USING GLOBAL CONSISTENCY CONSTRAINTS FOR DE-NOISING OF NATURAL IMAGES

A Project Report

Submitted in partial fulfilment of the

requirements for the Degree of

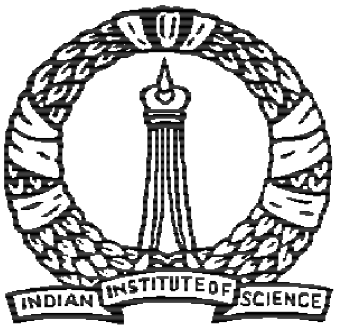
Master of Engineering

in

Faculty of Engineering

by

Pai Gautam Pravin



Electrical Engineering

Indian Institute of Science

BANGALORE – 560 012

JUNE 2013

Acknowledgements

I begin by expressing heartfelt gratitude to my advisor Venu Madhav Govindu for his selfless support and guidance all through the two years of my studies at IISc. It has been a pleasure to have worked under him and I have benefited hugely from his advice and tutelage. I am infinitely indebted to each and every one of my professors at IISc for the marvelous coursework and teaching. Their dedication towards their work is something I shall always remember and look up to. I thank all my fellow batch-mates, colleagues, lab-mates and friends: AjayRama, Abhiraam, Hari, Suraj, Mohammed and Krishanu for making these two years a very enriching experience indeed. And finally, I owe everything to my parents: Pravin and Maya for their unconditional love and support from the very beginning.

Abstract

De-noising of images is a classical problem in the Computer Vision and Image Processing literature and there have been a plethora of different algorithms developed for this ill-posed problem. With the advent of the non-local means *principle by Buades, most of the recent approaches have been of a non-parametric and data-adaptive nature. The development comes from the fact that images, and natural images in particular, show strong statistical redundancy in the behavior of its smaller units - patches, which have a strong tendency to recur across the same image. In this report, we propose to maximally exploit the geometric relationships that exist between the pixels of a natural image by introducing a notion of* consistency *in the multidimensional, non-parametric, and data adaptive filtering approach. We show that this notion of consistency can be viewed as a controlled diffusion on a graph constructed from the image. Finally, we propose a regularization framework for image de-noising using these ideas and show how enforcing consistency improves de-noising by increasing the connectivity of the graph built from very few neighbors.*

Contents

List of Tables

List of Figures

Chapter 1

Motivation

De-noising is a fundamental problem in any signal processing framework. Electronic systems process various kinds of different signals: audio, images, video, power, etc. and in most cases, by the virtue of either their processing or their acquisition, the signals get corrupted and this may lead to an erroneous functioning of the entire system. Hence de-noising or removal of the corruption is essential for a sturdy and robust signal processing system.

Imaging systems are no different to this scenario. Most modern imaging systems are digital where the acquisition of the signal occurs by the aggregation of light photons on a charged couple device (CCD). When photons strike the CCD film, depending to the wavelength and other factors, the CCD registers an image and this image formation is susceptible to noise perturbations where the magnitude of perturbation is inversely proportional to the number of pixels packed on the film. With the advent of high resolution camera's, more and more pixels are packed on the same unit area of the film and hence the image becomes more susceptible to noise.

Hence, de-noising can be dealt with from a systems side where the electronics of the imaging device are made less sensitive to noise which will lead to a higher cost of construction, or by using sophisticated de-noising algorithms to recover the image acquired from a lesser quality electronic system.

From an academic viewpoint, de-noising is a signal estimation problem where the

quality of estimation depends largely on our understanding of the structure and properties of the signal and the noise. Hence de-noising of images is still a pursued problem in Computer Vision and Image Processing because it involves building accurate models for natural images which is one of the largest open problems in low-level vision. These models can then be extended to other image processing tasks such as image de-blurring, super-resolution, inpainting etc.

Chapter 2

Introduction

2.1 Problem Formulation

The image perturbation can be modeled as follows, for all the pixels $i = 1 : MN$ in a $M \times N$ image, we have:

$$y_i = z_i + n_i \tag{2.1}$$

where z_i : true image intensity, y_i : observed image intensity, and n_i : perturbation or noise at pixel i. Although the image is a two-dimensional signal, for ease of representation we have used a single index - **i** for notation as we can simply vectorize the 2-D matrix into a single row or column and treat it as a vector.

We assume that the noise perturbation at each pixel is distributed normally with zero mean and finite variance, and that the noise is isotropic and identical throughout the image, i.e. in vector form

$$Y = Z + N \;\; where \;\; N \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right) \tag{2.2}$$

From equation (2.2) we can see that the problem is ill-posed, since we can have infinite combinations of image Z and noise N which yield the same observation Y. The de-noising problem is to estimate Z given only Y.

2.2 A Bayesian Estimation Perspective

All de-noising approaches have two parts: a noise model and a signal model. The noise model is the set of assumptions made on the statistics of the perturbation. Equations (2.1) and (2.2) clearly define the noise model for image de-noising. The image model correspondingly are the set of assumptions and beliefs made on the statistics of the behavior of the image.

Using a Bayesian view of estimation, the posterior $p(Z|Y)$ is written as:

$$p(Z|Y) \ \propto \ p(Y|Z) \ . \ p(Z) \tag{2.3}$$

now $p(Y|Z)$ is known given that we have clearly defined our noise model.

$$p(Y|Z) = \frac{1}{\sqrt{2\pi MN\sigma^2}} \, e^{\frac{-||Y-Z||^2}{2\sigma^2 MN}} \tag{2.4}$$

What remains to be evaluated is the image model $p(Z)$ in order to construct a MAP or an MMSE estimator. Thus the Bayes estimation perspective requires that we construct an accurate image model p(Z) for the domain of natural images.

There have been many attempts to build such models for natural images. [1, 2] are some examples where explicit models were built by observing the statistics of images and image gradients in particular. Although use of such models resulted in decent results as far as the gain in the signal to noise ratio is concerned, their performance was very far from the current de-noising algorithms like the BM3D[3] and the K-SVD [4]. The distinction comes from the fact that powerful parametric global models for natural images are hard to find and hence the de-noising performance is bounded by their ability to accurately describe natural images.

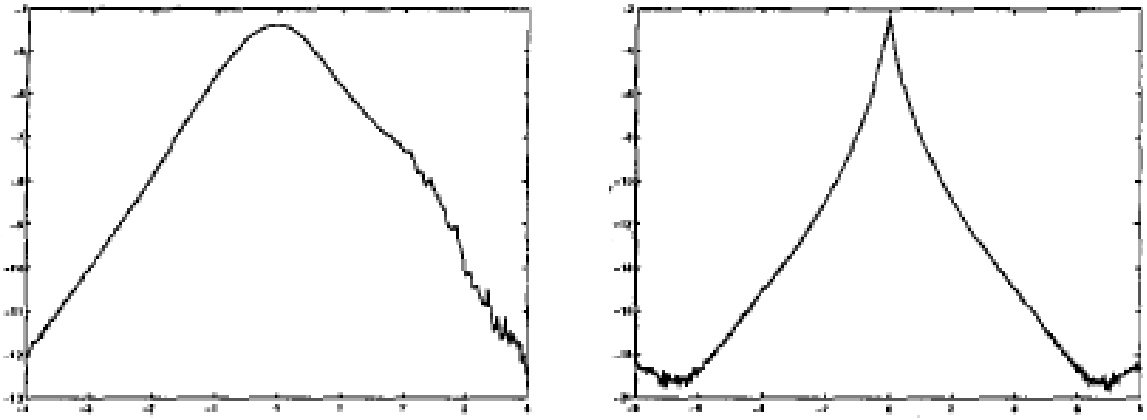


Figure 2.1: **Left:** log-histogram of $ln(I(i,j)) - ln(mean(I))$. **Right:** log histogram of $ln(I(i,j)) - ln(I(i,j+1))$. Figure taken from Statistics of Natural Images and Models, Huang and Mumford [1]

2.3 Image Statistics

Distinguishing signal from noise requires that we have some knowledge about both signal and noise. The noise and image models represent our knowledge about the noise and image although they are of a statistical nature. As mentioned previously, the early attempts on characterizing natural images was by observing the statistics of its gradients. The important observation in these papers was that the distribution of the magnitude of gradients of natural images resembles the generalized laplacian distribution which lead to the use of image priors p(Z) of the form:

$$p(d) = \frac{e^{-|\frac{d}{c}|^p}}{Z(c,p)} \tag{2.5}$$

where $|d(x) = \nabla Z(x)|$, the gradient at point x.
Later studies also involved the ensemble statistics of the wavelet co-efficients of natural images which show high kurtosis (a measure of peakedness of the distribution) and heavy tails. However most of these attempts failed to generate strong models for the joint statistics for a large number of pixels.

With the advent of Buades [6] seminal paper on the non-local means principle, study

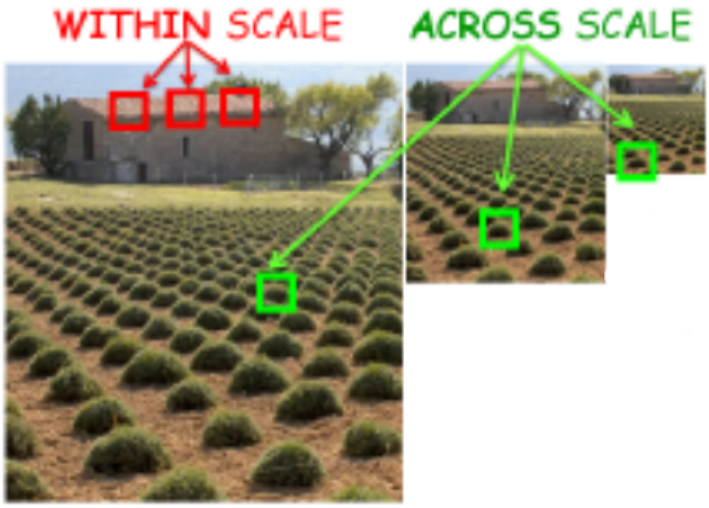


Figure 2.2: Demonstrating Patch Recurrence in Images, Figure from Internal Statistics of a Single Natural Image, Zontak and Irani [5]

of natural image statistics had a new perspective: self similarity. Instead of studying statistics of images as a whole, the image can be broken down into smaller units- patches and statistics of these patches is what all of the recent and state-of-the-art algorithms exploit.

It is observed that, patches in a natural image have a strong tendency to recur in the same image and at different rotations and scales. Michal Irani's paper [5] shows the statistics of these patch recurrences. Patches which have low frequency content (smooth regions, small edges etc.) are very likely to recur in nearby regions of the image whereas the patches with decent texture content have a lesser chance and may recur with larger distances amongst themselves.

Thus, the signal (image) can be broken down into smaller units and the large correlation of these units is exploited to have a best guess of the underlying image. The estimation is now done on a patch scale where similar looking patches are individually processed, de-noised and the image is then reconstructed out of these de-noised patches.

2.4 A Filtering Perspective

The Bayesian method of estimation requires that a prior be specified for natural images and the final estimator will depend on the nature of this prior. However, various linear filtering schemes have been proposed for de-noising without paying too much attention on what explicit model does the filter assume on the statistics of natural images.

