary parts of these complex numbers are Gaussian-distributed and independent (with the same variance), the absolute value is Rician-distributed. Similarly to the Poisson distribution, the Rician distribution can be well approximated with a Gaussian distribution, for higher mean values. The Rician Noise Generator block generates Rician distributed noise. The Rician probability density function is given by:

$$f(x) = \frac{x}{\sigma^2} i_0\left(\frac{mx}{\sigma^2}\right) exp\left(-\frac{x^2+m^2}{2\sigma^2}\right) \tag{1.5}$$

where σ is the standard deviation of the Gaussian distribution that underlies the Rician distribution noise and m^2 = ${m_i}^2$+${m_q}^2$, where m_i and m_q are the mean values of two independent Gaussian components.

1.2.4 Other Kinds of Noise

The types of noise possibly corrupting images are too numerous to list in this work. Other types of noise are salt and pepper, speckle noise, Brownian noise and so on. We are mostly interested in digital images, but images taken with analog cameras are also affected by noise, such as film grain. Physical degradation of old photographs such as daguerreotypes also causes a variety of artifacts, such as scratches. In this work we use additive WGN with different standard deviations.

1.3 Evaluating Denoising Results:

1.3.1 PSNR

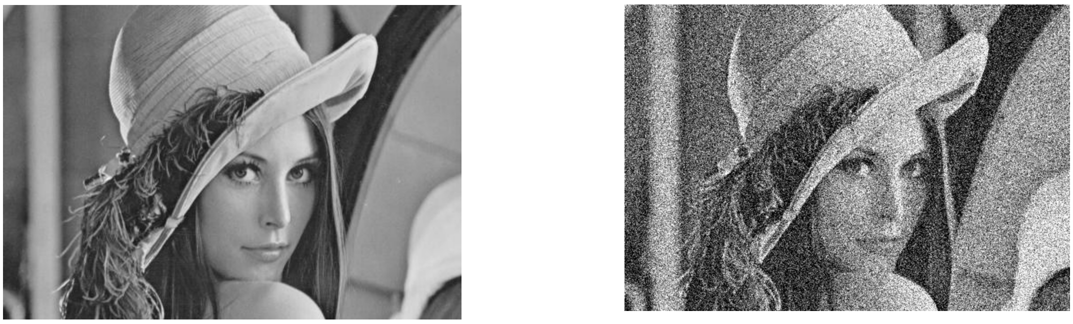
The most commonly used metric for image quality determination is the peak signal to noise ratio ($PSNR$), which is a full-reference metric and calculated between two images i_1 and i_2 as follows:

$$PSNR = 20log_{10}\left(\frac{MAX}{RMSE(i_1, i_2)}\right) \tag{ 1.6}$$

where, $RMSE$ refers to the root mean-square error between the main image and the reconstructed image and MAX equal to maximum possible pixel value of the images (255 for 8-bit images).

$$RMSE = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(i_{1_{ij}}i_{2_{ij}})^2} \tag{1.7}$$

$RMSE$ refers to the root mean square error between the main image and the reconstructed image $(\mathrm{i}_1, \mathrm{i}_2)$. PSNR is possibly the simplest of all image quality metrics. however, higher dB values tend to correlate with higher visual similarity between the two images x and y. Although, higher dB values do not all along indicate higher visual similarity, which is why expansive task has been put into detecting alternative metrics.



(a) Clean image (b) PSNR: 14.16 dB

Figure 1.4: Test image Lena. Corrupted with additive white Gaussian (AWG) standard deviation of noise = 50, PSNR shows the quality of image and this case is very low for noisy image. (a) Clean image (b) Noisy image

1.3.2 Other Image Quality Assessment:

Some image quality assessment aim to exploit known characteristics of the human visual system. The structural similarity index (SSIM) [4] is a comprehensive reference image quality assessment which divides the performance of similarity measurement into three factors: (a) luminance, (b) contrast, and (c) structure. Among other substances, the SSIM takes into account that the human visual system is sensitive to relative differences in luminance, rather than to absolute changes in luminance. The SSIM is a measure that is smaller or equal to 1. The measure is equal to 1 unique if the two images being compared are identical.

Other full-reference image quality assessments involve the information-content weighted PSNR (IW-PSNR [6]), information fidelity criterion (IFC [8]) and the visual image information (VIF [7]).Variants of the SSIM involve a multi-scale extension (MS-SSIM [5]) and the information-content weighted SSIM (IW-SSIM

[6]). No-reference image quality assessments involve DIIVINE [9], CBIQ [10]. Some other types of these measurements that we use rarely are LBIQ [11], BLIINDS [12], BRISQUE [13], and BIQI [14]. Many techniques can be used to capture deviations from the probable information of natural images, where these differences can be measured in various techniques. The PSNR is still the de facto standard in image denoising, however the SSIM is also sometimes utilized.

1.4 Thesis Description

In this thesis we mainly concentrate on the denoising problem as we have illustrated above with some kinds of noise and some filtering algorithms. One type of filtering is local filtering that we modifying in chapter three. The main contribution of this thesis is to describe and propose the kernel regression framework as an effective tool even for irregularly spaced samples. Kernel regression is known as a nonparametric approach that requires minimal assumptions, and hence the framework is one of the suitable approaches to the regression problem.

We modified the SKR method by using the Hessian matrix instead of gradient matrix and finding the structure tensor of the Hessian that makes the contours more accurate at the edges compared to the structure tensor of the gradient matrix.

The content of this thesis have been organized as follow. We discuss classic kernel regression in Chapter 2 and we expand it to our proposed adaptive kernel regression in Chapter 3. Chapter 4 provides details about simulation and the results are compared, with the state of the art algorithms. Chapter 5 pertains to conclusion of the thesis and future work.

Chapter 2

FILTERING AND KERNEL REGRESSION APPROACH

2.1 Mean Filter

Mean filter [1] behaves on an image by smoothing it which decreases the intensity variation between neighborhood pixels. The mean filter is nothing but an efficient sliding window spatial filter that swaps the center value in the window with the average of all the neighboring pixels values involving it. By performing this, it replaces pixels, which are unrepresentative of their surroundings. It is applied with a convolution mask, which grants a result that is a weighted sum of the values of a pixel and its neighbors.

It is also identified a linear filter, the mask or kernel is a square. Often a 3×3 square kernel is utilized. If the coefficients of the mask sum up to one, then the average brightness of the image is not changed. If the coefficients sum to zero, the average brightness is lost, and it returns a dark image. The mean or average filter behaves on the shift-multiply-sum principle.

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}$$

Figure 2.1: A constant weight 3×3 filter mask

Measuring the perspicuous convolution of an image with this kernel carries out the mean filtering process. It is consequential when the noise in the image is of impulsive type.

2.2 Median Filter

A median filter belongs to the class of nonlinear filters opposite the mean filter. The median filter likewise follows the moving window principle analogous to the mean filter. A 3×3, 5×5, or 7×7 kernel of pixels is scanned over pixel matrix of the entire image. The median of the pixels values in the window is measured, and then center pixel of the window is replaced with the quantized median. Median filtering is done by, first classifying all the pixel values from the surrounding neighborhood into numerical order and then swapping the pixel being considered with the middle pixel value. Note that the median value must be written to a separate array or buffer so that the outcomes are not distorted as the process is performed. We show the methodology as:

$$\left(\begin{array}{c|ccc|c} 1 & 1 & 2 & 5 & 4 \\ \cline{2-4} 3 & 3 & 2 & 1 & 3 \\ 4 & 4 & 5 & 1 & 0 \\ 5 & 1 & 1 & 2 & 3 \\ \cline{2-4} 0 & 0 & 2 & 1 & 3 \end{array}\right)$$

Figure 2.2: A 3×3 Median filter with median value equal to three.

The median is more robust compared to the mean. Hence, a single very unrepresentative pixel in a neighborhood will not affect the median value significantly. Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unreal pixel values when the filter straddles an edge [15].

2.3 Nonparametric Kernel Regression in 1-D

We have two methods for image processing, classical parametric method that relies on a particular model of signal and compute the parameters of the model in the presence of noise, and other one is totally in contrast to the parametric methods, non –parametric methods relies on just data to dictate the structure of the model, in which case this explicit model is called a regression function [16]. This method is used frequently for pattern detection and discrimination problems [17].

There are many concepts that are related to general framework indicated under diverse names such as bilateral filter, normalized convolution [18][19], edge-directed interpolation [20], and moving least-squares [21].

At first we threat the 1-D case where the measured data is given by:

$$y_i = z(x_i) + \varepsilon_i \qquad\qquad i = 1,2,3, \dots, p \qquad\qquad (2.1)$$

where $z(\cdot)$ is the (unknown) regression function, y_i is noisy sample, $x_i$is position of samples, p is number of samples in local analysis window and ε_i is independent and identically distributed (IID) zero mean noise value. $z(x_i)$ describes the intensity of the pixel of location x_i in a non-parametric manner. This regression function depends on the pixel intensities of neighboring pixels around a uniform or nonuniform grid. This kernel regression provides point wise estimation with minimal assumptions about global signal or noise models. It should be mentioned that we estimate the regression function $z(\cdot)$ by using Taylor series in local window and finding the polynomial bases of regression function at that window, we find the actual clean signal and the n-th derivatives of the signal of location x_i.

2.4 Mathematical Estimation for finding Kernel Regression

Because the specific form of $z(\cdot)$ (may remain unspecified), we assume that it is locally smooth to some order N, then to find the value of the function at any point x given the data, we can rely on a generic local expansion of the function near this point. Particularly, if the position of interest x is near sample at x_i then we have N-term Taylor series:

$$z(x_i) = z(x) + z'(x)(x_i - x) + \frac{1}{2!}z''(x)(x_i - x)^2 + \cdots + \frac{1}{N!}z^{(N)}(x)(x_i - x)^N \quad (2.2)$$

$$= \beta_0 + \beta_1(x_i - x) + \beta_2(x_i - x)^2 + \cdots + \beta_N(x_i - x)^N \quad (2.3)$$

where β_0=z(x), β_1=z′(x) and $\beta_N$is the n-th derivative of the function, now if we use the Taylor series as local representation of the regression function, by estimating the parameter β_0 can yield the desired estimate of the regression function based on the data. Indeed the local approximation can be built upon bases other than polynomials [3]. The parameters $\beta_{1,}\beta_{2,} \ldots, \beta_N$ will provide localized n-th derivatives of the regression function.

Because of the local approximation, we estimate the parameters $\beta_{1,}\beta_{2,} \ldots, \beta_N$ from the data by giving the nearby samples higher weight than samples placed farther away. By using a least-squares theory we can capture this idea and we can find $\beta_{1,}\beta_{2,} \ldots, \beta_N$ by optimization problem: (2.4)

$$\min_{\beta_n} \sum_{i=1}^{p} ([y_i - \beta_0 - \beta_1(x_i - x) - \beta_2(x_i - x)^2 - \cdots$$

$$- \beta_N(x_i - x)^N]^2 \frac{1}{h} K(\frac{(x_i - x)}{h})) \quad (2.4)$$

We defined the kernel function K (·) in order to penalize the distance away from the local position where the approximation is centered, h is smoothing parameter also called bandwidth controls the strength of this penalty. We can choose such functions: such as Gaussian, exponential, or other forms [3] as our kernel.

