
MIXED MULTISCALE BM4D FOR THREE-DIMENSIONAL OPTICAL COHERENCE TOMOGRAPHY DENOISING

A PREPRINT

Ashkan Abbasi¹, Amirhassan Monadjemi², Leyuan Fang³, Hossein Rabbani⁴, Bhavna Josephine Antony^{5,6}, and Hiroshi Ishikawa^{7,*}

¹Department of Ophthalmology, Casey Eye Institute, University of Isfahan, Iran. Oregon Health & Science University, USA.

²School of Continuing and Lifelong Education, National University of Singapore, Singapore

³College of Electrical and Information Engineering, Hunan University, China.

⁴Department of Biomedical Engineering, Medical Image and Signal Processing Research Center, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Iran.

⁵Electrical and Computer System Engineering, Faculty of Engineering, Monash University, Australia.

⁶Department of Infectious Diseases, Alfred Health, Australia.

⁷Department of Ophthalmology, Casey Eye Institute; and Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University, USA, (email: ishikawh@ohsu.edu).

*Corresponding author.

ABSTRACT

Since both 3D and multiscale processing often increase computations, there are a few optical coherence tomography (OCT) image denoising methods that offer both of these features. In this paper, we develop a multiscale extension for the block matching and 4D filtering (BM4D) method (which is a well-known 3D denoising method). We start our derivation in 2D by analyzing and extending the typical subbands denoising approach. Then, by empirically showing its shortcomings, we propose a method in which there is no need for directly denoising detail subbands, while it is possible to maintain the merits of both single-scale and multiscale methods. The proposed method is first examined thoroughly by constructing multiscale extensions for three single-scale natural image denoising methods. Then, we generalize our derivation to 3D by proposing mixed multiscale BM4D (mmBM4D) for OCT image denoising. We examined mmBM4D rigorously on three OCT image datasets captured by various imaging devices. The experiments reveal that mmBM4D significantly

outperforms its original counterpart and performs in par with state-of-the-art OCT denoising methods. In terms of peak-signal-to-noise-ratio (PSNR), mmBM4D surpasses the original BM4D by more than 0.68 decibels over the first dataset. In the second and third datasets, significant improvements in mean to standard deviation ratio, contrast to noise ratio, and equivalent number of looks are achieved. Furthermore, we presented experiments on retinal layer segmentation to evaluate the layer preservation quality of OCT denoising methods.

Keywords Wavelets · Sparse Representations · BM4D · Multiscale Denoising · Optical Coherence Tomography

1 Introduction

The retina is a layered structure in which the layers of neurons are interconnected by synapses. Optical coherence tomography (OCT) is a noninvasive imaging technique to obtain cross-sectional images of the retina with micrometer resolution. OCT is widely used in the diagnosis and management of patients with various ophthalmic disorders like macular degeneration and inflammatory retinal diseases [?]. In OCT, a low-coherence interferometry mechanism is used to measure back-scattering of light from tissues. Due to the nature of this image formation process, speckle noise is inevitable. The presence of noise heavily degrades the image quality and complicates analysis of morphological characteristics such as retinal layers [?, ?] and cysts [?, ?].

Early OCT denoising methods use a local neighborhood around each pixel to denoise it [?]. The OCT denoising quality was improved by leveraging more complex methods based on diffusion equations [?, ?] and total variations [?]. With the objective to better preserve statistical features of the data, Bayesian methods were utilized for modeling OCT images [?, ?, ?, ?, ?, ?]. However, due to modeling uncertainties, sophisticated priors should be utilized for the purpose of good performance, which usually leads to a time-consuming iterative inference [?, ?].

Nonlocal self-similarity (NSS) is one of the most successful priors for image modeling [?, ?, ?, ?], which has been employed multiple times by various OCT denoising methods in both spatial [?, ?] and transform [?, ?, ?] domains. Another promising prior is the sparsity of coefficients in some fixed transform domains such as wavelets [?, ?, ?, ?, ?], curvelets [?, ?, ?], and contourlets [?]. Although denoising methods based on these transforms are computationally efficient and inherently multiscale, exploiting fixed basis functions (atoms) restrict their effectiveness, which usually leads to some artifacts.

Recent OCT denoising methods have mostly employed data-driven approaches, which were either based on dictionary learning [?, ?, ?, ?, ?, ?, ?] or deep learning [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. In these studies, very promising results have been reported, however, adapting atoms for data substantially increases the inference time, and learning from high signal-to-noise-ratio (SNR) images restricts the applicability of a method because capturing high SNR OCT images is a difficult and costly procedure. In a very recent study, a combination of the low-rank approximation and tensor-based total variation has been successfully applied [?]. As opposed to the most recent methods, it is not supervised, however, due to the need for optimization during inference, it is computationally expensive.

With the objective of developing an effective OCT denoising method which does not rely on learning from high SNR images, reviewing the mentioned denoising methods reveals that, regardless of each method’s specific algorithm, exploiting sparsity, correlation between frames, and NSS priors are key features. Currently, the well-known block matching and 4D filtering (BM4D) method features all of these merits, and it was used previously as a benchmark method in some recent OCT denoising papers [?, ?, ?, ?, ?, ?, ?]. However, in these papers, BM4D was outperformed easily mostly due to the considerable artifacts in its results. This paper shows that the proposed multiscale extension for BM4D considerably reduces artifacts in a way that almost all of the mentioned OCT denoising methods can be outperformed in terms of both visual quality and feature preservation.

In order to develop an effective multiscale extension for BM4D, we first start our derivation in 2D, where we empirically analyze the subband denoising approach which was initially proposed by multiscale K-SVD (MS K-SVD) [?] for denoising natural images. By emphasizing on the shortcomings of this approach, we use a simple fusion method based on the wavelet subbands mixing [?, ?] to combine the results of single and multiscale methods. Although this method is simple, it greatly suppresses undesirable artifacts and maintains the low-frequency content and main structure of the image. Nevertheless, the need for directly denoising detail subbands makes it hard applying most image denoisers (e.g., BM4D) which are not developed specifically for denoising detail subbands. Therefore, we extend this method in a way that there is no need to explicitly denoise detail subbands. We have tested the proposed method for constructing multiscale extensions for multiple 2D single-scale denoisers. Finally, we extend our discussion to 3D and propose our multiscale extension for BM4D. All of the proposed multiscale extensions have been comprehensively examined on the genuine datasets. In addition, we have conducted retinal layer segmentation experiments on the results of denoising methods to reveal the effectiveness of our proposed method in comparison with other compared methods in preserving layer structure. Fig. 1 is based on the results reported in Section 6.2. This figure shows that our proposed mixed multiscale extensions for BM4D achieve superior results both in terms of quality and retinal layer preservation.

The rest of this paper is organized as follows. In Section 2, we briefly review the literature of multiscale denoising and ways to construct multiscale methods from single-scale ones. Our motivational example which emphasize the shortcomings of subbands denoising approach is presented in Section 3. Next, we propose our method in Section 4, and we generalize it for constructing multiscale extensions for various denoising methods in Section 5. Then, experiments on both natural and OCT images are presented in Section 6. Finally, we conclude this paper in Section 7.

2 Related Works

2.1 Multiscale Denoising

Image denoising methods often involve patch-based (or local) operations. When noise is weak, local modeling can achieve plausible results. However, as noise level increases, local image structures are substantially distorted by noise, and thus denoising through local operations becomes difficult. In this case, enlarging the effective modeling and denoising regions through multiscale processing is effective [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. Also, it is shown

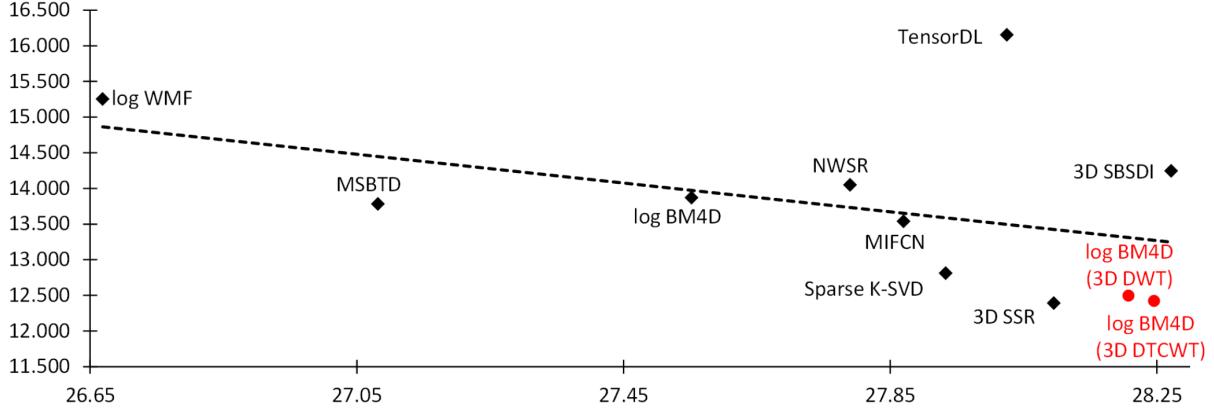


Figure 1: Plot of the relation between segmentation error and image quality score for different denoising methods when performing layer segmentation on results of the mentioned denoising methods. The vertical axis represents the sum of absolute error (SAE) for segmenting retinal layers from a dataset of OCT images, and the horizontal axis represents mean of peak signal to noise ratio (PSNR) for each denoising method. The dashed line shows the trend-line. Our proposed mixed multiscale methods are represented with red circles, and the compared methods are represented with black diamonds. This plot is drawn based on the results reported in Table 4 and 5. For more information, refer to Section 6.2.

that focusing on removing noise from high-frequency content in the most patch-based methods leads to considerable artifacts and low-frequency content loss [?, ?, ?, ?, ?, ?]. Even non-local methods such as non-local means [?, ?, ?] and block matching and 3D filtering (BM3D) [?] could not sufficiently reduce these types of artifacts [?, ?]. Nevertheless, promising results have been reported in several multiscale methods [?, ?, ?, ?, ?, ?, ?].

It is evident that multiscale denoising has appealing properties for processing OCT images, especially because noise in OCT images is not weak and preserving edges is of utmost importance for further analysis such as automatic layer segmentation. Therefore, here, we ask the question of how a 3D denoising method can be developed effectively to exploit the multiscale nature of the data.

