Model-Based Adaptive Non-Local Means Image Denoising

Hila Berkovich

Model-Based Adaptive Non-Local Means Image Denoising

Research Thesis

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Hila Berkovich

Submitted to the Senate of the Technion - Israel Institute of Technology

Iyar 5774 Haifa May 2014

Acknowledgment

The research thesis was done under the supervision of Prof. David Malah and Dr. Meir Barzohar in the Electrical Engineering Department.

I would like to express my deepest gratitude to my supervisor, Prof. David Malah, who has been a true mentor. I am privileged to have had his devoted guidance and to have benefited from his wealth of knowledge and tolerance. Moreover, I would also like to thank Dr. Meir Barzohar for his guidance through the entire research process. Many thanks to the devoted Signal and Image Processing Lab (SIPL) staff: Nimrod Peleg, Yair Moshe, Ziva Avni and Avi Rozen who jointly create the environment, which enabled and encouraged this research.

On a more personal note, I would like to thank my fiancee Tomer and my family for their constant mental support and encouragement.

The Generous Financial Help of The Technion is gratefully Acknowledged.

Part of this work was published and presented in the ISPA 2013 conference and the IMVC 2014: “Non-Local Means Denoising Using a Content-Based Search Region and Dissimilarity Kernel”.

Contents

List of Figures

List of Tables

Abstract

The need for efficient image restoration methods has grown with the massive production of digital images and movies of all kinds, often taken in poor conditions. Image denoising is used to find the best estimate of the original image given its noisy version. Among the vast image denoising methods that were suggested, patch-based approaches have drawn much attention in the image processing community. The Non-Local Means (NLM) denoising algorithm, first introduced by Buades et al. in 2005, takes advantage of image redundancy by comparing pixel neighborhoods within an extended search region in the image. Each pixel value is estimated as a weighted average of all other pixels in this search region. These pixels are each assigned a weight that is a function of the similarity between the local neighborhood of the Pixel Of Interest (POI) and their local neighborhood, such pixels with a similar neighborhood are assigned higher weights. The NLM denoising approach originally refers to Additive White Gaussian Noise (AWGN).

The participation of dissimilar pixels, which may be included in the extended search region, in the weighted averaging process, degrades the denoising performance. To eliminate their effect, researchers suggest creating an adaptive search region that excludes those pixels (e.g. [4,37]). These suggested methods are parameter dependent and involve heuristics. Another approch [17] suggests using a rectangular search region of various sizes, such that the window size is adapted to local structure.

In this thesis, we present a novel model-based method that extracts a set of similar pixels

for a given POI from its initial search region, using the statistical distribution of the NLM dissimilarity measure. Our approach does not require any parameter setting and provides better results than other compared adaptive search region approaches. Our proposed scheme was also compared to the standard NLM and was found to provide better performance both quantitatively and visually. We have also explored the effect of correlation between the dissimilarity elements of a given search region. Three sources of correlations were explored: correlation due to the comparison to the same reference patch, correlation due to patches overlap, and correlation due to overlap with the reference patch. We found that the correlations affect the dissimilarity model, however their effect on the denoising performance is insignificant. The model-based scheme was also integrated in the Block-Matching 3D (BM3D) state-of-the art denoising scheme, such that the computational complexity of the original BM3D is reduced while denoising results remain comparable. Besides the AWGN, we have explored our approach on Poisson noisy images as well. Poisson noise, which is signal dependent, is the noise type that characterizes images taken by a digital camera. In a similar manner to the Gaussian noise case, we have compared our proposed model-based approach to the standard NLM as well as the standard BM3D to the model-based BM3D. For the two methods, both NLM and BM3D, the tendency of the AWGN denoising was preserved, such that the proposed NLM provided better results both quantitatively and visually than the standard NLM, and the model-based BM3D provided comparable results to that of the standard BM3D.

List of Symbols and Abbreviations

Symbols

A_i	Similarity patch centered at pixel i
B^U, B^B	Estimated mean of normalized dissimilarities computed using the Uniform and the Box patch-kernel respectively
B^{Hard}	Maximal number of patches per group in BM3D Phase1
C^B_{min}, C^B_{max}	Clusters of matrix R^B
C^U_{min}, C^U_{max}	Clusters of matrix R^U
$C_{\widetilde{\mathbf{d}}^B_i}, C_{\widetilde{\mathbf{d}}^U_i}$	Covariance matrix of normalized dissimilarities computed using the Box and the Uniform patch-kernel respectively
D_{JS}	Jensen-Shannon divergence
D_{KL}	Kullback-Leibler divergence
F	Matrix form of Box patch-kernel
M	Linear size of search region in NLM
M^{Hard}	Linear search region patch size in BM3D Phase 1
N_i	Noise value at pixel i
O	Matrix that represents correlation between dissimilarities due to patches overlap
R	Matrix of normalized cardinality of adaptive search region
R^U, R^B	Matrix of normalized cardinality of adaptive search region, based on the Uniform and Box patch-kernels respectively
S_i	Set of global indices associated with the initial search region of size $M \times M$
S^D_i	Sub-set of the search region S_i that includes pixels with dissimilar neighborhood to that of pixel i

S_i^S	Sub-set of the search region S_i that includes pixels with similar neighborhood to that of pixel i
S_i^{SU}, S_i^{SB}	Sub-set of the search region S_i that includes pixels with similar neighborhood to that of pixel i, for dissimilarity computed using the Uniform or the Box patch-kernel respectively
T_{Direct}	Direct inverse Anscombe transform
$T_{Asym-Unbiased}$	Asymptotically unbiased inverse Anscombe transform
$T_{Exact-Unbiased-Close}$	Closed-form exact unbiased inverse Anscombe transform
TH_d	Dissimilarity threshold based on ALGORITHM I
TH^G	Estimated Variance threshold
$\hat{V}^U, \hat{V}^B$	Estimated variance of normalized dissimilarities computed using the Uniform and the Box patch-kernel respectively
W_i	Sum of weights of pixels associated with S_i (for weight normalization)
X_i	Noise-free pixel i
Y_i	Noisy pixel i
$d_i(j)$	Dissimilarity of pixel j with respect to pixel i, $j \in S_i$
$d_i^U(j), d_i^B(j)$	Dissimilarity, computed using the Uniform or Box patch-kernel, of pixel j with respect to pixel i , $j \in S_i$
$\widetilde{d}_i^U(j), \widetilde{d}_i^B(j)$	Normalized dissimilarity, computed using the Uniform or Box patch-kernel, of pixel j with respect to pixel i , $j \in S_i$
$\widetilde{\mathbf{d}}_i\left(\Psi_i^L\right)$	Vector of normalized dissimilarities arranged in ascending order
$\widetilde{d}_i\left(\Psi_i^L(m)\right)$	The normalized dissimilarity of the global index that corresponds to the m^{th} element of the set Ψ_i^L
f	Multiplication factor of the variance of the estimated variance
$f(y)$	Anscombe transform
h	NLM smoothing parameter
i	Lexicographic pixel location
p	Linear size of similarity patch in NLM

p^{Hard}	Linear similarity patch size in BM3D Phase 1
r_i	Elements of matrix R
r_i^U, r_i^B	Elements of matrices R^U, R^B respectively
$w_{i,j}$	Weight of pixel j with respect to pixel i
α_s	Normalized patch-kernel coefficients, $s \in [1, p^2]$
κ	Sum of squared patch-kernel coefficients
$\lambda(j)$	Non-Centrality parameter (non central Chi-Square distribution) of pixel j in S_i^D
σ_n^2	Additive noise variance
Ψ_i^L	Sub-set of the search region that includes the global indices of the pixels that are included in the search region and satisfy the no-overlap constraint, sorted in order of increasing dissimilarity values. $L \in [2, \lvert S_i \rvert]$ refers to the cardinality of the set
$\lVert \cdot \rVert_{2,a}$	Norm 2 operator weighted by a Gaussian of zero mean and a variance a

Abbreviations

AWGN	Additive White Gaussian Noise
BM3D	Block-Matching 3D
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semi-conductor
EM	Expectation Maximization
IID	Independent and Identically Distributed
JS Divergence	Jensen Shannon Divergence
KL Divergence	Kullback Leibler Divergence
K-SVD	K Singular Value Decomposition
LPA-ICI	Local Polynomial Approximation-Intersection of Confidence Intervals
MMSE	Minimum Mean Squared Error
NLM	Non-Local Means
PDF	Probability Distribution Function
POI	Pixel of Interest
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity
RGB	Red Green Blue
STD	STandard Deviation
SURE	Stein Unbiased Risk Estimator
UMVU Estimator	Uniformly Minimum Variance Unbiased Estimator
VST	Variance Stabilizing Transform

1. Introduction

1.1. Image Denoising

The need for efficient image restoration methods has grown with the massive production of digital images and movies of all kinds, often taken in poor conditions. No matter how good cameras are, an image improvement is always required. A digital image is generally encoded as a matrix of gray-level or color values. Each one of the pixel values Y_i is the result of a light intensity measurement, usually made by a charge coupled device (CCD) matrix coupled with a light focusing system or a complementary metal oxide semiconductor (CMOS). Each captor of the sensor is roughly a chamber in which the number of incoming photons is being counted for a fixed period corresponding to the obturation time. The two main limitations in image accuracy are categorized as blur and noise. Blur is intrinsic to image acquisition systems, as digital images have a finite number of samples and must satisfy the Shannon–Nyquist sampling conditions. Moreover, each pixel value is a result of photon count over the respective sensor chamber, which also depends on chamber's area. This integration adds to the blurring effect (known as aperture problem). The second main image perturbation is noise. In a first rough approximation, one can write $Y_i = X_i + N_i$, where Y_i is the observed value, X_i would be the "true" value at pixel i, namely the one which would be observed by averaging the photon counting on a long period of time, and N_i is the noise perturbation. The amount of noise added due to image acquisition is signal-dependent and characterized by a Poisson distribution, that is, N_i is larger when X_i is larger. However, in

many works, mostly with simulated noise, additive white Gaussian noise (AWGN) is used, so that the noise values at different pixels are assumed signal-independent and as realizations of an *i.i.d* (independent and identically distributed) random variable.

Image denoising is used to find the best estimate of the original image given its noisy version. Many methods for image denoising have been suggested, and a comprehensive review of them can be found in [5]. Among the proposed denoising schemes, patch-based methods have drawn much attention in the image processing community (e.g. [9]). Moreover, most of the suggested schemes deal with Gaussian noise model. An example of a recent patch-based denoising approach is the one introduced by Ram et al. [33]. Their scheme suggests to reorder the pixels in a given image based on their corresponding patches similarity and then apply a smoothing operator on the ordered pixels.

In 2005, Buades et al. [5] introduced the Non-Local Means (NLM) denoising algorithm which takes advantage of image redundancy by comparing pixel neighborhoods within an extended search region. Each pixel value is estimated as a weighted average of all the other pixels in this search region. These pixels are each assigned a weight that is proportional to the similarity between the local neighborhood of the reference pixel and their local neighborhood, such that pixels whose neighborhood is the most similar to the neighborhood of the reference pixel are given the largest weights. Moreover, the weights are controlled by a weight smoothing parameter (h), which steers their decay. It is increased with the noise variance in the image and it is usually set constant for the entire image. Since image pixels are highly correlated while Gaussian additive noise is typically $i.i.d.$, weighted averaging of these pixels results in noise reduction. Consequently, the uniqueness of the NLM approach lies in its ability to exploit spatial correlation in a defined neighborhood (search region) for noise removal.

Gaussian image denoising techniques can be exploited to deal with Poisson noise as well. In the Poisson case, a Variance Stabilizing Transform (VST), e.g., the Anscombe [26] transform is used in order to convert the signal-dependent noise to a Gaussian additive white

noise with unit variance. Then, NLM can be applied on the transformed noisy image. Finally, an inverse transform is applied on the denoised image.

1.2. Thesis Objectives

The NLM search region is usually a rectangular neighborhood, centered at the pixel of interest (POI), which may include pixels whose original gray value do not match the value of the original central pixel. Consequently, their participation in the weighted averaging process degrades denoising performance, even though they are assigned relatively small weights. To eliminate their effect, researchers, e.g., [4,37], suggest creating an adaptive search-region, which excludes those dissimilar pixels.

In this thesis, we present a novel model-based method, which defines a set of similar pixels to the POI from the initial search region, using the statistical distribution of the dissimilarity measure. Moreover, to enhance the denoising, the proposed method also adaptively assigns one of two patch-kernel types to each pixel, based on its local features. This patch-kernel is used for weight computation. We show that the suggested NLM modification improves the standard NLM performance both quantitatively and qualitatively. This approach is parameter free, since it is model-based, and that is its main uniqueness compared to other suggested methods for an adaptive search region.

We also refer to the correlation between the dissimilarities of the explored pixels within a defined search region. The correlation is due to overlap between compared neighborhoods and their comparison to the same reference patch, by definition.

Moreover, we apply our adaptive search region method to the BM3D [9,20] state of the art denoising approach and suggest a scheme that facilitates the original BM3D flow by using a model-based grouping for the first phase of the algorithm instead of the conventional grouping method. This proposed modification saves computations while providing comparable

denoising results.

1.3. Thesis Outline

The thesis is organized as follows: in Chapter 2, we overview the Non-Local Means (NLM) [5] neighborhood filter that takes advantage of the redundancy and self similarity of the image. The filter defines the denoised value of a given pixel by applying a weighted average using the pixels included in a defined search region, centered at the pixel of interest (POI). In addition, we review the NLM set of parameters and discuss how they can be optimized to provide the best denoising results.

Chapter 3 presents our novel model-based method for determining a pixel-wise adaptive search region using the statistical distribution of the dissimilarity measure. We discuss the statistical analysis behind the proposed approach and compare its performance to the standard NLM and to the LPA-ICI approach [35] that also suggests to use an adaptive search region. Appendix A provides additional comparison results between the standard NLM and the proposed model-based approach.

In Chapter 4, we suggest to expand the adaptive search region method by using an adaptive patch-kernel type, selected out of two possible types. NLM is commonly used with only one type of patch-kernel for the dissimilarity measure computation. In our analysis, two patch-kernels were explored: the Uniform and a Box patch-kernel. We suggest to select the pixel-wise appropriate kernel type based on local structure. Simulations reveal that the Uniform patch kernel is more adequate for smooth regions, whereas the Box patch-kernel is more adequate for texture or edges. Consequently, pixels that are characterized by a large adaptive search region, thus considered “smooth”, are associated with the Uniform kernel, and pixels that are characterized by a small search region are associated with the Box kernel. We compare the proposed adaptive NLM performance when applied using a single patch-

kernel to using an adaptive patch-kernel, and show that the adaptive kernel scheme improves denoising results.

Chapter 5 discusses the correlation between the dissimilarities of pixels associated to the adaptive search region and its effect on the model-based scheme presented in Chapter 3. We tackle the correlation analysis by considering first the correlation due to the mutual reference patch (Case 1) to whom all the patches in a given search region are compared, assuming no patch overlaps. Then we add the effect of overlap between patches, but not with the reference patch (Case 2), and finally we address the most general case in which overlapping patches may also overlap the reference patch (Case 3). The details of the statistical analysis are given in Appendix B and of the application of the analysis on the model-based method are elaborated in Appendix C.

Chapter 6 discusses the performance of the Correlation-Dependent Model-Based scheme, introduced in Chapter 5, and compares it to the model-based scheme of Chapter 4. In addition, we refer at the end of the chapter to the sensitivity of the two adaptive schemes and the standard NLM to an error in the noise variance estimation.

In Chapter 7, we suggest to integrate our model-based approach that defines an adaptive search region for each pixel in the image, into the BM3D [9] denoising method. This integration saves computations and eliminates the need for parameter calibration. We compare the denoising performances of the original BM3D and the BM3D combined with our model-based approach and show that they are comparable. Appendix D provides an elaborated explanation of the BM3D algorithm.

In Chapter 8, we refer to the Poisson noise model and present the flow that enables using the NLM denoising method that is suitable to an additive Gaussian noise, for Poisson noise denoising. This flow involves using a VST, specifically the Anscombe transform, which converts the Poisson noise to an additive Gaussian noise with unit variance. We explore the performance of the proposed adaptive NLM scheme and the model-based BM3D with

Poisson noise.

In Chapter 9, we present a summary of of the topics discussed in the thesis and suggest some open issues for future research.

2. Standard Non-Local Means Image Denoising

2.1. Introduction

In the presence of noise, the original pixel value is lost. Neighborhood filters (a class of filters to which the Non-Local Means filter is a member of) reduce the noise by selecting for each pixel i a set of pixels S_i characterized by both spatial proximity and similar gray level values. These filters proceed by replacing the gray level value of i by the average over the set S_i.

This chapter will focus on the Non-Local Means (NLM) [5] neighborhood filter that attempts to take advantage of the redundancy and self similarity of the image. The filter defines the denoised value of pixel i by applying a weighted average on the pixels assigned to the set S_i. The algorithm assigns a weight to a pixel $j \in S_i$ by comparing a small neighborhood around pixel j to a small neighborhood around the pixel of interest i (POI) . This weight is proportional to the similarity between the pixels' neighborhoods. In this manner, pixels with similar neighborhood to pixel i will be assigned a higher weight, thus have a more significant contribution to the weighted average process. Consequently, NLM provides a very efficient denoising procedure that preserves edges and texture while smoothing non-textured regions.

In this chapter, the NLM algorithm is described along with a proof of its consistency under stationarity conditions. Moreover, the importance of several parameters which affect the NLM performance is discussed.

2.2. Neighborhood Filters

Primitive neighborhood filters replace the gray value of a pixel with an average of the proximate pixels values. However, the most similar pixels to a given pixel have no reason to be spatially close to it. In 1999 Efros and Leung [14] used non-local self-similarities to synthesize textures and to fill in holes in images. Their algorithm scans a vast portion of the image in search of all the pixels whose neighborhood resembles to the neighborhood of the POI. The resemblance is evaluated by comparing windows, of defined size, centered at each compared pixel, and not just the value of the compared pixel itself (i.e., window center). This technique resembles the methodology of the sigma filter, invented by J.S. Lee [21] in 1983. This filter is motivated by the structure of the Gaussian PDF, and it smoothes the image noise by averaging only those neighborhood pixels which have the intensities, within a fixed sigma range, of the POI.

Let X and Y be the original and the observed noisy images, respectively. It is assumed that the original image is corrupted by a Gaussian noise N with a zero mean and a known standard deviation σ_n, such that,

$$Y = X + N,\ N \sim \mathcal{N}\left(0, \sigma_n^2\right) \tag{2.1}$$

Then, the filtered value by the sigma-filter strategy can be written as:

$$\hat{X}^{Lee}_{i,h,\rho} = \frac{1}{C_i} \sum_{j \in S_{i,\rho}} Y_j exp\left\{ -\frac{\left|Y_i - Y_j\right|^2}{h^2} \right\} \tag{2.2}$$

where only pixels inside the defined region $S_{i,\rho}$ are averaged, ρ defines the size of the search region, h controls the pixels dissimilarity and C_i is a normalization factor. The SUSAN algorithm [36] and the bilateral filter [39] make this process more symmetric by involving a bilateral Gaussian depending on both spatial proximity and gray level. This leads to the following:

$$\hat{X}^{BLF}_{i,h,\rho} = \frac{1}{C_i} \sum_{j \in S_{i,\rho}} Y_j exp \left\{ -\frac{\left|Y_i - Y_j\right|^2}{h^2} - \frac{|i-j|^2}{\rho^2} \right\} \tag{2.3}$$

The bilateral filters perform better denoising than Lee's statistical filter. They maintain sharp boundaries, since they average pixels belonging to the same region as the POI. However, Bilateral filters fail when the standard deviation of the noise exceeds the edge contrast.

2.3. The Non-Local Means (NLM) Algorithm

The NLM algorithm is inspired by the neighborhood filters. It takes advantage of the high degree of redundancy in any natural image by assuming that every small patch in a natural image has many similar patches in the same image. One can define a search region centered at pixel i, of size $M \times M$, such that $S_i = \left\{ j \,|\, |i-j| \leq \frac{M-1}{2} \right\}$. Specifically, in Texture synthesis [14], a sub-set of similar pixels, denoted $k \in S_i^S \subseteq S_i$, is extracted such that a patch around k resembles to a patch around i, by defining an adequate similarity measure. All pixels in that sub-set can be used for predicting the value at i. The fact that such a self-similarity exists proves image redundancy and matches the image regularity assumption.

In standard NLM, all the pixels that are included in S_i are used for the weighted averaging process, such that the weights are determined based on their resemblance to the POI, as explained next.

Assuming the noise model presented in eqn. (2.1), each pixel in the restored image is

derived as a weighted average of all gray values within a defined search region:

$$\hat{X}_i = \sum_{j \in S_i} w_{i,j} Y_j \tag{2.4}$$

where i represents a pixel index, S_i refers to a rectangular search region of size $M \times M$ centered at pixel i . The normalized weights, which can be referred to as similarity probabilities, are defined as:

$$w_{i,j} = \frac{1}{W_i} exp\left\{ -\frac{d_i(j)}{h^2} \right\} \tag{2.5}$$

such that , $W_i = \sum_{j \in S_i} w_{i,j}$, is a weight normalization factor, $d_i(j)$ is a dissimilarity measure, and h is the weight smoothing parameter, which is typically controlled manually in the algorithm. Choosing a very small h leads to noisy results almost identical to the input, while a very large h gives a uniform weighting, typically resulting in an overly-smoothed image. A more comprehensive discussion on the choice of this parameter will be held in section 2.5. The dissimilarity measure $d_i(j)$ is defined over the corresponding similarity patches as follows:

$$d_i(j) = \left\| Y(A_i) - Y\left(A_j\right) \right\|_{2,a}^2 \tag{2.6}$$

A_i represents a square similarity patch of size $p \times p$ centered at pixel i ($p < M$) and $Y(A_i)$ represents the pixel values of the corresponding similarity patch . The similarity patches may overlap within a given S_i and are defined such that their central pixel is included in S_i, not necessarily the whole patch. The vector norm is the Euclidean difference, weighted by a Gaussian kernel of zero mean and variance a that is used to smooth out the neighborhood while calculating the weights. This filter reduces the effect of differences in pixel intensities as they get spatially further away from the center of the patch. In practice, instead of a Gaussian kernel, simpler kernels are used: a Uniform kernel (which assigns the

same weights to all the pixels of the similarity patch), whose corresponding dissimilarity measure is denoted $d_i^U(j)$, and a Box kernel, illustrated in Figure 2.1, whose corresponding dissimilarity measure is denoted $d_i^B(j)$. Efros and Leung [14] showed that the L_2 distance is a reliable measure for the comparison of image windows in a texture search region. This measure is adapted to any additive white noise such that noise alters the distance between windows in a uniform way, as shown herein:

$$\mathrm{E}\left\|Y(A_i) - Y\left(A_j\right)\right\|_{2,a}^2 = \mathrm{E}\left\|X(A_i) - X\left(A_j\right)\right\|_{2,a}^2 + 2\sigma_n^2 \tag{2.7}$$

This equality shows that, in expectation, the Euclidean distance preserves the order of similarity between pixels. So the most similar pixels to i in Y are also expected to be the most similar pixels to i in X.

Refer to Figure 2.2 for a schematic view of a chosen search region and its respective similarity patches.

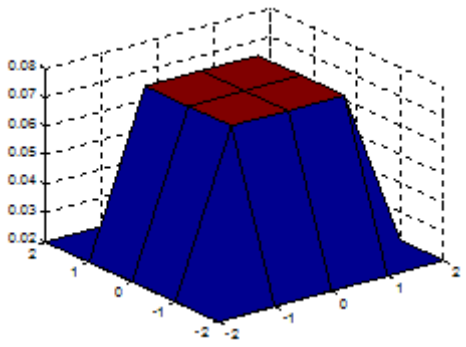


Figure 2.1.: Box kernel used for dissimilarity measure computation

Pay attention to the difference in weight definition between the NLM filter and the sigma-filter. The NLM defines pixel dissimilarity as the difference between pixel neighborhoods

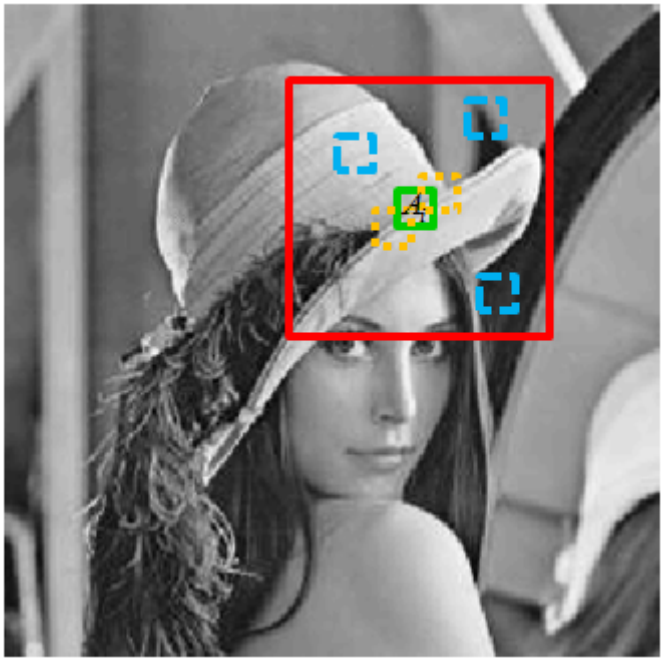


Figure 2.2.: The image Lena with a chosen search region S_i (marked in red) and respective similarity patches. The reference patch (A_i) is marked in green. Several compared patches $\left(A_j\right)$, within the search region, are marked by a light dashed orange contour (for patches which resemble A_i, i.e., having a small dissimilarity measure value) and a heavy dashed blue contour (for patches with high dissimilarity measure value).

(see eqn. (2.6)) , whereas the sigma-filter refers to the absolute difference between pixel values in a defined search region (see eqn. (2.2)).

The weights computation is a bottle neck of the NLM algorithm. Therefore, restricting the size of the search region M^2 is important for a practical implementation. Following recommendations presented in [32], that suggest to restrict the search region to $M \in [11,15]$ (see sub-section 2.5.3), the search region here is restricted to 11×11 $(M = 11)$ and the similarity patch is set to 5×5 $(p = 5)$. Consequently, for a $K \times K$ image, the final complexity of the algorithm is about $p^2 \times M^2 \times K^2$.

2.4. Statistical Consistency of the NLM

Estimation theory determines that if the original and observed images are considered as a realization of two random fields X and Y , then the best estimate of X is given by the conditional expectation

$$\hat{X}_i = \mathrm{E}\left[X_i|Y\right], \ \forall i \in X \tag{2.8}$$

The best estimate at a pixel i depends on the values of Y in the whole image. In many applications, e.g., texture synthesis, there is a restricted access to a small neighborhood of the pixel of interest (POI). In the denoising case, estimation of X_i may be based on a defined neighborhood centered at pixel i and not on the whole image. It can be shown that eqn. (2.8) also holds for a restricted neighborhood around pixel i, denoted S_i :

$$X_i = \mathrm{E}\left[X_i|Y\left(S_i\right)\right], \ \forall i \in X \tag{2.9}$$

Hence, the justification of restricting the search region in the NLM denoising process.

Image consistency means that the value at a given pixel i is directly influenced only by the values of its neighboring pixels [31]. Moreover, the defined search region is assumed to be stationary, i.e., it includes pixels whose characterizing distribution is space-invariant (for shifts within the region itself). Therefore, it can be concluded that the NLM methodology matches image consistency under stationarity conditions. This conclusion coincides with image redundancy, i.e., the fact that the image consists of many samples of every image detail. It is a crucial point in understanding the performance of the NLM algorithm. In addition, since the image is assumed to be a mixing process (refer to [31]), regions become more independent as their spatial distance increases, which is intuitively true for natural images.

2.5. Setting NLM Parameters

Exploring eqns. $(2.4) - (2.6)$ reveals that NLM denoising depends on setting the following parameters:

1. The similarity patch size $p \times p$, which is often set to 5×5 or 7×7, but should a priori be related to the scale of objects in the image, i.e., contrasted small details should be characterized by a smaller patch size than smooth regions.
2. The weight smoothing parameter h should be proportional to the typical distance between similar patches, and should also depend on the noise STD, as suggested in[40]. It should also be related to the regularity of the image, as will be explained in subsection 2.5.2.
3. The size of the search region S_i, $M \times M$, which has a dramatic impact on the computation time, but which also has an influence on the visual quality of the results.
4. The weight of the central pixel in a given S_i. This pixel should be treated differently than the other pixels in the search region, as it serves as the reference pixel and also participates in the weighted average procedure.
5. The patch kernel used for the dissimilarity measure computation.

In fact, these parameters are far from being independent, and setting their values is not an easy task. The rest of this section deals with the analysis of these parameters and their effect on to the NLM denoising performance.

2.5.1. Similarity-Patch Size

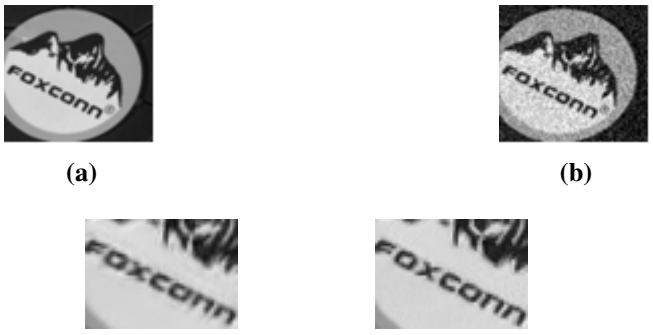
The dissimilarity measure can be defined as difference between similarity patches in either the spatial domain or in any adequate transform domain. The domain determines how the patches similarities are computed. Using a large patch size reduces the importance of low

contrasted small details, becoming more blurred. Moreover, it allows a more robust discrimination between areas that are not actually similar. In Figure 2.3(a), when a small similarity patch size is being used, Lena's skin looks mottled. It looks smoother with a larger patch size (Figure 2.3(b)), at the expense of blurry feathers. On the other hand, if the image has textures with highly contrasted transitions or that are characterized by rare pattern occurrences, using a too large patch prevents the algorithm from finding redundancies, as shown in Figures 2.4. For a large similarity patch size (Figure 2.4(c)), it is very difficult to find similar patches around the letters, thus a noisy halo appears. Using a smaller similarity patch size (Figure 2.4(d)) reduces the spread of the halo since it captures the local texture more accurately.

This phenomenon is referred to as the rare patch effect and is visually characterized by a noise halo around edges, due to lack of redundancy of the defined patches in the search region. Consequently, the patch size should ideally be chosen depending on the local scale of the image, where smooth regions define a large scale and edges or textures define a small scale. Deledalle et al. [10] address the problem of the rare patch effect by substituting the square isotropic patches of fixed size by spatially adaptive patch shapes that can exploit local image geometry. Their suggestion obviously complicates the implementation of the NLM filter and restricts the similarity patch shape to be chosen out of a selected bank of shapes. However, using a locally adapted h should make the choice of the size of the similarity patch less critical, as will be explained in subsection 2.5.2.

(a) **(b)**

Figure 2.3.: Denoising of the image Lena with $\sigma_n = 20$, $h = \sigma_n$, $M = 11$ and Box kernel for the dissimilarity measure computations. (a) Using 3×3 similarity patches, (b) Using 5×5 similarity patches.



(a) **(b)**

(c) **(d)**

Figure 2.4.: Choice of the patch size: (a) Original image [43], (b) Noisy image with $\sigma_n = 20$, (c) Zoomed-in view of NLM denoising with $\sigma_n = 20$, $h = \sigma_n$, $M = 11$ and $p = 9$, (d) Zoomed-in view of NLM denoising with $\sigma_n = 20$, $h = \sigma_n$, $M = 11$ and $p = 5$. Box kernel was used for the dissimilarity measure computations.

2.5.2. Weight-Smoothing Parameter

The weight-smoothing parameter (h), which quantifies how fast the weights decay with increasing dissimilarity of respective patches, is usually set to be proportional to the noise standard deviation [40]. Since the performance of the NLM filter is sensitive to h, its value

should be chosen carefully. There is apparently a contradiction between the idea of adapting the smoothing parameter (whether globally or locally) to the content of the image and the observation of several authors (e.g., [40]) that h should be set proportional to the noise standard deviation . Figure 2.5 shows that the relation between the optimal global parameter h and noise STD is indeed approximately linear and it is remarkable that the slope does not vary much between different images. This fact relates to the global effect of this parameter, however empirical results show that using a fixed global h value (where h is proportional to noise STD) causes over-smoothing of highly textured regions. Whereas, a low global value may preserve texture and edges, but results in insufficient denoising of smooth regions, which appear grainy. Consequently, a local value of h, which depends on the local structure of the image, should provide sufficient denoising of smooth regions as well as preserve highly textured areas. Therefore, this empirical rule of linear relation between the value of h and the noise STD only gives a rough idea of the optimal value. Duval et al. [12] reconcile these two points of view by suggesting a method that sets a locally adaptive h which is linear with the noise STD. Their method involves several runs of the NLM algorithm with different values of h. Then, a SURE estimator is calculated pixel-wise in order to select the local optimal value of h, out of the pre-selected values.

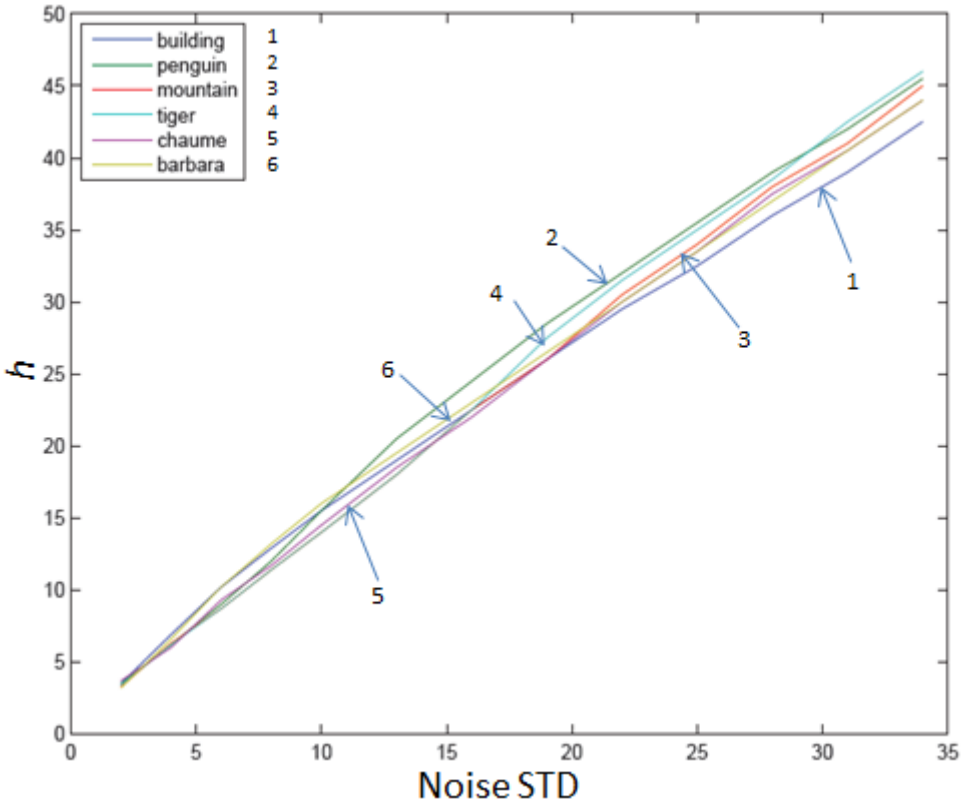
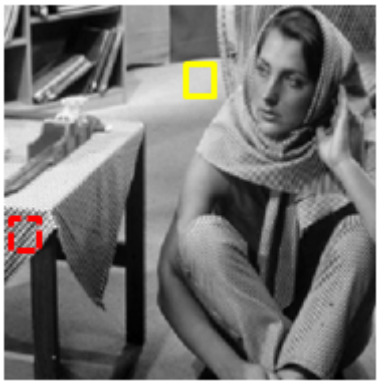


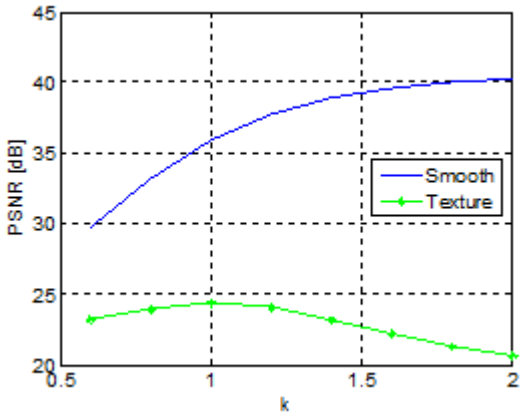
Figure 2.5.: Evolution of the optimal global smoothing parameter h when the noise level varies, for six different images [12].

Figure 2.6 demonstrates the importance of setting a local content-based value of h. It is based on a simulation we carried out, in which two representative patches were extracted from the image Barbara: one corresponds to a smooth region and the other corresponds to a textured region. The smoothing parameter was defined as $h = k \cdot \sigma_n$, and the NLM algorithm was applied on the selected image patch by using different values of the parameter k in order to determine which value provides the best denoising results. Each optimization is a result of averaging over ten noise realizations. Figure 2.6 (b) represents the optimization process over the smooth region, whereas Figure 2.6 (c) represents the process over the textured region. It can be seen that the smooth region requires a larger k compared to the textured region. This simulation shows that indeed h is sensitive to image local structure, as stated

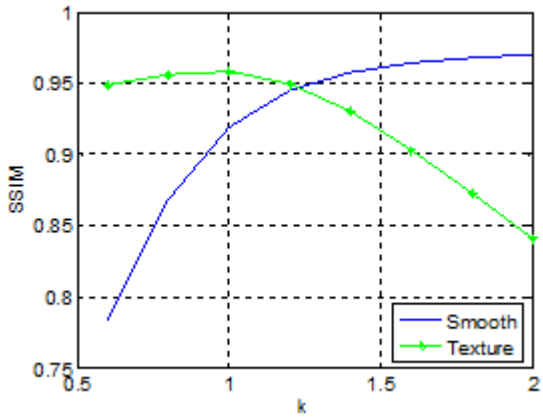
above.



(a)



(b)



(c)

Figure 2.6.: Simulation which demonstrates the sensitivity of h to image local structure. NLM denoising was applied on the noisy image Barbara (256×256) with $\sigma_n = 20$ and the parameters: $p = 5$, $M = 11$, $h = k \cdot \sigma_n$, where $k = [0.6, 2]$ and Box patch-kernel. (a) Barbara with two selected patches: the solid yellow patch corresponds to a smooth region, whereas the dashed red patch corresponds to a textured region, (b) PSNR vs. k curve, (c) SSIM vs. k curve.

2.5.3. Search-Region Size

Intuitively the search region should be as large as possible to have as many "copies" of the patch. However, it should be selected to be as small as possible, since it crucially affects

the computation time. Actually, most natural images are characterized by local similarities, thus there appears a need to use a very large search region. Salmon at al. [34] demonstrate that for most standard images, the gain of using a large search region size is insignificant for a parameter greater than 15, with a fixed choice of p (refer to Figure 2.7). Moreover, sizes higher than 15 have a negative influence on the NLM denoising performance for both medium and high noise levels. This phenomenon is due to the accumulation of small positive weights, leading to average of non-similar patches, hence biasing the estimation. Duval et. al [12] suggest to minimize the loss caused by a large search region by using truncated weights, i.e., a hard thresholding operator. When imposing a small threshold value, the NLM filter is almost insensitive to the increase of the search region. Of course, the weight thresholding parameter adds another degree of freedom to the algorithm, thus this suggestion is not optimal.

Levin at. al [23] study the relation between denoising gain and search region size requirements in patch-based denoising techniques. They conclude that each image pixel has a finite compact region of informative neighboring pixels based on its characterizing structure. Intuitively, the size distribution of these regions must directly impact both denoising error (small error implies higher PSNR) vs. search window size and its limit as the window size grows indefinitely. An infinite search region is obviously impractical. However, choosing only the most similar pixels to a POI (where similarity is measured based on neighborhood comparison, see eqn.(2.6)), within a finite defined search region, may serve as a practical solution to the content-based search region size. This suggested solution will be further discussed in chapter 3.

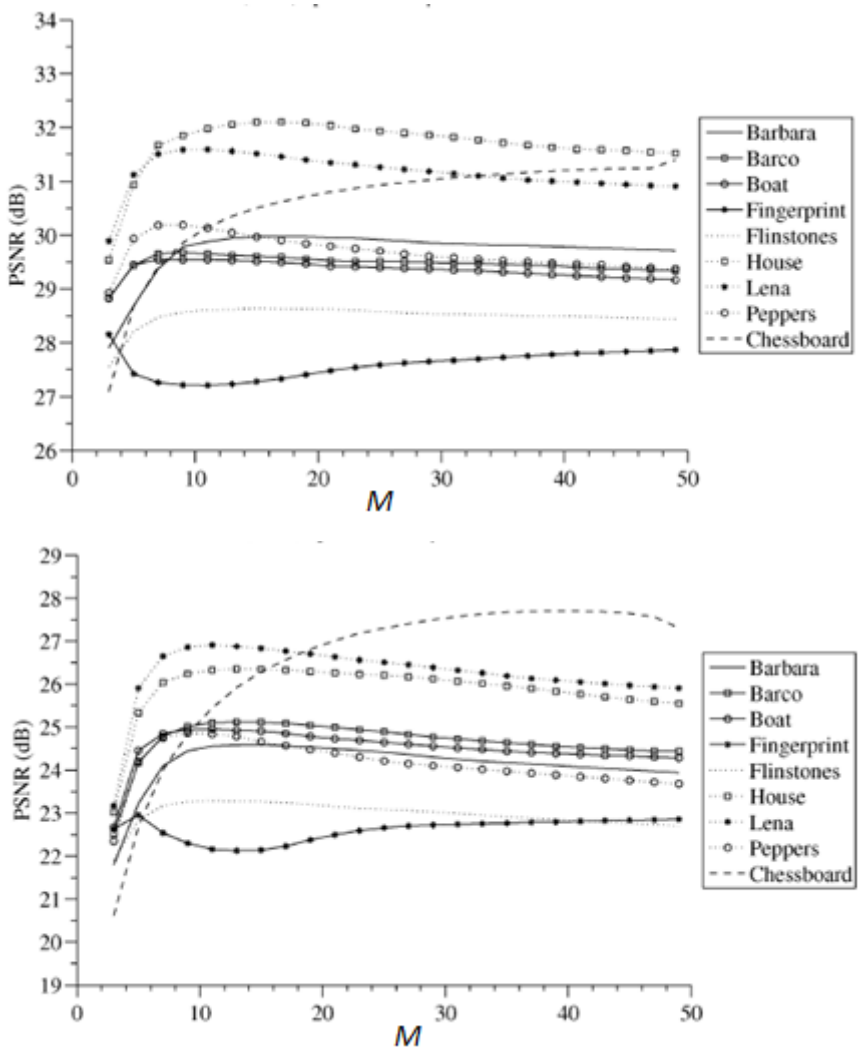


Figure 2.7.: [34] Influence of the size of the search region on the PSNR of the NLM procedure, with $p = 5$, weight of the central pixel is the maximal weight within the weights of the search region and h is chosen as the optimal value within the range $\left[\sqrt{3}\sigma_n, \sqrt{72}\sigma_n\right]$. The top figure refers to $\sigma_n = 20$ whereas the bottom figure refers to $\sigma_n = 50$.

Figure 2.8 displays the empirical PSNR as function of search region size. In this simulation, clean and noisy test patches pairs were used, such that for each noisy patch, the largest M, at which estimation is still reliable, was determined by comparing the denoising results with its paired noise-free patch. The patches were divided into groups, G_l, based on the largest

window size l ($M = [1, l]$) at which the estimate is still reliable. The figure displays, for each group, the empirical PSNR averaged over the group's patches as a function of window size M. Moreover, the mean gradient magnitude was computed for each group based on the noise-free data. As expected, groups that include mostly patches with large gradients (texture) are associated with a smaller search region size l. These groups of patches correspond to PSNR curves that are lower and also flatten earlier in the figure. In contrast, groups that include smooth patches are associated with small mean gradient magnitude and gain more from an increase in search region size. These patches correspond to the higher curves that flatten later in the figure. Consequently, it seems that pixels that correspond to smooth regions require a larger search region compared with pixels that correspond to textured regions. Moreover, theses pixels are characterized by a higher PSNR when denoised using a larger search region.