If we choose N=0, local linear filter is obtained, which is known as the Nadaraya-Watson Estimator (NWE) [22]. The form of this estimator is given below:

$$\hat{z}(x) = \frac{\sum_{i=1}^{p} k_h(x_i - x)\, y_i}{\sum_{i=1}^{p} k_h(x_i - x)} \tag{2.5}$$

where

$$K_h(x_i - x) = \frac{1}{h} k\left(\frac{x_i - x}{h}\right) \tag{2.6}$$

where h is a smoothing parameter for our kernel (K). This is the simplest filter for kernel regression method. As we increase the order, our local approximation become so complex, although increasing the order has effect that mentioned at [3][12].

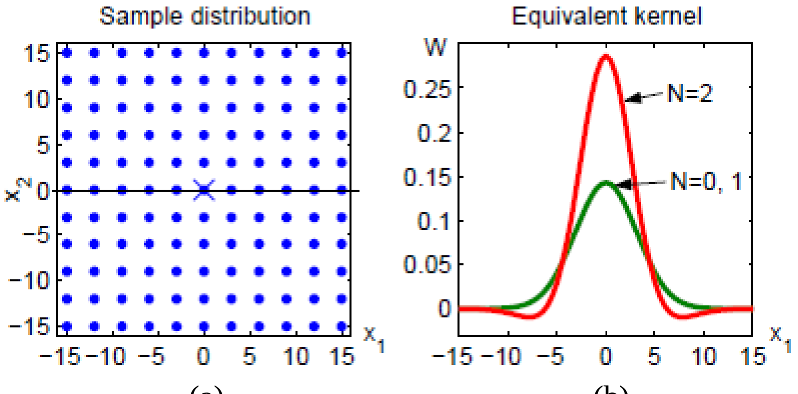
In general, lower order approximates, such as NWE, consequence in smoother signals (large bias and small variance) as there are fewer degrees of freedom. On the other hand over-fitting occurs in regressions using higher orders of approximation, approaching in small bias and large estimation variance.

2.5 Nonparametric Kernel Regression In 2-D

Analogous to the 1-D case in (2.1), the data model in 2-D is measured by:

$$y_i = z(x_i) + \varepsilon_i \qquad i = 1,2,3,\dots,p, \qquad x_i = [x_{1i}, x_{2i}]^T \tag{2.7}$$

where y_i is noisy sample at a sampling position x_i , (x_{1i}, x_{2i}) are spatial coordinates, $z(\cdot)$ is regression function that we want to estimate, ε_i is IID zero mean noise, p number of pixels that allocated in an arbitrary window around position x of interest as shown in Figure 2.3.



(a) (b)

Figure 2.3: Data model for 2-D kernel regression [23], (a) a regularly sampled data set and (b) a horizontal cross section of the equivalent kernels of orders N = 0, 1, 2

Local representation of the regression function with Taylor series (up to N order) is given by:

$$z(x_i) \approx z(x) + \{\nabla z(x)\}^T(x_i - x) + \frac{1}{2}(x_i - x)^T\{\mathcal{H}z(x)\}(x_i - x) + \cdots \tag{2.8}$$

$$= z(x) + \{\nabla z(x)\}^T(x_i - x) + \frac{1}{2}vec^T\{\mathcal{H}z(x)vec\{(x_i - x)(x_i - x)^T\} + \cdots \tag{2.9}$$

where ∇ and $\mathcal{H}$ are gradient (2×1) and Hessian (2×2) operators, respectively, and vec(·) is the vectorization operator, which lexicographically orders a matrix into a column-stack vector. Denoting vech(·) as the half-vectorization operator of the "lower triangular" part of a symmetric matrix like equation below:

$$vech\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} = [a_{11}\ a_{12}\ a_{13}a_{22}a_{23}a_{33}]^T \tag{2.10}$$

And supposing the symmetry of the Hessian matrix, the local representation in (2.9) is simplified to:

$$z(x_i) \approx \beta_0 + \beta_1^T(x_i - x) + \beta_2^T vech\{(x_i - x)((x_i - x)^T\} + \cdots \tag{2.11}$$

Comparison of (2.11) and (2.9) suggests that β_0 is the pixel value of interest, and the vectors β_1 and β_2 are the first and second derivatives, respectively.

$$\beta_0 = z(x) \tag{2.12}$$

$$\beta_1 = \nabla z(x) = \left[\frac{\partial z(x)}{\partial x_1} \quad \frac{\partial z(x)}{\partial x_2}\right]^T \tag{2.13}$$

$$\beta_2 = \mathcal{H} z(x) = \frac{1}{2}\left[\frac{\partial^2 z(x)}{\partial x_1^2} \quad 2\frac{\partial^2 z(x)}{\partial x_1 \partial x_2} \quad \frac{\partial^2 z(x)}{\partial x_2^2}\right]^T \tag{2.14}$$

As in the case of univariate data, the β_n's are calculated from the following optimization problem:

$$\min_{\beta n} \sum\nolimits_{i=1}^{p} ([y_i - \beta_0 - \beta_1^T(x_i - x) - \beta_2^T vech\{(x_i - x)(x_i - x)^T\}$$

$$- \cdots]^2 \, k_H \, (x_i - x)) \tag{2.15}$$

with

$$k_H(x_i - x) = \frac{1}{det(H)} k\big(H^{-1}(x_i - x)\big) \tag{2.16}$$

where k is the 2-D kernel function, H is the 2×2 smoothing matrix, We can rewrite (2.15) as weighted least squares optimization problem:

$$\hat{b} = argmin_b[(y - Xb)^T K (y - Xb)] \tag{2.17}$$

where

$$y = [\, y_1 \quad y_2 \quad y_3 \; \dots\dots \; y_p\,]^T$$

$$b = [\beta_0 \;\; {\beta_1}^T \; \dots\dots \; {\beta_N}^T]^T$$

$$K = diag[k_H(x_1 - x)\, k_H(x_2 - x) \dots k_H\big(x_p - x\big)]$$

$$X = \begin{bmatrix} 1 & (x_1 - x)^T & vech^T\{(x_1 - x)(x_1 - x)^T\} & \cdots \\ 1 & (x_2 - x)^T & vech^T\{(x_2 - x)(x_2 - x)^T\} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \big(x_p - x\big)^T & vech^T\{\big(x_p - x\big)\big(x_p - x\big)^T\} & \cdots \end{bmatrix}$$

with “diag” defining a diagonal matrix. Using the notation above, the optimization (2.17) provides the weighted least square estimator as:

$$\hat{b} = (X^T K X)^{-1} X^T K Y \tag{2.18}$$

With equation (2.18) we can find the estimate of the image (β_0) which is the desired and the n-the derivatives of z(·)(desired image or denoise image), just by premultiplied equation (2.18) with column vectors $e_1, e_{2,}$, .. depend on the purpose of estimation. where e_1 is a column vector with the first element equal to one, and the rest equal to zero and e_2 is a column vector with the second element equal to one, and the rest equal to zero.

This kernel function is bounded to be only a function of the spatial locations x_i and x, this formulation where only locator is used to find the kernel is known as (classical, or not data-adaptive) kernel regression in the nonparametric statistics literature.

In order to estimate the similarity of two pixels, in general, we can naturally employ both the spatial distance (Δx) and the gray level distance (Δz). The simplest way to incorporate the two distances is the Euclidean distance between points. However, a much more consequential way to combine the two distances (Δ's) is to define a “signal-induced" distance [23] which basically stands for a distance between the points calculated along the shortest path on the signal manifold (geodesic distance). where a geodesic is a path of shortest distance between two points.

In Figure 2.4 we show all kinds of distances that we mentioned before, z axis shows the gray-level distance and x axis shows the spatial distance.

It is also noteworthy that, classic kernel regression works based on the spatial distance, bilateral and non-local means filter work based on the Euclidean distance and Steering kernel regression works base on the geodesic distance, this thesis our kernel also uses geodesic distance.

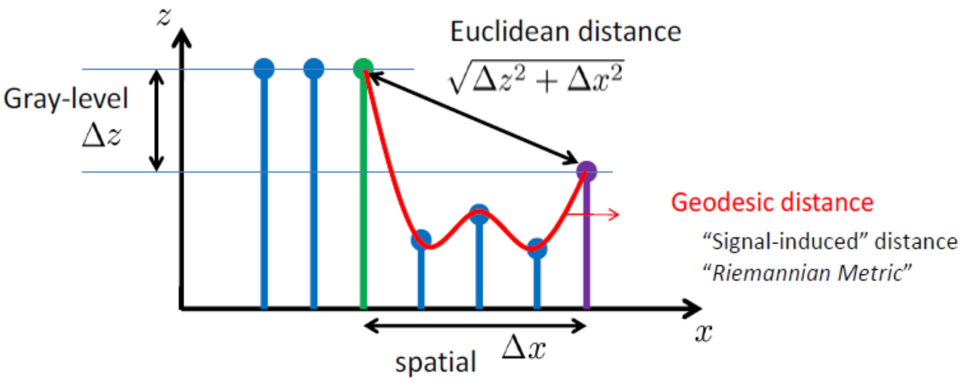


Figure 2.4: Showing the different distances between two pixels [26]

Classic kernel regression estimates the pixel z(x) by the combination of neighboring samples with linear weights, thus, the effective size and shape of the regression kernel are not adapted locally to image feature such as edges, Figure 2.5 shows the effect of this kernel.

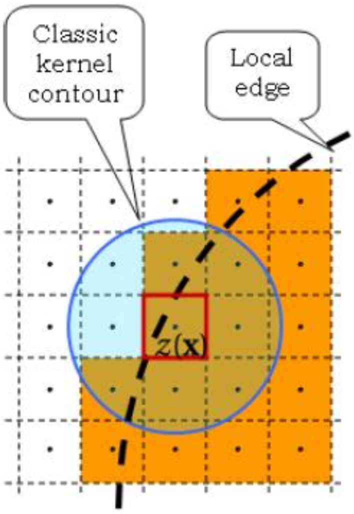


Figure 2.5: Applying the classic kernel regression to a local edge [26]

Chapter 3

PROPOSED SKR ALGORITHM VIA LOCAL LAPLACIAN

This proposed algorithm has three main parts: calculating the local Laplacian for each pixel in local window, creating the steering matrices with local Laplacian that we measured in part one and finding the structure tensor for each pixel, and creating the steering kernel from that steering matrix and after that we do iteration to suppress the effect of noise.

At each iteration we decrease the noise up to give the minimum MSE and then finish the iteration. The preference of this method compared to similar algorithms is to create the sharper image denoising which makes sharper edges and less blurred image . This chapter will cover each of these steps and algorithm behind them. This chapter starts with a short preface about finding adaptive kernels.

3.1 Locally Data-Adaptive Kernels

In the previous chapter we discussed about classic kernel regression which is based on the spatial distance between pixels, however, data-adaptive kernel regression relies not only on the spatial aspects (the sample ground and density), but also the photometric aspects of these samples (i.e., pixel values). Thus, the consequential size and form of the regression kernel are changed locally to image features.

Figure 3.1 shows a desired property of such a regression kernel when kernel is placed at the edges.

