STOCHASTIC BILATERAL FILTER AND STOCHASTIC NON-LOCAL MEANS FOR HIGH-DIMENSIONAL IMAGES

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Christina M. Karam

UNIVERSITY OF DAYTON

Dayton, Ohio

May, 2015
STOCHASTIC BILATERAL FILTER AND STOCHASTIC NON LOCAL MEANS FOR HIGH DIMENSIONAL IMAGES

Name: Karam, Christina M.

APPROVED BY:

Keigo Hirakawa, Ph.D.
Advisor Committee Chairman
Assistant Professor, Department of Electrical and Computer Engineering

Tarek Taha, Ph.D.
Committee Member
Associate Professor, Department of Electrical and Computer Engineering

Eric Balster, Ph.D.
Committee Member
Associate Professor, Department of Electrical and Computer Engineering

John G. Weber, Ph.D.
Associate Dean
School of Engineering

Eddy M. Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering
ABSTRACT

STOCHASTIC BILATERAL FILTER AND STOCHASTIC NON LOCAL MEANS FOR HIGH DIMENSIONAL IMAGES

Name: Karam, Christina M.
University of Dayton
Advisor: Dr. Keigo Hirakawa

We propose Stochastic Bilateral Filter (SBF) and Stochastic Non-local Means (SNLM)—fast image filtering aimed at processing high dimensional images (such as color and hyperspectral images). SBF and SNLM are comprised of an efficient randomized process, where it agrees with conventional bilateral filter (BF) or non-local means (NLM) on average. By Monte-Carlo, we repeat this process a few times with different random instantiations so that they can be averaged to attain the correct BF/NLM output. The computational bottleneck of the SBF is constant with respect to the color dimension, meaning the execution time for hyperspectral images is nearly the same as the grayscale images. For SNLM, it is constant with respect to the window and block sizes but is still dependent on the color dimension. They are considerably faster than the conventional and existing “fast” bilateral filter and “fast” non-local means implementations.
For Peter Dagher
ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Kiego Hirakawa for the guidance and support he showed me since I moved to Ohio, whether in class, or throughout my research. Without him, this thesis would not have been completed. He was understanding during difficult times, always encouraging, and the best adviser --and friend-- I could ever hope to have.

I would also like to thank my lab mates, who made this experience fun and interesting.

To my friends who have caused me to procrastinate more times than I would like to admit, yet still believed my luck, and eventual hard work, would pull me through...you were right.

To mom and dad, thank you for raising me the way you did; I hope this makes you proud. To my family, both near and far, I would not be where I am without you. Your constant encouragement in every step I take, the support, and the love you always send my way keeps me going.

Last but definitely not least, I would like to thank Peter Dagher, who encouraged me during my undergraduate studies, and who kept encouraging me during my first year at UD. Not a day goes by when I don’t think of you, of how positive you were until the very end, and of how thankful I am to have had you in my life. You are the reason I never gave up.
# TABLE OF CONTENTS

ABSTRACT ................................................................. iii
DEDICATION ............................................................... iv
ACKNOWLEDGMENTS ......................................................... v
LIST OF FIGURES .......................................................... vii
LIST OF TABLES ............................................................. viii

I. INTRODUCTION ......................................................... 1

II. BACKGROUND AND RELATED WORK ............................. 4
   2.1 Bilateral Filter .................................................. 4
   2.2 Fast Bilateral Filter ............................................ 5
   2.3 Non-Local Mean ................................................ 7
   2.4 Fast Non-Local Means .......................................... 7

III. PROPOSED: STOCHASTIC BILATERAL FILTERING AND STOCHASTIC NON-LOCAL MEANS ........................................... 9
   3.1 Main Result ..................................................... 9
   3.2 Implementation And Analysis ................................ 12

IV. RESULTS .............................................................. 16

V. CONCLUSION .......................................................... 20

BIBLIOGRAPHY ........................................................... 21
LIST OF FIGURES

3.1 Mean squared error (MSE) between ordinary and stochastic bilateral filters (i.e. $E\|g - \tilde{g}\|^2$), as a function of number of SBF iterations $L$. The data was obtained by averaging over 96 images. ................................................................. 14

3.2 Mean squared error (MSE) between ordinary and stochastic non-local means filters (i.e. $E\|g - \tilde{g}\|^2$), as a function of number of SNLM iterations $L$. The data was obtained by averaging over 96 images. ................................................................. 14