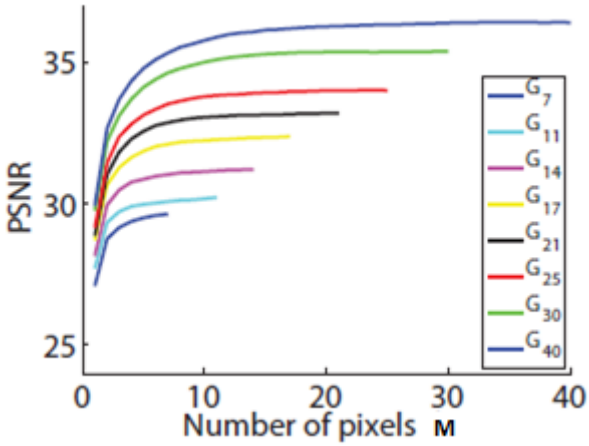


Figure 2.8.: [23] PSNR vs. size of search region d for patch groups G_l of varying complexity. Higher curves correspond to smooth regions, which flatten at larger patch dimensions. Textured regions correspond to lower curves which not only run out of samples sooner, but also their curves flatten earlier.

2.5.4. Reference-Pixel Weight

The role of the reference pixel (POI) is different in nature as it plays two roles at the same time. On one hand, it is the central pixel of the reference patch that is compared to the other patches, within a defined search region, and on the other hand it also participates in the averaging process together with all the others pixels of the search region. Several suggestions on how to set the weight of this pixel have arisen [11], as detailed herein. The central pixel weight shall be denoted $w_{i,i}$.

1. The original weight based on eqn. (1.5): $d_i(i) = 0$, $w_{i,i} = 1$ before normalization.
2. Assign the same value as the maximum of the other weights observed in the search region and then normalize the weights. Though this choice is not validated by theory, better results are obtained in practice.
3. Assign a zero weight, i.e., don't consider the central pixel in the weighted average process, $w_{i,i} = 0$.
4. Use Stein Unbiased Risk Estimator (SURE) [40] for weight computation. These weights are equivalent to replacing the central weight in the NLM procedure by $exp\left\{-\frac{2\sigma_n^2 M^2}{h^2}\right\}$ (without modifying the other weights), before normalization.

Salmon et al. [34] compared the performance of the different methods. Their results are illustrated in Figure 2.9. The histograms are ordered by the following methods: SURE estimator (4.), maximal weight (2.), original weight (1.) and zero weight (3.). It can be observed that for most of the images, with any noise level, the first two methods, i.e., the SURE weight and the maximal weight provide the best results. Consequently, in this work, the central pixel weight is chosen as the maximal weight within the pixels in the corresponding search region.

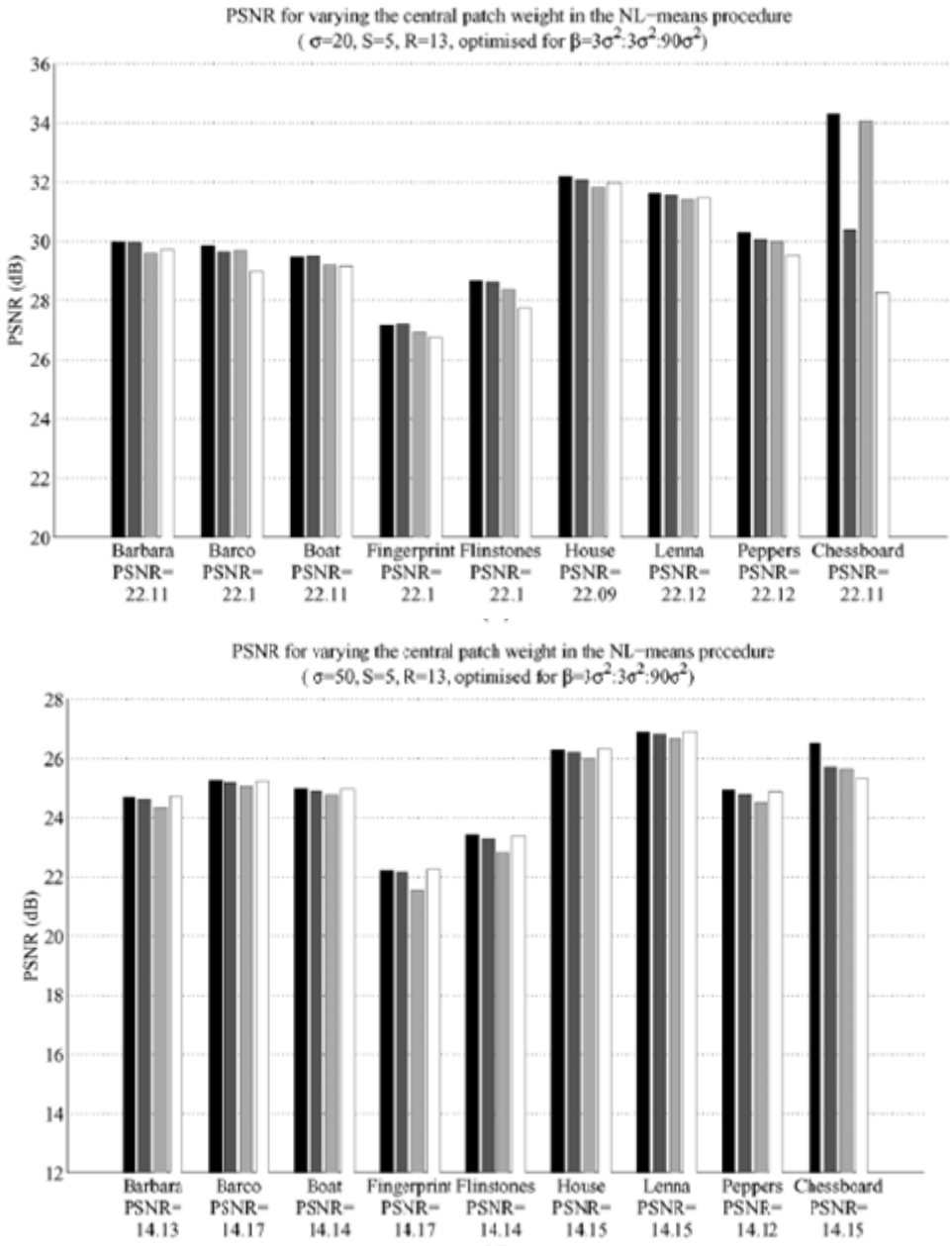
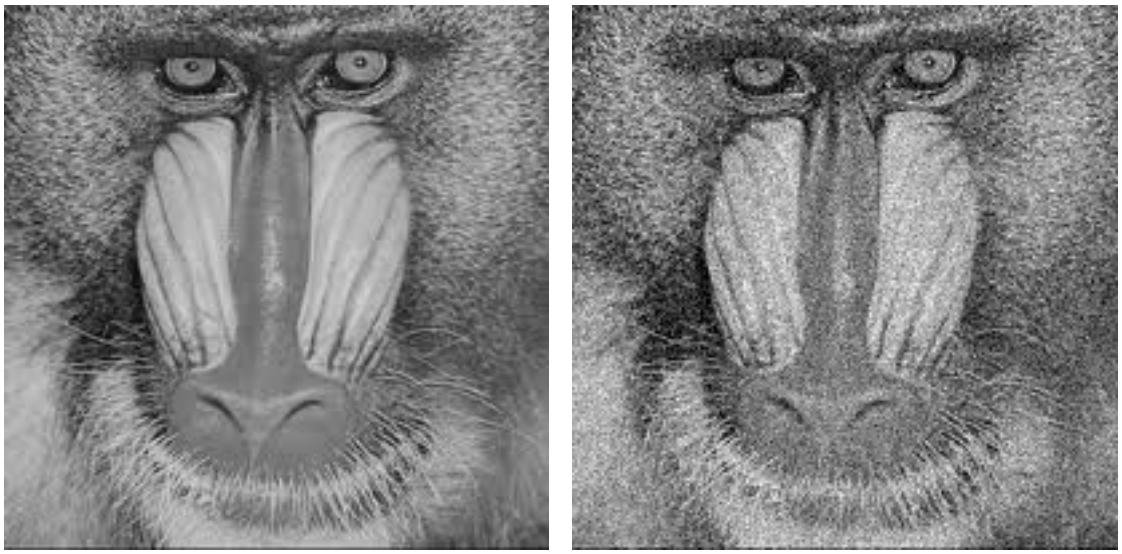


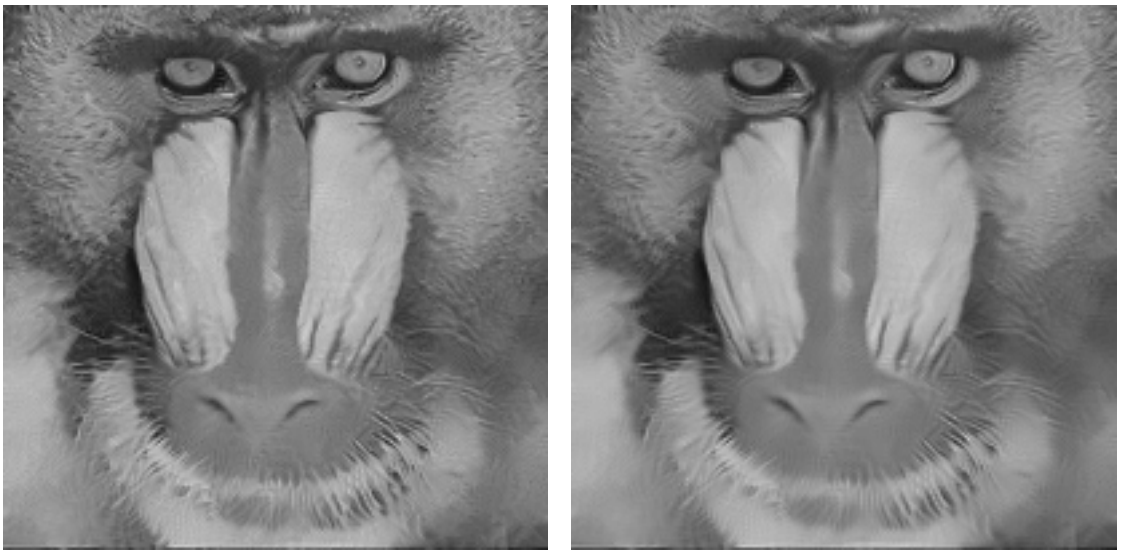
Figure 2.9.: [34] Comparing performance of NLM by changing the weight of the central pixel (in order from black to white: SURE, Max, Original, Zero) with two levels of noise and $p = 5$, $M = 13$ and h is chosen as the optimal value within the range $\left[\sqrt{3}\sigma_n, \sqrt{90}\sigma_n\right]$.The top figure refers to $\sigma_n = 20$ whereas the bottom figure refers to $\sigma_n = 50$. The PSNR given below the name of the image is the one obtained with the noisy version of each image.

2.5.5. Patch Kernel

As explained earlier, in sub-section 2.3, the dissimilarity measure is computed using the vector norm of the difference between the central patch and any compared patch (within the defined search region), weighted by a Gaussian kernel of zero mean and variance a, as shown in eqn. (2.6). The Gaussian kernel is used to smooth out the similarity patch while calculating the weights. This filter reduces the effect of differences in pixel intensities as they get spatially further away from the center of the patch. In practice, instead of a Gaussian kernel, simpler kernel is used, the Box kernel, as illustrated in Figure 2.1. Another, even simpler kernel that is used, is the Uniform kernel. It assigns the same weights to all the pixels of the similarity patch, thus does not weight the data according to its spatial distance from the patch center. Simulations suggest that the Uniform kernel is more adequate for smooth regions, whereas the Box kernel is more adequate for texture or edges. Consequently, there is an added value of using a Box kernel. Figure 2.10 (c), which represents a standard NLM denoising using a Box kernel, preserves structure (see Baboon's fur) but is characterized by granularity in smooth regions (see Baboon's nose). Fig. 2.10 (d) presents the denoising results of the standard NLM with a Uniform kernel. It shows that the granularity effect is reduced in smooth regions, but texture and edges are over-smoothed. Further discussion regarding the importance of the patch kernel will be held in chapter 3.



(a) **(b)**



(c) **(d)**

Figure 2.10.: Denoising of the image Baboon with $\sigma_n = 20$, using different patch kernels and $h = \sigma_n$, $p = 5$, $M = 11$. (a) Original image, (b) Noisy image, (c) Box kernel, (d) Uniform kernel.

2.6. Denoising Limits

Levin et. al [23] study absolute denoising limits and the convergence rate to them as a function of the search-region size. Scale invariance is a fundamental property of natural images. In order to characterize this property, a dead leaves image formation model is considered, e.g. [1], whereby an image is a random collection of piecewise constant segments, whose size is drawn from a scale-invariant distribution and whose intensity is drawn *i.i.d.* from a uniform distribution. This yields perfect correlation between pixels in the same region. To further simplify the analysis, an edge oracle, which gives the exact locations of edges in the image, is assumed. The optimal denoising is then to average all observations in a segment. For a pixel belonging to segment of size s pixels, the Minimum Mean Squared-Error (MMSE) is $\frac{\sigma_n^2}{s}$. Overall the expected reconstruction error with infinite-sized windows is $MMSE(s) = \int p(s) \cdot \frac{\sigma_n^2}{s}\, ds$ where $p(s)$ is the probability that a pixel belongs to a segment pf size s pixels. Alvarez at. al [2] show that scale-invariance implies that the probability that a random image pixel belongs to a segment of size s is of the form $p(s) \propto \frac{1}{s}$. To get a sense of the empirical size distribution of nearly-constant-intensity regions in natural images, Levin at al. perform a simple experiment inspired by[31]. For a random set of pixels $\{X_i\}$,the size $d(i)$ of the connected region whose pixel values differ from X_i by at most a threshold T was computed, i.e.: $d(i) = card\left\{ j|\ \left|X_j - X_i\right| \leq T \right\}$. The threshold T corresponds to the contrast within the segment, such that a lower value refers to a more uniform segment. The empirical histogram $h(d)$ of region sizes follows a power law behavior $h(d) \propto d^{-\alpha}$ with $\alpha \approx 1$, as suggested by [2]. Levin et al. model the MMSE, as a function of search-region size, denoted $M = |S_i|$, as $MMSE(M) = e + \frac{c}{M}$, where $e = MMSE(M \to \infty)$ and c is some constant. The MMSE value for an infinite search region is set based on extrapolation of experimental curves which are the result of denoising applied with a finite d. Extrapolating this parametric law gives a ballpark estimate of the best achievable denoising, suggesting that some improvement, although modest, is still possible (see Table 2.1). While the extrapo-

lated value may not be exact, their analysis suggest that there is an inherent limit imposed by the statistics of natural images, which cannot be broken, no matter how sophisticated future denoising algorithms will be. Table 2.1 compares the PSNR of existing algorithms (BM3D [9] and K-SVD [28]) to the predicted $PSNR_\infty$, over 20K test patches using the power law fit based on a pool of 10^8samples, i.e., for each test patch, the corresponding similar patches were extracted from the pool . The comparison suggests that depending on noise level σ_n, current methods may still be improved by $0.6 - 2.3$dB.

σ_n	35	50	75	100
Extrapolated Bound ($PSNR_\infty$)	30.6	28.8	27.3	26.3
K-SVD [28]	28.7	26.9	25	23.7
BM3D [9]	30	28.1	26.3	25

Table 2.1.: Extrapolated optimal denoising in PSNR, and the results of recent algorithms. It can be observed that a modest room for improvement still exists.

2.7. Chapter Summary

In this chapter, the Non-Local Means denoising scheme was introduced in detail. This filter is part of the neighborhood filters family. These filters estimate a POI by using a weighted average of pixels located in a search region associated with a POI. The weights are inversely proportional to the dissimilarity between a small neighborhood of the POI and a corresponding small neighborhood of pixels within the search region. This filter assumes stationarity of the search region data and depends on five parameters whose values affect the denoising performance. These parameters are the size of the similarity patch (p), the weight-smoothing parameter (h), the size of the search region (M), the weight of the central pixel $(w_{i,i})$, and the patch kernel type.

3. Model-Based Non-Local Means Image Denoising

3.1. Introduction

The Non-Local Means (NLM) denoising algorithm uses a weighted average of pixels, within a defined search region of the image, to estimate a noise-free pixel value. The search region is usually a rectangular neighborhood, centered at the pixel of interest (POI), which may include pixels whose original gray value do not match the value of the original central pixel. Consequently, their participation in the averaging process degrades denoising performance. To eliminate their effect, researchers suggest creating an adaptive search-region which excludes those dissimilar pixels.

This chapter presents a novel model-based method, which defines a set of similar pixels to the POI, from the initial search region, using the statistical distribution of the dissimilarity measure. Experimental results show that the proposed algorithm has better performance than the original one in terms of PSNR, Structural Similarity (SSIM) [41], and visual quality and is found to be more efficient than other examined approaches.

3.2. NLM Applied With An Adaptive Search Region - Prior Art

The uniqueness of the NLM approach lies in its ability to exploit spatial correlation in a defined neighborhood (search region) for noise removal. As mentioned in section 3.1, the search region may include pixels which differ in their original gray value from that of the POI and their participation in the averaging process may degrade the denoising performance. Their negative effect can be alleviated by the weight smoothing parameter (h). This parameter plays the same role as kernel support, such that the larger the support, the smoother the image becomes. Hence, for textural regions a smaller value of h should be used than for smooth regions, for the same given noise STD. In that manner, dissimilar pixels will be assigned a lower weight (refer to sub-section 2.5.2 for more details). As a result, there are NLM modifications that suggest using an adaptive h value [12, 11], which is matched to local structure. As discussed below, an alternative to using an adaptive h is to replace the isotropic square search region in the original NLM by an adaptive anisotropic region in which the most similar pixels to the POI are selected (based on comparing local neighborhoods). This anisotropic neighborhood can better exploit the local image structure. Consequently, denoising performance is improved, especially for pixels that belong to textured regions.

Mahmoudi et al. [29] propose to pre-classify the neighborhoods based on local average and gradients, but the calculation of the gradient is affected by noise. Coupe et al. [8] and Kervrann et al. [18] use patch average and patch variance to rule out dissimilar patches. Dinesh at al. [11] suggest a correlation-based patch classification method. The correlation is computed using an inner product between two normalized patches. Only patches, within the search region, whose correlation (with respect to the reference patch) is higher than a pre-defined threshold, are considered during the averaging process. The problem with these methods is that the measures which affect pre-selection of pixels use global thresh-

olds which are chosen somewhat heuristically and may vary based on image characteristics, hence imply lack of robustness. Azzabou et al. [3] suggest partitioning the image into two classes: noisy smooth zones and noisy texture/edge zones. Each pixel is characterized by a statistical model that defines a membership degree to each class. The method relies on a prior, which does not necessarily satisfy all explored images, and involves a Gaussian Mixture Model as well as EM optimization that are computationally expensive. Orchard et al. [32] propose an alternative strategy that uses the SVD to more efficiently eliminate dissimilar pixel pairs. The method relies on dimensionality reduction of the image patches by setting a global dimension for all patches. This dimension value may cause over-smoothing since texture and edge patches should be characterized with a higher dimension than smooth patches. In [4], Brox et al. suggest to classify pixels by using a cluster tree approach and K-Means (K=2) with pre-selected parameters that define the classification. This is equivalent to image segmentation based on iterative binary classification and hence is not necessarily robust under noisy conditions. Kervran et. al [17] propose to use an adaptive isotropic neighborhood, such that the proper search region size is set in a manner that it balances the accuracy of approximation and the stochastic error, at each spatial position. The drawback of this approach is that it restricts the search region to be rectangular, and it explores several search window sizes per each pixel, which makes it time-consuming. Sun et al. [37] present a method that determines a pixel-wise adaptively shaped search region, within which the image is homogeneous. The method is subjected to a contiguous search region shape that is not necessarily the best shape within the pre-defined search region.

The following section describes our suggestion for an adaptive search region. The novelty of our approach is that pixel association with any of these two groups is based on a probabilistic model that characterizes the dissimilarity measure between two compared patches. Consequently, each pixel is characterized by a content-based search region constructed from the similar pixels group.

3.3. Search Region Pixel-Classification - Proposed Model

The adaptive approaches, presented in the previous section, suggest to partition the given search region (which can also be defined as the entire image) into two groups, based on pixels' similarity to the POI. These approaches are parameter-dependent or restrict the set of similar pixels to the POI to be contiguous. We too propose to partition the search region of a given pixel i (S_i) into two sets: a set of similar pixels (with respect to the reference pixel i), denoted here S_i^S, and a complement set of dissimilar pixels, denoted S_i^D. However, unlike some of the earlier approaches, the set S_i^S is not restricted to be contiguous, and the partition is determined on the basis of a statistical model of the dissimilarity measure. Similarly to other approaches, the weighted averaging is applied only to the pixels in the set S_i^S. A schematic view of a partition is depicted in Figure 3.1.

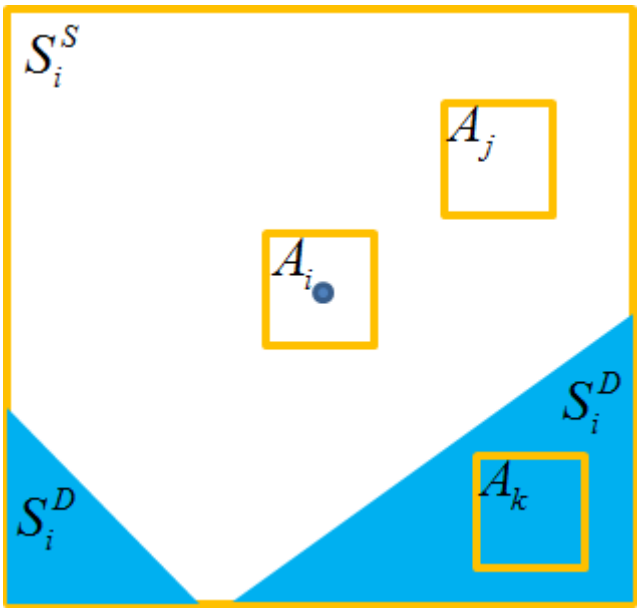


Figure 3.1.: Description of a search region divided into two sets. S_i^D is a group of pixels whose original gray value difference with respect to S_i^S is C (marked in light blue). A_j is a similarity patch included in the S_i^S set whereas A_k is a similarity patch included in the S_i^D set. Note that S_i^D is not restricted to be contiguous.

To derive the model, we refer at first to the normalized (by the noise variance) dissimilarity measure that uses a Uniform patch-kernel:

$$\frac{d_i^U(j)}{2\sigma_n^2} = \frac{1}{p^2}\frac{\left\|Y(A_i) - Y\left(A_j\right)\right\|_2^2}{2\sigma_n^2} = \frac{1}{p^2}\sum_{m\in A_i, l\in A_j}\left(\frac{Y_m - Y_l}{\sqrt{2}\sigma_n}\right)^2,\ j \in S_i^S \tag{3.1}$$

Refer to eqn. (2.6) for the general definition of the dissimilarity measure. A_i is the similarity patch centered at pixel i (the POI) and A_j is a compared similarity patch, centered at pixel $j \in S_i$. Moreover, the size of the similarity patches is set to be $p \times p$.

Assuming that the original gray values of all the pixels in S_i^S are the same as pixel i, i.e., $X_j = X_i\ \forall j \in S_i^S$ (refer to eqn. (2.1) for a description of the noise model), the following applies:

$$\sum_{m\in A_i, l\in A_j}\left(\frac{Y_m - Y_l}{\sqrt{2}\sigma_n}\right)^2 = \sum_{m\in A_i, l\in A_j}\left(\frac{N_m - N_l}{\sqrt{2}\sigma_n}\right)^2,\ \forall j \in S_i^S \tag{3.2}$$

The normalized noise difference $\frac{N_m - N_l}{\sqrt{2}\sigma_n}$ is distributed $\mathcal{N}(0,1)$ as a linear combination of two $i.i.d$ variables. We start with the simplifying assumption that the dissimilarity values related to the same reference patch over the search region are not correlated (in spite of patch overlaps and the use of the same reference patch). In chapter 4, we consider the effects of having a mutual reference patch, and patch overlaps.

LEMA 3.1 [15]: *Let $Z_1, ..., Z_k$ be independent, standard, normal variables, then the sum of their squares $Q = \sum_{m=1}^{k} Z_m^2$ is distributed according to the Chi-Square distribution with k degrees of freedom, i.e., $Q \sim \chi_k^2$.*

Under the current assumption and based on *Lema 3.1*, the sum, in eqn. (3.2), of p^2 squared independent standard normal variables has a *Chi-Square* distribution with p^2 degrees of freedom. For large p^2 , the *Chi-Square* distribution converges to a *Normal* distribution with

the following first two moments:

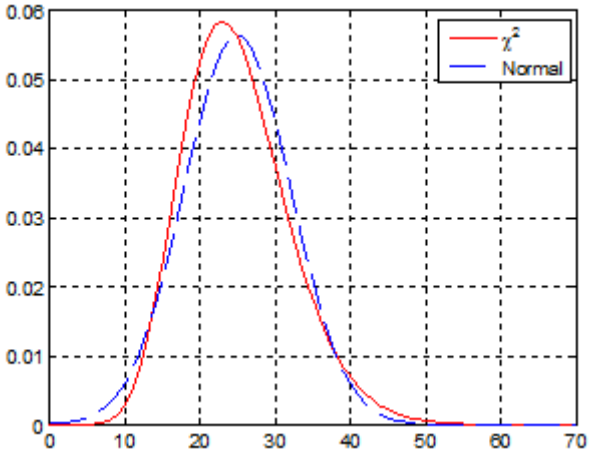
$$\chi^2_{p^2} \sim \mathcal{N}\left(p^2, 2p^2\right) \;\; for\; p \gg 1 \tag{3.3}$$

Figure 3.2(a) illustrates the goodness of fit between the two distributions, for $p^2 = 25$, as the similarity patch is typically chosen to be 5×5, i.e., $p = 5$. The figure shows that $p^2 = 25$ is sufficiently large to assume a convergence of the *Chi-Square* distribution to a Normal distribution. Figure 3.2(b) depicts the Kullbek Leibler (KL) Divergence[31] between the two distributions as a function of p.

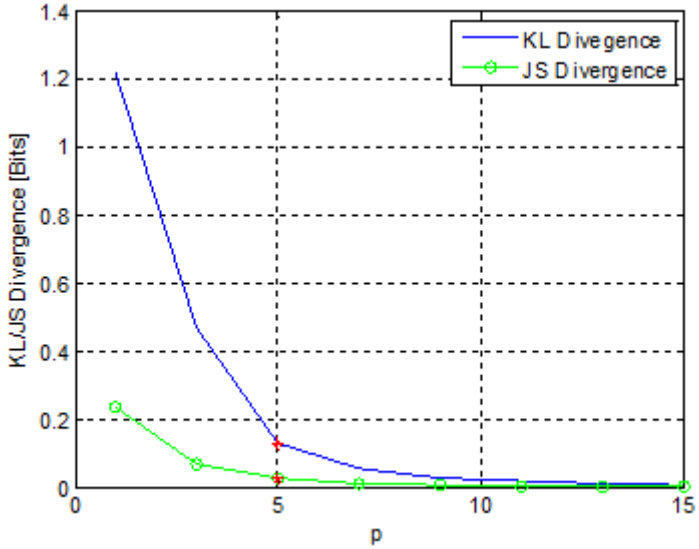
DEFINITION 3.1 [35]: *The Kullback-Leibler (KL) Divergence is a fundamental equation of information theory that quantifies the proximity (in bits) of two probability distributions. In our case, we wish to quantify how close is the Normal distribution, denoted here* $q^{\mathcal{N}}$ *to the Chi-Square, denoted* q^{χ^2}*. The corresponding KL Divergence is defined as* $D_{KL}\left(q^{\mathcal{N}}||q^{\chi^2}\right) = \sum_i q_i^{\mathcal{N}} log_2\left(\frac{q_i^{\mathcal{N}}}{q_i^{\chi^2}}\right)$*, where the index i refers to elements of an input vector and* $q_i^{\mathcal{N}}$, $q_i^{\chi^2}$ *are the respective discretized Normal and Chi-Square probabilities of the input vector elements.* D_{KL} *is non-negative, not symmetric in* $q^{\mathcal{N}}$ *and* q^{χ^2}*, zero if the distributions match exactly and can potentially equal infinity.*

DEFINITION 3.2 [19]: *The classical Jensen-Shannon Divergence is a non-negative symmetric derivative of the KL Divergence. It is defined as:* $D_{JS} = 0.5\left(D_{KL}\left(q^{\mathcal{N}}||q^{Mix}\right) + D_{KL}\left(q^{\chi^2}||q^{Mix}\right)\right)$ *where* $q^{Mix} = 0.5\left(q^{\mathcal{N}} + q^{\chi^2}\right)$ *is the average of the two distributions.*

The figure demonstrates that as the size of the similarity patch increases, the two explored distributions become more proximal, as expected. The red dots on the figure, which correspond to $p = 5$, show that for this patch size, the two distributions can be considered sufficiently proximal. Increasing p will provide a better approximation, but at the expense computational complexity.



(a)



(b)

Figure 3.2.: Goodness of fit between the Normal and Chi-Square distributions. (a) Normal and Chi-Square distributions for $p^2 = 25$ degrees of freedom, (b) KL Divergence and JS Divergence as a function of similarity patch size (p).

From the combination of eqns.(3.2) and (3.3) we get, for the uniform kernel:

$$\frac{p^2 d_i^U(j)}{2\sigma_n^2} \sim \chi^2_{p^2} \rightarrow \mathcal{N}\left(p^2, 2p^2\right) \quad for\ p \gg 1 \tag{3.4}$$

And hence the distribution of the *normalized* dissimilarity measure can be approximated (for sufficiently large p^2) by:

$$\tilde{d}_i^U(j) \triangleq \frac{d_i^U(j)}{2\sigma_n^2} \sim \mathcal{N}\left(1, \frac{2}{p^2}\right) \tag{3.5}$$

In the same manner, for a *general kernel*, the normalized dissimilarity measure is defined as follows:

$$\tilde{d}_i^G(j) \triangleq \frac{d_i^G(j)}{2\sigma_n^2} = \sum_{m\in A_i, l\in A_j, s\in\left[1,p^2\right]} \alpha_s \left(\frac{N_m - N_l}{\sqrt{2}\sigma_n}\right)^2, \; s.t. \sum_{s\in\left[1,p^2\right]} \alpha_s = 1, \tag{3.6}$$

where α_s represents the patch-kernel weights of a given $p \times p$ similarity patch (hence, $\alpha_s = p^{-2}$ for the Uniform patch kernel). This measure is distributed (in approximation) [19] as:

$$\tilde{d}_i^G(j) \sim \mathcal{N}\left(1, 2\sum_{s\in\left[1,p^2\right]} \alpha_s^2\right) \tag{3.7}$$

The *Box* patch kernel we use (for $p = 5$) is characterized by the following normalized coefficients α_s, arranged in a matrix form:

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1.888 & 1.888 & 1.888 & 0.5 \\ 0.5 & 1.888 & 1.888 & 1.888 & 0.5 \\ 0.5 & 1.888 & 1.888 & 1.888 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \cdot p^{-2} \tag{3.8}$$

We introduce the following definition for a General patch-kernel:

$$\kappa \triangleq \sum_{s\in\left[1,p^2\right]} \alpha_s^2 \tag{3.9}$$

The corresponding values of κ for the Uniform and the Box kernels (of size $p = 5$) are given by:

$$\begin{aligned} \text{For Uniform patch-kernel: } \kappa &= p^{-2} = 0.04, \ (p = 5) \\ \text{For Box patch-kernel: } \kappa &= 0.0578, \ (p = 5) \end{aligned} \tag{3.10}$$

The definition in eqn. (3.9) and the values of κ in eqn. (3.10) will be used from now on.

Thus, the normalized dissimilarities have the following distribution:

$$\tilde{d}_i^G(j), \tilde{d}_i^U(j) \sim \mathcal{N}(1, 2\kappa) \tag{3.11}$$

Model Application

We begin with the description of the proposed method for search region partitioning when the Uniform patch kernel is used.

Let's discuss the simplified case in which the search region can be divided into two uniform sub-regions, as illustrated in Figure 3.2. In the figure, S_i^D represents pixels whose original gray level difference with respect to the POI is C, i.e.,

$$Y_m - Y_l = C + N_m - N_l, \ \forall m \in A_i, l \in A_j, \textit{and} \ \forall j \in S_i^D \tag{3.12}$$

In a more general case, S_i is divided into multiple sub-regions, with a pixel-wise gray level difference $C_{m,l}$. This difference is defined for a pair of pixels m, l which corresponds to the same locations in similarity patches A_i, A_j respectively.

$$Y_m - Y_l = C_{m,l} + N_m - N_l, \ \forall m \in A_i, l \in A_j, \textit{ and} \ \forall j \in S_i^D \tag{3.13}$$

For this case, the following applies:

$$\sum_{m\in A_i,\, l\in A_j}\left(\frac{Y_m - Y_l}{\sqrt{2}\sigma_n}\right)^2 = \sum_{m\in A_i,\, l\in A_j}\left(\frac{C_{m,l} + N_m - N_l}{\sqrt{2}\sigma_n}\right)^2,\ \forall j \in S_i^D \tag{3.14}$$

The term on the r.h.s. of eqn.(3.11) has a Non-Central Chi-Square distribution with p^2 degrees of freedom and a Non-Centrality parameter λ for each pixel j in S_i^D:

$$\lambda(j) = \sum_{m\in A_i,\, l\in A_j}\left(\frac{C_{m,l}}{\sqrt{2}\sigma_n}\right)^2,\ \forall j \in S_i^D \tag{3.15}$$

Finally, the dissimilarity measure in S_i^D is characterized by the following first two moments (assuming N_i is *i.i.d* and $\kappa = p^{-2}$):

$$\forall j \in S_i^D:\quad \begin{array}{c} \mathrm{E}\left[\tilde{d}_i^U(j)\right] = 1 + \lambda(j)\,\kappa \\ \mathrm{Var}\left[\tilde{d}_i^U(j)\right] = 2\kappa + 4\lambda(j)\,\kappa^2 \end{array} \tag{3.16}$$

Therefore, the variance of the *normalized* dissimilarity measure vector of the entire search region is equal or larger than 2κ . It is important to note that we consider the $M^2 - 1$ elements of the dissimilarity vector as a realization of a random sequence of dissimilarity values in a $M \times M$ search region (excluding the POI whose corresponding dissimilarity is zero).

The search region can be divided into two sets on the basis of the aforementioned statistical analysis. Figure 3.3 illustrates the (theoretical) distribution of the two sets for a selected value of C. Setting the threshold which distinguishes between the two sets is not an easy task because practically $\lambda(j)$ is unknown. Hence, the distribution parameters of the S_i^D set are unknown. Consequently, we have decided to use a one-sided hypothesis which sets the threshold based on the characteristics of the S_i^S set. This threshold should be sufficiently robust, such that it will be able to cope with a potential cross-talk between the distributions of the two sets of pixels. A combined variance-mean threshold on the elements $\left\{\tilde{d}_i^U(j)\right\}_{j\in S_i}$, like $1 + 3\sqrt{2\kappa}$ (corresponding to three STDs from the mean, see eqn. (3.5)), can serve as

an example for such a threshold. However, this threshold turned out to be less reliable . Instead, setting a threshold that is based on the *accumulated variance* of the dissimilarity elements in S_i (refer to ALGORITHM I) turned out to be more robust. This threshold is set based on a sequential procedure that sorts the normalized dissimilarity elements in an ascending order and computes the variance of the sorted elements, starting with the two smallest elements and adding another element in each iteration based on the sorting order. The stopping criterion is set by comparing the computed variance, in each iteration, to the model variance threshold (2κ). In a similar manner to the accumulated variance threshold, an *accumulated mean* threshold was explored as well. However, experimental simulations have proven that this threshold is significantly less robust than the variance-based threshold and provides degraded denoising results.

Figure 3.4 provides a comparison between the three threshold suggestions for a given search region that is displayed in Figure 3.4 (a). Figure 3.4 (b) presents a dissimilarity histogram of an 11×11 search region extracted from a synthetic image that contains an edge. The dissimilarity measure was computed for all pixels in the defined initial search region S_i of size 11×11, centered at the POI. It is clear that the dissimilarity measure vector is characterized by at least two different populations. However, these two populations cross-talk and it is not an easy task to set the threshold which distinguishes between them. Defining a histogram-based threshold is not sufficiently robust since it depends on the number of bins assigned to the histogram. The model-based thresholds are marked on the figure, with the accumulated-variance threshold marked in red, the accumulated-mean threshold marked in black and the combined variance-mean threshold marked in green. By exploring the structure of the histogram, the red threshold appears to be more suitable than the two other thresholds. Figure 3.4 (c) presents the corresponding adaptive search regions $\left(S_i^S\right)$ that are the result of applying the different thresholds. The search regions are ordered from left to right, such that the region on the left is the result of the accumulated-variance threshold, the region in the center is the result of the accumulated-mean threshold, and the region on the right corresponds to

the combined variance-mean threshold. It is clear the accumulated-variance threshold (left region) provides the most accurate search region out of the three explored thresholds.

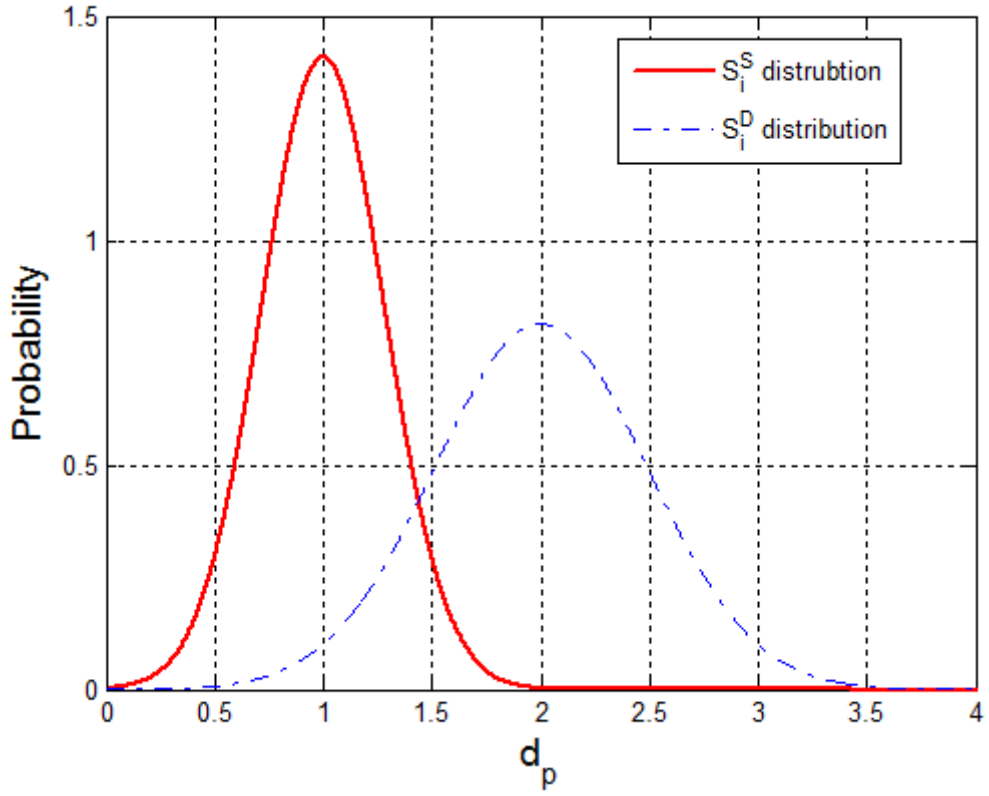
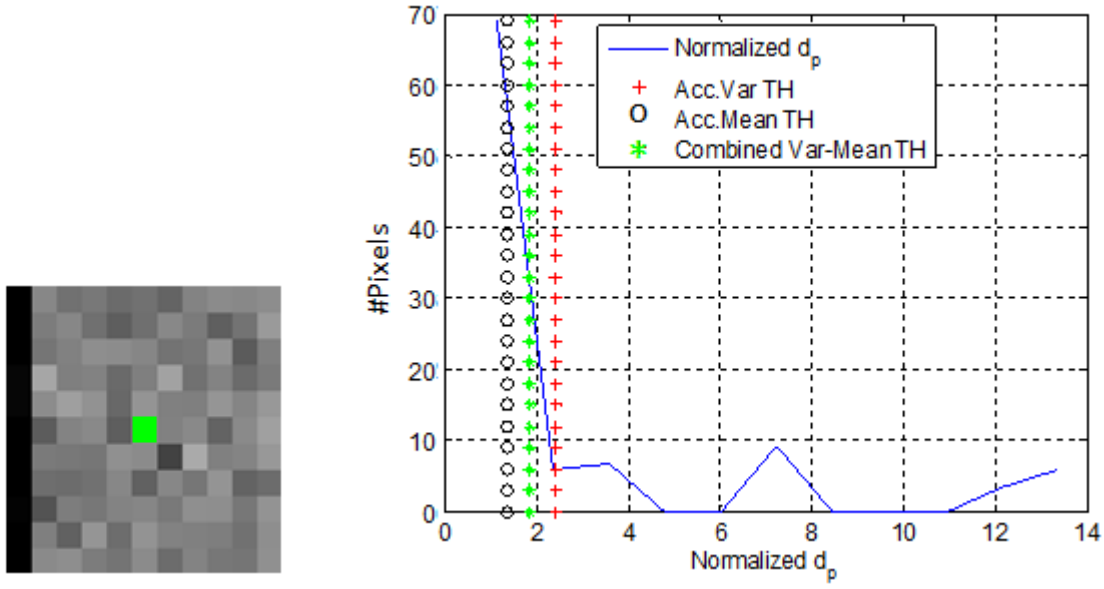
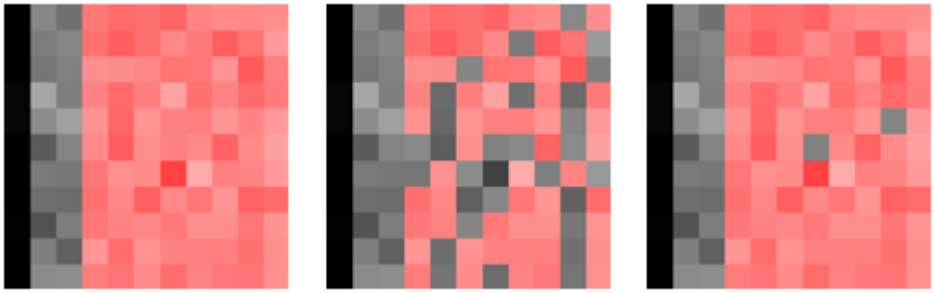


Figure 3.3.: An illustration of the (theoretical) distribution of the S_i^S (red) and the S_i^D (blue) sets, for a selected value of C and $p = 5$. S_i^D was assumed to be uniform, such that the gray level difference was set constant for the S_i^D sub-region.



(a) **(b)**



(c)

Figure 3.4.: A comparison between three suggested thresholds that satisfy the one-side hypothesis approach. (a) A selected search region from a noisy synthetic image ($\sigma_n = 20$) that contains an edge. The green pixel (in the center) corresponds to the POI. (b) A histogram of the normalized dissimilarity vector of the region in (a) (computation is based on $p = 5, M = 11$). The red mark corresponds to the accumulated-variance-based threshold, the black mark corresponds to the accumulated-mean-based threshold, and the green mark corresponds to a combined variance-mean-based threshold. (c) The adaptive search regions that correspond to the three explored thresholds. From left to right: accumulated-variance threshold, accumulated-mean threshold, and the combined variance-mean threshold.

