Scale-aware decomposition of images based on patch-based filtering

Ph.D. Dissertation

Bora Jin

Department of Electrical and Computer Engineering

Seoul National University

ABSTRACT

This dissertation presents an image decomposition algorithm based on patch-based filtering, for splitting an image into a structure layer and a texture layer. There are many applications through the decomposition because each layer can be processed respectively and appropriate manipulations are accomplished. Generally, structure layer captures coarse structure with large discontinuities and a texture layer contains fine details or proper patterns. The image decomposition is done by edge-preserving smoothing where structure layer can be obtained by applying smoothing filters to an image and then a texture layer by subtracting the filtered image from the original. The main contribution of this dissertation is to design an efficient and effective edge-preserving filter that can be adapted to various scales of images. The advantage of the proposed decomposition scheme is that it is robust to noise and can be extended to a noisy image decomposition, while conventional image decomposition methods cannot be applied to a noisy image decomposition and conventional image denoising methods are not suitable for image decomposition.

To be specific, a patch-based framework is proposed in this dissertation, which is efficient in image denoising and it is designed to smooth an image while preserving details and texture. Specifically, given a pixel, the filtering output is computed as the weighted average of neighboring pixels. For computing the weights, a set of similar patches is found at each pixel by considering patch similarities based on mean squared error (MSE) and other constraints. Then, weights between each patch and its similar patches are computed respectively. With the patch weights, all the pixels in a patch are updated at the same time while adapting to the local pixel weight. For better edge-preserving smoothing, the proposed algorithm utilizes two iterations which are performed through the same smoothing filter with different parameters. Also kernel bandwidth and the number of similar patches are tuned for multi-scale image decomposition.

The proposed decomposition can be applied to many applications, such as HDR tone mapping, detail enhancement, image denoising, and image coding, etc. In detail enhancement, the proposed smoothing filter is utilized to extract image detail and enhance it. In HDR tone mapping, a typical framework is used where the smoothing operator is replaced by the proposed one to reduce contrast range of a high dynamic range image to display it on low dynamic range devices. For image denoising, a noisy input is decomposed into structure/texture/noise and the noise layer is discarded while the texture layer is restored through the histogram matching. Also a novel coding scheme named as "structure scalable image coding scheme" is proposed where structure layer and salient texture layer are encoded for efficient image coding. Experimental results show that the proposed framework works well for image decomposition and it is robust to the presence of noise. Also it is verified that the proposed work can be utilized in many applications. In addition, by adopting the proposed method in decomposition of a noisy image, both image denoising and image enhancement can be achieved in the proposed framework. Furthermore, the proposed image coding method reduces compression artifact and improve the performance of image coding.

Key words: image decomposition, multi-scale decomposition, image denoising, image enhancement, image coding

Student number: 2010-30232

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Image decomposition

In the field of image processing, there are some applications that decomposes an image into a structure layer and a texture layer and then processed them separately and differently based on their properties that the structure layer captures coarse structure with large discontinuities, whereas a texture layer contains fine details or proper patterns. So it is useful to process each layer with appropriate processing method adapted to its property. For example, the techniques of high dynamic range (HDR) tone mapping aims to reduce the high dynamic range of image within the range of low dynamic range display devices [8, 9]. For this, it can be one of the easiest ways to compress the range linearly but it results in removal of fine details. Consequently, most techniques decompose an image into a base layer (structure layer) and a detail layer (texture layer) and reduce the range of the base layer only. Then the compressed base layer is recombined with the detail layer. Image decomposition can be applied to image detail enhancement, while detail layer is

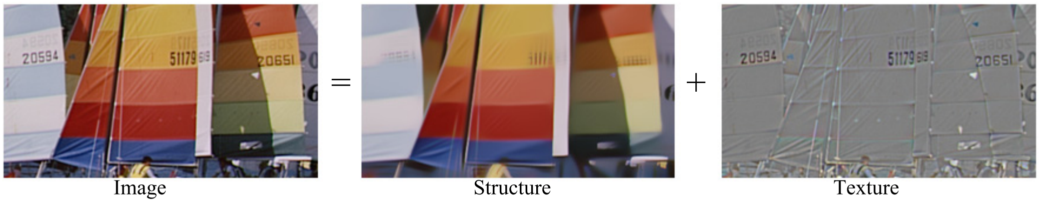


Figure 1.1: image decomposition

extracted via image decomposition and amplified to emphasize detail components [9, 1,3]. In the case of image compression applications, since the noise can be degraded the compression performance [10], image decomposition can also be utilized for image and video compression. Regarding noise as a kind of texture, better performance is achieved by compressing coarser structure without texture.

An example of image decomposition is shown in Fig. 1.1, for which smoothing algorithms are often used as shown in Fig. 1.2. Specifically, given an image I, structure layer I_S can be obtained by applying smoothing filters to an image and then a texture layer I_T by subtracting the filtered image from the original one as

$$I_T = I - I_S. \tag{1.1}$$

Thus, a smoothing filter is the most important in this process and many smoothing algorithms are suggested for years. Among them, one of the simplest methods is the Laplacian pyramid in which an image is smoothed out with Gaussian filter [9].

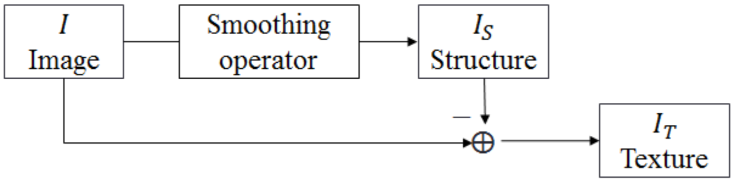


Figure 1.2: image decomposition via smoothing

However, as a linear filter like Gaussian filter blur region around the edges, which produces halo effect and gradient reversal artifact. Hence, many researchers focus on edge-preserving smoothing to reduce these artifacts. Bilateral filter (BLF) [11,12] is one of the popular non-linear filters due to its simplicity and effectiveness. It takes advantages of both spatial and photometric range weights and maintains edges well during smoothing process. So in [8], image decomposition based on BLF is used to reduce the contrast of HDR images. Also in [13], a set of images under varying light conditions is decomposed into different scales by applying BLF recursively. However, it is pointed out that BLF removes small textures effectively, but it is limited to increase spatial and range parameters progressively for removing large textures [1].

To overcome this, weighted least squares (WLS) framework is prosed [1], in which the edge-smoothing problem is formulated as an optimization framework. The WLS-based operator smoothes an image except the region with large gradient and thus it decomposes an image in edge-preserving manner. Another optimization framework, total variation (TV) model [14] is also adopted in image decomposition [15, 16, 2]. Standard TV model gives penalties to large gradient magnitude and it performs efficient texture removal. So many algorithms based on TV model are studied by adopting different regularization terms. Among them, relative total variation model (RTV) [2] introduces a novel regularization by combining a windowed total variation and inherent variation, and its decomposition produces more flattened structure layer than WLS without any assumption of the type of textures.

Else an image can be decomposed based on local extrema [3]. This method treats detail as oscillation between maxima and minima of intensities. So it locates both local maxima and minima in an image and constructs the maxima and minima envelopes by interpolating maxima and minima separately. Then the structural base layer is computed as the mean of extrema envelopes. In addition, rolling guidance filter [4] extracts structure from textured image through small structure removal and edge recovery steps. By applying Gaussian filter it removes small structure. Then it recovers smoothed edges with the output of previous iteration as a guidance. Patch-based image decomposition is also presented in [6] where patch weights depend on region covariance.

1.2 Image enhancement

The image enhancement is to make an image visually more pleasant to human visual system. To be specific, images can be more vivid through the edge-sharpening, contrast enhancement, detail enhancement, high-dynamic-range (HDR) image tone mapping and so on. Among them, the edge-sharpening accompanying detail enhancement is also an important topic in image processing, which increases the visual quality of the images [9, 17, 18].

The classical detail enhancement is to use the "unsharpening filter," where an image is low pass filtered and subtracted from the original, which leaves the high band signal that contains the edges and detail. The high band signal is then amplified and added to the low pass filtered image, which is the edge-sharpened result. When the Gaussian filter is used for the low pass filtering, its subtraction from the original is the Laplacian of Gaussian, and thus the subband images so formed are called the Laplacian Pyramids [9]. Like this, many algorithms for detail enhancement perform extracting detail first based on image decomposition. That is, after an image is decomposed into a structure layer and a detail layer, the detail layer is

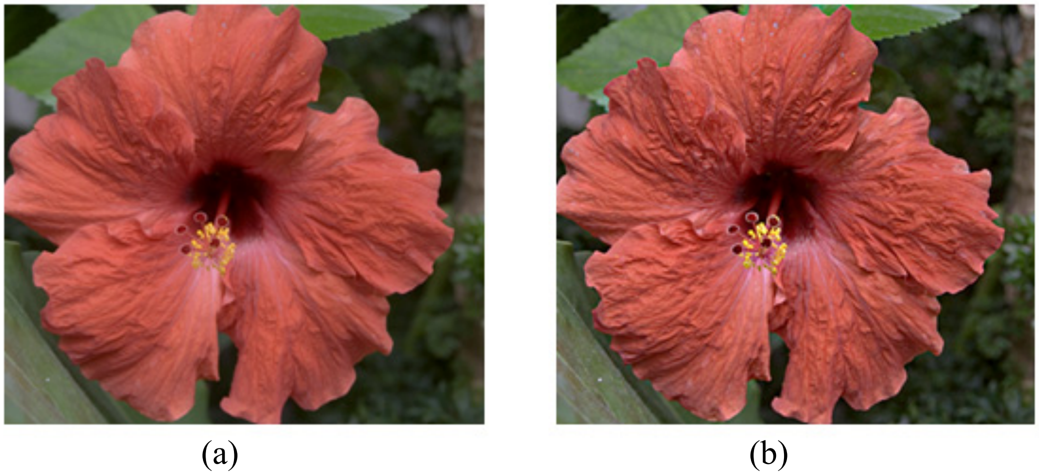


Figure 1.3: image detail enhancement (a) original image (b) detail-enhanced image

boosted and recombined with a structure layer. Consequently, the performance of this enhancement processes are in conformity with that of image decomposition. Fig. 1.3 is an example of detail-enhanced image.

Another form of image enhancement is contrast enhancement. For enhancing image contrast, histogram equalization is studied for decades and it is still a popular method [19, 20]. In histogram equalization, the gray levels of the image are remapped to stretch the histogram of an image or make it distributed equally and many adaptation schemes are proposed [21, 22].

It is becoming important to capture a scene with high dynamic range (HDR) cameras and display with HDR. Their exchange with low dynamic range (LDR) devices is also significant. Generally, an HDR tone mapping aims at reducing contrast range to display a high dynamic range image on low dynamic range devices [8, 9, 1]. In the literature, a simple form of tone mapping is based on global mapping function which maps the intensities in high dynamic range into ones in narrower range [23,23]. However, managing global contrast causes the loss of image detail while compression. Accordingly, as considering local contrast management, algorithms based on image decomposition are proposed [24, 9, 8]. For this, an image is decomposed into a structure layer and one or more detail layers first. Then the structure layer is compressed and recombined with texture layers. It is often performed by multiscale decomposition to enable detail enhancement simultaneously. High detail and reduced dynamic range can be produced with image decomposition. In [24], image decomposition is performed by a low curvature image simplifier (LCIS) which is inspired by anisotropic diffusion and this LCIS-based tries to preserve scene boundaries. Bilateral filter and Laplacian filter are used in HDR tone mapping, too [8, 9].

1.3 Image denoising

Image denoising is the process of restoring images corrupted by various noises such as additive white Gaussian noise (AWGN), Poisson noise, impulse noise and so on. It is fundamental in image formation, transmission and display systems, and thus a huge number of methods have been developed over decades. However, since an image can be corrupted by various noises, it is a challengeable problem and there is still further room for improvements. The overview of classical linear filtering and some of recently developed nonlinear methods can be found in [25], where the relations of different nonlinear methods are also well explained.

Generally, noise model is defined as

$$y = x + n \tag{1.2}$$

where y is a noisy image as the observation, x is the original one, and n is noise. As shown in Fig. 1.4, the purpose of image denoising is to restore x from given y, and peak-signal-to-noise ratio (PSNR) and mean squared error (MSE) are used to assess visual quality of image denoising. With point of view in image decomposition, image denoising decomposes a noisy image into an original one and noise. It shares the same issue with image decomposition, i.e., edge-preserving smoothing. So for suppressing the noise while keeping the important edges and structures in the image, many classical denoising methods utilize filtering schemes. For example, median filtering [26,27] is presented and nonlinear filtering methods such as bilateral filtering (BLF) [11], nonlocal means (NLM) filtering [28] and block matching 3D (BM3D) filtering [5] have been developed. These methods can be classified into spatial denoising and transform denoising according to where their processes are performed.

1.3.1 Spatial denoising

Spatial denoising manipulates pixel values directly to denoise an image. Many spatial denoising methods take the form of weighted averaging pixel values and the denoised intensity estimate at pixel $\mathbf{p}$ is defined as

$$\hat{I}(\mathbf{p}) = \frac{1}{W} \sum_{\mathbf{q}} w(\mathbf{p}, \mathbf{q}) I(\mathbf{q}) \tag{1.3}$$

where $w_{\mathbf{p},\mathbf{q}}$ is the weight between pixel p and q and W is a normalizing factor, i.e., $W = \sum_q w(\mathbf{p}, \mathbf{q})$. In spatial denoising, local self-similarity or nonlocal self-similarity is used. The underlying idea of methods based on nonlocal self-similarity is that similar patches with a given patch can be found in different regions within the image and $\mathbf{q}$ can be the entire pixel. On the other hand, only local neighbors are used as $\mathbf{q}$ in the methods based on local self-similarity. The BLF [11] is one of

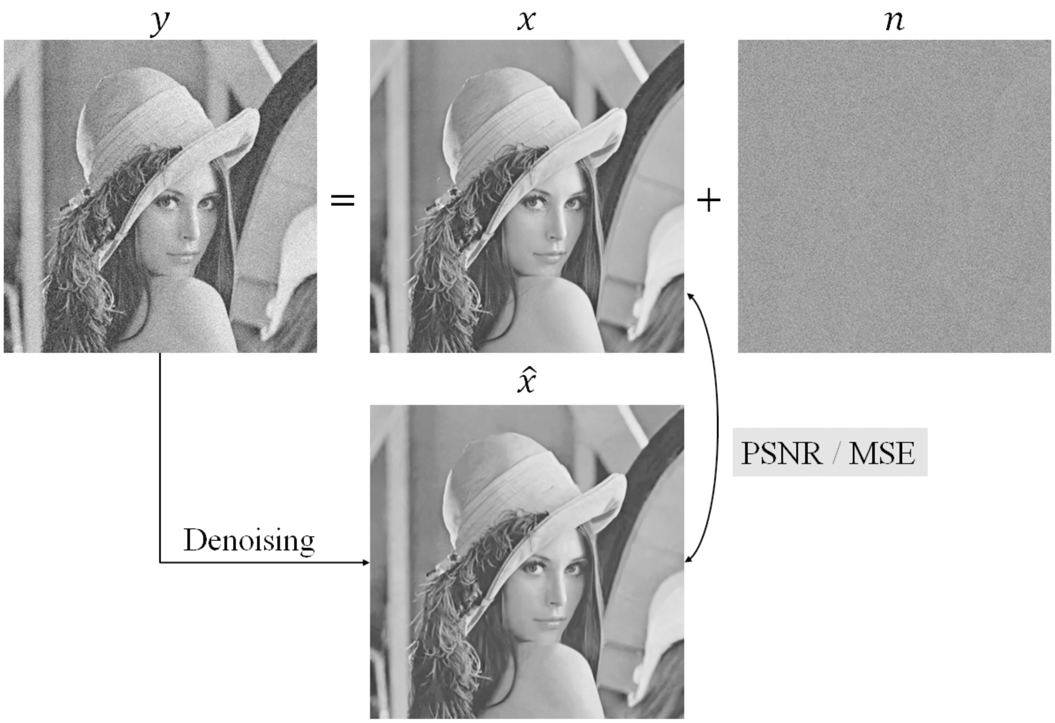


Figure 1.4: image denoising

local similarity and its weight is defined as

$$w(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma_d^2}\right)\exp\left(-\frac{\|I(\mathbf{p})-I(\mathbf{q})\|^2}{2\sigma_r^2}\right) \tag{1.4}$$

where $\mathbf{p}$ and $\mathbf{q}$ are pixel positions, $I(\mathbf{p})$ and $I(\mathbf{q})$ are intensity values at $\mathbf{p}$ and $\mathbf{q}$, and $N(\mathbf{p})$ is the set of neighboring pixels of $\mathbf{p}$.

NLM [28] and BM3D [5] are typical denoising based on nonlocal self-similarity. The weight function in NLM utilized similarity between patches and it is designed as

$$w(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{\|\Phi_{\mathbf{p}}-\Phi_{\mathbf{q}}\|}{h^2}\right) \tag{1.5}$$

where $\Phi_{\mathbf{p}}$ is the set of pixels within the patch centered at $\mathbf{p}$ and h is bandwidth of the filter. As each patch is used for computing the weight, NLM usually obtains better denoising results than pixel-based method like BLF but it is time-consuming process.

1.3.2 Transform domain denoising

The transform denoising methods are based on the idea that signals can be represented well as a linear combination of basis functions. It takes advantage of decomposition which make noise separated partly from the original image. Hence, image denoising can be achieved by manipulating transform coefficients and now it is important which transform can be selected for denoising. Among various transform such as the discrete cosine transform (DCT), the wavelet transform is the most-used one since it contains both spatial and spectral characteristic of an image. After an image is transformed into wavelet domain, large coefficients are regarded as the original image and small coefficients are considered as noise. Hence, thresholding techniques such as hard thresholding and soft thresholding are adopted [29,30]. Another method proposed in [27] takes Gaussian scale mixture (GSM) model to fit the neighboring coefficients and applies the Bayesian least squares (BLS) to manipulate the coefficients.

1.3.3 benefits of combined image decomposition and image denoising

A simple denoising example with a synthetic 1-D signal is shown, which motivates to perform noise removal in a decomposed image. It is performed to find if image decomposition increases the performance of filtering-based denoising. So BLF is applied to Laplacian subbands. Note that the kernel of the bilateral filter in eq. (2.4) is consisted of two terms, i.e., geometric and photometric terms. From this, it can be seen that the photometric weights would be kept large for wider area when a pixel $\mathbf{p}$ is in the flat area where $I(\mathbf{p})$ and $I(\mathbf{q})$ are similar, and hence many neighboring pixels can contribute for the denoising. On the contrary, when the pixel is in the non-flat area where $\|I(\mathbf{p}) - I(\mathbf{q})\|$ is large, the photometric weights diminish and thus the neighboring pixels less contribute for the denoising when compared with the case of flat area. To be specific, Fig. 1 shows an example for this, where Fig. 1(a) is the original noisy signal, and Fig. 1(b) and (c) show its Laplacian low and high subbands. It can be seen that the slope area in the original signal becomes a flat area and thus more noise reduction can be gained here due to the widened area of photometric similarity. For these three signals, the BLF is applied with the same parameters, $\sigma_d^2 = 4$ and $\sigma_r^2 = 49$, and the denoising results of overall area and specific areas (flat, edge, and slope areas as in Fig. 1(a)) are summarized in Table 1. In the table, "original" column is the mean square error (MSE) of a noisy signal of length 256 in Fig. 1(a), "original BLF" represents the BLF denoising of this signal, and "subband BLF" means to apply the BLF to both of low and high band signals in Fig.1 (b) and (c) and then adding them. Also, the "overall" means the MSE of the overall region of the signal in [0,255], and flat, edge and slope represent the areas as defined in Fig. 1(a). At the first row, it can be seen that the BLF of the signal greatly reduces the noise variance (from 9.12 to 3.24) and the subband BLF decreases the variance (to 2.11) a little bit more. When comparing the area wise results, it can be observed that the ratio of the denoising gain is the most significant in the slope area, whereas the subband BLF has worse gain at the edge area. Hence this supports the motivation that the an image decomposition like Laplacian subbands has the possibility of increasing the self-similarity-based filtering performance at the

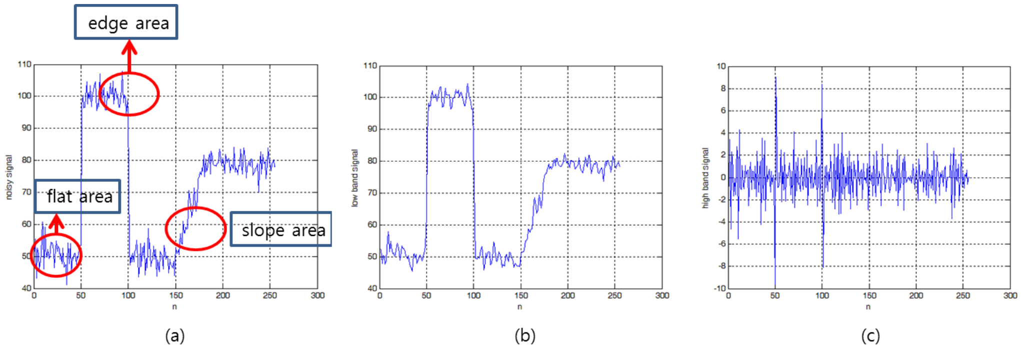


Figure 1.5: An example of the Laplacian subband decomposition for a 1-D signal. (a) original signal (b) low band signal (c) high band signal

slope areas which becomes flat in the Laplacian subbands. Also, the performance around edge area should be improved.