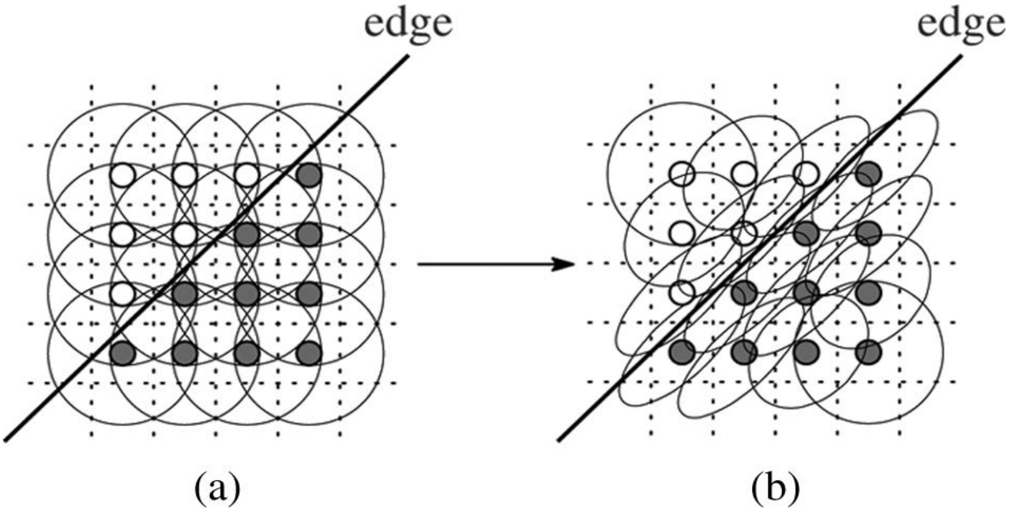


Figure 3.1: Kernel spread in a uniformly sampled data set. (a) Kernels in the classic method (b) Data-adapted kernels.

From Figure 3.1 we understand that by using classic kernel regression the contours have same shapes all over the image, but local adaptive kernel converts the contour shapes similar to the underlying image.

Therefore, the data-adaptive kernel suppresses noise while preserving local image structures. Locally Data-Adaptive Kernels (LARK) is structured similarly to (2.15) as an optimization problem given as below:

$$\min_{\beta n} \sum_{i=1}^{p} ([y_i - \beta_0 - \beta_1^T(x_i - x) - \beta_2^T vech\{(x_i - x)(x_i - x)^T\} - \cdots]^2 \ldots$$

$$\times \mathcal{K}_{adapt}\,(x_i - x, y_i - y)) \tag{3.1}$$

where the $\mathcal{K}_{\mathrm{adapt}}$ depends on the spatial sample coordinates $\mathrm{x_i}$'s and density as well as the photometric values $\mathrm{y_i}$ of the data. If we want to compare with classic kernel, adaptive kernel penalize, the spatial and photometric distance away from the local position where the approximation is centered.

3.2 Some Kinds of Locally Adaptive Filter

3.2.1 Bilateral Filter

The Bilateral filter proposed by Tomasi and Manduchi [24] [25] is ordinary choice of the adaptive kernel is used to suppress additive noise from images.

This adaptive kernel is to use separate terms for penalizing the spatial "distance" between the pixel position of interest x and its neighboring pixel positions $\{ x_i \}$, and the photometric "distance" between the pixel of interest y and its neighbors $\{ y_i \}$.

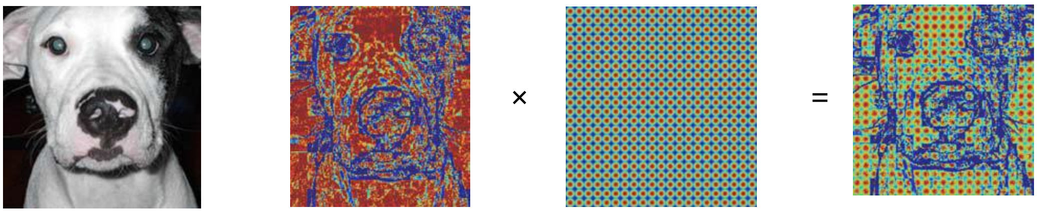
$$k_{bilat}(x_i - x, y_i - y) \triangleq k_{h_s}(x_i - x).k_{h_p}(y_i - y) \qquad (3.2)$$

where k is the Gaussian kernel function and h_s is the spatial smoothing scalar and $h_p$is the photometric smoothing scalar. The kernel functions are expressed as:

$$k_{h_s}(x_i - x) = \frac{1}{2\pi\sqrt{det(h_s^T h_s)}} exp\left\{-\frac{((x_i - x)^T (h_s^T h_s)^{-1}(x_i - x))}{2}\right\} \qquad (3.3)$$

$$k_{h_p}(y_i - y) = \frac{1}{2\pi\sqrt{det(h_p^T h_p)}} exp\left\{-\frac{((y_i - y)^T (h_p^T h_p)^{-1}(y_i - y))}{2}\right\} \qquad (3.4)$$

It is clear in the Figure 3.2 the photometric kernel (b) capture local structure of image consequentially with applying the $k_{h_p}(y_i - y)$ to the local patches and get high weight to the similar pixel intensity and low weight to the different pixel intensity between the pixel of interest *y* and its neighbors $\{y_i\}$ and spatial distance kernel $k_{h_s}(x_i - x)$ panelizing the spatial distance between the pixel position of interest x and its neighboring pixel positions $\{x_i\}$.



(a) (b) Pixel similarity (c) Spatial similarity (d) Bilateral kernel

Figure 3.2: Applying Bilateral kernels in separate way. (a) Sample image (b) Photometric kernel (c) Spatial distance kernel (d) Bilateral kernel

In this case if we have strong noise then our result is not Satisfactory, modified version of bilateral filter is Susan filter [25] proposed by Michel Elad which excludes the center pixel from the estimates.

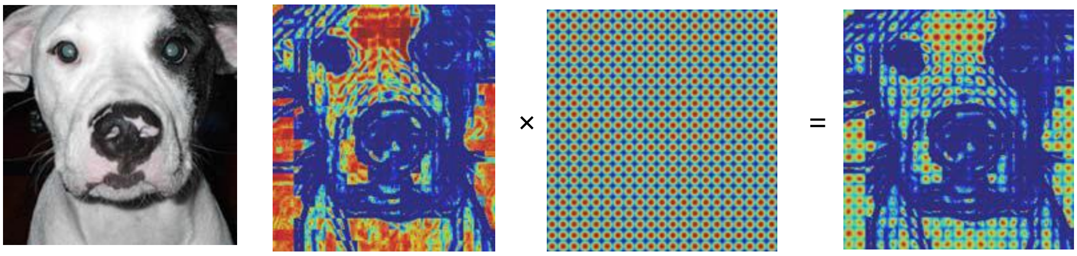
3.2.2 Non-Local Means (NL-Means) Filter

NL-means filter [26] [27], presented by Buades.et.al, is based on the natural redundancy of data in images. It is because of the fact that every small window in a natural image has many similar windows in the same image. The exclusivity of this filter is that the similarity of pixels has been more robust to noise by using a region comparison, rather than pixel comparison and also that matching patterns are not restricted to be local. That is, the pixels far away from the pixel being filtered are not penalized.

$$k_{nlm}(x_i - x, y_i - y) \triangleq k_{h_s}(x_i - x).k_{h_p}(y_i - y) \qquad (3.5)$$

where $\mathrm{y_i}$ is related to the patches instead of the pixels. And makes more smoothing effect compared by Bilateral filter.

It is clear from Figure3.3 the photometric kernel (b), captures the local structure of the image consequentially by applying the $\mathrm{k_{h_p}(y_i - y)}$ to the image and get high weight to similar patches intensity and low weight to different patch intensity between the pixel of interest $\{\mathrm{y}\}$ and its neighbors and pixel $\{\mathrm{y_i}\}$ and its neighbors and spatial distance kernel $k_{h_s}(x_i - x)$ panelizing the spatial distance between the pixel position of interest $\{x\}$ and its neighboring pixel positions $\{x_i\}$.



(a) (b) Patch similarity (c) Spatial similarity (d) NLM kernel

Figure 3.3: Applying NLM kernels in separate way. (a) Sample image (b) Photometric kernel (c) Spatial distance kernel (d) NLM kernel

3.3 Proposed Steering Kernel

The filtering mechanism that we design next takes the data-adaptive idea one step further, based upon the introductory nonparametric framework. In particular, we observe that the result of calculating the photometric kernel, $k_{h_p}(y_i - y)$ in (3.2) is to indirectly measure a function of the local gradient estimated between neighboring pixels and to use this estimate to weight the respective computations. H. Takeda [41] use gradient to find the first order structure tensor to estimate the steering kernel parameter but in this framework we use hessian matrix to find second order tensor to estimate the steering kernel parameter.

We use the second order classic kernel regression method for estimate of the image gradient of each pixel. Next, this estimate is used to approximate the dominant orientation of the local gradients in the image [28]. In a second filtering stage, this orientation details is then used to adaptively “steer” the local kernel, resulting in elongated contours expand along the directions of the local edge shape.

The SKR kernel function takes the form:

$$k_{adapt}(x_i - x, y_i - y) \triangleq k_{H_i^{steer}}(x_i - x) \qquad (3.6)$$

where $\mathrm{H}_i^{\mathrm{steer}}$'s are present the data-dependent complete (2×2) matrices which is called steering matrix or structure tensor. We indicate them as:

$$H_i^{steer} \ = h\tau_i c_i^{-.5} \qquad (3.7)$$

where again h and τ_i are the global smoothing parameter and the local agglomeration parameter, respectively, and c_i is (symmetric) steering matrix based on variations in the local gray-values.

Actually c_i capture the both spatial and photometric distance for regression function and it makes the shape to steer at local window. The distance that involves photometric and spatial distance together is called geodesic distance. Now, if we select a Gaussian kernel, i.e. plugging the steering matrix (3.5) into Gaussian kernel function (2.16), the steering kernel is mathematically represented as:

$$k_{h_i^{steer}}(x_i - x) = \frac{\sqrt{det(c_i)}}{2\pi h^2 \tau_i^2} exp\left\{-\frac{((x_i - x)^T c_i (x_i - x))}{2h^2\tau_i^2}\right\} \qquad (3.8)$$

For finding the unknown pixel β_0, the SKR (steering kernel) function takes all the steering matrices ($\mathrm{H}_i^{\mathrm{steer}}$) of the neighboring pixels ($\mathrm{y_i}$) around the position of interest x into account, and so, the steering kernel is not simply elliptic but it grants us weights that fit the local image structures more flexibly.

In the proposed method instead of using gradient for finding the structure of the local edges we use Laplacian to create the second order structure tensor for each pixel to finding $\mathrm{c_i}$ for each pixel. In this case we got the more accurate weights and the steering matrix become more flexible to shape of the edges and details.

An estimate of this second order structure tensor may be obtained as follows:

$$\widehat{c_i} = R_i^T R_i \tag{3.9}$$

where R_i is a stack of Laplacian vectors in a local analysis window.

$$R_i = \begin{pmatrix} \vdots & \vdots \\ z_{x1}(x_j) & z_{x2}(x_j) \\ \vdots & \vdots \end{pmatrix} \tag{3.10}$$

$\mathrm{z_{x1}}(\mathrm{x_j})$ and $\mathrm{z_{x2}}(\mathrm{x_j})$ are second derivatives along $\mathrm{x_1}$ (vertical) and $\mathrm{x_2}$ (horizontal) directions. The dominant local orientation of the Laplacians is then related to the eigenvectors of this estimated matrix.