The determination of the accumulated variance threshold is described in ALGORITHM I. This threshold is set based on an iterative procedure that sorts the $\tilde{d}_i^G(j)$ elements in an ascending order and computes the variance of the sorted elements, such that another element is added in each iteration based on the sorting order. The stopping criterion is set by comparing the computed variance, in each iteration, to the model variance threshold (2κ). Note that the values of $\tilde{d}_i^G(j)$ in a search region S_i are considered as a realization of the random sequence of M^2-1 values ($|S_i| = M^2$).

It is important to state that the statistical properties presented in eqn. (3.11) are adequate for random realizations of the normalized dissimilarity measure of the set of similar pixels, and do not represent the sorted dissimilarities. At the beginning of the accumulation process, when a small amount of dissimilarity elements are considered, the estimated mean of the explored elements is probably smaller than 1, since, in this stage, only the most similar pixels are considered. Similar observation is made with the estimated variance, that is smaller at the early accumulation stages and increases as pixels are added. However, as we approach the variance threshold, the accumulated pixels do satisfy the above-mentioned statistical characteristics.

ALGORITHM I: SEARCH REGION PIXEL CLASSIFICATION

1. For a given pixel i, calculate the normalized dissimilarity elements, denoted $\tilde{d}_i^G(j)\ \forall j \in S_i,\ j \neq i$ (see eqn. (3.1)).
2. Sort the normalized dissimilarity elements in an ascending order of values.
3. Compute an accumulated variance of the sorted elements by starting with the first two elements and adding one element at a time.
4. Stop accumulating elements when the computed variance exceeds $TH^G = 2\kappa$ (see eqns. (3.9)-(3.11)). The pixels associated with the accumulated dissimilarities define the S_i^S set.

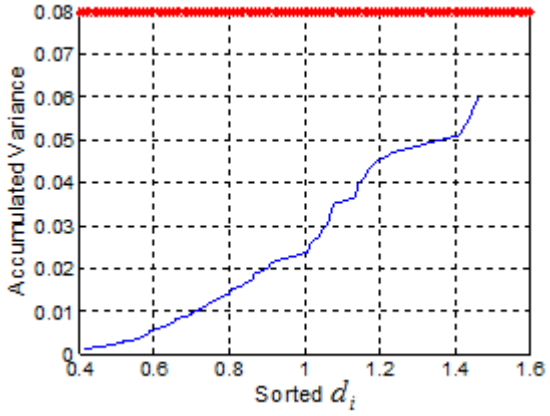
The dissimilarity value of the last pixel which participates in the accumulation process that defines S_i^S can serve as a threshold value, denoted TH_d , allowing to define S_i^S as follows:

$$S_i^S = \left\{ j \in S_i | \, \tilde{d}_i^G(j) \leq TH_d \right\} \tag{3.17}$$

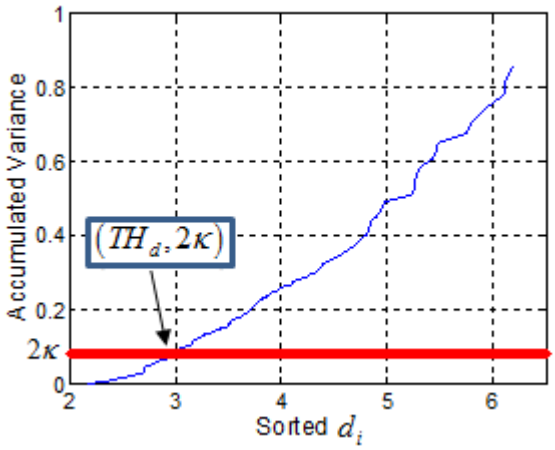
and hence the complementary set $S_i^D = \{S_i\} \setminus \{S_i^S\}$.

Refer to Figure 3.5 for an illustration of accumulated variance vectors, applied using the Uniform patch-kernel, obtained for a smooth and for a textured search region. Pay attention that for the smooth search region, all its respective pixels are included in its S_i^S since the accumulated variance of all these pixels is smaller than 2κ, whereas for the textured search region only a small fraction of its pixels is included. Hence, each pixel in the image is associated with its individual S_i^S set that defines which pixels will be included in the NLM averaging process. Moreover, each pixel is associated with a normalized cardinality value, $r_i = \frac{|S_i^S|}{M^2}$, which is defined as the number of pixels used to denoise it, normalized by the number of pixels in the original search region, and the image mask of normalized cardinality is denoted R. In this manner, smooth regions are characterized by a large value (closer to 1), whereas structural regions are associated with a smaller value (usually smaller than 0.5). A demonstration of such a matrix can be found in Figure 4.2 (b).

(a)



(b)



(c)

Figure 3.5.: Graphs of accumulated dissimilarity variance for noisy image Lena (256×256) where the dissimilarities are applied using the Uniform patch-kernel. (a) Lena with two marked search regions, the red one corresponds to a textured central patch and the dashed yellow corresponds to a smooth central patch, (b) Accumulated dissimilarity variance for the smooth patch, (c) Accumulated dissimilarity variance for the textured patch. The red line represents the 2κ value (for $p = 5$, $\kappa = 0.04$). All pixels in S_i whose dissimilarity is smaller or equal to TH_d (see ALGORITHM I) are associated with S_i^S.

3.4. Variance Threshold Validation

We wish to verify that the model-based variance threshold serves as an appropriate threshold, compared to an empirical one, in terms of PSNR. In Figure 3.6, we compare the denoising performance when using an estimated variance threshold to that of the model-based threshold, as suggested by ALGORITHM I. In our simulations, we explored different threshold values in the range $[0.5, 1.5] \cdot TH^G$, where $TH^G = 2\kappa$ (κ is set according to the Box patch-kernel, see eqn. (3.10)). The results were averaged over ten natural images given a specific noise level of $\sigma_n = 10, 20, 30$. The red filled circle on the solid blue curve corresponds to the denoising result using the model-based value, i.e., TH^G. It can be seen that the results that correspond to TH^G are located near the global maximum of the simulation curve for each explored noise level. This implies that the model-based variance threshold, used in ALGORITHM I, provides a good prediction of the empirical threshold value that results in a maximal PSNR, on average, in the given denoising scheme.

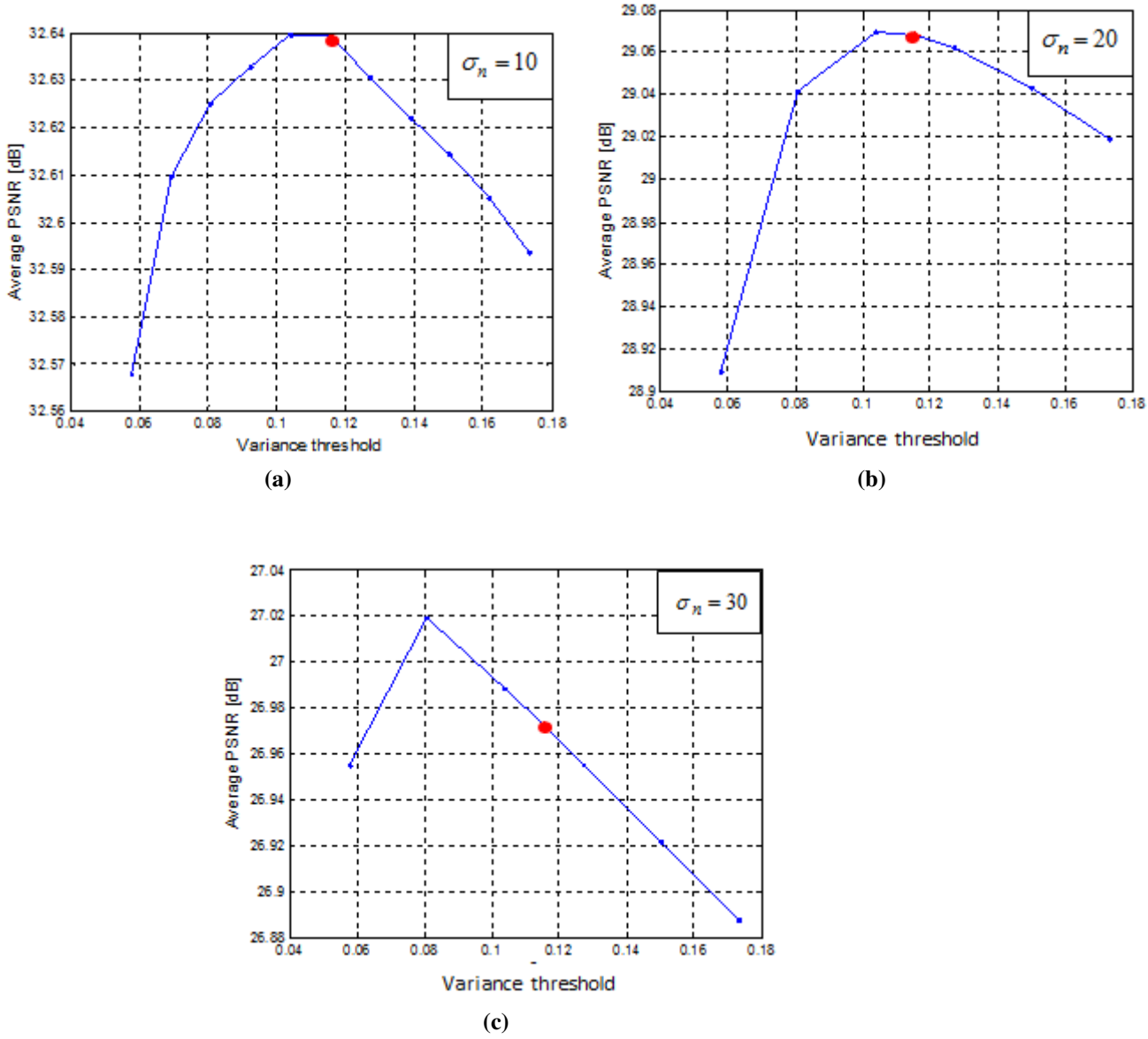


Figure 3.6.: Results of simulations that explore the model-based denoising results using various variance thresholds for images corrupted with various noise levels. The red filled circle on the solid blue curve corresponds to the denoising result using $TH^G = 2\kappa$, where κ corresponds to the Box patch-kernel. The following NLM parameters were used: $p = 5, M = 11, h = \sigma_n$. (a) $\sigma_n = 10$, (b) $\sigma_n = 20$, (c) $\sigma_n = 30$.

3.5. Variance Estimation Error

The accumulated variance of the normalized dissimilarity elements in a given search region is computed for the sorted elements, by starting with the variance of the two smallest dissimilarity values and adding one element at a time for the variance computation, as indicated in ALGORITHM I. As mentioned in the previous section, the discussed statistical properties presented in eqn. (3.11) are adequate for random realizations of the normalized dissimilarity measure of the set of similar pixels, and do not represent the sorted dissimilarities. Consequently, we refer to the point where the variance threshold is reached, then the dissimilarities of the accumulated pixels do satisfy the above-mentioned statistical characteristics.

Textured regions or regions that contain an edge are characterized by a small cardinality of the S_i^S set. Consequently, their respective final accumulated variance is computed for a relatively small number of considered elements, which makes the variance estimation error larger.

LEMA 3.2 [1]: *Let $Z_1,...,Z_L$ be a random sample that is distributed $\mathcal{N}\left(\mu,\sigma^2\right)$ with unknown mean and variance. The Sample Mean and Sample Variance are defined respectively as follows: $\overline{Z}=\frac{1}{L}\sum_{l=1}^{L}Z_l$, $\hat{V}=\frac{1}{L-1}\sum_{l=1}^{L}\left(Z_l-\overline{Z}\right)^2$. $\hat{V}$ is the Uniformly Minimum Variance Unbiased (UMVU) Estimator for σ^2 and has a Chi-Square distribution, such that*

$$\frac{(L-1)\hat{V}}{\sigma^2}\sim\chi^2_{L-1} \tag{3.18}$$

Since the Chi-Square distribution with $L-1$ degrees of freedom has a variance of $2(L-1)$ the standard error of estimating σ^2 by V^2 is:

$$\sigma_{\hat{V}}=\sigma^2\sqrt{\frac{2}{L-1}} \tag{3.19}$$

In our case, the normalized dissimilarity measure for a General patch-kernel has a *Normal* distribution with $\mu=1$, $\sigma^2=2\kappa$ (see eqn. (3.11)), as we approach the variance threshold

(see ALGORITHM I). Consequently, the variance estimation error, under this condition, is a function of $\left|S_i^S\right|$, that is:

$$\sigma_{\hat{V}} = 2\kappa\sqrt{\frac{2}{\left|S_i^S\right| - 1}} \tag{3.20}$$

Figure 3.6 presents the accumulated variance of sorted normalized dissimilarity elements for a pixel located on an edge (from the image Lena). It can be easily seen that the error of *UMVU* variance estimator is decreased as the number of the considered dissimilarity elements is increased, making the accumulated variance estimation more robust.

The variance STD presented in eqn.(3.20), or a specific factor of it can be added to the variance threshold during the accumulation process, such that it becomes:

$$TH^G = 2\kappa\left(1 + f\sqrt{\frac{2}{L-1}}\right) \tag{3.21}$$

f is a multiplication factor of the STD of the variance estimation error and κ is set according to the applied patch-kernel (see eqn. (3.10)). Simulations, presented in Figure 3.8, suggest that the optimal average PSNR is achieved for $f = 1$. However, this optimal f value has a minor contribution to the average PSNR (about 0.015 dB) with respect to using $f = 0$. When exploring the individual contribution of the utilization of $f = 1$ to the PSNR of specific images, we discovered that there is an increase that lies in the range of 0.08-0.14 dB. Since we wish to find a global f value that is adequate for different types of images, we suggest to continue here using a threshold value of 2κ (i.e., $f = 0$), and by that to preserve the parameter independency of the suggested approach. Figure 3.9 presents a typical histogram of the normalized cardinality of the adaptive search region, averaged over ten natural images. It can be seen that above 30% of the pixels are characterized with a relatively large S_i^S (most of the initial search region), hence associated with a small error term, which makes the selection of f insignificant for these pixels. The other fraction of the pixels are characterized

with a smaller S_i^S, hence associated with a larger value of variance computation error, such that a value of f that is larger than zero may have a more significant contribution to the corresponding variance threshold. However, as was explained above, the contribution of a value of f in the explored range (0-2) to the final PSNR is minor.

In sections 5.4 and 6.1, where consideration is given to the correlation between normalized dissimilarity elements, the variance threshold is modified, and a different value of f is used.

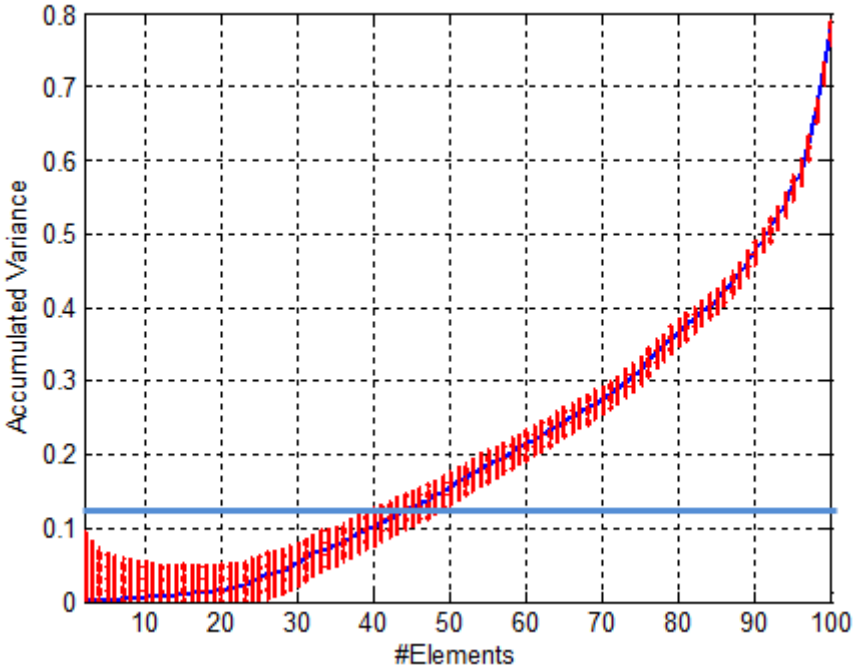


Figure 3.7.: Accumulated variance of sorted normalized dissimilarity elements for a pixel located on an edge (from the image Lena with $\sigma_n = 20,\ p = 5,\ M = 11$), computed using the Box patch-kernel. The error bars are added in red on top of the plot, such that the accumulated variance with the corresponding error STD is constrained to be positive. It can be seen that the error standard deviation decreases as the number of considered elements increases. The blue horizontal line corresponds to 2κ (for the Box patch-kernel).

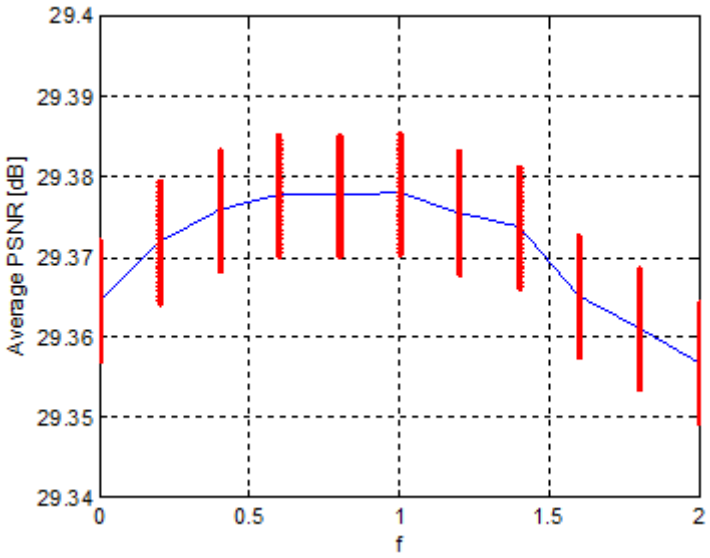


Figure 3.8.: Exploring the effect of adding a multiplication factor (f) of the variance error STD to the variance threshold. The curve represents averaged denoising results over ten images with different f values ($[0,3]$ with a stride of 0.2) and different noise levels ($[10,30]$ with a stride of 5). The red vertical lines correspond to variations due to different noise levels.

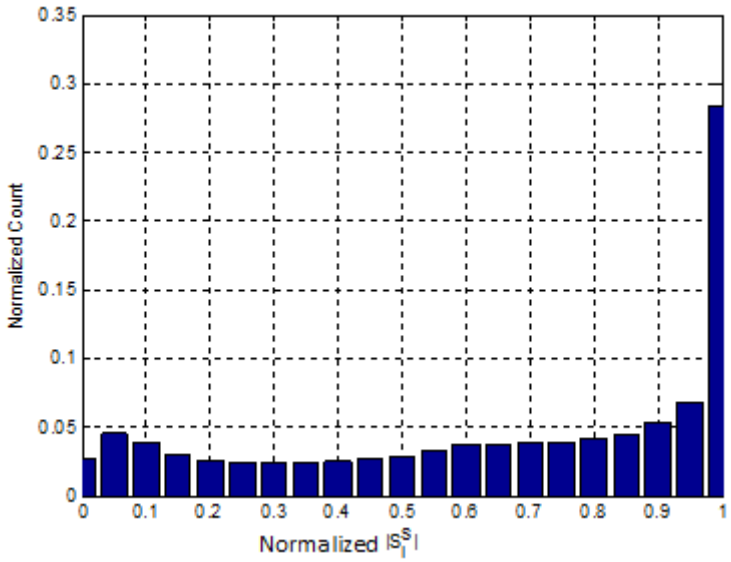


Figure 3.9.: Histogram of the normalized cardinality of the adaptive search region, accumulated over ten natural images corrupted with noise of $\sigma_n = 20$. NLM parameters: $f = 0$, $p = 5$, $M = 11$, $h = \sigma_n$.

3.6. Experimental Results

Two types of performance evaluation were conducted on natural images: an objective evaluation using the common measures of PSNR and SSIM [41] and a visual evaluation based on the perceived quality by a human observer. The following sub-sections describe the respective evaluations between the proposed adaptive search region scheme vs. the standard NLM, NLM with truncated weights and the LPA-ICI [37] method. Moreover, we have conducted a sanity check that verifies that the model-based variance threshold is indeed the threshold that provides the maximal PSNR, given the proposed adaptive NLM approach.

3.6.1. Comparison Between The Proposed Adaptive Search Region Scheme and Standard NLM

To evaluate the performance of our method, we have used several natural images corrupted by synthetic Gaussian noise (with $\sigma_n = 20, 30$). We compared the adaptive search region approach applied with either the Uniform or the Box patch-kernels to the standard NLM algorithm applied with the respective kernels. The same parameters were used for all the examined methods, i.e., a similarity patch of size 5×5 $(p = 5)$, a search region of size 11×11 $(M = 11)$ and $h = \sigma_n$. An objective evaluation that uses the common measures PSNR and SSIM [41] was conducted. Tables 3.1, 3.2 summarize the quantitative denoising results (objective evaluation) for different images with different noise levels. From their analysis, we can conclude that the proposed approach obtains somewhat higher PSNR and SSIM values than the standard NLM algorithm applied using either the Box or the Uniform patch-kernels. This tendency is preserved both for textural images (e.g., Baboon) and for smoother images (e.g., Lena). A visual evaluation based on the perceived quality by a human observer is presented in Figures 3.10, 3.11. The figures present a zoom-in view of the images Baboon and Lena and compare the two denoising NLM schemes, adaptive search region and standard, applied using both the Uniform and the Box patch-kernels. Figures 3.10 (b), (c) that correspond to the adaptive approach with Uniform and Box patch-kernels, respectively, are sharper than Figures 3.10 (d), (e) (refer to the eyes and fur) that corresponds to the Standard scheme with the same kernels. In Figure 3.11, the feathers and Lena's eyes appear sharper in the adaptive approach images (3.11 (b)-(c)) compared to their corresponding standard NLM images (3.11 (d)-(e)). Moreover, the use of the Uniform patch-kernel in the adaptive approach preserves the smoothness of Lena's face, as in its respective standard NLM image.

Image	σ_n	Standard NLM		NLM with Adaptive Search Region	
		PSNR [dB]	SSIM	PSNR [dB]	SSIM
Lena	20	30.11	0.87	**30.44**	**0.88**
Lena	30	28.03	0.81	**28.34**	**0.82**
Barbara	20	29.11	0.87	**29.4**	**0.88**
Barbara	30	26.92	0.8	**27.23**	**0.81**
Baboon	20	24.78	0.69	**25.36**	**0.73**
Baboon	30	23.53	0.6	**23.9**	**0.64**

Table 3.1.: Quantitative comparison between the Standard NLM and the suggested adaptive NLM, both applied using the **Uniform** patch-kernel.

Image	σ_n	Standard NLM		NLM with Adaptive Search Region	
		PSNR [dB]	SSIM	PSNR [dB]	SSIM
Lena	20	30.27	0.86	**30.3**	**0.87**
Lena	30	28.03	0.78	**28.08**	**0.79**
Barbara	20	29.19	0.87	**29.25**	**0.87**
Barbara	30	26.94	0.79	**27.02**	**0.8**
Baboon	20	25.54	0.74	**25.73**	**0.76**
Baboon	30	24.04	0.65	**24.16**	**0.67**

Table 3.2.: Quantitative comparison between the Standard NLM and the suggested adaptive NLM, both applied using the **Box** patch-kernel.

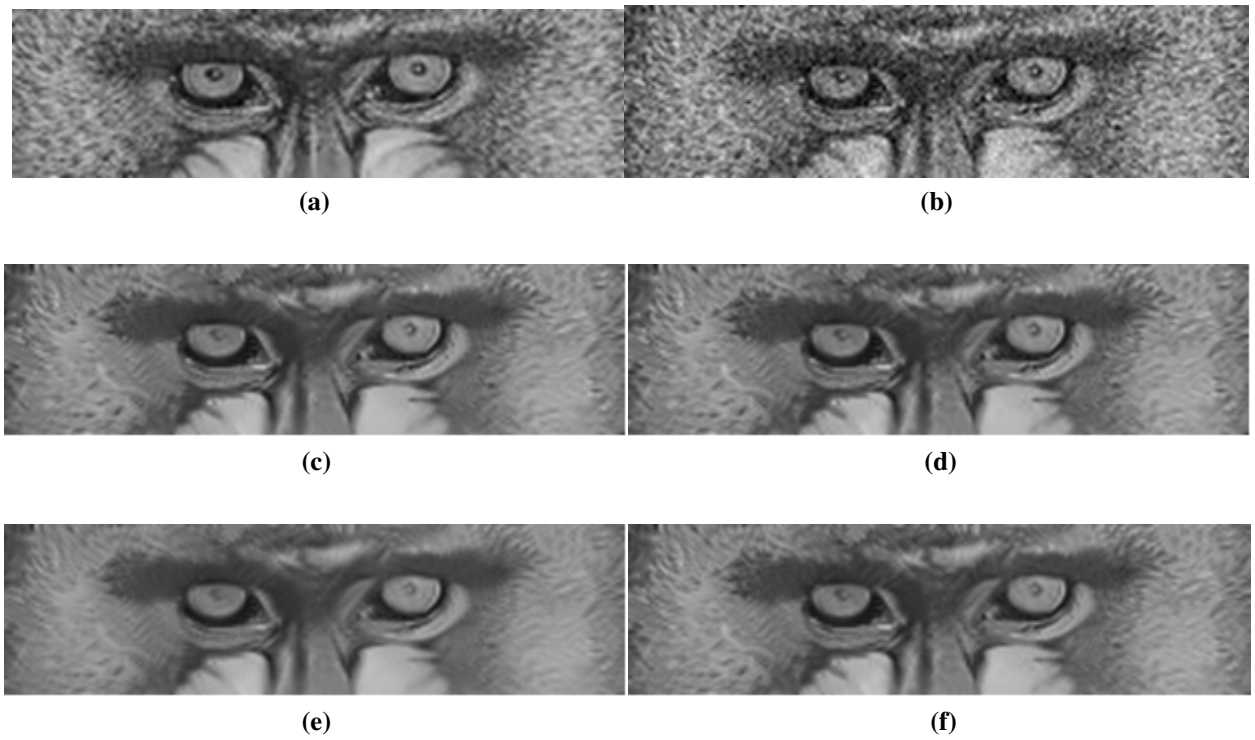


Figure 3.10.: Denoising variations of the image Baboon (204×204) with $\sigma_n = 20$, $p = 5$, $M = 11$, $h = \sigma_n$. A zoom-in view of the eyes and fur. (a) Cropped original image. (b) Cropped noisy image. (c) Denoised image using Adaptive scheme with a Uniform patch-kernel. (d) Denoised image using Adaptive scheme with a Box patch-kernel. (e) Denoised image using Standard NLM with a Uniform patch-kernel. (f) Denoised image using Standard NLM with a Box patch-kernel.

Figure 3.11.: Denoising variations of the image Lena (256×256) with $\sigma_n = 20,\ p = 5,\ M = 11,\ h = \sigma_n$. A zoom-in view of the eyes and hat feathers. (a) Cropped original image. (b) Cropped noisy image. (c) Denoised image using Adaptive scheme with a Uniform patch-kernel. (d) Denoised image using Adaptive scheme with a Box patch-kernel. (e) Denoised image using Standard NLM with a Uniform patch-kernel. (f) Denoised image using Standard NLM with a Box patch-kernel.

3.6.2. Comparison Between NLM Applied With The Suggested Adaptive Search Region and NLM with Truncated Weights

The straight-forward solution for using the most similar pixels in the denoising process is to use the pixels with the highest weights. In this manner, S_i^S will be constructed from the pixels that are associated with the most significant weights. This solution adds another degree of freedom that is the percentage of the highest weights that are considered. Tables 3.3 and 3.4 compare the adaptive search region scheme to NLM denoising that uses a fraction of the highest weights for the denoising process. Two fraction values were selected: 90% and 95% and explored for each patch-kernel (Uniform and Box). Table 3.3 presents the comparison for the Uniform patch-kernel and Table 3.4 presents the comparison for the Box patch-kernel. It can be seen that each patch-kernel has a different characterizing fraction that provides the best denoising results. The optimal fraction value for the Uniform and the Box patch-kernel denoising are 90% and 95%, respectively . NLM combined with the adaptive search region scheme provides better results, for any patch-kernel that is being used, in terms of both PSNR and SSIM than the manually selected fraction of most significant weights. The characterizing fraction may be image-dependent while the adaptive search region scheme is parameter-free and model-based.

Image	σ_n	100% weights PSNR [dB]/SSIM	95% weights PSNR [dB]/SSIM	90% weights PSNR [dB]/SSIM	Adaptive Search Region Scheme PSNR [dB]/SSIM
Lena	20	30.11/0.87	30.25/0.87	30.34/0.87	**30.44/0.88**
Lena	30	28.03/0.81	28.16/0.81	28.26/0.81	**28.34/0.82**
Baboon	20	24.78/0.67	24.97/0.7	25.12/0.72	**25.36/0.73**

Table 3.3.: Quantitative comparison between the Standard NLM applied with the Uniform patch-kernel, NLM with selection of a fraction of the most significant weights for each explored search region, and the suggested adaptive NLM.

Image	σ_n	100% weights PSNR [dB]/SSIM	95% weights PSNR [dB]/SSIM	90% weights PSNR [dB]/SSIM	Adaptive Search Region Scheme PSNR [dB]/SSIM
Lena	20	30.27/0.86	**30.33**/0.86	30.27/0.86	30.3/**0.88**
Lena	30	28.03/0.78	28.07/0.78	28.02/0.78	**28.08/0.82**
Baboon	20	25.54/0.67	25.7/**0.75**	25.36/0.73	**25.73/0.75**

Table 3.4.: Quantitative comparison between the Standard NLM applied with the Box patch-kernel, NLM with selection of a fraction of the most significant weights for each explored search region, and the suggested adaptive NLM.

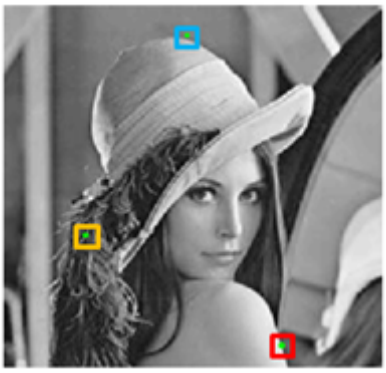
3.6.3. Comparison Between the Proposed Adaptive Search Region Scheme and LPA-ICI [37]

Section 3.2 provides a detailed overview of other adaptive NLM approaches that suggest to use an adaptive search region. This section will focus on the comparison between the Local Polynomial Approximation-Intersection of Confidence Intervals (LPA-ICI) approach [37] and our model-based approach. The LPA-ICI technique uses one-dimensional directional LPA kernels with pre-selected scales for eight directions. The optimal scale is set pixel-wise, based on the ICI approach, per each direction by computing the correlation between the POI and its corresponding neighborhood for the explored direction. The resultant search region is an octagon whose shape is based on the direction-wise spatial support. This method restricts the search region to be contiguous.

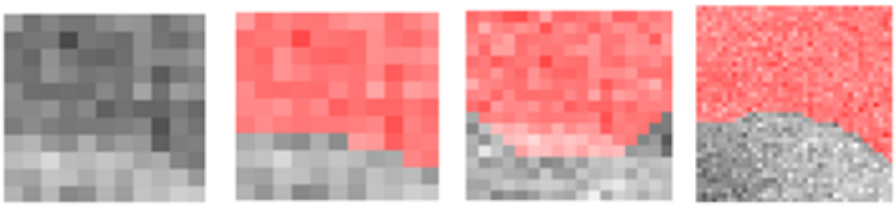
Figure 3.12 presents the resultant adaptive search region of different types of regions for both our model-based approach and the LPA-ICI approach [37]. For our approach, we used the following denoising parameters: $p = 5$, $M = 11$, $h = \sigma_n$, whereas the LPA-ICI [37] approach uses the same parameters values except for $M = 39$. For a fair comparison, we have also added our adaptive search region for $M = 39$. It can be clearly observed that for each explored region, our approach provides a more accurate adaptive search region, such

that only pixels with similar neighborhoods to that of the POI are included. An adequate search region assists in improving the denoising performance. The figure refers to three types of search regions:

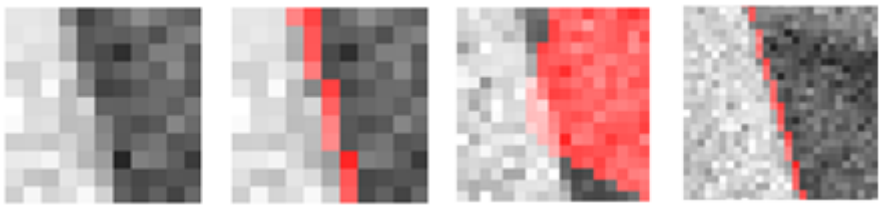
- Figure 3.12 (b) presents a search region that contains an edge, but the POI is distant from the edge center. The respective adaptive search region includes only pixels with similar neighborhoods to that of the POI, such that the search region contour lies along the edge. Notice that the resultant region of the LPA-ICI [37] approach is less accurate.
- Figure 3.12 (c) presents a search region that contains an edge. In this case the POI is located on the edge and the respective adaptive search region includes pixels located along the edge as well. The corresponding LPA-ICI [37] region includes pixels that relate to one side of the edge, but doesn't necessarily represent the edge itself.
- Figure 3.12 (d) presents a texture region. This example demonstrates one of the advantages of our approach, which does not restrict the search region to be contiguous, in contrast to the LPA-ICI method [37].



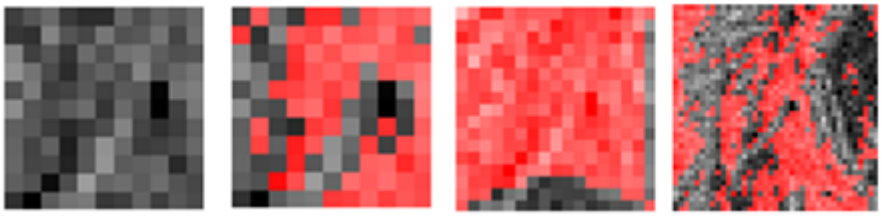
(a)



(b) Matches the orange marked region



(c) Matches the red marked region



(d) Matches the cyan marked region

Figure 3.12.: Adaptive search region based on our model-based method and the LPA-ICI [37] approach. (a) The image Lena with selected search regions. The green dot represents the POI for of each region, (b) A search region that contains an edge, however the POI is located far from the edge center, (c) A search region that contains an edge, and the POI is located on the edge, (d) A search region that contains texture. Figures (b)-(d) are ordered from left to right: initial search region, model-based search region for $M = 11$, LPA-ICI [37] search region for $M = 39$ and model-based search region for $M = 39$, marked in red.

3.7. Chapter Summary

In this chapter, we have introduced an Adaptive Model-Based NLM scheme. In this suggested scheme, only pixels whose neighborhood is similar to that of the POI are participating in the denoising process. The neighborhood similarity is based on the statistical properties of the dissimilarity measure. These properties classify the initial search region S_i into two sets; S_i^S that includes pixels with similar neighborhood to that of the POI, and S_i^D that includes pixels that are not included in the former set. This proposed approach is model-based and does not restrict the search region to be contiguous, an important quality for textural regions. By using an adaptive search region, only pixels whose defined neighborhood is similar to the neighborhood of the POI are selected to participate in the denoising process of the respective POI, and dissimilar pixels are avoided. In this manner, denoising is improved. Our suggested scheme results in a a more accurate adaptive search region that other common approaches. Moreover, it provides better denoising results than the standard NLM applied using either the Box or the Uniform patch-kernels.

4. Patch Kernel-Type Adaptation

NLM is commonly used with only one type of patch kernel for the dissimilarity measure computation (see eqn. (2.6)). In our analysis, two patch kernels were explored: the *Uniform patch kernel*, which assigns same weights to all the patch pixels, and a *Box patch kernel*, which assigns lower weights for distant pixels with respect to the patch center (see Figure 2.1).

Simulations suggest that the Uniform patch kernel is more adequate for smooth regions, whereas the Box patch kernel is more adequate for texture or edges. This conclusion is demonstrated in Figure 4.3 (b)-(c). The figure presents the denoising results of the image Lena with additive Gaussian noise of $\sigma_n = 20$. Figure 4.3 (b), which shows the results of standard NLM denoising using a Box patch kernel, preserves structure but is characterized by granularity in smooth regions. Figure 4.3 (c), which presents the denoising results of the standard NLM with a Uniform patch kernel, reduces the granularity effect, but over-smoothes texture and edges. Consequently, a combination of these two kernels, based on local structure, is suggested.

4.1. General Scheme

The information embedded in the matrix R , whose elements are $r_i = \frac{|S_i^S|}{M^2}$ (see section 3.3), can be used to estimate local structure, such that high r_i values correspond to smoother

regions, whose S_i^S set is larger. On the other hand, low r_i values correspond to edges or textured region characterized by a small set of S_i^S. For the sake of simplicity, i represents a running index, as if the matrix R is column-stacked.

We suggest to integrate the information embedded in the matrix R in order to assign to each pixel an adequate patch-kernel that is based on local structure. The suggested scheme, which is described in ALGORITHM II, requires the determination of two sets of similar pixels to pixel i: S_i^{SU}, S_i^{SB}. These sets are determined (based on ALGORITHM I) by using the Uniform and the Box patch kernels, respectively. As a result, two normalized cardinality matrices R^U, R^B are created for the noisy image. Their corresponding elements are denoted r_i^U, r_i^B. The data in each of these matrices is clustered, using K-Means with K=2, into two classes with two corresponding centroids. The cluster, denoted C_{max} , is associated with the larger centroid and represents pixels with a larger r_i value than the set of pixels, denoted C_{min} , associated with the smaller centroid. Hence, the cluster C_{max} contains pixels that typically relate to a smooth search region, whereas the cluster C_{min} contains pixels that are typically considered structural. We distinguish between the cluster C_{max} associated with the Uniform and the Box patch-kernels, denoted C_{max}^U, C_{max}^B respectively, as well as the cluster C_{min} associated with the Uniform and the Box patch-kernels, denoted C_{min}^U, C_{min}^B respectively.

ALGORITHM II: PATCH KERNEL-TYPE ADAPTATION

1. For each pixel i in the noisy image, do the following:

   a) Compute the normalized dissimilarity elements using the Uniform patch-kernel, denoted $\tilde{d}_i^U(j)$, $\forall j \in S_i$, $j \neq i$ (see eqns. (3.1)).

   b) Compute the variance of all the elements in the computed set of normalized dissimilarities. If the variance of all the computed elements is smaller or equal to $TH^U = 2\kappa$ (see eqns. (3.10),(3.11)), then pixel i is considered "smooth" under this test. Hence, $S_i^{SU} = S_i$, $r_i^U = 1$.

   c) Otherwise, perform the following:

      i. Compute the normalized dissimilarity elements using the Box patch-kernel, denoted $\tilde{d}_i^B(j)$, $\forall j \in S_i$, $j \neq i$.

      ii. Compute the variance of all the elements in this set. If their variance is smaller or equal to $TH^B = 2\kappa$ (see eqns. (3.10),(3.11)), then pixel i is considered "smooth" under this test. Hence, $S_i^{SB} = S_i$, $r_i^B = 1$.

      iii. Otherwise, compute the sets S_i^{SU}, S_i^{SB} using ALGORITHM I and the respective kernels. Then, compute the corresponding r_i^U, r_i^B to create the respective matrices R^U, R^B.

2. Cluster the data in each R^U, R^B into two clusters using K-Means with K=2, such that each matrix is divided into a set of "smooth" pixels and a set of structured pixels. The two cluster centroids of each matrix are compared to 0.5, and the following rule is applied for each pair of centroids:

   a) If the values of the two centroids of a given matrix are both larger than 0.5, then the clusters of the corresponding matrix are joined, associating all the pixels to the set of "smooth" pixels and the respective "structured" set becomes empty.

   b) If the values of the two centroids of a given matrix are both smaller than 0.5, then the clusters of the corresponding matrix are joined, associating all the pixels to the set of "structured" pixels and the respective "smooth" set becomes empty.

   c) Otherwise, clusters are not joined, and each corresponding matrix has its own pair of two clusters.

3. For each pixel i , set pixel weights (before normalization) according to the following rule:

$$\forall j \in S_i^S,\ j \neq i:\ w_{i,j} = \begin{cases} exp\left(-2\tilde{d}_i^U(j)\right) & r_i^U \in C_{max}^U \text{ and } r_i^B \in C_{max}^B \\ exp\left(-2\tilde{d}_i^B(j)\right) & Otherwise \end{cases}, \tag{4.1}$$

   where C_{max}^U, C_{max}^B are the sets associated with the centroids of the larger cluster in R_U, R_B matrices, respectively. The POI is assigned a weight that corresponds to the maximal weight given to any of the pixels included in S_i^S (see sub-section 2.5.4).

4. Normalize the computed weights by their sum $W_i = \sum_{j \in S_i^S} w_{i,j}$.

4.2. Simplified Adaptation Scheme

Since the computation of two normalized cardinality matrices is time consuming, we suggest a simplified scheme that is described in ALGORITHM III.

ALGORITHM III: SIMPLIFIED ADAPTATION SCHEME FOR PATCH KERNEL-TYPE ADAPTATION

1. For each pixel i in the noisy image, do the following:

   a) Compute the variance of *all* the elements $\tilde{d}_i^U(j)$, $\forall j \in S_i$, $j \neq i$ (see eqn. (3.1)). If the resultant variance is smaller or equal to $TH^U = 2\kappa$ (see eqns. (3.10), (3.11)), then pixel i is considered "smooth" and $S_i^S = S_i$, $r_i = 1$.

   b) Otherwise, compute the set S_i^S using ALGORITHM I and the Uniform patch-kernel. Then, compute the corresponding r_i to create the respective matrix R.

2. Cluster the data in the normalized cardinality matrix R , whose elements are r_i, into two clusters $\left(C_{max}^U, C_{min}^U\right)$ using K-Means with K=2, such that the cluster C_{max}^U having the higher centroid value represents the "smooth" pixels, and the cluster C_{min}^U respresents the "structured" pixels. The two cluster centroids are compared to 0.5, and the following rule is applied:

   a) If the two centroids values are both larger than 0.5, then the clusters are joined, associating all the pixels to the set of "smooth" pixels and the respective "structured" set becomes empty.

   b) If the two centroids values are both smaller than 0.5, then the clusters are joined, associating all the pixels to the set of "structured" pixels and the respective "smooth" set becomes empty.

   c) Otherwise, the two clusters are not empty, and the matrix R has a pair of clusters.

d) Compute the dissimilarity elements $\tilde{d}_i^B(j)$, $\forall j \in S_i$ for all the pixels i whose $r_i \in C_{min}^U$.

3. Set pixel weights (before normalization) according to the following rule:

$$\forall j \in S_i^S,\ j \neq i:\ w_{i,j} = \begin{cases} exp\left(-2\tilde{d}_i^U(j)\right) & r_i \in C_{max}^U \\ exp\left(-2\tilde{d}_i^B(j)\right) & Otherwise \end{cases}, \tag{4.2}$$

The POI is assigned a weight that corresponds to the maximal weight given to any of the pixels included in S_i^S (see sub-section 2.5.4).

4. Normalize the computed weights by their sum $W_i = \sum_{j \in S_i^S} w_{i,j}$.

This scheme may result in an inconsistent pixel classification into Smooth or Non-Smooth classes. On average, less than 5% of the pixels in the entire image are affected by this inconsistency. It stems from the fact that there are pixels that are considered structural according to the Uniform patch-kernel scheme, and considered "smooth" according to the Box patch-kernel scheme. Since the fraction of such pixels is relatively small, the two schemes provide very similar results. Hence, the less time consuming scheme in ALGORITHM III is recommended. The computational complexity, with respect to the standard NLM, is of course reduced when the simplified scheme of ALGORITHM III is being used, since Box-kernel based dissimilarities are computed for a smaller fraction of pixels in the image. In this manner, ALGORITHM III is characterized by an average increase of 14% of running time with respect to the standard NLM (applied using either the Uniform or the Box patch-kernel).