Table 1.1: Mean squared errors of the BLF results for the original and subband signals.

area	original	original BLF	subband BLF
overall ([0,255])	9.12	3.24	2.11
flat ([0,39])	10.8	4.58	2.86
edge ([80,119])	6.11	1.55	2.64
slope ([150,189])	8.53	3.45	1.67

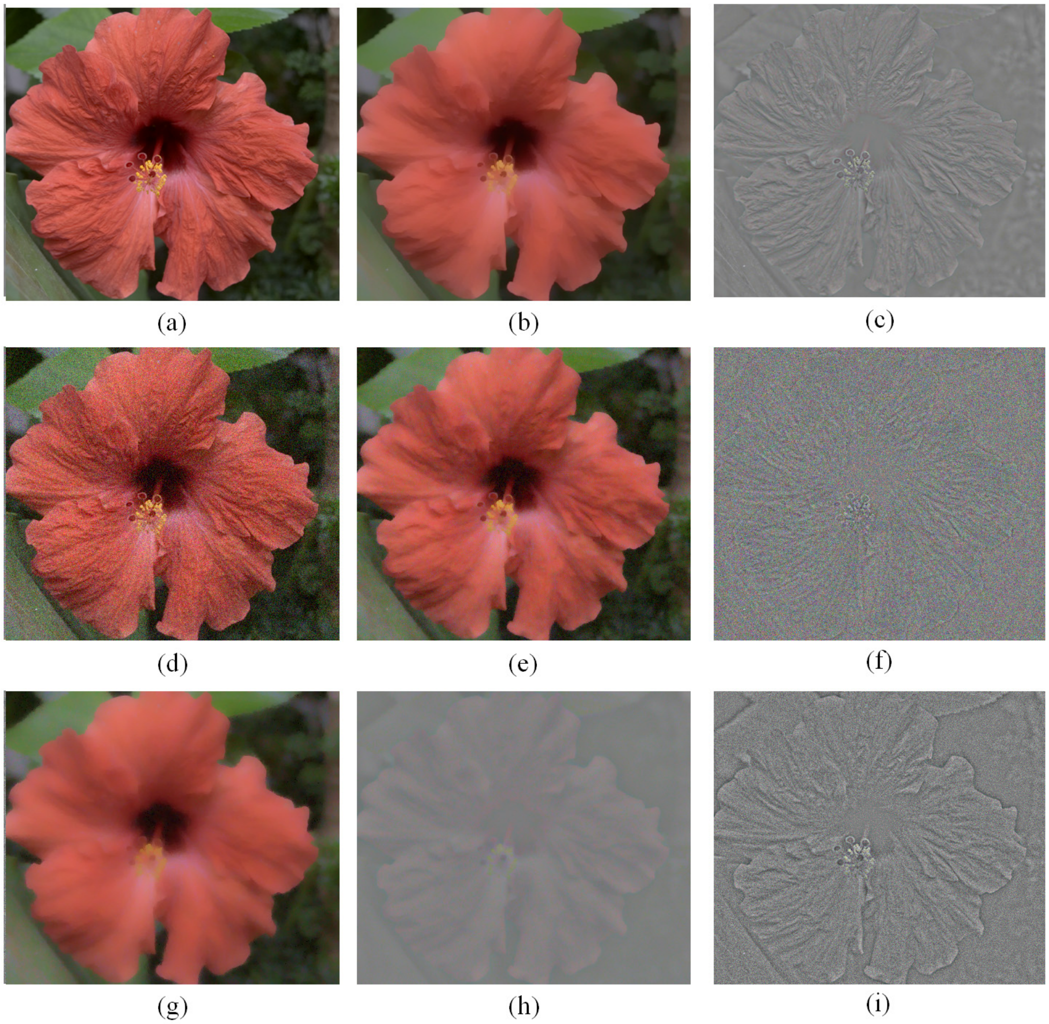
1.4 Summary

The motivation of this research is based on the concern that image decomposition and image denoising are not separable problems since noise can be regarded as small scales of texture. In other words, image denoising is to decompose a noisy image into structure, texture and noise, and to eliminate noise layer. Also they share common goals, i.e., edge-preserving smoothing. Nevertheless, each field is specialized and developed independently.

Even though conventional image decomposition methods such as TV model based methods [15,16,2] show their excellence in image decomposition, they cannot separate texture and noise. It is difficult to settle this problem by varying the parameters in those methods. For example, TV model [14] which is used in many decomposition schemes, takes the form of

$$\arg\min_{I_S} \sum_{\mathbf{p}} \{ \frac{1}{2\lambda} (I_S(\mathbf{p}) - I(\mathbf{p}))^2 + |\nabla I_S(\mathbf{p})| \} \tag{1.6}$$

where $\sum_{\mathbf{p}} |\nabla I_S(\mathbf{p})| = \sum \{ |(\partial_x I_S)_{\mathbf{p}}| + |(\partial_y I_S)_{\mathbf{p}}| \}$ is the sum of gradient magnitudes. While TV model-based method chooses different form of regularization term, this means the regularization term gives penalties to magnitudes of gradient. That is, the model is designed to preserve large gradients and suppress small gradients. Even so they are robust to extract structure in a noisy image, noise and texture can be separated by this framework. Fig. 1.6 and 1.7 show failure cases of decomposing texture and noise where WLS operator [1] and RTV-based method [2] are applied to extracted texture layer by adjusting parameters. Fig. 1.6 (a) is a noise-free original image, (b) its structure with $\alpha = 1.2, \quad \lambda = 0.4$ and (c) its texture. Also Fig. 1.6 (d) is a noisy image with additive white Gaussian noise $\sigma = 20$, (e) its decomposed structure and (f) extracted. Since Fig. 1.6 (d) still contains noise components, parameters are increased to be $\alpha = 1.2, \quad \lambda = 3.2$ to the extracted noise and obtain (g) structure. Then one more smoothing is performed on the extracted texture from texture + noise layer. Fig. 1.6 (h) is the texture after smoothing texture layer and (i) is the noise layer, ie., (d) can be exactly reconstructed by combining (g), (h), (i). Fig.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 1.6: Structure/texture/noise decomposition via WLS operator [1] (a) original image (b) structure ($\alpha = 1.2, \quad \lambda = 0.4$) (c) texture extracted by subtracting (b) from (a) (d) noisy image degraded by additive white Gaussian ($\sigma = 20$) (e) structure of (d) ($\alpha = 1.2, \quad \lambda = 0.4$) (f) texture of (d) ($\alpha = 1.2, \quad \lambda = 0.4$) (g) structure of (d) ($\alpha = 1.2, \quad \lambda = 3.2$) (e) extracted texture layer from (d)-(g) (f) extracted noise

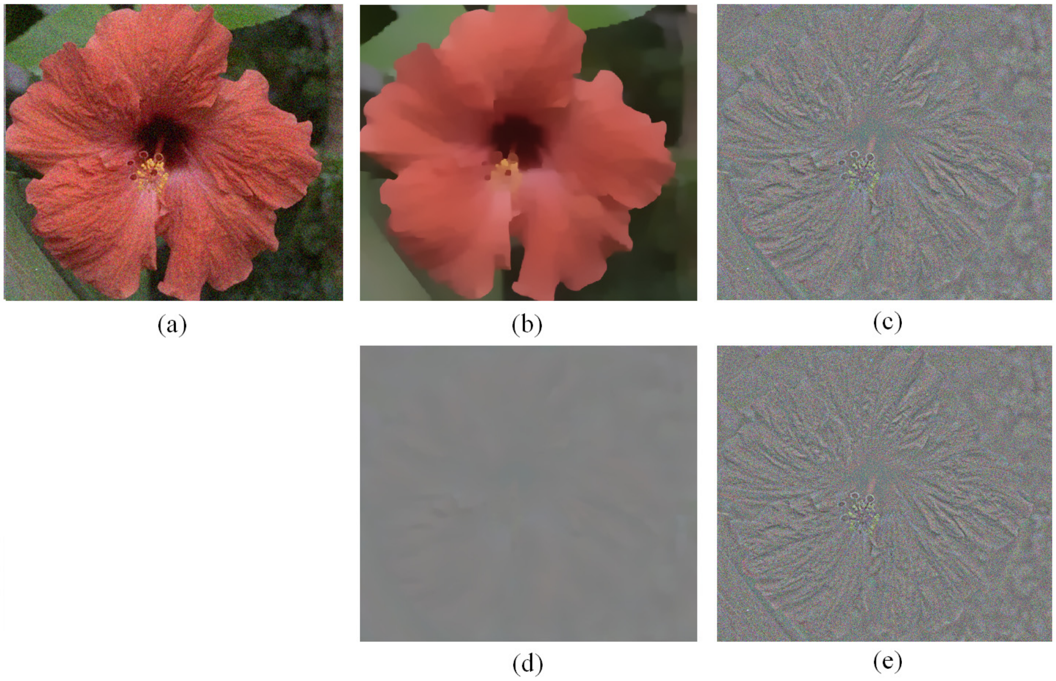


Figure 1.7: Structure/texture/noise decomposition via RTV operator [1] (a) noisy image degraded by additive white Gaussian ($\sigma = 20$) (b) structure of (a) (c) texture of (a), texture layer (c) is decomposed into (d) and (e) by one more RTV method

1.7 is the results of RTV-based decomposition. In these experiments, it is noted that the decomposition algorithm which is more robust to noise is required. In addition, many decomposition algorithms adopt optimization framework which increases computational complexities and causes convergence problems. Hence, straight-forward and patch-based framework are considered for robust decomposition algorithm to noise since patch-based framework is know for its excellence in image denoising.

Conventional patch-based framework used in image denoising can be represented by NLM [28] and BM3D [5]. They are designed to remove noise not texture and it is limited in applying them to image decomposition. For example, if NLM is used to

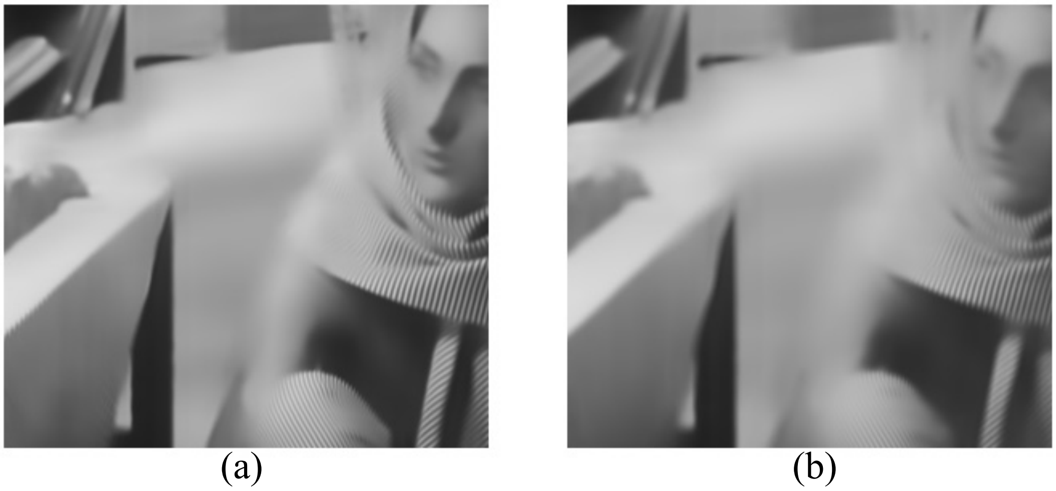


Figure 1.8: Nonlocal means filtering output (a) $\sigma = 30$, (b) $\sigma = 40$

decompose an image, the kernel bandwidth in eq. (1.5) should be widen. However, texture cannot be preserved well as increasing the bandwidth as shown in Fig. 1.8. Consequently, it is required to find ore edge-preserving technique which is robust to noise at the same time. In the proposed method, this problem is resolved by adjusting the number of patches as well as kernel bandwidth.

In this dissertation, a patch-based image decomposition framework is proposed and it is extended to a noisy image decomposition. Specifically, an image is decomposed by edge-preserving smoothing where its smoothed output is the weighted average of neighboring pixels. Given a pixel, a set of similar patches is found by measuring patch similarities based on mean squared error (MSE). Then, weights between each patch and its similar patches are computed respectively in the similar way of NLM [28]. With the patch weights, all the pixels in a patch are updated at the same time while considering local pixel weighs. For better edge-preserving smoothing, kernel bandwidth and number of similar patches are tuned. Experimental results show that the proposed framework work well for image decomposition and it is robust to the presence of noise. Also it is verified that the proposed method can be utilized in many applications such as HDR tone mapping and detail enhancement. In addition, by adopting the proposed method in decomposition of a noisy image, both image denoising and image enhancement can be achieved in the proposed framework. Furthermore, structure scalable image coding scheme is proposed based on the proposed decomposition where structure and only salient texture are encoded for efficient image coding.

Chapter 2

Related work

2.1 Image decomposition

2.1.1 Laplacian subbands

Constructing Laplaican subbands is one of the simplest image decomposition methods. When a signal is low pass filtered and the filtered output is subtracted from the original, then a high band signal is obtained. If this process is repeated for the low pass filtered signal, then a set of subband signals is obtained as in Fig. 2.1. The Laplacian pyramid for an image signal is constructed in this manner, where the low pass filter is a Gaussian filter with appropriate kernel bandwidth. More specifically, for a given image I, the Gaussian filter is applied iteratively with downsampling at every step. This process can be described as

$$\begin{aligned} G_0 &= I, \\ G_{k+1} &= \downarrow_2 (Gaussian(G_k)) \qquad for \quad k = 0, ..., n-1 \end{aligned} \tag{2.1}$$

where $\downarrow_2(\cdot)$ denotes the downsampling by 2 and $Gaussian(\cdot)$ is the Gaussian filtering. Then, the Laplacian subbands are defined as

$$\begin{aligned} L_{k+1} &= G_k - \uparrow_2 (G_{k+1}) \qquad for \quad k = 0, ..., n-2, \\ L_n &= G_n \end{aligned} \tag{2.2}$$

where $\uparrow_2(\cdot)$ denotes the upsampling by 2 and n is the level of the pyramid. In this paper, just two levels of Laplacian subband ($n = 2$) are used, where L_1 denotes the high-frequency (detail) subband and L_2 represents the low-frequency subband.

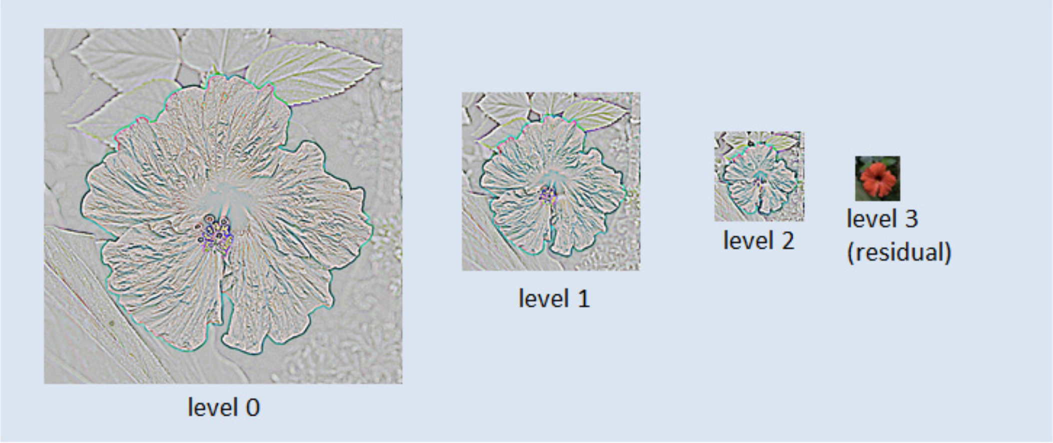


Figure 2.1: Laplacian subbands

2.2 Edge-preserving smoothing

2.2.1 Bilateral filtering

The bilateral filter (BLF) proposed in [11] is a non-linear filter that considers both of spatial and photometric distances of the neighboring pixels and its output is represented by weighted summation of them. So as shown in Fig. 2.2, its kernel is well adapted in image contents. It is well-known that the bilateral filter based on local self-similarity can remove noise or a fine detail in images with edge-preserving manner. In addition, it is simple and non-iterative. Hence, it has been shown to be effective in many applications such as image denoising, detail enhancement, tone mapping for high dynamic range (HDR) images and so on [31, 8, 32–34, 18]. Besides, since the state-of-the-art image denoising methods such as BM3D [5] and non-local means filter [28] require too much computation and high complexity, the bilateral filtering scheme can be a practical solution for mobile device and real-time implementation.

The BLF output is defined as

$$J(\mathbf{p}) = \frac{1}{W} \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q}) I(\mathbf{q}) \tag{2.3}$$

where $\mathbf{p}$ and $\mathbf{q}$ denote pixel positions, $N_{\mathbf{p}}$ is the neighbor of $\mathbf{p}$, $I(q)$ is the intensity of the image at a pixel $\mathbf{q}$, W is the normalizing factor as $W = \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q})$ and $w(\mathbf{p}, \mathbf{q})$ is the kernel of the BLF defined as [11]

$$w(\mathbf{p}, \mathbf{q}) = \exp\left(-\frac{\|\mathbf{p} - \mathbf{q}\|^2}{2\sigma_d^2}\right) \exp\left(-\frac{\|I(\mathbf{p}) - I(\mathbf{q})\|^2}{2\sigma_r^2}\right) \tag{2.4}$$

where σ_d is the bandwidth for the spatial distance and σ_r for the photometric distance. For successfully reducing the noise while keeping the edges, it is important to find the balance between the σ_d and σ_r, and also to find an appropriate size of the neighbor.

Recently, there have been several attempts to apply bilateral filter in some different domains [32,34,33]. In [32], an image is decomposed into the low-frequency and high-frequency subbands through wavelet decomposition and the noise is reduced by the combination of bilateral filtering and wavelet thresholding. The bilateral

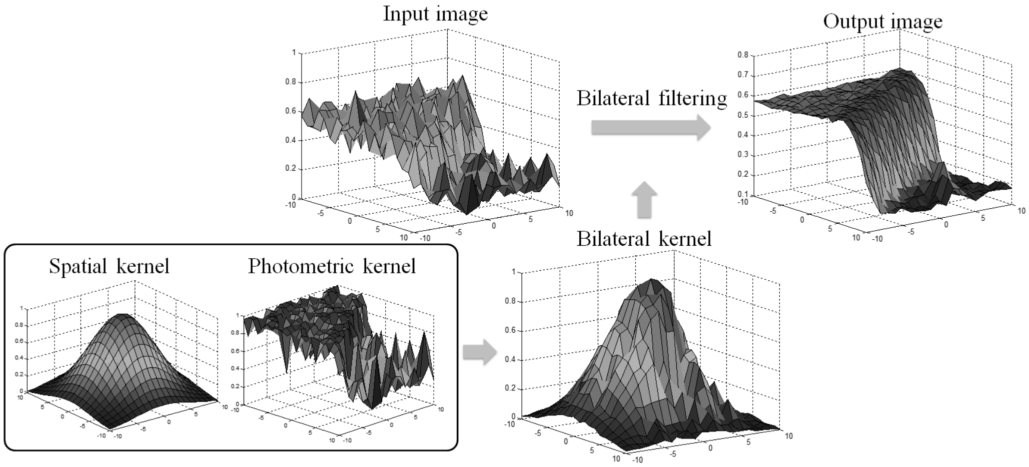
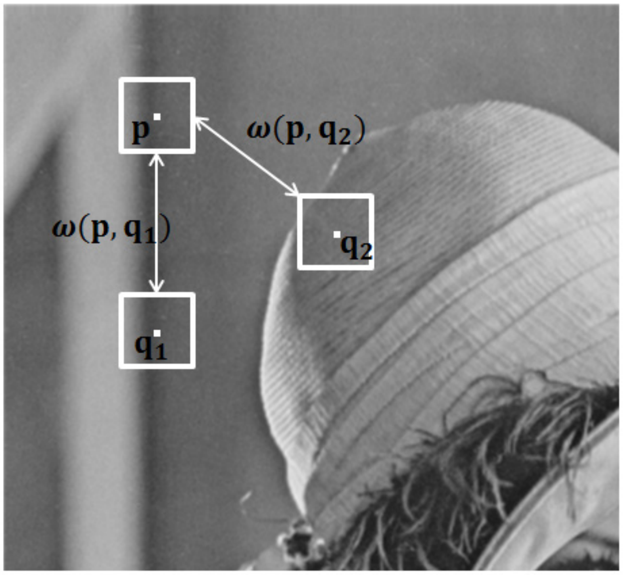


Figure 2.2: Bilateral filtering

filtering in the gradient domain proposed in [33] also shows good results. In [34], a recursive bilateral filter is proposed and extended to the gradient domain filtering. In these schemes, the properties of different domains can be exploited to enhance the denoising performance. In addition, the detail enhancement and image denoising can be achieved simultaneously.

Bandwidth selection

Three are two parameters in BLF. One is the spatial bandwidth σ_d in (2.4) and the other is the photometric (range) bandwidth σ_r. Since the performance of bilateral filter relies on selection of them, it is important to choose proper ones. In [32], the research for analyzing filtering performance according to parameters is tried. The experiments are done for understanding the relations of bandwidths and noise standard deviation σ_n. From the experiments, it is verified that that σ_d is insensitive to noise standard variation. Instead, proper range for $sigma_d$ is appealed as $[1.5, 2.1]$.



$\omega(\mathbf{p},\mathbf{q}_2)$
$\omega(\mathbf{p},\mathbf{q}_1)$
$\mathbf{p}$
$\mathbf{q}_1$
$\mathbf{q}_2$

Figure 2.3: Illustration of nonlocal means filtering scheme

On the other hand, it is noted that changing the noise standard deviation σ_n causes the change of the photometric bandwidth σ_r. That is, the optimal σ_r is varied according to σ_n and from the experiments, suggested value of σ_r is determined as

$$\sigma_r = 2\sigma_n. \tag{2.5}$$

2.2.2 Nonlocal means filtering

Nonlocal means (NLM) filter is proposed for image denoising [28]. Due to its effectiveness, it is adopted in many applications such as image inpainting [35] and super-resolution [36]. It is a patch-based technique and it assumes nonlocal self-similarity which means similar patches are not only in neighboring region, but also the entire region. So its output is the weighted average of all pixels in image. Specifically, Let us denote the square patch centered at pixel $\mathbf{p}$ as $\Phi_\mathbf{p}$ and the intensity value at $\mathbf{p}$ as $I(\mathbf{p})$. The output value at $\mathbf{p}$, $J(\mathbf{p})$ is determined by a weighted average as

$$J(\mathbf{p}) = \frac{1}{W} \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q}) I(\mathbf{q}) \tag{2.6}$$

where W is the normalizing factor as $W = \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q})$ and $w(\mathbf{p}, \mathbf{q})$ is the weight between $\mathbf{p}$ and $\mathbf{q}$. Here, $N(\mathbf{p})$ is supposed to be the entire image, but practically for better computing, it is often limited within a search window around $\mathbf{p}$. The strategy of weight is to give high value to similar pixels and low value to non-similar pixels. For example, in Fig. 2.3, $w(\mathbf{p}, \mathbf{q}_1)$ is larger than $w(\mathbf{p}, \mathbf{q}_2)$ as $\mathbf{q}_1$ is more similar patch than $\mathbf{q}_2$. So the weight is measured by as

$$w(\mathbf{p}, \mathbf{q}) = \exp\left(-\frac{\|\Phi_{\mathbf{p}} - \Phi_{\mathbf{q}}\|}{h^2} \right) \tag{2.7}$$

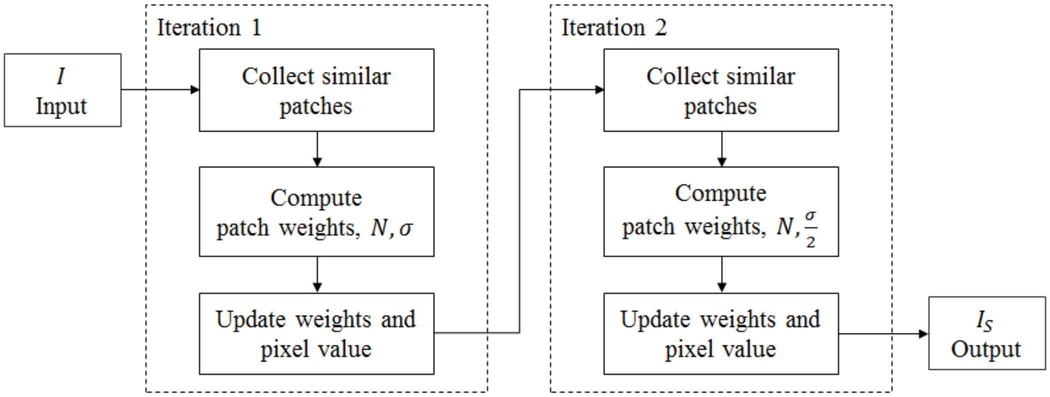
where $\Phi_{\mathbf{p}}$ is the squared patch centered at $\mathbf{p}$. Also h is the filtering bandwidth which controls permissable color range. In other words, Larger h gives more smoothing results.