4.1 Example bilateral filtering results ($\sigma = 10$, $\theta = 80$, noise variance = 20, window size = 100). The number of iterations for SBF was $L = 50$. Color image ($512 \times 768 \times 3$) was taken from the Kodak dataset. Hyperspectral image ($1040 \times 1392 \times 31$) was taken from a dataset in [1]. ................................................................. 17

4.2 Example non-local mean results ($\sigma = 10$, $\theta = 620$, noise variance = 20, window size = 10, block size = 3). The number of iterations for SNLM was $L = 10000$. Color image ($512 \times 768 \times 3$) was taken from the Kodak dataset. Hyperspectral image ($1040 \times 1392 \times 31$) was taken from a dataset in [1]. ................................................................. 18
LIST OF TABLES

3.1 Complexity analysis of bilateral filtering implementations. Complexity bottleneck is marked in red. \( M = \# \) color/spectrum of edge image, \( N = \# \) color/spectrum of filtering image, \( W = \) window size, \( Q = \# \) quantization steps, \( K = \# \) iterations in (2.4), \( L = \# \) iterations in SBF. ................................................................. 15

3.2 Complexity analysis of non-local means implementations. Complexity bottleneck is marked in red. \( M = \# \) color/spectrum of edge image, \( N = \# \) color/spectrum of filtering image, \( W = \) window size, \( B = \) block size, \( P = \) Bernoulli Probability for the weight in [2] or percentage of patches kept after preclassification in [3] \( Q = \# \) quantization steps, \( L = \# \) iterations in SNLM. ................................................................. 15
CHAPTER I

INTRODUCTION

Bilateral filter is an image smoothing filter, first introduced by Tomasi and Manduchi [4]. Image features such as edges and textures are preserved by the spatial kernel and range kernel that limit the averaging in bilateral filters to pixels that are spatially close and similar in intensity, respectively. Although bilateral filter has been proven useful for a number of image processing and computer vision applications—denoising [5], demoisacing [6], optical flow [7], tone mapping [8], exposure correction for videos [9], and flash/no-flash imaging [10] to name a few—one major limitation to bilateral filtering is the complexity of image-adaptive averaging weights.

A handful of “fast” bilateral filter implementations have been proposed in the past [11, 12]. The speedup stems from mathematical relations that allow convolution operators to replace the computational bottleneck of BF. For grayscale images, the complexity of these algorithms scales linearly with the image size. However, a straightforward extensions of these methods to color (3 color dimensions), multispectral (>3 spectral dimension), and hyperspectral (≫3 spectral dimensions) images increase computational and/or memory complexity of the convolutional operations exponentially. Hence ironically, the “fast” BF implementations is actually slower than the naive BF implementation (which scales linearly with color/spectral dimension). The complexity of “fast” BF implementations is prohibitively large for multispectral and hyperspectral images, in fact.
Non-local means is an extension of the bilateral filter that has shown advantages in case of noise removal [13]. It replaced the pixel-to-pixel similarity in range kernel with a patch-to-patch similarity measures that respects image structures such as edges better. Obviously, expansion of range kernel from a pixel to patches further increases complexity. The fast bilateral filter approaches do not generalize well to NLM, in much the same way they do not scale well with color dimensions. Though a few fast NLM implementations have been proposed [2, 3, 14, 15] with convolutional approaches, the complexity bottleneck remains an issue.

As with the existing fast BF approaches, the proposed SBF also replace the computational bottleneck of BF with convolutions. However, the number of required convolution operations remains constant for grayscale, color, multispectral, and hyperspectral images for the SBF. Though the number of multiplication operations in SBF technically scales linearly with the color/spectral dimension, it is small or negligible compared to the number of required convolution operations. Thus SBF is the fastest approach to BF of high dimensional images. The proposed SNLM is similar to the SBF in terms of replacing the bottleneck with convolutions. However, the number of convolutions is not invariant to the color dimensions, but rather to the window and block sizes, and that reduces the time consumption compared to the NLM approach, in particular in high dimensional images. In this paper, we propose stochastic bilateral filter (SBF)—a new fast bilateral filter implementation that processes hyperspectral images with nearly the same complexity as the grayscale images and stochastic non-local means (SNLM)—a new fast non-local means implementation whose complexity is invariant to the window and block sizes. This remarkable feat stems from an efficient randomized convolutional process, where it agrees with conventional bilateral filter and conventional non-local means on average. By Monte-Carlo, we repeat this process a few times with different random instantiations until averaged results attain the correct BF/NLM output. We prove that the rate at which SBF converges is invariant to the color/spectral dimension, meaning that the number
of convolution operations required by SBF remains constant for grayscale, color, multispectral, and hyperspectral images. Although the number of multiply/add operations in SBF technically scales linearly with the color/spectral dimension, it is negligible compared to the required convolution operations. The end result is a considerable speedup for bilateral filtering and non-local means filtering of hyperspectral images.