Figure 4.1 presents the clustered R matrix for the images Lena and Baboon. The red cluster corresponds to smooth pixels based on the Uniform kernel, i.e., $i \in C_{max}^U$, the blue cluster corresponds to pixels that are considered structural according to the two kernels, i.e., $C_{min}^U \cap C_{min}^B$, and the green cluster corresponds to the pixels that are considered structural according to the Uniform kernel but not according to the Box kernel, i.e., $i \in C_{min}^U \cap C_{max}^B$. It can be seen that for both Lena, whose majority of pixels are considered smooth, and Baboon, which is

characterized by a relatively large amount of structural pixels, the pixels associated with the green cluster are a clear minority (less than 5%). Moreover, there is no significant difference in performance in terms of PSNR and SSIM between the two methods (on average 0.01 dB and 0.001 SSIM difference).

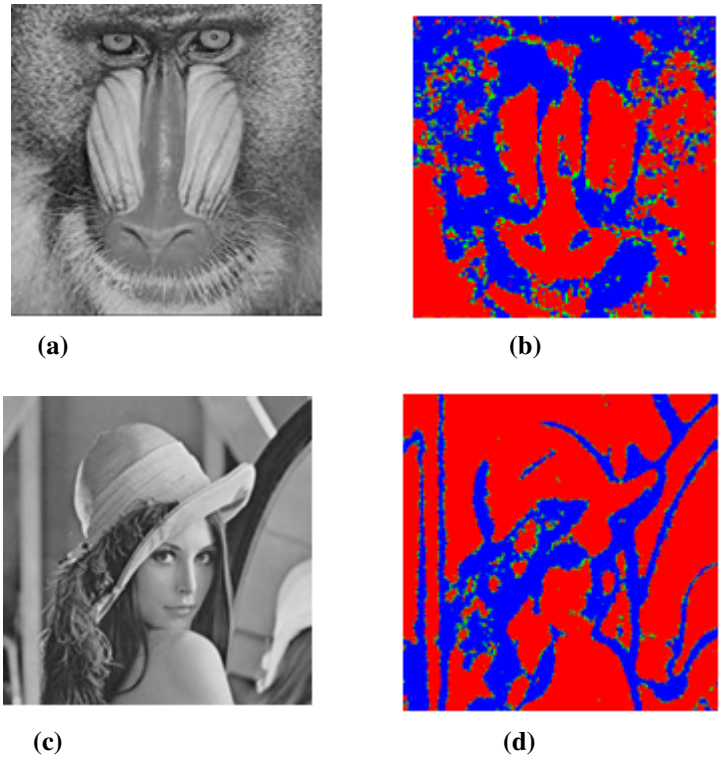
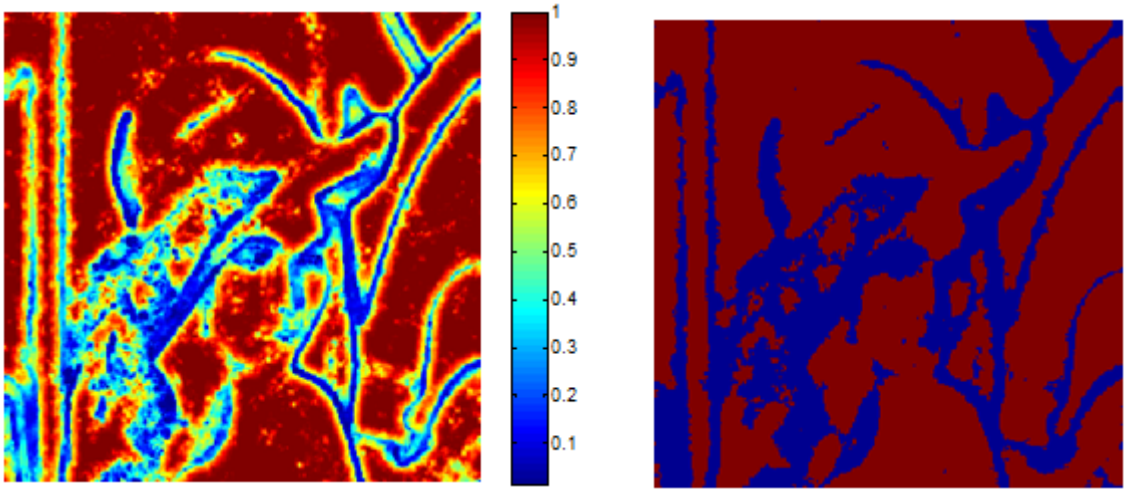


Figure 4.1.: Motivation behind the simplified scheme. (a) Baboon (204×204), (b) The clustered R matrix for the image in (a), (c) Lena (256×256), (d) The clustered R matrix for the image in (c). The R matrices were created for images corrupted with additive Gaussian noise of $\sigma_n = 20$, based on dissimilarity measures computed using $p = 5$, $M = 11$. The red cluster that corresponds to smooth regions is a result of the pixels $i \in C^U_{max}$. The blue cluster that corresponds to structural regions is a result of the pixels $i \in C^U_{min} \cap C^B_{min}$. The green cluster represents less than 5% of the pixels in both images and corresponds to the pixels $i \in C^U_{min} \cap C^B_{max}$.

Figure 4.2 (b) displays the normalized cardinality matrix R of the image Lena denoised using the simplified suggested scheme (i.e., ALGORITHM III). Figure 4.2 (c) displays the

classified matrix, such that red pixels correspond to smooth regions whose NLM weights are computed based on the Uniform patch kernel, whereas the blue pixels correspond to edges or textured regions whose respective NLM weights are computed using the Box patch kernel.

(a)



(b) **(c)**

Figure 4.2.: (a) Lena (256×256), (b) Normalized cardinality matrix (R) for noisy image with $\sigma_n = 20$, (c) Clustered elements of R using K-Means, with K=2. Red pixels correspond to smooth regions.

From now on, when we refer to the adaptive NLM scheme, we relate to the scheme that combines the adaptive search region and the adaptive patch-kernel altogether.

4.3. Experimental Results

Two types of performance evaluation were conducted on natural images: an objective evaluation using the common measures of PSNR and SSIM [41] and a visual evaluation based on the perceived quality by a human observer. The following sub-sections describe the respective evaluations between the two suggested schemes (general vs. simplified) and the simplified scheme vs. the standard NLM.

4.3.1. Comparison Between the General and the Simplified Kernel Adaptation Schemes

To verify that the Simplified scheme (ALGORITHM III) is a robust efficient approximation of the General kernel adaptation scheme (ALGORITHM II), the two schemes were compared experimentally. For this comparison, several natural images corrupted by synthetic Gaussian noise (with $\sigma_n = 20, 30$) were used with the same NLM parameters, i.e., a similarity patch of size 5×5 ($p = 5$), a search region of size 11×11 ($M = 11$) and $h = \sigma_n$.

Table 4.1 summarizes the quantitative denoising results (objective evaluation) for different images with different noise levels. It can be seen that the utilization of the Simplified scheme for the denoising process provides similar results when compared to the results of the General scheme. Besides the quantitative resemblance, the denoised images of both schemes are quite similar visually. Consequently, this simulation concludes that the Simplified scheme is a robust alternative to the General scheme. From now on, when we mention the kernel-type adaptation scheme, we refer to the simplified version. The combined adaptive scheme

refers to the combination of an adaptive search region with an adaptive kernel-type, that is, ALGORITHM III.

σ_n	20				30			
Image	General Scheme		Simplified Scheme		General Scheme		Simplified Scheme	
	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
Lena	30.48	0.878	30.49	0.879	28.31	0.815	28.32	0.815
Barbara	29.31	0.874	29.33	0.875	27.15	0.811	27.16	0.811
Peppers	30.54	0.888	30.55	0.888	28.38	0.836	28.39	0.836
Baboon	25.6	0.745	25.62	0.748	23.97	0.64	23.89	0.634

Table 4.1.: Quantitative comparison between the General kernel-type adaptation suggested scheme (ALGORITHM II) and the its Simplified version (ALGORITHM III).

4.3.2. Comparison Between the Combined Adaptive Scheme and Standard Non-Local Means

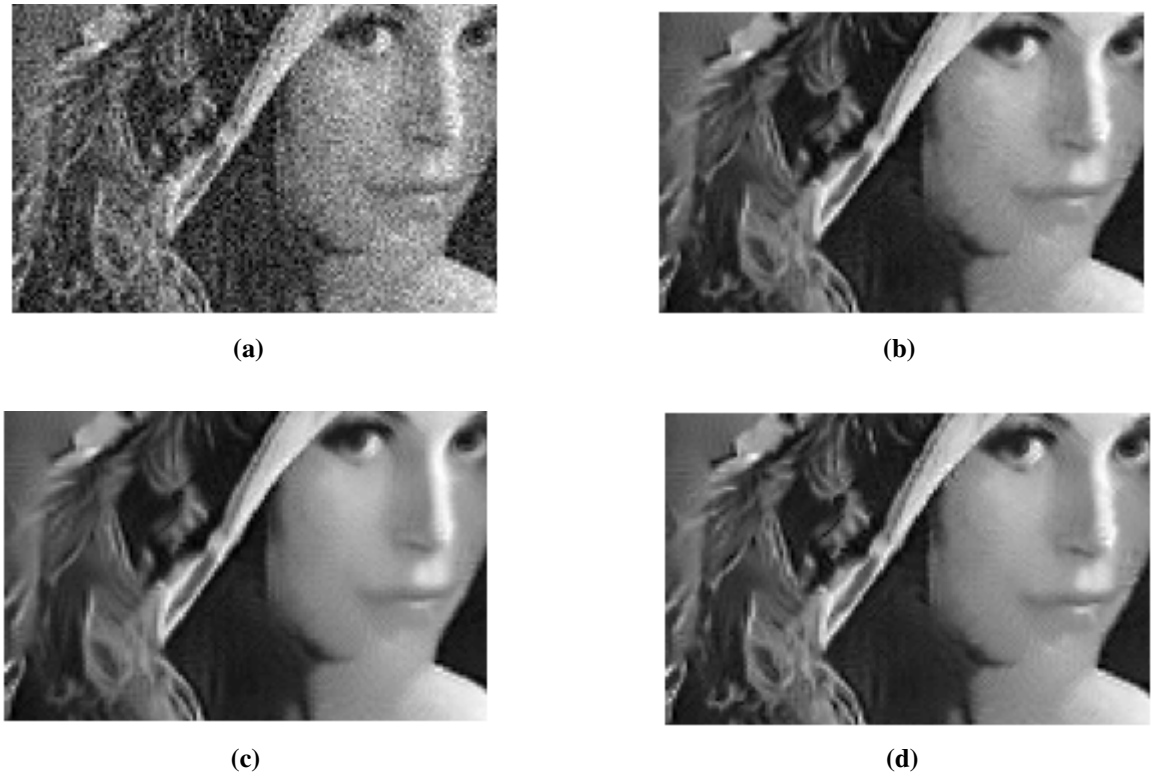
To evaluate the added value of the combined adaptive approach (adaptive search region and adaptive patch-kernel) to the NLM performance, we have used several natural images corrupted by synthetic Gaussian noise (with $\sigma_n = 20, 30$). We compared our approach to the standard NLM algorithm applied with both the Box and the Uniform patch-kernels. The same parameters were used for all the examined methods, i.e., a similarity patch of size 5x5 ($p = 5$), a search region of size 11x11 ($M = 11$) and $h = \sigma_n$. As was mentioned in section 3.5, both an objective and a visual evaluation were conducted. Table 4.2 summarizes the quantitative denoising results (objective evaluation) for different images with different noise levels. From its analysis, we can conclude that the proposed approach obtains somewhat higher PSNR and SSIM values than the conventional NLM algorithm for both the Box and the Uniform patch-kernels. This tendency is preserved both for textural images (e.g., Baboon) and for smoother images (e.g., Lena, Pepper). When using only an adaptive search

region combined with a fixed kernel (see Tables 3.1, 3.2), either the Uniform or the Box kernels, instead of an adaptive patch-kernel, the PSNR improvement is smaller. Average improvement based on three selected images with $\sigma_n = 20$ provides an increase of 0.07 dB and 0.38 dB for using NLM applied using the Box patch-kernel and NLM applied using the Uniform patch-kernel, respectively, compared to an average increase of 0.15 dB and 0.42 dB when adding the adaptive kernels.

A visual comparison is given in Figure 4.3, which compares the denoised images produced by the standard NLM (for both the Box and the Uniform patch-kernels) and our proposed Adaptive NLM, for the image Lena with $\sigma_n = 20$. Figure 4.3 (b), which represents a standard NLM denoising using a Box patch-kernel, preserves structure but is characterized by granularity in smooth regions. Figure 4.3 (c), which presents the denoising results of the standard NLM applied with Uniform patch-kernel, reduces the granularity effect, but oversmoothes texture and edges. On the other hand, Figure 4.3 (d), which presents the proposed method, is characterized by both preservation of structural information and reduced granularity in smooth regions. Moreover, since the proposed approach chooses only similar pixels from the search region, structures are sharper than in the standard NLM method even when it uses the Box patch-kernel. In the same manner, Figure 4.4 compares the denoising performance of the three explored approaches, NLM applied with Box patch-kernel, NLM applied with Uniform patch-kernel and our model-based approach, for the image Lena with different noise levels based on the PSNR measure. It can be clearly observed that the suggested model-based NLM method outperforms the standard NLM for any explored noise level. Although, the PSNR and SSIM differences are not that large, the visual comparison is highly significant. Appendix A provides additional examples of such a comparison.

Image	σ_n	NLM with Uniform patch-kernel		NLM with Box patch-kernel		NLM with Adaptive TH	
		PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
Lena	20	30.11	0.87	30.27	0.86	**30.48**	**0.88**
Lena	30	28.03	0.81	28.03	0.78	**28.32**	**0.82**
Barbara	20	29.11	0.87	29.19	0.87	**29.33**	**0.88**
Baboon	20	24.78	0.69	25.54	0.74	**25.62**	**0.75**
Pepper	20	30.28	0.88	30.39	0.87	**30.55**	**0.89**
Pepper	30	28.03	0.83	28.06	0.81	**28.39**	**0.84**

Table 4.2.: Quantitative comparison between the Standard NLM applied with both the Uniform and the Box patch-kernels and the suggested adaptive NLM.



(a) **(b)**

(c) **(d)**

Figure 4.3.: Denoising variations of the image Lena (256x256) with $\sigma_n = 20$, $p = 5$, $M = 11$, $h = \sigma_n$. A zoom-in view of the shoulder and face (flat regions) and of the feathers (texture). (a) Noisy image. (b) Denoised image using standard NLM with a Box patch-kernel. (c) Denoised image using standard NLM with a Uniform patch-kernel. (d) Denoised image using the Combined Adaptive Scheme (ALGORITHM III).

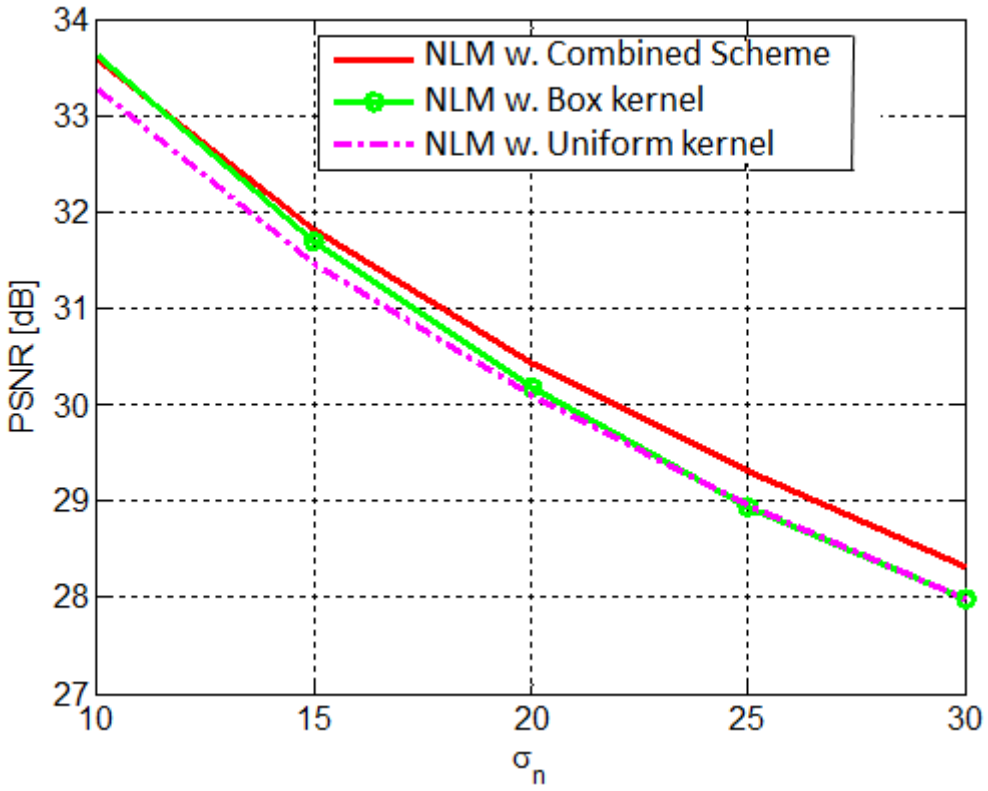


Figure 4.4.: A comparison between the three explored NLM approaches, i.e., standard NLM with either the Uniform patch-kernel or the Box patch-kernel, and our Adaptive approach, for the image Lena. The comparison is conducted for different noise levels with the following NLM parameters: $p = 5$, $M = 11$, $h = \sigma_n$.

4.3.3. Comparison Between The Adaptive Scheme (Algorithm III) and The Adaptive Search Region Size Approach [17]

Kervran et. al [17] present a somewhat different approach that uses an adaptive search region combined with the NLM denoising method. Instead of choosing pixels with similar neighborhood from a given initial search region of size $M \times M$, they select the most suitable pixel-wise search region size, out of five potential sizes $M = [1,3,5,9,17]$, in a manner that it balances the accuracy of approximation and the stochastic error, at each spatial position.

Their approach restricts the search region to be rectangular, thus contiguous, such that the optimal pixel-wise search region may include pixels whose neighborhood is different from that of the POI. This approach of optimal search region size selection (before applying NLM) is characterized by a computational complexity of $O\left(p^2 G \sum_{m\in[1,3,5,9,17]} M_m^2\right)$, where p represents the similarity patch size, and G refers to the number of pixels in the image. For comparison, our proposed approach for adaptive search region selection is characterized by a computational complexity of $O\left(p^2 G M^2\right)$. Hence, with $M = 11$ we obtain a significant reduction in the number of computations.

We wish to compare our approach to the approach proposed in [17], where both approaches are combined with the NLM denoising method. Since there is no available implementation of their respective algorithm, and its implementation by us is beyond the scope of the thesis, no visual comparison is conducted. Consequently, we compare the two approches in a quantitative manner, in terms of PSNR. Figure 4.5 presents a comparison with four selected images at different noise levels $\sigma_n = [10, 25]$, as reported in [17]. Since the paper results are reported for $p = 9$, we have implemented our approach with $M = 11$ and both $p = 5$ and $p = 9$. Our results represent the average PSNR over five noise realizations for each explored image. Two main conclusions can be derived from this comparison. The first one is the fact that the PSNR is improved for our approach when it is applied with $p = 9$, compared to $p = 5$. However, as was explianed in sub-section 2.5.1, increasing the similarity patch size causes, in some images, hallos around edges. The second conclusion refers to the fact the different images are characterized by different performance tendency, such that for the images Lena (a) and House (c), [17] presents better quantitative results, whereas for the images Barbara (b) and Peppers (d), our proposed apprach is quantitavely somewhat better. However, as mentioned before, the approach presented in [17] has a much higher computational complexity.

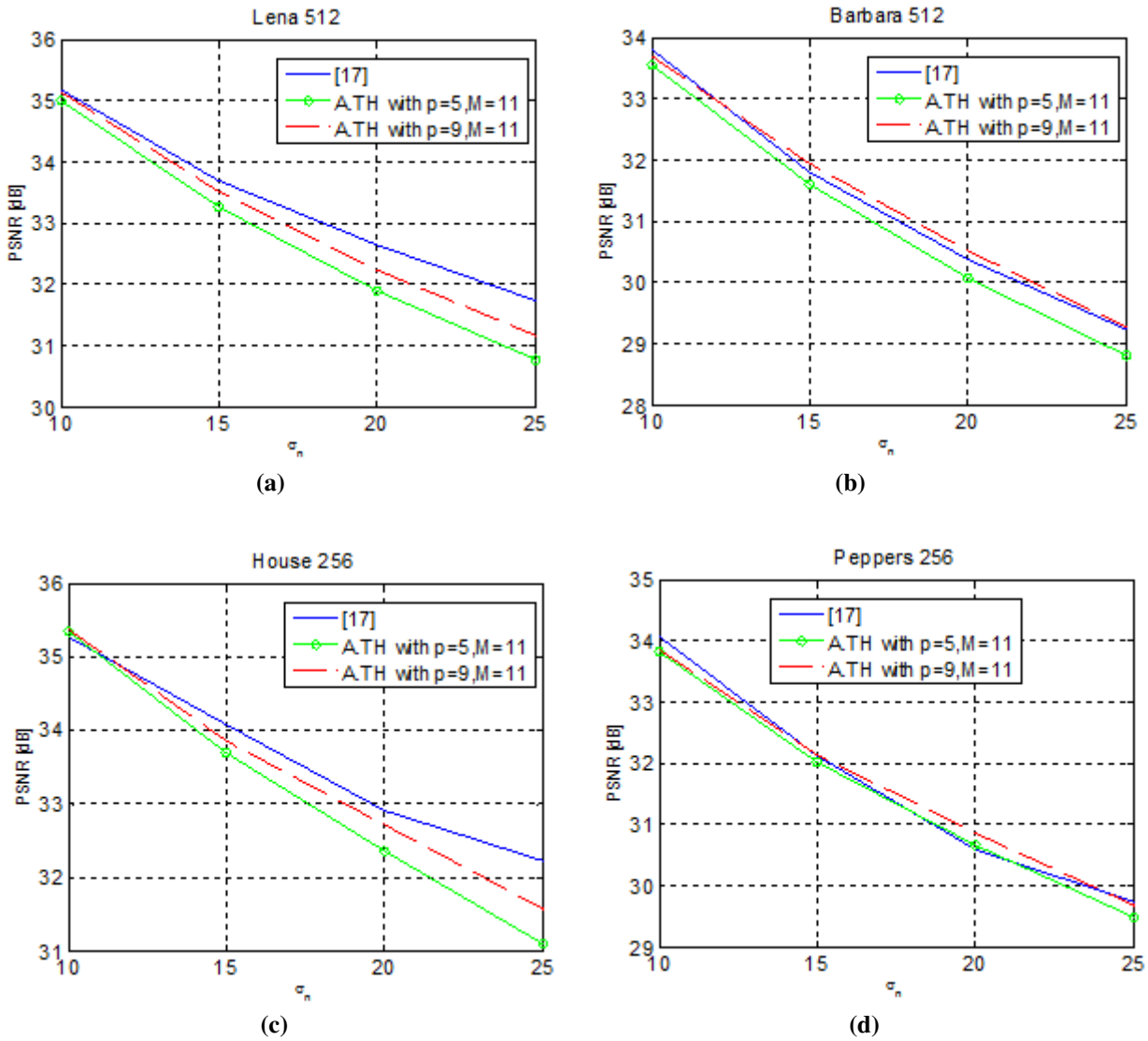


Figure 4.5.: Comparison between the NLM denoisng results when applied with our proposed adaptive search region approach vs. the adaptive search region size approach [17], for different noise levels $\sigma_n = [10,25]$. (a) Lena 512×512, (b) Barbara 512×512, (c) House 256×256, (d) Peppers 256×256. The blue line corresponds to the results of the approach presented in [17], the green circled line corresponds to our proposed approach applied with $p = 5, M = 11$, and the red dashed line corresponds to our proposed approach with $p = 9, M = 11$.

4.4. Chapter Summary

In this chapter, we have integrated the adaptive patch-kernel scheme into the adaptive search region scheme. Consequently, the cardinality of the S_i^S set, denoted $\left|S_i^S\right|$, is exploited to classify the pixels as "smooth" or "non-smooth", such that the pixels that are associated with the "smooth" cluster are characterized with a larger $\left|S_i^S\right|$ and will be denoised based on a dissimilarity measure computed using the Uniform kernel. The pixels that are associated with the other cluster are denoised based on a dissimilarity measure computed using the Box kernel. The combined approach was compared to the approach that uses an adaptive search region with a single patch-kernel type, as well as to the standard NLM and was found to provide better denoising results both quantitatively (PSNR and SSIM-wise) and qualitatively (visually). Moreover, we have compared our approach to the adaptive search region size approach [17] in terms of PSNR. The latter is characterized by a much higher computational complexity and its denoising results are comparable to ours.

5. Correlation Between Dissimilarities and Its Effect On The Model-Based Scheme

The statistical model in the previous chapter was developed under the simplifying assumption that the dissimilarity elements in a given search region S_i are not correlated. It is important to state that researchers, e.g., [31, 38], did not relate to any source of correlation between the dissimilarity elements and its effect on their statistical properties.

In this chapter we consider the correlation between normalized dissimilarities of patches in a given search region and its effect on the model-based scheme presented in Chapter 3. The reason that such a correlation exists is that the dissimilarities of all the patches in a given search region are computed with respect to the same reference patch. Furthermore, some patches may overlap each other and/or the mutual reference patch itself, contributing further to the correlation. To simplify the analysis, we tackle the issue by considering first the correlation due to the mutual reference patch (Case 1), assuming no patch overlaps. Then we add the effect of overlap between patches, but not with the reference patch (Case 2), and finally we address the most general case in which overlapping patches may also overlap the reference patch (Case 3). By arranging the dissimilarities in a vector form, we express the correlation between the vector elements via its covariance matrix and apply the results to derive the statistical properties of the empirical (estimated) variance used in the

proposed model-based denoising scheme.

5.1. Case 1: Correlation Between Dissimilarities of Patches That Do Not Overlap Each Other, Nor The Reference Patch

5.1.1. Case Definition

We discuss here the case of non-overlapping similarity patches, as illustrated in Figure 5.1. In this figure, the reference patch is denoted A_i and the selected compared patches are denoted A_j, A_k for $j,k \in S_i$. The compared patches satisfy the no-overlap criterion, that is: $\forall j \neq k,\ j,k \in S_i^S,\ A_j \cap A_k = \emptyset,\ A_j \cap A_i = \emptyset,\ A_k \cap A_i = \emptyset$. This means the these patches do not overlap each other, nor the reference patch.

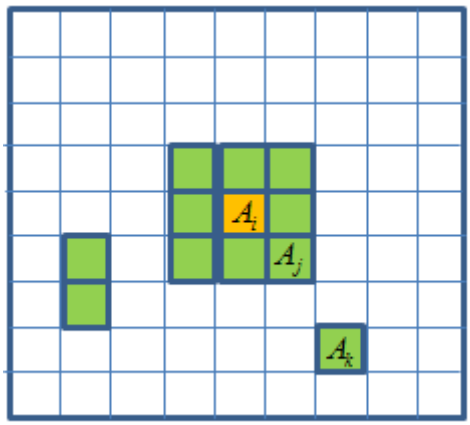


Figure 5.1.: Scheme of the grid that characterizes Case 1 where patches do not overlap each other, nor the reference patch. The reference patch A_i is colored in orange and the compared patches are colored in green (like A_j, A_k).

By definition, the normalized dissimilarity of a patch is computed with respect to the reference patch, thus the reference patch serves as a mutual member that adds a source of

correlation between dissimilarities of different compared patches. We will compute below the covariance matrix of the normalized dissimilarity elements arranged in a vector form. The distribution of each patch dissimilarity remains *Chi-Square* and can be approximated by a *Normal* distribution, as presented in eqn. (3.7). The off-diagonal elements of the respective covariance matrix refer to the cross-variance between the vectorized dissimilarity elements, and their elaborated analytic development for General and Uniform patch-kernels can be found in Appendix B.1. Here, we present the notation and the final forms of the respective covariance matrices.

Notation:

1. The set Ψ_i^L refers to a sub-set of the search region, $\Psi_i^L \subseteq S_i$, such that it includes the global indices of the pixels that are included in the search region and satisfy the no-overlap constraint, sorted in order of increasing dissimilarity values. $L \in [2, |S_i|]$ represents the number of dissimilarity elements that are being explored during the variance accumulation process (see ALGORITHM I). Hence, for a given L, this sub-set includes the global indices of the L smallest dissimilarities in S_i, out of the patches that satisfy the non-overlap criterion.

2. The dissimilarity vector that is built from the dissimilarity elements, associated with S_i, is denoted $\tilde{\mathbf{d}}_i$ and its elements are denoted $\tilde{d}_i(j)$, where $j \in S_i$. When we refer to the sorted vector elements, arranged by order of increasing dissimilarity values, we use the notation $\tilde{d}_i\left(\Psi_i^L(m)\right)$, such that the m^{th} element of the sub-set Ψ_i^L refers to a global index $j \in S_i$, where $m \in [1, L]$. The notation $\tilde{\mathbf{d}}_i\left(\Psi_i^L\right)$ refers to the vector of sorted dissimilarity elements. In order to distinguish between the dissimilarity vector that corresponds to the Uniform and a General patch-kernels, the notations $\tilde{\mathbf{d}}_i^U$ and $\tilde{\mathbf{d}}_i^G$ are introduced.

Figure 5.2 presents a search region S_i of size $M = 11$. We refer to pixels that are arranged in a lexicographic order. In this manner, the global pixel indices $j_1, ..., j_8$, marked

in yellow, correspond to pixels whose respective patches (of size $p = 5$) do not overlap each other, nor the reference patch. As an example, and without loss of generality, let's assume the following relation between patches normalized dissimilarity values: $\tilde{d}_i(j_2) < \tilde{d}_i(j_7) < \tilde{d}_i(j_3) < \tilde{d}_i(j_1) < \tilde{d}_i(j_4) < \tilde{d}_i(j_6) < \tilde{d}_i(j_8) < \tilde{d}_i(j_5)$. In this case, the set Ψ_i^L is defined as follows: $\Psi_i^L = \{j_2, j_7, j_3, j_1, j_4, j_6, j_8, j_5\}$.

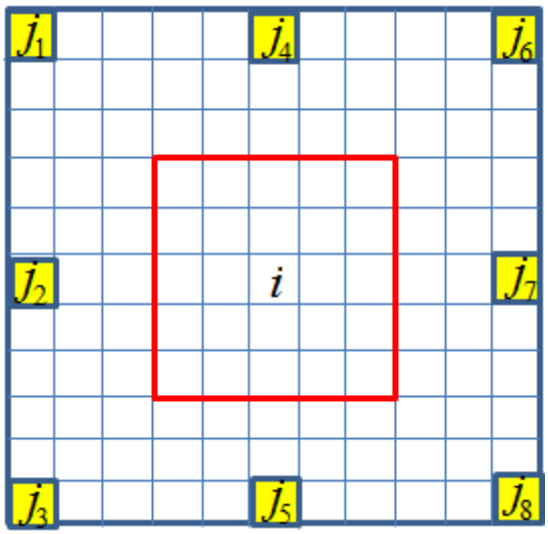


Figure 5.2.: A search region S_i of size $M = 11$. The reference patch (of size $p = 5$) is marked in red. The similarity patches (of size $p = 5$) associated with the pixels marked in yellow do not overlap each other, nor the reference patch.

Based on the new introduced notations, the covariance matrix for a General patch-kernel and L dissimilarity elements is circulant and obtained in Appendix B.1 as:

$$C_{\tilde{\mathbf{d}}_i^G} = E\left[\tilde{\mathbf{d}}_i^G \left(\tilde{\mathbf{d}}_i^G\right)^T\right] = \kappa \begin{bmatrix} 2 & 0.5 & .. & 0.5 \\ 0.5 & 2 & .. & 0.5 \\ | & | & | & | \\ 0.5 & 0.5 & .. & 2 \end{bmatrix}_{L\times L} \tag{5.1}$$

where $\kappa = \sum_{s\in[1,p^2]} \alpha_s^2$ (as defined in eqn. (3.9)) and α_s refers to the normalized coefficients

of a General patch-kernel within a similarity patch. Refer to eqn. (3.10) for the respective values of κ for the Box and the Uniform patch-kernels.

5.1.2. Effect on Statistical Properties of the Estimated Variance

The estimated variance of the normalized dissimilarities, for this case, is affected by the correlation term that was introduced in the previous sub-section (eqn. (5.1)), and its statistical analysis is required for setting the accumulated variance threshold, as explained in ALGORITHM I, presented in Chapter 3.

The sorting process deals with a set of normalized dissimilarity elements of size L, where $L \in [2, |S_i|]$. The set of sorted elements associated with S_i can be referred to as a vector $\tilde{\mathbf{d}}_i$ with elements $\tilde{d}_i\left(\Psi_i^L(m)\right)$, where $m \in [1, L]$ and $\Psi_i^L \subseteq S_i$, according to the notation introduced in sub-section 5.1.1.

The estimated (empirical) *unbiased* variance of a set of L dissimilarity elements (constituting $\tilde{\mathbf{d}}_i$) is defined as:

$$\hat{V} = \frac{1}{L-1} \sum_{l=1}^{L} \left(\tilde{d}_i\left(\Psi_i^L(l)\right) - B\right)^2 \tag{5.2}$$

where $B = \frac{1}{L}\sum_{l=1}^{L} \tilde{d}_i\left(\Psi_i^L(l)\right)$ is the estimated mean of the corresponding vector elements, and as V is a random variable.

Two important points should be emphasized:

- The L elements of the dissimilarity vector are considered here as different realizations of the same variable, just arranged in vector form, and not as an L-dimensional variable.
- The statistical properties presented in eqn. (3.11) were derived for random realizations of normalized dissimilarities of the patches whose central pixel is included in the set

of similar pixels, denoted S_i^S, but not of the sorted dissimilarities. At the beginning of the accumulation process, the estimated mean of the explored elements is expected to be smaller than 1, as at this stage, only the most similar pixels are considered and they are realizations of the lower part of the Gaussian bell. However, as we continue with this process and approach the variance threshold ($TH^G = 2\kappa$), the accumulated pixels do have the above-mentioned statistical characteristics of the unsorted set. Since we are interested in the threshold crossing point, the analysis of the estimated mean and the estimated variance that are discussed below refer to the unsorted case.

The distribution of the (unsorted) elements of the dissimilarity vector, for case 1, is (see also eqns. (3.10), (3.11)):

$$\tilde{d}_i^G(j) \sim \mathcal{N}(1, 2\kappa) \tag{5.3}$$

Consequently, the estimated mean B is a *Normal* random variable, as it is a sum of *Normal* random variables. The statistical properties of the estimated mean variable are derived in Appendix C.1 (eqn. (C.1)) and are as follows:

$$B^G \sim \mathcal{N}\left(1, \tfrac{L+3}{2L}\kappa\right) \tag{5.4}$$

As can be seen, the variance of the estimated mean is not decaying to zero for large L values. This is due to the non-zero correlation between the normalized dissimilarity elements (see eqn. (5.1)). It matches the result presented in [46] that states that the variance of a mean of L correlated variables with equal variance (in this case $\sigma^2 = 2\kappa$) and correlation coefficient ρ (in this case $\rho = 0.25$), is given by: $Var\left[B^G\right] = \frac{\sigma^2}{L} + \frac{L-1}{L}\rho\sigma^2$, which indeed gives eqn. (5.4).

After establishing the statistical properties of the estimated mean for the threshold crossing point, we can derive the properties of the estimated variance.

The estimated variance is not distributed *Chi-Square* since the dissimilarity elements are correlated. Therefore, we do not know its distribution type, but can derive its mean and variance (see Appendix C.1, eqn. (C.8)) for the threshold crossing point (considering the statistical properties of the estimated mean, as given in eqn. (5.4)):

$$\begin{aligned} \mathrm{E}\left[\hat{V}^G\right] &= \tfrac{3}{2}\kappa \\ \mathrm{Var}\left[\hat{V}^G\right] &= \tfrac{9}{2}\kappa^2 \tfrac{1}{L-1} \end{aligned} \tag{5.5}$$

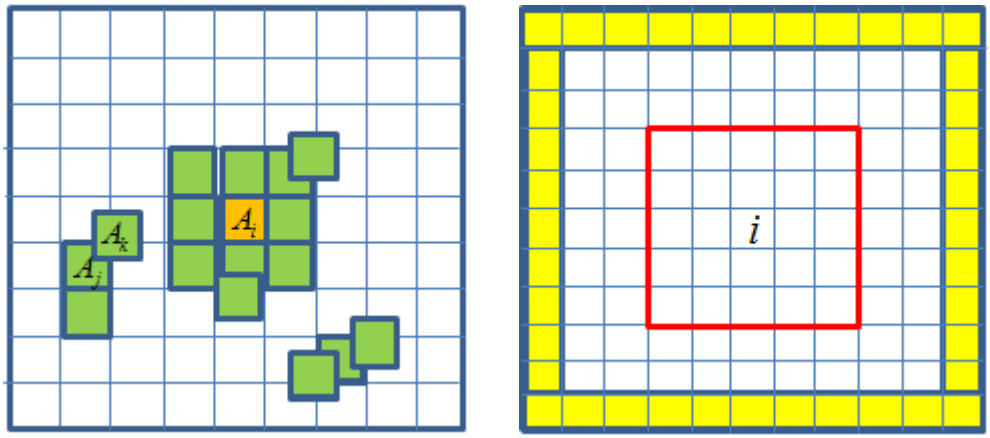
The correlation, in this case, causes a decrease of the mean of the estimated variance, as expected. Its value decreases from 2κ (eqn. (3.11)) to 1.5κ.

5.2. Case 2: Correlation Between Dissimilarities of Patches That Overlap Each Other, But Not The Reference Patch

5.2.1. Case Definition

We discuss here the case of similarity patches that overlap each other, but not the reference patch, as illustrated in Figure 5.3. In Figure 5.3 (a), the selected compared patches A_j, A_k for $j,k \in S_i$ overlap each other, such that the following overlap criterion is satisfied: $\forall j \neq k$, $j,k \in S_i^S$, $A_j \cap A_k \neq \emptyset$, $A_j \cap A_i = \emptyset$, $A_k \cap A_i = \emptyset$. Figure 5.3 (b) illustrates which pixels, whose patches (of size $p = 5$) do not overlap the reference patch, can participate in the denoising procedure of pixel i, given the overlap constraint and a search region of size $M = 11$. Consequently, the sub-set Ψ_i^L can include only the indices of the pixels marked in yellow.

As in Case 1, the reference patch A_i serves as a mutual member in the dissimilarity between it and the compared patches, thus inducing correlation between the corresponding dissimi-



(a) **(b)**

Figure 5.3.: (a) Scheme of the grid that characterizes Case 2 where patches can overlap each other, but not the reference patch. The reference patch A_i is colored in orange and the compared patches are colored in green (like A_j, A_k). (b) A search region S_i of size $M = 11$. The reference patch (of size $p = 5$) is marked in red. The yellow pixels are those whose similarity patches (of size $p = 5$) overlap each other, but not the reference patch. Pay attention that (a) refers to a patch-based grid, whereas (b) refers to a pixel-based grid.

larity elements. In this case, however, there is yet another source of correlation that stems from patches overlap. The covariance matrix is computed for the dissimilarity elements that are arranged in a vector form. The distribution of the patch dissimilarity elements remains *Chi-Square* and can be approximated by a *Normal* distribution, as presented in eqn. (3.7). The off-diagonal elements of the respective covariance matrix refer to the cross-variance between the vectorized dissimilarity elements. A detailed analytic development for General and Uniform patch-kernels can be found in Appendix B.2. Here, we present the final forms of the respective covariance matrices.

The covariance matrix for a General patch-kernel is shown in Appendix B.2 (eqn. (B.6)), to

be:

$$C_{\tilde{\mathbf{d}}_i^G} = E\left[\tilde{\mathbf{d}}_i^G \left(\tilde{\mathbf{d}}_i^G\right)^T\right] = \kappa \begin{bmatrix} 2 & 0.5 & .. & 0.5 \\ 0.5 & 2 & .. & 0.5 \\ | & | & | & | \\ 0.5 & 0.5 & .. & 2 \end{bmatrix} + 0.5O$$

where,

$$O = \begin{bmatrix} 0 & \cdots & \sum_{m \in O_{\Psi_i^L(1),\Psi_i^L(L)}} \alpha_{A_{\Psi_i^L(1)}(m)} \alpha_{A_{\Psi_i^L(L)}(m)} \\ \sum_{m \in O_{\Psi_i^L(1),\Psi_i^L(2)}} \alpha_{A_{\Psi_i^L(1)}(m)} \alpha_{A_{\Psi_i^L(2)}(m)} & \cdots & | \\ | & .. & | \\ \sum_{m \in O_{\Psi_i^L(L),\Psi_i^L(1)}} \alpha_{A_{\Psi_i^L(L)}(m)} \alpha_{A_{\Psi_i^L(1)}(m)} & .. & 0 \end{bmatrix} \tag{5.6}$$

O is a matrix obtained due to patches overlap and whose diagonal elements are all zero. The off-diagonal elements of the matrix O refers the the region of overlap between compared patches. For example, the notation $O_{\Psi_i^L(1),\Psi_i^L(2)} = \left\{t \,|\, t \in A_{\Psi_i^L(1)} \cap A_{\Psi_i^L(2)}\right\}$ refers to the global indices that represent the region of overlap between the patch that corresponds to the smallest dissimilarity and the one that corresponds to the next smallest dissimilarity and satisfy the overlap criterion. Moreover, the notation $A_{\Psi_i^L(m)}(t)$ represents the local index of the similarity patch centered at $\Psi_i^L(m)$ that corresponds to the global index t. Each off-diagonal element represents the sum of coefficient multiplication in the region of patch overlap, as illustrated in Figure 5.4. This figure presents two compared patches, of size 5×5, $A_{\Psi_i^L(m)}, A_{\Psi_i^L(s)}$, $m, s \in \left[1, M^2 - 1\right]$ whose region of overlap includes four pixels. The corresponding sum of coefficient multiplication is based on the local indices of the respective coefficients, as follows: $\alpha_{19}\alpha_1 + \alpha_{20}\alpha_2 + \alpha_{24}\alpha_6 + \alpha_{25}\alpha_7$.

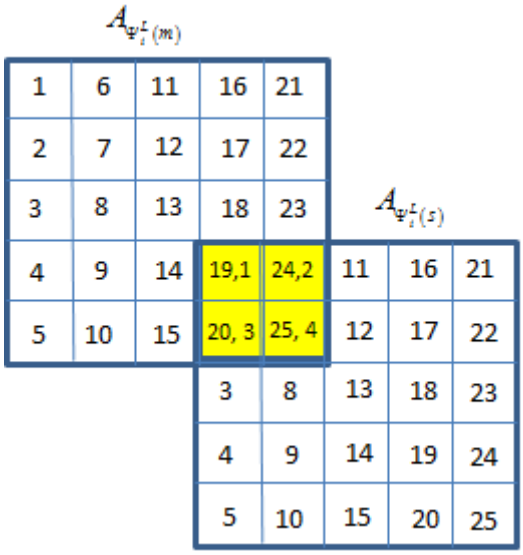


Figure 5.4.: An illustration of the overlap region between two compared patches with $p = 5$, and the local indices related to the region of overlap, marked in yellow. Each pixel in the region of overlap is characterized with two local indices. The left index is associated with the patch $A_{\Psi_i^L(m)}$ and the right index is associated with the patch $A_{\Psi_i^L(s)}$.

The covariance matrix that corresponds to the Uniform patch kernel has a simpler form of the matrix O, shown in Appendix B.2, eqn. (B.7), and given by assigning $\kappa = p^{-2}$:

$$O = \kappa^2 \begin{bmatrix} 0 & \dots & \left|O_{\Psi_i^L(1),\Psi_i^L(L)}\right| \\ \left|O_{\Psi_i^L(2),\Psi_i^L(1)}\right| & \dots & | \\ | & .. & | \\ \left|O_{\Psi_i^L(L),\Psi_i^L(1)}\right| & .. & 0 \end{bmatrix} \tag{5.7}$$

5.2.2. Effect on Statistical Properties of the Estimated Variance

We remind the reader that the statistical properties of the elements of the normalized dissimilarity vector, for this case, are similar to Case 1, as indicated in eqn. (5.3).