Chapter 3

Scale-aware decomposition of images based on patch-based filtering

3.1 Edge-preserving smoothing via patch-based framework

The proposed algorithm consists of three main steps: collecting similar patches at each pixel, computing patch weights and updating patch and weights. In the step of collecting similar patches, a set of patches are selected for each pixel by measuring patch similarities and additional constraints. In the second step, the weights between each pixel and its similar patches are calculated. In the third step, the values of all the pixels in a patch are updated by considering both patch weights and local pixel weights. Hence, the filtering output can be computed by the weighted summation



Iteration 1
Iteration 2
I Input
Collect similar patches
Compute patch weights, N, σ
Update weights and pixel value
Collect similar patches
Compute patch weights, $N, \frac{\sigma}{2}$
Update weights and pixel value
I_S Output

Figure 3.1: Block diagram of proposed image decomposition framework

of pixels in similar patches. Fig. 3.1 shows the block diagram of the proposed image decomposition framework. As seen in the block diagram, the proposed algorithm utilizes two iterations which are performed through the same smoothing filter with different parameters. In the proposed method, two types of parameters are used to control the degree of smoothing. In the first step, kernel bandwidth σ and number of patches N are utilized while $\frac{\sigma}{2}$ and N are used in the second step. That is, in the second iteration, the proposed method performs smoothing an image less than the first iteration by adjusting kernel bandwidth as $\frac{\sigma}{2}$.

More specifically, let us denote an input image as I and given a pixel $\mathbf{p} = (x, y)$, a pixel value at $\mathbf{p}$ as $I(\mathbf{p})$. A squared patch centered at $\mathbf{p}$ is represented as $\Phi_{\mathbf{p}}$, i.e.,

$$\Phi_{\mathbf{p}} = [I(\mathbf{p}_1), I(\mathbf{p}_2), ..., I(\mathbf{p}_{N_{\mathbf{p}}})], \tag{3.1}$$

$$\begin{bmatrix} I(\mathbf{p}_1) & \cdots & I(\mathbf{p}_1) & \cdots & I(\mathbf{p}_1) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ I(\mathbf{p}_{2r^2+r+1}) & \cdots & I(\mathbf{p}_{2r^2+2r+1}) & \cdots & I(\mathbf{p}_{2r^2+3r+1}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ I(\mathbf{p}_{N_\mathbf{p}-2r}) & \cdots & I(\mathbf{p}_{N_\mathbf{p}-r}) & \cdots & I(\mathbf{p}_{N_\mathbf{p}}) \end{bmatrix} \tag{3.2}$$

$$= \begin{bmatrix} I(x-r,y-r) & \cdots & I(x,y-r) & \cdots & I(x+r,y-r) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ I(x-r,y) & \cdots & I(\mathbf{p}) = I(x,y) & \cdots & I(x+r,y) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ I(x-r,y+r) & \cdots & I(x,y+r) & \cdots & I(x+r,y+r) \end{bmatrix}$$

where the patch window size is $(2r+1) \times (2r+1)$. Also $\mathcal{C}_\mathbf{p}$ is the set of pixels in search window where patches similar to $\Phi_\mathbf{p}$ would be found. First of all, similar patches are collected based on the mean squared error (MSE) as

$$d(\mathbf{p},\mathbf{q}) = \|\Phi_\mathbf{p} - \Phi_\mathbf{q}\|^2 = \frac{1}{N_\mathbf{p}} \sum_{k=1}^{N_\mathbf{p}} (I(\mathbf{p}_k) - I(\mathbf{q}_k))^2, \qquad I(\mathbf{p_k}) \in \Phi_\mathbf{p}. \tag{3.3}$$

where $N_\mathbf{p}$ is the number of patches in $\Phi_\mathbf{p}$. For better applications, the guidance image G can be used instead of I and then $d(\mathbf{p},\mathbf{q})$ can be approximated as

$$d(\mathbf{p},\mathbf{q}) = \|\Phi_\mathbf{p} - \Phi_\mathbf{q}\|^2 \approx \frac{1}{N_\mathbf{p}} \sum_{k=1}^{N_\mathbf{p}} (G(\mathbf{p}_k) - G(\mathbf{q}_k))^2, \qquad I(\mathbf{p_k}) \in \Phi_\mathbf{p}. \tag{3.4}$$

Guidance image G can be input image and other on purpose. For example, to remove specific scale of textures, it can be the Gaussian filtered image [4]. Then N patches constitute the set of similar patches, $\mathcal{C}_\mathbf{p}$ ascending order of MSE value, i.e.,

$$\mathcal{C}_\mathbf{p} = \{\mathbf{q}| \quad |\mathbf{p}-\mathbf{q}| > \epsilon, d(\mathbf{p},\mathbf{q}) < d(\mathbf{p},\mathbf{r}), r \notin \mathcal{C}_\mathbf{p}\}, \qquad n(\mathcal{C}_\mathbf{p}) \leq N. \tag{3.5}$$

Note that ϵ is used to avoid selecting $\mathbf{p}$. Now patch weights are computed between each patch and its similar patches as

$$w(\mathbf{p},\mathbf{q}) = \exp -\frac{\|\Phi_{\mathbf{p}} - \Phi_{\mathbf{q}}\|^2}{2\sigma^2}, \qquad \mathbf{q} \in \mathcal{C}_{\mathbf{p}}. \tag{3.6}$$

After computing patch weights, pixel values are calculated by the weighted summation of $\mathcal{C}_{\mathbf{p}}$ in this pixel update step and pixel values and weights are accumulated during filtering as in Fig. 3.2. The output pixel $I_S(\mathbf{p})$ is determined by

$$\begin{aligned} I_S(\mathbf{p}) = & \sum_{\mathbf{q}\in\mathcal{C}_{\mathbf{p}}} w(\mathbf{p},\mathbf{q}) \exp\left(-\frac{(I(\mathbf{p}) - I(\mathbf{q}))^2}{2\sigma^2}\right) \\ & + \sum_{\mathbf{r}\in\Psi_{\mathbf{p}}} \sum_{\mathbf{s}\in\mathcal{C}_{\mathbf{r}}} w(\mathbf{r},\mathbf{s}) \exp -\frac{(I(\mathbf{p}) - I(\mathbf{s}+\mathbf{p}-\mathbf{r}))^2}{2\sigma^2} \end{aligned} \tag{3.7}$$

where $\Psi_{\mathbf{p}} = \{\mathbf{r} | I(\mathbf{p}) \in \Phi_{\mathbf{r}} \quad \mathbf{r} \neq \mathbf{p}\}$. Also $w(\mathbf{p},\mathbf{q})$ is the patch weight and $\exp\left(-\frac{(I(\mathbf{p})-I(\mathbf{q}))^2}{2\sigma^2}\right)$ is the local weight as described in fig. 3.3. Fig. 3.4 shows image decomposition result varying σ and N.

3.2 Multi-scale image decomposition

The proposed image decomposition method can be adapted for multi-scale decomposition. It is done by sequential method and parallel method. These are chosen depending on purpose.

In the sequential method, the proposed smoothing operator is performed recursively. That is, after an image $I = I_S^0$ is once decomposed into structure I_S^1 and texture layers I_T^1, structure I_S^1 as an input is decomposed into next level structure I_S^2 and texture I_T^2. So a series of smoothing is processed as in 3.5 and n-level decomposition is formulated by

$$\begin{aligned} I_S^0(\mathbf{p}) = I(\mathbf{p}), \qquad & I_S^{k-1}(\mathbf{p}) = I_S^k(\mathbf{p}) + I_T^k(\mathbf{p}) \\ & I(\mathbf{p}) = \sum_k^n I_T^k(\mathbf{p}) + I_S^n(\mathbf{p}) \end{aligned} \tag{3.8}$$

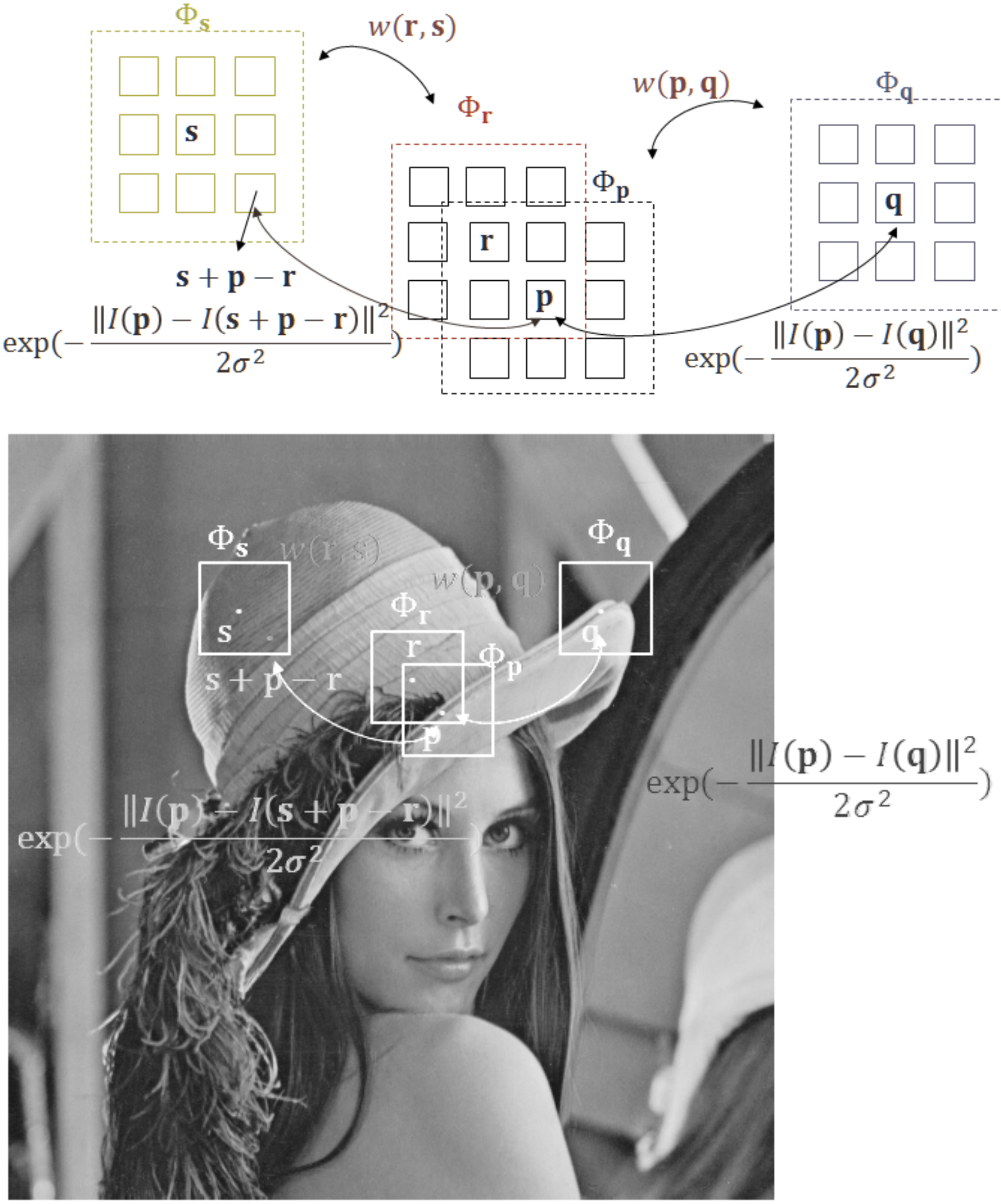

$\Phi_{\mathbf{s}}$
$w(\mathbf{r},\mathbf{s})$
$\Phi_{\mathbf{r}}$
$w(\mathbf{p},\mathbf{q})$
$\Phi_{\mathbf{q}}$
$\Phi_{\mathbf{p}}$
$\mathbf{s}$
$\mathbf{r}$
$\mathbf{p}$
$\mathbf{q}$
$\mathbf{s}+\mathbf{p}-\mathbf{r}$
$\exp(-\frac{\|I(\mathbf{p})-I(\mathbf{s}+\mathbf{p}-\mathbf{r})\|^2}{2\sigma^2})$
$\exp(-\frac{\|I(\mathbf{p})-I(\mathbf{q})\|^2}{2\sigma^2})$
$\Phi_{\mathbf{s}}$
$w(\mathbf{r},\mathbf{s})$
$w(\mathbf{p},\mathbf{q})$
$\Phi_{\mathbf{q}}$
$\Phi_{\mathbf{r}}$
$\mathbf{s}$
$\mathbf{r}$
$\Phi_{\mathbf{p}}$
$\mathbf{q}$
$\mathbf{s}+\mathbf{p}-\mathbf{r}$
$\mathbf{p}$
$\exp(-\frac{\|I(\mathbf{p})-I(\mathbf{q})\|^2}{2\sigma^2})$
$\exp(-\frac{\|I(\mathbf{p})-I(\mathbf{s}+\mathbf{p}-\mathbf{r})\|^2}{2\sigma^2})$

Figure 3.2: Illustration of updating patch

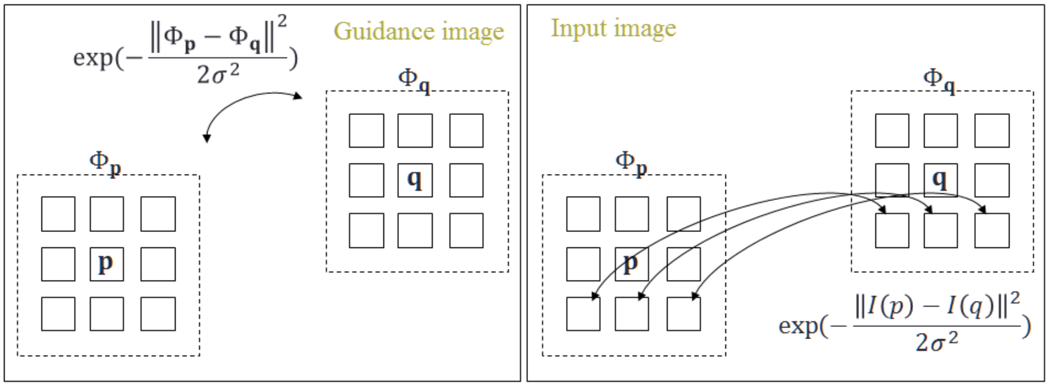

$\exp(-\frac{\|\Phi_p - \Phi_q\|^2}{2\sigma^2})$
Guidance image
Φ_q
q
Φ_p
p
Input image
Φ_q
q
Φ_p
p
$\exp(-\frac{\|I(p) - I(q)\|^2}{2\sigma^2})$

Figure 3.3: Comparison of patch weights and local weights

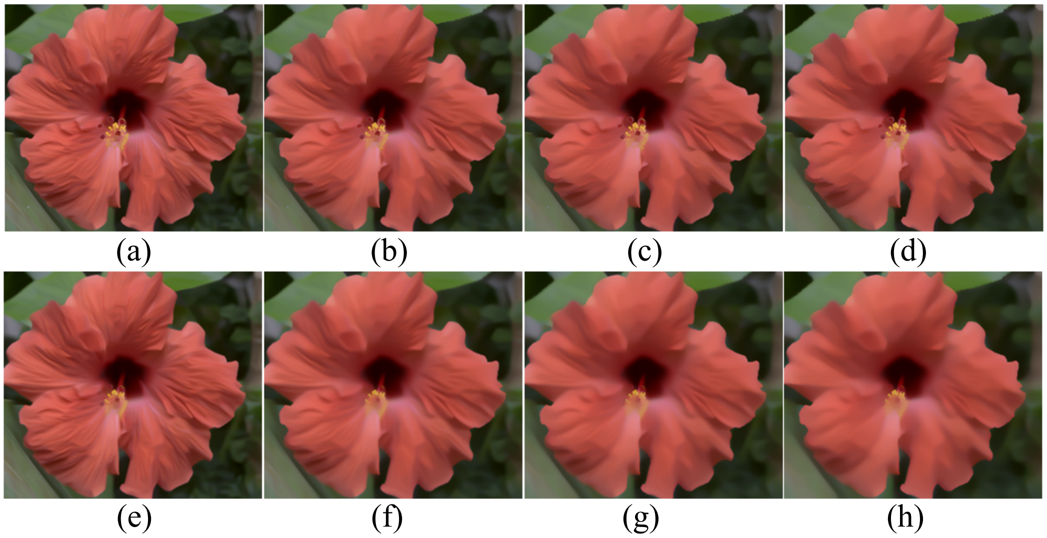


Figure 3.4: Image decomposition comparison (a) $simga = 10$, $N = 10$ (b) $simga = 10$, $N = 40$ (c) $simga = 10$, $N = 70$ (d) $simga = 10$, $N = 100$ (e) $simga = 20$, $N = 10$ (f) $simga = 20$, $N = 40$ (g) $simga = 20$, $N = 70$ (h) $simga = 20$, $N = 100$

where I_S^k is the structure layer obtained through k-th smoothing process. Hence, I_S^n becomes the coarsest structure layer and I_T^1 becomes the finest texture layer.

On the other hand, in the parallel method the multi-scale decomposition is carried by varying smoothing parameters such as the number of similar patches and kernel bandwidth. Fig. 3.6 illustrates the process of parallel method. As shown, by increasing parameters progressively, gradually coarser structure is obtained.

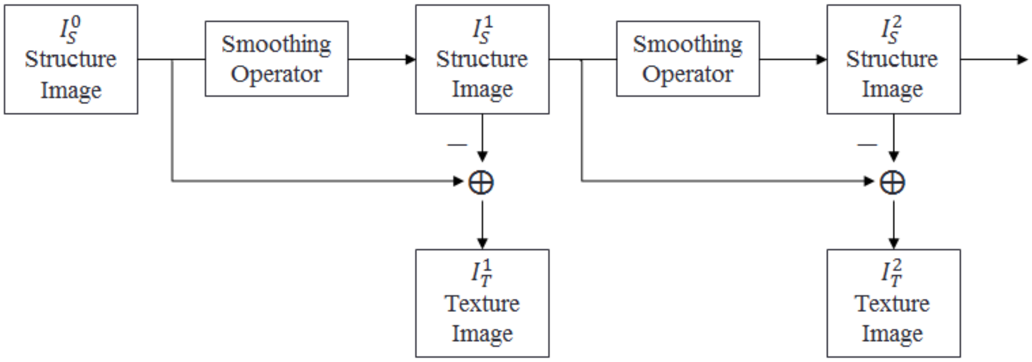


Figure 3.5: Sequential method

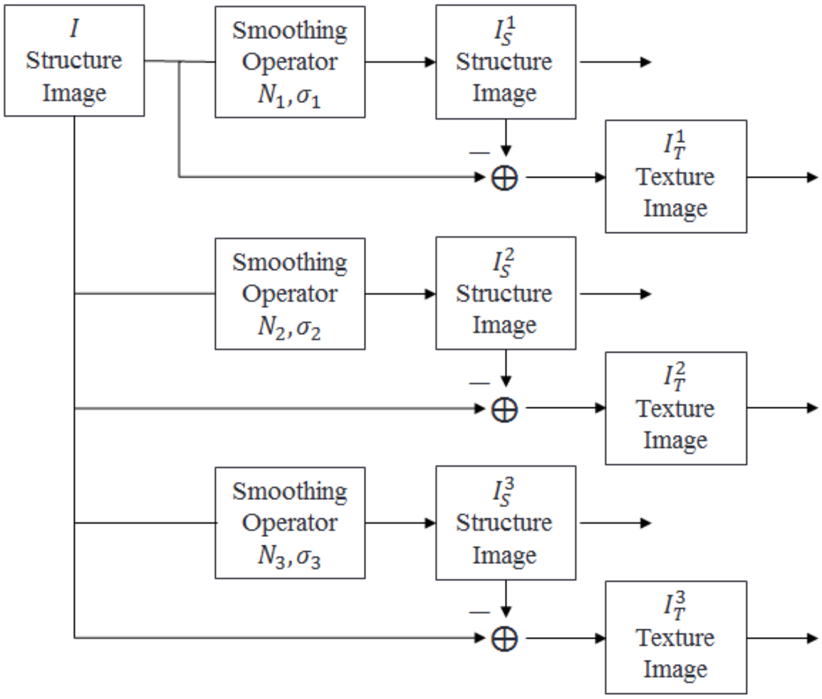


Figure 3.6: Parallel method

Chapter 4

Applications

4.1 Image enhancement

4.1.1 Detail enhancement

To enhance detail of an image, we extract a texture layer I_T from the image via image decomposition and boost it by multiplying constant. The result image with enhanced detail is represented as

$$I_{enhanced} = I_S + \alpha \times I_T, \quad \alpha > 1 \tag{4.1}$$

where I_S is a structure layer obtained by the proposed smoothing algorithm.

We tested image enhancement and compared enhancement results with WLS operator [1], relative total variation model [2], region covariance based method [6] and rolling guidance filter [4]. As depicted in Fig. 4.6, proposed method shows comparable result with state-of- the-art methods. It enhances detail of given image non-artificially and preserves proper color as well without reversing edge. Fig. 4.2 also provides enhanced result which shows the proposed method can be adopted in detail enhancement.