All of these methods propose an estimator where the de-noised estimate of every pixel can be written as a linear combination of other pixels . A simple example is the moving average filter where the estimated pixel intensity is an arithmetic average of the intensity of its neighborhood pixels.

$$\hat{z}_i = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} y_j \ , \tag{2.6}$$

where $\mathcal{S}$ is the neighborhood of pixel i.
The moving average filter assumes that the image intensity remains constant in neighborhoods of fixed size. This filter would perform best for regions of the image where the intensity is constant and there are no abrupt changes in image intensity. For regions like edges and texture, the moving average yields poor results because the estimate will smooth over the edges, texture and will result in loss of signal detail.

A simple way to improve the filter is to incorporate a weighted average, where the weights could be a function of the distance between the central pixel.

$$\hat{z}_i = \frac{1}{Z} \sum_{j \in \mathcal{S}} e^{\frac{-(i-j)^2}{2\sigma^2}} \ y_j \ , \tag{2.7}$$

Where Z is a normalization constant to ensure that the dynamic range of the signal remains within limits.

With the bi-lateral filter [7], the scheme of assigning weights of the linear filter became more sophisticated and the de-noising performance also improved. The bilateral filter has two terms in its weighting function: the space term just like the Gaussian smoothing filter and also an intensity term, penalizing pixels having intensities different than that

of the central pixel.

$$\hat{z}_i = \frac{1}{Z} \sum_{j \in \mathcal{S}} e^{\frac{-(i-j)^2}{2\sigma_x^2} + \frac{-(y_i - y_j)^2}{2\sigma_I^2}} y_j \tag{2.8}$$

Thus, pixels closest to the central (to be estimated) pixel in both space and intensity, affect it the most. Clearly this scheme of filtering does much better than its predecessor's as far as edges and strong textures are concerned, since pixels across an edge will not affect each other.

However, the bilateral filter still requires *similar* pixels to be close by. The Non-Local means filter does away with this term and the weights are now totally dependent on the neighborhood of the pixel - a patch.

$$\hat{z}_i = \frac{1}{Z} \sum_{j \in \mathcal{S}} e^{\frac{-||\mathbf{P_i} - \mathbf{P_j}||^2}{2\sigma^2}} y_j \ , \tag{2.9}$$

where $\mathbf{P_i}$ and $\mathbf{P_j}$ are the patches surrounding pixels i and j respectively. This filter epitomizes most non-parametric filtering schemes: The signal is broken down into smaller units - *patches* and the de-noised estimate uses the high correlation amongst these similar patches to define the output. The notion of similarity is now *similar neighborhoods* in the sum-of-squared error sense and the implicit assumption is that, this notion similarity-identical neighborhoods remains fairly robust to presence of noise i.e. even in the presence of noise, pixels with identical or close neighborhoods can be identified more reliably rather than comparing their individual intensities.

In practice however, the robustness of the weights in (2.9) depend on the noise level. For low noise levels the weights remain more or less the same, and change considerably with increasing perturbation. However, the non-local means weighting is much more robust than its predecessor's to increasing levels of noise, which is the reason why most state-of-the art algorithms choose to use it in principle for patch de-noising.

Chapter 3

A Unified Framework

3.1 The Weighted Least Squares Problem

Most of the non-parametric, data-adaptive filters, in the last section can be considered as solutions to a weighted least squares problem [8].
Thus, for all pixels $i = 1 : MN$, the de-noised estimate can be written as:

$$\hat{z}_i = \underset{z}{argmin} \sum_{j=1}^{j=MN} [y_j - z]^2 \, . \, K(i,j) \tag{3.1}$$

where as per convention, y_i is the noisy observation and $\hat{z}_i$ is the de-noised estimate at pixel i. $K(i,j)$ is a kernel which signifies the notion of *similarity* between pixels i and j. The solution to the above problem is obtained by simply differentiating the cost function with respect to z and equating to zero, we have :

$$\hat{z}_i = \sum_{j=1}^{j=MN} \frac{K(i,j)}{\sum_{j=1}^{j=MN} K(i,j)} \; y_j \tag{3.2}$$

$$\hat{z}_i = \sum_{j=1}^{j=MN} W(i,j) \; y_j \tag{3.3}$$

where

$$W(i,j) = \frac{K(i,j)}{\sum_{j=1}^{j=MN} K(i,j)} \tag{3.4}$$

Clearly, the weights in Equation (3.4) suggest that the estimate is a convex combination of the input i.e. if $K(i,j) \geq 0$ for all i,j then:

$$W(i,j) \geq 0 \tag{3.5}$$

$$\sum_{j=1}^{j=MN} W(i,j) = 1 \tag{3.6}$$

Equation (3.3), can be interpreted as an inner product of the noise image vector Y and a vector $\mathbf{w(i)}$:

$$\mathbf{w(i)} = [W(i,1), W(i,2), W(i,3), \ldots W(i,MN)]$$

then,

$$\hat{z}_i = \mathbf{w(i)}^T Y \tag{3.7}$$

Stacking the estimates for each pixel , in vector form (3.8) becomes:

$$\widehat{Z} = W\ Y \tag{3.8}$$

where, as per convention Y is the noisy image vectorized , $\widehat{Z}$ is the estimate vectorized and W is a stochastic matrix by virtue of 3.5 and 3.6. Row stochasticity is important in filtering since the dynamic range of the signal remains within limits despite any amount of repeated filtering.

Summarizing, the the unified framework [8], gives a nice interpretation to the filters: The estimate is a convex combination of the input where the weights are in proportion to a similarity kernel defined by $K(i,j)$.

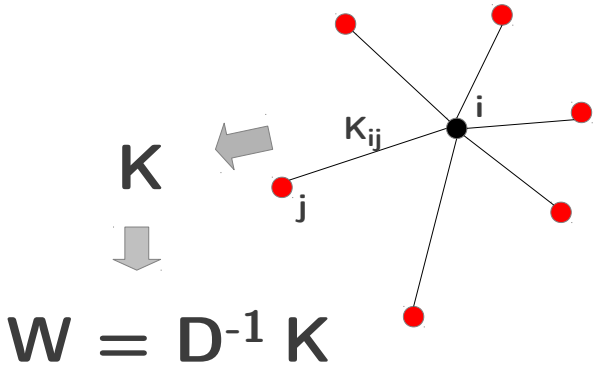


Figure 3.1: The Graph Perspective

3.2 The Graph Perspective

The weighted least squares problem (3.1) introduces a notion of similarity $K(i,j)$ based on which the matrix W is constructed. This motivates a graph interpretation of the entire filtering framework. Based on a notion of similarity defined by $K(i,j)$, we can populate the adjacency matrix of an undirected graph defined by considering each pixel as a node and the edges drawn as per $K(i,j)$.

For a given similarity kernel $K(i,j)$,

$$K = \begin{pmatrix} K(1,1) & K(1,2) & ... & K(1,MN) \\ K(2,1) & K(2,2) & ... & K(2,MN) \\ . & . & . & . \\ . & . & . & . \\ K(MN,1) & K(MN,2) & ... & K(MN,MN) \end{pmatrix}$$

define diagonal matrix D as:

$$D = \begin{pmatrix} D(1,1) & 0 & ... & 0 \\ 0 & D(2,2) & ... & 0 \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & ... & D(MN,MN) \end{pmatrix}$$

where, $D(i,i) = \sum_{j=1}^{j=MN} K(i,j)$
then the matrix W can be written as:

$$W = D^{-1}\, K \tag{3.9}$$

Thus , the matrix W is the row normalized adjacency matrix of a graph defined by the kernel $K(i,j)$

3.3 Choosing the Right Kernel

Any de-noising technique will result in some or the other form of *averaging* or *smoothing* as shown by Equation (3.3). The idea is to average *intelligently* i.e. smooth only those regions where the underlying signal does not have discontinuities.

The problem however, is in identifying these regions like edges and texture in the presence of noise. From a spectrum analysis perspective, noise which is assumed to be Additive White Gaussian will contain all frequencies, but the spectrum of the image is governed by its characteristics like constant regions, edges etc. Constant regions correspond to the low frequency components and hence noise in these regions can easily be identified and subdued using low pass filters. However, in regions where the signal contains high frequency components like edges and random texture, distinguishing signal and noise becomes a problem.

Therefore, an adaptive strategy must be employed where the strength of averaging or smoothing is different for different regions of the image. For constant regions, the averaging must be rigorous i.e. the weights will be more or less the same for all pixels in (3.3). For edges the averaging is weak only in a particular direction (across the edge) and for texture, the averaging must be weak.

Identifying these different regions is done by choosing the appropriate kernel $K(i,j)$. The ideal kernel must have the ability to distinguish all kinds of regions - constant or otherwise even in the presence of noise. For the moving average filter, the kernel was a box function of the spatial distance between pixels which clearly is not a good

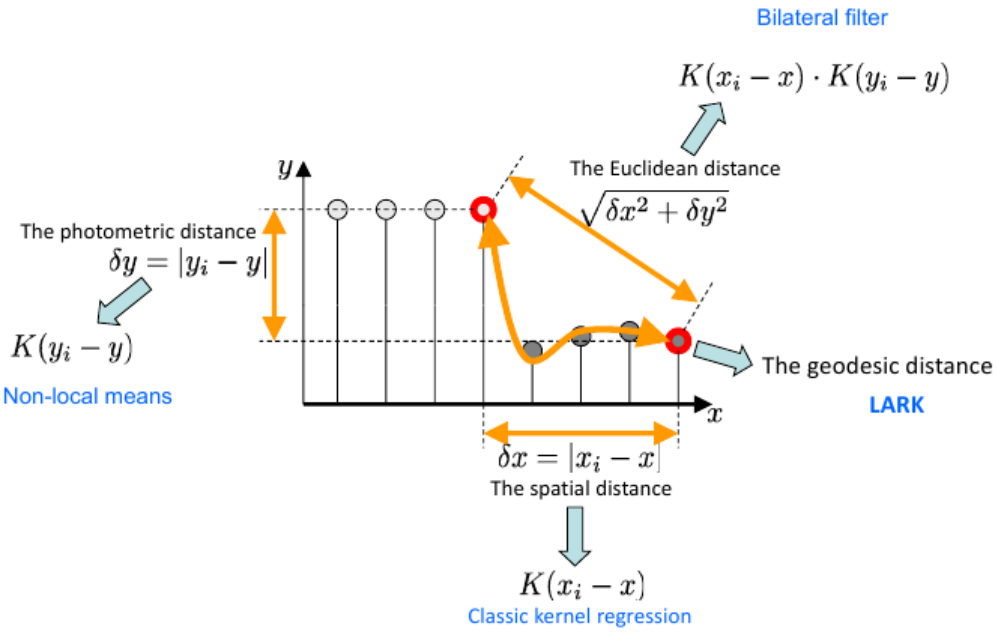


Figure 3.2: Summary of different kernels. Image borrowed from: A tour of Modern Image Filtering [8]

representative of the underlying image since pixels across an edge will be deemed similar by this kernel. With the bilateral filter the intensity term was added in addition to the space term. Although it does well for identifying strong edges, it is not very good as far as its robustness to noise is concerned and may not perform too well in constant regions with a moderate level of noise. With the non local means, the spatial term was removed completely and the intensity term was expanded to include the neighbors also which made it suitable not only for edges but also for repetitive texture patterns. In all our experiments, we have used the non-local means kernel as our similarity measure. However, these are not the only kernels that can be used for de-noising [9]. Any positive definite kernel which is a good representative of the underlying signal and robust to the nature of degradation can be used for improving the de-noising performance. Figure (3.2) summarizes the various kernels that may be chosen and their properties.