We emphasize that this is a theoretical paper aimed at reducing the complexity of BF/NLM for high dimensional image data. We do not at all suggest that the denoising ability of BF/NLM surpasses the state-of-art image denoising techniques [16–18] (it does not), nor do we attempt to improve the filtering qualities of BF/NLM. Nevertheless, BF/NLM continue to play important role in image processing and computer vision. This is partly due to the unmatched versatility and utility afforded by BF/NLM. For example, cross-bilateral filtering leverages image edges learned from a(n augmented) clean image to denoise noisy images. Bilateral filtering and non-local means also make use of all color/hyperspectral dimensions to preserve edges, while most alternative denoising techniques handle color/spectrum in an ad-hoc manner. Bilateral filtering and non-local means filtering are particularly suited for hyperspectral imaging, in the sense that the increased spectral samples collectively strengthen the evidence of edge presence. As a result, BF/NLM achieve higher denoising performance in hyperspectral imaging.
CHAPTER II

BACKGROUND AND RELATED WORK

2.1 Bilateral Filter

Bilateral filter smooths an image \( f : \mathbb{Z}^2 \rightarrow \mathbb{R}^N \) by sensing edges in “edge image” \( e : \mathbb{Z}^2 \rightarrow \mathbb{R}^M \) to yield its output \( g : \mathbb{Z}^2 \rightarrow \mathbb{R}^N \), where \( M \) and \( N \) denote color/spectral dimensions in \( e \) and \( f \), respectively [4]. In its original form, an output pixel \( g(i) \in \mathbb{R}^N \) is given by taking a weighted average of pixels in \( g \), as follows:

\[
g(i) = \frac{\int_{\Omega} w(i, j) \cdot \phi(e(i), e(j)) \cdot f(j) \cdot dj}{\int_{\Omega} w(i, j) \cdot \phi(e(i), e(j)) \cdot dj}, \tag{2.1}
\]

where \( \Omega \) is the set of all pixels in an image; the averaging weight for \( f(j) \) is given by the spatial kernel \( w(i, j) \) that represents the spatial proximity of location \( j \in \mathbb{Z}^2 \) to location \( i \in \mathbb{Z}^2 \); and the range kernel \( \phi(e(i), e(j)) \) representing the intensity proximity of \( e(j) \in \mathbb{R}^N \) to \( e(i) \in \mathbb{R}^N \). The denominator is used as a normalizing factor so that the weights locally add up to unity. In most cases, edge information is given by image \( f \) itself (i.e. \( e = f \)). When \( e \neq f \), (2.1) is known as cross-bilateral filter.

Gaussian function is a common choice for the spatial kernel \( w(i, j) \) and the range kernel \( \phi(e(i), e(j)) \):

\[
w(i, j) = \exp \left( -\frac{||i-j||^2}{2\sigma^2} \right),
\]

\[
\phi(e(i), e(j)) = \exp \left( -\frac{||e(i) - e(j)||^2}{2\theta^2} \right). \tag{2.2}
\]
where the user parameters $\sigma^2 > 0$ and $\theta^2 > 0$ control the degree of spatial and range smoothing, respectively (more smoothing with bigger parameters); and $\| \cdot \|$ denotes M-dimensional $\ell^2$ norm for $e(i) \in \mathbb{R}^M$. These kernels give rise to the edge-preservation. That is, pixels in the homogeneous regions ($e(j) \approx e(i)$) result in a range kernel close to 1, promoting averaging. On the other hand, the range kernel limits the contribution from the pixels across object boundary because $\phi(\cdot, \cdot)$ decays rapidly as $\|e(j) - e(i)\|$ becomes large. While Gaussian function yields the desired result in a single pass, it is computationally intensive.

The complexity of the bilateral filter is reported in Table 3.1. The window size $W$ refers to the support of $w(i,j)$ in (2.2). Although the support is $\Omega$ (i.e. entire image) technically, the window size $W$ can be made smaller in practice due to rapid decay of $w(i,j)$. Hence the integration in (2.1) has a linear complexity with respect $W^2$.

### 2.2 Fast Bilateral Filter

The fast BF implementations of [11, 12] are primarily developed for grayscale images. The interpretations we give below are their color/spectral imaging extensions—they are functionally equivalent to color/spectral BF as defined by (2.1) and (2.2), but computationally identical to the grayscale versions in [11, 12] when $M$ is set to 1. As the notation becomes dense for $M > 1$, we encourage readers to review details in [11, 12].