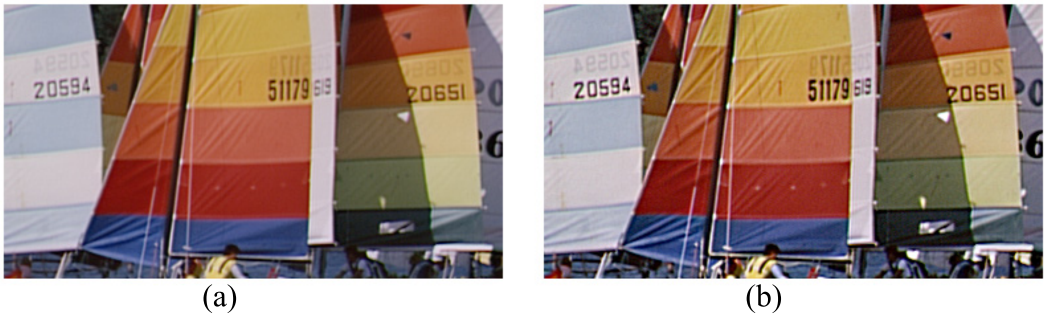


Figure 4.1: Detail enhancement result (a) original image (b) detail-enhanced image

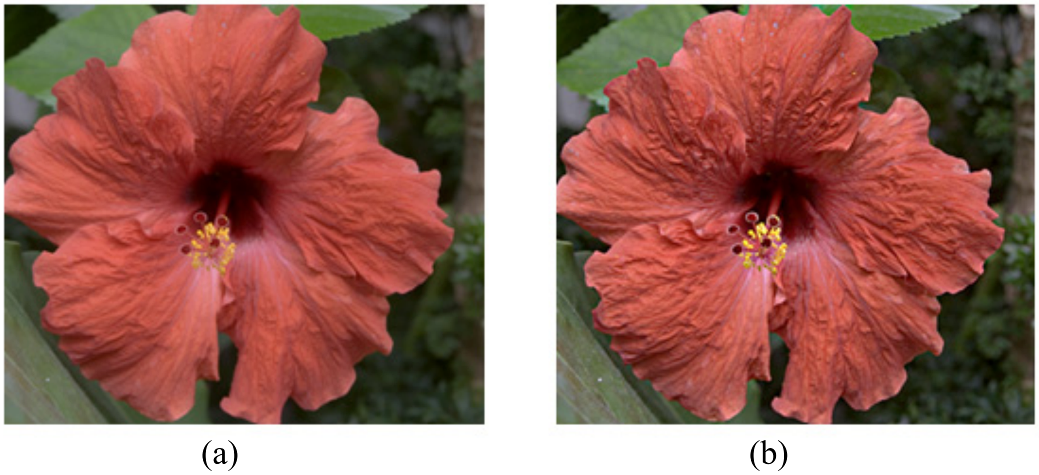


Figure 4.2: Detail enhancement result (a) original image (b) detail-enhanced image

Fig. 4.3 and Fig. 4.4 show comparison results with other methods such as WLS [1], RTV [2], local extrema [3], rolling guidance filter (RG) [4], BM3D [5] and region covariance [6]. Here, the results in Fig. 4.3 are produced by less smoothing than those in Fig. 4.4. In Fig. 4.3, image details are enhanced well, but some artifacts in the leaf are shown in (c), (d), (e) and (g). Also in Fig. 4.3 (f), since

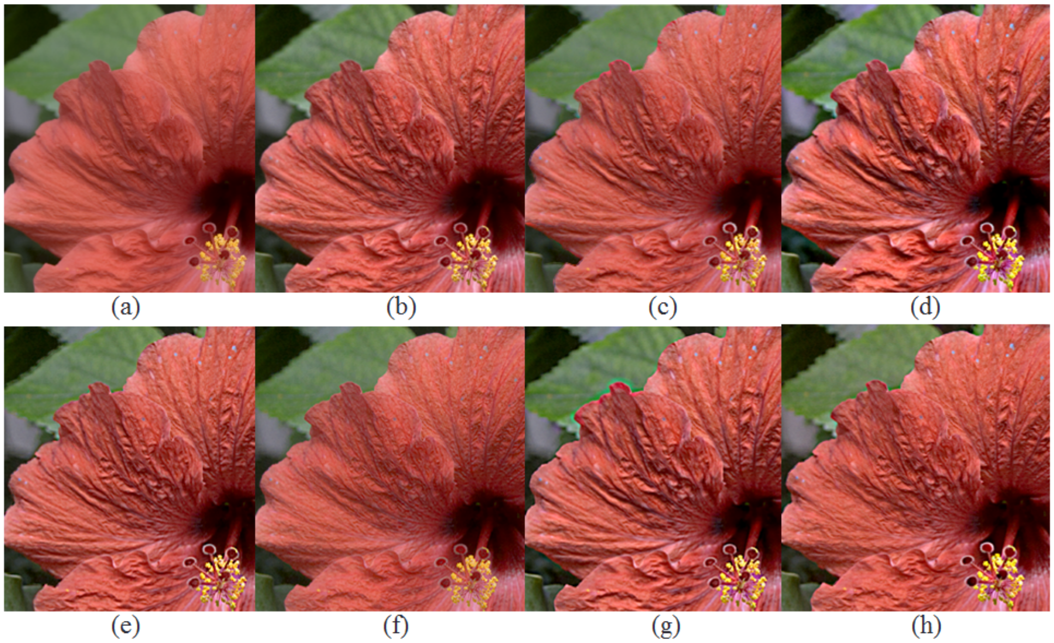


Figure 4.3: A comparison of Detail enhancement results (a) original image (b) detail-enhanced via WLS operator [1] (c) RTV [2] (d) local extrema [3](e) rolling guidance filter (RG) [4](f) BM3D [5] (g) region covariance [6] (h) proposed method

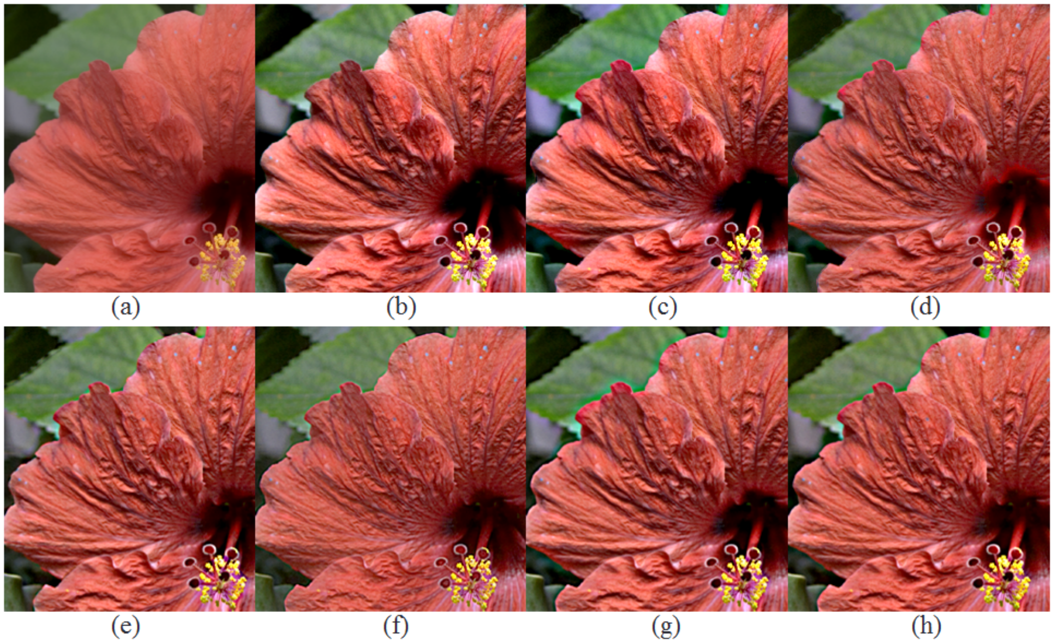


Figure 4.4: A comparison of Detail enhancement results (strong enhancement) (a) original image (b) detail-enhanced via WLS operator [1] (c) RTV [2] (d) local extrema [3](e) rolling guidance filter (RG) [4](f) BM3D [5] (g) region covariance [6] (h) proposed method

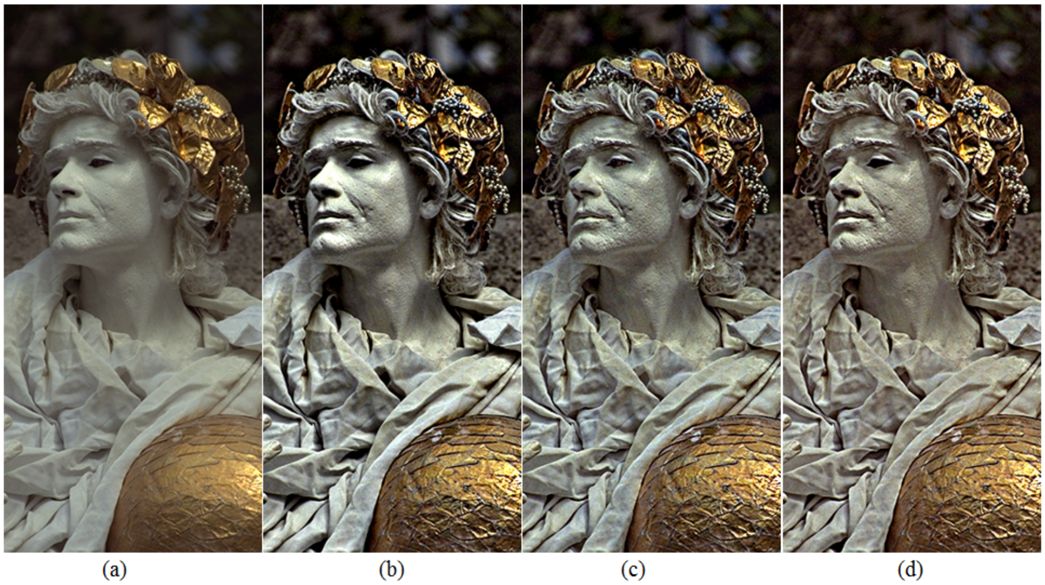


Figure 4.5: A comparison of Detail enhancement results (strong enhancement) (a) original image (b) detail-enhanced via WLS operator [1] (c) RTV [2] (d)proposed method

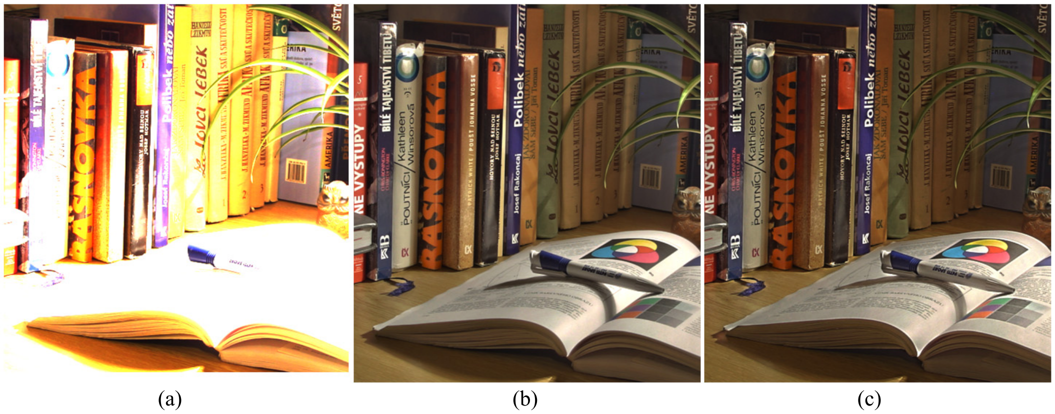

(a) (b) (c)

Figure 4.6: HDR tone mapping result 1 (a) original image (b) WLS (c) proposed method

BM3D is not for image decomposition, less details are extracted and enhanced. Fig 4.4 and Fig. 4.5 verify that WLS and RTV cause a change in colors and tones while the proposed method preserves the entire tones during enhancing image details.

4.1.2 HDR tone mapping

HDR tone mapping reduces contrast to display an high-dynamic-range image in low-dynamic range devices and the proposed decomposition is applied to it. Like the method in [1], the BLF in [8] is replaced with the proposed smoothing. Concretely, a log-luminance channel is extracted from a given HDR input by

$$L = 0.299 \times R + 0.587 \times G + 0.114 \times B, LL = \log(L). \tag{4.2}$$

where L is a luminance channel and LL, a log-luminance channel. Then the proposed edge-preserving smoothing is performed on LL repeatedly to smooth it more and more by incrementing the bandwidth and the number of patches. This series of processing constructs multiple structure layer u_0, u_1 and u_2 in order. Detail layers,

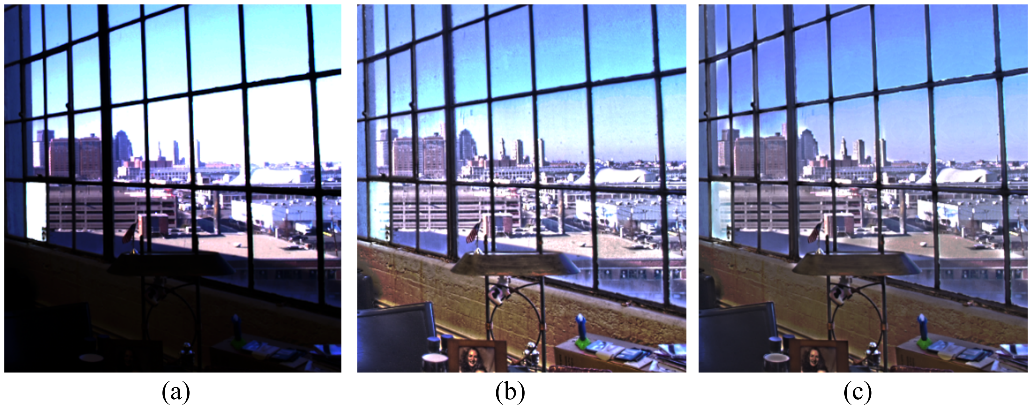


Figure 4.7: HDR tone mapping result 2 (a) original image (b) WLS (c) proposed method

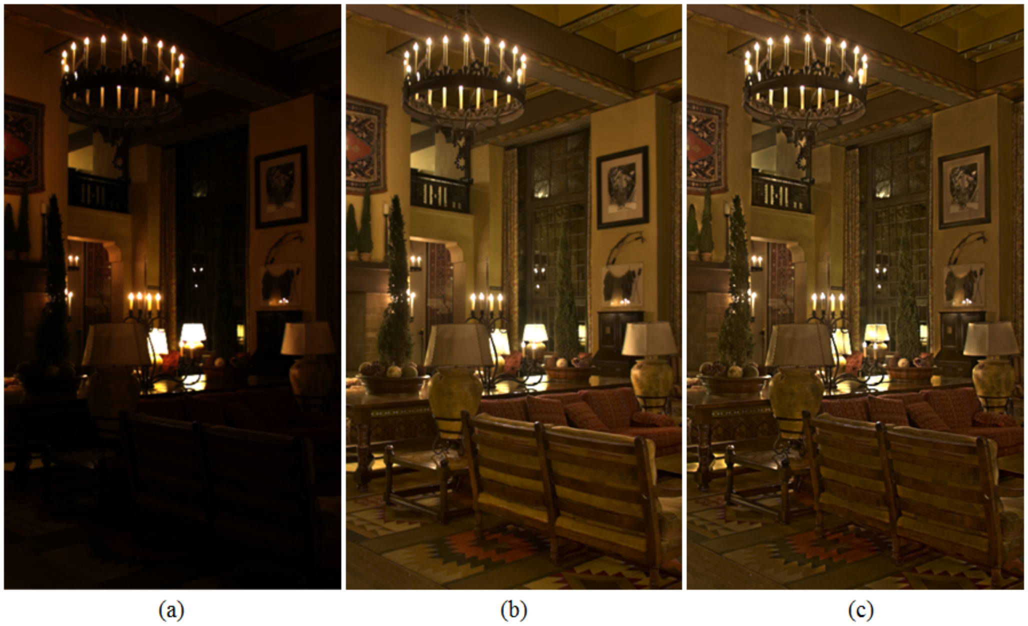


Figure 4.8: HDR tone mapping result 3 (a) original image (b) WLS (c) proposed method

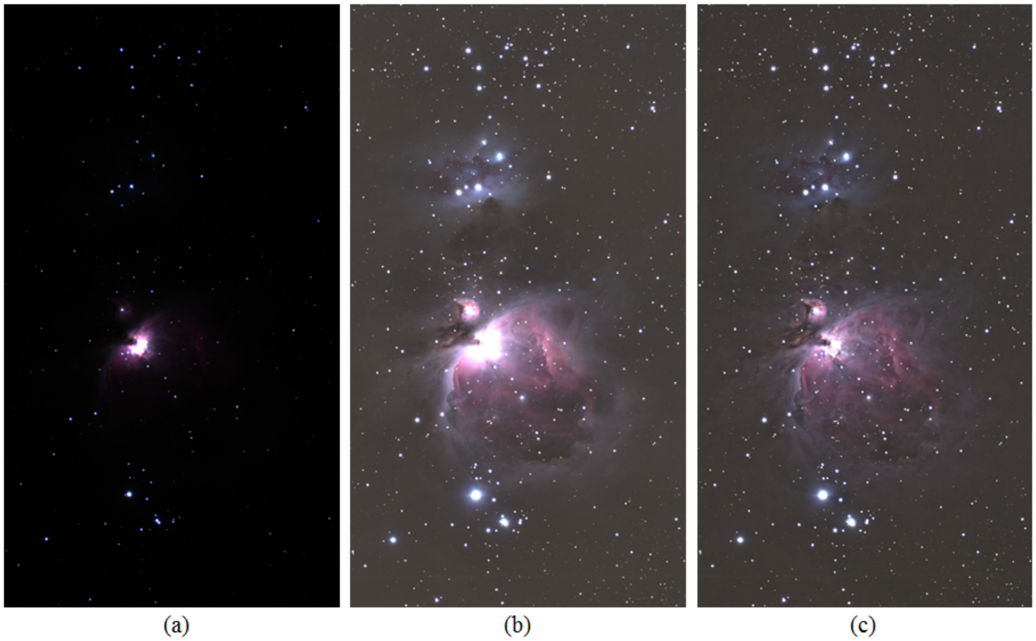


Figure 4.9: HDR tone mapping result 4 (a) original image (b) WLS (c) proposed method

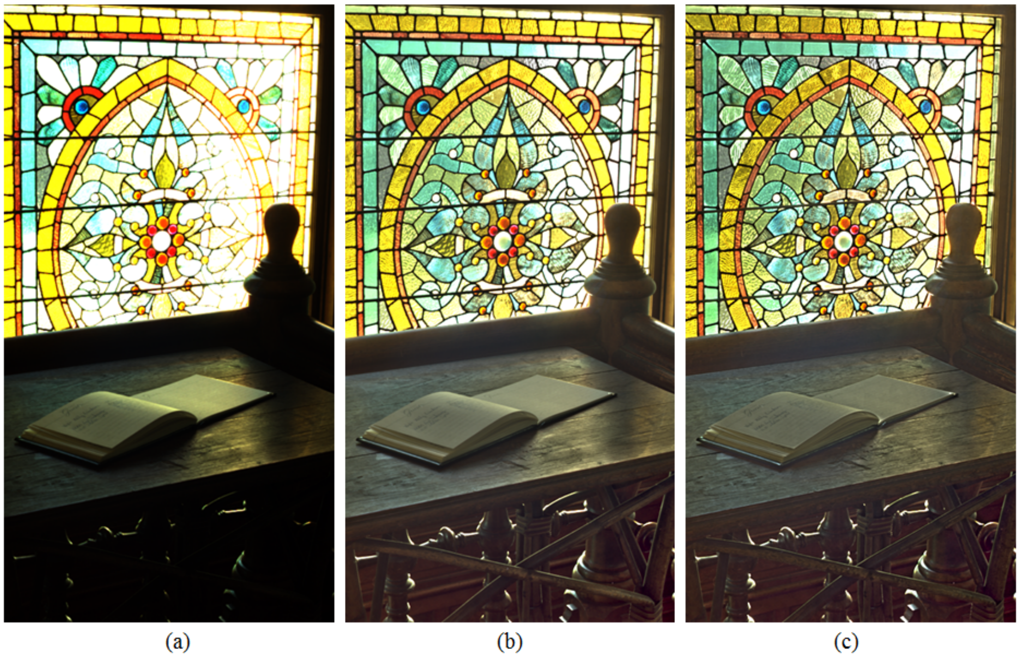


Figure 4.10: HDR tone mapping result 5 (a) original image (b) WLS (c) proposed method

d_0, d_1 and d_2 are computed as

$$u_0 = LL, \qquad d_n = u_n - u_{n+1}. \tag{4.3}$$

Then the output with compressed range is obtained by recombining the layers while boosting details and this means

$$\hat{L} = \exp(w_0 d_0 + w_1 d_1 + w_2 d_2 + w_3 d_3) \tag{4.4}$$

where weights w_0, w_3, w_3, w_3 are adjusted to balance multi-scale details.

The proposed HDR tone mapping results are compared with the method based on WLS operation [1] as depicted in Fig. 4.7 and Fig. 4.6. Original HDR range images are shown in Fig. 4.7 (a) and Fig. 4.6 (a). Through tone mapping process, their dynamic ranges are reduced. Fig. 4.7 (b) and Fig. 4.6 (b) are from the result of WLS operator and (c) from the proposed image decomposition. As shown, it turns out that the proposed algorithm gives pleasing visual quality without saturation and noticeable artifacts. Similar tendencies are observed in Fig. 4.8, 4.9 and 4.10.