In this case, compared patches may overlap each other, but not the reference patch. This overlap adds another source of correlation between the dissimilarity vector elements, as

described in detail in Appendix B.2. This correlation, together with the correlation described in Case 1, modify the statistical properties of the estimated mean and estimated variance.

The statistical properties of the estimated mean variable, at the threshold crossover point, are analyzed in appendix C.2 for the General (eqn. (C.9)) and the Uniform (eqn. (C.10)) patch-kernels. For simplicity, we present here the final form for only the Uniform patch-kernel, that is given by ($\kappa = p^{-2}$):

$$B^U \sim \mathcal{N}\left(1, \frac{L+3}{2L}\kappa + \frac{\kappa^2}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right|\right) \tag{5.8}$$

where $\Psi_i^L \subseteq S_i$ is a sub-set of the search region S_i that consists of the pixel global indices $j \in S_i$ that corresponds to sorted dissimilarity values, i.e., the L smallest dissimilarities values, under the consideration of the overlap criterion between patches A_j, A_k, presented in sub-section 5.1.2.

The mean of the estimated variance, for a General patch-kernel, is developed in Appendix C.2 (eqn. (C.12)). Here, we present the simplified form that characterizes the Uniform patch-kernel (eqn. C.3) and is based on assigning $\kappa = p^{-2}$:

$$E\left[\hat{V}^U\right] = \frac{3}{2}\kappa - \frac{\kappa^2}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right| \tag{5.9}$$

The variance of the estimated variance, in this case, involves a complicated development due to the complicated form of the covariance matrix (see eqn. (5.6)). In Case 1 the corresponding covariance matrix is circulant and a we could diagonalize it by applying a whitening transform, as explained in Appendix C.1. Then, we were able to use the transform domain, where the dissimilarities are not correlated, in order to compute the variance of the estimated variance. In the current case, applying a whitening transform creates complicated terms, whose analysis in not trivial due to the addition of the overlap terms. Further discussion can be found in Appendix C.2.

It can be seen from eqn. (5.9) that the correlation term due to the overlap between the patches themselves causes a decrease in the mean of the estimated variance, compared to its mean in Case 1 (i.e., 1.5κ). This decrease, however, is relatively small since it is proportional to κ^2, where $\kappa \ll 1$, even for just $p = 5$ that we use. Moreover, this overlap term has to be computed for each explored sub-set in each search region, which makes the computation impractical.

5.3. Case 3: Correlation Between Dissimilarities of Patches That Overlap Each Other And The Reference Patch

5.3.1. Case Definition

We finally discuss the most general case of similarity patches that overlap each other as well as the reference patch, as illustrated in Figure 5.5. In this figure, the selected compared patches A_j, A_k for $j,k \in S_i$ overlap each other, and each one of them also overlaps the reference patch A_i, satisfying the following overlap criterion: $\forall j \neq k,\ j,k \in S_i^S$, $A_j \cap A_k \neq \emptyset$, $A_j \cap A_i \neq \emptyset$, $A_k \cap A_i \neq \emptyset$.

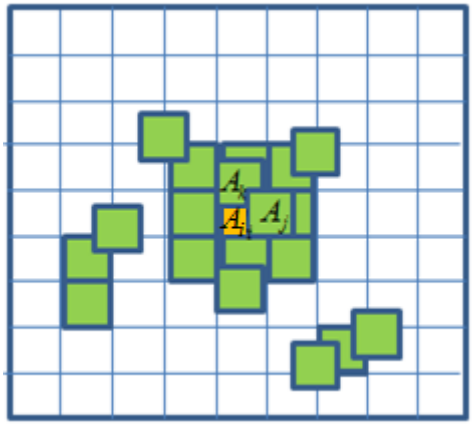


Figure 5.5.: Scheme of the grid that characterizes Case 3 where patches can overlap each other and the reference patch. The reference patch A_i is colored in orange and the Compared patches are colored in green (like A_j, A_k).

Here we get an additional source of correlation, as compared to the two previous cases, which is the overlap of the compared patches with the reference patch. The covariance matrix is computed for the dissimilarity elements that are arranged in a vector form. The off-diagonal elements of the respective covariance matrix refer to the cross-variance between the vectorized dissimilarity elements, and their elaborated analytic development for General and Uniform patch-kernels can be found in Appendix B.3. We present here the final forms of the respective covariance matrices.

The diagonal elements of the covariance matrix refer to the variance of each dissimilarity element. In this case, the distribution of patch dissimilarity is not *Chi-Square* anymore because not all the elements in the summation defining it (see eqn. (3.1)) are independent, in contrast to what was assumed by Buades et al. [4] and Thacker at. al [38]. The mean and variance of the dissimilarity for a General patch-kernel are analyzed in Appendix B.3, resulting in:

$$\mathrm{E}\left[\tilde{d}_i^G(j)\right] = 1,\ \mathrm{Var}\left[\tilde{d}_i^G(j)\right] = 2\kappa + \sum_{f \in O_{i,j}} \alpha^2_{A_i(f)} \alpha^2_{A_j(f)} \tag{5.10}$$

where $O_{i,j} = \left\{ m \,|\, m \in A_i \cap A_j \right\}$ is the set of pixels associated with the overlap between the similarity patch A_j and the reference patch A_i and the cardinality of the set is denoted $\left|O_{i,j}\right|$. The notation $\alpha_{A_i(f)}$ refers to the local index that corresponds to the global index f of the similarity patch A_i.

The corresponding statistics for the Uniform patch-kernel are obtained by assigning $\alpha_s = p^{-2}$, $\forall s \in \left[1, p^2\right]$, hence $\kappa = p^{-2}$:

$$\mathrm{E}\left[\tilde{d}_i^U(j)\right] = 1, \ \mathrm{Var}\left[\tilde{d}_i^U(j)\right] = 2\kappa + \left|O_{i,j}\right|\kappa^2 \tag{5.11}$$

As mentioned earlier, the cross-variance between the dissimilarity elements is also developed in Appendix B.3, for both General and Uniform patch-kernels. The cross-variance terms for the General kernel have complicated expressions and are shown only in the Appendix. For the Uniform kernel ($\kappa = p^{-2}$), the expression for the cross-variance between dissimilarity elements $j,k \in S_i^S$ is simpler and is given by:

$$\mathrm{Cov}\left[\tilde{d}_i^U(j), \tilde{d}_i^U(k)\right] = 0.5\kappa + 0.5\kappa^2\left(\left|O_{i,j}\right| + \left|O_{i,k}\right| + \left|O_{j,k}\right|\right) + \begin{cases} \kappa^2\left|O_{i,j}\right| & if \ \left|O_{i,j}\right| = \left|O_{i,k}\right| \\ 0 & Otherwise \end{cases} \tag{5.12}$$

5.3.2. Effect on Statistical Properties of the Estimated Variance

We remind the reader that unlike the former two cases, the statistical properties of the elements of the normalized dissimilarity vector for this case are not distributed *Chi-Square*, and their mean and variance are introduced in eqn. (5.10) for the General patch-kernel and eqn. (5.11) for the Uniform patch-kernel.

In this case, compared patches may overlap each other, as well as the reference patch. This overlap adds yet another source of correlation between the dissimilarity vector elements, as

described in detail in Appendix B.3. This correlation, together with the correlation described in Case 1, modify the statistical properties of the estimated mean and estimated variance.

In contrast to the two other explored cases, the estimated mean, in this case, is not a normal random variable since the dissimilarity elements are not distributed normally. Hence, we derive some of its statistical properties, but not its distribution type. The statistical properties of the estimated mean variable, at the threshold crossover point, are analyzed in appendix C.3 for the General patch-kernel (eqn. (C.14)) and the Uniform patch-kernel (eqn. (C.15)). For simplicity, we present here only the final form for a Uniform patch-kernel, that is given as follows:

$$\begin{aligned} \mathrm{E}\left[B^U\right] &= 1 \\ \mathrm{Var}\left[B^U\right] &= \tfrac{L+3}{2L}\kappa + \tfrac{\kappa^2}{L^2}\sum_{l=1}^{L}\left|O_{i,\Psi_i^L(l)}\right| + \\ &+ \tfrac{1}{2L^2p^4}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\left|O_{i,\Psi_i^L(l)}\right| + \left|O_{i,\Psi_i^L(k)}\right| + \left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right|\right) \\ &+ \tfrac{1}{L^2p^4}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,k\Psi_i^L(k)}\right|\right)\left|O_{i,\Psi_i^L(l)}\right|\right) \end{aligned} \tag{5.13}$$

where $\Psi_i^L \subseteq S_i$ is a sub-set of the search region S_i that consists of the pixel global indices $j \in S_i$ that corresponds to sorted dissimilarity values, i.e., the L smallest dissimilarities values, under the consideration of the overlap criterion between compared patches and the overlap with the reference patch. The notation $O_{i,\Psi_i^L(k)}, k \in [1,L]$ corresponds to the set of global indices in the overlap region between the reference patch A_i and the patch $A_{\Psi_i^L(k)}$,i.e., the patch that is associated to the k^{th} dissimilarity element in the sorted vector $\tilde{\mathbf{d}}_i^G\left(\Psi_i^L\right)$. The notation $\mathbf{1}(\cdot)$ refers to the Indicator function.

The mean of the estimated variance is developed in Appendix C.3 for a General patch-kernel (eqn. (C.16)) as well as for the Uniform patch-kernel (eqn. (C.17)). Here, we present the

final much simpler form that corresponds to the Uniform patch-kernel, that is given by:

$$\begin{gathered}\mathrm{E}\left[\hat{V}^{U}\right]=\frac{3}{2}\kappa+\frac{\kappa^{2}}{L}\sum_{l=1}^{L}\left|O_{i,\Psi_{i}^{L}(l)}\right|- \\ -\frac{\kappa^{2}}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left[\left|O_{i,\Psi_{i}^{L}(l)}\right|+\left|O_{i,\Psi_{i}^{L}(k)}\right|+\left|O_{\Psi_{i}^{L}(l),\Psi_{i}^{L}(k)}\right|\right] \\ -\frac{\kappa^{2}}{L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left[\mathbf{1}\left(\left|O_{i,\Psi_{i}^{L}(l)}\right|=\left|O_{i,\Psi_{i}^{L}(k)}\right|\right)\left|O_{i,\Psi_{i}^{L}(l)}\right|\right]\end{gathered} \tag{5.14}$$

The variance of the estimated variance, in this case, involves a complicated development due to the complicated form of the cross-covariance terms (see Appendix B.3, eqns. (B.9), (B.10)). Similarly to Case 2, applying a whitening transform to the covariance matrix, in order to compute the variance in a domain where the dissimilarities are not correlated, results in non-trivial analysis, as explained in Appendix C.3.

Here too, the correlation term due to the overlap of the patches with each other and/or the reference patch, causes just a small change in the mean of the estimated variance, compared to Case 1 (i.e., 1.5κ). This change is relatively small because it is proportional to κ^2, where $\kappa \ll 1$, even for just $p = 5$. As for the former case, these terms have to be computed for each explored sub-set in each search region, making the computation impractical.

5.4. Effect of Correlation on The Model-Based Scheme

The model introduced in Chapters 3,4 does not consider the correlation between the dissimilarity elements in a given search region S_i. The correlation analysis, presented above, provides us with a more accurate variance estimator compared to the one used in ALGORITHM III, introduced in Chapter 4.

In the previous sections of this chapter, we analyzed the statistical characteristics of the estimated mean and estimated variance of the normalized dissimilarity elements at the variance threshold crossover point, considering three sources of correlation. As was explained for cases 2 and 3, which consider overlaps both between patches and/or with the reference

patch, the computation of the respective mean of the estimated variance for each explored search region is impractical, and the overlap terms have anyway a minor contribution to the estimated variance mean that was obtained for Case 1 (i.e., 1.5κ), which relates only to correlation due to the mutual reference patch.

Thus, instead of using the threshold value in eqn. (3.21), which assumes no correlation, we use here a threshold value that is based on eqn. (5.5):

$$TH^G = \mathrm{E}\left[\hat{V}^G\right] + f \cdot STD\left[\hat{V}^G\right] = \tfrac{3}{2}\kappa\left(1 + f\sqrt{\tfrac{2}{L-1}}\right) \tag{5.15}$$

The factor f is a parameter that is selected empirically, as was done in Chapter 3, for the model that does not consider any correlation between dissimilarities. Its selection for the model that does consider correlation is discussed in the following chapter.

An attempt to enforce a situation where the only source of correlation would be the mutual reference patch, so we would have closed form simple expressions for the expectation and the variance of the estimated variance (as in Case 1), is not efficient for denoising because it requires increasing much the search region size (M^2) in order to get a sufficient number of patches in the denoising process of each pixel. We assume that a natural image is stationary within a given search region, and increasing its size may violate this assumption.

5.5. Chapter Summary

In this chapter we analyzed three sources of correlation between normalized dissimilarities, resulting from the comparison to the same reference patch A_i and the overlap between compared patches and possibly also with the reference patch. These sources of correlation are analyzed in an ascending order of complexity. The simplest case (Case 1) refers only to the correlation due to the mutual reference patch and is characterized by a circulant covariance matrix, whose unique structure was beneficial for the analysis of the variance of the esti-

mated variance for this case, as explained in Appendix C.1. The other two cases (Case 2, Case 3) are characterized by a more complicated form of a covariance matrix, due to the overlap between the compared patches and the overlap of a given patch with the reference patch. These three sources of correlation were not discussed before by other researchers in the field that explored the statistical properties of the NLM dissimilarity measure.

Moreover, based on the analysis of the covariance matrices for the three sources of correlation, we were able to derive statistical properties of the estimated mean and estimated variance at the variance threshold crossover point. For cases 2 and 3, the mean of the estimated variance depends also on the degree of overlap between the explored patches and the overlap with the reference patch. Consequently, these cases yield complicated terms. However, these terms are relatively small as they are proportional to κ^2, where $\kappa \ll 1$, even for just $p = 5$ that we use. Thus, their contribution to the mean of the estimated variance is minor compared to the value obtained in Case 1.

Finally, we use the correlation-dependent estimate of the estimated variance to set the threshold value in ALGORITHM III. The performance of the modified scheme, denoted the Correlation-Dependent Model-Based scheme, and its comparison to the original scheme of ALGORITHM III are discussed in the following chapter.

6. Performance of The Correlation-Dependent Model-Based Scheme

This chapter discusses the performance of the Correlation-Dependent Model-Based scheme, introduced in section 5.4, and compares it to the model-based scheme of ALGORITHM III, presented in Chapter 4. The difference between the two schemes is based on the effect of the correlation between the dissimilarities, within a given search region, on the estimated variance. As expected, the correlation causes a decrease in the variance between the corresponding elements, reducing it from 2κ (eqn. (3,11) to1.5κ (eqn. (5.5)). As suggested in Chapter 3 (section 3.5) for the original model-based scheme, and in Chapter 5 (section 5.4) for the correlation-dependent model-based scheme, the variance threshold is modified by adding a variance error correction term whose added value is discussed in this chapter. Finally, since the normalization of the dissimilarities assumes a prior knowledge of σ_n, which typically needs to be estimated, we refer at the end of the chapter to the sensitivity of the two adaptive schemes and the standard NLM to an error in the noise variance estimation.

6.1. Setting the Factor f

The factor f implies on the deviation of the threshold value from the mean of the estimated variance, according to the correlation-based model (see eqn. (5.15)), aiming to improve denoising performance. It is selected empirically with PSNR being the optimization measure. As in section 3.5, for the model that does not consider correlations, the simulations were performed using ten selected natural images, with additive Gaussian white noise at various levels, and exploring the denoising of each noisy image using different f values. The noise levels were $\sigma_n = [10, 35]$ with a stride of 5, and the explored f values were $f = [0, 3]$ with a stride of 0.2. The denoising results were averaged over all images and noise levels. Figure 6.1 presents the average PSNR as a function of f, with error bars, marked in red, indicating the variance over different noise levels for each examined f value . The global maximum is obtained for $f = 2$. Without considering the error STD (error bars), it can be seen that using an f value that is different from zero can improve the denoising results by 0.06 dB, on average, whereas for the no-correaltion threshold (section 3.5), using an f value that is larger than zero ($f = 1$) improved the denoising performance by only 0.01 dB, on average. When exploring the individual contribution of the utilization of $f = 2$ to the PSNR of specific images, we discovered that there is a maximal increase of 0.2 dB, which is more significant than the case of no-correlation, where the maximal increase (for $f = 1$) was 0.14 dB. Consequently, we suggest to continue here with $f = 2$.

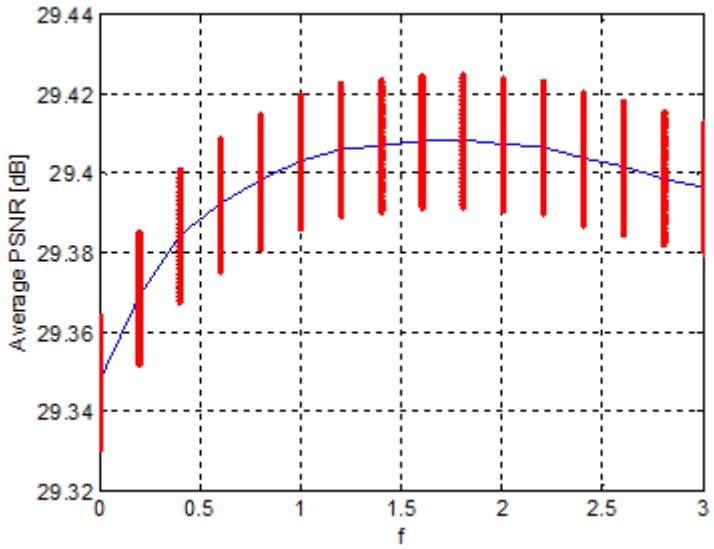


Figure 6.1.: f optimization curve

6.2. Performance Comparison of Denoising Schemes

The performance of the scheme presented in ALGORITHM III, with no correlation consideration, was compared to the scheme that considers correlation and introduced in section 5.4. Moreover, this latter scheme was also compared to the standard NLM applied using Uniform and Box patch-kernels. The denoising parameters that were used for the comparison are: $p = 5$, $M = 11$, $h = \sigma_n$, $f = 2$ for the correlation scheme and $f = 0$ for the no-correlation scheme. The following table presents the comparison results for different selected images and various noise levels. As was shown before (Chapter 4), ALGORITHM III without correlation consideration, is characterized with better denoising results, compared to the Standard NLM, both visually, PSNR-wise and SSIM-wise. The Correlation-Dependent Model-Based scheme is slightly better than its simplified version, mainly PSNR-wise. The PSNR difference, for the images and noise conditions presented in the table, is maximum 0.04 dB, and is not noticeable visually.

Image	σ_n	NLM w. Uniform PSNR [dB]/SSIM	NLM w. Box PSNR [dB]/SSIM	ALGORITHM III PSNR [dB]/SSIM	Correlation-Based ALGORITHM III PSNR [dB]/SSIM
Lena	20	30.11/0.87	30.25/0.87	30.48/0.88	**30.51/0.88**
Baboon	20	24.78/0.69	25.54/0.74	25.62/0.75	**25.64/0.75**
Barbara	30	26.92/0.8	26.94/0.8	27.16/0.81	**27.18/0.81**
Pirate	15	30.55/0.84	31.02/0.85	31.08/0.85	**31.12/0.85**

Table 6.1.: Quantitative comparison between the Standard NLM applied with Uniform and Box patch-kernels, the basic model-based NLM (ALGORITHM III without correlation consideration) with $f = 0$ and the correlation-dependent model-based NLM with $f = 2$ (ALGORITHM III with correlation consideration).

Figure 6.2 presents a performance comparison between the two model-based schemes as function of noise level. The displayed curves are the result of averaging over ten explored natural images. As observed from the results presented in Table 6.1, the Correlation-Dependent scheme is only slightly better, PSNR-wise, than the basic model-based scheme. It is more pronounced for low noise levels.

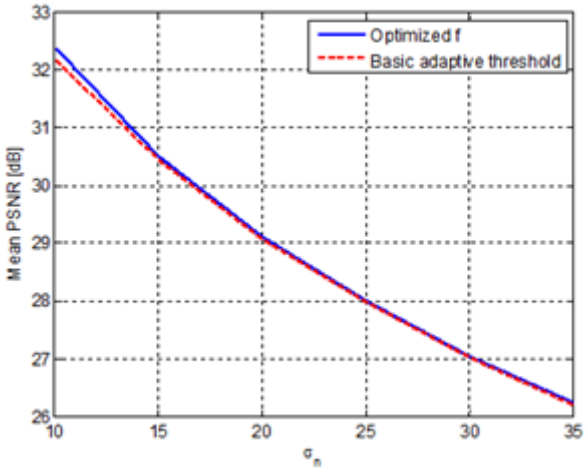


Figure 6.2.: Comparison between the two model-based NLM schemes: the blue curve refers to ALGORITHM III with correlation consideration, whereas the dashed red curve refers to ALGORITHM III without correlation consideration.

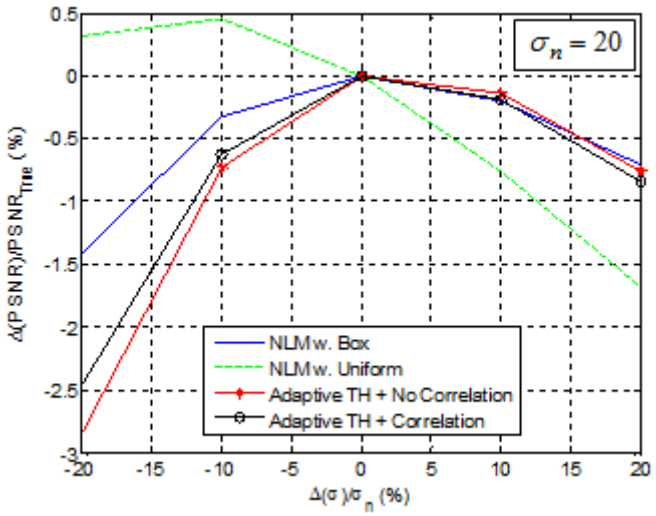
6.3. Sensitivity to Noise Level

In real scenarios, the noise level is an unknown parameter that has to be estimated. The estimation algorithm is beyond our scope, as there are quite a few suggested approaches in the literature (e.g., [8, 24]). We tested the sensitivity of our model-based approach and compared it to the sensitivity of the standard NLM. The objective of such an examination is to explore the quality of denoising when noise level estimation is not accurate. Figure 6.3 presents the sensitivity of four compared NLM approaches: standard NLM applied using the Box patch-kernel, standard NLM applied using the Uniform patch-kernel, Modified model-based NLM (with correlation consideration) with $f = 2$, and Basic model-based NLM (no correlation consideration). The sensitivity is measured for two selected noise levels $\sigma_n = 20, 30$ and the explored range of deviation for a given noise level is $[0.8\sigma_n, 1.2\sigma_n]$, where PSNR is the performance measure. The results were averaged over five natural images. The displayed curves refer to normalized PSNR differences between the PSNR obtained for an estimated noise level and PSNR obtained for the true noise level. The normalizing value was the PSNR obtained for the true noise level ($PSNR_{True}$). The NLM parameters that were used are $p = 5$, $M = 11$, $h = \sigma_n$.

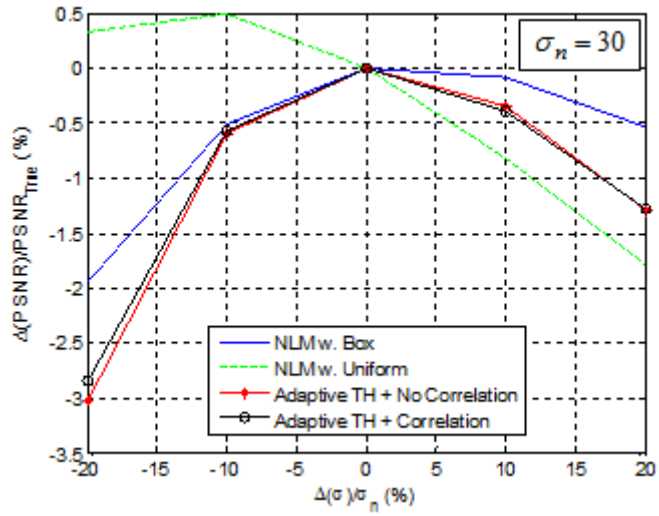
From the above comparison, one can observe that the standard NLM applied using the Box patch-kernel (solid blue curve) is somewhat less sensitive than the two adaptive approaches (black curves with circles for the correlation-dependent scheme and red curve with dots for the basic model-based scheme). As for the two model-based approaches, the sensitivity of the Correlation-Dependent model-based scheme is slightly lower than that of the Basic model-based scheme, when the estimated noise level is lower than the true level. The sensitivity tendency reverts when the estimated noise level is higher than the true level. The standard NLM applied using the Uniform patch-kernel (dashed green curve) is the most sensitive both when estimated noise level is lower and higher than the true noise level. Moreover, the PSNR increases when the noise level is underestimated, implying that a smaller value of the

smoothing parameter h (that is proportional to the noise level) should have been used when NLM is applied with the Uniform patch-kernel, to this particular set of noisy images.

Suggested noise level estimation approaches (e.g., [8]) are characterized by an estimation error of approximately 10%. A recently suggested approach by Liu et al. [24] is characterized by smaller estimation errors, of less then 2%. It can therefore be assumed that the performance sensitivity to noise level estimation errors up to 10%, for the four examined approaches, is comparable, with a change in PSNR of at most 0.5%.



(a)



(b)

Figure 6.3.: Simulations that explore the performance sensitivity (in PSNR) to noise level estimation error at two different noise levels. The solid blue curve corresponds to Standard NLM applied using the Box kernel, the dashed green curve corresponds to Standard NLM applied using the Uniform kernel, the red curve with dots corresponds to the model-based approach without correlation consideration and $f = 0$, and the black curve with circles corresponds to the model-based approach with correlation consideration and $f = 2$. (a) $\sigma_n = 20$, (b) $\sigma_n = 30$.

6.4. Chapter Summary

This chapter explores the performance of the modified model-based approach, i.e., ALGORITHM III with correlation consideration and compares it to the basic model-based approach, i.e., ALGORITHM III without correlation consideration, and to the standard NLM approach. Simulations show that the modified approach is visually comparable and is slightly better PSNR-wise than the basic scheme, which does not consider any source of correlation. Both approaches are better, mainly visually, than the standard NLM method, applied using either the Box or the Uniform patch-kernels.

We also explored the sensitivity to noise level estimation error of the two proposed model-based approaches and compared it to the sensitivity of the standard NLM. Within the range of the common estimation error (approximately 1-10%), all explored methods are comparable in their sensitivity, and the effect on PSNR is limited to 0.5%.

7. Block Matching 3D (BM3D) Combined with The Model-Based Scheme

BM3D [9] is considered to be the state-of-the-art denoising approach that achieves the best performance over other reported image denoising algorithms, however it is computationally expensive and requires multiple parameters setting. We suggest to integrate our model-based approach that defines an adaptive search region for each pixel in the image, with the BM3D denoising method. By introducing our model-based adaptive search region in one of the stages of the BM3D approach, we avoid the need for parameter calibration and save computations. We compare the denoising performances of the original BM3D and the BM3D combined with our model-based approach and show that they are comparable.

7.1. Brief Description of The BM3D Algorithm

BM3D is based on the fact that an image has a locally sparse representation in a given transform domain (e.g., Bi-orthogonal wavelet transform, DCT, Haar). This sparsity is enhanced by grouping similar 2D image patches into 3D groups. The grouping and filtering procedure are named Collaborative Filtering.

The basic BM3D algorithm assumes a Gaussian additive noise, using the noise model presented in eqn. (2.1). In general, the algorithm includes two main consecutive phases that consist of similar stages applied on different inputs, as presented in Figure 7.1. These stages are reviewed later on. The first phase receives as input the noisy image and estimates the noise-free image using a hard thresholding operator. This estimate is referred to as the *Basic Estimate*. The second phase is based both on the noisy image and on the basic estimate obtained in the first phase, and applies Wiener filtering. A detailed explanation of the stages of each phase can be found in Appendix D. Here, we present an overview of the main operations that characterize each of the two phases:

1. **Grouping** - Finding the image patches similar to a given reference patch, marked as *R* in Figure 7.1 (blue *R* refers to the noisy reference patch and red *R* refers to the basic estimate reference patch). The search for similar patches is limited to a defined search region, centered at the POI. Each phase uses a different patch similarity measure, where Phase 1 dissimilarity is measured in the transform domain (after coefficients shrinkage, applied using a hard thresholding operator) and Phase 2 dissimilarity is measured in image domain. Once the similar patches have been found, they are arranged in a 3D block, named a group. Phase 1 grouping is based on the noisy patches, whereas Phase 2 grouping is based on the denoised patches that were obtained in Phase 1. Phase 2 includes another group that is based on the noisy patches whose locations are extracted from the basic estimate grouping. The explored patches in each group may overlap, and each patch may be associated with different groups. Consequently, each pixel may be associated with different groups and different patches. Moreover, groups may differ in their cardinality.

2. **Collaborative Filtering -** A 3D isometric linear transform is applied to each group, followed by a shrinkage [9] of the transform spectrum. Phase 1 shrinkage is applied using a hard thresholding operator, whereas the shrinkage in Phase 2 is applied by Wiener filtering. Finally, an inverse linear transform is applied to estimate each patch

in the 3D group. This stage is described in Figure 7.1 as four consecutive steps: 3D transform, hard-thresholding for Phase1 and wiener filtering for Phase 2, inverse 3D transform, and finally block-wise estimates.

3. **Aggregation -** Since each pixel may be associated with different patches and different groups, it may posses several estimates. In this stage, all the estimates of a given pixel are aggregated using a weighted average, where each phase has its own weighting scheme.

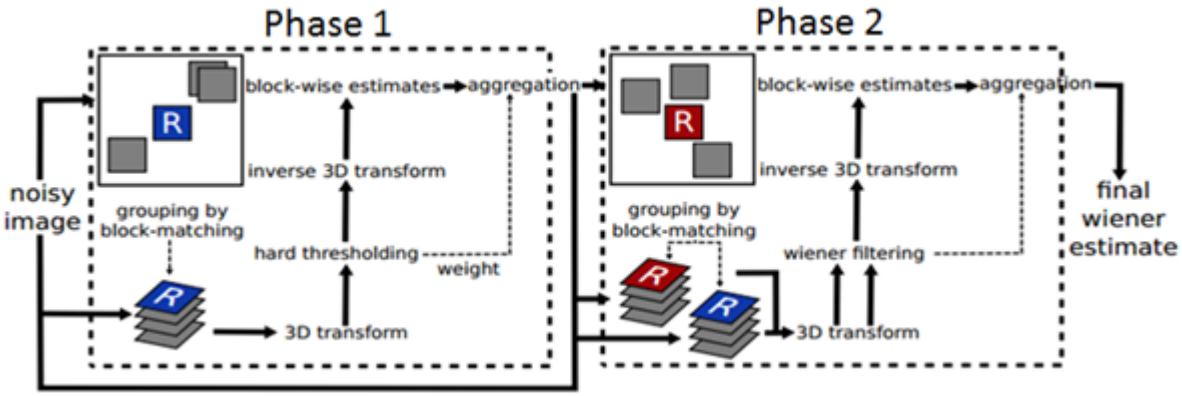


Figure 7.1.: Scheme of the BM3D algorithm [9]

7.2. Model-Based Block Matching

First phase grouping is applied in transform domain, i.e., the dissimilarity between patches of a defined search region S_i is computed using the transformed patches. Then, a hard-thresholding operator is applied, such that only the most similar transformed patches are selected to be associated with the corresponding group. We suggest a more efficient method to create a group of similar patches in the first phase (Basic Estimate) of the BM3D algorithm. This method relies on the model-based NLM approach, which is used to define a set of pixels whose neighborhood is similar to a neighborhood of a given reference pixel. Our

alternative method is based on a model that characterizes similar patches, and if we refer to the model that does not consider correlation between dissimilarities it does not require any parameter setting. Recall that for the model that does consider correlation, a single empirical parameter was set based on PSNR optimization. However, since the two models provided similar results for pixel selection (see Table 6.1), we continue here with the model that does not consider correlation, and hence is parameter-free. Consequently, it reduces the degrees of freedom of the first phase grouping from two to zero and saves computations since there is no need to use the transform domain.

The BM3D algorithm uses similarity patches of an even size, e.g., $p^{Hard} = 8$, in contrast to the NLM approach that uses an odd size, e.g., $p = 5$. Therefore, the model-based approach is applied using the Uniform patch-kernel that assigns same weights to all the patch pixels, and not other kernels, such as the Box patch-kernel that are defined based on an odd patch size and assigns higher weights to pixels that are located closer to the patch center.

ALGORITHM IV below explains our grouping methodology for the Basic Estimate phase. It relies on ALGORITHM I, which was introduced in Chapter 3. The similarity patch size is defined to be $p^{Hard} \times p^{Hard}$, and the search region size is defined to be $M^{Hard} \times M^{Hard}$. The algorithm input is an initial search region of a given POI, and its output is the most similar patches extracted from the resultant S_i^S, arranged in a 3D array.

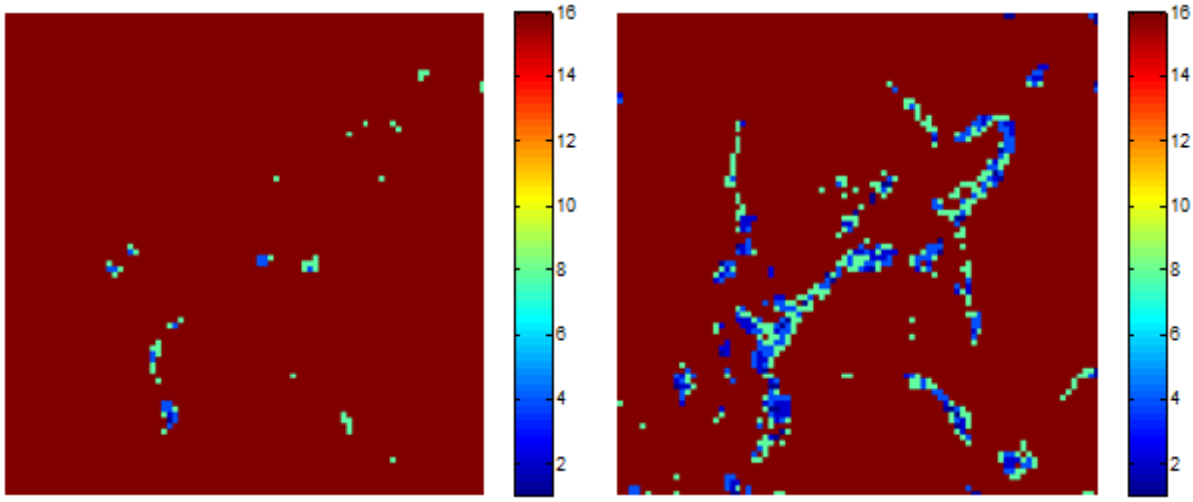
ALGORITHM IV: MODEL-BASED BASIC ESTIMATE GROUPING

1. Compute the dissimilarity between patches A_i and A_j, $j \in S_i$ in the image domain, based on the dissimilarity definition in eqn. (2.6) using a Uniform patch-kernel.
2. Normalize the computed dissimilarity elements by the noise STD, as defined in eqn. (3.5).
3. Sort the normalized dissimilarity elements in an ascending order.
4. Compute accumulated variance, by starting with the first two elements (of smallest dissimilarity) and adding one element at a time. Stop the accumulation process once the accumulated variance is above the model-based threshold or when the cardinality of the accumulated patches equals B^{Hard}, according to which condition comes first. Thus, each group contains at most B^{Hard} most similar patches.
5. The accumulated pixels define the set S_i^S, and the patches associated with this set of pixels are candidates to form the corresponding group of the reference patch A_i.

The suggested grouping can be applied only on the original noisy patches, whose noise model is known to be additive Gaussian, thus it is appropriate only for the grouping process in the Basic Estimate phase. The grouping in the Wiener phase is applied on the patches extracted from the Basic Estimate, i.e., after an initial denoising. In order to apply the model-based approach on the denoised patches, we are required to estimate their noise model, which is not necessarily Gaussian and additive. Moreover, even if the noise model of the denoised patches could have been estimated, there is no guarantee that the corresponding dissimilarity variance that is required to set the adaptive search region could be estimated.

Figure 7.2 displays Phase 1 group cardinality map for the two grouping approaches, applied on the image Lena with $\sigma_n = 20$. The following parameters were used to create the corresponding groups: $p^{Hard} = 8, M^{Hard} = 39, B^{Hard} = 16$. It can be clearly seen that pixels located on edges are characterized with smaller groups for the model-based grouping

approach, such that dissimilar patches are not selected for the denoising process. This signifies the advantage of our proposed grouping with respect to the basic estimate grouping.



(a) **(b)**

Figure 7.2.: Basic estimate group cardinality map, applied on the image Lena with $\sigma_n = 20$ using the parameters: $p^{Hard} = 8, M^{Hard} = 39, B^{Hard} = 16$. (a) Original Basic Estimate grouping. (b) Model-Based grouping.

The following table summarizes the difference between our proposed grouping methodology, denoted Model-Based Grouping, and the BM3D Basic Estimate Grouping.

Index	BM3D Basic Estimate Grouping	Number of Operations per search region	Model-Based Grouping	Number of Operations per search region
1	Transform patches	$0.5M^2 plog_2 p$ arithmetic		
2	Apply hard thresholding operator on transformed patches	$M^2 p^2$ comparisons		
3	Compute dissimilarities in transform domain	$2M^2 p^2$ arithmetic	Compute normalized dissimilarities in image domain	$2M^2 p^2$ arithmetic
4	Sort dissimilarities in an ascending order	$log_2 M^2$ comparisons	Sort normalized dissimilarities in an ascending order	$log_2 M^2$ comparisons
5	Apply hard-thresholding operator on computed dissimilarities	$log_2 M^2$ comparisons	Accumulated variance computation and variance threshold application	$0.72M^2$ arithmetic $log_2 M^2$ comparisons
6	Choose at most B^{Hard} most similar patches	0	Choose at most B^{Hard} most similar patches	0

Table 7.1.: Comparison between the flow of the Model-Based Grouping and that of the BM3D Basic Estimate Grouping.

By comparing the two grouping methods, we can see that the Model-Based approach saves computations as there is no need to apply a transform and the hard-thresholding operator on the transformed patches (operations 1-2 of the BM3D Basic Estimate grouping in the table). When we use a search region of size $M \times M$ and a similarity patch of size $p \times p$, these two stages require $0.5M^2 plog_2 p$ arithmetic operations for patch transformation (for an efficient transform implementation), in addition to $M^2 p^2$ comparison operations for hard-thresholding application. Operations 3,4, and 6 have similar complexity for the two grouping methods. Operation 5 of the Model-Based method is more complicated than its respective operation in the original grouping method, because of the accumulated variance computation module. This module requires approximately $0.72M^2$ arithmetic operations, assuming (based on simulations) that 30% of the pixels in a natural image have an adaptive

search region of size M^2 and the other 70% have an average of $0.6M^2$ pixels in their adaptive search region. Hence, the entire model-based grouping requires $M^2\left(2p^2+0.72\right)$ arithmetic operations and $2log_2M^2$ comparisons, whereas the entire basic estimate grouping requires $M^2\left(0.5plog_2p+2p^2\right)$ arithmetic operations and $M^2p^2+2log_2M^2$ comparisons. The ratio between the basic estimate and the model-based grouping for the arithmetic operations is $\frac{0.5plog_2p+2p^2}{0.72+2p^2}$ and for the comparisons is $1+\frac{M^2p^2}{2log_2M^2}$, such that for $p=8$ and $M=39$ the basic estimate grouping approach requires overall approximately 9% more arithmetic operations and $4.6\cdot10^3$ times more comparisons per search region. The overall running time of the two grouping methods in Matlab for an image of size 256×256, a search region of size 39×39, a similarity patch of size 8×8, and $B^{Hard}=16$ lasts 270 seconds for the basic estimate grouping method and 240 seconds for the model-based grouping. This is a reduction of more than 11% in running time for the grouping step and 4.5% in running time for the whole BM3D process. On top of that, our proposed approach is parameter independent.

7.3. Experimental Results

We compare the Basic Estimate denoising results for the two methods of Basic Estimate grouping and also the final results, after the second phase . For our proposed grouping method, we use 2κ as the variance threshold (see ALGORITHM I), where $\kappa=\left(p^{Hard}\right)^{-2}$. This threshold refers to the basic model-based scheme, that does not consider correlations between dissimilarities. .

Table 7.2 presents the BM3D results after applying both the hard-thresholding phase (Phase 1) and the Wiener phase (Phase 2), where the grouping of Phase 1 is applied either using the BM3D approach or the model-based approach. The PSNR and SSIM differences between the two grouping approaches, both after Phase 1 and Phase 2, are insignificant and no visual differences were observed, as can be seen in the example presented in Figure 7.3. This figure presents the denoising results of the image Peppers with $\sigma_n=20$ after applying the

two phases of the BM3D algorithm. Figure 7.3 (b) refers to the denoising result using the basic estimate grouping, whereas Figure 7.3 (c) refers to denoising result using the model-based grouping. As mentioned earlier, no visible difference can be observed although this image is characterized by the largest PSNR difference (after Phase 2), in favor of the model-based grouping, among the images that are presented in Table 7.2.

The original grouping method uses dissimilarities computed based on the transformed patches after shrinkage, such that the noise is reduced. On the other hand, the proposed grouping method achieves a comparable performance by using dissimilarities computed in the image domain, where the noisy patches are being used, as well as the statistical model that characterizes these dissimilarities. Consequently, we can conclude that the model-based grouping can serve as a simpler grouping alternative to the basic estimate grouping.

Image	σ_n	BM3D Grouping Phase 1 Output PSNR [dB]/SSIM	Model-Based Grouping Phase 1 Output PSNR [dB]/SSIM	BM3D Grouping Final Output PSNR [dB]/SSIM	Model-Based Grouping Final Output PSNR [dB]/SSIM
Baboon	20	25.83/0.77	25.86/0.77	26.2/0.79	26.2/0.79
Barbara	20	29.73/0.88	29.76/0.88	30.29/0.9	30.29/0.9
Barbara	30	27.66/0.82	27.63/0.82	28.24/0.85	28.21/0.85
Peppers	20	30.89/0.9	30.99/0.9	31.46/0.92	31.5/0.92
Peppers	30	28.56/0.85	28.6/0.85	29.29/0.88	29.32/0.88

Table 7.2.: Quantitative comparison of denoising results following BM3D Phase 1 and Phase 2, using two different grouping approaches for Phase 1: original Basic Estimate and Model-Based.

(a) (b) (c)

Figure 7.3.: Visual comparison between the denoising results of the two grouping methods in Phase 1 of the BM3D algorithm, based on the image Peppers with $\sigma_n = 20$. (a) Noisy image. (b) Final output using the Basic Estimate grouping in Phase 1. (c) Final output using the Model-Based grouping in Phase 1.

7.4. Chapter Summary

This chapter overviews the BM3D denoising method that is considered to be the state-of-the-art denoising approach. It consists of two main phases, each of them having three stages: Grouping, Collaborative Filtering and Aggregation. We suggest to replace the grouping method in the first phase, i.e., in the basic estimate phase, which compares patch dissimilarity in the transform domain, by the simpler model-based approach that compares dissimilarities in the image domain. The proposed grouping method is parameter free, it saves computations since is does not require patch transformation, and was found to provide similar denoising results compared to the original basic estimate grouping approach, both quantitatively, using the PSNR and SSIM measures, and visually, both after Phase 1 and after Phase 2 of the BM3D algorithm.