For having a more convenient form of the steering matrix (second order structure tensor), our steering matrix can be decomposed into three components related to eigenvalue decomposition as follows:

$$c_i = \gamma_i \mathcal{U}_{\theta_i} \Lambda_i \, \mathcal{U}_{\theta_i}{}^{\mathrm{T}} \tag{3.11}$$

where $\mathcal{U}_{\theta_i}$ is the rotation matrix , Λ_i is the elongation matrix and γ_i is scaling parameter.

$$\mathcal{U}_{\theta_i} = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix} \tag{3.12}$$

$$\Lambda_i = \begin{bmatrix} \sigma_i & 0 \\ 0 & \dfrac{1}{\sigma_i} \end{bmatrix} \tag{3.13}$$

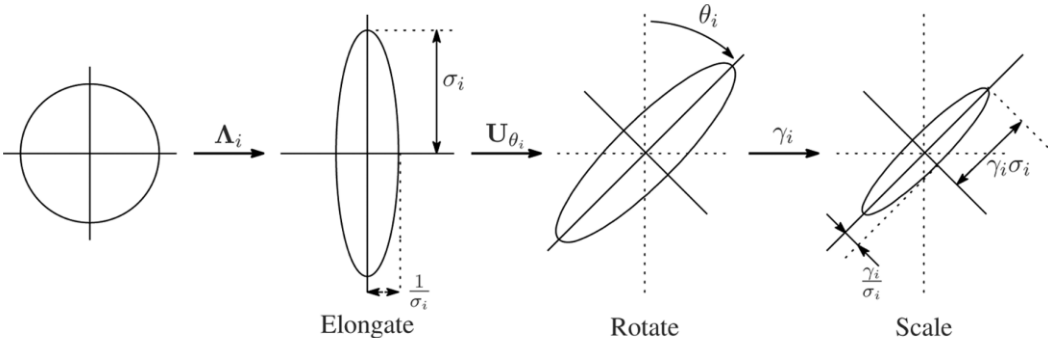


Figure 3.4: Effects of three components on the size of the regression kernel.

Figure 3.4 is a schematic representation showing the effects of the steering matrix and its elements on the size and shape of the KR footprint.

The Hessian matrix as derived from the second order Taylor series expansion can be used to describe the second structure tensor around a point. For giving more information about the structure tensor and comparison of the different order tensor we refer to R.D.Kriz [44] and [45] which compare the effect of order structure tensor.

The second structure tensor matrix (metric tensor) given by three parameters γ_i , σ_i and θ_i which are scaling, elongation and rotation parameters, respectively. Following the work in [28], the dominant orientation of the local Laplacian field is the singular vector related to the smallest (nonzero) singular value of the local Laplacian matrix R_i.

$$R_i = \begin{pmatrix} \vdots & \vdots \\ z_{x1}(x_j) & z_{x2}(x_j) \\ \vdots & \vdots \end{pmatrix} = u_i s_i v_i^T = u_i \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} [v_1 \quad v_1]^T \tag{3.14}$$

where $u_i s_i v_i^T$ is the compacted singular value decomposition of R_i , and s_i is a diagonal (2×2) matrix outlining the energy in the dominant directions. Then, the second column of the (2×2) orthogonal matrix v_i , $v_2 = [v_{12}, v_{22}]T$, describes the dominant orientation angle θ_i as:

$$\theta_i = arctan\left(\frac{v_{12}}{v_{22}}\right) \tag{3.15}$$

The singular vector related to the smallest nonzero singular value s_2 of R_i forms the dominant orientation of the local Laplacian field.

The elongation parameter α_i can be scaled due to the power of the dominant Laplacian direction as shown below:

$$\alpha_i = \frac{s_1 + \lambda'}{s_2 + \lambda'} \tag{3.16}$$

where $\acute{\lambda}$ is a regularization parameter [32] for the kernel elongation, which dampens the effect of the noise and limits the ratio from becoming degenerate.

At the end we defined the scaling parameter by:

$$\gamma_i = (\frac{s_1 s_2 + \lambda''}{M})^{\omega} \tag{3.17}$$

where λ'' is again a "regularization" parameter, which dampens the effect of the noise and preserves γ_i from becoming zero, M is the number of samples in the local analysis window, and ω is the structure sensitivity parameter.

This parameter changes size of the contours in kernels, depend on the area, in texture areas it become small and in smooth region it become big. We use these references for calculating this parameter [28] [29] [30].

The regularization parameters λ' and λ'' are used to prohibit the shape of the kernel from becoming infinitely narrow and long. In practice, it suffices to keep these numbers reasonably small, and, therefore, in all experiments in this proposed method, we fixed their values equal to $\lambda' = 1.0$ and $\lambda'' = 0.01$, respectively.

The structure sensitivity ω (typically $0 \leq \omega \leq 0.5$) controls how strongly the size of the kernel is affected by the local structure. The product of the singular values indicates the amount of energy of the local signal structure: the larger the product, the stronger and the more complex the local structure is. A large ω is preferable when the given signal carries severe noise.

Each pixel in this framework has the specific structure tensor which is based on the elongation, rotation and scaling parameters. In Table 3.1 we want to compare these parameters by using the first order structure tensor and second order structure tensor for the specific pixels in our image. We select four random pixels in Lena image and define $x_i = [x_{1i}, x_{2i}]$ as pixel position where (x_{1i}, x_{2i}) are spatial coordinates. Note that the above numbers related to first order structure tensor and the below numbers related to second order structure tensor in each block of this table and all of these parameters is computed at the first iteration of both SKR (first order structure tensor) and proposed (second order structure tensor) methods.

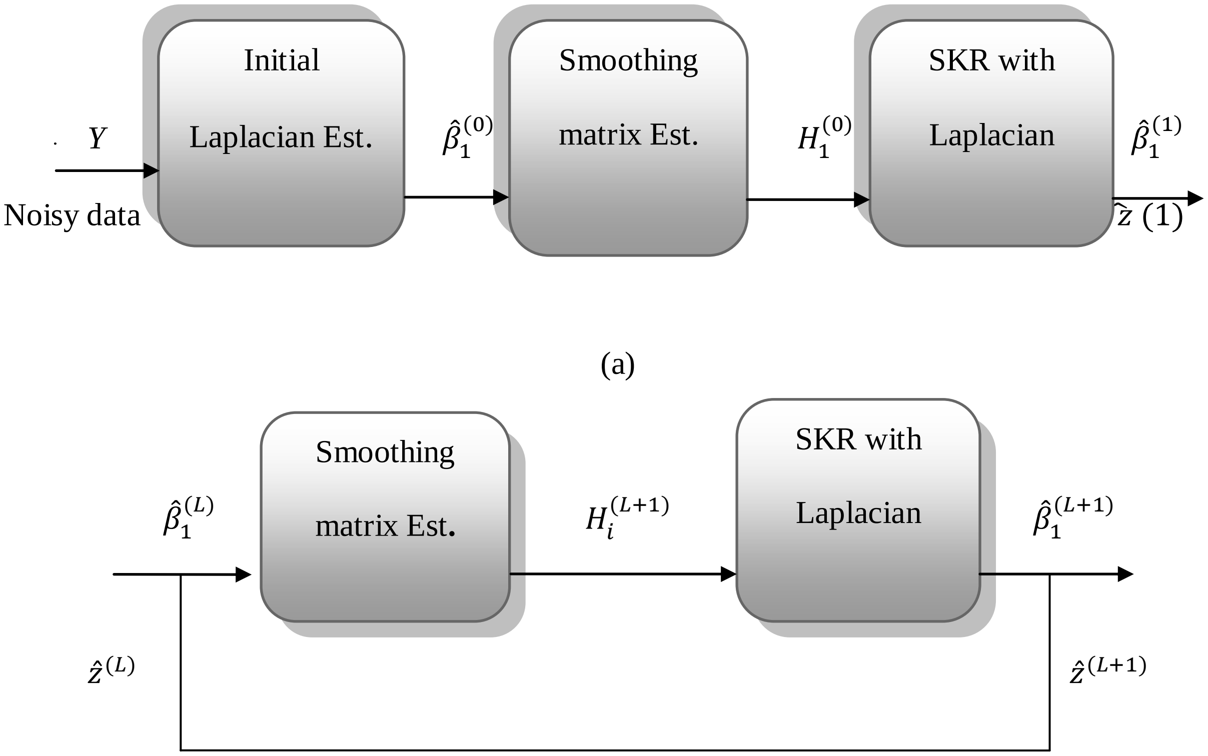
Table 3.1: Quantitative comparison of the different structure tensor parameters

Pixel position(x_i)	Scaling(γ_{x_i})	Rotation(θ_{x_i})	Elongation(σ_{x_i})
(126,175)	1.54 1.71	0.71 0.72	16.11 23.35
(50,3)	1.25 1.79	0.69 .0.68	13.98 24.60
(38,75)	3.20 3.82	0.05 0.06	14.76 21.99
(250,55)	1.57 1.95	0.94 0.93	13.81 23.04

With respect to the Table 3.1 we realize that by using the second order structure tensor, the elongation (σ_{x_i}) has increased significantly and due to this, the kernel becomes stretched. The scaling parameter (γ_{x_i}) has increased and result in attenuate the kernels shape. On the other hand comparing the rotation parameter (θ_{x_i}) we understand that the orientation of the kernel become stable. As an indeed estimate Figure 3.4 and Table 3.1 suggest that using the second order structure tensor does the image structure more accurately.

When we find the second order structure tensor ($\mathrm{c_i}$) for each pixel, we can built the adaptive steering kernel with equation (3.8) by applying the equation (2.18) for finding polynomial matrix (b) and using $\mathrm{k}_{\mathrm{h}_\mathrm{i}^{\mathrm{steer}}}$ instead of $\mathrm{k_H}$ at diag (k), we can find the estimate of the image (β_0) wich is desire for us and the n-the derivatives of $\hat{z}$ $(\cdot)$, just by premultiplied equation (2.18) with column vectors $\mathrm{e}_1, \mathrm{e}_{2,}, ..$ depend on the desired estimation.

Figure 3.5 shows the Block diagram representation of the proposed method, part (a) is Initialization step to find direct Laplacian and use it for finding the second order structure tensor from the second derivative of x direction and y direction in hessian matrix, part (b) is the Iteration to decrease the noise [31], in this block diagram L is the number of iteration and $\hat{z}$ (.) is the output denoise image and y is the noisy image.



(a)

(b)

Figure 3.5: Block diagram representation of the proposed method.

The difference between this framework and SKR is the order of the structure tensor, SKR uses first order structure tensor that makes the edges smooth and be more intensity to the noise in local patches, but by using the second order tensor, these effects are decreased. A good choice for c_i (structure tensor) will effectively spread the kernel function along the local edges, as shown in Figure 3.1 (b). We will see the effects of these changes at next part in objective and subjective comparison between this framework to other algorithms.

Chapter 4

SIMULATIONS AND RESULTS

In this chapter, we demonstrate the proposed algorithm result over the sample images by using Matlab platform. In this work we use three spatial filters to compare quantitatively and visually these filters. We use bilateral, non local means, SKR method to compare whit our proposed method.

To show the quantitative performance PSNR measurement is used. We add white Gaussian noise with different standard deviation to compare our result. Note that in all experiments in this thesis we used Gaussian type kernel functions [33].