Our work is inspired in part by the fast BF implementation in [12]. It leverages the following convergence:

$$
\phi(e(i), e(j)) = \lim_{K \to \infty} \frac{1}{2KM} \sum_{k_1=0}^{K} \ldots \sum_{k_M=0}^{K} \left( \sum_{k_1=0}^{K} \ldots \sum_{k_M=0}^{K} c_k \right) \times \exp \left( \frac{j(2k_1 - K)(f_1(i) - f_1(j))}{\theta \sqrt{K}} + \ldots + \frac{j(2k_M - K)(f_M(i) - f_M(j))}{\theta \sqrt{K}} \right) 
$$

(2.3)
This relation allows BF in (2.1) to be rewritten as: 
\[ g(i) \approx \sum_{k \in \Lambda_K} c_k e^{j(2k^T - K_0) e(i) \theta} \left\{ w(i) *_i e^{-j(2k^T - K_0) e(i) \theta} f(i) \right\}, \tag{2.4} \]

where \(*_i\) denotes convolution over \(i\). This approximation is valid when \(K\) is sufficiently large, and the integration is now a convolution. Although convolution can be executed efficiently via the fast Fourier transforms or integral images [12], it is still the computational bottleneck of (2.4). The required number of convolution is \(2N \cdot (K + 1)^M\), which grows at an exponential rate with the increase of the color/spectral dimension \(M\) of the edge image \(e(i)\). This implies that even though the computational cost for grayscale image \((M = 1)\) is significantly improved, complexity of non-gray image \((M > 1)\) can become prohibitively large (even when \(K\) is small). The complexity of (2.4) is summarized in Table 3.1.

In another fast bilateral filter implementation proposed by [11], the range kernel \(\phi(\cdot, \cdot)\) in (2.1) is replaced by an integral:
\[ \phi(e(i), e(j)) = \int_{\mathbb{R}^M} \exp\left(-\frac{\|e(i) - u\|^2}{2\theta^2}\right) \delta(u - e(j)) du, \tag{2.5} \]

where \(\delta(\cdot)\) is an impulse function and \(u \in \mathbb{R}^M\). Substituting (2.5) into (2.1), BF is interpreted as a \(2 + M\) dimensional linear filtering. Specifically, define new Gaussian filter and sparse signal as
\[ w(i, u) = \exp\left(-\frac{\|i\|^2}{2\sigma^2} - \frac{\|u\|^2}{2\theta^2}\right) \tag{2.6} \]
\[ n(i, u) = f(i) \delta(u - e(i)). \]

Then (2.1) can be thought of as:
\[ g(i) = \left. \frac{w(i, u) *_{i,u} n(i, u)}{w(i, u) *_{i,u} \delta(u - e(i))} \right|_{u=e(i)}, \tag{2.7} \]

where \(*_{i,u}\) is the convolution over indexes \(i\) and \(u\). Because convolution can be implemented efficiently, this is a considerable speedup. To carry out (2.7), however, the memory requirement for \(n(i, u)\) is \(N \cdot Q^M\) per pixel, where \(Q\) is the number of quantization steps used in each color/spectral
component in $e(n)$. Although the computational cost for grayscale image ($M = 1$) is significantly improved, (2.7) is prohibitively complex for high dimensional images ($M > 1$) (even when $Q$ is small). See Table 3.1.

### 2.3 Non-Local Mean

Non-local mean smooths an image $f : \mathbb{Z}^2 \to \mathbb{R}^N$ by sensing edges in $e : \mathbb{Z}^2 \to \mathbb{Z}^M$ not based on intensity proximity but instead on the similarities of image patches near the pixel of interest. The output $g : \mathbb{Z}^2 \to \mathbb{Z}^N$ is computed as:

$$h(i) = \frac{\int_{\Omega} w(i,j) e^{-\sum_{k \in \Gamma} v(k) \frac{||e(i-k) - e(j-k)||^2}{2\theta^2}} f(j) d\,j}{\int_{\Omega} w(i,j) e^{-\sum_{k \in \Gamma} v(k) \frac{||e(i-k) - e(j-k)||^2}{2\theta^2}} d\,j},$$

(2.8)

where $\Gamma$ represents pixels in a $B \times B$ image block weighted by $v(k)$. Obviously, the complexity of NLM is higher than the BF (unless $B = 1$ which makes NLM in (2.8) identical to (2.1)).