8. Model-Based Adaptive Approach and The Poisson Noise Model

8.1. Introduction

Many imaging devices, such as digital cameras and any device equipped with a CCD or a CMOS sensor, capture images by successive photon-to-electron, electron-to-voltage, and voltage-to-digit conversion. The two predominant sources of noise in digital image acquisition are the stochastic nature of the photon-counting process at the detectors and the intrinsic thermal and electronic fluctuations of the acquisition devices. Under standard illumination conditions, the second source of noise, which is signal-independent, is stronger than the first one. This motivates the usual Additive White Gaussian Noise (AWGN) assumption. However, in many applications such as fluorescence microscopy, astronomy, and endoscopy, only a few photons are collected by the photo-sensors, due to various physical constraints (low-power light source, short exposure time etc.). Under these imaging conditions, the major source of noise is strongly signal-dependent. Consequently, it is more reasonable to model the output of the detectors as a Poisson-distributed random vector.

The pixels in the sensor are arranged in a Bayer mosaic pattern [30], such that for each 2×2 set of pixels, two diagonally opposed pixels have green filters, and the other two have red and blue filters. Since G (stands for Green) carries most of the luminance information, its

sampling rate is twice that of R (Red) and B (Blue). Demosaicing [30] is the problem of interpolating back the image captured with a Bayer pattern, so that every pixel in the sensor is associated with a full RGB value.

In this chapter, we consider the model of signal-dependent noise and do not relate to any signal independent noise that may be added to it [27], i.e., we refer to the purely Poisson random noise. We explain how images corrupted with signal-dependent noise can be denoised by removing this signal-dependency and converting the noise model to be additive Gaussian with known variance. Once the Poisson noise has been transformed to Gaussian noise, the NLM denoising approach can be used, and we can explore the performance of the model-based approach on this type of images. We will not elaborate here on the demosaicing process, and refer the reader to [30] for more information.

8.2. Noise Model

Let Y_i be an observed pixel value obtained through an image acquisition device. Each Y_i is considered to be an independent random Poisson variable whose mean $X_i \geq 0$ is the underlying intensity value to be estimated. Explicitly, the discrete Poisson probability of each Y_i is obtained by:

$$P(Y_i|X_i) = \frac{X_i^{-Y_i} e^{-Y_i}}{Y_i!} \tag{8.1}$$

In addition to being the mean of the Poisson variable Y_i , the parameter X_i is also its variance:

$$\mathrm{E}[Y_i|X_i] = \mathrm{Var}[Y_i|X_i] = X_i \tag{8.2}$$

Poisson noise can be formally defined as:

$$N_i = Y_i - \mathrm{E}[Y_i|X_i] \tag{8.3}$$

Consequently, the following is trivially obtained: $\mathrm{E}[N_i|X_i] = 0$ and $\mathrm{Var}[N_i|X_i] = \mathrm{Var}[Y_i|X_i] = X_i$. Since the noise variance depends on the true intensity value X_i, Poisson noise is signal-dependent. More specifically, the standard deviation of the noise at pixel i equals $\sqrt{X_i}$. Due to this behavior, the effect of Poisson noise increases, i.e., the signal-to-noise ratio decreases, as the intensity value decreases: $SNR_i = \frac{X_i^2}{X_i} = X_i$.

8.3. Anscombe Transform - Variance Stabilizing Transform (VST)

Researchers (e.g. [42, 22]) suggest to apply image denoising techniques directly on the noisy image. These specific denoising methods are suitable for the *Poisson* noise model. However, the NLM denoising approach is originally intended for a Gaussian noise model. Hence, if we wish to use the NLM for Poisson denoising, a transform that converts the Poisson noise to a Gaussian noise is required. This is the Anscombe transform [43] that is described herein.

In applied statistics, a variance-stabilizing transformation (VST) is a data transformation that is specifically chosen to allow the application of analysis of variance techniques. The aim behind the choice of a variance-stabilizing transformation is to find a simple function to be applied on a given data set, such that the variance of the transformed data values is not related to their mean value. In this manner, the data-dependence of the noise variance is removed, so that it becomes constant throughout the whole data Y. Moreover, if the transformation is also normalizing, i.e., it results in a *Normal (*Gaussian*)* noise distribution, the

intensity values X_i can be estimated with a conventional denoising method designed for additive white Gaussian noise. In practice, neither exact stabilization nor exact normalization are possible [13], therefore, approximate or asymptotical results are employed. One of the most popular variance-stabilizing transformations is the Anscombe transformation [26]:

$$f(y) = 2\sqrt{y + \frac{3}{8}} \tag{8.4}$$

Applying eqn. (8.4) to Poisson distributed data gives a signal whose noise is asymptotically additive standard *Normal*. In order to understand why a VST transforms a Poisson distributed data to a Normally distributed data, we refer the reader to [7, 47].

The denoising of $f(y)$ produces a signal $\hat{f}(y)$ that can be considered as an estimate of $\mathrm{E}[f(y)|x]$, where x is the noise-free signal. In order to obtain the desired estimate of x, an inverse transformation should be applied on $\hat{f}(y)$. The *direct* algebraic inverse of eqn. (8.4) is:

$$T_{Direct}\left(\hat{f}(y)\right) = f^{-1}\left(\hat{f}(y)\right) = \left(\frac{\hat{f}(y)}{2}\right)^2 - \frac{3}{8} \tag{8.5}$$

However, the resulting estimate of x is biased, because the non-linearity of the transformation f means that generally $\mathrm{E}[f(y)|x] \neq f(\mathrm{E}[y|x])$, and thus $f^{-1}(\mathrm{E}[f(y)|x]) \neq \mathrm{E}[y|x]$. Another possibility is to use the adjusted inverse [26] that provides *asymptotically unbiased* solution for large counts:

$$T_{Asym-Unbiased}\left(\hat{f}(y)\right) = \left(\frac{\hat{f}(y)}{2}\right)^2 - \frac{1}{8} \tag{8.6}$$

Since the bias is especially significant for low counts, as shown in [26], this solution is not appropriate for these counts. The problem of bias can be solved by finding the *exact unbiased* inverse that maps $\mathrm{E}[f(y)|x]$ to $\mathrm{E}[y|x]$, where $\mathrm{E}[y|x] = x \geq 0$ is the noise-free value of the pixel. This involves computing the infinite sum (see eqns. (8.1), (8.4)) $\mathrm{E}[f(y)|x] =$

$\sum_{y=0}^{\infty} f(y) P(y|x) = 2\sum_{y=0}^{\infty} \left(\sqrt{y+\frac{3}{8}} \frac{x^y e^{-x}}{y!} \right)$ that can be interpreted as a maximum likelihood inverse [26]. Makitalo et al. [28] suggested a closed-form expression that approximates the exact unbiased inverse transform as:

$$T_{Exact-Unbiased-Closed}\left(\hat{f}(y)\right) = max(0, \frac{1}{4}\hat{f}(y)^2 + \frac{1}{4}\sqrt{\frac{3}{2}}\hat{f}(y)^{-1} - \frac{11}{8}\hat{f}(y)^{-2} + \frac{5}{8}\sqrt{\frac{3}{2}}\hat{f}(y)^{-3} - \frac{1}{8}) \tag{8.7}$$

In low count Poisson image denoising, the results obtained by using eqn. (8.7) and a state-of-the-art Gaussian denoising algorithm are dramatically better than those obtained with the asymptotically unbiased inverse (eqn. (8.6)), and better than what is achieved with currently existing methods specifically designed for Poisson denoising [26].

To conclude, given a noisy image with Poisson noise, the denoising flow consists of the following stages:

1. Apply forward Anscombe transform based on eqn. (8.4).
2. Denoise the noisy image using a AGWN denoising method, assuming $\sigma_n = 1$.
3. Apply the exact unbiased inverse Anscombe transform based on eqn. (8.7).

8.4. Comparison Between Standard NLM and Model-Based NLM

In a similar manner to the comparison conducted for Gaussian noise (sub-section 3.6.1), we wish to compare the denoising performance of the standard NLM, applied using either the Uniform or the Box patch-kernels, to the performance of the proposed adaptive model-based NLM, both with and without correlation consideration. Since the Poisson noise is proportional to pixel intensity, noisy images with different noise levels were created by

normalizing the noise-free image by its maximal intensity and multiplying the normalized image by a factor, denoted Q_{Poiss}, such that Q_{Poiss} is the highest intensity in the resultant image. Then, the noisy value of each pixel is drawn from a Poisson distribution whose parameters are set by the noise-free pixel value. Lower Q_{Poiss} implies on a noisier image with a lower SNR, due to lower pixel intensities ($SNR_i = X_i$,where X_i is the noise-free pixel value).

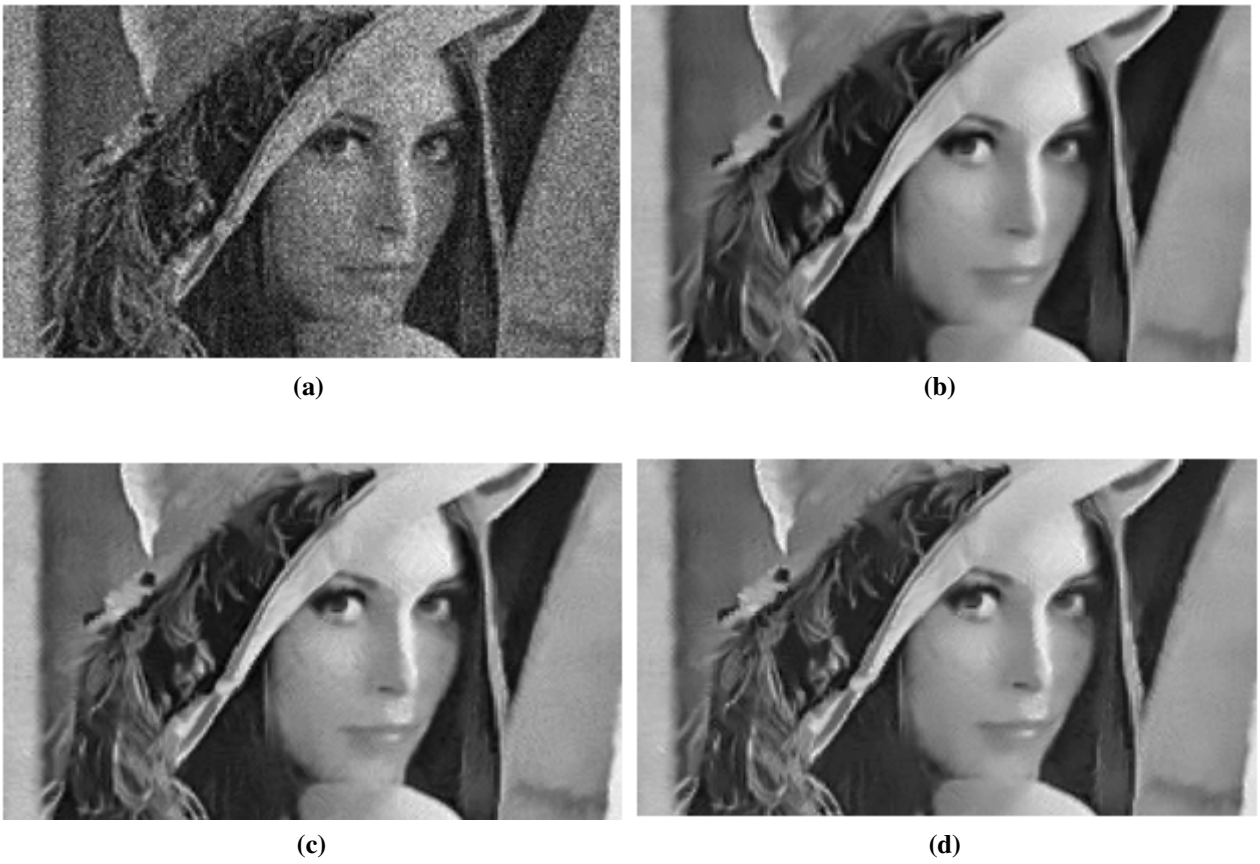
For the denoising process, same parameters were used for all the examined methods, i.e., a similarity patch of size 5×5 $(p = 5)$, a search region of size 11×11 $(M = 11)$ and $h = \sigma_n$. For the proposed adaptive model-based approach, we used $f = 0$ for the no-correlation method and $f = 2$ for the correlation-based method. An objective evaluation that uses the common measures PSNR and SSIM [41] was conducted. Table 8.1 summarizes the quantitative denoising results (objective evaluation) for different images with different noise levels. From its analysis, we can conclude that the proposed model-based approach, with and without correlation consideration, obtains somewhat higher PSNR and SSIM values than the standard NLM algorithm applied using either the Box or the Uniform patch-kernels, as was shown for the Gaussian noise model. The model-based approach that considers correlation has slightly higher PSNR than the approach that does not consider correlation (maximum 0.06 dB for the explored images and explored noise levels). The simulations also show that there is no clear visual distinction.

Image	Initial PSNR (Q_{Poiss})	Standard NLM w. Uniform	Standard NLM w. Box	Model-Based NLM no correlation	Model-Based NLM w. correlation
Lena	22.58 (100)	30.62/0.88	30.73/0.87	30.9/0.89	**30.96/0.89**
Lena	18.8 (50)	28.52/0.82	28.44/0.79	28.82/0.83	**28.84/0.83**
Barbara	22.27 (100)	29.17/0.87	29.25/0.87	29.35/0.88	**29.41/0.88**
Barbara	18.95 (50)	27.14/0.81	27.14/0.8	27.39/0.82	**27.43/0.82**
Baboon	22.32 (100)	25.28/0.72	26.08/0.77	26.15/0.77	**26.19/0.77**
Baboon	19 (50)	23.84/0.62	**24.39/0.67**	24.23/0.65	24.25/0.66
Peppers	22.67 (100)	30.66/0.88	30.7/0.88	30.82/0.89	**30.88/0.89**
Peppers	19.2 (50)	28.74/0.85	28.63/0.82	28.92/0.85	**28.95/0.85**

Table 8.1.: Quantitative comparison between the Standard NLM, applied with either the Uniform or the Box patch-kernels and the Proposed Model-Based NLM (ALGORITHM III) with and without correlation consideration.

A visual evaluation based on the perceived quality by a human observer is presented in Figures 8.1, 8.2. The figures present a zoom-in view of the images Baboon and Lena and compare three denoising methods: Standard NLM applied using the Uniform kernel, Standard NLM applied using the Box kernel, and the Proposed model-based NLM with correlation consideration. We have chosen to display only the results of the correlation-based approach and not also the results of model-based NLM without correlation consideration because, as mentioned earlier, there is no noticeable difference between the resultant images and the former approach provides slightly higher PSNR than the latter. Figures 8.1 (b) that corresponds to NLM with Uniform kernel is over-smoothed in textured areas, as the hat's feathers. Figure 8.1 (c) that corresponds to NLM with Box kernel preserves the feathers texture, however smooth regions, such as Lena's face appear grainy. In Figure 8.1 (d) that corresponds to ALGORITHM III (NLM with a model-based search region and an adaptive patch-kernel type), introduced in Chapter 4, with correlation consideration, both texture and smoothness are preserved. The same tendency can be concluded for Figure 8.2, that presents a zoom-in view of the eyes, nose and fur of the image Baboon. NLM with Uniform kernel (Figure 8.2 (b)) over-smoothes the fur and the eyes, whereas NLM with Box kernel preserves texture, but also appears grainy in smooth regions, like the nose. In

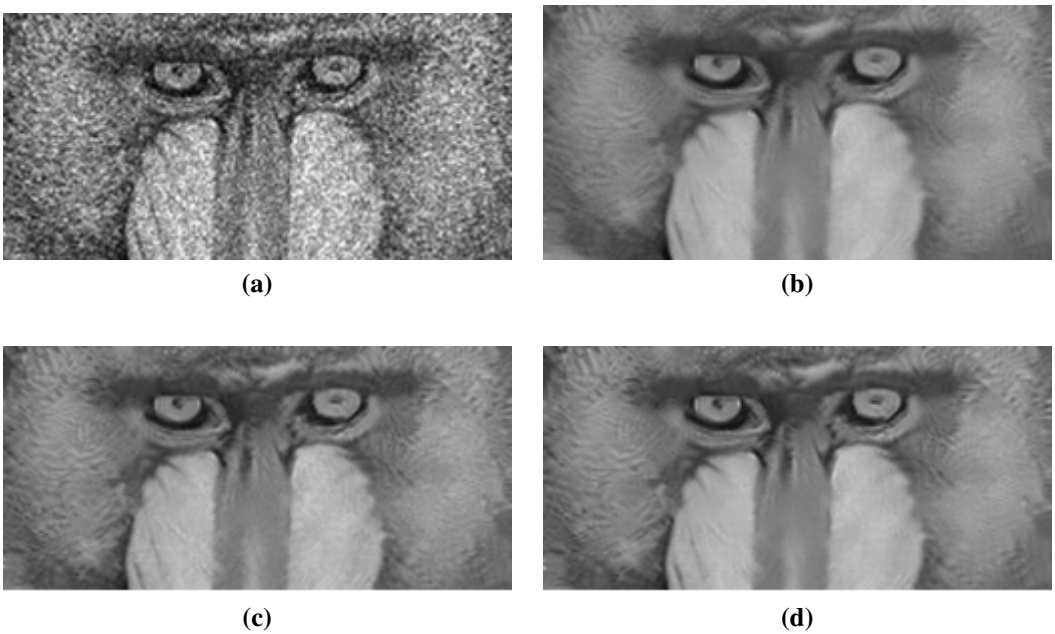
Figure 8.2 (d) that corresponds to ALGORITHM III with correlation consideration texture in the eyes and fur as well as nose smoothness are preserved.



(a) **(b)**

(c) **(d)**

Figure 8.1.: Denoising results of the image Lena (256×256) with Poisson noise, $Q_{Poiss} = 100$, $p = 5$, $M = 11$, $h = \sigma_n$. A zoom-in view of the eyes and hat feathers. (a) Cropped noisy image. (b) Denoised image using Standard NLM with a Uniform patch-kernel. (c) Denoised image using Standard NLM with a Box patch-kernel. (d) Denoised image using ALGORITHM III with correlation consideration.



(a) **(b)**

(c) **(d)**

Figure 8.2.: Denoising results of the image Baboon (204×204) with Poisson noise, $Q_{Poiss} = 50$, $p = 5$, $M = 11$, $h = \sigma_n$. A zoom-in view of the eyes, nose and fur. (a) Cropped noisy image. (b) Denoised image using Standard NLM with a Uniform patch-kernel. (c) Denoised image using Standard NLM with a Box patch-kernel. (d) Denoised image using ALGORITHM III with correlation consideration.

8.5. Comparison Between Standard BM3D and Model-Based BM3D

In a similar manner to the comparison conducted for Gaussian noise (section 7.3), we wish to compare the denoising performance of the standard BM3D after both Phase 1 and Phase 2 to the model-based BM3D after each of these two phases, when the input image is characterized by a Poisson noise model. We remind the reader that the difference between the two approaches stems from the different grouping method of Phase 1. The standard BM3D Phase 1 grouping creates 3D groups by applying a hard-thresholding operator to the patches dissimilarity in the transform domain, whereas the model-based grouping uses

ALGORITHM I with the Uniform patch-kernel, i.e., sets a model-based variance threshold on patches dissimilarity in the image domain. We have also explored the added-value of the correlation-based model to the BM3D grouping by comparing the standard BM3D grouping in Phase 1 to the model-based grouping applied with either the correlation-based threshold, i.e., eqn. (5.15) with $f = 2$ or the no-correlation threshold, i.e., eqn. (3.21) with $f = 0$.

As in the previous section, noisy images with different noise levels were created by normalizing the noise-free image by its maximal intensity, multiplying the normalized image by a factor, denoted Q_{Poiss}. The noisy value of each pixel is drawn from a Poisson distribution whose parameters are set by the noise-free pixel value. For the denoising process, same parameters were used for all the examined methods, i.e., a similarity patch of size 8×8, and a search region of size 39×39. An objective evaluation that uses the common measures PSNR and SSIM [41] was conducted. Tables 8.2, 8.3 summarize the quantitative denoising results, for Phase 1 and Phase 2, for different images with different noise levels . From their analysis, we can conclude that the model-based BM3D is comparable to the standard BM3D with or without correlation consideration. Consequently, we recommend the user to use the model without correlation consideration. The PSNR differences between the three explored methods are negligible, and no visual difference is noticeable.

Image	Initial PSNR (Q_{Poiss})	Standard BM3D	Model-Based BM3D no correlation	Model-Based BM3D w. correlation
Lena	22.46 (100)	31.47/0.9	31.43/0.9	31.4/0.9
Lena	18.84 (50)	29.31/0.85	29.31/0.85	29.31/0.85
Barbara	22.23 (100)	29.8/0.89	29.83/0.89	29.81/0.89
Barbara	18.93 (50)	27.67/0.83	27.7/0.83	27.7/0.83
Baboon	22.99 (100)	26.31/0.79	26.32/0.79	26.3/0.79
Baboon	19.72 (50)	24.57/0.69	24.59/0.69	24.59/0.69

Table 8.2.: Quantitative comparison between the Standard BM3D and the Model-Based BM3D, applied with either the correlation-based threshold and no-correlation threshold. All explored methods are explored after **Phase 1** of the BM3D algorithm.

Image	Initial PSNR (Q_{Poiss})	Standard BM3D	Model-Based BM3D no correlation	Model-Based BM3D w. correlation
Lena	22.46 (100)	32.08/0.92	32.07/0.92	32.07/0.92
Lena	18.84 (50)	29.99/0.88	29.959/0.88	29.95/0.88
Barbara	22.23 (100)	30.24/0.91	30.26/0.91	30.25/0.91
Barbara	18.93 (50)	28.32/0.86	28.34/0.86	28.34/0.86
Baboon	22.99 (100)	27.1/0.83	27.09/0.83	27.08/0.83
Baboon	19.72 (50)	24.72/0.71	24.72/0.7	24.72/0.7

Table 8.3.: Quantitative comparison between the Standard BM3D and the Model-Based BM3D, applied with either the correlation-based threshold and no-correlation threshold. All explored methods are explored after **Phase 2** of the BM3D algorithm.

8.6. Chapter Summary

In this chapter, we have introduced the intensity-dependent Poisson noise model that characterizes images taken by a digital camera. We have explained how the common denoising algorithms that are used to handle additive Gaussian noise can be exploited to deal with this type of noise, by utilizing the Anscombe transform. Finally, we have compared the denoising performance of ALGORITHM III, applied with and without correlation consideration, to the standard NLM. This algorithm that was introduced in Chapter 4, sets a model-based search region and an adaptive patch-kernel type per pixel. Similar comparison was conducted for the BM3D denoising method and the two Phase 1 grouping approaches. We have concluded that the performance tendency obtained for Gaussian noise is preserved also for the Poisson noise, i.e., the model-based approach is somewhat better both objectively and subjectively than the standard NLM. As for the BM3D method, the model-based BM3D saves computations and remains comparable to the standard BM3D.

9. Conclusion

9.1. Summary

In this thesis, we have explored the Non-Local Means denoising method. This method, inspired by neighborhood filters [14], takes advantage of the high degree of redundancy in any natural image by assuming that every small patch in a natural image has many similar patches in the same image. The NLM estimate a POI by using a weighted average of pixels located in a search region associated with that POI. The weights are exponential terms that are inversely proportional to the dissimilarity between a small neighborhood of the POI and a corresponding small neighborhood of pixels within the search region. This method assumes stationarity of the search region data and depends on five parameters whose values affect the denoising performance. These parameters are the size of the similarity patch (p), the weight-smoothing parameter (h), the size of the search region (M), the weight of the central pixel $(w_{i,i})$, and the patch-kernel type.

The search region is usually a rectangular neighborhood, centered at the POI, which may include pixels whose original gray value do not match the value of the original central pixel. Consequently, their participation in the weighted averaging process degrades denoising performance. To eliminate their effect, researchers (e.g., [32, 37]) suggest creating an adaptive search-region which excludes those dissimilar pixels, such that the initial search region is segmented into two sets: a set of similar pixels to the POI and a complementary set of dis-

similar pixels. In this thesis we present a novel adaptive model-based method, which defines a set of similar pixels to the POI, from the initial search region, using the statistical distribution of the dissimilarity measure. This proposed approach does not restrict the search region to be contiguous, an important quality for textured regions. We further improve the denoising performance by using an adaptive patch-kernel that is set according to the cardinality of the adaptive search region, denoted $\left|S_i^S\right|$. This combined approach was compared to the approach that uses an adaptive search region with a single patch-kernel type, as well as to the standard NLM and was found to provide better denoising results both quantitatively (PSNR and SSIM-wise) and qualitatively (visually). As for computational complexity, the running time of the proposed appraoch is increased by 14% on average with respect to the standard NLM, applied using either the Uniform or the Box patch-kernels. Another comparison that was conducted is to the adaptive search region size approach [17], which is much more complicated than our proposed approach and provides comparable denoising results.

The above-mentioned model-based scheme does not relate to the correlation between the dissimilarity elements of a given search region. As shown in Chapter 5, there are three possible sources of correlation that should be referred to. The first source is due to the comparison to the same reference patch during dissimilarity computation, the second origin is due to patches overlap with each other, and the third origin is due to patches overlap with the reference patch. The first source of correlation provides the main contribution to the model. The estimated variance of the normalized dissimilarities that are associated with a given adaptive search region, is decreased, as expected, when correlation is considered. In order to compensate for variance error computation caused by using a small sample size, we have added to the variance threshold of ALGORITHM III a variance correction term. This correction term is the variance of the estimated variance variable, which is sample size dependent, multiplied by a factor f (see section 5.4). The correction causes a slight reduction in the estimated variance threshold with respect to the case where no correlation is considered. The added value of this correction term was also explored for the no correlation case

and was found to be redundant there since the estimated variance threshold was high enough to include all the relevant pixels. We have compared the performance of the model-based search region created with no correlation consideration to that of the model-based search region created with correlation consideration and the correction factor. Simulations suggest that there is no significant difference between these two modification, thus the scheme that does not consider correlation is preferred due to its reduced computational complexity.

In addition, we have suggested to exploit the model-based adaptive search method in the BM3D Phase 1 (Basic Estimate) grouping. The original Phase 1 grouping methodology, which compares patch dissimilarity in the transform domain, is replaced by a simpler approach that compares dissimilarities in the image domain. The proposed grouping method is parameter free, it saves computations (11% for the Phase1 grouping step and 4.5% for the overall scheme) since is does not require patch transformation, and was found to provide similar denoising results compared to the original basic estimate grouping approach, both quantitatively and visually.

Finally, we refer to the intensity-dependent Poisson noise model that characterizes images taken by a digital camera. The common denoising algorithms that are used to handle additive Gaussian noise can be exploited to deal with this type of noise, by utilizing the Anscombe transform. We have compared the denoising performance of our proposed model-based approach applied with and without correlation consideration to the standard NLM. Similar comparison was conducted for the BM3D denoising method and the two Phase 1 grouping approaches. We have concluded that the performance tendency obtained for Gaussian noise is preserved also for the Poisson noise, i.e., the model-based approach is somewhat better both objectively and subjectively than standard NLM. As for the BM3D method, the model-based BM3D saves computations and remains comparable to the standard BM3D.

9.2. Future Work

As discussed in Chapter 5, there are three sources of correlation between the dissimilarities of a given search region. We have derived the final estimated variance expression only for the first source that is due to the comparison to the same reference patch. The two other sources are characterized by complicated terms of covariance matrices due to patch overlap terms, thus analyzing the variance of the estimated variance is not an easy task. If we wish to be more precise, we should add the contribution of patches overlap to the variance of the estimated variance that is used in the computation of the adaptive search region. This analysis may be a part of a future research.

Another topic that can be explored is the expansion of the model-based image denoising technique to video denoising. In this case, the denoising of each pixel can be based on both the defined search region of the given frame at time t and of the previous frame at time $t-1$. In this manner, the adaptive search region can be expanded, such that more pixels are considered in the weighted averaging process. One should explore whether it improves the denoising performance.

When dealing with Poisson noise, other variance stabilizing transforms (VST) can be explored besides the Anscombe transform. Maybe an alternative VST combined with the proposed model-based approach will provide better denoising results than the commonly used Anscombe.

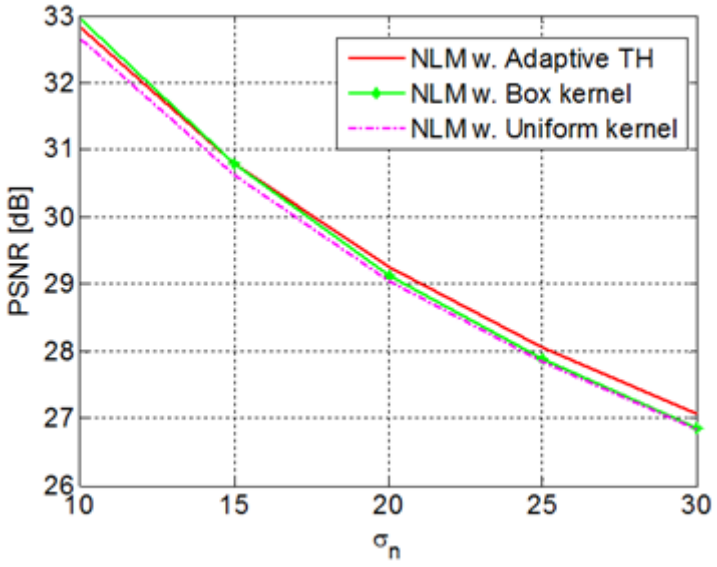
Currently, two types of patch-kernels are used. One may try to find a better kernel that suits the "non-smooth" regions and explore its performance using the proposed adaptive model-based scheme compared to the commonly used Box patch-kernel. A suggestion of such a kernel optimization may involove finding the most suitable slope of the Box kernel, see Figure 2.1.

Finally, color components can be added to the model. Instead of using only intensity-based dissimilarities, one can add color information, such that the adaptive search region is based

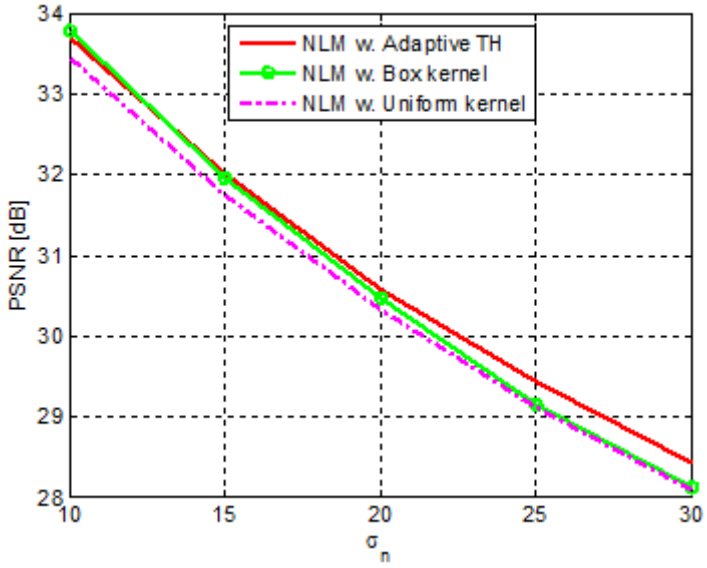
on a measure that is the result of three dissimilarity values for each pixel in the corresponding search region. Other color spaces, such as Lab [50], can be explored as well.

A. Additional Examples Of Comparison Between The Standard NLM and The Model-Based NLM

This appendix presents additional graphs that demonstrate the added value of the Model-based approach (refer to Chapter 3) applied with NLM denoising, compared to the standard NLM approach, as a continuation to the example that is presented in sub-section 3.6.2. Figure A1.1 illustrates the improvement of the suggested approach PSNR-wise for different noise levels. For all the explored cases, same NLM parameters were used for denoising, as detailed in the figure.



(a)



(b)

Figure A.1.: A comparison between the three explored NLM approaches, i.e., standard NLM with either the Uniform patch-kernel or the Box patch-kernel, and our Model-Based approach, for different images. The comparison is conducted for different noise levels with the following NLM parameters: $p = 5$, $M = 11$, $h = \sigma_n$. (a) Barbara (256×256), (b) Peppers (256×256).

B. Distribution of Normalized Dissimilarities For Different Cases of Patches Overlap

In this appendix we develop the cross-variance terms of the normalized dissimilarity values, associated with pixels included in the S_i^S set. We refer to three sources of correlation, as discussed in Chapter 5, starting from the simplest Case 1, where similarity patches do not overlap at all, and the correlation between dissimilarities is due to the comparison to a mutual reference patch. Then, we add another degree of complexity by allowing patches to overlap each other, but not the reference patch (Case 2). Finally, we refer to the most general case (Case 3), in which patches may overlap each other and/or the reference patch.

B.1. Case 1: Correlation Between Dissimilarities of Patches That Do Not Overlap

This case refers to the similarity patches A_j, A_k for $j,k \in S_i$ that satisfy the non-overlap criterion: $\forall j \neq k,\ j,k \in S_i^S$, $A_j \cap A_k = \emptyset$, $A_j \cap A_i = \emptyset$, $A_k \cap A_i = \emptyset$, i.e., the patches do not overlap each other, nor the reference patch A_i. By definition, the normalized dissimilarity elements, within a given S_i, share a mutual member, which is the reference patch. Consequently, these elements are correlated and their cross-variance term is developed herein for a General patch-kernel.

Before continuing with the cross-variance analysis, we wish to clarify the definition of global and local indices. We need to distinguish between a global index (with respect to the image), which represents the N_i values (based on the noise model where $Y_i = X_i + N_i$), and a local index (with respect to the similarity patch), which represents the patch-kernel coefficients $\alpha_s, s \in \left[1, p^2\right]$. Figure B.1 presents a global coordinate system of a selected similarity patch (marked in yellow), on the right, and its respective local coordinate system, on the left. For example, the third global coordinate corresponds to the first local coordinate, such that the kernel first coefficient α_1 corresponds to the third pixel value in the image N_3.

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

1	12	23	34	45	56	67	78	89	100	111
2	13	24	35	46	57	68	79	90	101	112
3	14	25	36	47	58	69	80	91	102	113
4	15	26	37	48	59	70	81	92	103	114
5	16	27	38	49	60	71	82	93	104	115
6	17	28	39	50	61	72	83	94	105	116
7	18	29	40	51	62	73	84	95	106	117
8	19	30	41	52	63	74	85	96	107	118
9	20	31	42	53	64	75	86	97	108	119
10	21	32	43	54	65	76	87	98	109	120
11	22	33	44	55	66	77	88	99	110	121

Figure B.1.: The grid on the right corresponds to global coordinates of the image. The yellow patch is a selected similarity patch of size 5×5. The right patch grid presents the patch global coordinates, whereas the left patch grid presents its local coordinates.

We begin with the distribution of the normalized dissimilarity variable, which was proven to be approximately *Normal* (see section 3.3) and use the definition for κ introduced in Chapter 3 (eqn. (3.9)):

$$\tilde{d}_i^G(j) = \frac{d_i^G(j)}{2\sigma_n^2} \sim \mathcal{N}(1, 2\kappa) \tag{B.1}$$

By the definition of cross-variance, the following applies for patches A_j, A_k for $j, k \in S_i^S$:

$$Cov\left[\tilde{d}_i^G(j),\tilde{d}_i^G(k)\right]=\mathrm{E}\left[\left(\tilde{d}_i^G(j)-\mathrm{E}\left[\tilde{d}_i^G(j)\right]\right)\cdot\left(\tilde{d}_i^G(k)-\mathrm{E}\left[\tilde{d}_i^G(k)\right]\right)\right]=$$
$$=\mathrm{E}\left[\left(\tilde{d}_i^G(j)-1\right)\cdot\left(\tilde{d}_i^G(k)-1\right)\right]=\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]-\mathrm{E}\left[\tilde{d}_i^G(j)\right]-\mathrm{E}\left[\tilde{d}_i^G(k)\right]+1=$$
$$=\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]-1$$

We need to analyze the term $\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]$, based on the definition of the normalized dissimilarity measure:

$$\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]=\mathrm{E}\left[\sum_{m\in A_i,f\in A_j,l_1\in\left[1,p^2\right]}\alpha_{l_1}\left(\frac{N_m-N_f}{\sqrt{2}\sigma_n}\right)^2\cdot\sum_{m\in A_i,s\in A_k,l_2\in\left[1,p^2\right]}\alpha_{l_2}\left(\frac{N_m-N_s}{\sqrt{2}\sigma_n}\right)^2\right]=$$
$$=\ \frac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{m\in A_i,f\in A_j,l_1\in\left[1,p^2\right]}\alpha_{l_1}\left(N_m-N_f\right)^2\cdot\sum_{m\in A_i,s\in A_k,l_2\in\left[1,p^2\right]}\alpha_{l_2}\left(N_m-N_s\right)^2\right]$$

Note: The cross-products $N_i\cdot N_j,\ \forall i\neq j$ and any other version of them will be neglected since their expectation is zero and they do not contribute to the sum.

$$\mathrm{E}\left[\tilde{d}_i^U(j)\cdot\tilde{d}_i^U(k)\right]=\frac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{m\in A_i,l\in\left[1,p^2\right]}\alpha_l^2N_m^4\right]+$$
$$+\frac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{m\in A_i,l_1\in\left[1,p^2\right]}\alpha_{l_1}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in\left[1,p^2\right]\setminus\{l_1\}}\alpha_{l_2}N_f^2\right)\right]+$$
$$+\frac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{m\in A_i,l_1\in\left[1,p^2\right]}\alpha_{l_1}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right]+$$
$$+\frac{1}{4\sigma_n^4}\mathrm{E}\left[\left(\sum_{l\in A_j,l_1\in\left[1,p^2\right]}\alpha_{l_1}N_l^2\right)\cdot\left(\sum_{m\in A_i,l_2\in\left[1,p^2\right]}\alpha_{l_2}N_m^2\right)\right]+$$
$$+\frac{1}{4\sigma_n^4}\mathrm{E}\left[\left(\sum_{l\in A_j,l_1\in\left[1,p^2\right]}\alpha_{l_1}N_l^2\right)\cdot\left(\sum_{s\in A_k,l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right]$$

Dealing with each addend separately and following the next rules:

1. Exchange the order of summation and expectation.
2. N_i are $i.i.d$, thus share the same distribution and $\mathrm{E}\left[N_i^2N_j^2\right]=\mathrm{E}\left[N_i^2\right]\cdot\mathrm{E}\left[N_j^2\right]=\sigma_n^4$
3. For $N_i\sim\mathcal{N}\left(0,\sigma_n^2\right)$, the 4^{th} moment is known to be: $\mathrm{E}\left[N_i^4\right]=3\sigma_n^4$

$$
\begin{aligned}
&(i)\ \mathrm{E}\left[\textstyle\sum_{m\in A_i, l\in[1,p^2]} \alpha_l^2 N_m^4\right] = \textstyle\sum_{m\in A_i, l\in[1,p^2]} \alpha_l^2 \mathrm{E}\left[N_m^4\right] = 3\sigma_n^4 \textstyle\sum_{l\in[1,p^2]} \alpha_l^2 \\
&(ii)\ \mathrm{E}\left[\textstyle\sum_{m\in A_i, l_1\in[1,p^2]} \alpha_{l_1} N_m^2 \cdot \left(\textstyle\sum_{f\in A_i\setminus\{m\}, l_2\in[1,p^2]\setminus\{l_1\}} \alpha_{l_2} N_f^2 + \textstyle\sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} N_s^2\right)\right] = \\
&\quad = \textstyle\sum_{m\in A_i, l_1\in[1,p^2]} \alpha_{l_1} \mathrm{E}\left[N_m^2\right] \cdot \left(\textstyle\sum_{f\in A_i\setminus\{m\}, l_2\in[1,p^2]\setminus\{l_1\}} \alpha_{l_2} \mathrm{E}\left[N_f^2\right] + \textstyle\sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} \mathrm{E}\left[N_s^2\right]\right) = \\
&\quad = \sigma_n^4 \textstyle\sum_{l_1\in[1,p^2]} \alpha_{l_1} \left(2 - \alpha_{l_1}\right) = \sigma_n^4 \left(2 - \textstyle\sum_{l_1\in[1,p^2]} \alpha_{l_1}^2\right) \\
&(iii)\ \mathrm{E}\left[\left(\textstyle\sum_{m\in A_j, l_1\in[1,p^2]} \alpha_{l_1} N_m^2\right) \cdot \left(\textstyle\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} N_f^2 + \textstyle\sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} N_s^2\right)\right] = \\
&\quad = \mathrm{E}\left[\textstyle\sum_{m\in A_j, l_1\in[1,p^2]} \alpha_{l_1} N_m^2\right] \cdot \mathrm{E}\left[\textstyle\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} N_f^2 + \textstyle\sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} N_s^2\right] = \\
&\quad = \left[\textstyle\sum_{m\in A_j, l_1\in[1,p^2]} \alpha_{l_1} \mathrm{E}\left[N_m^2\right]\right] \cdot \left[\textstyle\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} \mathrm{E}\left[N_f^2\right] + \textstyle\sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} \mathrm{E}\left[N_s^2\right]\right] = 2\sigma_n^4
\end{aligned}
$$

Finally, the cross-product expectation is obtained as:

$$
\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] = \frac{1}{4\sigma_n^4}\left(3\sigma_n^4 \sum_{l\in[1,p^2]} \alpha_l^2 + 2\sigma_n^4 - \sigma_n^4 \sum_{l\in[1,p^2]} \alpha_l^2 + 2\sigma_n^4\right) = 1 + \frac{\sum_{l\in[1,p^2]} \alpha_l^2}{2} = 1 + \frac{\kappa}{2}
$$

The cross-variance term for a General patch kernel is given by:

$$
\forall j,k \in S_i^S :\ \mathrm{Cov}\left[\tilde{d}_i^G(j), \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] - 1 = \frac{\kappa}{2} \tag{B.2}
$$

The covariance matrix for a General patch-kernel is obtained by:

$$
C_{\mathbf{d}_i^G} = E\left[\tilde{\mathbf{d}}_i^G \left(\tilde{\mathbf{d}}_i^G\right)^T\right] = \kappa \begin{bmatrix} 2 & 0.5 & .. & 0.5 \\ 0.5 & 2 & .. & 0.5 \\ | & | & | & | \\ 0.5 & 0.5 & .. & 2 \end{bmatrix}_{L\times L} \tag{B.3}
$$

For the Uniform patch kernel, we assign $\kappa = p^{-2}$.

B.2. Case 2: Correlation Between Dissimilarities of Patches That Overlap Each Other, But Not The Reference Patch

This case refers to the similarity patches A_j, A_k for $j,k \in S_i$ that satisfy the overlap criterion: $\forall j \neq k,\ j,k \in S_i^S,\ A_j \cap A_k \neq \emptyset,\ A_j \cap A_i = \emptyset,\ A_k \cap A_i = \emptyset$, i.e., the patches overlap each other, but not the reference patch A_i. As for the former case, the corresponding normalized dissimilarities are correlated by definition due to the mutual reference patch. Moreover, this case is characterized by another source of correlation that is the patch overlap. The cross-variance term is developed herein for a General patch-kernel.

We begin with the distribution of the normalized dissimilarity measure, which was proven to be approximately *Normal* (see eqn. (3.11)):

$$\tilde{d}_i^G(j) = \frac{d_i^G(j)}{2\sigma_n^2} \sim \mathcal{N}(1, 2\boldsymbol{\kappa}) \tag{B.4}$$

As was shown in the previous sub-section,the cross-variance for patches $A_j, A_k\ j,k \in S_i^S$ is defined as:

$$\mathrm{Cov}\left[\tilde{d}_i^G(j), \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] - 1$$

We need to analyze the term $\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right]$, based on the definition of the normalized dissimilarity measure:

$$\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\sum_{m\in A_i, f\in A_j, l_1\in[1,p^2]} \alpha_{l_1} \left(\frac{N_m - N_f}{\sqrt{2}\sigma_n}\right)^2 \cdot \sum_{m\in A_i, s\in A_k, l_2\in[1,p^2]} \alpha_{l_2} \left(\frac{N_m - N_s}{\sqrt{2}\sigma_n}\right)^2\right] =$$
$$= \frac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{m\in A_i, f\in A_j, l_1\in[1,p^2]} \alpha_{l_1} \left(N_m - N_f\right)^2 \cdot \sum_{m\in A_i, s\in A_k, l_2\in[1,p^2]} \alpha_{l_2} \left(N_m - N_s\right)\right]$$

Remarks:

1. The cross-products $N_i \cdot N_j\ \forall i \neq j$ and any other version of them will be neglected since their expectation is zero and they do not contribute to the sum.