4.2 Image denoising

4.2.1 A noisy image decomposition

Image denoising is regarded as decomposition problem of a noisy image which result in a structure layer, a texture layer and a noise layer. Since noise is generally a kind of small textures, the extracted texture layer after one step of decomposition contains both texture and noise. Consequently, one more decomposition process is required but it is performed on the texture layer unlike conventional image decomposition methods. This is for decomposition of noise and texture and image denoising is

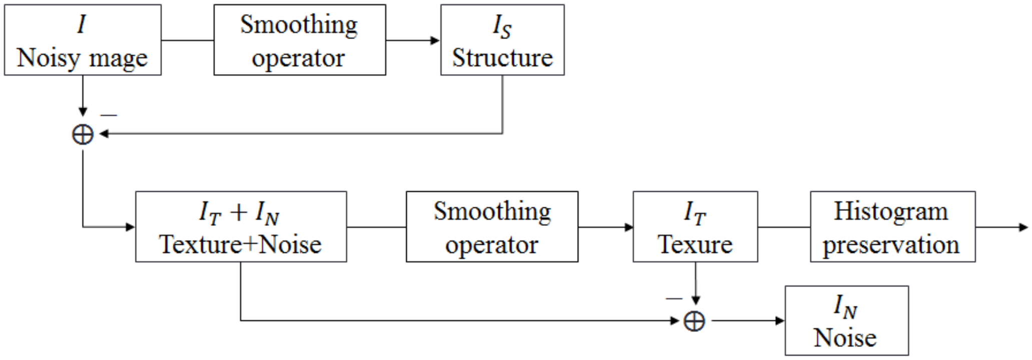


Figure 4.11: Block diagram of decomposing a noisy image

achieved by eliminating the noise layer. The entire decomposition of a noisy image is described in 4.11 and result image of each process is provided in fig. 4.12. Note that the final texture layer goes through histogram preservation process. It is because smoothing step no matter how good causes loss of high frequency details. Hence, the histogram adjustment step is proposed to preserve the properties of the original histogram, which is demonstrated in the next section. Further, in fig. 4.13, it is verified that the proposed method are more robust to noise than other methods.

4.2.2 texture enhancement via histogram preservation

As stated in the introduction, one of the reasons for choosing the decomposition-based filtering scheme is to enhance the edges while reducing the noise. A straightforward method would be to amplify the denoised high band and then add this signal to the denoised low band image, like the original unsharp mask method. However, since the edge components in the high band have been smoothed in the filtering process, the straightforward method might add the smeared edges. Hence, the process is performed to restore the edge strength of the high band image as strong as the

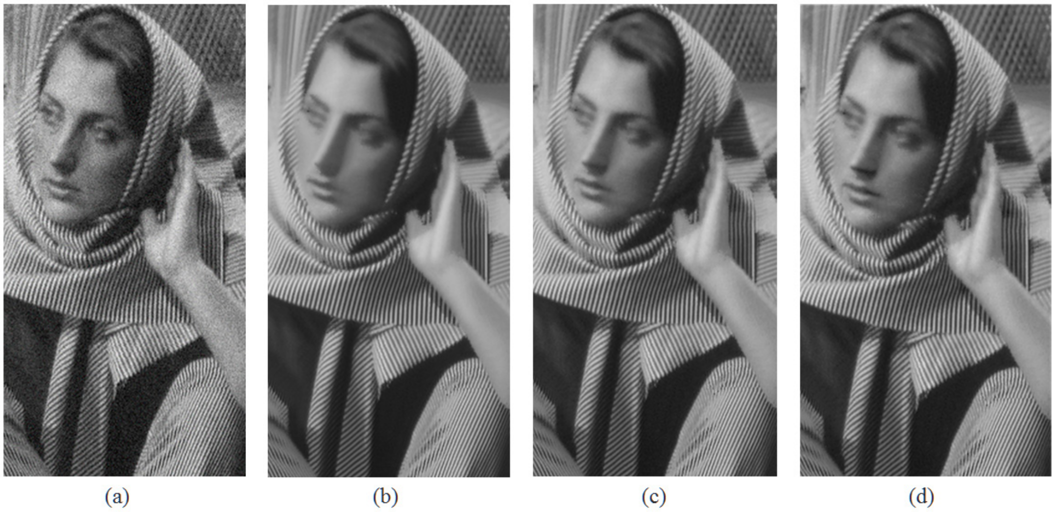


Figure 4.12: Decomposition of a noisy image (a) noisy image (b) structure I_S (c) structure + texture $I_S + I_T$ (d) structure + texture via histogram preservation

original one and then this restored edges are added. For this, the idea of gradient histogram preservation (GHP) in [37] is adopted, which is to impose a constraint that the processed image has the same gradient histogram as the estimated original one. Specifically, for the noisy image model:

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \tag{4.5}$$

where $\mathbf{x}$ is the original image, $\mathbf{n}$ is the noise, and $\mathbf{y}$ is the observed noisy image, the processed image is constrained to have similar gradient histogram as $\mathbf{x}$. In [37], considering the histogram of gradients of $\mathbf{y}$ as the discretization of the pdf of gradient distribution of $\mathbf{y}$, the gradient histogram of the original image $\mathbf{x}$ is found by solving

$$\arg\min_{H_x}\{\|H_y - H_x \otimes H_n\|^2 + c \cdot R(H_x)\} \tag{4.6}$$

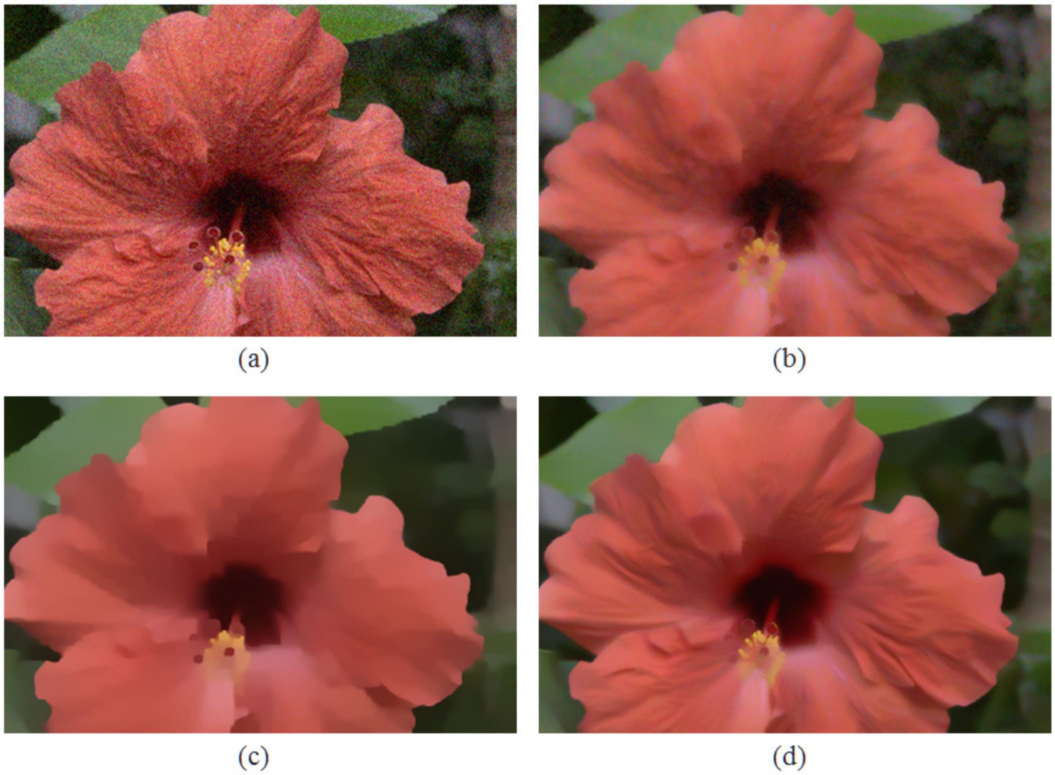


Figure 4.13: Edge-preserving smoothing of noisy image (a) noisy image (b) WLS [1] (c) RTV [2] (d) proposed

where H_x, H_y and H_n are the gradient histograms of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{n}$ respectively, $\otimes$ is the convolution operator, and $c \cdot R(H_x)$ is a regularization term. For solving this problem, H_y is estimated from the observed data and H_n is modeled as a hyper-Laplacian distribution as [37]:

$$p_x = k \exp(-\kappa |x|^\gamma) \tag{4.7}$$

where k is normalization factor.

Note that the high band image texture I_T in the proposed scheme is also a kind of gradient image, where the above GHP approach can be applied. Applying the proposed decomposition to eq. (4.5), the high band relationship is determined as

$$\mathcal{D}(\mathbf{y}) = \mathcal{D}(\mathbf{x}) + \mathcal{D}(\mathbf{n}) \tag{4.8}$$

where $\mathcal{D}(\cdot)$ is the operator that extracts the texture of the input image, and hence $\mathcal{D}(\mathbf{y}) = I_T + I_N$ and $\mathcal{D}(\mathbf{x}) = I_T$ in the problem. Like the GHP approach, the histogram of $\mathcal{D}(\mathbf{x})$ is found so that the histogram of $\hat{I_T}$ is matched to this one. Denoting the histogram of $\mathcal{D}(\mathbf{x})$ as the "reference histogram" H_r obtained in a similar manner as eq. (4.6), except that the positive and negative coefficients are considered separately in order not to diminish the peaks of coefficients that appear around the edges. To be specific, H_r is obtained as

$$H_r = \arg\min_{H}\{\|H_{y,+} - H \otimes H_n\|^2 + \|H_{y,-} + H \otimes H_n\|^2 + c \cdot R(H_r)\}. \tag{4.9}$$

where $H_{y,+}$ is the histogram of positive values in $\hat{I_T}$, $H_{y,-}$ for the negative values, and H_n is the histogram of $\mathcal{D}(\mathbf{n})$ that is modeled as eq. (4.7). The range of parameters for solving this problem is set the same as [37], i.e., $\kappa \in [0.001, 3]$ and $\gamma \in [0.02, 1.5]$. Then, the histogram of I_T is matched to H_r, which is denoted as $\mathcal{HE}(I_T)$ in Fig. 4.11 and the edge enhanced texture is obtained as $\hat{I_T} = \lambda \cdot I_T + (1-\lambda) \cdot \mathcal{HE}(I_T)$ occasionally.

4.2.3 image denoising via subband BLF

For fast and practical image denoising, smoothing operators in the 4.11 is replaced with Laplacian pyramid in (2.2) and the proposed subband BLF respectively. The overall process is described in 4.15 where $\lambda = 1$ corresponds to the proposed Laplacian subband filtering explained above, and $0 \leq \lambda < 1$ gives the edge enhanced results by histogram preservation. Laplacian pyramid is used to decompose an image into a structure layer and a texture/noise layer and then the proposed subband

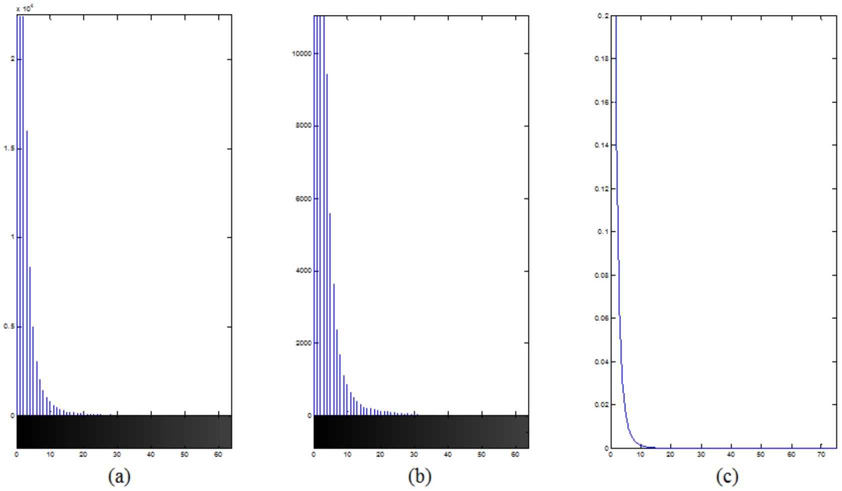


Figure 4.14: Histograms of texture layer I_T and modeled histogram H_r (a) histogram of input texture I_T (positive coefficients) (b) histogram of input texture I_T (negative coefficients) (c) histogram modeled as a hyper-Laplacian

BLF is applied to a texture/noise layer. The proposed subband BLF (SBLF) is designed to be performed on a texture layer, which is used to decompose a texture/noise layer into a texture layer and noise. By using this pixel-based method instead of the proposed patch-based work, the time complexity is reduced while denoising effectively.

For a given input image, subband decomposition is performed first as eqs. (2.1) and (2.2) to obtain the low band signal as a structure layer $L_2 = I_S$ and high band $L_1 = I_T + I_N$. For the low band image L_2, we apply the conventional BLF with $\sigma_d = 1.8$ and $\sigma_r = \sigma$ as suggested in [32], where σ is the noise variance. As stated above, the filtering scheme for the high band image L_1 is focused on, especially at the edge area. The basic idea is to give larger weights to the pixels that have similar edge intensities as well as pixel intensities. Also, when it is highly probable that a

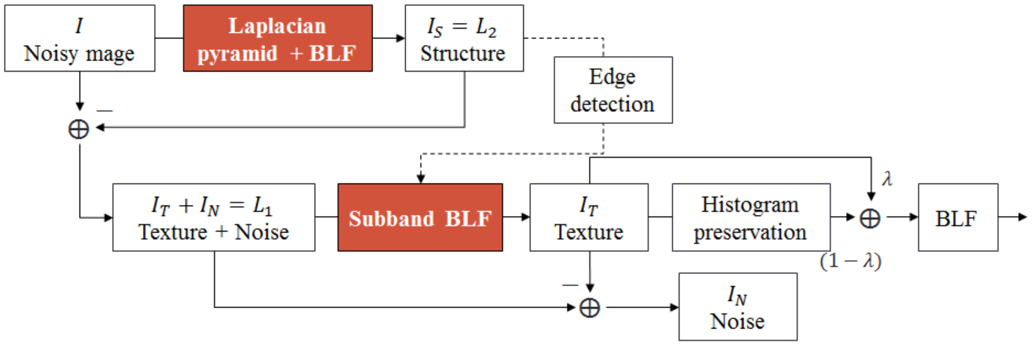


Figure 4.15: Noisy image decomposition via subband BLF

pixel is on the edge, it needs to be less affected by the neighboring pixels. These ideas are encoded into the photometric kernel of BLF as

$$w(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{\|\mathbf{p}-\mathbf{q}\|^2}{2\sigma_d^2}\right)\exp\left(-\frac{\|I(\mathbf{p})-I(\mathbf{q})\|^2+\|h(\mathbf{p})-h(\mathbf{q})\|^2}{4\sigma_r^2(\mathbf{p})}\right) \quad (4.10)$$

were $\sigma_r^2(p)$ is the pixel dependent bandwidth and $h(\mathbf{p})$ is the intensity of the pixel $\mathbf{p}$ in the histogram-equalized image of L_1 which will be explained later in more detail. Comparing this kernel with that of the original BLF in eq. (2.4), it can be seen that the spatial term (first term) is unchanged and only the photometric kernel (second term) is different. The bandwidth for the photometric kernel is now pixel dependent and the pixel difference in the high band ($\|h(\mathbf{p}) - h(\mathbf{q})\|$) is considered in weight control.

For these edge dependent modifications, edge information is extracted from the BLF of low band image L_2, which is denoted as $\hat{L_2}$. For determining whether a pixel is an edge pixel or not, the Laplacian of Gaussian filter and then thresholding are applied. Specifically, $\hat{L_2}$ is convolved with the kernel defined as

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.11}$$

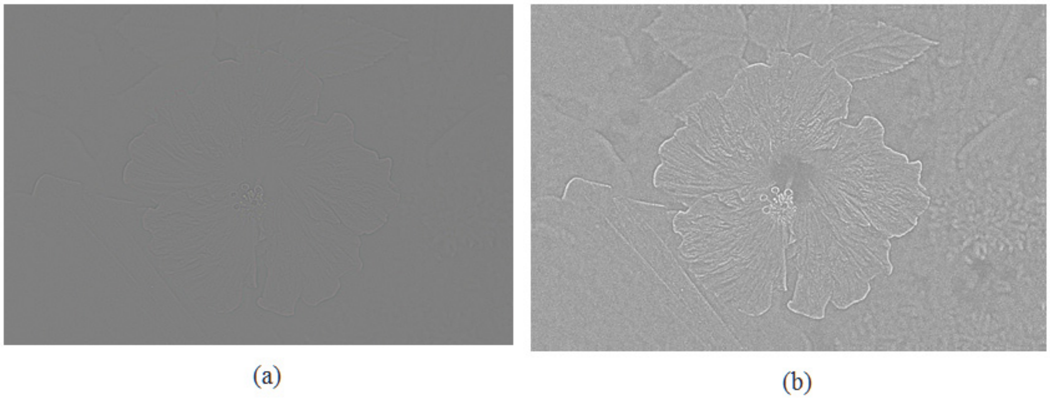
and then the output pixels larger than 75% of the mean value are considered the edge pixels. This gives an edge map $E(p)$ which is 1 when the pixel p belongs to an edge, and 0 if not. For simplicity, the edge map is obtained from the approximate intensity component (image of $(R+G+B)/3$), and this edge map is applied to all of color components equally. With this edge map, the kernel bandwidth is determined as

$$\sigma_r(\mathbf{p}) = \begin{cases} \sigma & \text{if } E(\mathbf{p}) = 1 \\ 2\sigma & \text{otherwise.} \end{cases} \tag{4.12}$$

It can be seen that the kernel bandwidth is small when the pixel is on the edge, so that the neighboring pixels less contribute to the averaging and thus the edge intensities are less changed. Conversely, the pixels in the flat areas are more strongly filtered than the edge pixels.

The second modification is the inclusion of $\|h(\mathbf{p})-h(\mathbf{q})\|$ in the photometric term, where $h(\mathbf{p})$ is the pixel value of histogram equalization of L_1, in order to assign large weights to the pixels with similar edge intensities as shown in fig. 4.16. The reason for using histogram equalized high band image is to make the contribution of edge intensity as large as the pixel intensity and also to stress the strong edges. More specifically, instead of directly using $L_1(\mathbf{p})$ in the photometric term, its histogram-equalized intensity $h(\mathbf{p})$ is used, which is normalized into the range of $[0, 255]$ (Note that $L_1(\mathbf{p})$ can have negative value). Denoting the output of proposed BLF of L_1 as $\hat{L}_1$, the final denoised image is obtained as $\hat{L}_1 + \hat{L}_2$.

Throughout the experiments, it is found that the low-band (L_2) filtering with a variety of parameter changes does not much affect the overall performance. Hence, just the original BLF with $\sigma_r = \sigma$ is applied for the low-band filtering.


(a)

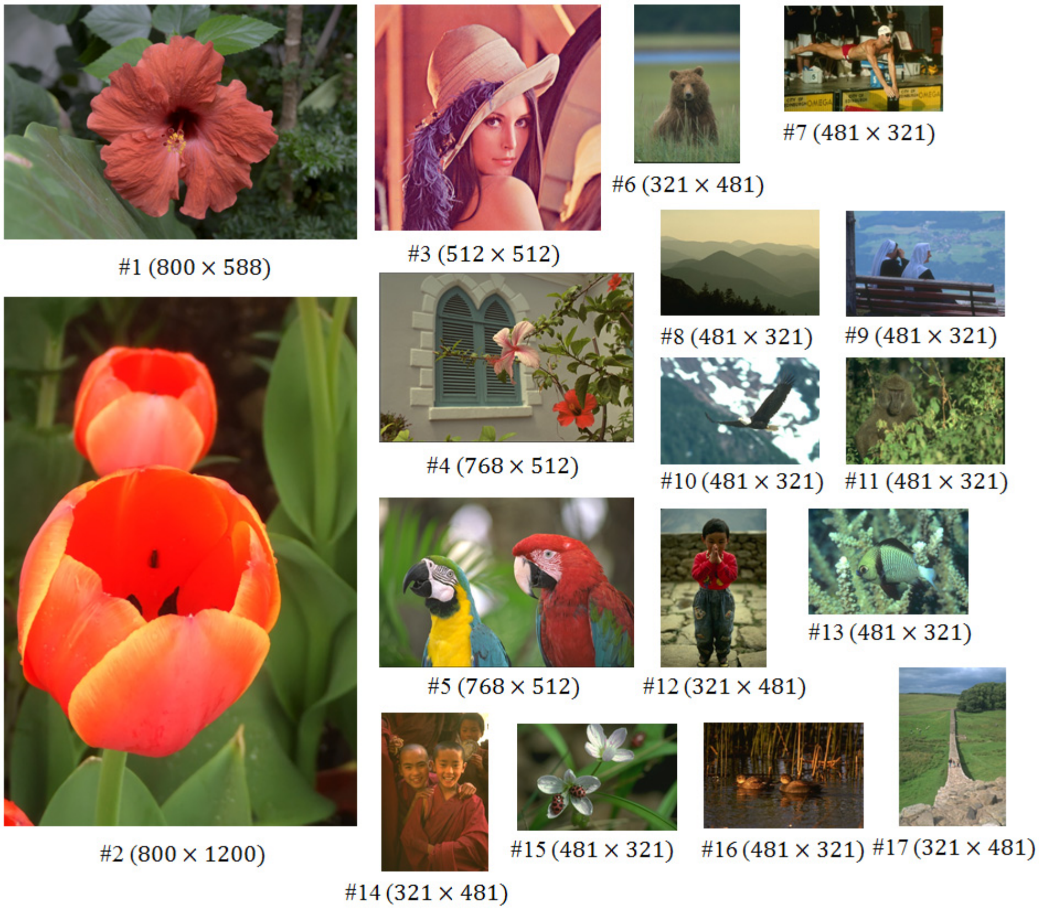
(b)

Figure 4.16: the effect of histogram equalization (a) L_1 (b) h

4.2.4 Experimental results of image denoising

Experiments on pseudo white and Poisson noise

To evaluate the performance of the proposed method, several images in Fig. 4.17 are tested. Each image is corrupted by the additive Gaussian noise (AWGN) with σ_n and Poisson noise with Q and restored by the proposed bilateral filter. We compare the proposed subband BLF algorithm with the original bilateral filter proposed [11], multiresolution bilateral filter [32], BLS-GSM [27], NLM [28], and BM3D [5] with the authors' source code. According to [32], we set $\sigma_d = 1.8$ and $\sigma_r = 2\sigma_n$. Also 11×11 windows are used for the original BLF, multiresolution BLF (MBLF), and proposed method. MBLF is implemented in MATLAB only and others are implemented in MATLAB and C++ through MATLAB MEX functions. These tasks are performed on a PC with Intel Core i5 CPU and 4 GB RAM.



#1 (800 × 588)
#2 (800 × 1200)
#3 (512 × 512)
#4 (768 × 512)
#5 (768 × 512)
#6 (321 × 481)
#7 (481 × 321)
#8 (481 × 321)
#9 (481 × 321)
#10 (481 × 321)
#11 (481 × 321)
#12 (321 × 481)
#13 (481 × 321)
#14 (321 × 481)
#15 (481 × 321)
#16 (481 × 321)
#17 (321 × 481)

Figure 4.17: Test images from Kodak dataset, BSDS 500 [7], etc.