As the test set, we used some of the standard gray-scale images commonly known as: Barbara, Boat, Cameraman, Parrot, Stream, House, Lena, peppers and Man (Figure 4.1). In this work we use Lena, Boat, Stream and parrot images for our experiments.

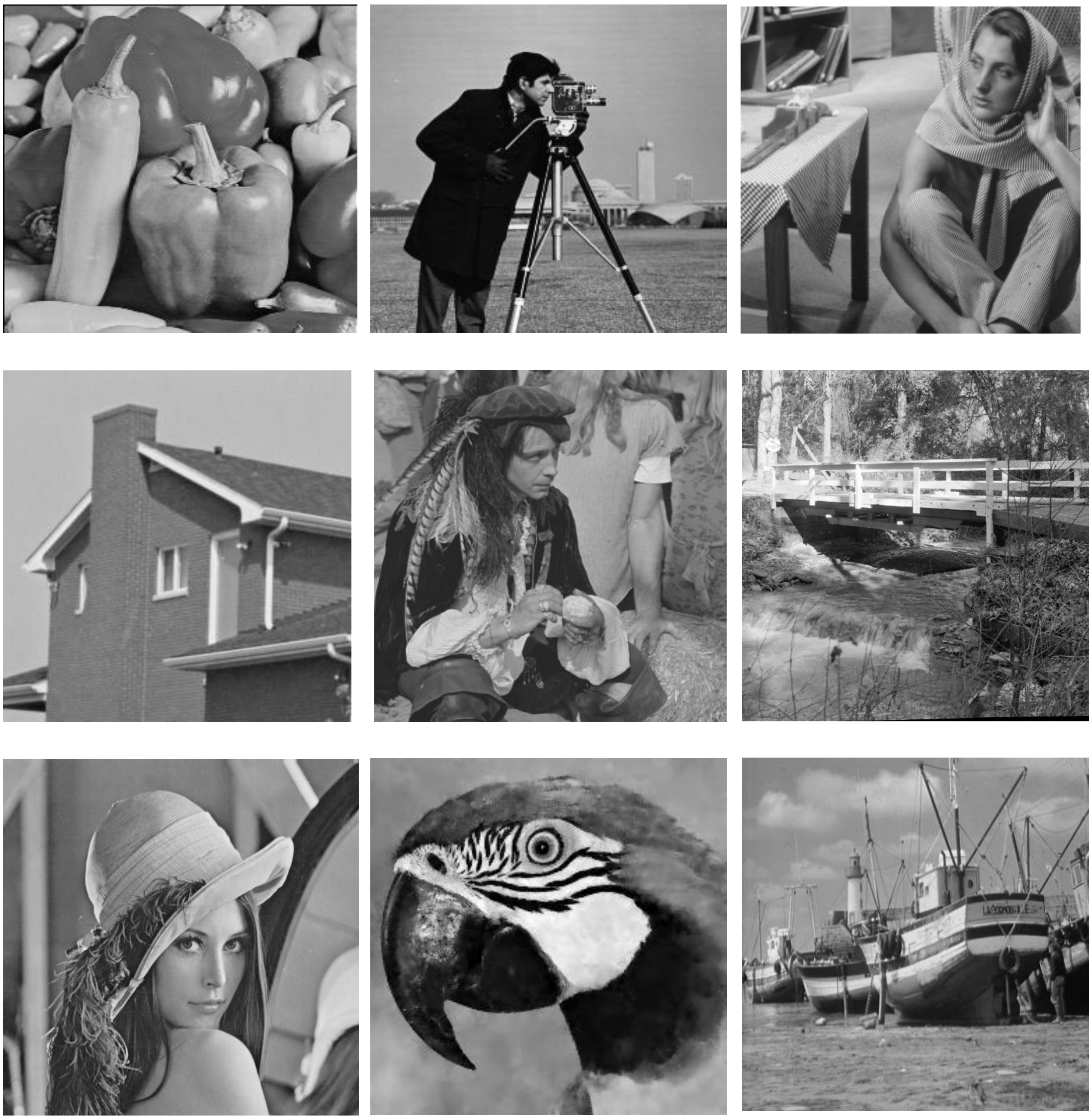


Figure 4.1: The standard testing images used in the experiment.

We first carried out an experiment to determine the convergence properties of the algorithm. Table 4.1 and Figure 4.2 present result of this experiment. We see that increasing the number of iterations make decreasing the variance of the approximation, it also caused improved bias (which known as blurriness). Therefore, in some iteration, a minimum MSE is captured. An example of the characteristic performance of (MSE) at the different number of iterations are shown in Table 4.1 and related chart.

Table 4.1: Experimental performance on Lena, Boat, Stream, Parrot images and given MSE at each iteration of the proposed algorithm

iteration	MSE δ=25	MSE δ=25	MSE δ=25	MSE δ=25
Num\ Pic	Lena	Boat	Stream	Parrot
1	618.01	610.95	603.02	567.1
2	393.06	411.74	454.23	378.74
3	245.89	274.03	338.92	261.59
4	123.98	199.63	270.18	199.97
5	124.55	155.91	229.07	163.89
6	95.78	127.49	203.76	140.67
7	76.16	108.30	188.16	125.25
8	62.78	95.47	178.95	115.03
9	53.95	87.29	174.17	108.75
10	48.49	82.59	172.53	105.42
11	42.42	80.52	173.16	104.26
12	47.87	83.84	174.73	105.83
13	49.87	85.62	176.62	107.76
14	52.76	87.62	178.67	109.19
15	57.76	89.76	182.86	111.67
16	62.76	93.78	187.63	116.76
17	73.47	110.57	200.57	128.85

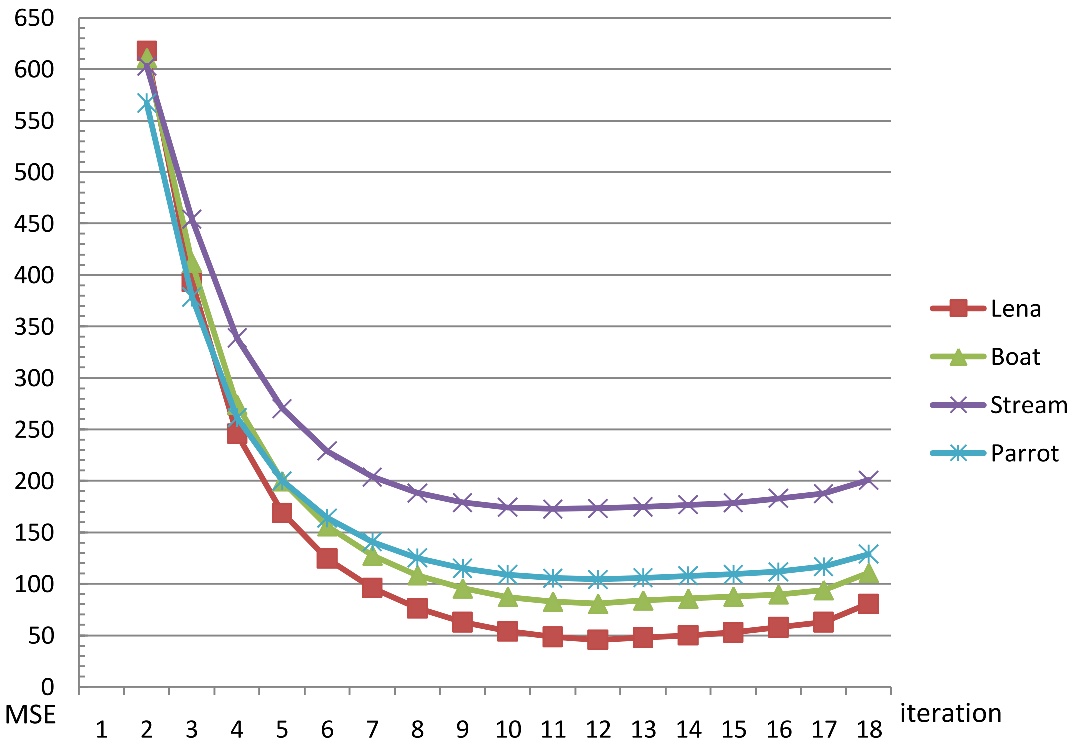
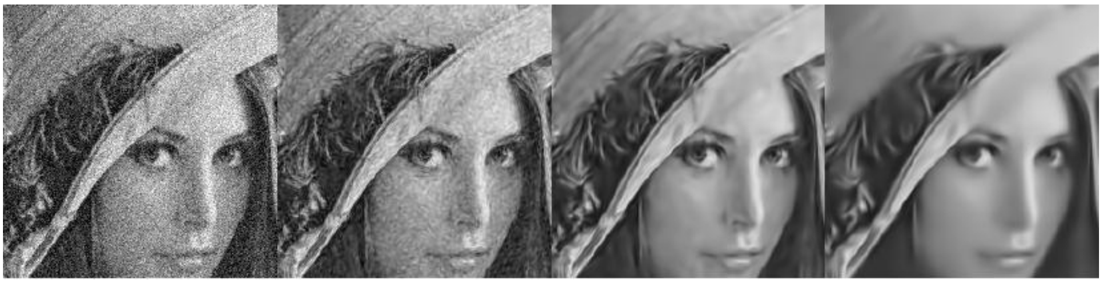


Figure 4.2: Experimental performance on Lena, Boat, Stream, Parrot images and given MSE at each iteration of the proposed algorithm, X axis shows the number of iteration and Y axis shows the MSE value.



(a) Input image (MSE = 625) (b) 4iterations (MSE = 123) (c) 10 iteration (MSE = 48) (d) 17 iteration (MSE =73)