Since both 3D and multiscale processing usually increase the computational costs of a method, there are very few OCT image denoising methods that have both of these properties simultaneously. In [?], a 3D dictionary based on dual-tree complex wavelet transform is offered for initializing 3D K-SVD [?, ?] and 3D Sparse K-SVD [?] denoising methods. Though 3D (Sparse) K-SVD is able to exploit the correlation between nearby frames, there is no guarantee to preserve the multiscale nature of the initial dictionary during the dictionary learning process. Better results have been reported by exploiting the subband denoising approach (initially proposed by MS K-SVD (Section 2.3)) in the 3D curvelet domain [?]. However, this method could not employ NSS prior, and therefore its performance is significantly below the current state-of-the-art methods such as [?, ?, ?, ?]. Also, due to heavy computations of 3D curvelets, usually OCT images were resized before feeding the method [?, ?]. Further, as our experiments with various 2D and 3D methods reveals in this paper, the subbands denoising approach (Section 2.3) is prone to generate high-frequency artifacts (Section 3). In contrast, our proposed method (Section 4) avoids directly denoising detail subbands and combines

the merits of single-scale and multiscale denoisers. Furthermore, due to the computational efficiency of the proposed method, there is no need to resize relatively big OCT volumes (e.g., $512 \times 1000 \times 100$ pixels) before denoising.

2.2 Constructing Multiscale Methods from Single-scale Ones

Tremendous efforts have been made over the years to construct multiscale denoising methods from single-scale ones. A considerable amount of these works has been done with the aim of integrating multiscale analysis into data-driven sparse methods, which are intrinsically single-scale [?, ?, ?, ?, ?, ?, ?, ?, ?]. In [?, ?], the authors showed how a wavelet basis can be adapted to the statistics of natural images, and achieved slightly better coding efficiency [?] and denoising [?]. But these methods are fairly elaborate and computationally demanding, since they rely on sampling techniques to infer sparse codes. The work by [?] has extended K-SVD [?, ?] to simultaneously use different sizes of atoms, each size corresponding to a different scale. They have reported very promising image restoration results, though their method is extremely computationally demanding.

In a different approach, off-the-shelf single-scale denoising methods are directly applied in the multi-resolution analysis domain [?, ?, ?, ?, ?, ?]. In this way, the denoising method acts as a complex shrinkage operator and the frequency selectivity of subbands acts as a divide and conquer strategy which could result in sparser representations, and thus simpler denoising subproblems. In these studies, the multi-resolution analysis domain was mainly wavelets [?] or Laplacian pyramid [?]. Some representative works in the Laplacian pyramid domain include the multiscale meta-procedure proposed by [?], multiscale EPLL (MSEPLL) [?], conservative scale recomposition [?], and MSND [?]. The discrete cosine pyramid was also exploited in MS DCT [?, ?] with notable results.

Wavelets were also successfully used to construct multiscale methods from single-scale ones [?, ?, ?, ?]. In [?], a method based on filtering principal component coefficients of wavelet packet subbands was presented. One of the most notable wavelet based construction methods is multiscale K-SVD (MS K-SVD) [?]. In this method, the well-known K-SVD image denoising method [?] was directly used to filter all wavelet subbands (including the approximate subband). They reported that MS K-SVD was more powerful than pure K-SVD in recovering textures, edges, and main image structures. However, on average, MS K-SVD was not able to outperform its single-scale baseline. Later, Fused K-SVD [?] proposed a joint sparse representation framework [?] over a learned dictionary to fuse the results of single and multiscale methods. Although the artifacts are suppressed greatly, the computational cost increases significantly since the joint sparse representations are obtained using concatenated input vectors and dictionaries.

2.3 Constructing Multiscale Methods via Subband Denoising

In this subsection, we briefly review the subband denoising method proposed by MS K-SVD [?] as a general construction that can be used to create a multiscale method from a single-scale one. Since various forms of wavelet transforms and filter banks have been proposed over the years, this approach has a potential to boost single-scale methods for different types of images by offering a range of filterbanks.

The steps of the subband denoising method are listed in Algorithm 1. On the first line, it applies a J -scale wavelet transform (denoted by $\text{WT}(\cdot)$) to the input image \mathbf{Y} and decomposes it into a set of subbands $\mathcal{S} = \{\mathbf{A}^J, \mathcal{D}^J\}$ which includes the approximate subband at scale J (\mathbf{A}^J) and a set (\mathcal{D}^J) containing the detail subbands. Next, each subband is denoised via a single-scale denoising method (denoted by $\text{Dn}(\cdot)$) in line 4. Then, the denoised image ($\hat{\mathbf{X}}_M$) is reconstructed by computing the inverse wavelet transform ($\text{iWT}(\cdot)$) at line 6.

Algorithm 1 multiscale Denoiser (mDn)

Input: \mathbf{Y} : Noisy image, σ_S : Noise standard deviation of the subband S , p : Patch size, and J : Number of scales.
Hyper-parameters: $\text{Dn}(\cdot)$: Single-scale denoiser, $\text{WT}(\cdot)$ and $\text{iWT}(\cdot)$: Wavelet transform and its inverse. The analysis and synthesis filterbanks are denoted by af and sf , respectively.

```

1:  $\mathcal{S} = \text{WT}(\mathbf{Y}, J, af)$  //  $\mathcal{S}$  contains approximate and detail subbands:  $\{\mathbf{A}^J, \mathcal{D}^J\}$ 
2:  $\hat{\mathcal{S}} = \{\}$  // An empty set is initialized.
3: for each subband ( $\mathbf{S} \in \mathcal{S}$ ) do
4:    $\hat{\mathbf{S}} = \text{Dn}(\mathbf{S}, \sigma_S, p)$  // The single-scale denoising method denoises each subband.
5:    $\hat{\mathcal{S}} = \hat{\mathcal{S}} \cup \hat{\mathbf{S}}$  // The Denoised subband will be added to the set.
6: Output:  $\hat{\mathbf{X}}_M = \text{iWT}(\hat{\mathcal{S}}, J, sf)$ 
```

Although, in the original MS K-SVD’s paper [?], mDn (Algorithm 1) was realized using K-SVD denoising [?] and 2D discrete wavelet transform (DWT) [?], here, we emphasize that this method is general enough to be implemented with other single-scale denoisers or wavelet transforms. It can also be used to implement 3D multiscale denoising methods. We note that our upcoming experiments support this claim. Therefore, we consider mDn’s single-scale denoiser, the type of J -scale wavelet transform, and filter bank as hyper-parameters of this method. In the following, after analyzing the shortcomings of mDn, we develop our proposed way of constructing a multiscale denoising method.

3 Motivational Example

For analyzing mDn (Algorithm 1), it is required to realize it with a specific single-image denoiser ($\text{Dn}(\cdot)$) and wavelet transform ($\text{WT}(\cdot)$). According to [?], we can use K-SVD denoising [?] in place of $\text{Dn}(\cdot)$ and 2D DWT [?] with discrete Meyer filterbank in place of $\text{WT}(\cdot)$. Here, we use the same wavelet transform and filterbank. But, we substitute the sparsifying transform learning denoising (TLD) [?] method in place of $\text{Dn}(\cdot)$ due to its simplicity and computational advantages over K-SVD. The results with K-SVD will be reported in Section 6.1.

By replacing TLD into mDn, we call the resultant method as mTLD. Similar to TLD’s original papers [?, ?, ?], we set the patch size (p) equal to 11×11 pixels. And, we estimate the noise level (σ) from the noisy image using a noise estimation method [?, ?]. We experimentally found that we can use the noise level of the input image for denoising its subbands $\sigma_S = \sigma$ (Algorithm 1, line 4).

The outputs of TLD and its multiscale version (mTLD) for denoising a test image are shown in Fig. 2 (b). In this example, mTLD is implemented with a 1-scale ($J=1$) DWT, and the results are shown in Fig. 2 (c) and (d). The quality of denoised images is quantized using the Peak Signal-to-Noise-Ratio (PSNR) and structural similarity index (SSIM) [?].

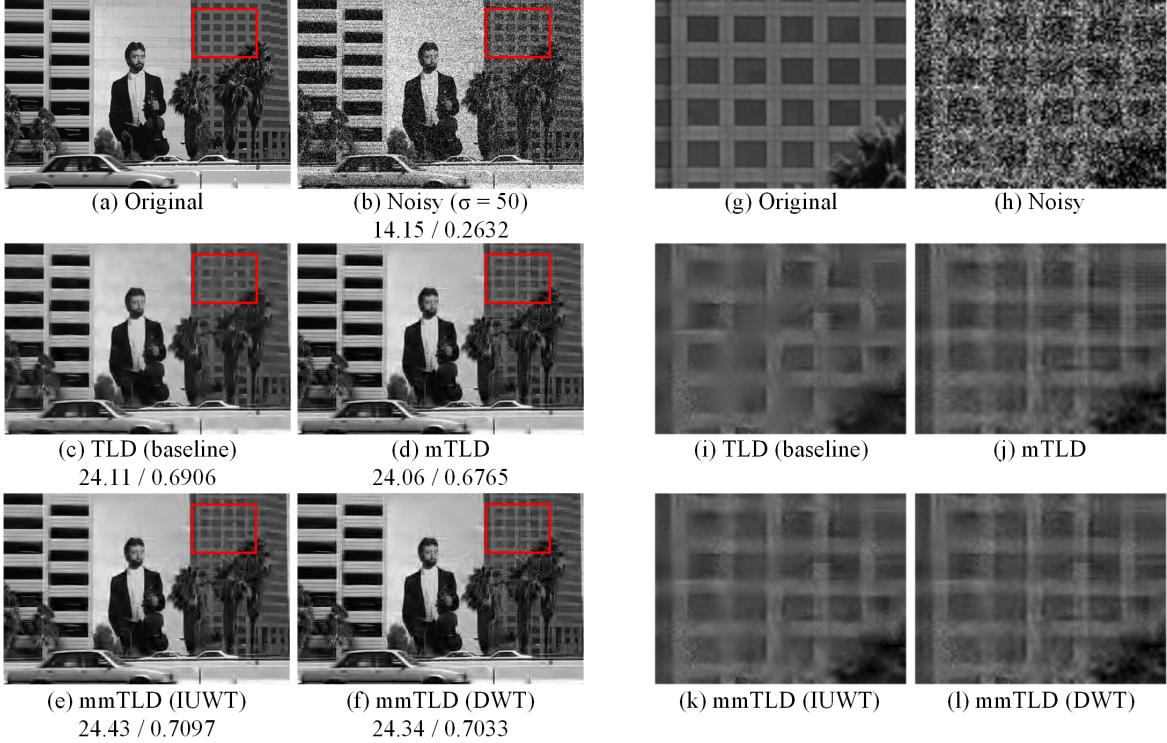


Figure 2: Visual comparison of the denoising results by a single-scale denoiser (here, TLD) and its multiscale extensions constructed with: 1. mDn (Algorithm 1) which is denoted by mTLD, and 2. our proposed mmDn which is denoted by mmTLD. Zoomed region for mTLD (j) shows substantial artifacts. These artifacts are mitigated in the results of mmTLD due to using subbands mixing. Here, mmTLD is implemented using two different types of wavelet transform at its mixing stage. The transform type is mentioned in the parenthesis. For each method, the PSNR and SSIM results are reported below.