Chapter 4

Global Consistency Constraints

4.1 A Notion Of Consistency

Most non-parametric filtering approaches can be simplified to the form of equation (3.8) where based on a kernel, the estimate is a linear function of the noisy image. However, strictly speaking the filter is not a linear estimator, since the matrix W is also computed from the observation-Y.

$$\widehat{Z} = W(Y)\, Y \tag{4.1}$$

The matrix W as we have seen is motivated from the construction of a graph built out of the image using a kernel which has properties as defined in section 3.3. W encodes the *geometric relationships* existing among the pixels of the image, the nature of the relationship defined by the structure and form of the kernel. In the case of the non-local means kernel, pixels having similar neighborhoods will be identified and the graph edges amongst such pixels will be strong, which equivalently means that the averaging in (3.8) amongst these pixels will be rigorous.

However, the framework as posed in (3.8) does not put any constraint on the geometric relationships of the de-noised estimate, i.e. the graph structure of the estimate is independent of the graph structure of the true or the noisy signal.

The notion of *global consistency* stems from the fact that there is a need for estimates

of the form (3.8) to retain the graph structure despite the filtering step of (3.8). If a group of pixels are similar before the filtering, the must remain similar post filtering step thereby maintaining the structure of the graph. Formulating the constraint, if W encodes the relationships of the true signal, then the estimate $\widehat{Z}$ must satisfy:

$$\widehat{Z} = W\widehat{Z} \tag{4.2}$$

Equation (4.2) suggests that, the estimate must belong to a space where, any amount of further filtering must not alter it. This can be interpreted as the transformation of the signal onto a space where the signal and noise are completely separated.

$$\widehat{Z} = W\widehat{Z} = W^2\widehat{Z} = W^3\widehat{Z} = ... \tag{4.3}$$

We must be careful not to mis-interpret Equation (4.3). We are **not** looking for a filter that has no effect on the noisy image, rather the *oracle* filter or graph W is given to us and we require that the estimate obey the geometric relationships as encoded in this graph. How to obtain this true graph structure given only the noisy image is what we address in section 5.2, but for now given the matrix W which encodes the geometric relationships of the desired estimate, we have the de-noising problem formulated as:

$$\widehat{Z} = \underset{Z}{argmin}\ \ ||Z - Y||^2 \ \ subject\ to\ \ Z = WZ \tag{4.4}$$

Thus, in the space of all estimates satisfying (4.2), we choose the one closest to the noisy signal, alternately we believe natural images to lie on a lower dimensional manifold which we locally characterize using the graph W. Noising an image will put it off the manifold and hence to estimate it, we choose the point on this manifold which is closest to the noisy observation.

4.2 Connections to Spectral Graph Theory

Looking closely at our consistency constraint again:

$$\begin{aligned} \widehat{Z} &= W\,\widehat{Z} \\ (I-W)\widehat{Z} &= 0 \end{aligned} \tag{4.5}$$

In Spectral Graph Theory, the matrix $(I-W)$ is one of the various versions of the Graph Laplacian

$$L = I - W = I - D^{-1}K \tag{4.6}$$

There are many different versions of the Graph Laplacian and each one has its own set of properties which characterize the properties of the graph. Laplacians are used extensively in spectral clustering [10] since their spectrum encodes the *"clusterness"* of the graph. We will see that in the case of the global consistency constraint a similar interpretation holds true. Rewriting our global consistency constraint in terms of the Graph Laplacian:

$$L\,\widehat{Z} = 0 \tag{4.7}$$

Thus, global consistency requires that the estimate lie in the null space of the Graph Laplacian L. Although L has many properties, we enumerate the important ones below which allow us to better understand what global consistency means.

1. *The multiplicity k of the eigenvalue 0 of L equals the number of connected components* $\{A_1, ..., A_k\}$ *in the graph.*

2. *The eigenspace of eigenvector 0 of L is spanned by the indicator vectors* $\mathbf{1_{A_1}}, \mathbf{1_{A_2}}, \mathbf{1_{A_3}}, ...\mathbf{1_{A_k}}$ *for each of these components.*

In effect the consistency constraint requires that the estimate $\widehat{Z}$ be a linear combination of the indicator vectors of each component class of the graph,

$$i.e.\ \ \widehat{Z} = \alpha_1\,\mathbf{1_{A_1}} + \alpha_2\,\mathbf{1_{A_2}} + \alpha_3\,\mathbf{1_{A_3}} \cdots + \alpha_k\,\mathbf{1_{A_k}} \tag{4.8}$$

Equation (4.8) gives us a clear picture of the de-noising strategy: we first cluster the image pixels into component classes where pixels in each component class have identical or similar neighborhoods, and (4.8) ensures that all of them must have the same intensity given by the corresponding weight α_i. The weights for each component class $\{\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_k\}$ are chosen such that the error $||\widehat{Z} - Y||^2$ is minimum which is exactly what (4.4) requires.

4.3 Spectral Analysis of the Filters

We have posed the de-noising strategy as an objective function in (4.4) without actually probing the nature of the graphs W built from natural images. A little more analysis on the matrices like W in (3.8) is required to assess the working of the consistency constraints. Writing equation (3.9) again:

$$\begin{aligned} W &= D^{-1}\, K \\ &= D^{-\frac{1}{2}}\, \{D^{-\frac{1}{2}}\, K\, D^{-\frac{1}{2}}\}\, D^{\frac{1}{2}} \\ \therefore\ W &= D^{-\frac{1}{2}}\, \mathcal{K}\, D^{\frac{1}{2}} \end{aligned} \tag{4.9}$$

where, $\mathcal{K} = D^{-\frac{1}{2}}\, K\, D^{-\frac{1}{2}}$

W is a non-negative, positive definite, stochastic matrix. Non-negativity comes from the fact that the kernel matrix K and diagonal matrix D are both non-negative. (3.5) and (3.6) justify the row stochasticity. From (4.9) we see that W is similar to a symmetric positive definite matrix $\mathcal{K}$ and hence W and $\mathcal{K}$ share the same eigenvalues.

It interesting to probe the spectra of our filtering matrix W, and since W and $\mathcal{K}$ share the same spectrum we can explore the eigen-structure of $\mathcal{K}$. It turns out the matrix $\mathcal{K}$ is related to one other version of the Graph Laplacian popularly known as the *Normalized*

Graph Laplacian:

$$L_{sym} = I - D^{-\frac{1}{2}} K D^{-\frac{1}{2}} \tag{4.10}$$

$$L_{sym} = I - \mathcal{K} \tag{4.11}$$

Writing the eigen-value equation for L_{sym}, we have:

$$\begin{aligned} L_{sym}x &= \alpha x \\ \therefore\ x - \mathcal{K}x &= \alpha x \\ \therefore\ \mathcal{K}x &= (1-\alpha)x \end{aligned}$$

Therefore, if x is an eigenvector of L_{sym} with eigen-value α, then it is also an eigen-vector of $\mathcal{K}$ with the corresponding eigenvalue $\mu = 1 - \alpha$.

Now, L_{sym} is symmetric and positive semi-definite [10] which means the eigenvalues of L_{sym} are real and greater than or equal to zero. Therefore, the eigen-values of $\mathcal{K}$ are bounded above by 1. They are also bounded below by zero since $\mathcal{K}$ is symmetric and positive semidefinite. Therefore if $\{\mu_1, \mu_2 \dots \mu_n\}$ are the eigenvalues of $\mathcal{K}$ (and equivalently W) then, for all i:

$$0 \leq \mu_i \leq 1 \tag{4.12}$$

We now know that, matrices of the form (3.8) have their spectrum bounded in the [0,1] region. The spectrum of the Graph Laplacian L of equation (4.6) is also bounded in [0,1] which can be verified using the same arguments. We see that just like the case of $\mathcal{K}$ and L_{sym} , the eigenvalues of W and L are also complements of each other, i.e. if λ is the eigenvalue of W the $(1 - \lambda)$ is the eigenvalue of the L corresponding to the same eigenvector. This complementary nature is in accordance with the fundamental properties of W and L. W is a smoothing matrix and hence behaves like a low-pass filter. The Laplacian as we know from image processing [11], behaves as a high-pass filter and therefore it should only be natural that their performance and hence spectra be complementary to one another. While W enhances low frequency components L

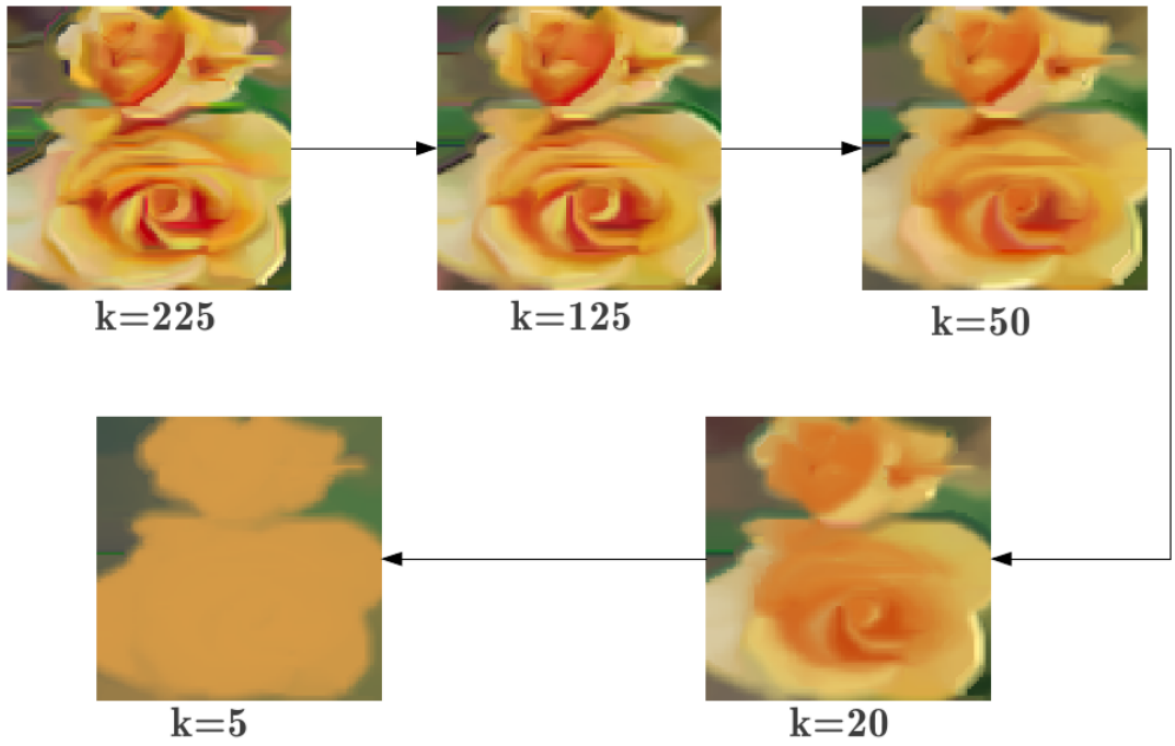


Figure 4.1: Projection of image onto the space spanned by the smallest "k" eigenvectors of the graph laplacian

enhances the higher ones as we can see in Figure (4.1)

An important point to note is that, since W is row stochastic, $\lambda = 1$ will be an eigenvalue of multiplicity at least 1, since the all ones vector $\mathbf{1}$ is always an eigenvector of W with eigenvalue 1.

$$W\mathbf{1} = \mathbf{1} \tag{4.13}$$

This means the the graph will always have one connected component. If eigenvalue $\lambda = 1$ has multiplicity more than one, then that multiplicity value will decide the number of connected components or clusters, and the eigenvectors for each of them will be the indicator vectors for each component class, which equivalently means that the multiplicity of eigenvalue 0 of L equals the number of connected components with the same set of eigenvectors.