Table 4.1 describes denoising results of images corrupted by the additive Gaussian noise (AWGN) with $\sigma = 20, 30, 40, 50$. As shown in the Table 4.1, the proposed method yields better results than BLF and MBLF. Further, whereas it is a kind of pixel-based methods which is generally more vulnerable but more practical than patch-based ones, it outperforms a typical patch-based method, NLM in the case of high noise variance. Although its PSNRs are lower than BM3D, it has the advantages in terms of computation time and complexity as seen in 4.1. Fig. 4.18

shows one of restored results. It demonstrates that subband BLF promises better visual quality than other methods and comparable results with BM3D despite lower PNSR. Besides, Fig. 4.18 (h) assures that the process of image enhancement provides pleasing results.

Table 4.2 also shows denoising results when images are corrupted by Poisson noise. We added Poisson noise with $Q \in \{5, 10, 15\}$ to each image. It is verified that subband BLF outperform BLF, MBLF, and even NLM.

Table 4.1: PSNR comparison results for additive white Gaussian noise with σ_n

Image	σ_n	Local self-similarity					Nonlocal self-similarity	
		BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	20	31.6794	31.9323	33.8950	33.8895	33.9912	33.3431	35.8375
	30	29.3073	29.4261	31.7667	31.8053	31.9595	30.7662	33.6504
	40	27.4443	27.0962	29.6679	29.9504	30.3387	28.6293	31.5308
	50	26.0250	25.0727	28.1345	28.3033	28.8437	26.8978	29.9591
2	20	32.9749	33.2897	35.5475	35.2767	35.5815	35.4776	36.6106
	30	29.8528	29.5258	32.1670	31.9508	32.1319	31.8976	33.0533
	40	27.5926	26.6223	29.4518	29.3156	29.7362	29.2211	30.2152
	50	25.8208	24.3292	27.3741	27.1749	27.7712	27.0151	28.0509
3	20	29.5742	30.3730	31.5846	30.9477	31.2679	31.8571	32.9763
	30	27.4829	28.6958	29.6329	29.5150	29.7952	29.9456	31.4283
	40	25.9551	27.2244	28.0216	28.2743	28.4112	28.1528	29.7727
	50	24.7668	25.9430	26.6088	27.1353	27.4058	26.4483	28.7892
4	20	28.9022	30.0833	31.9435	30.5835	31.3030	32.6272	34.7826
	30	26.6148	27.6982	29.5136	28.7128	29.3479	29.7242	32.3814
	40	25.0913	25.8124	27.6326	27.3614	27.8489	27.2984	30.2180
	50	23.9548	24.2227	26.0823	26.1541	26.4535	25.1335	29.0437
5	20	30.9226	31.6095	33.3512	32.1059	32.5655	33.4586	35.4615
	30	28.3882	29.0038	30.6294	30.0301	30.6165	30.8806	33.2934
	40	26.5845	26.7319	28.6867	28.5237	29.2085	28.9082	31.2421
	50	25.2343	24.7728	26.8821	27.2045	27.8678	27.1664	29.8016
6	20	31.3201	32.6138	33.8225	33.5283	33.6509	33.7352	35.8833

	30	29.0220	30.4443	32.4017	32.0848	32.1853	31.9146	34.0272
	40	27.2790	28.2125	31.3444	30.7425	31.0878	30.2546	32.4741
	50	25.8628	26.2051	30.1969	29.4428	30.0478	28.7224	31.7233
7	20	30.4979	31.2362	33.2919	31.9643	32.4950	33.1599	35.7414
	30	28.1647	28.7057	31.2204	30.3517	30.8560	30.7644	33.6985
	40	26.5483	26.5278	29.4311	29.1206	29.7163	28.8654	31.8135
	50	25.2710	24.4774	27.8250	27.8153	28.3661	27.1031	30.4806
8	20	30.3259	30.7226	31.6622	32.3661	32.7538	32.4851	34.1278
	30	27.6526	27.8349	29.1567	29.7130	30.1721	29.5019	31.0695
	40	25.6668	25.5379	27.1845	27.5696	27.9187	27.1221	28.7032
	50	24.1793	23.6836	25.2317	25.8535	26.3615	25.0768	27.0269
9	20	28.5583	28.9596	30.6504	30.1767	30.6660	31.2895	33.2452
	30	26.4573	26.3106	28.0796	28.1855	28.5467	28.2735	30.5383
	40	24.9711	24.2361	25.8615	26.4964	26.7047	25.7901	28.1555
	50	23.7114	22.4669	24.2702	24.9009	25.0663	23.6596	26.3629
10	20	28.9684	29.6138	31.3675	31.6605	31.6543	31.2497	33.3269
	30	27.1900	27.6733	29.2816	29.9158	29.7016	28.7399	31.2765
	40	25.8052	25.7907	27.5601	28.3284	28.1986	26.3427	29.4868
	50	24.7704	24.0634	26.1686	26.9749	27.0048	24.4219	28.4078
11	20	29.7327	30.1930	31.8022	31.6099	31.8518	31.7344	34.1093
	30	27.5835	27.6991	29.2003	29.4726	29.6222	28.5796	31.4683
	40	26.0026	25.6153	27.4096	27.7887	27.8895	26.3862	29.3017
	50	24.7143	23.8605	25.7574	26.3342	26.5497	24.6990	27.7145
12	20	28.5115	29.0784	30.9551	30.3149	30.5899	31.1675	33.2284
	30	26.4951	26.9067	28.7451	28.5976	28.7228	28.8376	30.9240
	40	25.1260	25.1497	27.0012	27.3580	27.4152	26.9429	29.2662
	50	24.0792	23.5972	25.6690	26.2281	26.2232	25.1579	28.2475
13	20	31.5452	32.0284	33.7145	33.6012	34.0474	33.7485	36.1165
	30	29.1706	29.2877	31.1478	31.3642	31.8337	30.7781	33.4574
	40	27.3659	26.8768	28.9921	29.4073	30.0110	28.3528	30.9108
	50	25.8349	24.6919	27.2418	27.5212	28.0419	26.2902	28.7832
14	20	31.0387	31.3631	33.0023	32.5708	33.1066	33.7123	35.9191
	30	28.6250	28.5758	30.3031	30.5868	31.1799	30.5336	33.2031
	40	26.8590	26.0783	28.3290	28.8154	29.4035	28.2653	30.7346
	50	25.5486	24.0992	26.6015	27.3602	27.9302	26.4955	29.1329

15	20	29.2370	29.5998	30.8136	30.5740	30.7395	30.9515	32.9437
	30	27.3628	27.9516	29.2355	29.2343	29.2801	28.9399	31.1266
	40	25.9133	26.5167	28.0673	28.2442	28.3154	27.5401	29.6110
	50	24.7727	25.1012	27.1200	27.2924	27.4839	26.3898	28.8131
16	20	27.8444	27.6034	29.1273	28.6031	29.0561	29.5986	32.0654
	30	25.9188	25.2429	26.7201	26.7651	27.1110	26.3124	29.3661
	40	24.5894	23.4094	24.7600	25.3502	25.3923	24.0598	27.1518
	50	23.5213	21.8728	23.3981	24.1003	24.0880	22.4760	25.3342
17	20	32.7103	34.5430	35.6234	35.6674	36.1768	36.4398	37.5250
	30	29.9369	31.2216	33.4702	32.9760	33.3581	33.6113	34.5780
	40	27.8516	28.5983	31.3656	30.6255	31.1115	30.9495	31.8928
	50	26.2928	26.4323	29.5636	28.7517	29.3792	28.8643	30.0609
Avg.	20	30.2555	30.8731	32.4797	32.0847	32.4410	32.7080	34.7000
	30	27.9544	28.3649	30.1572	30.0742	30.3777	30.0001	32.2671
	40	26.2733	26.2375	28.2804	28.4278	28.7475	27.8283	30.1459
	50	24.9624	24.4054	26.7133	26.9734	27.3462	26.0010	28.6901
Avg. time	(sec)	0.2971	10.3835	17.2237	1.0496	1.6537	25.3411	5.8549

Table 4.2: PSNR comparison results for Poisson noise

Image	Q	Local self-similarity					Nonlocal self-similarity	
		BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	5	32.9474	32.2942	32.5604	33.9158	33.6460	31.7862	35.7180
	10	31.6517	30.4416	32.0477	32.3254	32.4932	29.8999	34.2759
	15	30.9226	29.1446	30.5474	31.3624	31.5622	28.6041	33.2375
2	5	32.6603	31.9920	32.5760	32.8464	33.1194	32.5680	33.3375
	10	29.8982	29.1885	29.9337	30.0923	30.4616	29.7621	30.5083
	15	28.3087	27.5458	28.2237	28.4888	28.8353	28.0551	28.8865
3	5	28.5304	29.6791	28.7796	30.1783	30.2822	28.8773	31.8382
	10	27.1771	28.2888	27.4379	28.7222	28.7241	27.3655	30.1377
	15	26.2929	27.3980	27.1803	27.6600	27.8185	27.0401	28.9333
4	5	28.0572	29.6012	29.5626	30.2409	30.8311	29.7546	33.6546
	10	26.4676	27.6728	28.2923	28.6199	29.1260	27.8181	31.7713

	15	25.7800	26.5266	27.7651	27.6787	28.0336	26.6028	30.4430
5	5	30.3216	30.8260	30.3239	31.3062	31.6425	30.3956	34.0762
	10	28.4768	28.7685	28.9643	29.4338	29.8800	28.8715	31.9897
	15	27.3614	27.3844	27.9394	28.2943	28.8375	28.0808	30.6739
6	5	31.9297	32.2599	31.4390	33.0248	33.0831	31.3576	34.7484
	10	30.8194	30.5589	30.8745	32.0504	32.1184	30.0446	33.6319
	15	30.2222	29.1207	31.2517	31.2596	31.3442	29.4280	32.8599
7	5	29.6739	30.4118	30.0423	31.2509	31.5222	29.3095	34.6661
	10	28.3809	28.2856	28.9907	29.8879	30.3524	27.7685	33.0593
	15	27.6936	26.8059	29.3206	29.0268	29.5240	27.4791	31.9049
8	5	28.2500	28.4255	28.3466	29.4297	29.6461	28.5906	30.1870
	10	25.8856	25.8762	26.1665	26.7587	26.9426	26.2118	27.2840
	15	24.4493	24.3873	24.8280	25.1588	25.3818	24.8496	25.6552
9	5	28.0706	29.1018	29.4688	30.1800	30.3597	29.4709	33.0635
	10	26.6741	27.2846	28.5283	28.7467	28.7843	27.2990	31.2016
	15	25.9656	26.1796	26.7397	27.9588	27.8463	26.0379	30.0464
10	5	28.8831	29.4276	30.0148	31.3568	31.1390	29.7278	32.6500
	10	27.8802	27.9964	28.7563	29.8511	29.3882	27.3240	31.1192
	15	27.4028	26.9265	27.9094	28.9117	28.5604	25.6081	30.2343
11	5	29.5365	30.0382	30.1752	31.5122	31.5537	29.3369	33.8069
	10	28.3256	28.3580	28.8217	30.0518	29.8965	26.9207	32.1791
	15	27.7447	27.2156	28.3706	29.1063	28.9979	26.1556	31.0028
12	5	27.6072	28.1619	28.1212	29.5238	29.5007	27.9824	31.7674
	10	26.3065	26.3717	27.0471	27.9643	27.8676	26.3422	29.8103
	15	25.5783	25.2363	26.6315	26.9470	26.8828	25.9462	28.6412
13	5	32.0150	32.4269	31.6687	33.3938	33.5789	32.6053	35.5010
	10	30.5937	30.6959	29.9627	31.7383	31.8907	30.0511	33.8554
	15	29.7145	29.4146	29.4981	30.6168	30.8686	28.4237	32.4779
14	5	31.2368	31.5797	31.6847	32.3691	32.5747	31.7102	34.8674
	10	29.6350	29.5338	29.8394	30.8186	31.1010	29.2284	33.0338
	15	28.7648	28.2227	28.1174	29.8880	30.2727	28.2336	31.8545
15	5	28.5837	28.7374	28.5589	29.9596	29.8765	28.1096	31.4455
	10	27.5850	27.3168	27.5229	28.6577	28.5360	26.7409	29.9984
	15	26.9981	26.3929	27.2907	27.8465	27.8287	26.7237	29.0602
16	5	28.1392	28.9779	28.6754	30.3397	30.1752	29.0209	33.0324

	10	26.6853	27.2935	27.4899	28.3641	28.3926	27.6298	30.9771
	15	25.9588	26.4298	26.9550	27.6548	26.6335	26.0549	30.1389
17	5	33.6342	33.7754	34.2991	35.1089	35.3384	33.9634	36.6987
	10	31.3931	31.5094	32.8083	32.8809	33.1721	31.7328	34.0991
	15	30.2021	30.0352	31.6157	31.2593	31.5348	30.2098	32.3833
Avg.	5	30.0045	30.4539	30.3704	31.5257	31.6394	30.2686	33.5917
	10	28.4609	28.5554	29.0285	29.8214	29.9487	28.2948	31.7019
	15	27.6094	27.3157	28.2461	28.7717	28.8684	27.2667	30.4961
Avg. time		0.3078	10.8430	16.8132	1.1335	2.0406	21.5664	6.9882

Experiments on mixed noise

We do some experiments on mixed noise. In the experiments, images are degraded by various noises mixed by Poisson noise, Gaussian noise, and impulse noise. Since all the algorithm except MBLF do not use noise estimation, we tested with several $\sigma_n \in \{10, 20, 30, ..., 70\}$ and chose the best result for each image. Table 4.3 shows the PSNR comparison results for denoising images added by 20% impulse noise, Table 4.4 shows the case of 20% impulse noise + Gaussian noise $\sigma_n = 10$, and Table 4.5 presents the case of 10% impulse noise + Poisson noise $Q = 10$. Through the results, it is demonstrated that the proposed algorithm works better than the state-of-the-art methods with less computation time.

Experiments on real noise

When an image is captured by camera in insufficient light condition, slow shutter speed or high ISO is required to produce a visible image. However, slow shutter speed lets more light reach to the lens of camera and at the same time it makes a captured image blurred without a tripod while another solution, high ISO induces

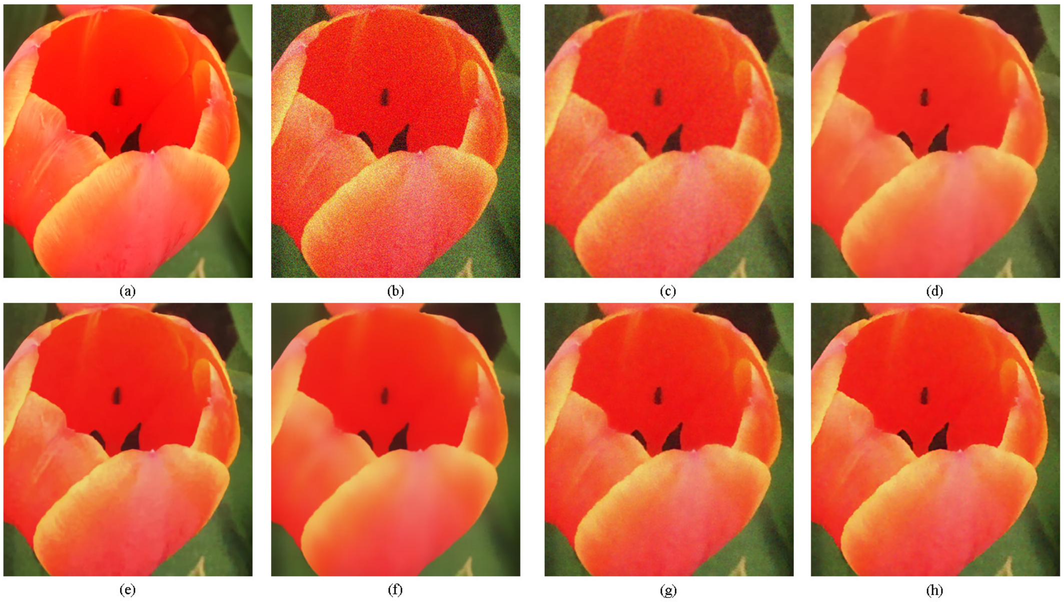


Figure 4.18: Results of denoising the image corrupted by AWGN $\sigma_n = 40$ (a) original image (b) noisy image (c) BLF (Lab), 26.6263 dB (d) MBLF, 26.6223 dB (e) BM3D 30.2152 dB (f) NLM 29.2211 dB(g) subband BLF, 29.3156 dB, (h) subband BLF + enhancement via subband histogram preservation, 29.7362 dB

Table 4.3: PSNR comparison results for mixed noise (20% impulse noise)

Image	Local self-similarity					Nonlocal self-similarity	
	BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	24.2018	22.3321	24.1887	24.8946	25.5637	23.6376	24.9905
2	25.2758	22.3321	23.1560	24.6518	25.1281	22.8560	24.9531
3	23.4455	23.5109	23.2620	24.7831	25.0209	23.5753	23.2513
4	24.3097	23.5109	24.3265	24.7915	25.4917	24.3059	24.8292
5	24.2705	23.0465	23.4489	24.9239	25.2416	23.7099	25.9463
6	24.5625	22.8714	26.7785	24.6312	24.9877	24.0875	24.9552
7	26.7080	24.5795	23.9426	27.2216	27.9954	25.9881	27.2638
8	27.4187	24.5795	22.8013	26.9462	27.4291	24.5127	27.1690
9	24.9481	23.2055	21.4086	24.7405	25.4531	23.2277	25.4997
10	23.1654	23.2055	22.9440	24.1613	24.6602	22.3793	23.8574
11	21.8711	22.1039	22.7081	22.3840	23.0829	21.1000	22.6501
12	24.2072	22.1039	23.8970	24.4611	24.7991	22.3230	24.5317
13	22.7976	21.2694	22.7738	23.8494	24.4214	22.3291	24.0320
14	23.7069	22.8539	23.0868	24.8992	24.9745	22.7604	25.3259
15	24.1294	22.8539	25.2027	24.2629	24.1815	21.0988	25.2427
16	23.8777	21.9349	20.6449	23.9269	24.8300	23.5069	23.5867
17	23.9372	24.5466	24.4997	23.9191	24.6917	22.8812	23.9613
Avg.	24.2843	22.9906	23.4747	24.6734	25.1737	23.1929	24.8262
Avg. time	0.3120	9.3545	16.8914	0.9256	1.5416	38.2350	9.0230

Table 4.4: PSNR comparison results for mixed noise (20% impulse noise + Gaussian noise $\sigma_n = 10$)

Image	Local self-similarity					Nonlocal self-similarity	
	BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	25.4552	22.1891	24.1444	24.7558	25.5305	23.5478	24.9216
2	23.7768	21.0752	22.8482	23.3077	24.4494	23.3204	22.9711
3	24.3323	23.4815	23.2735	24.7094	25.4398	24.2775	24.7898
4	24.2892	23.0925	24.3643	24.9520	25.2857	23.7272	25.9443
5	24.7113	22.8267	23.4414	24.6117	25.2226	24.0864	24.9208
6	27.5909	24.5983	26.7542	27.2233	28.0096	25.9194	27.2793
7	25.1151	22.3752	24.0196	25.0142	25.7913	24.5317	25.5305
8	23.5459	23.2134	22.7675	24.0731	24.5248	22.3253	23.8019
9	22.1189	19.7029	21.3331	22.3726	23.0503	21.0883	22.6279
10	24.3848	22.1769	22.9668	24.5119	24.8412	22.4003	24.5818
11	24.0068	21.3128	22.7450	23.8465	24.4128	22.4418	24.0698
12	24.1744	22.9088	23.9241	24.9614	25.0423	22.8593	25.3869
13	24.3663	22.0438	22.8545	24.0156	24.8735	23.5837	23.7107
14	24.2580	21.5176	23.0955	23.9533	24.7053	22.8613	24.0113
15	25.8520	24.5380	25.1631	26.0130	26.0355	24.5009	26.3651
16	21.6917	19.3832	20.7904	21.6338	22.1698	20.7998	21.8605
17	25.2929	22.4147	24.4745	24.8613	26.0848	24.3419	24.3968
Avg.	24.4096	22.2853	23.4682	24.4010	25.0276	23.3302	24.5394
Avg. time	0.2988	10.3789	16.7026	1.1930	1.8961	32.0858	8.3840

Table 4.5: PSNR comparison results for mixed noise (10% impulse noise + Poisson noise $Q = 10$)

Image	Local self-similarity					Nonlocal self-similarity	
	BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	28.4158	25.4233	27.4656	28.2487	29.2835	27.6470	29.1604
2	25.8129	23.6747	25.0903	25.5515	26.6082	26.0011	25.4205
3	25.1491	25.3809	24.9301	26.0402	26.5509	25.6082	26.7212
4	25.1560	24.7387	25.7451	26.3853	26.9163	26.2451	28.4774
5	26.0058	24.7914	25.1994	26.4017	27.1395	26.3872	27.7189
6	28.9993	26.8902	28.9912	29.5343	30.2031	28.8034	30.5975
7	26.4841	24.6991	26.4335	27.2403	28.0310	27.0361	28.9383
8	23.1222	22.8339	22.4087	23.5417	24.1565	23.3715	23.5693
9	24.2926	22.6400	24.2504	25.2764	25.9606	24.9965	26.2820
10	26.1391	24.4213	25.6495	26.9526	27.1979	25.2780	27.6961
11	26.1645	24.2364	25.2243	26.6826	27.2050	25.8733	27.6886
12	24.8639	23.8919	24.7310	25.8516	26.0308	25.0271	27.1157
13	27.2892	24.9263	26.2754	27.4459	28.3717	27.0570	27.8019
14	26.8431	24.4638	26.1080	27.1334	27.9553	26.7907	28.0307
15	26.3177	25.4183	25.7355	26.9212	27.0969	26.1018	27.8844
16	23.9856	22.1646	23.4939	24.6034	25.0621	24.3040	25.7068
17	27.4375	26.3678	27.7605	27.7814	28.9324	27.5485	27.8591
Avg.	26.0281	24.5272	25.6172	26.5642	27.2178	26.1221	27.4511
Avg. time	0.2996	10.3959	16.2441	1.0685	1.7206	33.1900	9.4430