Figure 2 (c) and (d) show that although the main image structures (or low-frequency information) are better recovered through mTLD, the high frequency artifacts in mTLD's output negatively affect the result. TLD does not suffer from such artifacts, and it effectively removes high frequency noise, but it also removes the prominent image structures and degrades the low frequency content. Our experiments in Section 6.1 confirms that this does not happen accidentally, and in general mTLD is inferior to TLD. In general, implementing mDn with other single-scale denoisers also leads to similar results. The same observation has been reported in [?] when MS K-SVD [?] and pure K-SVD denoising [?] methods were compared.

In order to restore the damaged structures in Fig. 2 (c), a reader may suggest that we can carefully control the amount of denoising in TLD. This can be done by controlling the sparsity level used for recovering patches in TLD, because it has a direct impact on the amount of denoising. In TLD [?, ?, ?], the algorithm updates the sparse representations until the error $\|Y - \hat{U}\|$ for recovering noisy patches Y by \hat{U} falls below a given threshold $nC^2\sigma^2$; where C is a fixed constant, n is the dimension of patches (here, $n = 11 \times 11 = 121$) and σ is the standard deviation of noise. Therefore, C is considered as a hyper-parameter which controls the sparsity of representations. To visually inspect the effects of different C values, we change it with steps of 0.05 in Fig. 3. Comparing Fig. 3 (b) and (c), it can be seen that although the default value of C is reasonable for this specific image, when we set C to 0.99 (Fig. 3 (b)), PSNR maximizes, and

the damaged structures are better recovered. However, note that this is achieved at the expense of decreasing SSIM and tolerating more artifacts. It turns out that even with this careful hyper-parameter adjustment, TLD cannot efficiently recover the damaged parts without sacrificing quality.

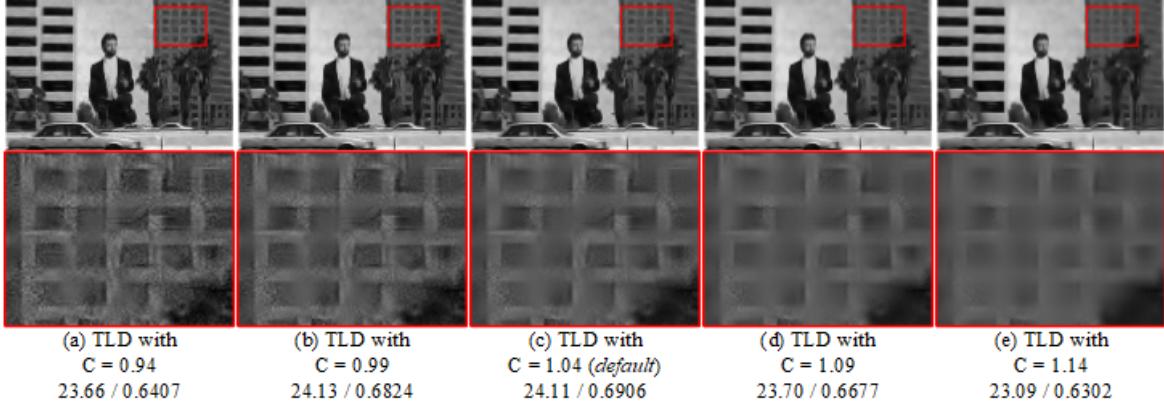


Figure 3: Effects of using different sparsity levels (or error thresholds) on the output of TLD. The C constant modifies the sparsity level in TLD [?, ?, ?]. By increasing this constant, the representations in TLD become sparser, and, therefore, noise suppression becomes stronger. For each method, the PSNR and SSIM are reported below.

A number of reasons can be thought of for explaining why mTLD generally leads to inferior results than TLD: First, denoising a subband is not an easy task due to its low signal to noise ratio [?]. This is why a lot of efforts have been made over the years to introduce more effective wavelet shrinkage methods. Second, to denoise subbands using TLD, careful parameter adjustment for each subband might be required, which is quite a cumbersome procedure. In the following section, we present our proposed methods to mitigate mDn’s shortcomings. Our final method avoids directly denoising subbands while it holds the advantages of mDn in preserving low-frequency content.

4 Proposed Method

4.1 Mixed Multiscale Denoiser (mmDn): Initial Version

Using wavelet subbands mixing technique [?, ?], we can combine the merits of both single-scale and multiscale denoisers: (i) The main image structure and low frequency content of multiscale denoiser’s output, and (ii) The high frequency content of single-scale denoiser’s output. We call the resultant method the mixed multiscale denoiser (mmDn) method.

The main steps of the initial version of our proposed mmDn method are listed in Algorithm 2. At first, the input image is denoised through both the single-scale denoiser ($D_n(\cdot)$ at line 1) and multiscale denoiser ($mD_n(\cdot)$ at line 2). Then, their outputs are separately decomposed using a K -scale wavelet transform (denoted by $WT_X(\cdot)$). Next, the wavelet subbands mixing is applied (in line 7) to create a new set ($\hat{\mathcal{S}}_X$) which comprises of the approximate subband ($\hat{\mathbf{A}}_M^K$) of $mD_n(\cdot)$ and detail subbands ($\hat{\mathcal{D}}_S^K$) of $D_n(\cdot)$. Finally, the inverse wavelet transform is computed on ($\hat{\mathcal{S}}_X$) to reconstruct the final denoised image ($\hat{\mathbf{X}}_{MM0}$).

Algorithm 2 Initial Version of the Mixed Multiscale Denoiser (mmDn-0)

Input: Noisy image (\mathbf{Y}) and its noise standard deviation (σ), Patch size (p), Number of scales (J), Number of scales (K) for the mixing stage.

Hyper-parameters: Hyper-parameters of Algorithm 1, Type of wavelet transform (WT_X) and its filterbank (af and sf).

-
- 1: $\hat{\mathbf{X}}_S = \text{Dn}(\mathbf{Y}, \sigma, p)$ // $\text{Dn}(\cdot)$: Single-scale denoiser
 - 2: $\hat{\mathbf{X}}_M = \text{mDn}(\mathbf{Y}, \sigma, p, J)$ // $\text{mDn}(\cdot)$: Multiscale denoiser (Algorithm 1)
 - 3: $\hat{\mathcal{S}}_S = \text{WT}_X(\hat{\mathbf{X}}_S, K, af)$ // $\hat{\mathcal{S}}_S$ will be $\{\hat{\mathbf{A}}_S^K, \hat{\mathcal{D}}_S^K\}$
 - 4: $\hat{\mathcal{D}}_S^K = \text{Get the detail subbands from } \hat{\mathcal{S}}_S$
 - 5: $\hat{\mathcal{S}}_M = \text{WT}_X(\hat{\mathbf{X}}_M, K, af)$ // $\hat{\mathcal{S}}_M$ will be $\{\hat{\mathbf{A}}_M^K, \hat{\mathcal{D}}_M^K\}$
 - 6: $\hat{\mathbf{A}}_M^K = \text{Get the approximate subband from } \hat{\mathcal{S}}_M$
 - 7: $\hat{\mathcal{S}}_X = \{\hat{\mathbf{A}}_M^K, \hat{\mathcal{D}}_S^K\}$
 - 8: **Output:** $\hat{\mathbf{X}}_{MM0} = \text{iWT}_X(\hat{\mathcal{S}}_X, K, sf)$
-

In order to analyze mmDn-0 (Algorithm 2) in practice, similar to mDn (Algorithm 1), we need to realize mmDn-0 by specifically determining its hyper-parameters. Here, we again replace $\text{Dn}(\cdot)$ with TLD as its single-scale denoiser. The multiscale denoiser ($\text{mDn}(\cdot)$) is realized as we have described in the previous section, and we use 2D DWT in place of its J -scale wavelet transform ($\text{WT}(\cdot)$). Therefore, there is one more left hyper-parameter in mmDn: The type of a K -scale wavelet transform ($\text{WT}_X(\cdot)$) which is used for subband mixing in mmDn-0. We should notice that this transform is completely independent from the wavelet transform used in ($\text{mDn}(\cdot)$), and we are free to choose the appropriate transform for each stage. To show this, we use two different types of wavelet transforms in place of $\text{WT}_X(\cdot)$. Firstly, we realize $\text{WT}_X(\cdot)$ with a 1-scale ($K=1$) isotropic undecimated wavelet transform (IUWT) with an Astro filter bank [?]. Secondly, we realize $\text{WT}_X(\cdot)$ with a 1-scale ($K=1$) 2D DWT. Since we are using mmDn to construct multiscale extensions of TLD, we call the resultant method mmTLD. The outputs of mmTLD with these transforms are shown in Fig. 2 (e) and (f), respectively. It can be seen that, contrary to the realization of mDn (Fig. 2 (d), denoted by mTLD), our proposed mmDn based methods (mmTLD with IUWT, and mmTLD with DWT) do not suffer from high frequency artifacts while they are able to preserve main structure of the image. The PSNR and SSIM also reflect the superiority of our proposed methods.

To demonstrate why the mixing stage of mmTLD works in practice, let's show its mixing stage when IUWT is for the mixing stage (i.e., $\text{WT}_X(\cdot)$ performs a 1-scale IUWT in Algorithm 2). Then, in line 3, the result of the single-scale denoiser (i.e., Fig. 2 (c)) is decomposed, and then in line 3, the result of the multiscale denoiser (i.e., Fig. 2 (d)) is decomposed. In each level, IUWT decomposes its input into one approximate and one detail subbands. Here, we use 1-scale decomposition. Therefore, the resultant subbands can be easily shown in Fig. 4. In this figure, the first two images illustrate the subbands obtained from TLD's output (denoted by $\hat{\mathcal{S}}_S$ in Algorithm 2) and the second two images show the subbands of MTLD's output ($\hat{\mathcal{S}}_M$). This figure clearly shows that the low frequency content in the approximate

subband of MTLD (c) is much richer than the corresponding subband of TLD (a). Conversely, the detail subband of MTLD (d) is noisier than the corresponding subband of TLD (b). Therefore, line 7 of mmDn-0 (Algorithm 2) takes the strength of TLD and mTLD (Fig. 4 (b) and (c)) and discards their weaknesses (Fig. 4 (a) and (d)) to produce the final output (Fig. 2 (e)).