Thus, the nature of the spectrum of W or L, determines the performance of the filter and also the redundancy of the underlying signal. If the spectrum dies of very quickly,

it means that the underlying image can be decomposed into groups of strong clusters, equivalently the filtering will be rigorous and de-noising performance will be good. A slow decay in the spectrum may suggest, weak clusters in certain regions of the image, and the filtering in these regions will be correspondingly weak.

We conducted an experiment where we compute the smallest (closest to zero) k eigenvectors of the Graph Laplacian and project our image onto this space. Figure (4.1) shows the results for some values of k. Clearly, larger values of k result in more detail being preserved. As suggested previously, the eigenspace of the laplacian forms a basis in which the image can be represented. The lowest eigenvectors (eigenvectors corresponding to the smallest eigenvalues) encode the strongest features of the image like strong edges and smooth regions. which also means that pixels in these regions will be strongly clustered. With increasing the eigenvalues the corresponding eigenvectors encode smaller details like small edges, texture etc. where the clustering in these groups becomes relatively weaker.

4.4 A Regularization Framework

Natural images in general contain constant regions, edges, repetitive textures etc, for which clustering is strong and also random fluctuations, small edges, texture etc where one may not find similar neighbors for every patch.

Therefore, the matrix W computed from the image will contain only one connected component - the entire image, which means that the output of (4.4) will be a constant image. Hence there is a need to relax the consistency constraint for those pixels which do not belong to any component class because the patches around them do not have good neighbors. The simplest way to do this is to use a penalty term for deviating from the consistency constraint:

$$\widehat{Z} = \underset{Z}{argmin} \;\; ||Z - Y||^2 + \lambda \, ||LZ||^2 \tag{4.14}$$

(4.14) is a quadratic cost function and may be easily solved by differentiating with respect

to Z

$$\begin{aligned} f(Z) &= ||Z - Y||^2 + \lambda \, ||LZ||^2 \\ \therefore \quad \nabla f(Z) &= 2(Z - Y) + 2(\lambda L^T L)Z \end{aligned} \tag{4.15}$$

Setting gradient to zero we get the system of equations:

$$[I + \lambda L^T L] \, \widehat{Z} = Y \tag{4.16}$$

$$\text{Let } C = I + \lambda L^T L$$

$$\therefore \; \widehat{Z} = C^{-1} \, Y \tag{4.17}$$

We started from the generic multidimensional filter of (3.8) and motivated the idea of consistency which any filter of the form of (3.8) must obey. We developed a regularization framework to enforce the consistency and in the process arrived at yet another linear filter in (4.17) just like (3.8). We now probe the nature of this new filter: $C = [I + \lambda L^T L]^{-1}$ to understand the difference in performance for both of them. Henceforth we refer the filter in (4.17) as **consistency filter**.

Consider the properties of $C = [I + \lambda L^T L]$, because the inverse does not have neat explicit form. Firstly, it is easy to see that C is symmetric and therefore C^{-1} will be symmetric as well. C is also a positive definite matrix for $\lambda \geq 0$, since: for any $x \neq 0, \; x \in \mathbb{R}^{\mathbb{MN}}$

$$x^T C x = x^t \{I + \lambda L^T L\} x = x^T x + \lambda x^T L^T L x = ||x||^2 + \lambda ||Lx||^2 > 0 \tag{4.18}$$

Now C being symmetric and positive definite, C^{-1} will also be positive definite.

Importantly, consider the left or right multiplication of C with the all ones vector $\mathbf{1}$.

$$C \; \mathbf{1} = [I + \lambda L^T L]\mathbf{1} = \mathbf{1} + \lambda L^T L \; \mathbf{1} \tag{4.19}$$

but, from (4.13): $W\mathbf{1} = \mathbf{1}$, i.e. $(I - W)\mathbf{1} = 0 \implies L\mathbf{1} = 0$

$$\therefore\ C\ \mathbf{1} = \mathbf{1} \implies C^{-1}\mathbf{1} = \mathbf{1} \tag{4.20}$$

Thus, C^{-1} is symmetric and has its rows and columns sum to 1, hence it shows properties of a *doubly stochastic matrix*. However, we have to check if C^{-1} is element-wise non-negative to confirm double stochasticity. Now

$$\begin{aligned} C &= I + \lambda\ L^T L \\ &= I + \lambda (I - W)^T (I - W) \end{aligned}$$

$$\therefore\ \ C = (\lambda + 1)I - \lambda(W + W^T - W^T W) \tag{4.21}$$

The matrix C in Equation (4.21) is of the form of a non-singular M-matrix [12], i.e. matrices which can be expressed in the form $sI - G$ where matrix $G = [g_{ij}]$ is such that $g_{ij} \geq 0\ \ \forall i\ \neq\ j$ *and* $\rho(G)\ <\ s$. The fundamental property of an M-matrix is that its inverse is proved to be element-wise non-negative [12, 13], Let $B = (W + W^T - W^T W)$.

Therefore, if C has to be a M-matrix then, $B = (W + W^T - W^T W)$ must satisfy:

$$b_{ij}\ \geq\ 0\ \forall\ i\ \neq\ j \tag{4.22}$$

$$\lambda\rho(B)\ <\ (\lambda + 1) \tag{4.23}$$

The first condition requires that,

$$w_{ij} + w_{ji} \geq \sum_p w_{jp} w_{pi}\ \ \forall\ i \neq j \tag{4.24}$$

We conducted a series of simulations to verify (4.24). We find that although (4.24) does not strictly hold true for all row stochastic matrices, for matrices of the form (4.9) (i.e. the set of row stochastic matrices similar to a positive definite matrix) which are large

and sparse (Ref section 5.1) (4.24) holds true. The second condition requires that:

$$\rho(B) < 1 + \frac{1}{\lambda} \tag{4.25}$$

Now since the matrix B is symmetric: its eigen-values will be real, and importantly, B is the complement of a positive definite matrix, i.e.

$$(I - W)^T(I - W) = I - B \tag{4.26}$$

which means that the eigen-values of B are upper bounded by 1 since eigen-values of $(I - W)^T(I - W)$ are lower bounded by 0

$$\therefore \;\; \rho(B) \leq 1 \tag{4.27}$$

and thus (4.25) holds true. Next we probe the spectrum of C and thereby C^{-1}, consider the eigen-value equation of C:

$$\begin{aligned} Cx &= \mu x \\ \therefore \; x + \lambda L^T L x &= \mu x \\ \therefore \; L^T L x &= \frac{(\mu - 1)}{\lambda}\, x \end{aligned}$$

But clearly $L^T L$ is a positive semidefinite symmetric matrix, since $x^T L^T L x = ||Lx||^2 \geq 0$, and therefore,

$$\mu - 1 \geq 0 \implies \mu \geq 1 \implies \mu^{-1} \leq 1 \tag{4.28}$$

Equation (4.28) along with the fact that C and C^{-1} are symmetric positive definite matrices implies that if $\{\gamma_1, \gamma_2 \ldots \gamma_n\}$ are the eigenvalues of C^{-1} then, for all i:

$$0 < \gamma_i \leq 1 \tag{4.29}$$

From (4.28) we can relate the singular values of L to that of C^{-1} as,

$$\sigma^2 = \frac{(\mu - 1)}{\lambda} \tag{4.30}$$

$$\therefore \ \mu^{-1} = \frac{1}{1 + \lambda\, \sigma^2} \tag{4.31}$$

Consider, the Singular Value Decomposition of the Laplacian L.

$$L = U\Sigma V^T \tag{4.32}$$

Now, in terms of the right-singular vectors of L,

$$\begin{aligned} C &= I + \lambda L^T L = I + \lambda\, V\Sigma^2 V^T \\ &= V\, \{I + \lambda\Sigma^2\}\, V^T \\ \therefore \ C^{-1} &= V\{I + \lambda\Sigma^2\}^{-1}\, V^T \end{aligned} \tag{4.33}$$

The matrix $[I + \lambda\Sigma^2]$ is a diagonal matrix where the i^{th} element is as per Equation (4.31), i.e.

$$\mu_i^{-1} = \frac{1}{1 + \lambda\, \sigma_i^2} \tag{4.34}$$

Equation (4.33) is the eigen-value decomposition of our filter matrix C^{-1}. Now orthonormal matrix V=[$\mathbf{v_2}$,$\mathbf{v_2}\ldots\mathbf{v_n}$] where the set $\{\mathbf{v_2}$,$\mathbf{v_2}\ldots\mathbf{v_n}\}$ are the right singular vector's of L then:

$$C^{-1} = \frac{1}{1 + \lambda\, \sigma_1^2}\, \mathbf{v_1}\mathbf{v_1}^T + \frac{1}{1 + \lambda\, \sigma_2^2}\, \mathbf{v_2}\mathbf{v_2}^T \cdots \frac{1}{1 + \lambda\, \sigma_n^2}\, \mathbf{v_n}\mathbf{v_n}^T \tag{4.35}$$

$$\therefore \ C^{-1} = \sum_{i=1}^{i=MN} \frac{1}{1 + \lambda\, \sigma_i^2}\, \mathbf{v_i}\mathbf{v_i}^T \tag{4.36}$$

Equation (4.36) gives a nice interpretation of the consistency filter. The filter can be expanded as a basis of orthonormal rank-1 matrices comprising of the self outer product of the right singular space of L. Since L is positive definite, the lowest (closest to 0) singular-vectors of L are nothing but the indicator vectors of each component class and

the corresponding singular value σ for such vectors is 0. Therefore, the first k values of the sum in (4.36) are nothing but the outer products of the indicator vectors of each component class, i.e.

$$\therefore\ C^{-1} \ = \ \frac{1}{|A_1|}\ \mathbf{1_{A_1}}\mathbf{1_{A_1}}^T + \cdots + \frac{1}{|A_k|}\ \mathbf{1_{A_k}}\mathbf{1_{A_k}}^T + \sum_{i=k+1}^{i=MN}\ \frac{1}{1+\lambda\ \sigma_i^2}\ \mathbf{v_i}\mathbf{v_i}^T \quad (4.37)$$

The matrices corresponding to the indicator vectors i.e. $\frac{1}{|A_1|}\ \mathbf{1_{A_1}}\mathbf{1_{A_1}}^T + \cdots + \frac{1}{|A_k|}\ \mathbf{1_{A_k}}\mathbf{1_{A_k}}^T$ will have all weights uniform in each component class, and they make no distinction in intra class weights. The matrices corresponding to the non-zero singular values correspond to the details in i.e. they will contain negative elements which show intra class differences. On the whole the penalty term λ controls the corresponding influence of each basis. Now the filtered estimate $\widehat{Z}$ can be written as:

$$\begin{aligned}
\widehat{Z} \ &= \ C^{-1}Y \\
&= \ \Big[\sum_{i=1}^{i=MN}\ \frac{1}{1+\lambda\ \sigma_i^2}\ \mathbf{v_i}\mathbf{v_i}^T\Big]Y \\
&= \ \sum_{i=1}^{i=MN}\ \left[\frac{\mathbf{v_i}^T Y}{1+\lambda\ \sigma_i^2}\right]\ \mathbf{v_i} \\
\therefore\ \widehat{Z} \ &= \ \sum_{i=1}^{i=MN}\ \beta_i(\lambda)\ \mathbf{v_i} \qquad (4.38) \\
where\ \ \beta_i(\lambda) \ &= \ \frac{\mathbf{v_i}^T Y}{1+\lambda\ \sigma_i^2} \qquad (4.39)
\end{aligned}$$

Equation (4.39) shows that the consistency filter can be written as a linear combination of the right singular space of the graph Laplacian L, with the weights as a function of λ. Thus depending on how sparse we assume our image to be in this transformed space, we can choose λ and hence choose only the first-k best vectors for expansion.