### 2.4 Fast Non-Local Means

The fast NLM implementations of [3] and that of [2] are used for high-dimensional images. They are functionally equivalent to the NLM defined by (2.8), but computationally different as seen in Table 2.

The work proposed in [3] is based on convolutions. However, not all the pixels are taken into consideration. There is a pre-classification process based on three statistical moments which eliminates the dissimilar pixels based on a specific threshold [15]. The percentage of pixels is represented as "P" in Table 2. In (2.9) the $\delta$ takes a value of 1 or 0, determining whether the pixel is taken into account or not. This makes the process of non-local means filtering faster by eliminating a number computations from the original process.
The concept of [2] is similar to that of [3], in the sense that only a certain number of pixels are taken into account instead of all the pixels. However, the decision process is not the same. There is no pre-classification process, but a random binomial probability represented by P in Table 2. In equation (2.9), the δ accounts for a pixel taken into account or not. In this case, the assumption is that taking an average of a sample of the pixel and averaging yields a similar result as averaging out all the pixels in the sample.

\[
h(i) = \frac{\int_\Omega \delta w(i, j) e^{-\sum_{k \in \Gamma} v(k) \frac{||e(i-k) - e(j-k)||^2}{2\theta^2}} f(j) \, dj}{\int_\Omega \delta w(i, j) e^{-\sum_{k \in \Gamma} v(k) \frac{||e(i-k) - e(j-k)||^2}{2\theta^2}} \, dj}, \quad (2.9)
\]
CHAPTER III

PROPOSED: STOCHASTIC BILATERAL FILTERING AND STOCHASTIC NON-LOCAL MEANS

3.1 Main Result

As we saw in Section 2.2, the key step to accelerating nonlinear filtering is to replace the computation of image content-adaptive weights with convolution [11]. In this section, we propose a novel approach called stochastic bilateral filtering (SBF) that replaces the deterministic range kernel $\phi(\cdot, \cdot)$ with a random convolutional kernel, whose rate of Monte-Carlo convergence does not depend on the color/spectral dimensionality of the edge image $e(i)$. Thus, SBF is ideal for high dimensional images such as hyperspectral images.

Lemma 1 (Unbiased estimate of Gaussian function). Let $X \in \mathbb{R}^M$, $X \sim \mathcal{N}(0, \Sigma)$ denote a normal random vector, and $e \in \mathbb{R}^M$ is a constant vector. Then

$$\mathbb{E}[\cos(X^T e)] = \exp\left(-\frac{1}{2} e^T \Sigma e\right). \quad (3.1)$$

The variance of $\cos(X^T e)$ is no greater than $\frac{1}{2}$.

Proof. Recall multivariate characteristic function:

$$\mathbb{E}[\exp(\pm jX^T e)] = \exp\left(-\frac{1}{2} e^T \Sigma e\right). \quad (3.2)$$
Thus, (3.1) follows from

$$
E[\cos(X^T e)] = \frac{1}{2} E \left[ \exp(jX^T e) + \exp(-jX^T e) \right].
$$

(3.3)

The variance of $\cos(X^T e)$ is given by

$$
E[\cos(X^T e)^2] - E[\cos(X^T e)]^2
= \frac{1}{2} E \left[ 1 + \cos(2X^T e) \right] - \exp(-e^T \Sigma e)
= \frac{1}{2} + \frac{1}{2} \exp(-2e^T \Sigma e) - \exp(-e^T \Sigma e)
= \frac{1}{2} (1 - \exp(-e \Sigma e))^2 \leq \frac{1}{2}
$$

(3.4)

The significance of the lemma is that it replaces the range Gaussian kernel with the expected value of a random phenomenon. We may approximate the quantity in (3.1) by generating the normal random vector $L$ times, evaluating $\cos(X^T e)$ each time, and taking their ensemble average. This random approach will converge to the Gaussian function in the mean square error sense with at a rate upper-bounded by $\frac{1}{2L}$. What is remarkable is that the rate $\frac{1}{2L}$ is invariant to the vector length $M$ and image content (though the rate is even faster when $e^T \Sigma e$ is small).

**Theorem 1** (Stochastic Bilateral Filter). Let $X \sim \mathcal{N}(0, I/\theta^2)$ be a length $M$ random vector where $I \in \mathbb{R}^{M \times M}$ is an identity matrix. The SBF $\tilde{g}(i)$ defined as

$$
\tilde{g}(i) = \frac{E \cos(X^T e(i)) \{ w(i) \ast_i (\cos(X^T e(i)) f(i)) \} + \sin(X^T e(i)) \{ w(i) \ast_i (\sin(X^T e(i)) f(i)) \}}{E \cos(X^T e(i)) \{ w(i) \ast_i \cos(X^T e(i)) \} + \sin(X^T e(i)) \{ w(i) \ast_i \sin(X^T e(i)) \}}
$$

(3.5)

is equivalent to BF $g(i)$ in (2.1).