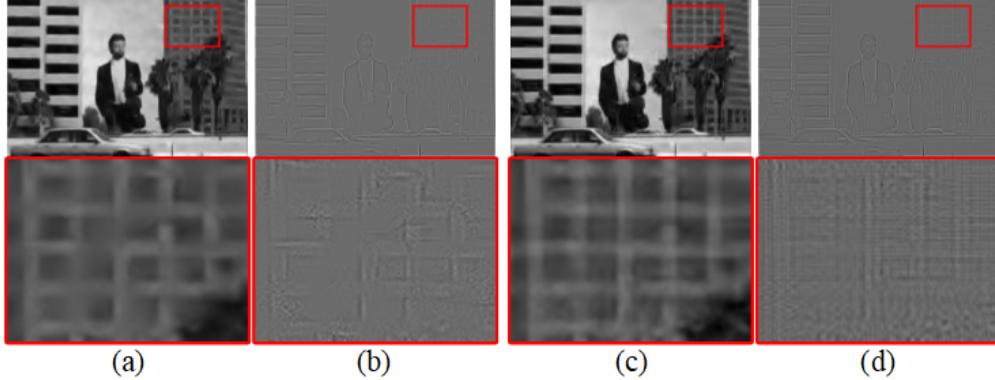


Figure 4: Illustration of the subbands obtained by applying 1-scale ($K=1$) IUWT decompositions on TLD and MTLD outputs'. In the first two images, the TLD subbands are shown: (a) the approximate subband (\hat{A}_S^K), and (b) the detail subband (\hat{D}_S^K). In the last two images, the subbands of MTLD's output are shown: (c) the approximate subband (\hat{A}_M^K), and (d) the detail subband (\hat{D}_M^K).

4.2 Removing the Need to Directly Denoise Detail Subbands and Increasing the Number of Scales

In this section, we want to address two main deficiencies of our initially proposed mmDn (denoted by mmDn-0 in Algorithm 2). First, similar to mDn, our mmDn-0 needs to directly denoise detail subbands. This is because mmDn-0 includes mDn as one of its main steps. Second, if we use more than one decomposition level in the mixing stage (i.e., $K>1$ in Algorithm 2), the output quality tends to the quality of the single-scale denoiser. This is because mmDn-0 only uses the last approximate subband of the multi-scale denoiser (mDn(.)). Therefore, by increasing the number of scales, the contribution of the multiscale denoiser decreases substantially.

As it has been mentioned in the previous section, two independent wavelet transforms are used in mmDn-0 (Algorithm 2): 1) $WT(\cdot)$ is used for denoising stage inside mDn (line 2), and 2) $WT_X(\cdot)$ is used for subbands mixing. There might be some benefits if we are free to choose these transforms in practice. For example, Fig. 2 shows that although both the realization of mmDn-0 (mmTLD with IUWT (e), and with DWT (f)) use DWT for their denoising stages, exploiting IUWT for the subbands mixing stage enhances the output quality in terms of both PSNR and SSIM. However, when these two transforms are selected to be the same, a considerable amount of computations become redundant, and more importantly, it provides a way to bypass directly denoising detail subbands.

In mmDn-0 subbands of mDn are extracted in line 5. When both wavelet transforms ($WT(\cdot)$ and $WT_X(\cdot)$) are the same, the $\hat{\mathcal{S}}$ in mDn and $\hat{\mathcal{S}}_M$ in mmDn-0 becomes equal. Therefore, the inverse wavelet transform at the end of mDn becomes redundant, and we can use $\hat{\mathcal{S}}_M = \hat{\mathcal{S}}$ in mmDn-0. Additionally, in mmDn-0 only the approximate subband

from $\hat{\mathcal{S}}$ is used. This implies that there is no need to denoise the set of detail subbands ($\hat{\mathcal{D}}_M^K$) inside mDn, and we can only pass the last denoised approximate subband to mmDn-0.

According to the above paragraph, when $\text{WT}(\cdot)$ and $\text{WT}_X(\cdot)$ are identical, we can summarize the steps as follows: 1. Denoise the noisy input image \mathbf{Y} , and get its set of detail subbands ($\hat{\mathcal{D}}_S^J$) at scale J , 2. Get the J -th approximate subband (\mathbf{A}^J) of the noisy image, and denoise it ($\hat{\mathbf{A}}^J$), and 3. Using $\hat{\mathcal{S}}_X = \{\hat{\mathbf{A}}^J \cdot \hat{\mathcal{D}}_S^J\}$, compute the inverse wavelet transform. It is evident that, in this procedure, the denoising steps are all performed through the single-scale denoiser ($D_n(\cdot)$). Compared to mmDn-0 (Algorithm 2), this procedure saves the operations needed to denoise detail subbands since they are not used in mmDn-0. Also, the forward and inverse transforms computations in mDn (Algorithm 1) are not required anymore. Therefore, this simplified method reduces the computational cost of mmDn-0 substantially.

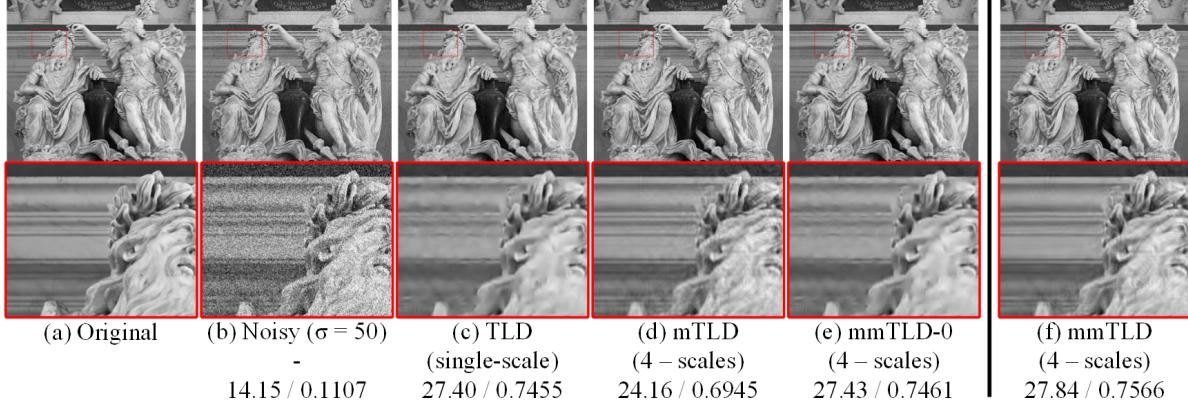


Figure 5: Effect of increasing the number of scales on the performance of multiscale extensions. A noisy image, and the results of a single-scale denoiser (TLD), and its multiscale extensions based on mDn (mTLD), mmDn-0 (mmTLD-0), and mmDn (mmTLD) are shown. TLD (c) substantially damages the low frequency content of the image. mTLD (d) preserves low frequency content, but it suffers from artifacts. We expect that mmTLD-0 (e) will behave like mTLD because it essentially combines high frequency content of TLD and low frequency content of mTLD. However, since it borrows the last approximate subband from mTLD, the borrowed content from TLD dominates the final result when the number of decomposition scales increases in the mixing stage. This weakness is solved in our final version of the mmDn method which is realized here with mmTLD (f). The original image (a) is a high-resolution image (with spatial size of 2040×1740 pixels) from the validation set of DIV2K dataset [?, ?]. The image (a) is corrupted in (b) by adding a white Gaussian noise with $\sigma = 50$. For each method, the PSNR and SSIM results are reported below.

The second issue with mmDn-0 (Algorithm 2) happens when the number of scales increases. Naturally, we expect that by increasing the number of scales (to some extent), more details can be recovered. However, in practice, the opposite happens for mmDn-0. In Fig. 5, it can be seen that using 4-scale transforms to implement mmTLD-0 significantly reduces the effectiveness of this method in comparison with mTLD which realizes mDn (Algorithm 1). This is because mDn effectively uses the information of all decomposition stages. But, mmDn-0 only uses the last approximate subband. Therefore, by increasing the number of scales, more information is borrowed from the single-scale denoiser, and thus, its performance decreases and the overall quality of our proposed mmDn-0 method tends to the single-scale denoiser. This argument can be validated by comparing the quality metric values for TLD and mmTLD-0 results in Fig. 5 (c) and (e).

To fix this issue for mmDn-0, the detail subbands should be processed separately, similar to mDn. However, in contrast to mDn, we want to indirectly denoise detail subbands. This problem can be solved effectively by storing all approximate subbands across all levels of the decomposition scales. Then, we can extract the set of detail subbands at each scale by performing a 1-scale wavelet transform at that scale. This provides a way to simultaneously perform denoising and subbands mixing across all decomposition levels. We list the steps of our final version of the proposed mmDn method in Algorithm 3. Firstly, it stores the noisy image (the finest scale) and all of its approximate subbands in an array (denoted by AP in Algorithm 3). Secondly, the method starts with the last two approximate subbands (AP[$s - 1$] and AP[s]) and denoises them (in lines 7 to 8). Thirdly, the method performs the subbands mixing (line 13). It reconstructs the denoised approximate subband at the finer scale (AP[$s - 1$]) by changing its approximate subband with AP[s] (coarser scale). Finally, these steps are repeated across all scales to reconstruct the final image. It is easy to see that when the number of scales is set to 1 ($J = 1$), mmDn (Algorithm 3) is exactly equivalent to mmDn-0 (Algorithm 2). However, when the number of scales increases, mmDn better preserves the advantages of the multiscale denoising. Fig. 5 (e) and (f) compares the denoising qualities of these two methods for their realizations when $Dn(\cdot) = TLD$ and $WT(\cdot) = DWT$ with 4 decomposition scales.

Algorithm 3 Final Version of the Mixed Multiscale Denoiser (mmDn)

Input: Noisy image (\mathbf{Y}) and its noise standard deviation (σ), Patch size (p), Number of scales (J).

Hyper-parameters: Hyper-parameters of Algorithm 1.

```

1: AP[0] = { $\mathbf{Y}$ }                                // AP is an array to store the approximate subbands.
2: for  $s = 1$  to  $J$  do
3:    $\mathcal{S} = WT(AP[s - 1], 1, af)$            // 1-scale wavelet transform
4:    $\mathbf{A} =$  Get the approximate subband from  $\mathcal{S}$ 
5:   AP[ $s$ ] = { $\mathbf{A}$ }
6: for  $s = J$  to 1 do
7:   AP[ $s - 1$ ] =  $Dn(AP[s - 1], \sigma, p)$       // Denoising finer scale
8:   AP[ $s$ ] =  $Dn(AP[s], \sigma, p)$                 // Denoising coarser scale
9:    $\mathcal{S}_F = WT(AP[s - 1], 1, af)$ 
10:   $\mathcal{D}_F =$  Get the detail subbands from  $\mathcal{S}_F$ 
11:   $\mathcal{S}_C = WT(AP[s], 1, af)$ 
12:   $\mathbf{A}_C =$  Get the approximate subband from  $\mathcal{S}_C$ 
13:   $S_X = \{\mathbf{A}_C, \mathcal{D}_F\}$ 
14:  AP[ $s - 1$ ] = iWT( $S_X, 1, sf$ )
15: Output:  $\hat{\mathbf{X}}_{MM} = AP[0]$ 
```

5 Generalization

Until now, we have discussed two methods for constructing multiscale extensions of a given single-scale denoiser: 1. mDn (Algorithm 1) [?], and 2. our proposed mmDn (Algorithm 3). In the previous sections, we set the single-scale denoiser ($Dn(\cdot)$) to only TLD [?], and mostly used 2D DWT [?]. However, as we have mentioned earlier, the mDn (Algorithm 1) and mmDn (Algorithm 3) methods are quite general and we can realize them with other single-scale denoisers and wavelet transforms. For example, mDn was previously realized by K-SVD denoising method [?], and they called it MS K-SVD [?]. In this section, we further generalize mDn and our proposed mmDn to introduce multiscale extensions for two other 2D image denoising methods. 1. K-SVD denoising [?], and 2. Spatially adaptive

iterative singular-value thresholding (SAIST) [?]. Moreover, we generalize our discussion from 2D to 3D and introduce an efficient multiscale extension for BM4D. In summary, we will demonstrate the results of following methods in Section 6:

Table 1: We have developed multiscale extensions for the following 2D and 3D single-scale denoisers. Multiscale extensions are constructed based on either mDn (Algorithm 1, which was initially proposed in [?]), or our proposed mmDn (Algorithm 3) methods.