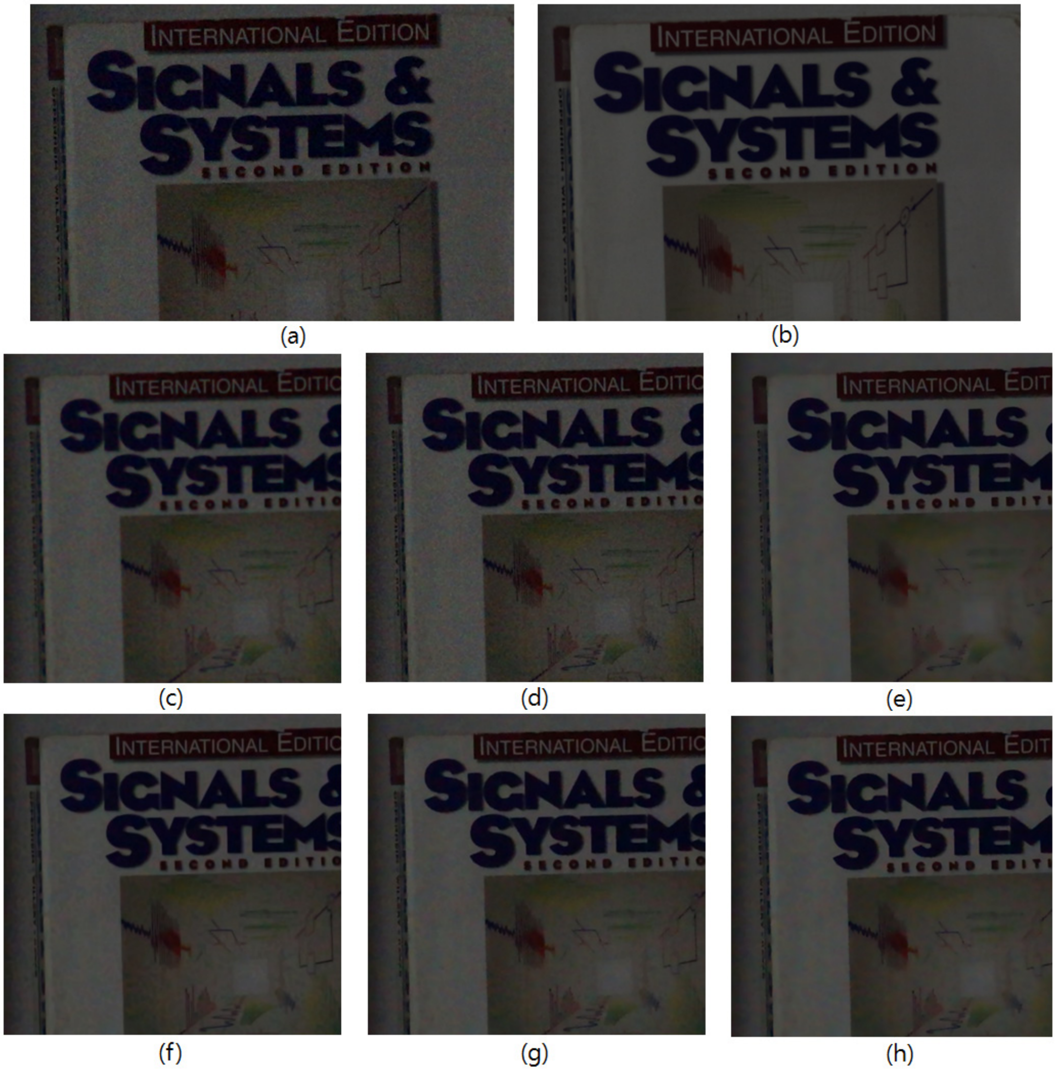


Figure 4.19: Image denoising of real noisy image result 1 (a) real noisy image (b) ground truth (c) BLF, 36.6848 dB (d) MBLF 35.5875 dB (e) NLM, 37.8776 dB (f) BM3D, 38.4951 dB (g) SBLF 37.8706 dB (h) SBLF+SHP, 37.7653 dB

Table 4.6: PSNR comparison results for real noise

Image	Local self-similarity					Nonlocal self-similarity	
	BLF	MBLF	BLS-GSM	SBLF	SBLF+SHP	NLM	BM3D
1	32.8029	31.68	33.0025	34.0362	34.0253	34.183	34.431
2	36.6846	35.5875	37.0479	37.8706	37.7653	37.8776	38.4951
3	22.6241	22.7788	22.2161	22.9023	22.8644	22.5071	22.6712
Avg.	30.7039	30.0154	30.7555	31.603	31.5517	31.5226	31.8658

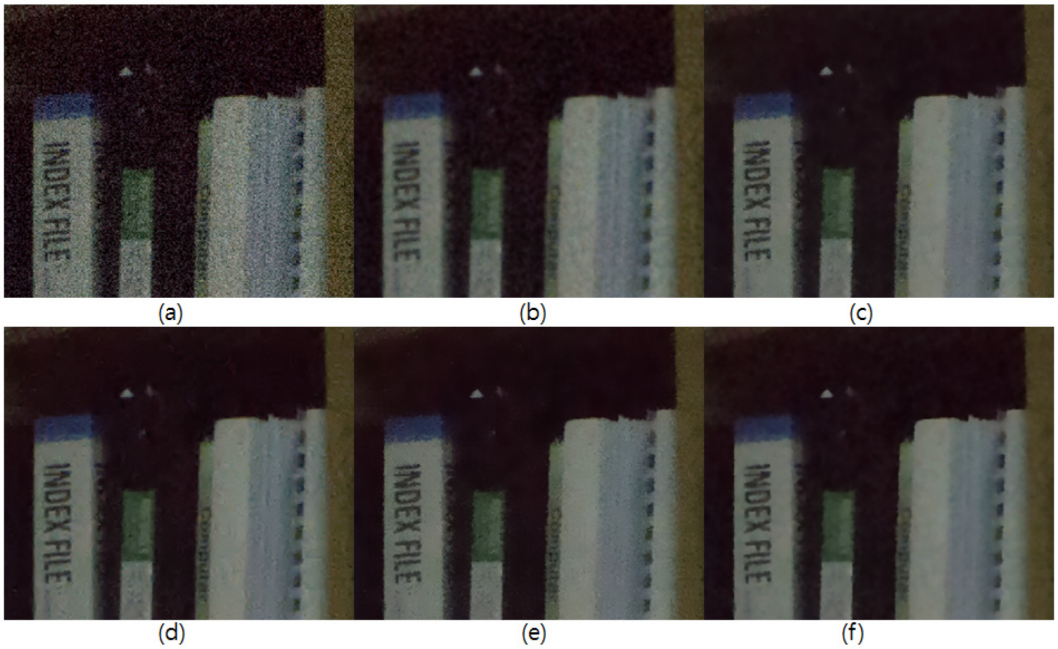


Figure 4.20: Image denoising of real noisy image result 2 (a) real noisy image (b) BLF (c) MBLF (d) BLS-GSM (e) NLM (f) SBLF+SHP

noise. So for experiments on real noise, we obtained several images captured in low light condition with high ISO and then denoised them by BLF, MBLF, NLM, BM3D and the proposed subband BLF. Table 4.6 illustrates PSNR results where the ground truth image is generated by averaging 30 images taken by fixed camera. It shows that the proposed method denoise real noise well. For comparing visual quality, Fig. 4.19 and Fig. 4.20 are also provided and it is presented that the proposed method gives visually pleasing results.

4.3 Image coding

4.3.1 Structure scalable image coding framework

When an image contains a lot of textures and/or noise, the performance of its compression becomes worse than the image with less textures [38, 39]. It is due to high frequency components in texture and noise. To improve image/video coding technique, video coding scheme using texture analysis and synthesis are studied [39, 39]. In this framework, texture region which can be synthesized is analyzed to encode non-texture region and texture region is reconstructed by texture synthesis. It is verified that it enables bit-savings, but texture synthesis techniques are limited to textures with repeated pattern. Hence, another effective coding scheme is required.

First of all, some experiments are done to investigate the effect of texture on image coding. Fig. 4.21 shows compression results of the original image (a) and extracted structure (b) via BLF. Also Fig. 4.22 (a), (b), (c) are the result images obtained by compressing the original image at given bit per pixel(bpp) and (d), (e), (f) are obtained by compressing the structure layer where the peak-signal-to-ratios (PSNRs) are computed between the original image and each compressed image.

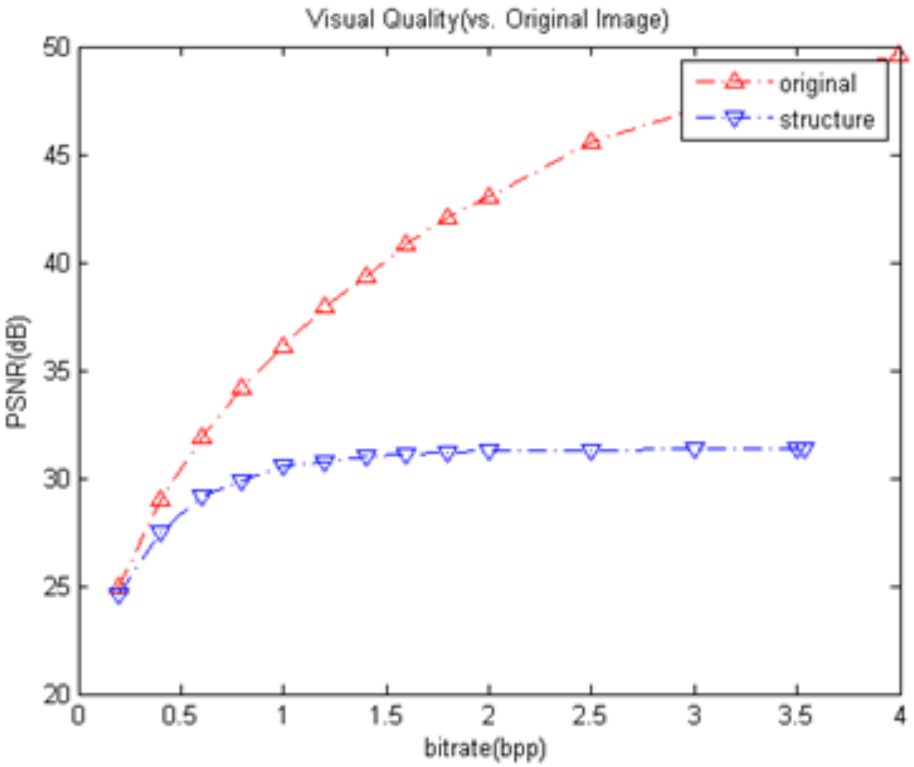


Figure 4.21: Rate-Distortion curve of compressing the original image and its structure layer

They show PSNRs of compressed structure image are lower, of course, but some artifacts induced by compression are reduced. In addition, it is known that the human visual is more attentive to salient objects [40, 41]. All things considered, we propose a new structure scalable image coding framework based on the proposed decomposition.

The proposed structure scalable image coding scheme is described in Fig. 4.23. The key is to remain textures and details only in salient region since human visual system is more sensitive to change in salient region, but it is not in other region. For

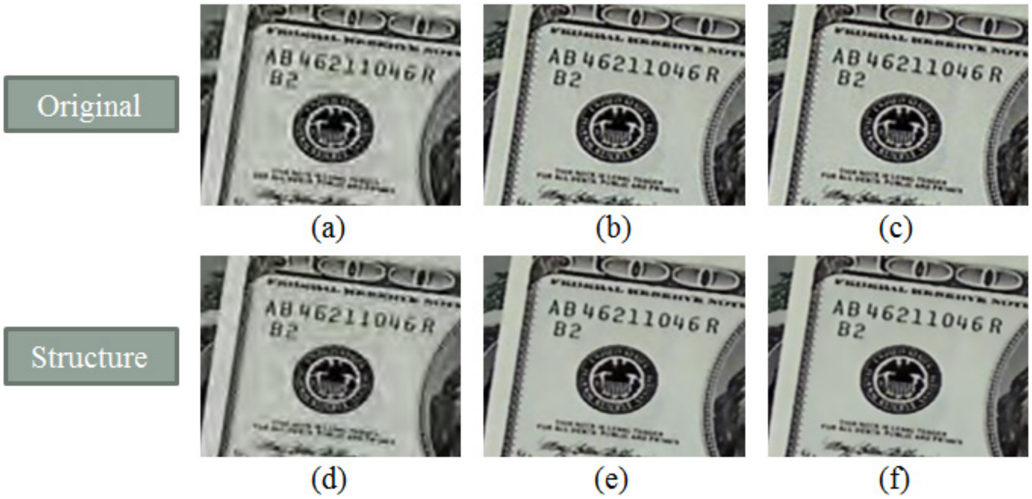


Figure 4.22: The result images obtained by compressing the original image and its structure layer, original image: (a)0.1983 bpp (24.94 dB) (b) 0.9979 bpp(36.12 dB) (c) 2.498 bpp (45.53 dB), structure layer: (d) 0.1981 bpp (24.63 dB) (e) 0.9983 bpp (30.55 dB) (f) 2.498 bpp (31.31 dB)

this, given an image, saliency detection proposed in [41]is preceded before extracting structure to generate a saliency mask. Then, the proposed decomposition method is adopted for separate the structure and the texture. With the saliency mask, the texture layer corresponding salient region recombined with the structure layer. Namely, textures are removed in the image except salient region and encode the recombined image. Meanwhile, this scheme is named as “structure scalable coding” since bit stream can be controlled according to the transmission channel condition. It is done by adjusting decomposition parameters which manipulate the degree of smoothing. For example, in the case of requiring more bit-savings, an image is smoothed a lot to leave large scale of structure.

To verify effectiveness of the proposed work, images are encoded by JPEG

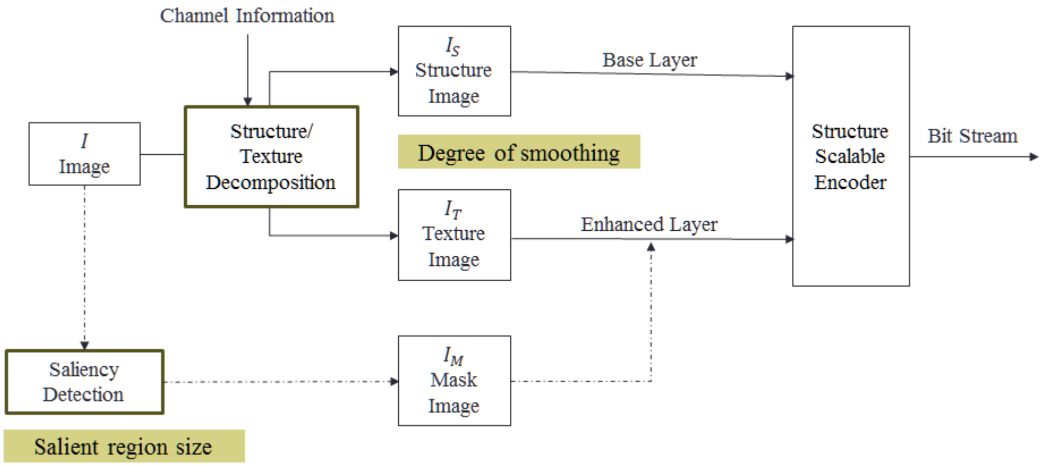


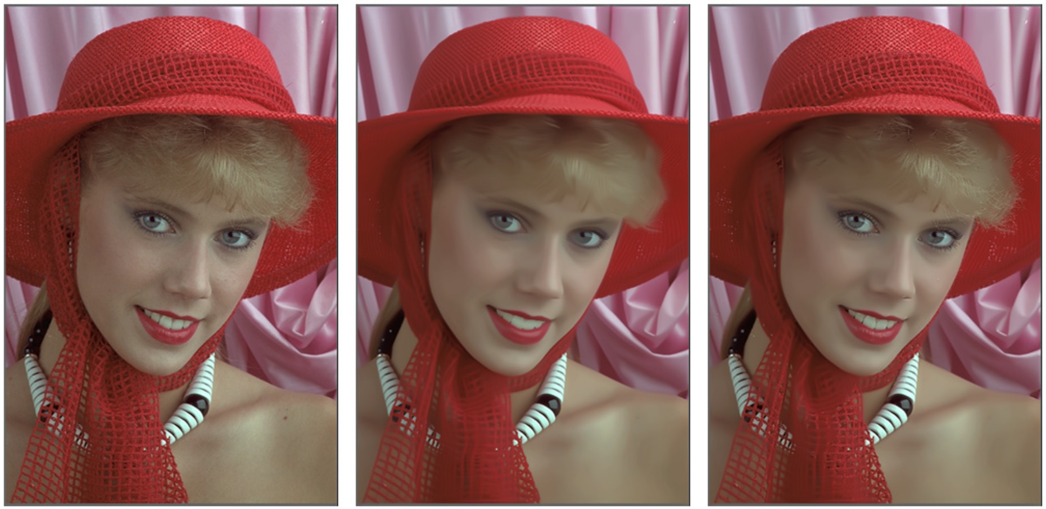
Figure 4.23: Block diagram of structure scalable image coding framework

2000 [42] and compared it with the compressed original images. Table 4.7 demonstrates lossless compression results of Kodak image. It represents bit per pixels (bpp) of compressing the original images (the second column), structure layers (the third column) and the images produced by the proposed framework (the fourth column). As shown in Table 4.7, compressing structure layers requires less bits than the original and the proposed scheme gives bit-savings up to 36%. Fig. 4.24 are for comparing subject visual quality where original image (a), structure image (b) and structure layer recombined salient texture (c). Through efficient saliency detection, Fig. 4.24 (c) shows a relatively small difference from the original image and even better quality with correction of skin texture. Fig. 4.25 depicts rate-distortion (R-D) curves of lossy compression of Fig. 4.26 (a). The PSNRs are computed between the original image and the compressed one. As shown in Fig. 4.25 (a), when the image is encoded through the proposed framework, the PSNRs of overall region at the same bpp with the original image compression are lower. However, it is demon-

Table 4.7: JPEG Lossless compression results of Kodak images

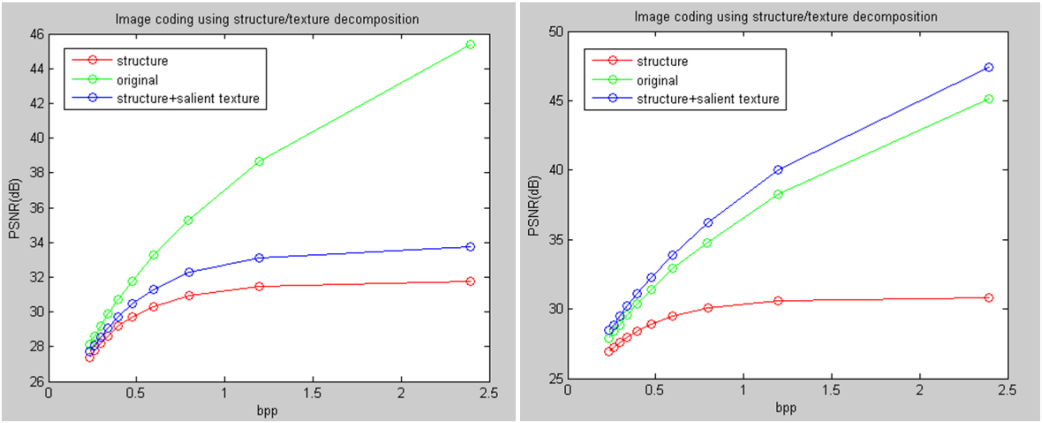
#img	Original image (bpp)	Structure image (bpp)	Structure+ salient texture (bpp)	Gain (%)
1	10.3842	8.934	8.9084	14.21
2	9.1627	5.3811	5.7978	36.72
3	8.0917	5.5926	5.8459	27.75
4	9.3595	6.5025	7.1130	24.00
5	10.8167	10.5243	10.2775	4.98
6	9.591	7.4153	7.3933	22.91
7	8.5037	6.6404	6.6028	22.35
8	11.1389	9.9328	9.8094	11.93
9	9.0531	5.6023	5.7665	36.30
10	9.2183	6.1483	6.1442	33.35
Avg.	9.532	7.2674	7.3659	23.45

strated that the proposed algorithm works better in the saliency-detected region (Fig. 4.25 (b)), which shows the proposed algorithm reduces compression artifacts and achieve efficient compression like Fig. 4.27, Fig. 4.28, Fig. 4.29, Fig. 4.30 and Fig. 4.31.