**Proof.** By Lemma 1, it is clear that

$$
E[\cos(X^T (e(i) - e(j)))] = \phi(e(i), e(j)).
$$

(3.6)
By trigonometric identity, we expand (3.6) as
\[
\mathbb{E} \left[ \cos \left( X^T (e(i) - e(j)) \right) \right] = \mathbb{E} \left[ \cos \left( X^T e(i) \right) \cos \left( X^T e(j) \right) + \sin \left( X^T e(i) \right) \sin \left( X^T e(j) \right) \right].
\]
(3.7)

Substituting (3.7) into (2.1) proves the theorem.

As was the case previously, the convolution is the critical path for the SBF. As a side note, a more complex range kernel (where \( \phi(\cdot, \cdot) \) may involve \( e^T \Sigma e \) instead of \( \|e\|^2/\theta^2 \) can be implemented with no additional cost.

**Corollary 1** (complex exponential SBF). Let \( X \sim \mathcal{N}(0, I/\theta^2) \) be a length \( M \) random vector where \( I \in \mathbb{R}^{M \times M} \) is an identity matrix. A new SBF \( \hat{g}(i) \) defined as
\[
\hat{g}(i) = \frac{\mathbb{E} \left[ \exp(j X^T e(i)) \{ w(i) * e(j) \} \exp(-j X^T e(i))f(i) \right]}{\mathbb{E} \left[ \exp(j X^T e(i)) \{ w(i) * e(j) \} \right]}
\]
(3.8)
is also equivalent to BF \( g(i) \) in (2.1).

Though on the onset \( \hat{g} \) seems to be less complex than \( \tilde{g} \), there is no computational advantage to \( \hat{g} \) in practice. This is due to the fact that convolutions with complex number requires filtering of real and imaginary components separately.

**Theorem 2** (Stochastic Non-Local Means). Let \( X \sim \mathcal{N}(0, I/\theta^2) \) be a length \( M \) random vector where \( I \in \mathbb{R}^{M \times M} \) is an identity matrix. The SBF \( \tilde{h}(i) \) defined as
\[
\tilde{h}(i) = \frac{\mathbb{E} \left[ \exp(-j(X*e)(i)) \int_{\Omega} \{ w(i, j) \} e^{-j(X*e)(j)} f(j) dj \right]}{\mathbb{E} \left[ \exp(-j(X*e)(i)) \right] \int_{\Omega} \{ w(i, j) \} e^{-j(X*e)(j)} dj}
\]
(3.9)
is also equivalent to NLM \( h(i) \) in (2.8) where \( X*e = \sum X_k e(i-k) \).

**Proof.** By multivariate characteristic function, it is clear that:
\[
\mathbb{E}[e^{jX_k(e(i-k)-e(j-k))}] = \exp \left( -\frac{(e(i-k)-e(j-k))^2}{2\theta^2} \right).
\]
(3.10)
where $X_k \sim \mathcal{N}(0, I/\theta^2)$.

Substituting (3.10) in (2.8) proves the theorem.

### 3.2 Implementation And Analysis

Due to the fact that the convergence of the randomized filter is invariant to the color/spectral dimension, we are left to simply choose $L$ that is large enough to meet a certain mean squared error tolerance of $1/2L$. The following pseudo-codes outline the steps of SBF and SNLM required to carry out (3.5) and (3.9) in Theorems 1 and 2 respectively.

**Algorithm 1 Stochastic Bilateral Filter**

initialize numerator $n(i) \leftarrow 0$
initialize denominator $d(i) \leftarrow 0$

for $L$ times do
    generate $X \sim \mathcal{N}(0, I/\theta^2)$
    compute $y(i) \leftarrow X^T e(i)$
    compute $c(i) \leftarrow \cos(y(i))$ and $s(i) = \sin(y(i))$
    update $n(i) \leftarrow n(i) + c(i) \{w(i) * (c(i)f(i))\}$
    update $n(i) \leftarrow n(i) + s(i) \{w(i) * (s(i)f(i))\}$
    update $d(i) \leftarrow d(i) + c(i) \{w(i) * c(i)\}$
    update $d(i) \leftarrow d(i) + s(i) \{w(i) * s(i)\}$

end for

set $\tilde{g}(i) \leftarrow n(i)/d(i)$

The complexity of the SBF pseudo-code is recorded in Table 3.1. The number of multiply/add operations in SBF scales linearly with $ML$, which stems from the step “$y(i) \leftarrow X^T e(i)$” in Algorithm 1. The remainder of Algorithm 1 is entirely invariant to color/spectral dimension $M$. As stated earlier in Lemma 1, the convergence rate of SBF is no worse than $1/2L$, which is independent of $M$. This implies that the number of SBF iterations $L$ needed to meet a certain error tolerance remains constant for grayscale, color, multispectral, and hyperspectral images. Hence, convolution