Single-scale Method	Multiscale Extension	Comments
TLD [?]	mTLD	In mDn, set $D_n(\cdot)$ to TLD and WT to 2D DWT.
TLD [?]	mmTLD	In mmDn, set $D_n(\cdot)$ to TLD and WT to 2D DWT.
K-SVD denoising [?]	mK-SVD	In mDn, set $D_n(\cdot)$ to K-SVD denoising and WT to 2D DWT. This method was called MS K-SVD in [?].
K-SVD denoising [?]	mmK-SVD	In mmDn, set $D_n(\cdot)$ to K-SVD denoising and WT to 2D DWT.
SAIST [?]	mSAIST	In mDn, set $D_n(\cdot)$ to SAIST and WT to 2D DWT.
SAIST [?]	mmSAIST	In mmDn, set $D_n(\cdot)$ to SAIST and WT to 2D DWT.
BM4D [?]	mmBM4D	In mmDn, set $D_n(\cdot)$ to BM4D and WT to a 3D wavelet transform (e.g., 3D DWT). More details will be explained in the next subsection.

5.1 Mixed Multiscale BM4D (mmBM4D)

In this section, we generalize our discussion from 2D to 3D, and apply our proposed mmDn (Algorithm 3) method to develop a multiscale extension for the well-known BM4D method in order to use it for 3D OCT image denoising. Note that a multiscale extension based on mDn (Algorithm 1, which was initially proposed by [?]) cannot be developed easily for BM4D, since it is not easy to tune BM4D for directly denoising detail subbands (which contain only high frequency content). In contrast to mDn, our proposed mmDn avoids directly denoising detail subbands. Therefore, BM4D can be used naturally in mmDn to denoise the approximate subbands at different decomposition levels.

In order to effectively use mmDn (Algorithm 3) for constructing a multiscale extension for BM4D (namely, mmBM4D), it is worth mentioning that BM4D was previously used multiple times to denoise 3D OCT images [?, ?, ?, ?, ?, ?, ?]. In these works, researchers have used this method in the spatial domain. However, we here show that the performance of BM4D may greatly be enhanced for OCT denoising if we use it in the logarithm domain. Exploiting the logarithm domain to convert the multiplicative to additive noise is a common practice for OCT images, especially for methods based on fixed transform domains such as wavelets [?]. Therefore, we here present our results both in the spatial and logarithm domains.

In Fig. 6 (a) and (b), one slice of a noisy 3D OCT image is shown along with its corresponding registered and averaged image. The averaged image was obtained by a dedicated and costly imaging procedure. This image can be

considered as the ground-truth [?, ?, ?]. The rest of images in the left column ((c), (e), and (g)) show the results of applying BM4D and its multiscale realizations in the spatial domain. Note that, here, we realize mmBM4D with two types of 3D wavelet transforms: 1. 3D DWT [?], and 2. 3D dualtree discrete complex wavelet transform (3D DTCWT) [?]. The first one is denoted by mmBM4D (3D DWT) and the second one is denoted by mmBM4D (3D DTCWT). The right column ((d), (f), and (h)) shows the results of each method in the logarithm domain. As we have mentioned earlier, in contrast to the results reported in [?, ?, ?, ?, ?, ?, ?], it can be seen that exploiting the logarithm domain can enhance the denoising results substantially. We will report more results and discussions in the upcoming section.

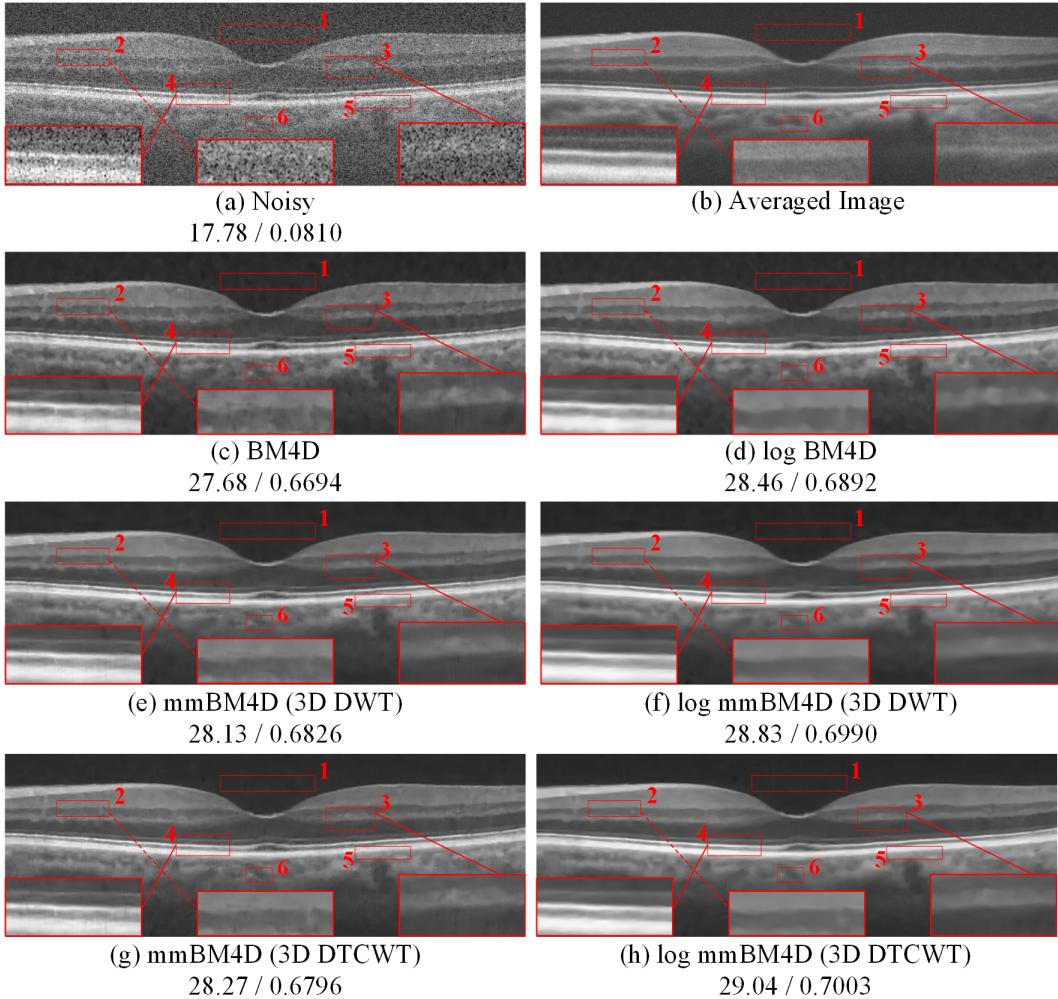


Figure 6: Visual comparison of the denoising results by BM4D and its two multiscale extensions realized with 3D DWT and 3D DTCWT. The transform type is mentioned in the parenthesis. The prefix "log" shows that the corresponding method is applied in the logarithm domain. For each method, the PSNR and SSIM results are reported below.

6 Experimental Results

In this section, we present the qualitative and quantitative results of the methods discussed in the previous section. The compared methods, datasets, and parameters are thoroughly presented in the following subsections. The source code to reproduce the experiments will be made publicly available on <https://github.com/ashkan-abbasi66>.

6.1 2D Natural Image Denoising

6.1.1 Datasets

To evaluate the performance of our proposed multiscale methods with their single-scale counterparts in removing synthetic Gaussian noise from gray-scale images, two datasets are used here: 1. Classic test images [?]: includes *Barbara*, *Boat*, *Cameraman*, *Fingerprint*, *Hill*, *Lena*, *Couple*, *Pentagon*, and *Man*. All images have 512×512 pixels, except for *Pentagon* which has 1024×1024 pixels, and 2. CSR test set [?]: a set of relatively high-resolution test images (e.g., 1423×949 pixels). Note that both of these datasets were employed previously for the assessment of multiscale denoising methods [?, ?]. We corrupt all of the images from these datasets by adding white Gaussian noise with three different standard deviations (15, 25, and 50) from relatively weak to strong noise levels.

6.1.2 Results

In Fig. 7, we show visual comparisons between the results of three single-scale denoisers and their corresponding multiscale extensions which are developed based on mDn (Algorithm 1) and our proposed mmDn (Algorithm 3) methods. This figure shows that, as it is expected, the single-scale denoising methods cannot recover the building facades. To recover the zoomed region, it seems that non-local information might be beneficial because it is a dominant pattern in this figure. However, even SAIST which is able to use these information fails to recover most parts of it. Multiscale processing is helpful in this case. It can be seen that our proposed multiscale extensions (whose names begin with "mm") significantly improve the results in comparison with both their corresponding single-scale and mDn based (whose names are started with "m") methods. Note that mK-SVD was called MS K-SVD in [?].

The observations and explanations can be validated by the average quantitative results reported in Table 2 and 3. The results obtained on the classic test images are reported in Table 2. It can be seen that in almost all cases our proposed subbands mixing based multiscale methods improve significantly over their corresponding baselines. For each multiscale method, the gain over its single-scale counterpart for all three noise levels are averaged and then reported in the last column of the table. Similar to the other studies such as [?, ?, ?, ?, ?, ?], it can be seen that the amount of gains obtained by our proposed mmDn based methods depend on the noise level, baseline methods, and datasets. Generally no gain from multiscale processing is expected when noise standard deviation (σ) is below 20 [?]. Table 2 and 3 show that when noise is relatively weak ($\sigma = 15$), both mDn and mmDn based methods fail to have significant gains except for mmSAIST. But, when noise level increases, the other methods also show considerable gains.