Equations (4.36) define the behavior of the consistency filters. Depending on the the kernel chosen, the behavior of multi-dimensional filters of the form (3.8) can been interpreted from the nature of the spectrum of such matrices. Equation (4.31) shows how we have a one-parameter (λ) control of the spectrum of consistency filter. If λ is

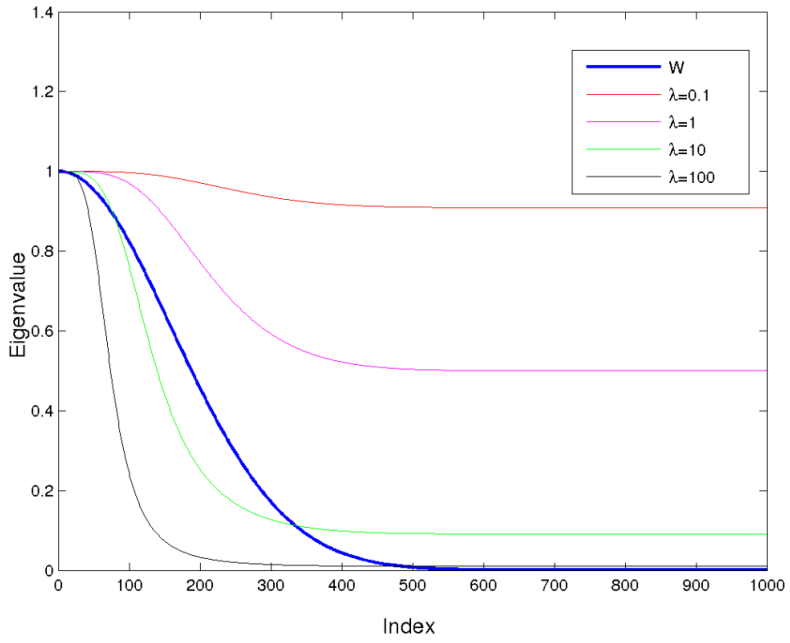


Figure 4.2: Spectrum of Consistency filter for different λ's

small then the spectrum of our consistency filter will be flat which means that there is no change in the output. If λ is large then, for all the non-zero singular values the output spectrum will die very rapidly. When the singular value is zero, the output eigenvalue will be 1, which suggests that making lambda very large will result in preserving only the component classes. If there is only 1 component class (only one zero-singular value) then the output matrix reduces to the self outer product of the all ones vector. If there is more than one component class, then the output will be the sum of the self outer-products of the indicator vectors of each component class.

$$\lim_{\lambda \to \infty} C^{-1} = \frac{1}{|A_1|} \mathbf{1_{A_1}} \mathbf{1_{A_1}}^T + \frac{1}{|A_2|} \mathbf{1_{A_2}} \mathbf{1_{A_2}}^T \dots \frac{1}{|A_k|} \mathbf{1_{A_k}} \mathbf{1_{A_k}}^T \tag{4.40}$$

Figure (4.2) compares the original filter W and consistency filter C^{-1} in (4.17). This plot was generated for the standard Gaussian kernel $K(i,j) = e^{\frac{-(i-j)^2}{2\sigma^2}}$. It demonstrates the fact that by choosing a larger λ we are in effect imposing that the image lie in a lower dimensional space where the cardinality of the space is determined by the number of connected components.

Equation (4.40) gives us a notion of diffusion. The consistency filter when constructed

for large λ's makes no distinction for pixels belonging to a group or cluster unlike its counterpart in (3.8) which fails to identify the global connectivity of similar pixels. Thus all inter-group weights are equal which means that the filter C^{-1} involves diffusing the pixel intensity over each cluster, which is similar in spirit to anisotropic diffusion as mentioned [14, 8].

Summarizing, imposing consistency on the multidimensional filters like in (3.8) has yielded yet another multidimensional filter like (3.8) with very special properties.

1. The consistency filter has a spectrum compacted in [0,1]. Thus the filter is stable and any amount of repeated iterations does not alter the dynamic range of the signal.

2. The consistency filter is symmetric and stochastic making it a special case of doubly stochastic matrices. Conceptually the symmetry signifies the fact that global similarity relationships are made symmetric, i.e. if pixel-i effects the estimate of pixel-j by α then pixel-j will also affect the estimate of pixel-i by the same amount, thus bringing the two pixels closer in the graph. [15] shows some of the advantages of doubly stochastic filters, and symmetrizing the ones which are not. This is in accordance with our notion of consistency which requires that geometric relationships be preserved.

3. The consistency filter gives a parametric control over the spectrum of the output. Thus, if we have strong reason to believe that the underlying signal low dimensional, despite the findings of (3.8) in the presence of noise, λ may be adjusted optimally to give the best guess of the low-dimensional signal.

4. The consistency filter is a linear combination of the right singular vectors of the graph laplacian, which encode the global geometry of the image. The vectors corresponding to low singular values encode the strongest features and as the singular values increase, the corresponding singular vectors encode more minute details. Thus, with a single parameter λ we can control the weights of the combination giving preference to the vectors corresponding to smaller singular values.

Chapter 5

Implementation

5.1 Constructing the Graph

In order to solve (4.17), we first need to construct the graph i.e. the appropriate matrices in (4.17). We observe that the dimensionality of our system of equations is huge, since for an M×N image, the matrix C will be of the order: MN×MN which for a 256×256 image will be 65000×65000. Storing so many elements and processing them becomes infeasible at these numbers. Hence, we do not populate every element of the graph K, rather we compute only the first **n** nearest neighbors and assign weights to them, the rest are approximated as zero. This makes the graph very sparse and since only $n.MN$ are non-zero in $(MN)^2$ elements. We store and process it as a sparse matrix in MATLAB.

For finding the nearest neighbors, we do a brute force search around each pixel using a predetermined search window. This is done as follows: we first construct a dictionary of patches from the image, i.e. for every pixel: access the patch surrounding it and store it in a dictionary. Now, once the dictionary is ready, for every pixel in the search window, compare the patch SSD with the central pixel and sort them to get the best n similar patches and populate each row of the graph K. Once the graph matrix is constructed, normalizing it, constructing the laplacian and matrix C in (4.17) is trivial. The system of equations in (4.17) is solved using MATLAB's inbuilt conjugate gradient solver which is reasonably fast.

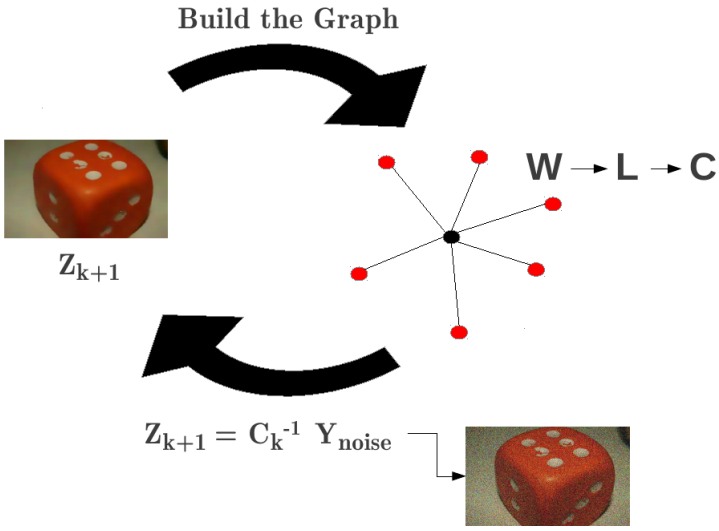


Figure 5.1: Estimating the true graph Iteratively

5.2 An Iterative Scheme

As mentioned in section 4.1, we ideally require the true geometry of the underlying signal in order to enforce consistency. We observe that the graph built out of the noisy image is quite sensitive to the level of noise. Thus, on increasing levels of noise, the graph constructed from the noisy image is quite different from that of the true signal. This can be explained from the fact that when the level of noise becomes at par with the image features then distinguishing image and noise becomes impossible. Thus and edge with drop of 30 intensity units becomes undetectable under a noise variance of $\sigma^2 = 30$. Thus, SSD computations for such regions will yield erroneous matches and hence a corrupted graph.

One way to find a suitable graph is to compute it on a pre-filtered pilot and iterate on that. Thus as the image gets de-noised on each iteration so does the graph. However, care must be taken not to compute the graph on an over-smoothed image which fail to encode the minute details of the image.

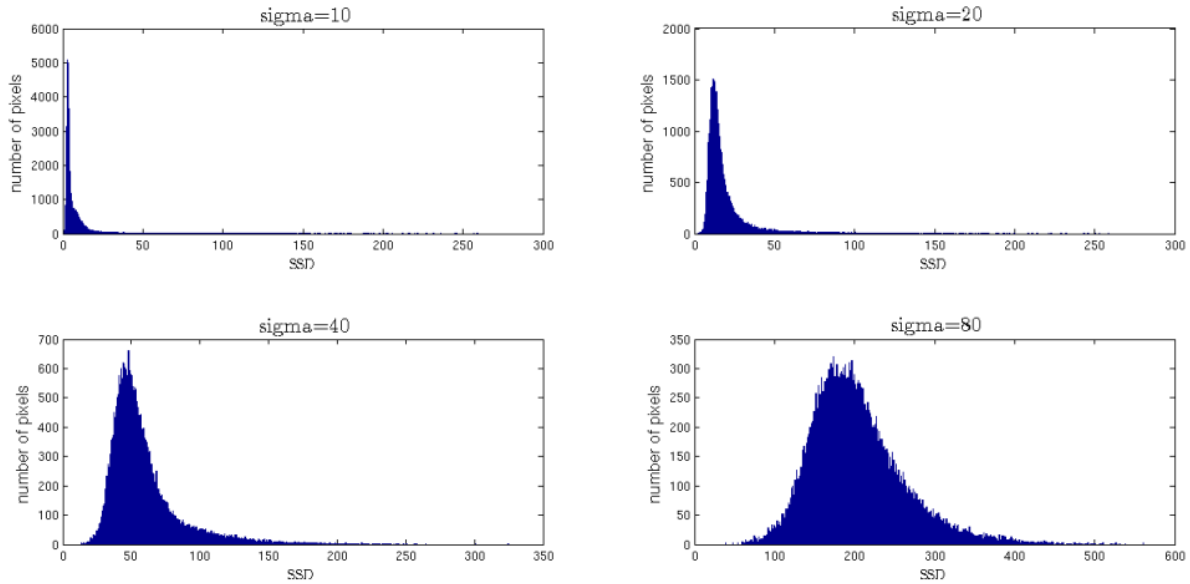


Figure 5.2: Histograms of the patch-SSD of the 1^{st} nearest neighbor for various noise levels of the cameraman image

5.3 Adaptive Patch Strategy

The purpose of regularization is to relax the consistency constraint on those pixels which do not belong to strong clusters, and in our experiments we observed that the consistency filter was doing poorly in such regions of rapid image fluctuations and texture.