12
Algorithm 2 Stochastic Non-Local Means

initialize numerator \( n(i) \) and denominator \( d(i) \) to zero.

for \( L \) times do
  compute \( y(i) \leftarrow 0 \)
  for \( c \) times do
    generate \( X \sim N(0, I/\theta^2) \)
    compute \( y(i) \leftarrow y(i) + X^T \ast e_c(i) \)
  end for
  compute \( c(i) \leftarrow \cos(y(i)) \) and \( s(i) = \sin(y(i)) \)
  update \( n(i) \leftarrow n(i) + c(i) \{ w(i) \ast_i (c(i) \ast f(i)) \} \)
  update \( n(i) \leftarrow n(i) + s(i) \{ w(i) \ast_i (s(i) \ast f(i)) \} \)
  update \( d(i) \leftarrow d(i) + c(i) \{ w(i) \ast_i c(i) \} \)
  update \( d(i) \leftarrow d(i) + s(i) \{ w(i) \ast_i s(i) \} \)
end for

set \( \tilde{h}(i) \leftarrow n(i)/d(i) \)

operation—which is the bottleneck complexity of SBF—is performed \((2N + 2)L\) times, regardless of color/spectral dimension \( M \).

Recalling (3.4), the variance of \( \cos(\langle X^T (e(i) - e(j)) \rangle \) is actually smaller if \( \exp(-\eta \langle\eta\rangle^2) \) is large. This fact is reflected in Figure 3.1 showing that the mean squared error (MSE) between the ordinary and stochastic bilateral filters (i.e. \( \mathbb{E}\|g - \tilde{g}\|^2 \)) was lower when \( \theta \) was large. Considering MSE as a function of the number of SBF iterations performed \((L)\), the constant offset in Figure 3.1 between \( \theta = 20 \) and \( \theta = 50 \) curves is also not unexpected (compare \( \frac{1}{2\pi L} (1 - \exp(-\eta \langle\eta\rangle^2))^2 \) for \( \theta = 20 \) and \( \theta = 50 \) in log scale). Notice that the convergence rate in Figure 3.1 is even higher at the initial stages of the iteration. This is due to the fact that SBF in (3.5) is a ratio—the denominator dominates MSE in the first few iterations, and the numerator determines the long-term convergence rate. The spatial parameter \( \sigma \) had negligible influence on convergence rate.

The complexity of the SNLM pseudo-code is recorded in Table 3.2. The convergence rate of SNLM depends on the color dimension of the edge image and the filtering image, but is invariant with respect to \( W^2 \) and \( B^2 \). This implies that the convolution operation—which is also the
Figure 3.1: Mean squared error (MSE) between ordinary and stochastic bilateral filters (i.e. $E \| g - \tilde{g} \|^2$), as a function of number of SBF iterations $L$. The data was obtained by averaging over 96 images.

Figure 3.2: Mean squared error (MSE) between ordinary and stochastic non-local means filters (i.e. $E \| g - \tilde{g} \|^2$), as a function of number of SNLM iterations $L$. The data was obtained by averaging over 96 images.
Table 3.1: Complexity analysis of bilateral filtering implementations. Complexity bottleneck is marked in red. $M=$# color/spectrum of edge image, $N=$# color/spectrum of filtering image, $W=$window size, $Q=$# quantization steps, $K=$# iterations in (2.4), $L=$# iterations in SBF.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Per Pixel</th>
<th>Per Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>multiply</td>
<td>divide</td>
</tr>
<tr>
<td>Tomasi [4]</td>
<td>$W^2$</td>
<td>1</td>
</tr>
<tr>
<td>Paris [11]</td>
<td>$(M + 2)W^2$</td>
<td>1</td>
</tr>
<tr>
<td>Chaudhury [12]</td>
<td>$(M + 4N + 2)L$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