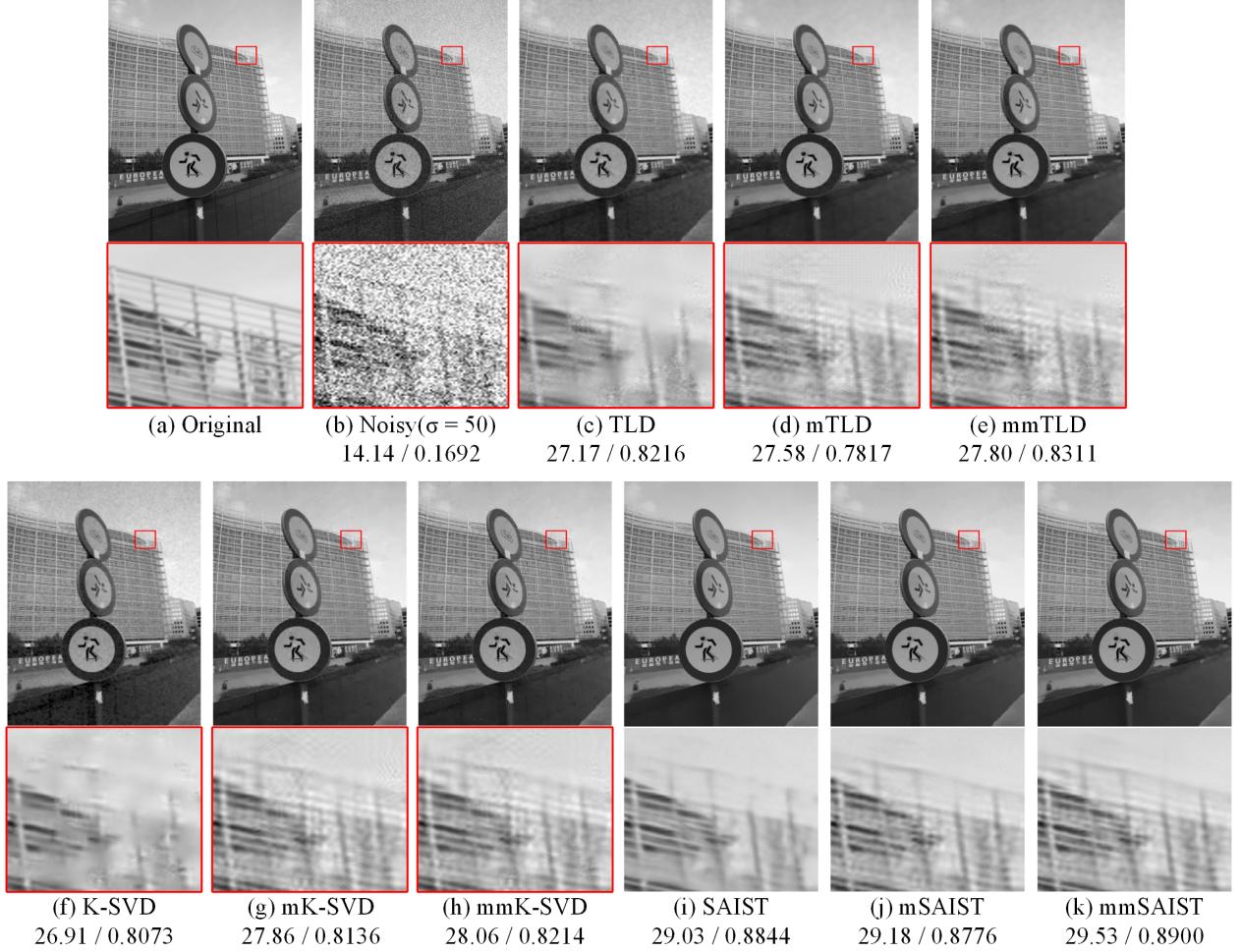


Figure 7: Visual comparison of the results obtained by three groups of single-scale and multiscale denoising methods. In each group, there is one single-scale (TLD, K-SVD, and SAIST) and two multiscale denoising methods whose names begin with "m" or "mm". For each method, the PSNR and SSIM results are reported below.

These tables show that for both datasets, mSAIST leads to unsatisfactory results (on average, it shows either decreased performance or no improvement). This is because SAIST was not originally designed to directly denoise high frequency content. Instead, our mmDn based version of it (mmSAIST), bypasses this difficulty by extracting the denoised detail subbands from the denoised approximate subbands. Therefore, mmSAIST consistently improves the results on both datasets. The best result with mDn based methods achieved by mK-SVD (or MS K-SVD [?]). This method was highly engineered in order to denoise detail subbands. In contrast, our corresponding mmDn based method (mmK-SVD) was just uses the default parameters of the K-SVD denoising [?] to denoise the approximate subbands. Since the nature of the approximate subbands are more similar to the input image itself, almost no tuning is needed for developing mmK-SVD, while it has better performance.

Table 2: Mean of PSNR (dB) of each method for Gaussian image denoising of classic test images. For each group of compared methods, the best PSNR result is shown in **bold**. Also, averaged PSNR gains for each multiscale method is reported in the last column.

Methods	15	25	50	Gain
Noisy Images	24.61	20.17	14.15	-
TLD	32.03	29.45	26.10	-
mTLD	31.87	29.46	26.30	0.02
mmTLD	32.05	29.59	26.43	0.16
K-SVD	32.13	29.58	26.00	-
mK-SVD	31.95	29.65	26.68	0.19
mmK-SVD	32.11	29.81	26.69	0.30
SAIST	32.16	30.03	27.03	-
mSAIST	31.68	29.60	26.98	-0.32
mmSAIST	32.30	30.16	27.20	0.15

Table 3: Mean of PSNR (dB) of each method for Gaussian image denoising of gray-scale high-resolution images of CSR test set. For each group of compared methods, the best PSNR result is shown in **bold**. Also, averaged PSNR gains for each multiscale method is reported in the last column.

Methods	15	25	50	Gain
Noisy Images	24.61	20.17	14.16	-
TLD	33.36	30.49	26.90	-
mTLD	33.31	30.61	27.15	0.11
mmTLD	33.47	30.74	27.33	0.26
K-SVD	33.59	30.57	26.83	-
mK-SVD	33.27	30.63	27.41	0.11
mmK-SVD	33.48	30.81	27.47	0.26
SAIST	33.40	31.28	27.89	-
mSAIST	33.38	31.10	28.09	0.00
mmSAIST	33.80	31.48	28.20	0.30

6.2 3D OCT Image Denoising

The experiments with natural images (Section 6.1.2) show that when noise is not weak, the proposed mmDn based methods can greatly boost the performance over single-scale ones. Noise in OCT images is usually not weak. Therefore, we expect that harnessing the idea of multiscale denoising for OCT images enhances the result. In this section, we demonstrate our OCT denoising results. As we have introduced in Section 5.1, here, the baseline single-scale denoiser is BM4D. Although BM4D is not considered a dedicated OCT denoising method, we show that its extension based on mmDn can boost its performance in a way that it can compete well with various dedicated OCT denoising methods. We also enriched our experiments with OCT layer segmentation results to show that multiscale BM4D is able to preserve layer structures, which are among the most important cues for clinicians.

6.2.1 Datasets

In order to evaluate the 3D OCT denoising methods, we conduct denoising on three publicly available datasets. The first dataset contains 18 spectral domain OCT (SDOCT) images captured by a commercial Bioptigen SDOCT imaging device [?, ?, ?]. Due to providing high signal-to-noise-ratio (SNR) images, this dataset is very popular for evaluating

OCT image reconstruction methods [?, ?]. An example of a high SNR image is illustrated in Fig. 6 (b), which was the result of an averaging and registration procedure [?]. Each subject was either diagnosed as normal or age-related macular degeneration (AMD), and there are four noisy nearby OCT images for each test image, and thus, 3D OCT processing is possible. Specifically, each input volume has a size of $450 \times 900 \times 5$ voxels.

The second dataset contains 13 SDOCT volumes captured from the macular area in the retina by a Topcon 3D OCT-1000 imaging device. The subjects were without any anomalies, and the size of each volume is $650 \times 512 \times 128$ voxels [?]. The third dataset contains SDOCT volumes of size $512 \times 1000 \times 100$ voxels captured by a Biotigen imaging device from 269 AMD and 115 normal subjects. In contrast to the first dataset, there are no high SNR images in the second and third datasets. Therefore, the denoising results are only assessed either visually or through non-reference image quality assessment metrics.

6.2.2 Metrics

We exploit different image reconstruction metrics to assess the quality of denoising. The PSNR, SSIM, mean-to-standard-deviation ratio (MSR) [?], contrast-to-noise-ratio (CNR) [?], and equivalent number of looks (ENL) [?] are used here. In contrast to PSNR and SSIM, the other metrics do not need reference images and are computed locally. Therefore, it is required to select a few regions of interest (ROIs) in the images. The contrast between foreground regions (e.g., red box 2-6 in Fig. 6) and background noise is measured through the CNR metric. The background noise is taken into account using a region from the background area (e.g., red box 1 in Fig. 6). MSR is a sign of good feature recovery without considering the background regions. CNR considers both background and foreground regions, and it is big when ROIs contain prominent features. ENL is a measure of smoothness in the background. Large ENL values indicate stronger noise suppression and smoother background areas.

6.2.3 Denoising Results on the First OCT Image Dataset

In Fig. 8, a visual comparison of the denoising results obtained using 11 denoising methods is presented. All parameters involved in the compared methods were optimally assigned or chosen as described in their reference papers. We use a 5-scale 3D DWT and 3-scale 3D DTCWT to implement our proposed mixed multiscale BM4D extensions (mmBM4D). Note that the number of decomposition levels are fixed during all experiments on this dataset. Increasing the number of decomposition levels hardly improves the results while negatively contributing to the runtime. The compared methods in Fig. 8 are as follows: Sparse K-SVD [?], tensor dictionary learning (Tensor DL) [?], wavelet based denoising of multiple frame data (WMF) [?], multi-scale sparsity based tomographic denoising (MSBTD) [?], nonlocal weighted sparse representation (NWSR) [?], 3D sparsity based simultaneous denoising and interpolation (3D SBSDI) [?], 3D segmentation based sparse reconstruction (3D SSR) [?], and multi-input fully-convolutional network (MIFCN) [?]. Among these methods, only MSBTD, NWSR are single-frame, and the others are 3D denoising methods.

As can be seen in Fig. 8, all 3D denoising methods (except log WMF) outperform 2D denoising counterparts (MSBTD and NWSR). Sparse K-SVD and Tensor DL suppress noise while introducing blurring and artifacts. 3D SBSDI further

reduces the artifacts, but it results in a slightly noisy reconstruction. 3D SSR reduces the noise greatly, however, its results are not natural-looking, due to exploiting segmentation during denoising. MIFCN preserves the layers, however, its result's boundaries are somewhat blurry. Overall, the proposed log mmBM4D methods can effectively reduce noise while preserving many structures, compared to the average image. Comparing different realizations of log mmBM4D through 3D DWT and 3D DTCWT, it can be seen that 3D DTCWT leads to a sharper denoising quality. This is consistent with the results reported in [?], where they found that 2D denoising through CWT better preserves edges in OCT images. Furthermore, the superiority of proposed methods is reflected in both PSNR and SSIM scores.