Therefore, we adopted a new strategy to handle such regions: adaptively change the patch-size for different regions of the image. This requires a preprocessing step of identifying these regions in the first place.

We tried different texture detection schemes to classify the image into different regions like LARK [9] and the Harris function [16]. However we developed a simpler scheme and found it to best suit our purpose. Since we are concerned on identifying the weak cluster regions of the image based on patch neighborhoods, we perform a simple threshold step on the histogram of errors of the first nearest neighbor of every pixel. Figure (5.2) shows the histograms of the sum of squared difference (SSD) of the 1^{st} nearest neighbor for different noise levels. We can clearly see that on increasing levels of noise, the distribution looks close to a chi-squared distribution (square of a Gaussian random variable). This is to be expected since, most of the image comprises of smooth regions and strong textures,

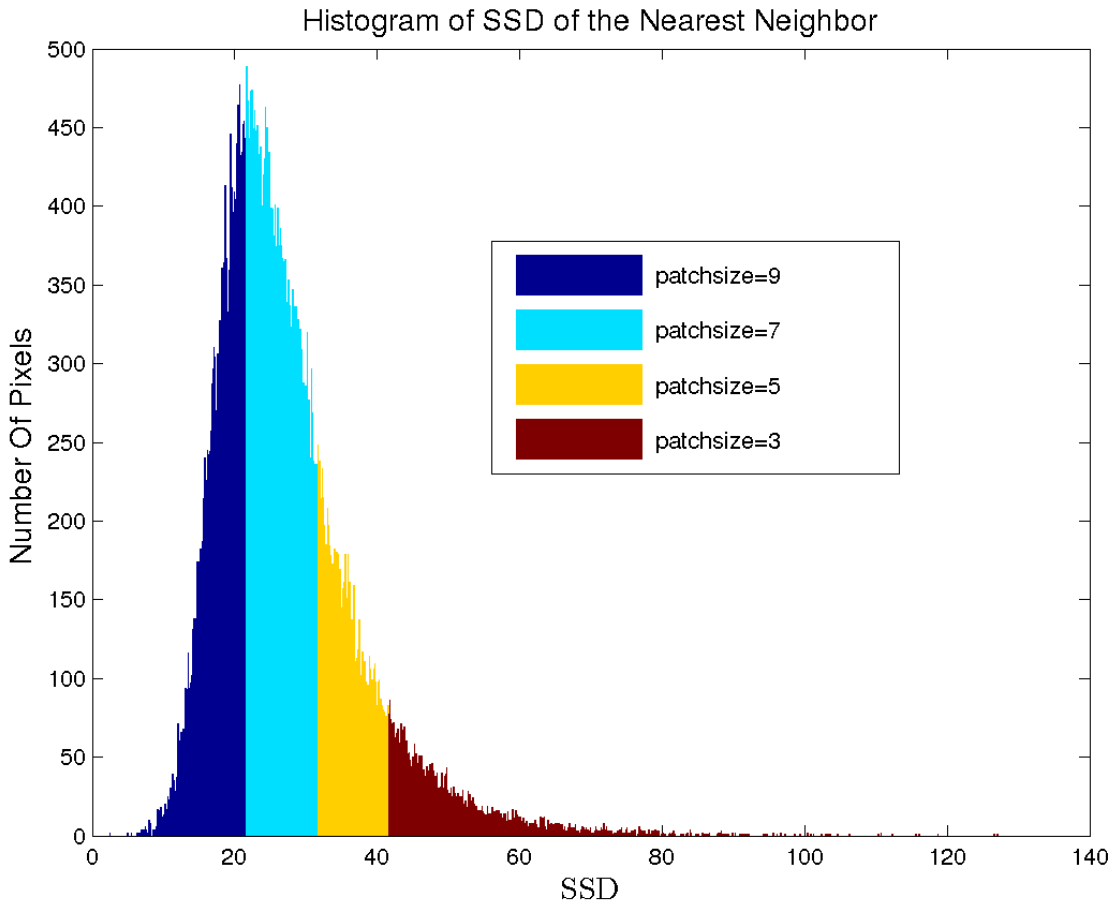


Figure 5.3: Segmenting the patch-SSD histogram and assigning corresponding patch-sizes

which means that most of the detected similar patches are indeed noisy versions of an underlying base patch. We divide the histogram into 4 regions and for each region associate a corresponding patch-size and penalty weight.

Let Mo and σ be the mode and variance of the histogram respectively. Let $\beta > 0$ be a tuning parameter, and $SSD(i)$ denotes the patch-SSD of the 1^{st} nearest neighbor at pixel i.

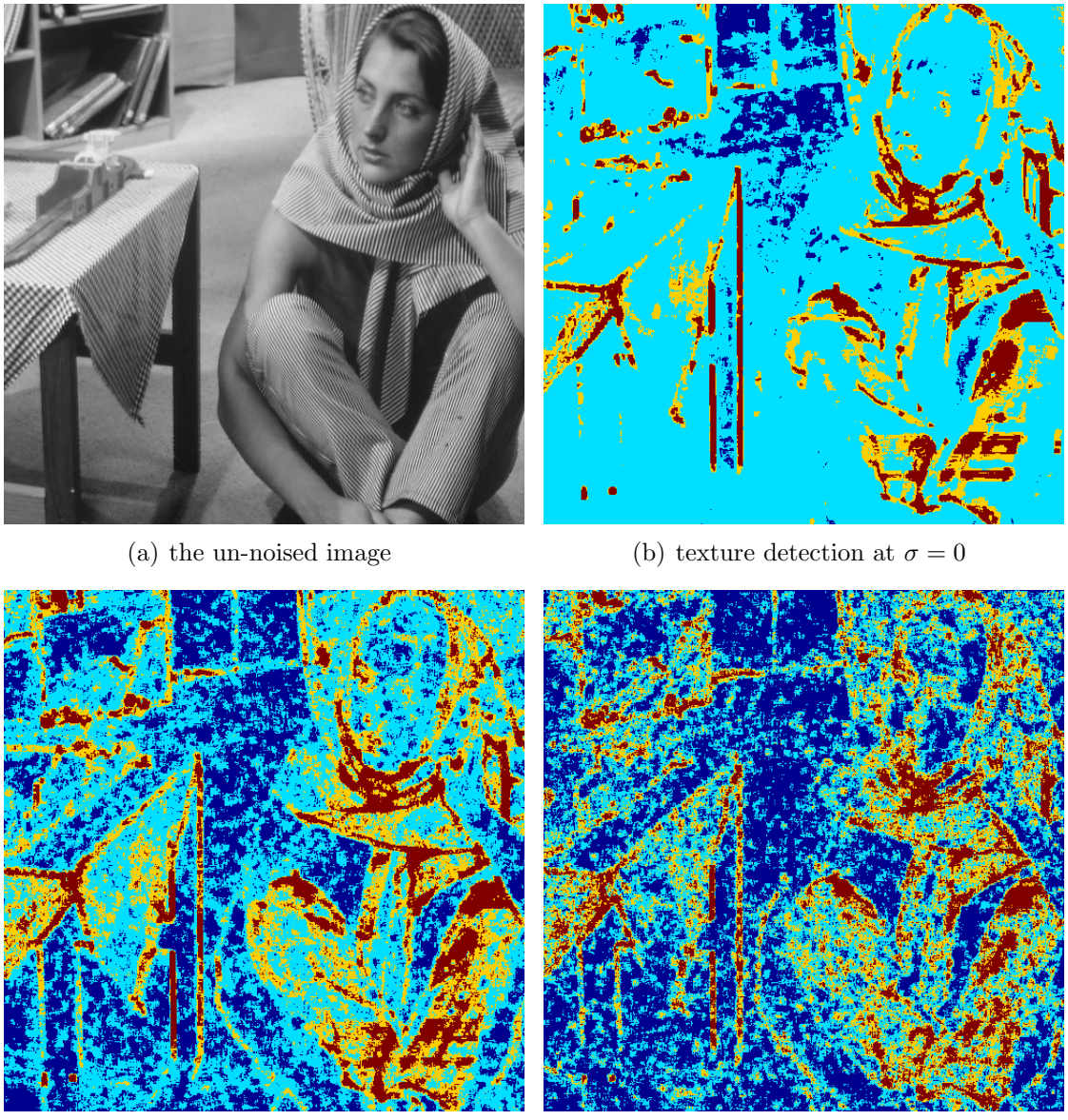
$SSD(i) < Mo$	Region-1, choose patch-size=9.
$Mo \leq SSD(i) < (Mo + \beta\sigma)$	Region-2, choose patch-size=7.
$(Mo + \beta\sigma) \leq SSD(i) < (Mo + 2\beta\sigma)$	Region-3, choose patch-size=5.
$SSD(i) > (Mo + 2\beta\sigma)$	Region-4, choose patch-size=3.

With this identification, we now construct our graph adaptively, i.e. depending on which part of the histogram the pixels belongs, we use a corresponding patch-size and populate each row of the graph. Figure's (5.3) and (5.4) shows the regions which show bad clustering for various levels of noise.

Yet another way of adopting adaptively to different regions is to modify the penalties in (4.14). Currently the deviation from the consistency is penalized in the L_2 sense, where λ is the penalty weight, the same for all pixels. We can adopt a differential penalty scheme by having different λ's for different regions based on the detection scheme explained. Conceptually our cost function (4.14) will get modified as:

$$\widehat{Z} = \underset{Z}{argmin} \ \ ||Z - Y||^2 + \ ||\Lambda L Z||^2 \tag{5.1}$$

where the matrix Λ is a diagonal matrix populated with different weights depending on the findings of Figure (5.2). All properties of the consistency filter hold except that now we have a multi-parameter control over the spectrum of our output filter.



(a) the un-noised image

(b) texture detection at $\sigma = 0$

(c) texture detection at $\sigma = 15$

(d) texture detection at $\sigma = 30$

Figure 5.4: Detecting different regions of the image for various levels of noise. For color-code refer Figure: 5.3

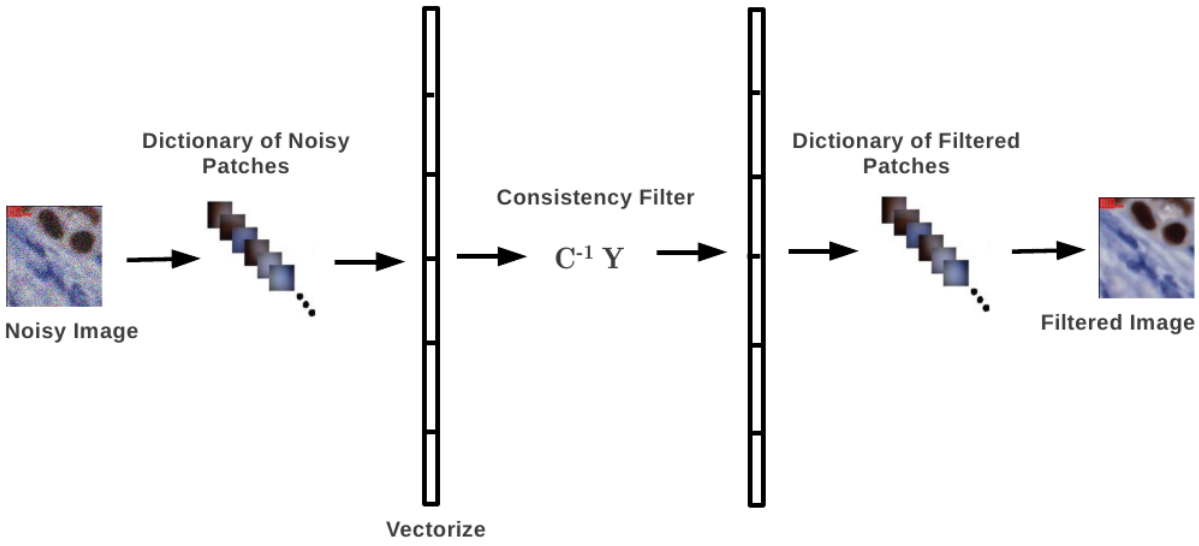


Figure 5.5: Implementing the Consistency Filter on a Patch Scale

5.4 Patch-Based Implementation

So far our formulation was dealt on a pixel basis i.e. pixels are considered to be the nodes of the graph and a similarity measure is defined for them. However, the same formulation may be extended to patches i.e. we can consider patches as nodes and the same similarity measure is now defined between *patches as a whole*. Thus, the image is first broken down into a dictionary of patches , where the number of patches (length of the dictionary) equals the number of pixels. This dictionary is now in converted into a single vector and Equation (3.8) and all subsequent analysis holds for this vector.