2. The term $l \in A_j(m)$ refers to the patch local index that corresponds to the global index m in the similarity patch A_j.

3. The set of global indices included in the region of patch overlap is defined as: $O_{j,k} = A_j \cap A_k = \left\{ m \,|\, m \in A_j \cap A_k \right\}$.

$$
\begin{aligned}
&\mathrm{E}\left[\tilde{d_i^G}(j) \cdot \tilde{d_i^G}(k)\right] = \tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{\ m \in A_i, l \in \left[1,p^2\right]\ } \alpha_l^2 N_m^4\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{\ m \in A_i, l_1 \in \left[1,p^2\right]\ } \alpha_{l_1} N_m^2 \cdot \left(\sum_{\ f \in A_i \setminus \{m\}, l_2 \in \left[1,p^2\right] \setminus \{l_1\}\ } \alpha_{l_2} N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\sum_{\ m \in A_i, l_1 \in \left[1,p^2\right]\ } \alpha_{l_1} N_m^2 \cdot \left(\sum_{\ s \in A_k, l_3 \in \left[1,p^2\right]\ } \alpha_{l_3} N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\left(\textstyle\sum_{m \in A_j \setminus \left\{O_{j,k}\right\}, l_1 \in \left[1,p^2\right] \setminus A_j(m)} \alpha_{l_1} N_m^2\right) \cdot \left(\textstyle\sum_{f \in A_i, l_2 \in \left[1,p^2\right]} \alpha_{l_2} N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\left(\textstyle\sum_{m \in A_j \setminus \left\{O_{j,k}\right\}, l_1 \in \left[1,p^2\right] \setminus A_j(m)} \alpha_{l_1} N_m^2\right) \cdot \left(\textstyle\sum_{s \in A_k, l_3 \in \left[1,p^2\right]} \alpha_{l_3} N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m \in O_{j,k}, l_1 \in A_j(m)} \alpha_{l_1} N_m^2 \cdot \left(\textstyle\sum_{f \in A_i, l_2 \in \left[1,p^2\right]} \alpha_{l_2} N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m \in O_{j,k}, l_1 \in A_j(m)} \alpha_{l_1} N_m^2 \cdot \left(\textstyle\sum_{s \in A_k \setminus \{m\}, l_3 \in \left[1,p^2\right] \setminus A_k(m)} \alpha_{l_3} N_s^2 + \alpha_{A_k(m)} N_m^2\right)\right]
\end{aligned}
$$

Dealing with each addend separately and following the next rules:

1. Substitute the order of summation and expectation.
2. N_i are $i.i.d$, thus share the same distribution and $\mathrm{E}\left[N_i^2 N_j^2\right] = \mathrm{E}\left[N_i^2\right] \cdot \mathrm{E}\left[N_j^2\right] = \sigma_n^4$
3. For $N_i \sim \mathcal{N}\left(0, \sigma_n^2\right)$, the 4^{th} moment is defined as: $\mathrm{E}\left[N_i^4\right] = 3\sigma_n^4$

$$
\begin{aligned}
&(i)\ \ \mathrm{E}\left[\textstyle\sum_{m \in A_i, l \in \left[1,p^2\right]} \alpha_l^2 N_m^4\right] = \textstyle\sum_{m \in A_i, l \in \left[1,p^2\right]} \alpha_l^2 \mathrm{E}\left[N_m^4\right] = 3\sigma_n^4 \textstyle\sum_{l \in \left[1,p^2\right]} \alpha_1^2 \\
&(ii)\ \ \mathrm{E}\left[\textstyle\sum_{m \in A_i, l_1 \in \left[1,p^2\right]} \alpha_{l_1} N_m^2 \cdot \left(\textstyle\sum_{f \in A_i \setminus \{m\}, l_2 \in \left[1,p^2\right] \setminus \{l_1\}} \alpha_{l_2} N_f^2\right)\right] + \\
&+\mathrm{E}\left[\textstyle\sum_{m \in A_i, l_1 \in \left[1,p^2\right]} \alpha_{l_1} N_m^2 \cdot \left(\textstyle\sum_{s \in A_k, l_3 \in \left[1,p^2\right]} \alpha_{l_3} N_s^2\right)\right] = \\
&= \textstyle\sum_{m \in A_i, l_1 \in \left[1,p^2\right]} \alpha_{l_1} \mathrm{E}\left[N_m^2\right] \cdot \left(\textstyle\sum_{f \in A_i \setminus \{m\}, l_2 \in \left[1,p^2\right] \setminus \{l_1\}} \alpha_{l_2} \mathrm{E}\left[N_f^2\right]\right) + \\
&+\textstyle\sum_{m \in A_i, l_1 \in \left[1,p^2\right]} \alpha_{l_1} \mathrm{E}\left[N_m^2\right] \cdot \left(\textstyle\sum_{s \in A_k, l_3 \in \left[1,p^2\right]} \alpha_{l_3} \mathrm{E}\left[N_s^2\right]\right) =
\end{aligned}
$$

$$= \sigma_n^4 \sum_{m\in A_i, 1_1\in[1,p^2]} \alpha_{l_1} \cdot \left(\sum_{l_2\in[1,p^2]} \alpha_{l_2} - \alpha_{l_1} + \sum_{l_3\in[1,p^2]} \alpha_{l_3}\right) =$$

$$= \sigma_n^4 \sum_{1_1\in[1,p^2]} \alpha_{l_1} (2-\alpha_{l_1}) = \sigma_n^4 \left(2 - \sum_{l_1\in[1,p^2]} \alpha_{1_1}^2\right)$$

$$(iii)\ \mathrm{E}\left[\left(\sum_{m\in A_j\setminus\{O_{j,k}\}, l_1\in[1,p^2]\setminus A_j(m)} \alpha_{l_1} N_m^2\right) \cdot \left(\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} N_f^2 + \sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} N_s^2\right)\right] =$$

$$= \mathrm{E}\left[\sum_{m\in A_j\setminus\{O_{j,k}\}, l_1\in[1,p^2]\setminus A_j(m)} \alpha_{l_1} N_m^2\right] \cdot \mathrm{E}\left[\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} N_f^2 + \sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} N_s^2\right] =$$

$$= \left[\sum_{m\in A_j\setminus\{O_{j,k}\}, l_1\in[1,p^2]\setminus A_j(m)} \alpha_{l_1} \mathrm{E}\left[N_m^2\right]\right] \cdot \left[\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} \mathrm{E}\left[N_f^2\right] + \sum_{s\in A_k, l_3\in[1,p^2]} \alpha_{l_3} \mathrm{E}\left[N_s^2\right]\right] =$$

$$= \sigma_n^4 \left[\sum_{m\in A_j\setminus\{O_{j,k}\}, l_1\in[1,p^2]\setminus A_j(m)} \alpha_{l_1}\right] \cdot \left[\sum_{l_2\in[1,p^2]} \alpha_{l_2} + \sum_{l_3\in[1,p^2]} \alpha_{l_3}\right] =$$

$$= 2\sigma_n^4 \left(1 - \sum_{m\in O_{j,k}} \alpha_{A_j(m)}\right)$$

$$(iv)\ \mathrm{E}\left[\sum_{m\in O_{j,k}} \alpha_{A_j(m)} N_m^2 \cdot \left(\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} N_f^2 + \sum_{s\in A_k\setminus\{m\}, l_3\in[1,p^2]\setminus A_k(m)} \alpha_{l_3} N_s^2 + \alpha_{A_k(m)} N_m^2\right)\right] =$$

$$= \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \mathrm{E}\left[N_m^2\right] \cdot \left(\sum_{f\in A_i, l_2\in[1,p^2]} \alpha_{l_2} \mathrm{E}\left[N_f^2\right] + \sum_{s\in A_k\setminus\{m\}, l_3\in[1,p^2]\setminus A_k(m)} \alpha_{l_3} \mathrm{E}\left[N_s^2\right]\right) +$$

$$+ \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \alpha_{A_k(m)} \mathrm{E}\left[N_m^4\right] = \sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \cdot \left(\sum_{l_2\in[1,p^2]} \alpha_{l_2}\right) =$$

$$+ \sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \cdot \left(\sum_{l_3\in[1,p^2]\setminus A_k(m)} \alpha_{l_3} + 3\alpha_{A_k(m)}\right) =$$

$$= \sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \cdot \left(2 + 2\alpha_{A_k(m)}\right) = 2\sigma_n^4 \left(\sum_{m\in O_{j,k}} \alpha_{A_j(m)} + \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \alpha_{A_k(m)}\right)$$

Finally, the cross-product expectation is defined as:

$$\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] = \frac{1}{4\sigma_n^4} \left(3\sigma_n^4 \sum_{l\in[1,p^2]} \alpha_l^2 + 2\sigma_n^4 - \sigma_n^4 \sum_{l_1\in[1,p^2]} \alpha_{1_1}^2 + 2\sigma_n^4\right) +$$

$$+ \frac{1}{4\sigma_n^4} \left(-2\sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} + 2\sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} + 2\sigma_n^4 \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \alpha_{A_k(m)}\right) =$$

$$= \frac{1}{2} \left(\sum_{l_1\in[1,p^2]} \alpha_{1_1}^2 + 2 + \sum_{m\in O_{j,k}} \alpha_{A_j(m)} \alpha_{A_k(m)}\right)$$

The cross-variance term for a General patch kernel is defined as:

$$\forall j,k \in S_i^S: \quad \mathrm{Cov}\left[\tilde{d}_i^G(j), \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] - 1 = \frac{\kappa}{2} + \frac{\sum_{m\in O_{j,k}} \alpha_{A_j(m)} \alpha_{A_k(m)}}{2} \quad \text{(B.5)}$$

In order to define the covariance matrix, we introduce the notation Ψ_i^L (see section 5.1.1). The set Ψ_i^L refers to a sub-set of the search region, such that $\Psi_i^L \subseteq S_i$. This set includes the global indices of the pixels that are included in the search region and satisfy the respective overlap criterion, arranged by order of increasing dissimilarity values. $L \in [2, |S_i|]$ represents the number of dissimilarity elements that are being explored during the variance accumulation process (see ALGORITHM I in Chapter 3). Hence, for a given L, this sub-set includes the global indices of the L smallest dissimilarities in S_i, out of the patches that satisfy the non-overlap criterion. The notation $\tilde{\mathbf{d}}_i\left(\Psi_i^L\right)$ refers to the sorted dissimilarity vector.

The covariance matrix for a General patch-kernel is defined as (where $\kappa = \sum_{l \in [1,p^2]} \alpha_l^2$)

$$C_{\tilde{\mathbf{d}}_i^G} = E\left[\tilde{\mathbf{d}}_i^G \left(\tilde{\mathbf{d}}_i^G\right)^T\right] = \kappa \begin{bmatrix} 2 & 0.5 & .. & 0.5 \\ 0.5 & 2 & .. & 0.5 \\ | & | & | & | \\ 0.5 & 0.5 & .. & 2 \end{bmatrix} + 0.5O$$

where,

$$O = \begin{bmatrix} 0 & \cdots & \sum_{m \in O_{\Psi_i^L(1),\Psi_i^L(L)}} \alpha_{A_{\Psi_i^L(1)}(m)} \alpha_{A_{\Psi_i^L(L)}(m)} \\ \sum_{m \in O_{\Psi_i^L(1),\Psi_i^L(2)}} \alpha_{A_{\Psi_i^L(1)}(m)} \alpha_{A_{\Psi_i^L(2)}(m)} & \cdots & | \\ | & .. & | \\ \sum_{m \in O_{\Psi_i^L(L),\Psi_i^L(1)}} \alpha_{A_{\Psi_i^L(L)}(m)} \alpha_{A_{\Psi_i^L(1)}(m)} & .. & 0 \end{bmatrix} \tag{B.6}$$

For the Uniform patch kernel, we assign $\kappa = p^{-2}$, then the matrix O is obtained by

$$O = \kappa^2 \begin{bmatrix} 0 & \cdots & \left|O_{\Psi_i^L(1),\Psi_i^L(L)}\right| \\ \left|O_{\Psi_i^L(2),\Psi_i^L(1)}\right| & \cdots & | \\ | & .. & | \\ \left|O_{\Psi_i^L(L),\Psi_i^L(1)}\right| & .. & 0 \end{bmatrix} \tag{B.7}$$

B.3. Case 3: Correlation Between Dissimilarities of Patches That Overlap Each Other and The Reference Patch

This case refers to the similarity patches A_j, A_k for $j,k \in S_i$ that satisfy the overlap criterion: $\forall j \neq k,\ j,k \in S_i^S$, $A_j \cap A_k \neq \emptyset$, $A_j \cap A_i \neq \emptyset$, $A_k \cap A_i \neq \emptyset$, i.e., the patches overlap each other and/or the reference patch A_i. In addition to the correlation between the normalized dissimilarities that is due to the mutual reference patch, there are two other origins that stem from the overlap of the patches between themselves and with the reference patch. The cross-variance term is developed herein for a General patch-kernel.

First, we discuss the distribution of the normalized dissimilarity in this case. The distribution of this variable is no longer *Chi-Square*, and the corresponding variance term is developed herein.

We define the set of indices that are associated with the region of overlap between patches A_i and A_j as : $O_{i,j} = A_i \cap A_j = \left\{ m \,|\, m \in A_i \cap A_j \right\}$.

The variance of sum of *dependent* random variables, in this case, is expressed as:

$$\mathrm{Var}\left[\tilde{d}_i^G(j)\right] = \mathrm{Var}\left[\sum_{m\in A_i, f\in A_j, l\in\left[1,p^2\right]} \alpha_l \left(\frac{N_m - N_f}{\sqrt{2}\sigma_n}\right)^2\right] = \sum_{m\in A_i, f\in A_j, l\in\left[1,p^2\right]} \alpha_l^2 Var\left[\left(\frac{N_m - N_f}{\sqrt{2}\sigma_n}\right)^2\right] +$$

$$+2\sum_{m\in A_i, f\in A_j, k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)} \mathrm{Cov}\left(\left(\frac{N_m - N_k}{\sqrt{2}\sigma_n}\right)^2, \left(\frac{N_k - N_f}{\sqrt{2}\sigma_n}\right)^2\right)$$

The first term on the r.h.s of the equation is already known from previous computations:

$$\sum_{m\in A_i, f\in A_j, l\in\left[1,p^2\right]} \mathrm{Var}\left(\alpha_l \frac{N_m - N_f}{\sqrt{2}\sigma_n}\right)^2 = \frac{1}{4\sigma_n^4}\sum_{m\in A_i, f\in A_j, l\in\left[1,p^2\right]} \alpha_l^2 \mathrm{Var}\left(N_m^2 + N_f^2 - 2N_m N_f\right) =$$

$$= \frac{1}{4\sigma_n^4}\sum_{m\in A_i, f\in A_j, l\in\left[1,p^2\right]} \alpha_l^2 \left[2\mathrm{Var}\left[N_m^2\right] + 4Var\left[N_m N_f\right]\right] =$$

$$= \frac{1}{4\sigma_n^4}\sum_{l\in\left[1,p^2\right]} \alpha_l^2 \left[4\sigma_n^4 + 4\sigma_n^4\right] = 2\sum_{l\in\left[1,p^2\right]} \alpha_l^2 = 2\kappa$$

The second term of the r.h.s of the equation refers to the overlap of the compared patch A_j with the reference patch A_i. In the following development, the cross-products $N_i \cdot N_j\ \forall i \neq j$ and any other version of them are neglected since their expectation is zero and they do not contribute to the sum.

$$2\sum_{m\in A_i, f\in A_j, k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)} \mathrm{Cov}\left(\left(\frac{N_m - N_k}{\sqrt{2}\sigma_n}\right)^2, \left(\frac{N_k - N_f}{\sqrt{2}\sigma_n}\right)^2\right) =$$

$$= 2\sum_{m\in A_i, f\in A_j, k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)} \left(\mathrm{E}\left[\left(\frac{N_m - N_k}{\sqrt{2}\sigma_n}\right)^2 \cdot \left(\frac{N_k - N_f}{\sqrt{2}\sigma_n}\right)^2 \right] - \mathrm{E}\left(\frac{N_m - N_k}{\sqrt{2}\sigma_n}\right)^2 \cdot \mathrm{E}\left(\frac{N_k - N_f}{\sqrt{2}\sigma_n}\right)^2 \right) =$$

$$= \frac{2}{4\sigma_n^4} \sum_{m\in A_i, f\in A_j, k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)} \left(\mathrm{E}\left[\left(N_m^2 + N_k^2 - 2N_m N_k\right) \cdot \left(N_k^2 + N_f^2 - 2N_k N_f\right) \right] - \ldots \right)$$

$$\left(\ldots - \mathrm{E}\left(N_m^2 + N_k^2 - 2N_m N_k\right) \cdot \mathrm{E}\left(N_k^2 + N_f^2 - 2N_k N_f\right) \right) =$$

$$= \frac{1}{2\sigma_n^4} \sum_{m\in A_i, f\in A_j, k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)}^2 \left(\mathrm{E}\left[N_m^2 N_k^2 + N_m^2 N_f^2 + N_k^4 + N_k^2 N_f^2 \right] - 4\sigma_n^4 \right) =$$

$$= \frac{1}{2\sigma_n^4} \sum_{k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)} 2\sigma_n^4 = \sum_{k\in O_{i,j}} \alpha_{A_i(k)}\alpha_{A_j(k)}$$

The mean of the normalized dissimilarity remains the same, namely 1. Consequently the mean and variance of this variable are as follows:

$$\forall j \in S_i^S : \ \mathrm{E}\left[\tilde{d}_i^G(j)\right] = 1, \ Var\left[\tilde{d}_i^G(j)\right] = 2\kappa + \sum_{f\in O_{i,j}} \alpha_{A_i(f)}\alpha_{A_j(f)} \tag{B.8}$$

Now, we begin with the analysis of the cross-variance terms.

As for Case 2, we define the set of global indices included in the region of overlap between patches A_j and A_k, $j,k \in S_i^S$ as: $O_{j,k} = A_j \cap A_k = \left\{ m \,|\, m \in A_j \cap A_k \right\}$.

The cross-variance is defined as: $\mathrm{Cov}\left[\tilde{d}_i^G(j), \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] - 1$

We need to analyze the term $\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right]$, based on the definition of the normalized dissimilarity:

$$\mathrm{E}\left[\tilde{d}_i^G(j) \cdot \tilde{d}_i^G(k)\right] = \mathrm{E}\left[\sum_{m\in A_i, f\in A_j, l\in[1,p^2]} \alpha_l \left(\frac{N_m - N_f}{\sqrt{2}\sigma_n}\right)^2 \cdot \sum_{m\in A_i, s\in A_k, l\in[1,p^2]} \alpha_l \left(\frac{N_m - N_s}{\sqrt{2}\sigma_n}\right)^2 \right] =$$

$$= \frac{1}{4\sigma_n^4} \mathrm{E}\left[\sum_{m\in A_i, f\in A_j, l\in[1,p^2]} \alpha_l \left(N_m - N_f\right)^2 \cdot \sum_{m\in A_i, s\in A_k, l\in[1,p^2]} \alpha_l \left(N_m - N_s\right)^2 \right]$$

Notes:

1. The cross-products $N_i \cdot N_j \ \forall i \neq j$ and any other version of them will be neglected since their expectation is zero and they do not contribute to the sum.

2. The structure of the similarity patches defines the following rule: if $\left|O_{j,k}\right| > 0$ and both $\left|O_{i,j}\right| > 0$ and $\left|O_{i,k}\right| > 0$, then $O_{j,k} = O_{i,j,k}$

3. The notation $\mathbf{1}(\cdot)$ refers to the Indicator function.

$$
\begin{aligned}
&\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right] = \tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in A_i, l\in\left[1,p^2\right]}\alpha_l^2 N_m^4\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{i,j}}\alpha_{A_i(m)}\alpha_{A_j(m)}N_m^4\right] + \tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\alpha_{A_k(m)}N_m^4\right] \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\alpha_{A_k(m)}N_m^4\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in A_i\setminus\left\{O_{i,k}\right\}, l_1\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_1}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\}, l_2\in\left[1,p^2\right]\setminus\{l_1\}}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in A_i\setminus\{O_{i,k}\}, l_1\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_1}N_m^2\cdot\left(\sum_{s\in A_k, l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{i.k}}\alpha_{A_i(m)}N_m^2\cdot\left(\sum_{m\in O_{i.k}}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{i.k}}\alpha_{A_i(m)}N_m^2\cdot\left(\sum_{s\in A_k\setminus\{m\}, l_3\in\left[1,p^2\right]\setminus A_k(m)}\alpha_{l_3}N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathrm{E}\left[\textstyle\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k\setminus\{m\}, l_3\in\left[1,p^2\right]\setminus A_k(m)}\alpha_{l_3}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i, l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k, l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\}, l_2\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{i,j}\setminus\left\{O_{j,k}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\}, l_2\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{i,j}\setminus\left\{O_{j,k}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k, l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i, l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i, l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k, l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{i,j}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\}, l_2\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right] + \\
&+\tfrac{1}{4\sigma_n^4}\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\textstyle\sum_{m\in O_{i,j}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k, l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right] +
\end{aligned}
$$

$+\frac{1}{4\sigma_n^4}\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)4\mathrm{E}\left[\sum_{m\in O_{i,j},f\in O_{i,k}}\alpha_{A_j(m)}\alpha_{A_k(f)}N_m^2N_f^2\right]$

Analyzing each addend separately:

$(i)\ \mathrm{E}\left[\sum_{m\in A_i,l\in[1,p^2]}\alpha_l^2N_m^4\right]=3\sigma_n^4\sum_{l\in[1,p^2]}\alpha_l^2$

$(ii)\ \mathrm{E}\left[\sum_{m\in O_{i,j}}\alpha_{A_i(m)}\alpha_{A_j(m)}N_m^4\right]=3\sigma_n^4\sum_{m\in O_{i,j}}\alpha_{A_i(m)}\alpha_{A_j(m)}$

$(iii)\ \mathrm{E}\left[\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\alpha_{A_k(m)}N_m^4\right]=3\sigma_n^4\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\alpha_{A_k(m)}$

$(iv)\ \mathrm{E}\left[\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\alpha_{A_k(m)}N_m^4\right]=3\sigma_n^4\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\alpha_{A_k(m)}$

$(v)\ \mathrm{E}\left[\sum_{m\in A_i\setminus\{O_{i,k}\},l_1\in[1,p^2]\setminus A_i(m)}\alpha_{l_1}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in[1,p^2]\setminus\{l_1\}}\alpha_{l_2}N_f^2\right)\right]=$

$=\sigma_n^4\sum_{m\in A_i\setminus\{O_{i,k}\},l_1\in[1,p^2]\setminus A_i(m)}\alpha_{l_1}\cdot(1-\alpha_{l_1})=$

$=\sigma_n^4\left(1-\sum_{m\in O_{i,k}}\alpha_{A_i(m)}-\sum_{m\in A_i\setminus\{O_{i,k}\},l_1\in[1,p^2]\setminus A_i(m)}\alpha_{l_1}^2\right)$

$(vi)\ \mathrm{E}\left[\sum_{m\in A_i\setminus\{O_{i.k}\},l_1\in[1,p^2]\setminus A_i(m)}\alpha_{l_1}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in[1,p^2]}\alpha_{l_3}N_s^2\right)\right]=$

$=\sigma_n^4\sum_{m\in A_i\setminus\{O_{i.k}\},l_1\in[1,p^2]\setminus A_i(m)}\alpha_{l_1}=\sigma_n^4\left(1-\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\right)$

$(vii)\ \mathrm{E}\left[\sum_{m\in O_{i.k}}\alpha_{A_i(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in[1,p^2]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right]=$

$=\sigma_n^4\sum_{m\in O_{i.k}}\alpha_{A_i(m)}\left(1-\alpha_{A_i(m)}\right)$

$(viii)\ \mathrm{E}\left[\sum_{m\in O_{i.k}}\alpha_{A_i(m)}N_m^2\cdot\left(\sum_{s\in A_k\setminus\{m\},l_3\in[1,p^2]\setminus A_k(m)}\alpha_{l_3}N_s^2\right)\right]=$

$=\sigma_n^4\sum_{m\in O_{i.k}}\alpha_{A_i(m)}\left(1-\alpha_{A_k(m)}\right)$

$(ix)\ \mathrm{E}\left[\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k\setminus\{m\},l_3\in[1,p^2]\setminus A_k(m)}\alpha_{l_3}N_{fs}^2\right)\right]=$

$=\sigma_n^4\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\left(1-\alpha_{A_k(m)}\right)$

$(x)\ \mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{i,j}\setminus\{O_{i,j,k}\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in[1,p^2]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right]=$

$=\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\sigma_n^4\sum_{m\in O_{i,j}\setminus\{O_{j,k}\}}\alpha_{A_j(m)}\left(1-\alpha_{A_i(m)}\right)$

$(xi)\ \mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{i,j}\setminus\{O_{i,j,k}\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in[1,p^2]}\alpha_{l_3}N_s^2\right)\right]=$

$=\sigma_n^4\sum_{m\in O_{i,j}\setminus\{O_{j,k}\}}\alpha_{A_j(m)}$

$$(xii)\,\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)\sigma_n^4\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\left(1-\alpha_{A_i(m)}\right)$$

$$(xiv)\,\mathbf{1}\left(O_{j,k}=O_{i,j,k}\right)E\left[\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i,l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}=O_{i,j,}\right)\sigma_n^4\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}$$

$$(xv)\;\mathbf{1}\left(O_{j,k}{=}O_{i,j,k}\right)E\left[\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}{=}O_{i,j,k}\right)\sigma_n^4\sum_{m\in A_j\setminus\left\{O_{i,j}\right\}}\alpha_{A_j(m)}$$

$$(xvi)\;\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{i,j}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i\setminus\{m\},l_2\in\left[1,p^2\right]\setminus A_i(m)}\alpha_{l_2}N_f^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\sigma_n^4\sum_{m\in O_{i,j}}\alpha_{A_j(m)}\left(1-\alpha_{A_i(m)}\right)$$

$$(xvii)\,\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{i,j}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\sigma_n^4\sum_{m\in O_{i,j}}\alpha_{A_j(m)}$$

$$(xviii)\mathbf{1}\left(O_{j,k}{\neq}O_{i,j,k}\right)E\left[\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i,l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}{\neq}O_{i,j,k}\right)\sigma_n^4\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}$$

$$(xix)\;\mathbf{1}\left(O_{j,k}{\neq}O_{i,j,k}\right)E\left[\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{s\in A_k,l_3\in\left[1,p^2\right]}\alpha_{l_3}N_s^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}{\neq}O_{i,j,k}\right)\sigma_n^4\sum_{m\in A_j\setminus\left\{O_{j,k},O_{i,j}\right\}}\alpha_{A_j(m)}$$

$$(xiii)\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\mathrm{E}\left[\sum_{m\in O_{j,k}}\alpha_{A_j(m)}N_m^2\cdot\left(\sum_{f\in A_i,l_2\in\left[1,p^2\right]}\alpha_{l_2}N_f^2\right)\right]=$$

$$=\mathbf{1}\left(O_{j,k}\neq O_{i,j,k}\right)\sigma_n^4\sum_{m\in O_{j,k}}\alpha_{A_j(m)}$$

$$(xx)\,\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)4\mathrm{E}\left[\sum_{m\in O_{i,j},f\in O_{i,k}}\alpha_{A_j(m)}\alpha_{A_k(f)}N_m^2N_f^2\right]=$$

$$=\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)4\sigma_n^4\sum_{m\in O_{i,j},f\in O_{i,k}}\alpha_{A_j(m)}\alpha_{A_k(f)}$$

Finally, the cross-product expectation is obtained as follows:

$$\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]=0.5\sum_{l\in\left[1,p^2\right]}\alpha_l^2+0.5\sum_{m\in O_{i,j}}\alpha_{A_i(m)}\alpha_{A_j(m)}$$

$$+0.5\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\alpha_{A_k(m)}+0.5\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\alpha_{A_k(m)}+1+$$

$+\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)\sum_{m\in O_{i,j},f\in O_{i,k}}\alpha_{A_j(m)}\alpha_{A_k(f)}$

And the cross-variance term for a General patch kernel is given by:

$$\begin{gathered}\forall j,k\in S_i^S:\ \mathrm{Cov}\left[\tilde{d}_i^G(j),\tilde{d}_i^G(k)\right]=\mathrm{E}\left[\tilde{d}_i^G(j)\cdot\tilde{d}_i^G(k)\right]-1=\\ 0.5\kappa+0.5\sum_{m\in O_{i,j}}\alpha_{A_i(m)}\alpha_{A_j(m)}+\\ +0.5\sum_{m\in O_{i,k}}\alpha_{A_i(m)}\alpha_{A_k(m)}+0.5\sum_{m\in O_{j,k}}\alpha_{A_j(m)}\alpha_{A_k(m)}+\\ +\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)\sum_{m\in O_{i,j},f\in O_{i,k}}\alpha_{A_j(m)}\alpha_{A_k(f)}\end{gathered}\tag{B.9}$$

For the Uniform patch kernel, we use $\alpha_l=p^{-2},\forall l\in\left[1,p^2\right]$, that is $\kappa=p^{-2}$, and the corresponding cross-variance term is obtained by:

$$\forall j,k\in S_i^S:\ Cov\left[\tilde{d}_i^U(j),\tilde{d}_i^U(k)\right]=\frac{1}{2p^2}+\frac{\left|O_{i,j}\right|}{2p^4}+\frac{\left|O_{i,k}\right|}{2p^4}+\frac{\left|O_{j,k}\right|}{2p^4}+\mathbf{1}\left(\left|O_{i,j}\right|=\left|O_{i,k}\right|\right)\frac{\left|O_{i,j}\right|}{p^4}\tag{B.10}$$

C. Statistical Properties of The estimated variance For Different Cases of Patches Overlap

In this appendix we analyze the statistical properties of the estimated variance of the normalized dissimilarity elements within a given search region, arranged in a vector form. These properties are used to set an S_i^s set per pixel, as explained in section 5.4.

C.1. Case 1: Correlation Between Dissimilarities of Patches That Do Not Overlap

In sub-section 5.1.2, we defined the estimated mean of the normalized dissimilarity elements within a given search region S_i as $B = \frac{1}{L}\sum_{l=1}^{L} \tilde{d}_i\left(\Psi_i^L(l)\right)$, where Ψ_i^L is the set of global indices sorted by order of ascending normalized dissimilarities. The (unbiased) estimated variance is defined using the estimated mean as $\hat{V} = \frac{1}{L-1}\sum_{l=1}^{L}\left(\tilde{d}_i\left(\Psi_i^L(l)\right) - B\right)^2$. In practice, the estimated mean and estimated variance are computed for the sorted dissimilarity elements. We refer to the statistical properties of these variables at the variance threshold crossover point, i.e., when L represents the cardinality of the corresponding adaptive search region.

We start with the analytic development of the statistical properties of the estimated mean and then continue with the estimated variance. The development is conducted using the General patch-kernel and then simplified for the Uniform patch-kernel.

Notation: We distinguish between the notation of the estimated mean computed using the General patch-kernel and the Uniform patch-kernel, and refer to them as B^G, B^U, respectively. In a similar manner, we distinguish between the corresponding estimated variances, denoted $\hat{V}^G, \hat{V}^U$, respectively.

Estimated mean Analysis:

The expectation of the estimated mean is obtained by:

$$\mathrm{E}\left[B^G\right] = \frac{1}{L}\mathrm{E}\left[\sum_{l=1}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)\right] = \frac{1}{L}\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right] = 1$$

For the computation of the variance of the estimated mean, we analyze the second moment using the respective statistical properties of the dissimilarity vector (see eqns. (B.1), (B.2)). We remind the reader that the following definition was used in Chapter 3 (eqn. (3.9)): $\kappa = \sum_{s\in[1,p^2]} \alpha_s^2$, where α_s are the normalized patch-kernel coefficients.

$$\mathrm{E}\left[B^G\right]^2 = \frac{1}{L^2}E\left[\sum_{l=1}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]^2 = \frac{1}{L^2}\mathrm{E}\left[\sum_{l=1}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)^2 + \sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right] =$$

$$= \frac{1}{L^2}\left(\sum_{l=1}^{L} E\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)^2\right] + \sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L} E\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right]\right) =$$

$$= \frac{1}{L^2}\left(L(2\kappa+1) + L(L-1)(0.5\kappa+1)\right) = \frac{L+3}{2L}\kappa + 1$$

Now, the variance of the estimated mean is obtained by:

$$\mathrm{Var}\left[B^G\right] = \mathrm{E}\left[B^G\right]^2 - \left(\mathrm{E}\left[B^G\right]\right)^2 = \frac{L+3}{2L}\kappa$$

The estimated mean is a normal random variable as a linear combination of normal random

variables, thus the following applies:

$$B^G \sim \mathcal{N}\left(1, \frac{L+3}{2L}\kappa\right) \tag{C.1}$$

The estimated mean that corresponds to the dissimilarity elements that were computed using the Uniform patch-kernel is obtained by assigning $\alpha_s = \frac{1}{p^2}$, $\forall s \in \left[1, p^2\right]$, that is, $\kappa = p^{-2}$.

Estimated variance Analysis:

The expectation of the estimated variance is obtained by:

$$\mathrm{E}\left[\hat{V}^G\right] = \frac{1}{L-1}\mathrm{E}\left[\sum_{l=1}^{L}\left(\tilde{d_i^G}\left(\Psi_i^L(l)\right) - B^G\right)^2\right] = \frac{1}{L-1}\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d_i^G}\left(\Psi_i^L(l)\right)\right]^2 - \frac{L}{L-1}\mathrm{E}\left[B^G\right]^2 =$$

$$= \frac{L}{L-1}\left(2\kappa + 1 - \frac{L+3}{2L}\kappa - 1\right) = \frac{3}{2}\kappa$$

Analyzing the variance of the estimated variance in this domain, denoted the *Correlation domain*, is complicated due to the need to compute statistical properties of high order correlation terms. In order to avoid the analysis of these terms, we need to transform the correlated dissimilarity variables to another domain, in which they are uncorrelated, denoted the *Transform domain*. This transform is actually a whitening procedure of the covariance matrix.

DEFINITION C.1 [46]**:** *The Whitening Transform converts a given covariance matrix, of a Gaussian zero mean data Z of dimensions $L \times b$, to a unit diagonal matrix. This process is also known as de-correlation. The transform is given by $T = \Lambda^{-0.5}Q^T$, where Λ is the eigen-values (diagonal) matrix and Q is the respective eigen-vectors matrix whose columns are the orthonormal eigen-vectors.*

In our case, we wish to apply the whitening transform to the dissimilarity vector with subtracted estimated mean, whose dimensions are $L \times 1$, such that $L \in [2, |S_i|]$. The transformed

dissimilarity vector is defined as:

$$\hat{\mathbf{d}}_i^G = T \cdot \left(\tilde{\mathbf{d}}_i^G - B^G\right) = \Lambda^{-0.5} Q^T \cdot \left(\tilde{\mathbf{d}}_i^G - B^G\right) \tag{C.2}$$

The covariance matrix of the correlated dissimilarity elements, computed using a General patch-kernel, is given by

$$C_{\tilde{\mathbf{d}}_i^G} = \kappa \begin{bmatrix} 2 & 0.5 & ... & 0.5 \\ 0.5 & 2 & ... & 0.5 \\ | & .. & .. & | \\ 0.5 & .. & .. & 2 \end{bmatrix}_{L \times L} \tag{C.3}$$

As can be seen, this matrix is circulant, thus characterized with a unique structure.

The eigen-values matrix of dimensions $L \times L$ is given by:

$$\Lambda = Q^T C^G_{\tilde{\mathbf{d}}_i^G} Q = 0.5\kappa \begin{bmatrix} 3 & 0 & ... & 0 \\ 0 & 3 & ... & 0 \\ | & .. & .. & | \\ 0 & .. & .. & L+3 \end{bmatrix} \tag{C.4}$$

The eigen-vectors are defined as follows:

$$\begin{aligned} &\forall l \in [1,L] : \sum_{k=1}^{L} Q_{kl}^2 = 1 \ (Orthonormality) \\ &\forall l \in [1,L-1] : \sum_{k=1}^{L} Q_{kl} = 0 \\ &l = L, \forall k \in [1,L] : Q_{kL} = \frac{1}{\sqrt{L}} \end{aligned} \tag{C.5}$$

where Q_{kl} is the k^{th} element of the l^{th} eigen-vector.

Consequently, the transformed elements (eqn. (C.3)) are obtained by

$$\begin{aligned} &\forall l \in [1, L-1]: \ \hat{d}_i^G(l) = \sqrt{\tfrac{2}{3\kappa}} \sum_{k=1}^{L} \left[Q_{kl} \left(\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right) \right] \\ &l = L: \ \hat{d}_i^G(L) = \sqrt{\tfrac{2}{L(L+3)\kappa}} \sum_{k=1}^{L} \left(\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right) = 0 \end{aligned} \tag{C.6}$$

The whitening transform, in this case, reduces the dimensionality of the transformed dissimilarity vector by 1 ($\hat{d}_i^G \left(\Psi_i^L(L) \right) = 0$).

We wish to verify that the statistical properties of the of the first $L-1$ elements of the transformed vector are characterized by a standard normal distribution:

$$\mathrm{E}\left[\hat{d}_i^G(l)\right] = \sqrt{\tfrac{2}{3\kappa}} \left[\sum_{k=1}^{L} Q_{kl} \mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right] \right] = 0$$

$$Var\left[\hat{d}_i^G(l)\right] = \mathrm{E}\left[\hat{d}_i^G(l)\right]^2 = \tfrac{2}{3\kappa} E \left[\sum_{k=1}^{L} Q_{kl} \mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right] \right]^2 =$$

$$= \tfrac{2}{3\kappa} \sum_{k=1}^{L} Q_{kl}^2 \mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right]^2 +$$

$$+ \tfrac{2}{3\kappa} \sum_{k=1}^{L} \sum_{g=1, g\neq k}^{L} Q_{kl} Q_{gl} \mathrm{E}\left[\left(\tilde{d}_i^G \left(\Psi_i^L(l) \right) - B^G \right) \left(\tilde{d}_i^G \left(\Psi_i^L(g) \right) - B^G \right) \right]$$

We define the normal variable $R_k = \tilde{d}_i^G \left(\Psi_i^L(k) \right) - B^G$, $k \in [1, L]$ and wish to find its statistical properties:

$$\forall k \in [1, L]: \ \mathrm{E}[R_k] = 0$$

$$\forall k \in [1, L]: \ \mathrm{E}\left[R_k^2\right] = \mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(k) \right) \right]^2 + \mathrm{E}\left[B^G\right]^2 - 2\mathrm{E}\left[B^G \tilde{d}_i^G \left(\Psi_i^L(k) \right) \right] = 1 + 2\kappa + 1 +$$

$$+ \tfrac{L+3}{2L}\kappa - \tfrac{2}{L} E \left[\left(\tilde{d}_i^G \left(\Psi_i^L(k) \right) \right)^2 + \sum_{g=1, g\neq k}^{L} \tilde{d}_i^G \left(\Psi_i^L(k) \right) \tilde{d}_i^G \left(\Psi_i^L(g) \right) \right] = \tfrac{3}{2} \tfrac{L-1}{L} \kappa$$

$$\forall k, f \in [1, L], \ k \neq f: \ \mathrm{E}\left[R_k R_f\right] = \mathrm{E}\left[\left(\tilde{d}_i^G \left(\Psi_i^L(k) \right) - B^G \right) \left(\tilde{d}_i^G \left(\Psi_i^L(f) \right) - B^G \right) \right] =$$

$$= \mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(k) \right) \tilde{d}_i^G \left(\Psi_i^L(f) \right) \right] - 2\mathrm{E}\left[\tilde{d}_i^G \left(\Psi_i^L(k) \right) B^G \right] + \mathrm{E}\left[B^G\right]^2 = \tfrac{3}{2L} \kappa$$

Going back to the computation of the variance of the transformed dissimilarity element:

$$Var\left[\hat{d}_i^G \left(\Psi_i^L(l) \right) \right] = \tfrac{2}{3\kappa} \left[\sum_{k=1}^{L} Q_{kl}^2 \mathrm{E}\left[R_k^2\right] + \sum_{k=1}^{L} \sum_{g=1, g\neq k}^{L} Q_{kl} Q_{gl} \mathrm{E}[R_k R_g] \right] =$$

$$= \frac{2}{3\kappa}\left[\frac{3}{2}\frac{L-1}{L}\kappa\underbrace{\sum_{k=1}^{L}Q_{kl}^2}_{=1}+\frac{3}{2L}\kappa\underbrace{\sum_{k=1}^{L}Q_{kl}\sum_{g=1,g\neq k}^{L}Q_{gl}}_{=-1}\right]=1$$

The relations $\sum_{k=1}^{L}Q_{kl}^2=1$ and $\sum_{k=1}^{L}Q_{kl}\sum_{g=1,g\neq k}^{L}Q_{gl}=-1$ are based on eqn. (C.6).

After verifying the characteristics of the whitening transform applied to the zero-mean dissimilarity vector, we wish to return to the computation of the variance of the estimated variance variable. This variance is defined in matrix notation based on the inverse transform, i.e.:

$$\tilde{\mathbf{d}}_i^G - B^G = Q\Lambda^{0.5}\hat{d}_i^G \tag{C.7}$$

Consequently, the estimated variance is obtained by:

$$\hat{V}^G = \frac{1}{L-1}\left(\tilde{\mathbf{d}}_i^G - B^G\right)^T\left(\tilde{\mathbf{d}}_i^G - B^G\right) = \frac{1}{L}\left(\hat{\mathbf{d}}_i^G\right)^T\Lambda\hat{\mathbf{d}}_i^G =$$

$$= \frac{3}{2(L-1)}\kappa\sum_{l=1}^{L-1}\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2+\frac{L+3}{2(L-1)}\kappa\left(\underbrace{\hat{d}_i^G\left(\Psi_i^L(L)\right)}_{=0}\right)^2 = \frac{3}{2(L-1)}\kappa\sum_{l=1}^{L-1}\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2$$

In order to compute the variance of the term above, we need to analyze the second moment of the estimated variance variable:

$$\mathrm{E}\left[\hat{V}^G\right]^2 = \frac{9}{4(L-1)^2}\kappa^2\mathrm{E}\left[\sum_{l=1}^{L-1}\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\right]^2 =$$

$$= \frac{9}{4L^2}\kappa^2\mathrm{E}\left[\sum_{l=1}^{L-1}\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^4+\sum_{l=1}^{L-1}\sum_{k=1,k\neq l}^{L-1}\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\left(\hat{d}_i^G\left(\Psi_i^L(k)\right)\right)^2\right] =$$

$$= \frac{9}{4(L-1)^2}\kappa^2\left[3(L-1)+\sum_{l=1}^{L-1}\sum_{k=1,k\neq l}^{L-1}\underbrace{\mathrm{E}\left[\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\left(\hat{d}_i^G\left(\Psi_i^L(k)\right)\right)^2\right]}_{=1}\right] =$$

$$= \frac{9}{4(L-1)^2}\kappa^2\left[3(L-1)+(L-1)(L-2)\right] = \frac{9}{4}\frac{L+1}{L-1}\kappa^2$$

The relation $\mathrm{E}\left[\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\left(\hat{d}_i^G\left(\Psi_i^L(k)\right)\right)^2\right]=\mathrm{E}\left[\left(\hat{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\right]\mathrm{E}\left[\left(\hat{d}_i^G\left(\Psi_i^L(k)\right)\right)^2\right]=1, \forall l\neq k, l,k\in[1,L]$ is true due to the independency of the variables $\hat{d}_i^G\left(\Psi_i^L(l)\right), \hat{d}_i^G\left(\Psi_i^L(k)\right)$.