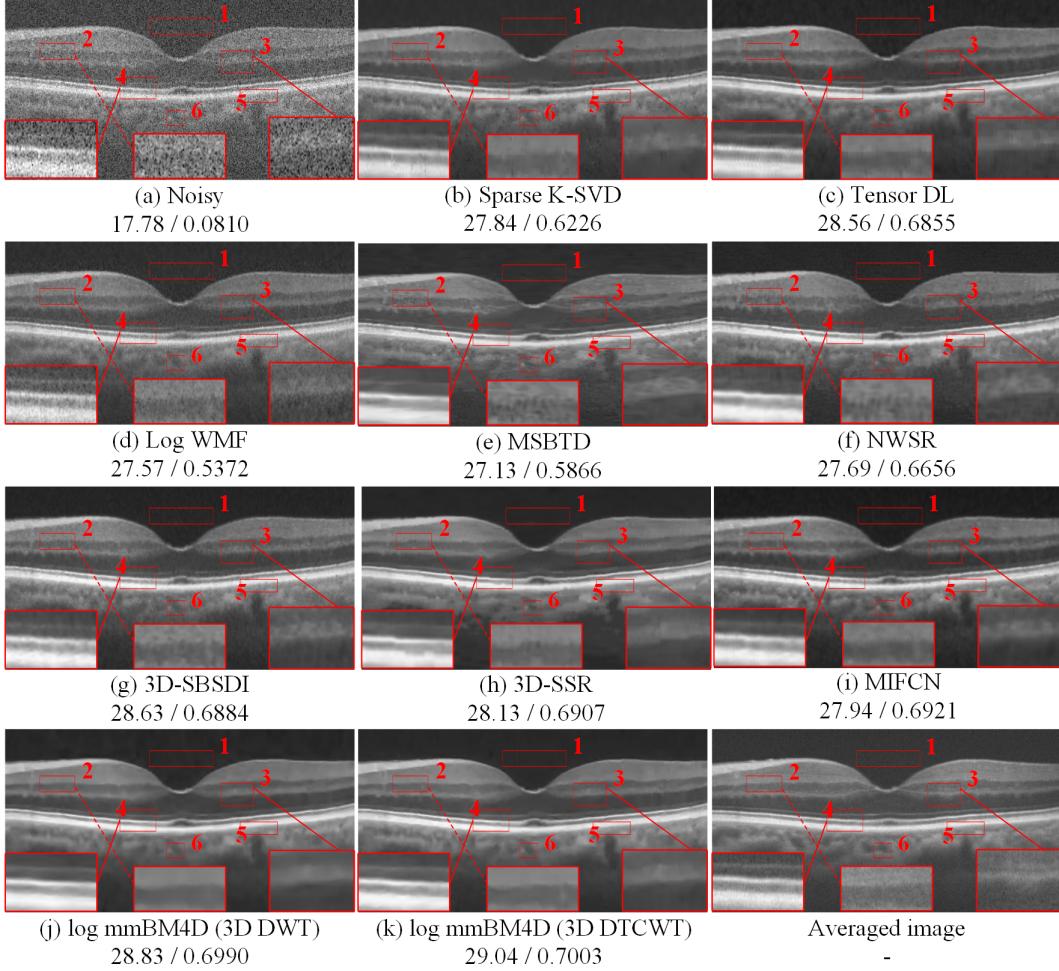


Figure 8: Visual comparison of the 3D OCT denoising results by the compared methods and our proposed multiscale BM4D extensions. For each method, the PSNR and SSIM results are reported below.

The observations and explanations can be validated by the average quantitative results reported in Table 4. This table shows a comprehensive evaluation of the compared methods on the first dataset (Section 6.2.1). In addition to the compared methods in Fig. 8, we also report the average PSNR and SSIM results of two other recent OCT denoising methods: Laplacian scale mixture-generalized extreme value (LSM-GEV) [?] and DnCNN [?]. It can be seen that, in terms of average PSNR, MSR, and CNR, the best result is achieved by 3D SBSDI. However, note that 3D SBSDI is a supervised denoising method which is specifically designed for OCT images. In contrast, our log mmBM4D methods

are unsupervised, i.e., log mmBM4D does not need high SNR images for training its parameters. Considering the hard and costly procedure of obtaining these images, mmBM4D has wider applicability in comparison with supervised methods. Nevertheless, the proposed log mmBM4D's realizations outperform the original BM4D by more than 0.68 dB, and they achieve the best SSIM results, which indicates better edge recovery. Edges in OCT images are of the utmost importance since they play a prominent role in layer segmentation, which is a common analysis over OCT images. We will examine the edge preservation quality of the compared methods in Section 6.2.4.

Table 4: Mean of PSNR (dB), SSIM, MSR, CNR, ENL of each method for denoising 18 OCT images. The best quality metric results obtained by the compared methods are shown in **bold** and the second one is underlined. Additionally, the average processing time per frame (TPF) for denoising OCT volumes with size $450 \times 900 \times 5$ voxels is reported. Furthermore, the last column indicates whether each method is unsupervised or not. The star "*" shows that the output/source code of a method is not publicly available, and the values are copied from the original paper.

Methods	PSNR	SSIM	MSR	CNR	ENL	TPF	Unsupervised?
Noisy Images	17.74	0.0867	3.18	1.61	6.97	-	-
Sparse K-SVD	27.35	0.6145	7.99	3.46	3027.38	7.9	✓
Tensor DL	28.03	0.6734	8.06	3.63	1116.08	4.75	✓
log WMF	26.67	0.5347	7.29	3.52	232.88	0.80	✓
MSBTD	27.08	0.5902	7.43	3.34	1643.06	780.0	✗
NWSR	27.79	0.6686	<u>8.67</u>	3.45	1514.89	66.5	✗
3D SBSDI	28.27	0.6853	8.72	3.75	2572.91	9.5	✗
3D SSR	28.10	0.6905	8.04	3.57	<u>5225.34</u>	-	✗
MIFCN	27.87	0.6879	8.34	<u>3.73</u>	2340.38	0.2	✗
DnCNN*	28.08	0.6910	n\ a	n\ a	n\ a	n\ a	✗
LSM-GEV*	<u>28.25</u>	0.6879	n\ a	n\ a	n\ a	150	✓
BM4D (baseline)	27.52	0.6597	7.14	3.32	1507.68	4.02	✓
log BM4D	27.89	0.6822	7.47	3.43	1664.02	4.02	✓
log mmBM4D (3D DWT)	28.21	0.6925	8.07	3.58	3720.29	11.2	✓
log mmBM4D (3D DTCWT)	<u>28.25</u>	<u>0.6924</u>	8.12	3.65	7530.18	6.5	✓

Table 4 also shows that, in terms of runtime, the fastest method is MIFCN. This method is essentially a neural network which exploits the power of GPU computations. The third fastest method is BM4D. Our proposed log mmBM4D (3D DTCWT) has a reasonable overhead. It adds only 2.5 seconds per frame to incorporate the ability of 3D processing into the original BM4D. However, when 3D DWT is used instead of 3D DTCWT, the running time increases. This difference between the two realizations of mmBM4D is due to the difference in the number of decomposition levels, filterbanks, and wavelet transforms. For log mmBM4D (3D DWT), we have tested different filterbanks including

Daubechies wavelets, Symlets, etc [?]. We empirically found that Daubechies 7 (db7) filterbank is a good choice for this method.

The filter length for db7 is $2 \times 7 = 14$, and the input depth is 5 (Section 6.2.1). Because we have used log mmBM4D (3D DWT) with 5 scales, according to the MATLAB's documentation for "dwt3" function, the approximate subbands have the following depths: 9, 11, 12, 12, 12. This is because when the filter length is a and the input dimension is b , performing 1-scale decomposition results in a depth equal to $\lfloor(a+b)/2\rfloor$. However, the computation for 3D DTCWT is different. 3D DTCWT requires its input dimensions to be a power of two, then it halves the input dimensions on each scale. To perform 3 scales of 3D DTCWT on the input with depth equal to 5, we need to pad it to become a power of two (here, it becomes 8). Then, the approximate subbands have the following depths: 4, 2, 2. Note that for computing 3D DTCWT, the input dimension must be even and greater than or equal to 4 [?, ?]. Therefore, on the second scale, we need to again pad the volume to become 4.

6.2.4 Segmenting Layers of Noisy and Denoised OCT Images

Even though improving the visual quality of OCT images is desirable, in practical applications it is more important to increase the quality of high-level analysis such as layer segmentation and image classification. Here, this question is triggered: Do the proposed methods for restoring OCT images improve the quality of retinal layer segmentation? To answer this question, we have exploited two open source programs: CASEREL [?, ?] and LiveLayer [?]. First, we used CASEREL to automatically segment retinal layers on noisy and denoised OCT images. Then, the results were compared with the segmentation maps created for all test images (Section 6.2.1) by an expert through the manual layer segmentation tab of LiveLayer [?]. The segmented borders are internal limiting membrane (ILM), nerve fiber layer-ganglion cell layer (NFL-GCL), inner plexiform layer-inner nuclear layer (IPL-INL), outer plexiform layer-outer nuclear layer (OPL-ONL), inner segments of photoreceptors-outer segments of photoreceptors (IS-OS), Bruch's membrane-Choroid (BM-Choroid) from top to bottom, respectively. For each image, we compute the unsigned border positioning error for each segmented border by calculating the absolute difference of vertical values (for each pixel) of the segmented and ground-truth borders [?]. Finally, the average errors are summed up and indicated by the sum of absolute error (SAE) for each method.

We have summarized the retinal layer segmentation errors of the compared methods evaluated on the first dataset (Section 6.2.1) in Table 5. The average error made by each method for segmenting each layer is reported separately, and their sum is reported in the last column (SAE). These results reveal that identifying borders in the noisy images leads to the highest error, and all denoising methods can help reduce the segmentation errors to some extent. In terms of SAE, the best method for preserving layer structures is 3D SSR. This is expected because 3D SSR [?] is not only a supervised denoising method, but it also internally exploits an advanced segmentation method through its reconstruction process. Therefore, it is expected that its denoising results preserve layer structures better than the other compared methods. Nevertheless, our proposed log mmBM4D methods can achieve very similar SAEs in comparison with 3D SSR. Also, the results show that incorporating multiscale information into BM4D [?] through the proposed approach

can greatly enhance its results in a way that our multiscale extensions for BM4D are well comparable (both in terms of denoising scores and segmentation errors) with sophisticated methods such as 3D SSR and other dedicated OCT denoising methods.

Here, because we have reported both denoising scores and segmentation errors for all compared methods, it is possible to use the reported results in Table 4 and 5 to draw the plot of PSNR versus SAE in Fig. 1. This figure shows that the proposed multiscale extensions for BM4D can achieve a good denoising quality score while it also preserves layer structures. Furthermore, it reveals that while generally there is a correlation between quality scores and segmentation errors, there might be some contradictions in practice. For example, 3D SBSDI achieves the best result in terms of PSNR. However, probably due to remaining noise and artifacts in its results (e.g., Fig. 8 (g)), its corresponding segmentation error is higher than what is expected.