Thus, instead of de-noising pixels like in (3.8) and (4.17) we now de-noise patches and then aggregate these patches to form the final estimate. The process of de-noising patches is identical to the pixel process, the difference lying in the dimensions of our laplacian and associated matrices. Thus, instead of $MN \times MN$ the dimensions of the matrix for patch based de-noising becomes $MN(patchsize)^2 \times MN(patchsize)^2$.

Conceptually, the main difference lies in what our base units are and the relationships between them. When we associate similarities to patches, we mean that every pixel in one patch is similar to the corresponding pixel in the other. When we do de-noising using these relationships, we will have a huge set of de-noised patches which we would have to

aggregate to form the de-noised image. This means that in the process of aggregation, every pixel will have multiple estimates from different patches and we would have combine them in some fashion to to get the final estimate.

For our case, we perform the aggregation using a weighted mean, i.e. the net estimate at each pixel is a weighted combination of its estimates from various patches where, the weights are decided based on distance between the concerned pixel and the center of the patch. Thus, the patch having its center as the concerned pixel gets the maximum weight and the weight dies down as the distance increases.

In our experiments, we observe that, in general patch based de-noising does marginally better than its pixel based counterparts. This can be intuitively explained by noticing that for de-noising using patches, we have many more number of constraints (and hence estimates) per-pixel which makes the averaging more rigorous and supplementary to a better de-noising performance.

Chapter 6

Results

In the next few pages we have compiled the results of the consistency filter and compared it with other algorithms. A major factor in the behavior of the consistency filter is the penalty factor λ. As we have mentioned in section 4.4, λ signifies the our belief of the magnitude of *clusterness* of the graph. For various noisy images we plotted iso-noise curves i.e. PSNR vs λ for the consistency filter Figure (6.1) shows one such plot. We can see that depending on the noise levels, the curve is convex in nature with a particular λ maximizing the PSNR, after which the PSNR saturates to a fixed level. This is intuitive since, after a point the weights in (4.17) become uniform and hence changing lambda does not change the filter as much.

Thus, the iso-noise curves suggest an optimal λ^* for which the imposition of consistency is at its peak. In our experiments we have chosen λ^* based on thumb rule, a better way would be to choose it based on minimization of some estimated risk or error function like Steins Unbiased Risk Estimator (SURE). However we see that the drop in PSNR from the peak to the saturation is not very large and hence any λ on the greater side of the peak should do fine.

A look at the PSNR's in tables 6.1-6.4 tells us how consistency vastly improves from the baseline filtering procedure of (3.8). We can clearly see that consistency filter of the oracle graph i.e. the graph built from the original un-noised image shows a considerable gain in PSNR performance. It also signifies that that high noise levels considerably

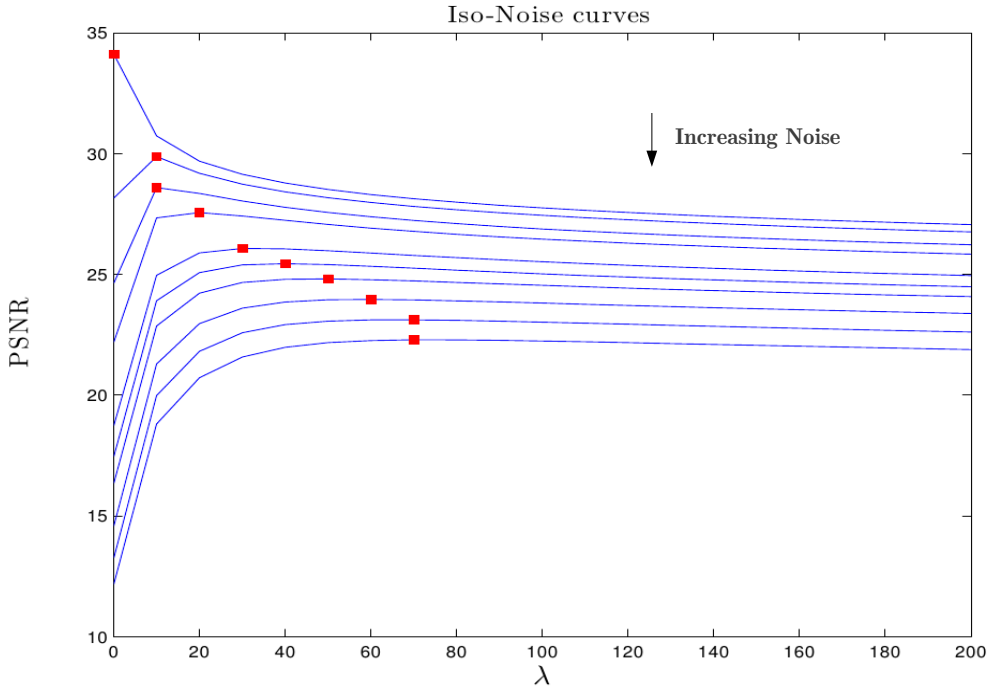


Figure 6.1: PSNR vs λ for various noise levels for couple image. Red dots show the mode/maximum of each curve

distort the geometric relationships of the graph.

A comment on the behavior of the plots of 6.7 and 6.8 : Clearly the performance of KSVD, the Consistency and the Oracle Consistency filters are somewhat linearly dependent on the input- noise PSNR. However, the non-local means and the bilateral, have saturation characteristics for high PSNR's since at low noise levels, these filters (predominantly the bilateral filter) become close to the standard Gaussian smoothing filter, i.e. the estimate at each pixel is an average of its neighbors.

For the non-local means filter, the number of nearest neighbor chosen was 5, for the consistency filter it was chosen adaptively as shown in section 5.3. For the bilateral filter, the window size was 7 and the σ's of the space and intensity terms were 50 each.

Table 6.1: PSNR of different Algorithms for Boat image (GrayScale)

σ	**Bilateral**	**Non-Local Means**	**Consistency**	**Oracle**	**KSVD**
10	25.55	26.62	29.96	32.19	33.39
20	25.50	25.93	27.95	31.13	29.42
30	25.27	25.05	26.41	30.17	27.29
40	24.75	24.01	25.28	29.60	25.87
60	20.55	21.73	23.52	28.72	24.00

Table 6.2: PSNR of different Algorithms for Barbara image (GrayScale)

σ	**Bilateral**	**Non-Local Means**	**Consistency**	**Oracle**	**KSVD**
10	26.44	29.58	31.68	34.21	32.6
20	26.26	28.38	29.09	32.74	29.47
30	26.06	26.84	27.27	31.59	27.58
40	25.26	25.29	26.00	30.77	26.26
60	20.75	22.39	24.17	29.43	24.25

Table 6.3: PSNR of different Algorithms for House image (Color)

σ	**Bilateral**	**Non-Local Means**	**Consistency**	**Oracle**	**BM3D**
10	30.64	28.73	31.51	32.33	36.18
20	30.10	28.31	30.17	31.53	33.84
30	29.02	27.57	29.03	30.78	32.29
40	26.62	26.96	28.18	30.14	30.55
60	18.51	24.99	26.54	29.20	29.7

Table 6.4: PSNR of different Algorithms for Lena image (Color)

σ	**Bilateral**	**Non-Local Means**	**Consistency**	**Oracle**	**BM3D**
10	29.6160	28.8533	31.8241	32.9962	36.31
20	29.2250	28.3524	30.2378	31.8189	33.10
30	28.3809	27.6963	28.9118	30.8800	31.00
40	26.2287	26.9326	27.8686	30.0994	29.18
60	18.5820	25.4581	26.3199	28.9192	27.9

(a) noisy image: $\sigma = 20$

(b) Bilateral Filter. PSNR: 26.02 dB

(c) Non-Local Means Filter. PSNR: 28.3 dB

(d) Consistency Filter. PSNR: 29.03 dB

Figure 6.2: Results for the Barbara image

(a) noisy image: $\sigma = 20$

(b) Bilateral Filter

(c) Non-Local Means Filter

(d) Consistency Filter

Figure 6.3: Results for the Barbara image (Zoomed)