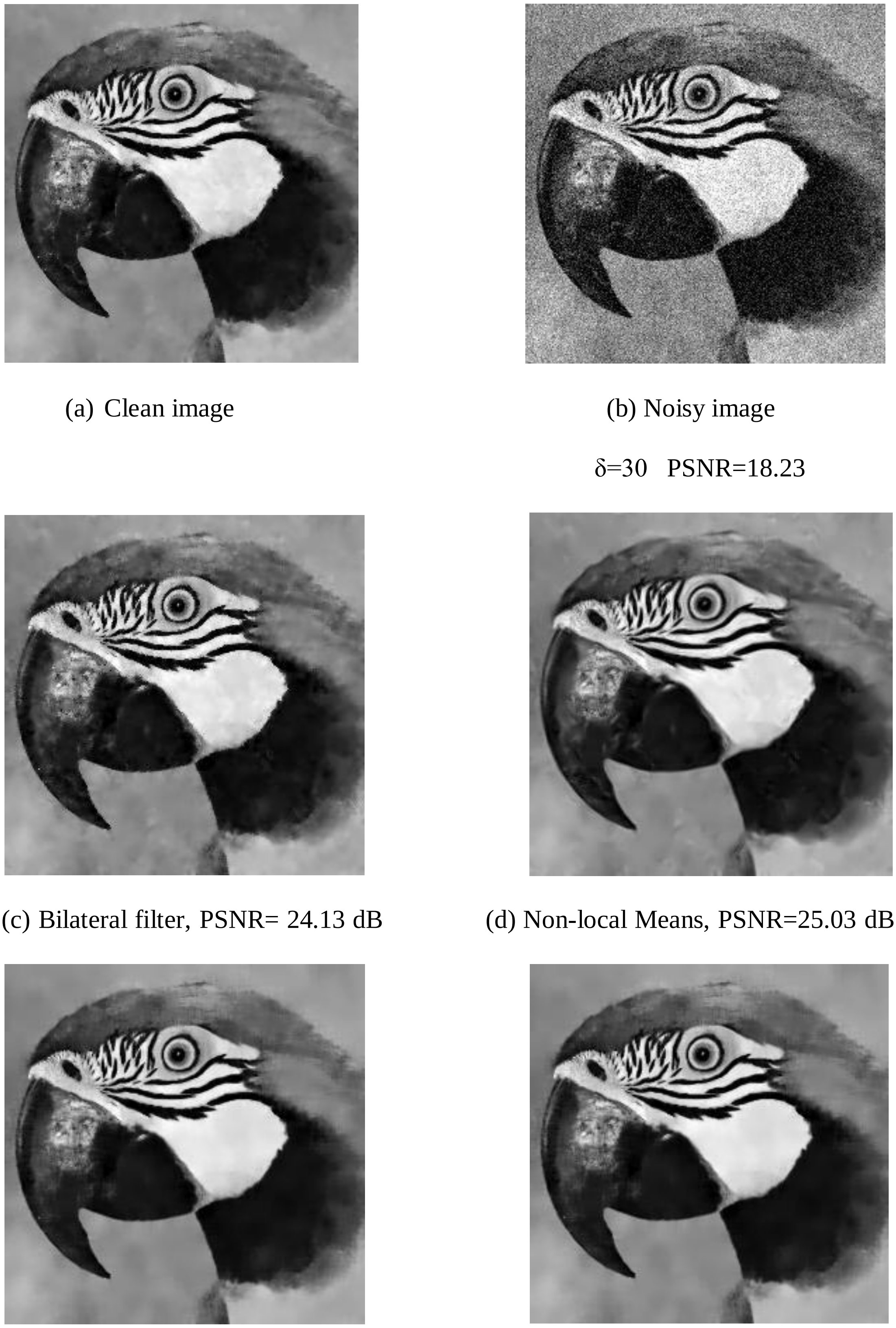
Figure 4.3: An example of the behavior of mean square error of the iterative Proposed method and its estimated images at the different number of iterations.

These experiments which have shown in the Table 4.1 illustrate that the MSE value decreases and at a certain number of iterations, the MSE hits bottoms, then, MSE start increasing and it leads to decreasing the PSNR which is not our desire.

In Figures 4.4 to 4.8, we show the results obtained with all four denoising algorithms (Bilateral filter [24] [25], Non-local Means [39], SKR filter [41] and Proposed method) on images Lena, Boat, Stream and Parrot with three differences (δ).

We summarize our work by creating the table that involves five kinds of denoising algorithms and 3 different standard deviations.

Comparisons show the superiority of this method to others especially in preserving the sharpness of edge regions and decreasing the artifacts in the smooth regions of the filtered image.



(a) Clean image

(b) Noisy image

δ=30 PSNR=18.23

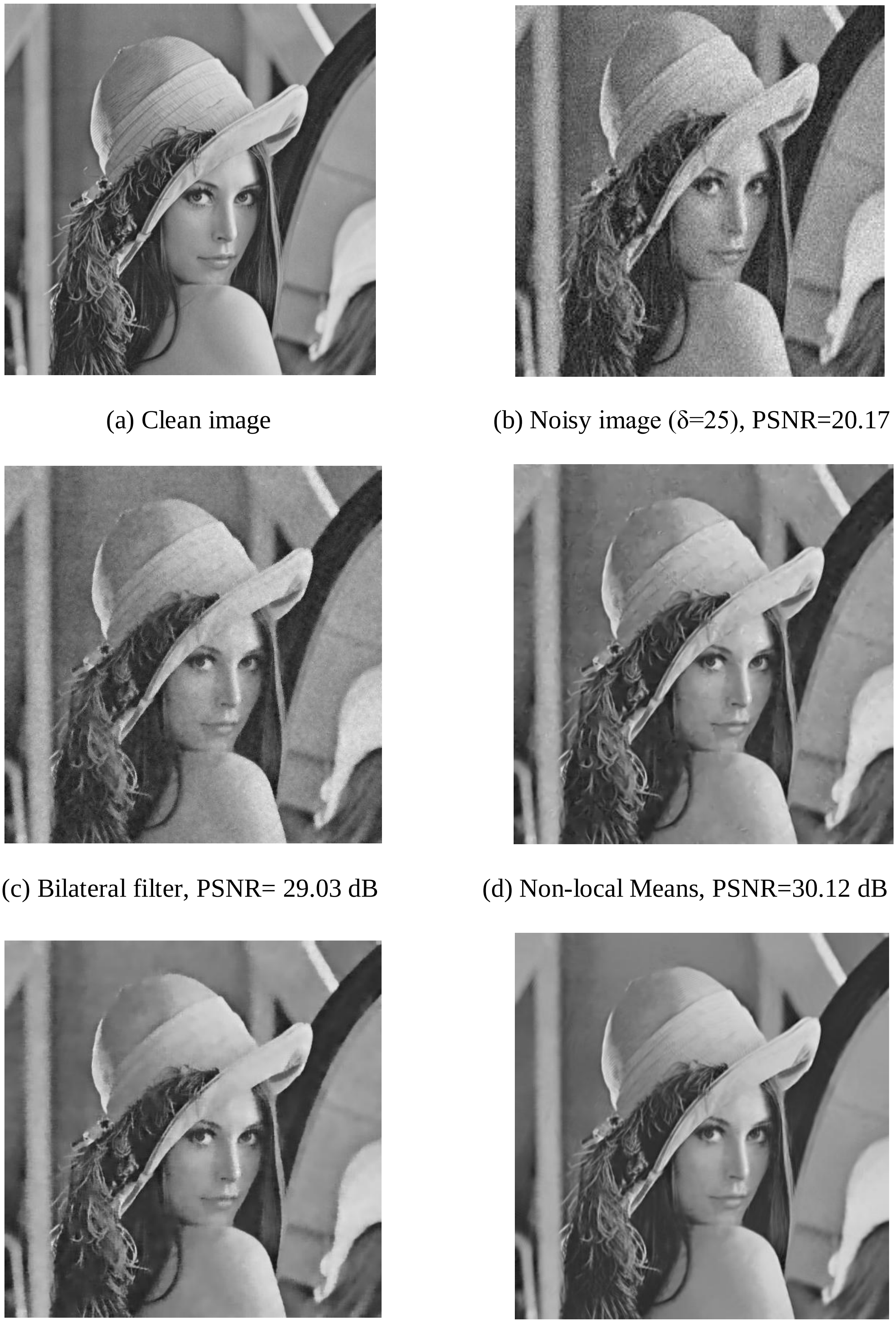
(c) Bilateral filter, PSNR= 24.13 dB

(d) Non-local Means, PSNR=25.03 dB

(e) SKR filter, PSNR=25.90 dB

(f) Proposed method, PSNR= 26.03 dB

Figure 4.4: Visual and quantitative comparison of Parrot image with δ=30
(a) original parrot image, (b) Noise-ridden image δ=30 (c) Bilateral filter (3.2) [25], (d) Non-local Means (3.3) [26] [27], (e) Iterative SKR filter with number of iteration (N) =12 [41], (f) Proposed method with (N=12)



(a) Clean image

(b) Noisy image (δ=25), PSNR=20.17

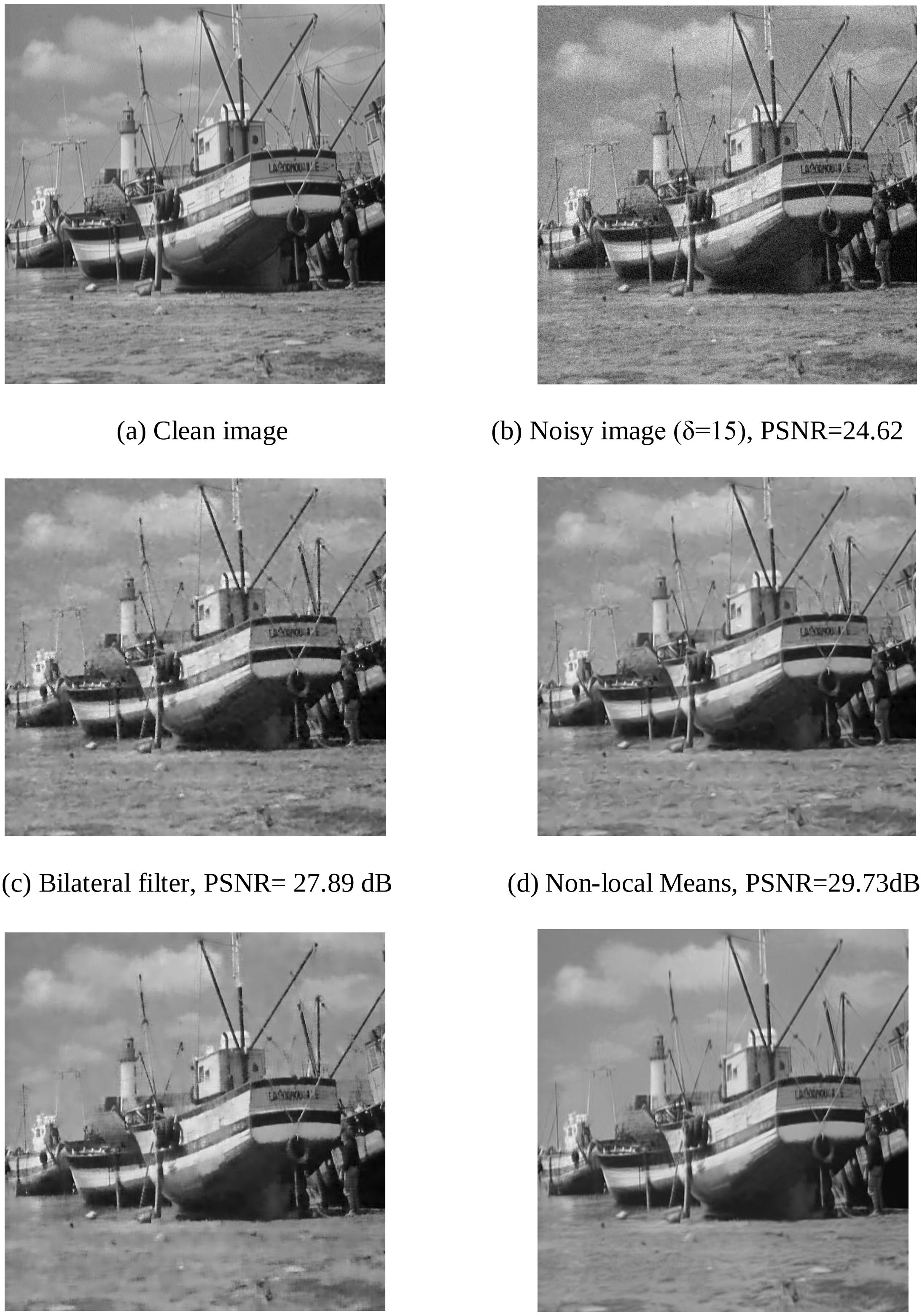
(c) Bilateral filter, PSNR= 29.03 dB

(d) Non-local Means, PSNR=30.12 dB

(e) SKR filter, PSNR=31.33dB

(f) Proposed method, PSNR=31.53dB

Figure 4.5: Visual and quantitative comparison of Lena image with δ=25 (a)Clean Lena image, (b) Noise-ridden image δ=25 (c) Bilateral filter (3.2) [25], (d) Non- Local Means (3.3) [26] [27], (e) Iterative SKR filter with number of iteration (N) =12 [41], (f) Proposed method with (N=12)