Table 5: Average border position errors obtained by segmenting retinal layers from OCT images denoised by the compared methods. Sum of the errors for each method is reported in the last column (SAE).

Methods	ILM	NFL-GCL	IPL-INL	OPL-ONL	ONL-IS/OS	BM-Choroid	SAE
Noisy	2.090	7.018	5.357	5.233	2.136	2.165	24.000
log WMF	1.706	2.415	3.256	4.466	1.649	1.759	15.251
MSBTD	1.332	2.159	3.118	4.280	1.219	1.673	13.781
NWSR	1.355	2.178	3.250	4.241	1.225	1.800	14.049
3D SBSDI	1.316	3.091	3.064	3.863	1.306	1.603	14.244
3D SSR	1.211	1.914	2.817	3.661	1.034	1.754	12.391
MIFCN	1.300	2.493	2.994	3.809	1.138	1.803	13.536
log BM4D	1.396	2.400	2.826	3.961	1.226	2.060	13.869
log mmBM4D (3D DWT)	1.381	2.023	2.635	3.536	1.167	1.753	12.496
log mmBM4D (3D DTCWT)	1.432	2.025	2.594	3.567	1.147	1.658	<u>12.422</u>

6.2.5 Denoising Results on the Second and Third OCT Image Datasets

In this section, we further evaluate the applicability and performance of our proposed mixed multiscale methods on two other datasets (Section 6.2.1).

As we have mentioned earlier, the second dataset contains 13 volumes, each one has 128 slices. To quantitatively and qualitatively assess the denoising quality of the compared 3D denoising methods, we randomly selected three slices from each volume. Then, each slice is denoised through the compared method by exploiting 7 slices around it (i.e., the input volume size is $650 \times 512 \times 8$ voxels). The denoising results of the original BM4D and our proposed multiscale extensions are shown in Fig. 9. Furthermore, the average MSR, CNR, and ENL metric values (computed over $13 \times 3 = 39$ images) are presented in Table 6. It is evident that our proposed mixed multiscale BM4D extensions greatly improve over the original BM4D method, for which the output is dominated by artifacts.

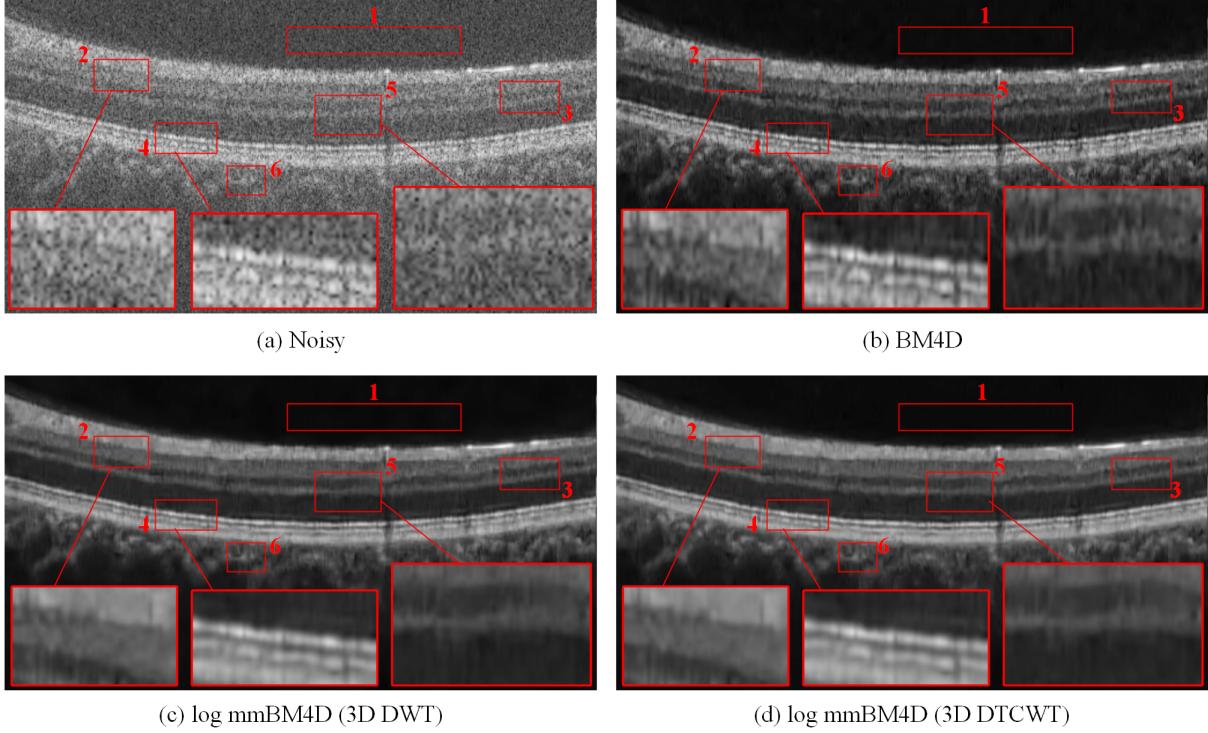


Figure 9: Visual comparison of the 3D OCT denoising results obtained by denoising a test image from the second dataset through BM4D and the proposed multiscale BM4D extensions.

Table 6: Mean of MSR, CNR, ENL of each method for 3D denoising 39 TOPCON OCT images. The best quality metric results obtained by the compared methods are shown in **bold** and the second one is underlined. Additionally, the average processing time per frame (TPF) for denoising OCT volumes with size $650 \times 512 \times 8$ voxels is reported in the last column.

Methods	MSR	CNR	ENL	TPF
Noisy Images	4.24	1.86	29.82	-
BM4D (baseline)	6.83	2.90	<u>3335.03</u>	4.8
log BM4D	7.87	3.17	4282.53	4.8
log mmBM4D (3D DWT)	<u>8.01</u>	<u>3.20</u>	<u>6067.87</u>	5.6
log mmBM4D (3D DTCWT)	8.15	3.25	8994.30	5.0

In the third dataset, the volumes are relatively high-resolution ($512 \times 1000 \times 100$ voxels). In contrast to the experiments on the first and second datasets, here, we fed the whole volume to the denoising method. Due to the computational costs or intrinsic method limitations (e.g., MIFCN only accepts 5 slices per volume, supervised methods are not trained on this dataset), the application of most compared denoising methods reported in Section 6.2.3 to denoise a whole volume is not straightforward. However, due to the flexibility and computational advantages of BM4D and subsequently the

proposed multiscale extensions, we can feed a whole volume to BM4D or log mmBM4D at a reasonable runtime. A visual comparison of the denoising results for one test slice from this dataset is shown in Fig. 10. We also report the average metric values for denoising 2 normal and AMD OCT volumes at Table 7. These metric values were computed based on five random slices per volume.

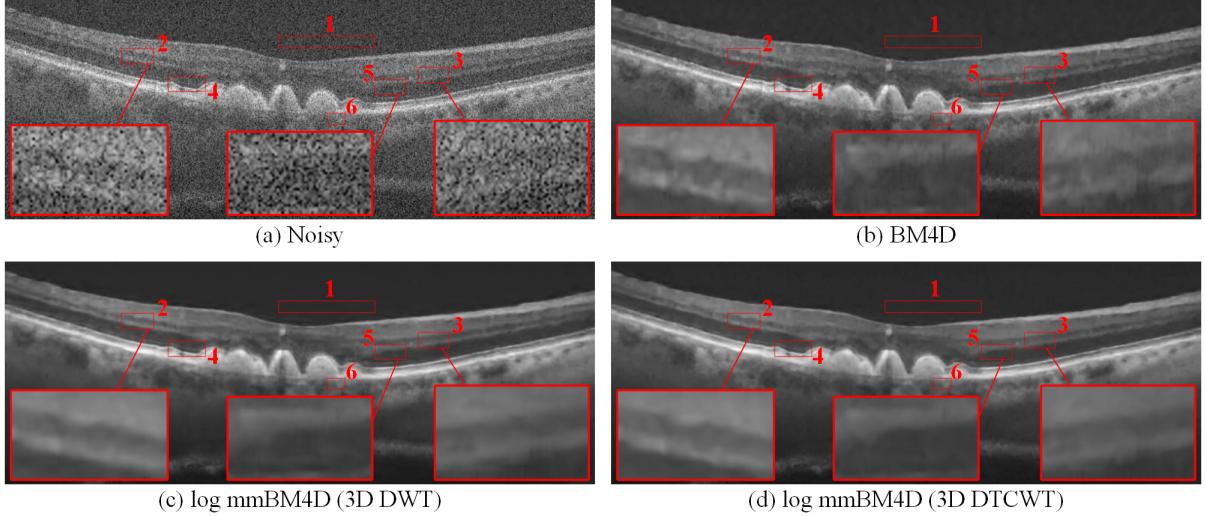


Figure 10: Visual comparison of the 3D OCT denoising results obtained by denoising a test image from the third dataset through BM4D and the proposed multiscale BM4D extensions.

Table 7: Mean of MSR, CNR, ENL of each method for denoising the third OCT image dataset. The best quality metric results obtained by the compared methods are shown in **bold** and the second one is underlined. Additionally, the average processing time per frame (TPF) for denoising OCT volumes with size $512 \times 1000 \times 100$ voxels is reported in the last column.

Methods	MSR	CNR	ENL	TPF
Noisy Images	3.05	1.31	6.73	-
BM4D (baseline)	7.77	3.13	2335.46	17.95
log BM4D	8.40	3.30	2375.00	17.95
log mmBM4D (3D DWT)	<u>8.25</u>	<u>3.30</u>	9643.98	20.51
log mmBM4D (3D DTCWT)	8.68	3.41	<u>7395.83</u>	20.86

7 Conclusion

This paper presents a method for constructing multiscale denoising methods from single-scale ones. We start with the subbands denoising approach which was initially proposed by multiscale K-SVD [?]. Then, by emphasizing on its shortcomings, we propose the mixed multiscale method. Our proposed method avoids directly denoising detail

subbands, and instead, extracts those subbands from the approximate subbands at finer level. Therefore, it is possible to combine the advantages of single-scale and multiscale methods in an efficient way. We have exploited the proposed method to construct multiscale extensions for multiple 2D and 3D denoising methods. The experiments show that our multiscale extensions can significantly boost their single-scale counterparts. In Particular, we propose a multiscale extension for the well-known BM4D benchmark denoising method which can outperform several dedicated recent OCT image denoising methods. We do not restrict our evaluations to the usual quantitative and qualitative image quality assessments experiments, and further, we evaluate the retinal layer preservation quality of the proposed OCT denoising method by conducting comprehensive segmentation experiments.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References