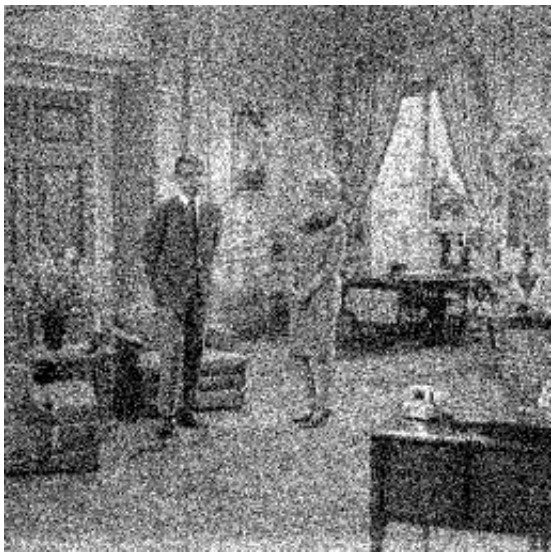
(a) noisy image: $\sigma = 20$

(b) Bilateral Filter. PSNR: 26.8 dB

(c) Non-Local Means Filter. PSNR: 26.4 dB

(d) Consistency Filter. PSNR: 28.1 dB

Figure 6.4: Results for the Boat image



(a) noisy image: $\sigma = 40$

(b) Bilateral Filter. PSNR: 24.56 dB

(c) Non-Local Means Filter. PSNR: 23.81 dB

(d) Consistency Filter. PSNR: 24.02 dB

Figure 6.5: Results for the Couple image

(a) noisy image: $\sigma = 50$

(b) Bilateral Filter. PSNR: 22.69 dB

(c) Non-Local Means Filter. PSNR: 24.92 dB

(d) Consistency Filter. PSNR: 27.99 dB

Figure 6.6: Results for the Cap image

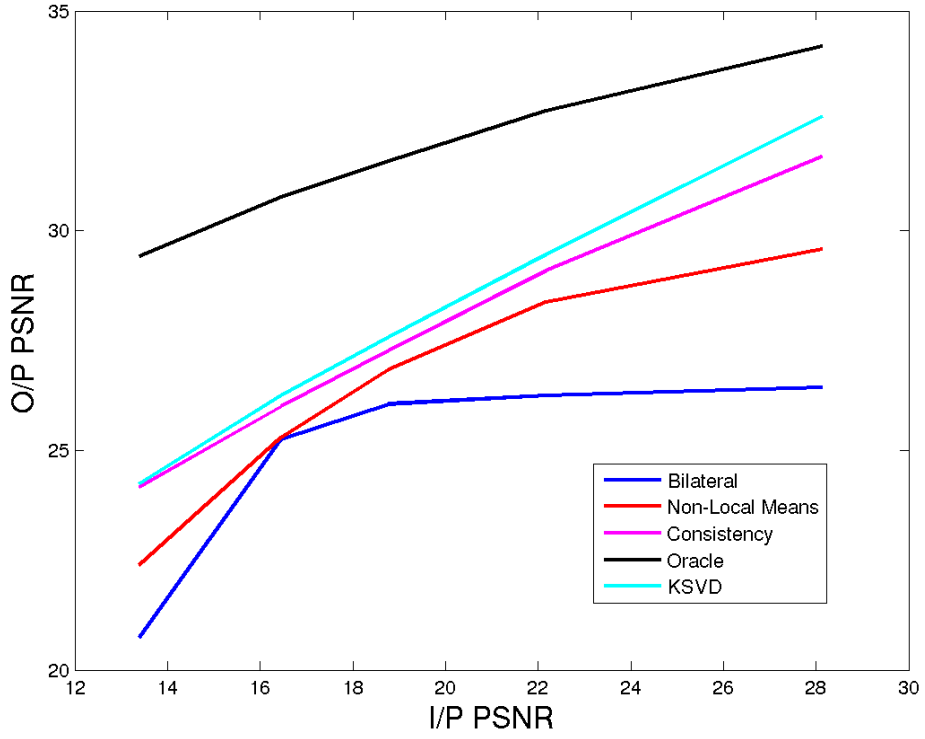
(a) noisy image: $\sigma = 50$

(b) Bilateral Filter. PSNR: 22.08 dB

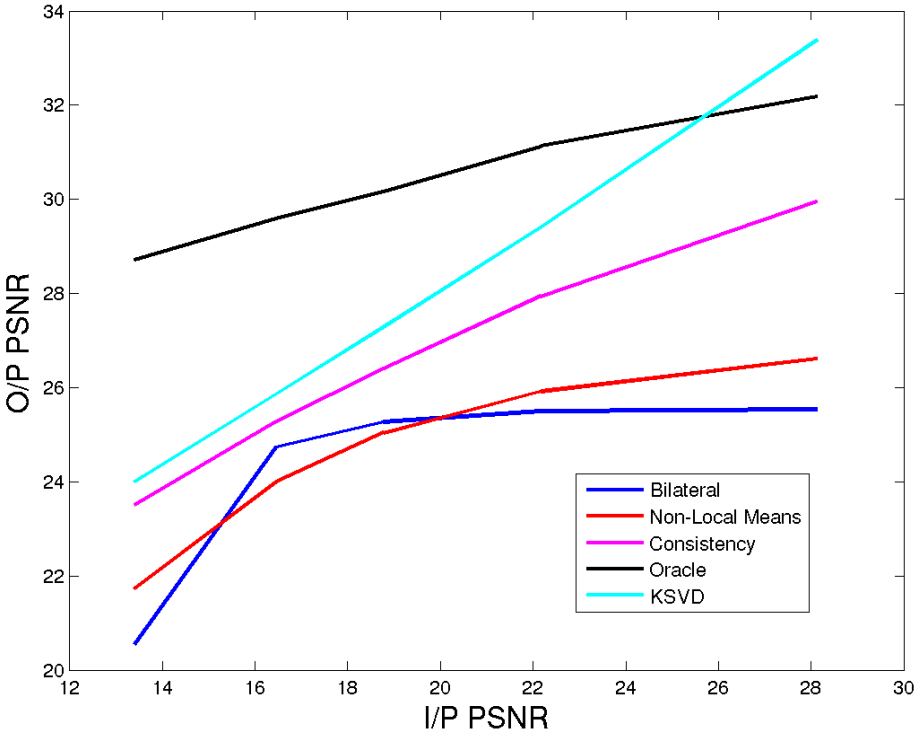
(c) Non-Local Means Filter. PSNR: 24.44 dB

(d) Consistency Filter. PSNR: 27.07 dB

Figure 6.7: Results for the House image

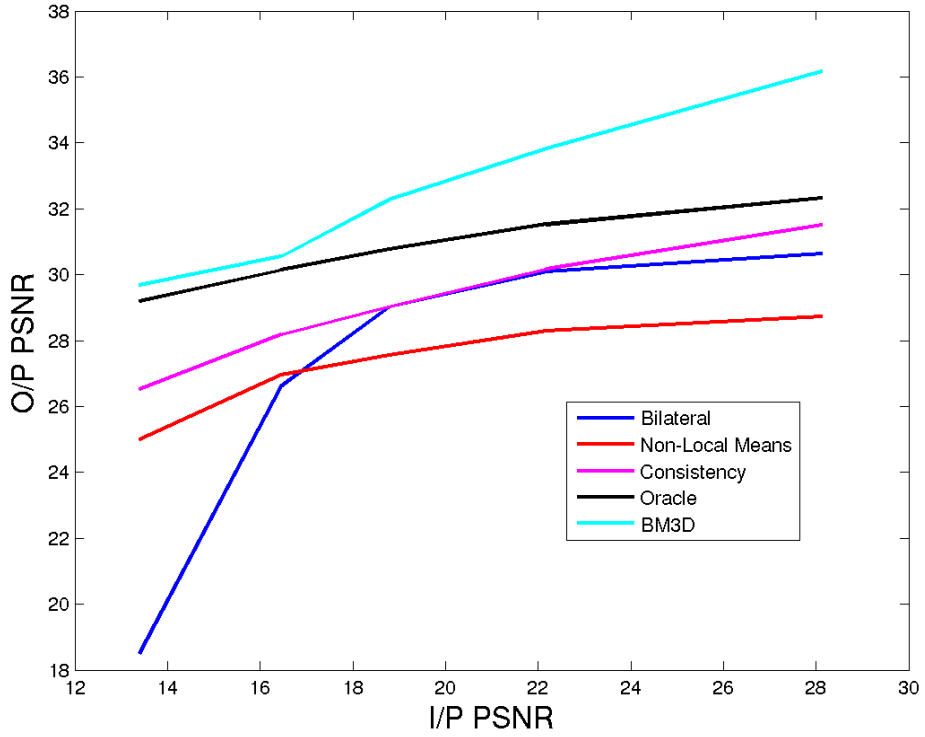


(a) Barbara

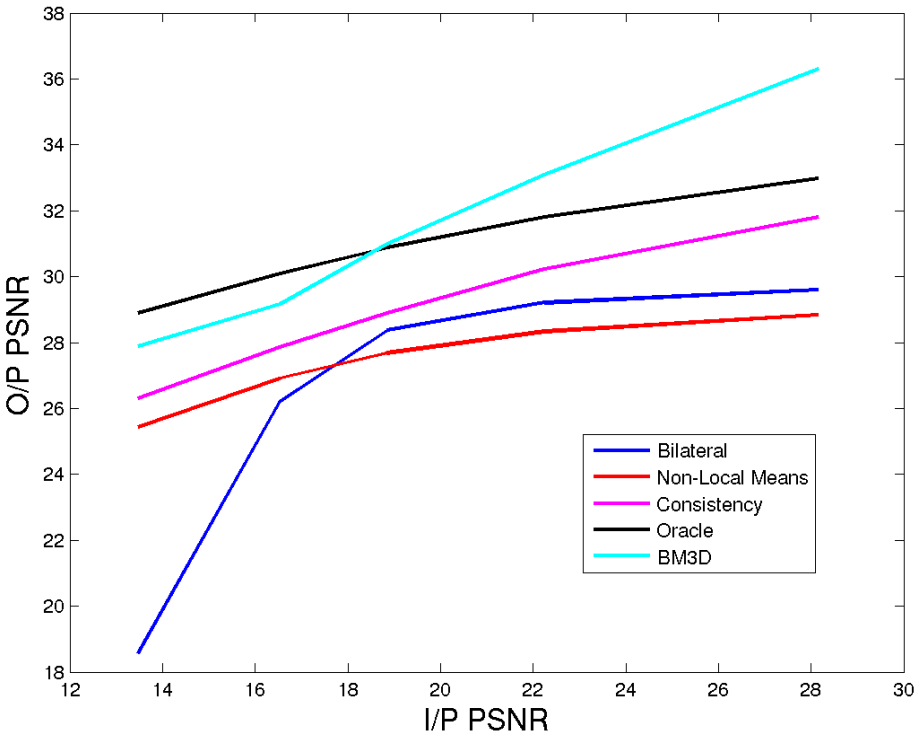


(b) Boat

Figure 6.8: PSNR Comparison for various images



(a) House (Color)



(b) Lena (Color)

Figure 6.9: PSNR Comparison for various images

Chapter 7

Conclusions and Future Work

Enforcing consistency in the multidimensional filtering framework exploits the strong connectivity existing among the the pixels in a natural image. It works best when the underlying signal has strong self-similarity in terms of the *clusterness* of its patches. In this project we have explored the effect of enforcing consistency it in an L_2 sense, i.e. we have penalized deviation from ideal in a squared manner. However, preliminary experiments show that the distribution of the error $\mathbf{e}$=LZ for natural images show heavy tails and peakedness, suggesting a Laplacian distribution as mentioned in section 2.3. therefore, one possible future experiment would be to extend the it to a L_1 framework where it might do much better.

Despite, maximally exploiting global geometric relationships, we see that a regularization framework for consistency reduces it to a linear filter of the form (3.8). Therefore, the performance of this filter will be bounded and may not do as well as the highly non-linear algorithms like BM3D [3] and K-SVD[4]. Thus one could come up with a upper bound on the performance of filters of the form of (3.8).

Finally, fundamental to the global consistency argument is the fact that the true graph is available and we see its importance in the results of tables 6.1-6.3. As mentioned in section 5.2, we estimate it iteratively. However, more analysis on the perturbation of the graph and in particular, the perturbation of matrices W, L, C with noise and its effect on the algorithm is of need.

References

[1] J. Huang and D. Mumford, "Statistics of natural images and models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, vol. 1. IEEE, 1999.

[2] E. P. Simoncelli, "Statistical modeling of photographic images," *Handbook of Video and Image Processing*, pp. 431–441, 2005.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[5] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*. IEEE, 2011, pp. 977–984.

[6] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 2. IEEE, 2005, pp. 60–65.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision, 1998*. IEEE, 1998, pp. 839–846.

[8] P. Milanfar, "A tour of modern image filtering: new insights and methods, both practical and theoretical," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 106–128, 2013.

[9] 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, 2007.

[10] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[11] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Gatesmark Publishing Tennessee, 2009, vol. 2.

[12] A. Berman and R. J. Plemmons, *Nonnegative matrices*. SIAM, 1979.

[13] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHUP, 2012, vol. 3.

[14] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[15] P. Milanfar, "Symmetrizing smoothing filters," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 263–284, 2013.

[16] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[17] P. Chatterjee and P. Milanfar, "Patch-based locally optimal denoising," in *18th IEEE International Conference on Image Processing (ICIP), 2011*. IEEE, 2011, pp. 2553–2556.

[18] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.