The variance of the estimated variance is obtained by:

$Var\left[\hat{V}^G\right]=\mathrm{E}\left[\hat{V}^G\right]^2-\left(\mathrm{E}\left[\hat{V}^G\right]\right)^2=\frac{9}{4}\frac{L+1}{L-1}\kappa^2-\frac{9}{4}\kappa^2=\frac{9}{2}\kappa^2\frac{1}{L-1}$

To summarize, the statistical properties of the estimated variance of the dissimilarity elements, computed using a General patch-kernel, are as follows:

$$\begin{aligned}\mathrm{E}\left[\hat{V}^G\right]&=\tfrac{3}{2}\kappa\\ Var\left[\hat{V}^G\right]&=\tfrac{9}{2}\kappa^2\tfrac{1}{L-1}\end{aligned}\tag{C.8}$$

By assigning $\kappa=p^{-2}$, we obtain the statistical properties of the estimated variance of the dissimilarity elements, computed using the Uniform patch-kernel.

C.2. Case 2: Correlation Between Dissimilarities of Patches That Overlap Each Other, But Not The Reference Patch

The analysis of the statistical properties of the estimated mean and variance for this case are elaborated herein.

Estimated mean Analysis:

The expectation of the estimated mean is obtained by:

$\mathrm{E}\left[B^G\right]=\frac{1}{L}\mathrm{E}\left[\sum_{l=1}^{L}\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]=\frac{1}{L}\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]=1$

For the computation of the variance of the estimated mean, we use the statistical propeties of the corresponding dissimilarity vector (see eqns. (B.7), (B.8)). We define $O_{j,k}=\left\{s|s\in A_j\cap A_k\right\}$

as the set of global indices that are associated with the region of overlap between similarity patches A_j and $A_k,\ j,k \in S_i^S$, and denote the set cardinality $\left|O_{j,k}\right|$.

$$\mathrm{E}\left[B^G\right]^2 = \frac{1}{L^2} E\left[\sum_{l=1}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]^2 = \frac{1}{L^2}\mathrm{E}\left[\sum_{l=1}^{L}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right)^2 + \sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L} \tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right] =$$

$$= \frac{1}{L^2}\left(\sum_{l=1}^{L}\left[\mathrm{E}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\right] + \sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right]\right) =$$

$$= \frac{1}{L^2}\left(L(2\kappa+1) + L(L-1)(0.5\kappa+1)\right) + \frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(f)}\alpha_{A_{\Psi_i^L(k)}(f)}\right) =$$

$$= \frac{L+3}{2L}\kappa + \frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(f)}\alpha_{A_{\Psi_i^L(k)}(f)}\right) + 1$$

Now, the variance of the estimated mean is obtained by:

$$\mathrm{Var}\left[B^G\right] = \mathrm{E}\left[B^G\right]^2 - \left(\mathrm{E}\left[B^G\right]\right)^2 = \frac{L+3}{2L}\kappa + \frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(f)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)$$

The estimated mean is a normal random variable as a linear combination of normal random variables, thus the following applies:

$$B^G \sim \mathcal{N}\left(1, \frac{L+3}{2L}\kappa + \frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(f)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)\right) \tag{C.9}$$

The estimated mean that corresponds to the dissimilarity elements that were computed using the Uniform patch-kernel is obtained by assigning $\alpha_s = \frac{1}{p^2},\ \forall s \in \left[1, p^2\right]$, that is $\kappa = p^{-2}$:

$$B^U \sim \mathcal{N}\left(1, \frac{L+3}{2L}\kappa + \frac{\kappa^2}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right|\right) \tag{C.10}$$

The variance of the estimated mean consists of a supplement (the second term) that is based on the relative location of the explored patches with respect to each other, and hence is changed for different sub-sets of explored dissimilarities.

Estimated variance Analysis:

The expectation of the estimated variance is obtained by:

$$\mathrm{E}\left[\hat{V}^G\right] = \frac{1}{L-1}\mathrm{E}\left[\sum_{l=1}^{L}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right) - B^G\right)^2\right] = \frac{1}{L-1}\left(\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]^2 - L\mathrm{E}\left[B^G\right]^2\right) =$$

$$= \frac{L}{L-1}\left(2\kappa + 1 - \frac{L+3}{2L}\kappa - 1\right) - \frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}}(f)\alpha_{A_{\Psi_i^L(k)}}(f)\right) =$$

$$= \frac{3}{2}\kappa - \frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}}(f)\alpha_{A_{\Psi_i^L(k)}}(f)\right)$$

The analysis of the variance of the estimated variance in the Correlation domain is complicated as stated for the former case. Its analysis in the Transform domain is not trivial as well, due to the form of the covariance matrix:

$$C_{\tilde{\mathbf{d}}_i^G} = E\left[\tilde{\mathbf{d}}_i^G\left(\tilde{\mathbf{d}}_i^G\right)^T\right] = \kappa\begin{bmatrix} 2 & 0.5 & .. & 0.5 \\ 0.5 & 2 & .. & 0.5 \\ | & | & | & | \\ 0.5 & 0.5 & .. & 2 \end{bmatrix} + 0.5O$$

where,

$$O = \begin{bmatrix} 0 & \cdots & \sum_{m\in O_{\Psi_i^L(1),\Psi_i^L(L)}}\alpha_{A_{\Psi_i^L(1)}}(m)\alpha_{A_{\Psi_i^L(L)}}(m) \\ \sum_{m\in O_{\Psi_i^L(1),\Psi_i^L(2)}}\alpha_{A_{\Psi_i^L(1)}}(m)\alpha_{A_{\Psi_i^L(2)}}(m) & \cdots & | \\ | & .. & | \\ \sum_{m\in O_{\Psi_i^L(L),\Psi_i^L(1)}}\alpha_{A_{\Psi_i^L(L)}}(m)\alpha_{A_{\Psi_i^L(1)}}(m) & .. & 0 \end{bmatrix} \tag{C.11}$$

Each explored pixel is associated with a different covariance matrix for different L values. The eigen-decomposition of this matrix yields different eigen-values in contrast to the simplified structure of the covariance matrix in Case 1. Consequently, the analysis of this matrix is impractical and will not be further developed.

To conclude, the mean of the estimated variance, in this case, for the General patch-kernel is:

$$\mathrm{E}\left[\hat{V}^G\right] = \frac{3}{2}\kappa - \frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{f\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}}(f)\,\alpha_{A_{\Psi_i^L(k)}}(f)\right) \tag{C.12}$$

To obtain the mean of the estimated variance for the Uniform patch-kernel, we assign $\alpha_s = p^{-2}, \forall s \in \left[1, p^2\right]$, hence $\kappa = p^{-2}$:

$$\mathrm{E}\left[\hat{V}^U\right] = \frac{3}{2}\kappa - \frac{\kappa^2}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right| \tag{C.13}$$

As explained in section 5.2.2, the correlation term due to to overlap between the patches is smaller by at least one order of magnitude than the variance mean that corresponds to Case 1, i.e., 1.5κ.

C.3. Case 3: Correlation Between Dissimilarities of Patches That Overlap Each Other and The Reference Patch

The analysis of the statistical properties of the estimated mean and variance for this case are elaborated herein.

Estimated mean Analysis:

The expectation of the estimated mean is obtained by:

$$\mathrm{E}\left[B^G\right] = \frac{1}{L}\mathrm{E}\left[\sum_{l=1}^{L}\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right] = \frac{1}{L}\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right] = 1$$

For the computation of the variance of the estimated mean, we use the statistical properties

of the respective dissimilarity vector (see eqns. (B.9), (B.10)).

$$\mathrm{E}\left[B^G\right]^2 = \frac{1}{L^2}E\left[\sum_{l=1}^{L}\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]^2 = \frac{1}{L^2}\mathrm{E}\left[\sum_{l=1}^{L}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\right] +$$
$$+\frac{1}{L^2}\mathrm{E}\left[\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right] =$$
$$= \frac{1}{L^2}\left(\sum_{l=1}^{L}\left[\mathrm{E}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right)^2\right] + \sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\tilde{d}_i^G\left(\Psi_i^L(k)\right)\right]\right) =$$
$$= \frac{1}{L^2}\left(L(2\kappa+1) + \sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)} + L(L-1)(0.5\kappa+1)\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)} + \sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\mathbf{21}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}\right) =$$
$$= \frac{L+3}{2L}\kappa + \frac{1}{L^2}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)} + 1 +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)} + \sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\mathbf{21}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)$$

Now, the variance of the estimated mean is obtained by:

$$\mathrm{Var}\left[B^G\right] = \mathrm{E}\left[B^G\right]^2 - \left(\mathrm{E}\left[B^G\right]\right)^2 = \frac{L+3}{2L}\kappa + \frac{1}{L^2}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)} +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)} + \sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) +$$
$$+\frac{1}{L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}$$

In contrast to the two other explored cases, the estimated mean, here, is not a normal random variable since the dissimilarity elements are not distributed normally.

The estimated mean that corresponds to the dissimilarity elements that are computed using the Uniform patch-kernel is obtained by assigning $\alpha_s = \frac{1}{p^2}$, $\forall s \in \left[1, p^2\right]$, hence $\kappa = p^{-2}$.

To conclude, the statistical properties of the estimated mean for the General patch-kernel are:

$$\begin{gathered}
\mathrm{E}\left[B^G\right] = 1 \\
\mathrm{Var}\left[B^G\right] = \frac{L+3}{2L}\kappa + \frac{1}{L^2}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}} \alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)} + \\
+\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}} \alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)} + \sum_{m\in O_{i,\Psi_i^L(k)}} \alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) + \\
\frac{1}{2L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right) + \\
+\frac{1}{L^2}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L} 1\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}} \alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}
\end{gathered} \tag{C.14}$$

The statistical properties of the estimated mean for the Uniform patch-kernel are obtained by assigning $\alpha_s = p^{-2}$, $\forall s \in \left[1, p^2\right]$, that is $\kappa = p^{-2}$:

$$\begin{gathered}
\mathrm{E}\left[B^U\right] = 1 \\
\mathrm{Var}\left[B^U\right] = \frac{L+3}{2L}\kappa + \frac{\kappa^2}{L^2}\sum_{l=1}^{L}\left|O_{i,\Psi_i^L(l)}\right| + \\
+\frac{1}{2L^2p^4}\sum_{l=1}^{L}\sum_{k=1,k\neq j}^{L}\left(\left|O_{i,\Psi_i^L(l)}\right| + \left|O_{i,\Psi_i^L(k)}\right| + \left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right| + 2\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,k\Psi_i^L(k)}\right|\right)\left|O_{i,\Psi_i^L(l)}\right|\right)
\end{gathered} \tag{C.15}$$

The variance of the estimated mean consists of a supplement that is based on the relative location of the explored patches with respect to each other and the reference patch, and hence is changed for different sub-sets of explored dissimilarities.

Estimated variance Analysis:

The expectation of the estimated variance is obtained by:

$$\mathrm{E}\left[\hat{V}^G\right]=\frac{1}{L-1}\mathrm{E}\left[\sum_{l=1}^{L}\left(\tilde{d}_i^G\left(\Psi_i^L(l)\right)-B^G\right)^2\right]=\frac{1}{L-1}\left(\sum_{l=1}^{L}\mathrm{E}\left[\tilde{d}_i^G\left(\Psi_i^L(l)\right)\right]^2-L\mathrm{E}\left[B^G\right]^2\right)=$$
$$=\frac{L}{L-1}\left(2\kappa+1-\frac{L+3}{2L}\kappa-1\right)+\frac{1}{L-1}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)}-\frac{1}{L(L-1)}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)}-$$
$$-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)}+\sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)-$$
$$-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)-$$
$$-\frac{1}{L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right|=\left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)=$$
$$=\frac{3}{2}\kappa+\frac{1}{L}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)}-$$
$$-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)}+\sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)-$$
$$-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)$$
$$-\frac{1}{L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right|=\left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)$$

The analysis of the variance of the estimated variance in the Correlation domain is complicated as stated for cases 1 and 2. Its analysis in the Transform domain is not trivial as well, as this case is a more general, thus much more complicated version of Case 2. In a similar manner to Case 2, each pixel is associated with a different covariance matrix for different L values (see eqns. (B.9), (B.10) for the cross-variance terms), thus its computation and eigen-decomposition analysis are impractical, hence will not be further developed.

To conclude, the mean of the estimated variance for the General patch-kernel is as follows:

$$\begin{aligned}\mathrm{E}\left[\hat{V}^G\right]&=\frac{3}{2}\kappa+\frac{1}{L}\sum_{l=1}^{L}\sum_{f\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(f)}\alpha_{A_{\Psi_i^L(l)}(f)}-\\&-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{i,\Psi_i^L(l)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(l)}(m)}+\sum_{m\in O_{i,\Psi_i^L(k)}}\alpha_{A_i(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)-\\&-\frac{1}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\sum_{m\in O_{\Psi_i^L(l),\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(m)}\right)\\&-\frac{1}{L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right|=\left|O_{i,\Psi_i^L(k)}\right|\right)\sum_{m\in O_{i,\Psi_i^L(l)},f\in O_{i,\Psi_i^L(k)}}\alpha_{A_{\Psi_i^L(l)}(m)}\alpha_{A_{\Psi_i^L(k)}(f)}\right)\end{aligned}\tag{C.16}$$

The mean of the estimated variance for the Uniform patch-kernel is obtained by assigning $\alpha_s = p^{-2}, \forall s \in \left[1, p^2\right]$, that is $\kappa = p^{-2}$:

$$\begin{aligned} \mathrm{E}\left[\hat{V}^U\right] = \tfrac{3}{2}\kappa + \tfrac{\kappa^2}{L}\sum_{l=1}^{L}\left|O_{i,\Psi_i^L(l)}\right| - \\ -\tfrac{\kappa^2}{2L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\left|O_{i,\Psi_i^L(l)}\right| + \left|O_{i,\Psi_i^L(k)}\right| + \left|O_{\Psi_i^L(l),\Psi_i^L(k)}\right|\right) \\ -\tfrac{\kappa^2}{L(L-1)}\sum_{l=1}^{L}\sum_{k=1,k\neq l}^{L}\left(\mathbf{1}\left(\left|O_{i,\Psi_i^L(l)}\right| = \left|O_{i,\Psi_i^L(k)}\right|\right)\left|O_{i,\Psi_i^L(l)}\right|\right) \end{aligned} \tag{C.17}$$

D. Block-Matching 3D (BM3D)

In Chapter 7, we propose to combine the model-based adaptive search region with the BM3D [9] denoising scheme. Here, we present a detailed overview of the BM3D algorithm, that consists of two main consecutive phases. Refer to Figure 7.1 for a schematic description of the algorithm flow.

D.1. Phase I

The reference noisy patch is denoted A_i (marked as R in Figure 7.1) and its size is $p^{Hard} \times p^{Hard}$.

Grouping

The original noisy image Y is searched in a A_i-centered $M^{Hard} \times M^{Hard}$ neighborhood, denoted S_i, for patches similar to the reference patch. The set of similar patches is simply defined by:

$$\begin{aligned} G^{Hard}(A_i) &= \left\{ A_j,\, j \in S_i \middle|\, d_{p^{Hard}}\left(A_i, A_j\right) \leq \tau^{Hard} \right\} \\ d_{p^{Hard}}\left(A_i, A_j\right) &= \frac{\left\| \gamma(A_i) - \gamma\left(A_j\right) \right\|_2^2}{\left(p^{Hard}\right)^2} \end{aligned} \tag{D.1}$$

where τ^{Hard} is the distance threshold for $d_{p^{Hard}}\left(A_i, A_j\right)$ under which two patches are assumed similar. $d_{p^{Hard}}\left(A_i, A_j\right)$ is the normalized quadratic distance between patches, that is measured in the transform domain, $\gamma(A_i)$ is a hard-thresholding operator applied in transform domain with a corresponding threshold of $\lambda_{2D}^{Hard}\sigma_n$, and σ_n is the noise STD (see eqn. (2.1)). In this manner, only transformed values above the defined threshold $\lambda_{2D}^{Hard}\sigma_n$ are used for the dissimilarity computation. The 3D group, which is built by stacking up the patches matched to A_i, is denoted $G^{Hard}(A_i)$. Patches in $G^{Hard}(A_i)$ are sorted in an ascending order of dissimilarity$\left(d_{p^{Hard}}\left(A_i, A_j\right)\right)$, such that the most B^{Hard} similar patches are chosen and the dimensions of the 3D group are $p^{Hard} \times p^{Hard} \times B^{Hard}$.

Collaborative Filtering

Once the 3D-block $G^{Hard}(A_i)$ is built, the collaborative filtering is applied. A 3D isometric linear transform is applied to the group, followed by a shrinkage of the transform spectrum. Finally the inverse linear transform is applied to estimate each patch in the 3D group

$$\hat{G}^{Hard}(A_i) = \left(T_{3D}^{Hard}\right)^{-1}\left\{\gamma\left(T_{3D}^{Hard}\left\{G^{Hard}(A_i)\right\}\right)\right\} \tag{D.2}$$

where γ is a hard thresholding operator with threshold $\lambda_{2D}^{Hard}\sigma_n$:

$$\gamma(x) = \begin{cases} 0 & if\ |x| \leq \lambda_{2D}^{Hard}\sigma_n \\ x & Otherwise \end{cases} \tag{D.3}$$

For practical purposes, the 3D transform T_{3D}^{Hard} is made up of two separable transforms: a 2D transform denoted by T_{2D}^{Hard} applied on each patch in the group $G^{Hard}(A_i)$, and a 1D transform denoted by T_{1D}^{Hard} applied along the third dimension of the 3D group. The choice of these transforms is discussed in [20].

Aggregation

Once the collaborative filtering is completed, we get an estimate for each patch in a 3D group. Each patch can be associated with several groups and since patches can overlap, each pixel is associated with different patches. Therefore, each pixel possess several estimates. In this stage, all the estimates of a given pixel are aggregated using a weighted average, with the following weights:

$$w_{A_i}^{Hard} = \begin{cases} \left(\sigma_n^2 K_{A_i}^{Hard}\right)^{-1} & if\ K_{A_i}^{Hard} \geq 1 \\ 1 & Otherwise \end{cases} \tag{D.4}$$

where $K_{A_i}^{Hard}$ is the number of non-zero coefficients of the 3D group in the transform domain.

This weighting scheme assigns a priority to homogeneous patches (characterized by small value of $K_{A_i}^{Hard}$), such that patches that contain an edge are assigned with a smaller weight compared to homogeneous ones. Dabov et. al [9] found that a satisfactory choice for aggregation weights would be ones that are inversely proportional to the total sample variance of the corresponding block-wise estimates. Thus, noisier block-wise estimates should be awarded smaller weights.

The result is an artifact reduction around the edges that avoids the classic ringing effects that is observed in transform threshold methods. The basic estimate after this first phase is given by:

$$\hat{X}_i^{Basic} = \frac{\sum_{i \in G^{Hard}(A_j)} w_{A_j}^{Hard} \sum_{A_k \in G^{Hard}(A_k)} \hat{A}_k^{Hard}(i)}{\sum_{i \in G^{Hard}(A_j)} w_{A_j}^{Hard}} \tag{D.5}$$

where $\hat{A}_k^{Hard}(i)$ is the basic estimate of pixel i that is associated to patch A_k. The external sum goes over all the 3D groups that include pixel i and the internal sum goes over all

the patches of the corresponding groups and uses the basic estimate of pixel i from these patches.

D.2. Phase II

The size of the patches in this phase is $p^{Wiener} \times p^{Wiener}$.

Grouping

Given the basic estimate $\hat{X}^{Basic}$ of the true image, obtained in phase 1, the denoising can be improved by performing grouping within this basic estimate and collaborative empirical Wiener filtering. Because the noise of the basic estimate is assumed to be significantly attenuated, the thresholding-based distance described in eqn. (D.1) is replaced with the normalized squared l_2-distance computed within the basic estimate itself.

$$\begin{aligned} G_{Hard}^{Wiener}(A_i) &= \left\{ \hat{A}_j^{Hard} | \, d_{p^{Wiener}}\left(\hat{A}_i^{Hard}, \hat{A}_j^{Hard}\right) \leq \tau^{Wiener} \right\} \\ d_{p^{Wiener}}\left(\hat{A}_i^{Hard}, \hat{A}_j^{Hard}\right) &= \frac{\left\| \hat{A}_i^{Hard} - \hat{A}_j^{Hard} \right\|_2^2}{\left(p^{Wiener}\right)^2} \end{aligned} \tag{D.6}$$

where τ^{Wiener} is a set dissimilarity threshold. The group $G_{Hard}^{Wiener}(A_i)$ is a 3D structure that consists the basic estimate blocks. Another 3D group is constructed by using the corresponding noisy blocks, i.e.,

$$G_{Noisy}^{Wiener}(A_i) = \left\{ A_j | \, d_{p^{Wiener}}\left(\hat{A}_i^{Hard}, \hat{A}_j^{Hard}\right) \leq \tau^{Wiener} \right\} \tag{D.7}$$

Collaborative Filtering

The empirical Wiener shrinkage coefficients are defined by using the energy of the 3D transform coefficients of the basic estimate group $G_{Hard}^{Wiener}(A_i)$:

$$W_{A_i}^{Wiener} = \frac{\left|T_{3D}^{Wiener}\left\{G_{Hard}^{Wiener}(A_i)\right\}\right|^2}{\left|T_{3D}^{Wiener}\left\{G_{Hard}^{Wiener}(A_i)\right\}\right|^2 + \sigma_n^2} \tag{D.8}$$

Then, the collaborative Wiener filtering of $G_{Noisy}^{Wiener}(A_i)$ is realized as the element-by-element multiplication of the 3D transform coefficients of the noisy data $T_{3D}^{Wiener}\left\{G_{Noisy}^{Wiener}(A_i)\right\}$ with the Wiener shrinkage coefficients $W_{A_i}^{Wiener}$. Subsequently, the inverse transform produces the group of estimates

$$\hat{G}^{Wiener}(A_i) = \left(T_{3D}^{Wiener}\right)^{-1}\left\{W_{A_i}^{Wiener} \cdot T_{3D}^{Wiener}\left\{G_{Noisy}^{Wiener}(A_i)\right\}\right\} \tag{D.9}$$

The choice of the transform T_{3D}^{Wiener} is discussed in [20].

Aggregation

In a similar manner to the first phase, each patch, and therefore each pixel, have several estimates that need to be aggregated. The weights are chosen to be inversely proportional to the total sample variance, such that:

$$w_{A_i}^{Wiener} = \left(\sigma_n^2 \left\|W_{A_i}^{Wiener}\right\|_2^2\right)^{-1} \tag{D.10}$$

The final estimate after this second phase is given by:

$$\hat{X}_i^{Wiener} = \frac{\sum_{i \in G_{Noisy}^{Wiener}(A_j)} w_{A_j}^{Wiener} \sum_{A_k \in G_{Noisy}^{Wiener}(A_k)} \hat{A}_k^{Wiener}(i)}{\sum_{i \in G_{Noisy}^{Wiener}(A_j)} w_{A_j}^{Wiener}} \tag{D.11}$$

where $\hat{A}_k^{Wiener}(i)$ is the final estimate of pixel i that is associated to patch A_k.

Bibliography

[1] S. Ahn and J.A. Fessler, “Standard Errors of Mean, Variance, and Standard Deviation Estimators”, July 2003.

[2] L. Alvarez, Y. Gousseau, and J. Morel, “The Size of Objects In Natural Images”, Advances in Imaging and Electron Physics, Vol. 111, pp. 167-242, 1999.

[3] N. Azzabou, N. Paragios, F. Guichard, "Uniform and Textured Regions Separation in Natural Images Towards MPM Adaptive Denoising", in Proceedings of SSVM, pp. 418-429, 2007.

[4] A. Buades, B. Coll, and J.M. Morel, "A Review Of Image Denoising Algorithms, With a New One", SIAM Journal on Multiscale Modeling and Simulations Vol. 4, No. 2, pp. 490–530, 2005.

[5] A.Buades, B. Coll, and J.M. Morel, “Nonlocal Image and Movie Denoising”, International Journal of Computer Vision, Vol. 76, pp. 123-139, 2008.

[6] T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient Nonlocal Means for Denoising of Textural Patterns", IEEE Transactions on Image Processing, pp. 1083 – 1092, July 2008.

[7] G. Casella and R.L. Berger, “Statistical Inference”, Book, pp. 243

[8] P. Coupé et. al, "An Optimized Blockwise Nonlocal Means Denoising Filter for 3-D Magnetic Resonance Images", IEEE Trans Med Imaging, Vol. 27, No. 4, pp. 425–441, April 2008.

[9] K. Dabov et al., “Image Denoising By Sparse 3-D Transform Domain Collaborative Filtering”, IEEE Trans Image Processing, Vol. 16, Issue 8, pp. 2080-2095, 2007.

[10] C.A Deledalle, V. Duval, and J. Salmon, “Non-Local Methods with Shape-Adaptive Patches (NLM-SAP)”, Journal of Mathematical Imaging and Vision archive, Vol. 43, Issue 2, pp. 103-120, June 2012.

[11] P.J. Dinesh, V.K. Govindan, and A.T. Mathew, "Robust Estimation Approach for Nonlocal-Means Denoising Based On Structurally Similar Patches", International Journal Open Problems in Computer Science, Vol. 2, No. 2, pp. 311-331, June 2009.

[12] V. Duval, Jean-Francois Aujol, and Yann Gousseau, “On The Parameter Choice For The Non-Local Means”, SIAM Journal on Image Sciences, hal-00468856, version 1, Mar 2010.

[13] B. Efron, “Transformation Theory: How Normal Is A Family of Distributions? “, The Annals of Statistics, Vol. 10, No. 2, pp. 323-339, 1982.

[14] A. Efros and T. Leung, “Texture Synthesis By Non Parametric Sampling, IEEE International Conference on Computer Vision, Volume 2, pp. 1033-1038, 1999.

[15] M. Elad and M. Aharon, “Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries”, IEEE Transactions on Image Processing, Vol. 15, Issue 12, pp. 3736-3745, December 2006.

[16] S. Gabler and C. Wolff, "A Quick and Easy Approximation To The Distribution Of A Sum Of Weighted Chi-Square Variables", Statistische Hefte 28, pp. 317-325 (1987).

[17] C. Kervrann and J. Boulanger, “Optimal Spatial Adaptation for Patch-Based Image Denoising”, IEEE Transactions on Image Processing, Vol. 15, No. 10, pp. 2866-2878, October 2006

[18] C. Kervrann, J. Boulanger, and P. Coupé, “Bayesian Non-Local Means Filter, Image

Redundancy and Adaptive Dictionaries for Noise Removal", in Proceedings of SSVM, pp. 520–532, 2007.

[19] P.W. Lamberti, A.P. Majtey, A. Borras, M. Casas, and A. Plastino, “Metric Character of The Quantum Jensen-Shannon Divergence”, Phys. Review, A 77, May 2008.

[20] M. Lebrun, “An Analysis and Implementation of the BM3D Image Denoising Method”, Image Processing On Line, 2 (2012), pp. 175-213.

[21] J.S. Lee, ”Digital Image Smoothing and The Sigma Filter”, Computer Vision, Graphics and Image Processing, Vol. 24, pp. 255-269, 1983.

[22] S. Lefkimmiatis, P. Maragos, and G. Papandreou, “Bayesian Inference on Multiscale Models For Poisson Intensity Estimation: Applications to Photon-Limited Image Denoising”, IEEE Transaction on Image Processing, Vol. 18, No. 8, pp. 1724-1741, August 2009.

[23] A. Levin et al., “Patch Complexity, Finite Pixel Correlations and Optimal Denoising”, Proceedings of the 12th European conference on Computer Vision, Volume Part V, pp. 73-86, 2012.

[24] X. Liu, M. Tanaka and M. Okutomi , “Noise Level Estimation Using Weak Textured Patches of a Single Noisy Image”, Proceedings of IEEE International Conference on Image Processing (ICIP 2012), pp. 665-668, September 2012.

[25] F. Luisier, T. Blu, and M. Unser, “Image Denoising in Mixed Poisson–Gaussian Noise”, IEEE Transactions On Image Processing, Vol. 20, No. 3, pp. 696-708, March 2011.

[26] M. Mäkitalo and A. Foi, “Optimal Inversion of The Anscombe Transformation In Low-Count Poisson Image Denoising , IEEE Transactions Image Processing, Vol. 20, No. 1, pp.99-109, January 2011.

[27] M. Mäkitalo and A. Foi, “Optimal Inversion of The Generalized Anscombe Transfor-

mation For Poisson-Gaussian Noise", IEEE transactions on image processing, Vol. 22, No. 1, pp. 91-103, January 2013.

[28] M. Mahmoudi and G. Sapiro, "Fast Image and Video Denoising Via Nonlocal Means of Similar Neighborhoods," IEEE Signal Processing Letters, Vol.12, No.12, pp. 839-842, 2005.

[29] M. Mäkitalo and A. Foi, "A Closed-Form Approximation of the Exact Unbiased Inverse of the Anscombe Variance-Stabilizing Transformation", IEEE Transactions on Image Processing, Vol. 20, No. 9, September 2011.

[30] H.S. Malvar, L.W. He, and R. Cutler, "High-Quality Linear Interpolation For Demosaicing of Bayer-Patterned Color Images", ICASSP 2004.

[31] M. Mohri, and A. Rostamizade, "Stability Bounds for Stationary φ-mixing and β-mixing Processes", Journal of Machine Learning Research, Vol. 11, pp. 789-814, 2010.

[32] J. Orchard, M. Ebrahimi, A. Wong, "Efficient Nonlocal- Means Denoising Using The SVD",15th IEEE International Conference on Image Processing, pp. 1732 – 1735, October 2008.

[33] I. Ram, M. Elad, and I. Cohen, "Image Processing using Smooth Ordering of its Patches", IEEE Transactions on Image Processing, Vol. 22, Issue 7, pp. 2764-2774, July 2013.

[34] J. Salmon, "On Two Parameters For Denoising With Non-Local Means", Signal Processing Letters IEEE, Vol. 17, Issue 3, pp. 269-272, March 2010.

[35] J. Shlens, "Notes on Kullback-Leibler Divergence and Likelihood Theory", Systems Neurobiology Laboratory, Salk Insitute for Biological Studies, La Jolla, CA 92037, August 2007.

[36] S.M. Smith and J.M. Brady, "Susan - A New Approach to Low Level Image Processing", International Journal of Computer Vision, Volume 23 (1), pp. 45-78, 1997.

[37] W. Sun, M. Han, "Adaptive Search Based Non-Local Means Image De-noising", Image and Signal Processing CISP '09, pp. 1- 4, 2009.

[38] N.A. Thacker, J.V. Manjon and P.A. Bromiley, "A Statistical Interpretation of Non-Local Means", IET Computer Vision (in press, 2009).

[39] C. Tomasi and R. Manduchi, "Bilateral Filtering For Gray and Color Images," 6^{th} International Conference on Computer Vision, pp. 839-46. 1998.

[40] D. Van De Ville, and M. Kocher, "SURE-Based Non-Local Means" , IEEE Signal Processing Letters, Vol. 16, No. 11, pp. 973-976, November 2009.

[41] 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, April 2004.

[42] B. Zhang, J.M. Fadili, and J.L. Starck, "Wavelets, Ridgelets, and Curvelets For Poisson Noise Removal", IEEE Transaction on Image Processing, Vol. 17, No. 7, pp. 1093-1108, July 2008.

[43] http://demo.ipol.im/demo/bcm_non_local_means_denoising

[44] http://sipi.usc.edu/database/?volume=textures&image=12

[45] https://en.wikipedia.org/wiki/Chi-squared_distribution

[46] http://www.engr.uky.edu/~lgh/classes/ee640/WhiteCovarianceMatrix.pdf

[47] http://en.wikipedia.org/wiki/Central_limit_theorem

[48] http://en.wikipedia.org/wiki/Variance#Sum_of_correlated_variables

[49] http://www.colourphil.co.uk/lab_lch_colour_space.shtml

[50] http://mobyle-serpico.rennes.inria.fr/cgi-bin/portal.py#forms::NDSafir

הסרת רעש מתמונות

בשיטה של ממוצעים לא-מקומיים מסתגלת ומבוססת מודל

הילה ברקוביץ

הסרת רעש מתמונות

בשיטה של ממוצעים לא-מקומיים

מסתגלת ומבוססת מודל

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר

מגיסטר למדעים בהנדסת חשמל

הילה ברקוביץ

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

אייר התשע"ד חיפה מאי 2014

המחקר נעשה בהנחיית פרופ' דוד מלאך וד"ר מאיר בר-זוהר בפקולטה להנדסת חשמל

תודות

ברצוני להודות לפרופ' דוד מלאך וד"ר מאיר בר זוהר על הנחייתם לאורך כל המחקר. ברצוני להודות במיוחד לפרופ' דוד מלאך על ששימש לי כמנטור, על מסירותו ועל סבלנותו שאינם ברורים מאליהם. בנוסף, אודה לצוות מעבדת SIPL: נמרוד פלג, יאיר משה, זיוה אבני, ואבי רוזן, אשר עזרו ביצירת סביבה המתאימה לביצוע מחקר זה. לסיום, אודה למשפחתי ולארוסי תומר, על תמיכתם ועידודם לכל אורך הדרך.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי

חלק מעבודה זו פורסם בכנס IMVC 2014 והוצג בכנס ISPA 2013: "Non-Local Means Denoising Using a Content-Based Search Region and Dissimilarity Kernel"

תקציר

הצורך בשיטות יעילות לשחזור תמונות הולך ומתרחב בשל הייצור ההמוני של תמונות דיגיטליות וסרטוני וידאו מסוגים שונים, המצולמים לעיתים בתנאי תאורה לקויים. למעשה, אין זה משנה מהי איכות המצלמה, שיפור התמונה תמיד נחוץ. תמונה דיגיטלית מיוצגת לרוב כמטריצה של פיקסלים, כאשר כל פיקסל משוייך עם רמת אפור או צבע מסויים. הערך של כל פיקסל הינו תוצאה של מדידת כמות האור המגיעה לכל תא בחיישן של המצלמה, על ידי ספירת כמות הפוטונים הפוגעים באותו תא, בהתאם לזמן החשיפה המוגדר במצלמה. שתי המגבלות העיקריות בשיטת רכישה זו הן יצירת טשטוש ורעש. טשטוש הינו מגבלה פנימית של המצלמה, הנובעת ממבנה החיישן ומשטח תא אופייני, על פניו מתבצעת הסכימה של הפוטונים הפוגעים כדי ליצור את ערך הפיקסל המתאים לתא הנידון. המגבלה השנייה נובעת מרעש, כך שכמות הרעש המתווספת בתהליך רכישת התמונה הינה תלויית אות ומאופיינת בפילוג פואסוני. בהתאם לפילוג זה, ככל שיותר פוטונים פוגעים בתא מסויים, כך רמת הרעש המתווספת לערך הפיקסל המשוייך עם אותו תא גדלה. עם זאת, בעבודות מחקר רבות, בוחנים את ביצועי האלגוריתמים עם רעש מלאכותי גאוסי אדיטיבי, או מניחים שהוא כזה, עם שונות ידועה ותוחלת השווה אפס. הרעש הגאוסי בפיקסלים שונים מהווה ריאליזציות שונות של משתנים גאוסיים בלתי-תלויים ובעלי התפלגות זהה (*i.i.d.*).

מטרת סינון רעש מתמונות היא לשחזר את התמונה המקורית בהינתן תמונה רועשת. בין השיטות לסינון רעש שהוצעו, השיטות המבוססות על חלוקת התמונה לבלוקים (patches) עוררו תשומת לב ניכרת בשנים האחרונות. אחת משיטות אלה, שבה נדון בתזה, הינה שיטת הממוצעים הלא-מקומיים (Non-Local Means), אשר הוצגה לראשונה על ידי Buades בשנת 2005. השיטה מנצלת את היתירות הקיימת בתמונה טבעית, על ידי השוואה בין סביבות קטנות של פיקסלים באיזור חיפוש מורחב, המוגדר בתמונה סביב הפיקסל שאותו מעוניינים לשחזר. באופן זה, רמת האפור של כל פיקסל משוערך נתונה כממוצע משוקלל של כל הפיקסלים האחרים באיזור החיפוש. המשקל של כל פיקסל באיזור החיפוש הוא פרופרציוני לדמיון בין הסביבה של הפיקסל אותו מעוניינים לשחזר ובין סביבת הפיקסל שאליו משווים באיזור החיפוש. כך, פיקסלים בעלי סביבה הדומה לסביבת פיקסל העניין יהיו בעלי משקלים גבוהים יותר וישפיעו יותר על ערכו הסופי. מאחר שהפיקסלים בתמונה הם בעלי קורלציה גבוהה, בעוד שמרכיבי הרעש הגאוסי הינם בלתי תלויים, המיצוע המשוקלל של הפיקסלים באיזור החיפוש מפחית את הרעש. לסיכום, הייחודיות בשיטת הממוצעים הלא-מקומיים נעוץ ביכולת השיטה לנצל קורלציה מרחבית באיזור חיפוש נתון לצורך סינון רעש.

איזור החיפוש המורחב הינו בדרך כלל סביבה ריבועית, הממורכזת סביב הפיקסל אותו מעוניינים לשחזר. סביבה זו עלולה לכלול פיקסלים שרמת האפור המקורית שלהם שונה מרמת האפור המקורית של פיקסל

העניין. כתוצאה מכך, השתתפותם של אותם פיקסלים בתהליך המיצוע המשוקלל יכולה לפגוע בביצועי תהליך סינון הרעש. על-מנת לבטל את השפעתם של הפיקסלים הללו על תהליך המיצוע, הציעו חוקרים שונים ליצור איזור חיפוש מסתגל (אדפטיבי), אשר אינו מכיל את הפיקסלים הללו. השיטות שהוצעו תלויות בכיול פרמטרים ומערבות היוריסטיקות.

בתיזה זו, אנו מציעים שיטה חדשה, מבוססת מודל, אשר מייצרת עבור כל פיקסל, שאותו מעוניינים לשחזר בתמונה, סביבת חיפוש מסתגלת מתוך איזור החיפוש המורחב ההתחלתי. שיטה זו מתבססת על הפילוג הסטטיסטי של ערכי מדד השוני המאפיין את שיטת הממוצעים הלא-מקומיים. בהינתן רעש גאוסי אדיטיבי, אנו מניחים שההפרשים בין סביבות של פיקסלי מקור דומים נתונים על-ידי הפרשי הרעש באותן סביבות. הפרשים אילו ניתנים למידול כבעלי התפלגות חי-בריבוע (Chi-Square), הניתנת לקירוב ע"י התפלגות נורמלית, תחת תנאים מתאימים. באופן זה, אנו ממדלים את ערכי מדד השוני של פיקסלים, שסביבתם דומה לסביבת פיקסל העניין, כבעלי התפלגות גאוסית עם מאפיינים סטטיסטיים ידועים. בשיטה המוצעת, אנו מסתמכים על השונות המאפיינת פילוג זה ומתייחסים לממד השוני של פיקסלים בעלי סביבות דומות כריאליזציות שונות של המשתנה הגאוסי הנידון. הייחודיות של השיטה המוצעת נעוץ בהיותה מבוססת מודל ועל כן אינה מצריכה כיול פרמטרים, בניגוד לשיטות אחרות שהציעו למצוא איזור חיפוש מסתגל. בנוסף, השיטה אינו מגבילה את איזור החיפוש האדפטיבי להיות רצוף, תכונה חשובה למרקמים, בניגוד לשיטות מוצעות אחרות. השווינו שיטה זו לשיטת הממוצעים הלא-מקומיים הסטנדרטית עבור תמונות טבעיות שונות עם רמות רעש שונות. השיטה המוצעת התגלתה כבעלת ביצועים טובים יותר הן כמותית והן ויזואלית מאשר השיטה הסטנדרטית המקבילה.

שיטת הממוצעים הלא-מקומיים משתמשת בגרעין מתאים לצורך חישוב מדד השוני. על-פי רוב, אחד משני סוגי גרעינים נמצא בשימוש: גרעין אחיד, לפיו כל פיקסל בסביבת הפיקסל הנידון מקבל משקל זהה, וגרעין "קופסה" ("Box"), לפיו פיקסלים המרוחקים ממרכז הסביבה המושווית מקבלים משקלים נמוכים יותר. סימולציות מראות כי הגרעין האחיד מתאים יותר לאיזורים חלקים ואילו גרעין ה"קופסה" מתאים יותר לאיזורים המתארים שפות או מרקמים. בתיזה זו, אנו מציעים להשתמש גם בסוג גרעין מסתגל באופן מקומי, בנוסף לאיזור החיפוש המסתגל. הסתגלות סוג הגרעין נקבעת לפי גודל איזור החיפוש המסתגל, כך שפיקסלים המשוייכים לאיזור חיפוש גדול ולכן מתארים בסבירות גבוהה איזורים חלקים, ישוייכו לגרעין האחיד ואילו פיקסלים בעלי איזור חיפוש קטן יותר, המתארים בסבירות גבוהה איזורי מרקם או שפות, ישוייכו לגרעין הקופסה. השיטה, שמשלבת איזור חיפוש מסתגל וסוג גרעין מקומי מסתגל, הושוותה הן לשיטה בה רק איזור החיפוש הוא מסתגל והן לשיטת הממוצעים הלא-מקומיים הסטנדרטית והתגלתה כבעלת ערך מוסף ביחס לשתי השיטות. כלומר, שימוש בגרעין המתאים למבנה המקומי בתמונה משפר את תוצאות הסרת הרעש, כך שאיזורים חלקים מוחלקים ואילו איזורי מרקמים או שפות שומרים על חדות.

בנוסף לכך, בחנו את מידת ההשפעה של הקורלציה בין איברי מדד השוני של שיטת הממוצעים הלא-מקומיים באיזור חיפוש נתון. שלושה מקורות של קורלציה נבחנו: קורלציה כתוצאה מהשוואה לאותו בלוק ייחוס, קורלציה כתוצאה מחפיפה בין הבלוקים, וקורלציה כתוצאה מחפיפה של בלוקים לבלוק הייחוס. נמצא כי לקורלציה יש השפעה על המודל אך השפעתה על מידת הסרת הרעש אינה משמעותית.

שיטת איזור החיפוש המסתגל המוצעת שולבה גם בשיטת התאמת בלוקים תלת-ממדיים Block Matching 3D (BM3D), הנחשבת כיום לשיטת הסרת הרעש הטובה ביותר. כתוצאה מהשילוב בין השיטות מתקבל חיסכון בחישובים ללא פגיעה בביצועי הסרת הרעש.

טכניקות סינון רעש גאוסי יכולות להתמודד גם עם רעש בעל התפלגות פואסונית. על-מנת לאפשר זאת, יש להשתמש בהתמרה מייצבת שונות (Variance Stabilizing Transform – VST), כדוגמת התמרת Anscombe. התמרה זו ממירה אות בעל התפלגות פואסונית לאות בעלת התפלגות גאוסית עם שונות השווה לאחד ותוחלת השווה לאפס. לאחר השימוש בהתמרה, ניתן להשתמש בשיטת הממוצעים הלא-מקומיים לצורך סינון הרעש, כאשר הקלט הוא התמונה המותמרת. לאחר סינון הרעש, יש לבצע התמרה הופכית על התמונה המסוננת. באופן דומה לסימולציות שנערכו עבור רעש גאוסי לבן אדיטיבי, ערכנו השוואה עבור תמונות עם רעש פואסוני בין השיטה המבוססת מודל המוצעת לבין שיטת הממוצעים הלא-מקומיים הסטנדרטית. בנוסף, ערכנו השוואה בין שיטת התאמת הבלוקים התלת-מימדיים המבוססת מודל לשיטה הסטנדרטית. התקבל כי עבור שתי השיטות שבחנו, נשמרה המגמה שהתקיימה עבור מודל הרעש הגאוסי. שיטת הממוצעים הלא-מקומיים המבוססת מודל הניבה תוצאות טובות יותר מאשר השיטה הסטנדרטית. כמו-כן, שיטת התאמת הבלוקים התלת-מימדיים המבוססת מודל הניבה תוצאות דומות לתוצאות השיטה הסטנדרטית המתאימה תוך חסכון בחישובים.