Table 3.2: Complexity analysis of non-local means implementations. Complexity bottleneck is marked in red. $M=$# color/spectrum of edge image, $N=$# color/spectrum of filtering image, $W=$window size, $B=$block size, $P=$Bernoulli Probability for the weight in [2] or percentage of patches kept after preclassification in [3] $Q=$# quantization steps, $L=$# iterations in SNLM.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Per Pixel</th>
<th>Per Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>multiply</td>
<td>divide</td>
</tr>
<tr>
<td>Buades [13]</td>
<td>$(B(M - 1) + 1 + N)W^2$</td>
<td>1</td>
</tr>
<tr>
<td>Chan [2]</td>
<td>$(B(M - 1) + 1 + N)W^2P$</td>
<td>$W^2$</td>
</tr>
<tr>
<td>Dauwe [3]</td>
<td>$(B(M - 1) + 1 + N)W^2P$</td>
<td>$W^2$</td>
</tr>
<tr>
<td>Goossens [14]</td>
<td>$3W^2M$</td>
<td>1</td>
</tr>
<tr>
<td>proposed SNLM</td>
<td>$(4N + 2)L$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

bottleneck complexity of the SNLM—is performed $(2N + 2 + M)L$ times. As for all the other version of the Non-Local means algorithms, they are all dependent on $W^2$ which makes the window size an inconvenience, in particular for hyperspectral images, since they are also dependent on the color/spectral dimensions of the edge and filtering images $M$ and $N$ respectively.
CHAPTER IV

RESULTS

Results shown in Figure 4.1 confirm that bilateral filtering largely succeeds in smoothing of an image while preserving edges by averaging pixels of similar intensities within a spatial neighborhood. We used the complete noisy color/hyperspectral images in Figure 4.1(a,e) as the edge reference image \( e(n) \) to filter the noisy green/550nm images in Figure 4.1(b,f), respectively. Given the same noise variance, BF achieves higher denoising performance in hyperspectral imaging, as predicted, as evidenced by the details of bricks and windows in Figure 4.1(f). More importantly, the results obtained from BF and SBF are nearly identical. Differences are hard to spot—note the wheel and handle of the second from left biker in Figure 4.1(d), and the edges of the rooftops of the buildings in Figure 4.1(h), for example.

Similarly, results shown in Figure 4.2 confirm that non-locals means filtering yields the same results as the bilateral filtering, where the edges are preserved. Again, we used the complete noisy color/hyperspectral images in Figure 4.2(a,e) as the edge reference image \( e(n) \) to filter the noisy green/550nm images in Figure 4.2(b,f), respectively. Given the same noise variance, NLM achieves higher denoising performance in hyperspectral imaging, as predicted, as evidenced by the details of bricks and windows in Figure 4.2(f). More importantly, the results obtained from NLM and SNLM are nearly identical. Differences are hard to spot—note the wheel and handle of the second from left
Figure 4.1: Example bilateral filtering results ($\sigma = 10$, $\theta = 80$, noise variance = 20, window size = 100). The number of iterations for SBF was $L = 50$. Color image ($512 \times 768 \times 3$) was taken from the Kodak dataset. Hyperspectral image ($1040 \times 1392 \times 31$) was taken from a dataset in [1].
Figure 4.2: Example non-local mean results ($\sigma = 10$, $\theta = 620$, noise variance =20 , window size =10, block size = 3 ). The number of iterations for SNLM was $L = 10000$. Color image ($512 \times 768 \times 3$) was taken from the Kodak dataset. Hyperspectral image ($1040 \times 1392 \times 31$) was taken from a dataset in [1].
biker in Figure 4.2(d), and the edges of the rooftops of the buildings in Figure 4.2(h), for example.

We developed a comparable implementation of BF, SBF, NLM and SNLM in Matlab. The reported execution times in Figures 4.1 and 4.2 are based on Matlab R2013a running on Acer Aspire computer with Intel i7 and 12GB DDR3 RAM. Although the speedup is not impressive in the RGB color images, improvement is considerable in hyperspectral images.
CHAPTER V

CONCLUSION

We proposed stochastic bilateral filter (SBF) and stochastic non-local means (SNLM), aimed at speeding up bilateral filtering (BF) and non-local means (NLM) for high dimensional image data. SBF and SNLM rely on a random convolutional process that matches BF and NLM on average. We proved that Monte-Carlo convergence rate of SBF remains invariant to color/spectral dimension, meaning hyperspectral images can be filtered as quickly as grayscale images, and that SNLM remains invariant to the window size. Experimental results confirm significant speed improvements of SBF and SNLM for high dimensional images.
BIBLIOGRAPHY