(a) Clean image

(b) Noisy image (δ=15), PSNR=24.62

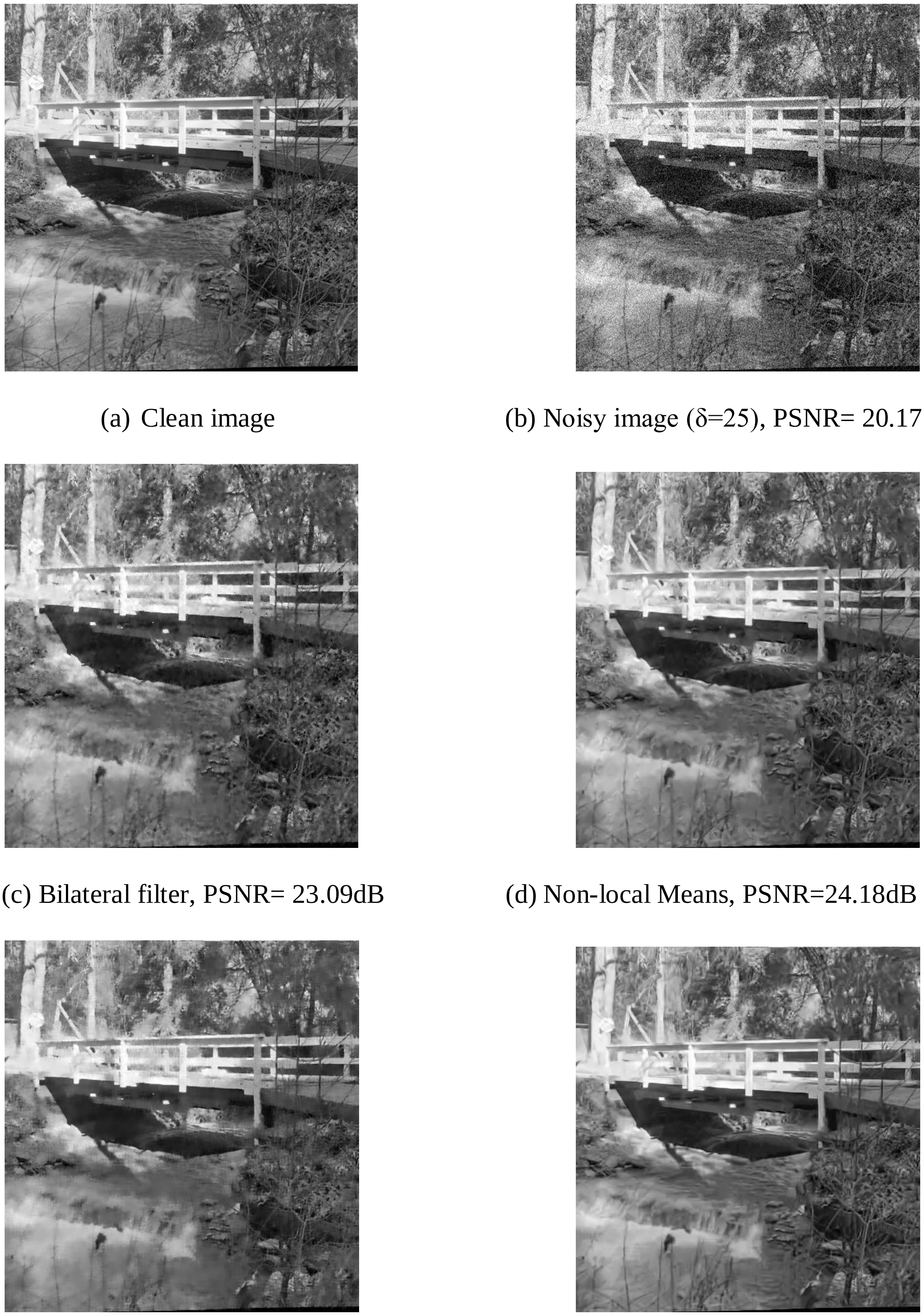
(c) Bilateral filter, PSNR= 27.89 dB

(d) Non-local Means, PSNR=29.73dB

(e) SKR filter, PSNR= 31.27dB

(f) Proposed method, PSNR= 31.527dB

Figure 4.6: Visual and quantitative comparison of Boat image with δ=15 (a)Clean boat image, (b) Noise-ridden image δ=15 (c) Bilateral filter (3.2) [25], (d) Non-local Means (3.3) [26] [27], (e) Iterative SKR filter with number of iteration (N) =12 [41], (f) Proposed method with (N=12)



(a) Clean image

(b) Noisy image (δ=25), PSNR= 20.17

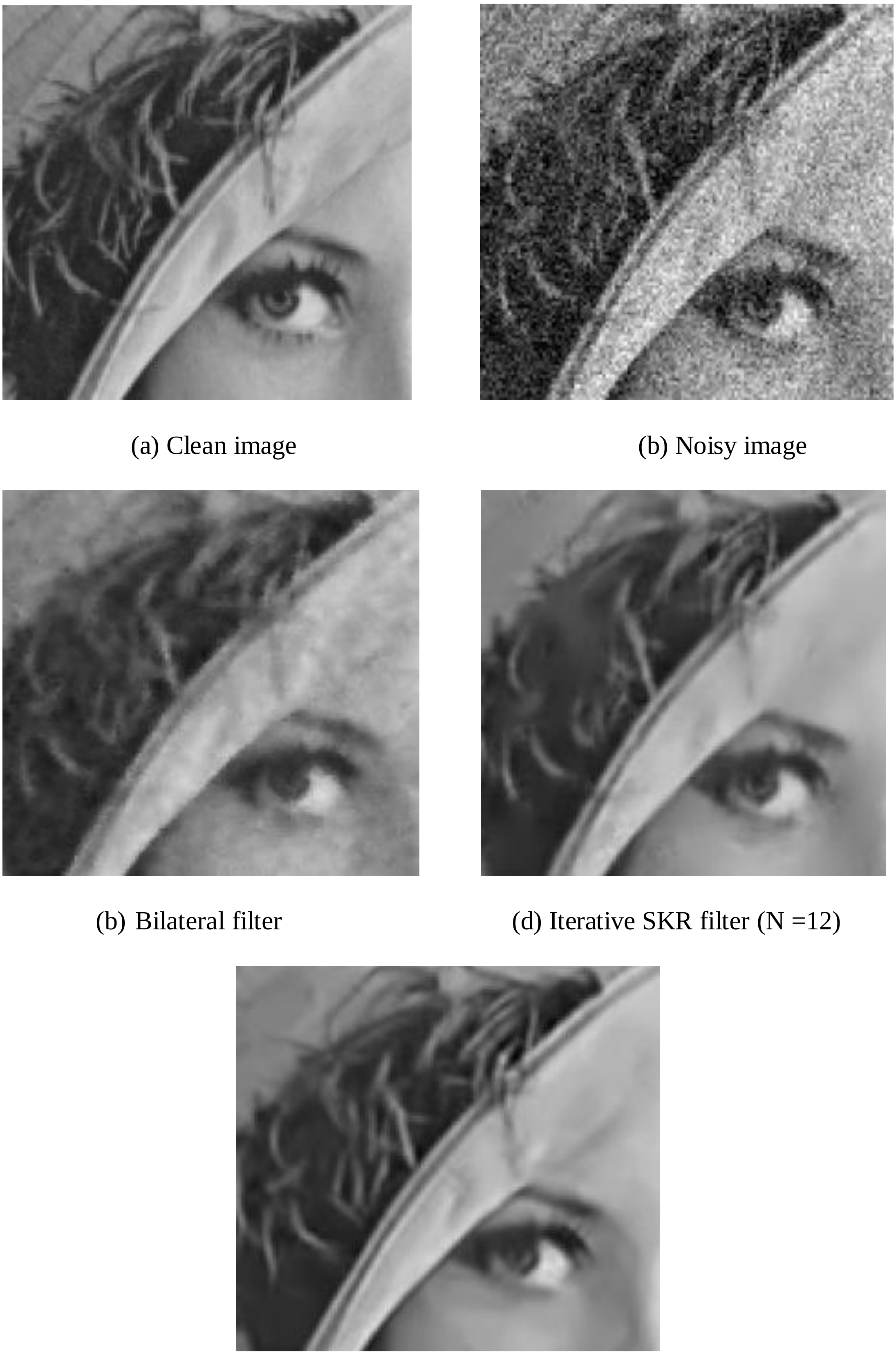
(c) Bilateral filter, PSNR= 23.09dB

(d) Non-local Means, PSNR=24.18dB

(e) SKR filter, PSNR=25.49dB

(f) Proposed method, PSNR= 25.76dB

Figure 4.7: Visual and quantitative comparison of Stream image with δ=25 (a) Clean stream image, (b) Noisy image δ=25 (c) Bilateral filter (3.2) [25], (d) Non-local Means (3.3) [26] [27], (e) Iterative SKR filter with number of iteration (N) =12 [41], (f) Proposed method with (N=12)



(a) Clean image

(b) Noisy image

(b) Bilateral filter

(d) Iterative SKR filter (N =12)

(e) Proposed method (N=12)

Figure 4.8: An example of WGN removal with the Lena enlarged selected regions (a) Clean, (b) Noisy, (c) Bilateral, (d) Iterative SKR and (e) Proposed images shown in (Figure 4.5)

Figure 4.4 (a) shows the parrot noise-ridden image with δ=30. As it is observable all of the edges become sharper in (f) which is the result by proposed algorithm compared to (c) Bilateral, (d) Non-local Means, but compare to SKR method strong edges and high different intensity become shaper in (f).

In Figure 4.5 and Figure 4.6 we compare denoising method for two well-known bench mark images, Lena and boat. The experiments show our result is more visually pleasing when compared to (c), (d) and (e) which introduce artifacts in the smooth regions of the filtered image. The proposed method approach strongly reduces these artifacts.

Figure 4.7 has many texture area and it makes subjective evaluations become challenging, but the proposed filter is found to be the best in filtering smooth and complex regions with quite little distortion and giving the best visual quality among all filters compared here.

Figure 4.8 shows the enlarged selected regions of the respective images shown in Figure 4.5, in this case artifacts in the smooth regions of filter (c) and (d) is shown better and density of these artifacts are higher than compare to our proposed kernel.

These comparisons prove that using the second order structure tensor in addition to performing the process more accurate to finding the edges, introduce less artifacts in the smooth regions of the filtered image.

Table 4.2 shows the denoising performance of some popular methods (Bilateral [24] [42], NLM [26], SKR[41], BM3D[43]) under WGN corruption, compared to proposed algorithm. Results noted are average PSNR with different standard deviations (δ), δ =15 (top), δ = 25 (middle) & δ = 50 (bottom) over 4 different images.

Table 4.2: Denoising performance of 5 different methods

methods image	Bilateral PSNR(dB)	NLM PSNR(dB)	SKR (12iteration) PSNR(dB)	BM3D PSNR(dB)	Proposed method (12iteration) PSNR(dB)
Barbara	29.87 25.15 22.77	31.63 27.97 24.13	32.41 29.93 25.96	33.09 30.67 26.65	32.72 30.11 26.18
Lena	30.68 29.03 25.53	32.43 30.12 24.41	33.54 31.33 27.94	33.90 31.92 28.49	33.71 31.53 28.09
bout	27.89 25.43 22.76	29.73 27.86 23.15	31.27 29.38 25.47	32.11 30.12 26.13	31.52 29.61 25.71
House	32.26 30.49 25.18	33.17 31.16 26.97	34.53 32.51 28.71	34.89 32.89 29.25	34.65 32.76 28.99

The summarized of the result are observable in table. Comparing the PSNR that reached from this table, we can see that our proposed algorithm is operating better than SKR [41] with an average PSNR increase of around 0.3 dB over different standard deviations and images. Although the performance in the proposed algorithm have been improved to the SKR method but with comparing to the BM3D [43] algorithm we can understand this proposed algorithm is improvable.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This work is concerned with the design of a filter for the denoising algorithm. Introducing the SKR structure and its properties lead to present the new steering matrix that hold on the Laplacian of pixels. The SKR filter is motivated by Hiroyuki Takeda work for improving the shape of the kernel in spatial domain. In this work we complemented Takeda’s approach with a new method of designing.

Considering the requirements of steering kernel regression such as gradient in X and Y axis and tensor structure, a new design of SKR filter via Laplacian in X and Y direction gives new tensor structure that adapt more complexity to the underlying image. This method result in sharper denoised images and increases PSNR by two or three percent depending on our sample image and standard deviation of the noise.

We can use this method for other purposes of image processing such as interpolation, super resolution and deblurring.

In order to enhance the operation of the kernel regression, we proposed a data-adaptive alternative to the classic kernel regression approach, where we discover the optimal filter coefficients from not only the spatial distances between the pixel of interest and its neighbors but also the photometric distances (differences in pixel intensity). This approach smoothed pixels along the local orientation structure. We generalized the bilateral filter [24] [25] by the non local mean filter [39], and proposed this algorithm and its iterative implementation which improves the filtering performance further. We showed the effectiveness of the proposed iterative algorithm for denoising.

5.2 Future work

In the future, the most important part of work which needs more investigation would be the model selection for choosing the kernel which penalized the spatial distance and photometric distance for neighboring pixels. In our work we used Gaussian kernel function but there are numerous choices for the kernel function K (xi −x) such as Epanechnikov, Biweight and triangle. By using the different kernel we approach the different result and optimizing this result would be finding the best choice for selecting the kernel. Table 5-1 shows the different choice of kernel.

One other important issue that can be worked on it in future is a possibility of improving the performance of kernel regression by the choice of the distance metric. The distance metrics are classified into two types: non-adaptive and adaptive. Minkowski, Manhattan, and Euclidean (l_2,l_1 and l_p-norm, respectively) are classified as non-adaptive distance metrics.

These metrics are definitely most consequential when their choice matches the statistical distribution of the data being treated. For example, when the data are corrupted by white Gaussian noise, Euclidean distance is an appropriate choice.

It should be noted that, in our method approach, we compute the weight values based on the distances between pixels measured by a data-adaptive metric.

Table 5.1: Different choice of kernels

name	Kernel functions
Epanechnikov	$K(u) = \frac{3}{4}(1 - u^2)\, \mathbf{1}_{\{\lvert u\rvert \leq 1\}}$
Quartic **(biweight)**	$K(u) = \frac{15}{16}(1 - u^2)^2\, \mathbf{1}_{\{\lvert u\rvert \leq 1\}}$
Triweight	$K(u) = \frac{35}{32}(1 - u^2)^3\, \mathbf{1}_{\{\lvert u\rvert \leq 1\}}$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$

REFRENCES

[1] Scott E Umbaugh, “Computer Vision and Image Processing,” *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2239–2248, August 2008.

[2] L. Gagnon, F. D. Smaili, “Speckle Noise Reduction of Airborne SAR Images with Symmetric Daubechies Wavelets”, vol. 2, no. 6, pp. 1422-1424, 1996.

[3] H. Gudbjartsson and S. Patz , “The Rician Distribution Of Noisy MRI Data. Magnetic Resonance in Medicine ,” vol. 34, no. 6,pp 910-918, July 1995.

[4] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. “ Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp 600-612, 2004.

[5] Z. Wang, E.P. Simoncelli, and A.C. Bovik. “Multiscale structural similarity for image quality assessment,” *In Conference of the Thirty-Seventh Asilomar on Signals,* volume 2, pages 1398-1402. April, 2003.

[6] J. Weickert. “Anisotropic diffusion in image processing,” *ECMI Series,* Teubner-Verlag, Stuttgart, Germany, 1998.

[7] H.R. Sheikh and A.C. Bovik. “Image information and visual quality,” *IEEE Transactions on Image Processing,* vol. 15, no. 2, pp 430-444, 2006.

[8] H.R. Sheikh, A.C. Bovik, and G. De Veciana, “An information_delity criterion for image quality assessment using natural scene statistics,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp 2117-2128, 2005.

[9] A.K. Moorthy and A.C. Bovik, “Blind image quality assessment: From natural scene statistics to perceptual quality,” *IEEE Transactions on Image Processing* vol. 20, no. 12, pp 3350-3364, 2011.

[10] P. Ye and D. Doermann, “No-reference image quality assessment using visual codebooks*,” IEEE Transactions on Image Processing,* vol. 21, no. 7, pp 3129-3138, July 2012.

[11] H. Tang, N. Joshi, and A. Kapoor, “ Learning a blind measure of perceptual image Quality,” *In International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 6, pp. 597–600, April 1997.

[12] M.A. Saad, A.C. Bovik, and C. Charrier, “Blind image quality assessment: A natural scene statistics approach in the dct domain,” *IEEE Transactions on Image Processing,* vol. 21, no. 8, pp. 3339-3352, April 2012.

[13] A. Mittal, A. Moorthy, and A. Bovik, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on Image Processing* vol. 17, no. 6, pp. 1043–1092, July 2009.

[14] A.K. Moorthy and A.C. Bovik, “A two-step framework for constructing blind image quality indices,” *IEEE Signal Processing Letters* , 2010.

[15] E. A. Nadaraya, “On estimating regression,” *Theory of Probability and its Applications*, pp. 141–142, September 1964.

[16] M. P.Wand and M. C. Jones, “Kernel Smoothing, ser. Monographs on Statistics and Applied Probability,” London; Chapman and Hall, January 1995.

[17] P. Yee and S. Haykin, “Pattern classification as an ill-posed, inverse problem: a regularization approach,” *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, ICASSP, vol. 1, pp. 597–600, April 1993.

[18] H. Knutsson and C. F.Westin, “Normalized and differential convolution – methods for interpolation and filtering of incomplete and uncertain data,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Regocnition* (CVPR), pp. 515–523, June 1993.

[19] T. Q. Pham, L. J. van Vliet, and K. Schutte, “Robust fusion of irregularly sampled data using adaptive normalized convolution,” *EURASIP Journal on Applied Signal Processing,* Article ID 83268, January 2006.

[20] X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Transactions on Image Processing,* vol. 10, no. 10, pp. 1521–1527, October 2001.

[21] N. K. Bose and N. Ahuja, “Superresolution and noise filtering using moving least squares,” *IEEE Transactions on Image Processing,* vol. 15, no. 8, pp 186-204.

[22] E. A. Nadaraya, “On estimating regression Theory of Probability and its Applications,” pp. 141–142, September 1964.

[23] S. Osher and L. I. Rudin, “Feature-oriented image enhancement using shock filters,” *SIAM Journal on Numerical Analysis*, vol. 27, no. 4, pp. 919–940, 1990.

[24] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” Proceeding of the 1998 IEEE *International Conference of Compute Vision*, Bombay, India, pp. 836–846, January 1998.

[25] M. Elad, “On the origin of the bilateral filter and ways to improve it,” *IEEE transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1150, 2002.

[26] A. Buades, B. Coll, and J. Morel, “A non-local algorithm for image denoising,” *IEEE international conference on computer vision and pattern recognition*, India, 2010.

[27] A. Buades, B. Coll, and J. Morel, “On image denoising methods,” Technical Report 2004-15, CMLA, July 2004.

[28] X. Feng and P. Milanfar, “Multiscale principal components analysis for image local orientation estimation,” *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2002.

[29] K. V. Mardia, J. T. Kent, and J. M. Bibby, “ Multivariate Analysis,” London , New York: Academic Press, 1979.

[30] A. Edelman, “Eigenvalues and condition numbers of random matrices*,” SIAM journal on Matrix Analysis and Aplications*, vol. 9, pp. 543–560, 1988.

[31] M. Charest and P. Milanfar, “On iterative regularization and its application,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 406–411, March 2008.

[32] X. Zhu and P. Milanfar, “Automatic parameter selection for denoising algorithms using a no-reference measure of image content,” *submitted to IEEE Transactions on Image Processing*, 2009.

[33] H. Takeda, H. Seo, and P. Milanfar, “Statistical approaches to quality assessment for image restoration,” *Proceedings of the International Conference on Consumer lectronics*, January 2008.

[34] K. Q. Weinberger and G. Tesauro, “Metric learning for kernel regression,” *Proceedings of the Eleventh InternationalWorkshop on Artificial Intelligence and Statistics*, pp. 608–615, July 2007.

[35] J. van deWeijer and R. van den Boomgaard, “Least squares and robust estimation of local image structure,” *Scale Space International Conference*, vol. 2695, no. 4. 161–184, September 2009.

[36] H. Takeda, “Locally adaptive kernel regression methods for multi-dimensional signal processing,” *Ph.D. dissertation*, University of California, Santa Cruz, September 2010.

[37] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising methods, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530.

[38] P. Chatterjee and P. Milanfar, “Clustering-based denoising with locally learned dictionaries,” *IEEE Transactions on Image Processing, vol*. 18, no. 7, pp. 1438, 2009.

[39] T. Brox, O. Kleinschmidt, and D. Cremers, “Efficient nonlocal means for denoising of textural patterns*,” IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1083–1092, July 2008.

[40] D. D. Muresan and T. W. Parks, “Adaptive principal components and image denoising,” *in Proceedings of IEEE International Conference on Image Processing*, vol. 1, no. 8, pp. 101–104, Barcelona, Spain, September 2003.

[41] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349– 366, February 2007.

[42] Nilamani Bhoid “evelopment of Some Novel Spatial-Domain and Transform-Domain Digital Image Filters,” *IEEE Transactions on Image Processing*, vol. 17, no. 20, pp. 145–171, July 2010.

[43] Harold C. Burger, Christian J. Schuler, and Stefan HarmelingImage, “Plain Neural Networks Compete with BM3D,” *Max Planck Institute for Intelligent Systems*, Tubingen, Germany.

[44] Xiaofei He, Jiajun Bu, Chun Chen and Ziyu Guan “Image representation using Laplacian regularized nonnegative tensor factorization,” *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 112– 153, 2007

[45] R.D. Kriz, M. Yaman, M. Harting and A.A. Ray4 “Visualization of Zeroth, Second, Fourth, Higher Order Tensors, and Invariance of Tensor Equations,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 649– 776, February 2009