(a) (b) (c)

Figure 4.24: (a) original image (b) structure layer (c) structure + salient texture layer



(a) (b)

Figure 4.25: Rate-distortion curves (a) overall image (b) saliency-detected region

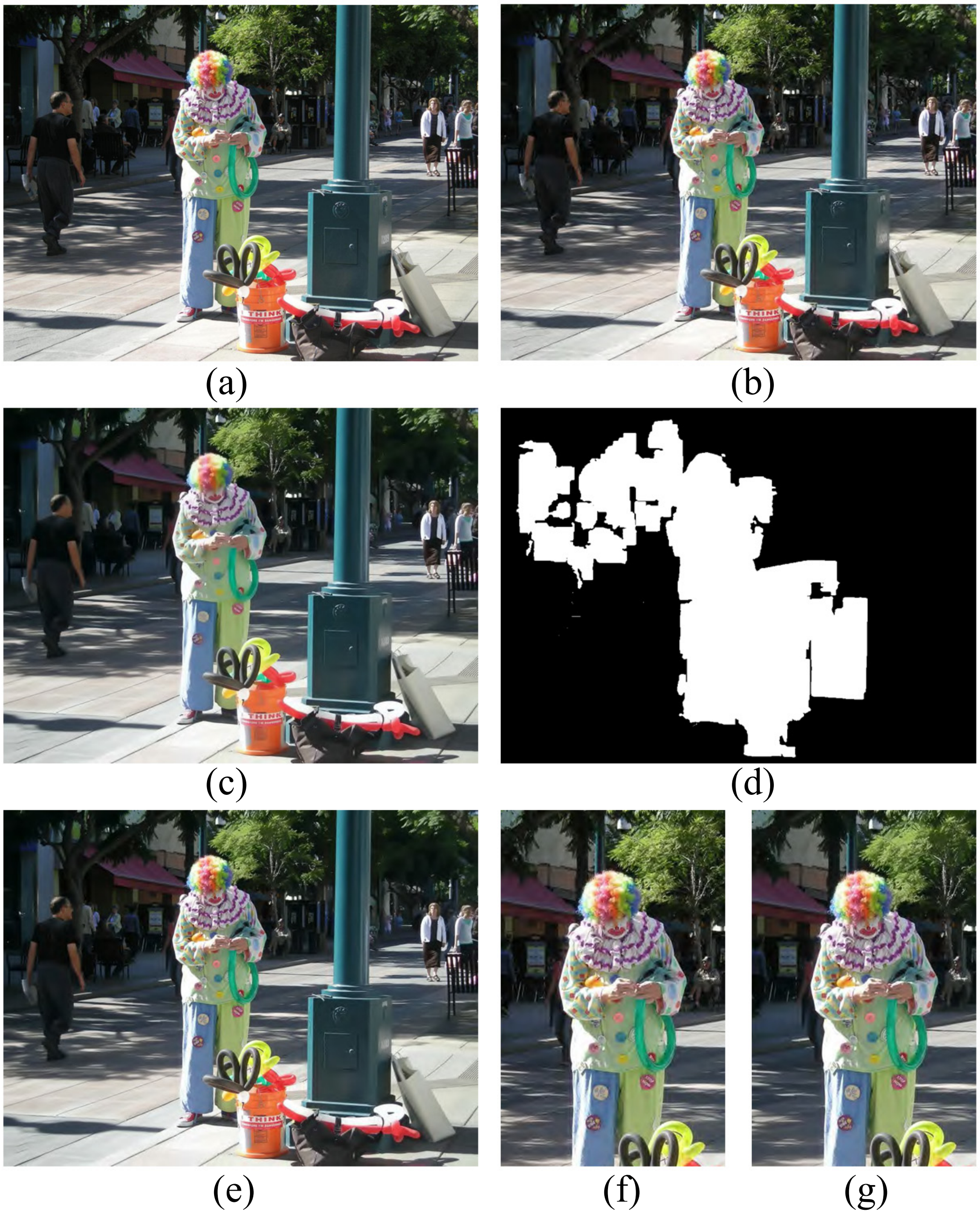
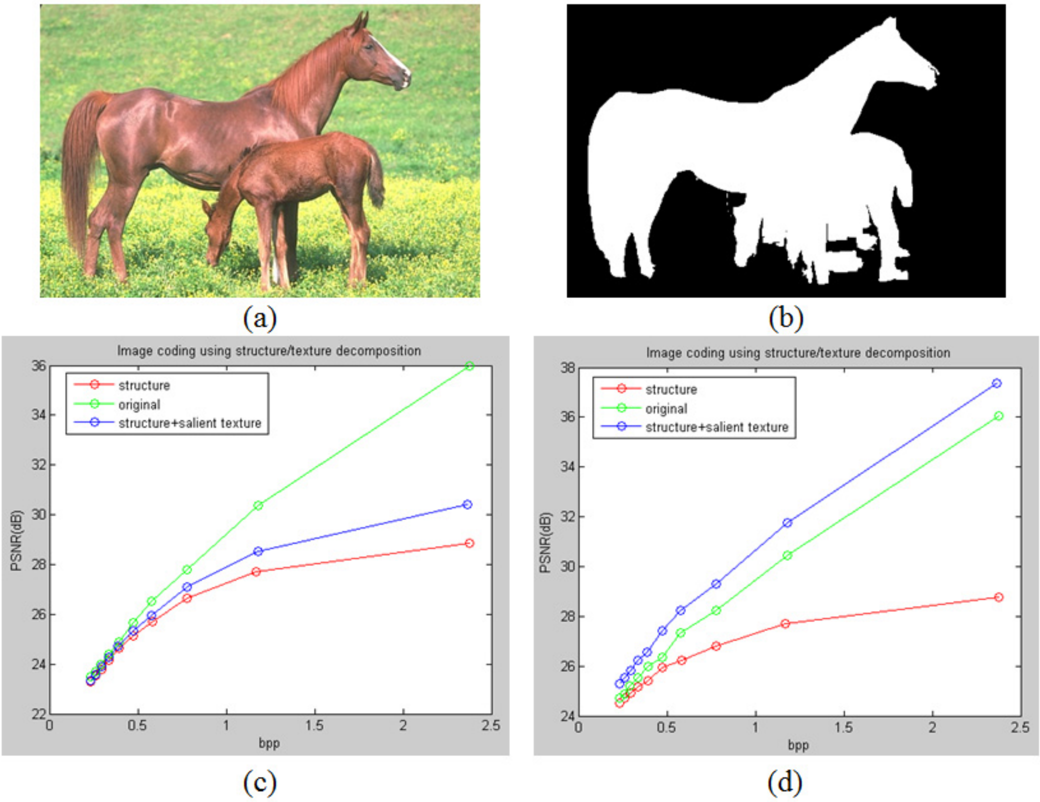


Figure 4.26: Lossy compression results 1 (bpp:0.3956) (a) original image (b) compressed original image (overall: 30.7200 dB, salient region: 30.3549 dB) (c) compressed structural image(overall: 29.1682 dB, salient region: 28.4062 dB) (d) saliency mask (salient region is expressed in white) (e) compressed structural image (overall: 29.1682 dB, salient region: 28.4062 dB) (f) close-up of (b) (g) close-up of (e)



(a)

(b)

(c)

(d)

Figure 4.27: Lossy compression results 2 (a) original image (b) saliency mask (c)R-D curve for overall region (d) R-D curve for salient region

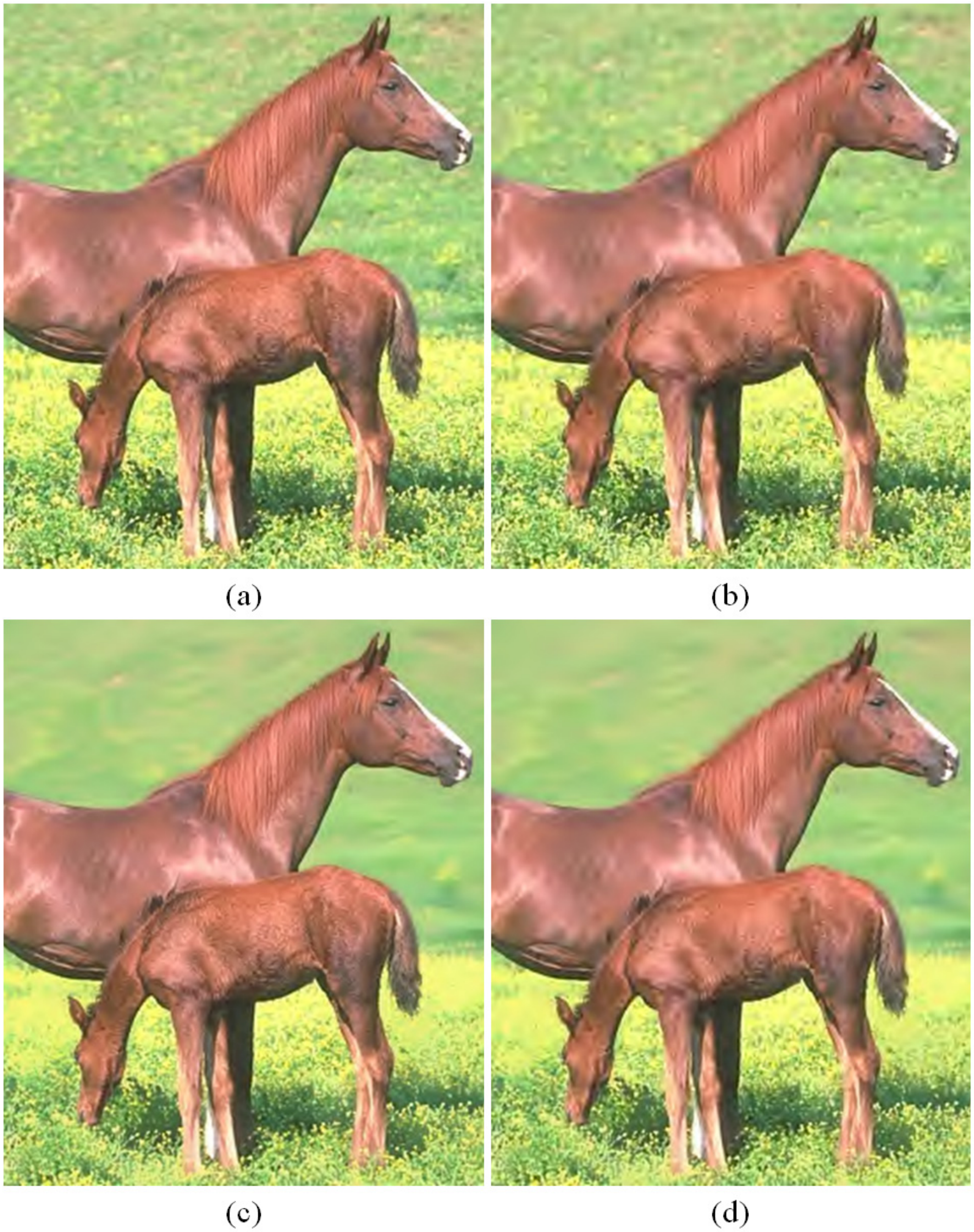


Figure 4.28: Lossy compression results images of Fig. 4.27 (a) compressed original image (20:1) (b) compressed original image (30:1) (c) compressed structure + salient texture (20:1) (d) compressed structure + salient texture

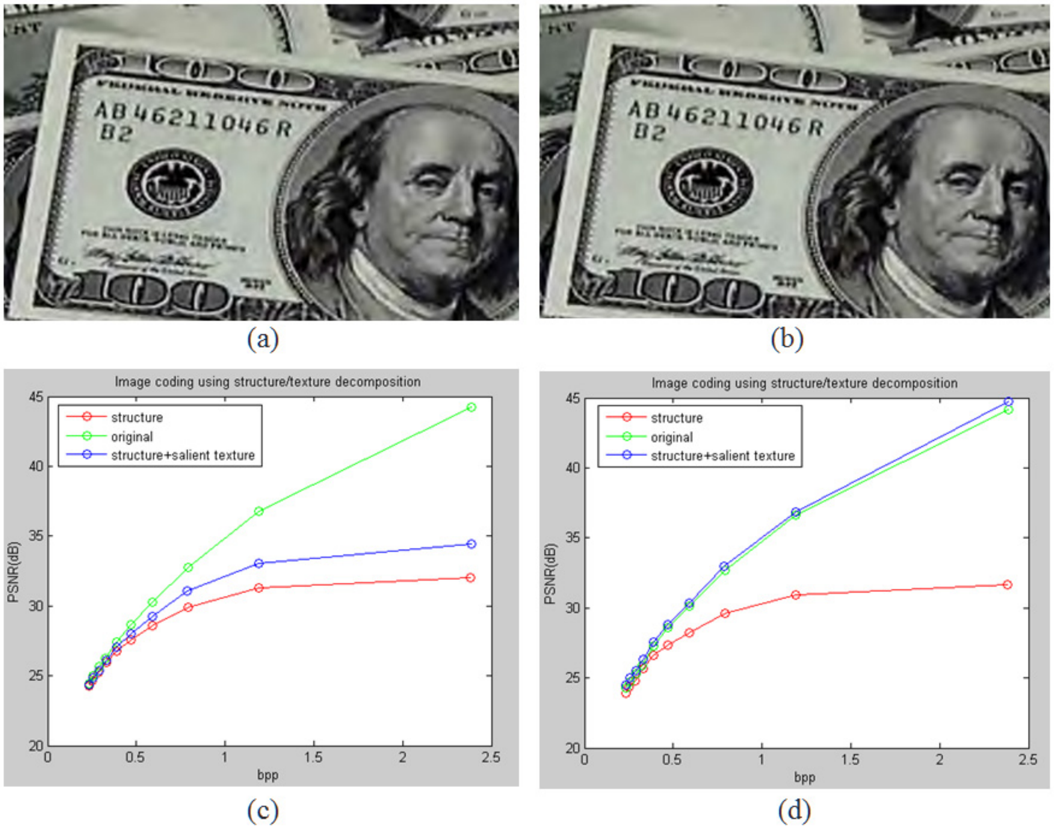
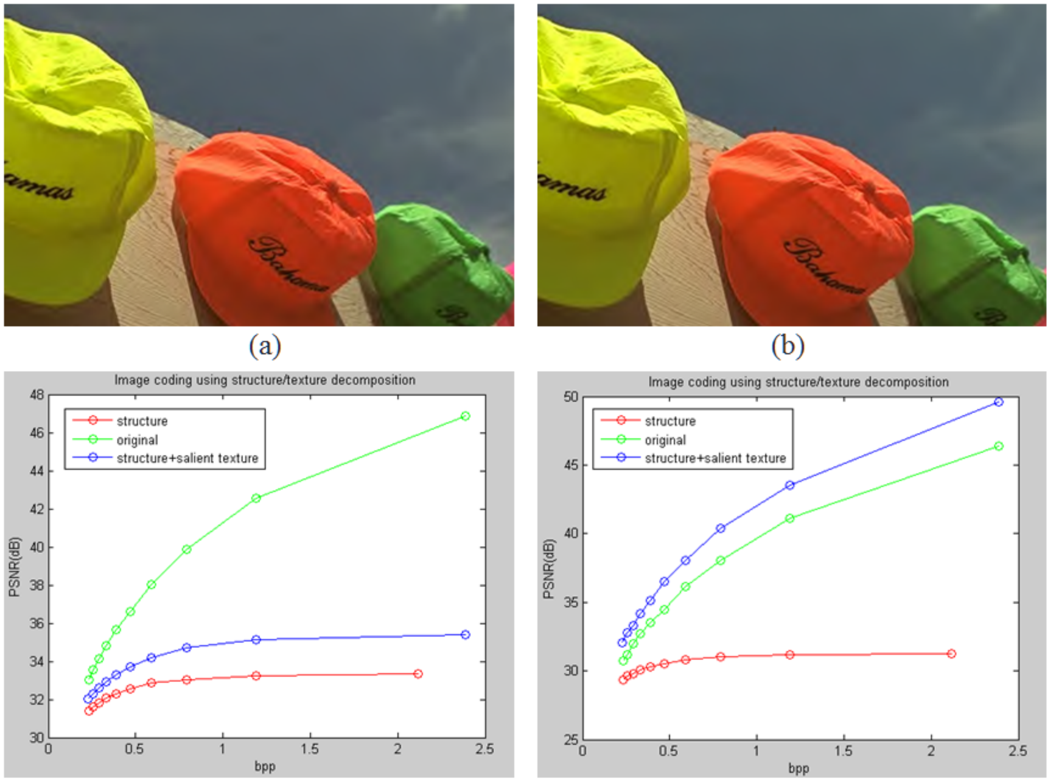

(a)
(b)
(c)
(d)

Figure 4.29: Lossy compression results images 3 (a) compressed original image (50:1) (b) compressed structure + salient texture (c)R-D curve for overall region (d) R-D curve for salient region



(a)

(b)

(c)

(d)

Figure 4.30: Lossy compression results images 4 (a) compressed original image (60:1) (b) compressed structure + salient texture (c)R-D curve for overall region (d) R-D curve for salient region

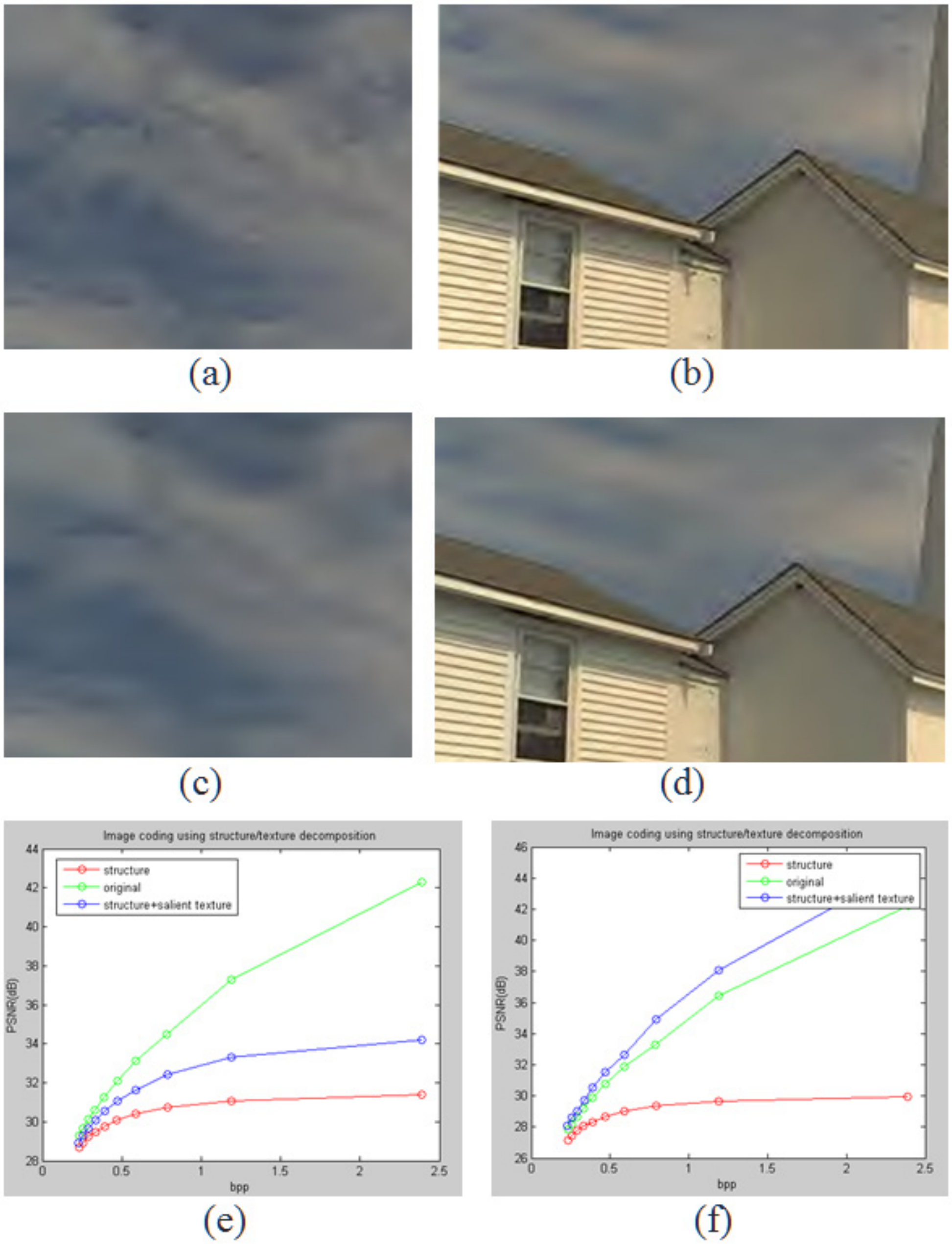

(a) (b) (c) (d) (e) (f)

Figure 4.31: Lossy compression results images 5 (a) compressed original image (70:1) (b) compressed original image (70:1) (c) compressed structure + salient texture of (a) (d) compressed structure + salient texture of (b) (e)R-D curve for overall region (f) R-D curve for salient region

Chapter 5

Conclusion

This dissertation has presented a patch-based image decomposition framework for decomposing an image into a structure layer and a texture layer. Like conventional methods, the proposed image decomposition is performed by edge-preserving smoothing where a structure layer can be obtained by applying smoothing filter to an image and a texture layer can be extracted by subtracting the structure layer from the original image. However, the proposed method is motivated by the concern that noise can be regarded as small scale of texture and image decomposition and image denoising are not separable problems. Hence it takes advantages of patch-based framework which is efficient in image denoising and edge-preserving smoothing is done by patch-based filtering. Also it tries to solve the over-smoothing problem existing in conventional patch-based framework by controlling number of patches used in filtering. Consequently, it provides pleasing results of image decomposition and facilitates extension to a noisy image decomposition.

The proposed filtering output is obtained through three main steps, i.e., collecting similar patches at each pixel, computing patch weights and updating patch and weights. In the step of collecting similar patches, a set of patches are selected for each pixel by measuring patch similarities and additional constraints. In the second step, the weights between each pixel and its similar patches are calculated. In the third step, the values of all the pixels in a patch are updated by considering both patch weights and local pixel weights. Hence, the filtering output can be computed by the weighted summation of pixels in similar patches. More specifically, given a pixel, a set of similar patches is found by measuring patch similarities based on mean squared error (MSE) and additional constraints. Then, patch weights between each patch and its similar patches are computed respectively in the similar way of conventional patch-based framework. With the patch weights, all the pixels in a patch are updated at the same time while considering local pixel weighs.

The proposed decomposition can be applied to many applications, such as detail enhancement, HDR tone mapping, image denoising, and image coding, etc. In detail enhancement, the proposed edge-preserving smoothing works well for extraction and enhancement of image detail. In HDR tone mapping, a typical framework is utilized where smoothing operator is replaced by the proposed smoothing operator and the dynamic range of a high dynamic range image is reduced for displaying the image into low dynamic range devices. For image denoising, a noisy input is decomposed into structure/texture/noise by the proposed method and the noise layer is discarded while the texture layer is restored through the histogram preservation process. Also a novel coding scheme named as "structure scalable image coding scheme" is proposed for efficient image coding. In the proposed image coding scheme, image decomposition and saliency detection are performed to encode structure layer and salient texture layer. Experimental results show that the proposed framework can perform well in many applications. In addition, it is verified that the proposed method is robust to the presence of noise. Also by adopting the proposed method in decomposition of a noisy image, both image denoising and image enhancement can be achieved in the proposed framework. Furthermore, the proposed method for image coding is efficient in reduction of compression artifact and improvement of the coding performance.

Bibliography

[1] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–67:10, Aug. 2008. [Online]. Available: http://doi.acm.org/10.1145/1360612.1360666

[2] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 139, 2012.

[3] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 147:1–147:9, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618493

[4] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 815–830.

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

[6] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 176, 2013.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[8] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, July 2002.

[9] S. Paris, S. W. Hasinoff, and J. Kautz, "Local laplacian filters: edge-aware image processing with a laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, p. 68, 2011.

[10] O. K. Al-Shaykh and R. M. Mersereau, "Lossy compression of noisy images," *Image Processing, IEEE Transactions on*, vol. 7, no. 12, pp. 1641–1652, 1998.

[11] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, Y. Smith, Ed. Stoneham: Butterworth-Heinemann, 1998, pp. 839–846.

[12] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "A gentle introduction to bilateral filtering and its applications," in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, pp. 1:1–1:50. [Online]. Available: http://doi.acm.org/10.1145/1401132.1401134

[13] M. A. Raanan Fattal and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, vol. 26, no. 3, p. to appear, 2007.

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

[15] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decompositionmodeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.

[16] S. Osher, A. Solé, and L. Vese, "Image decomposition and restoration using total variation minimization and the h 1," *Multiscale Modeling & Simulation*, vol. 1, no. 3, pp. 349–370, 2003.

[17] G. Deng, "A generalized unsharp masking algorithm," *Image Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1249–1261, May 2011.

[18] K. He, J. Sun, and X. Tang, "Guided image filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, June 2013.

[19] J. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 889–896, May 2000.

[20] M. Kaur, J. Kaur, and J. Kaur, "Survey of contrast enhancement techniques based on histogram equalization," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 2, no. 7, 2011.

[21] J. Stark, “Adaptive image contrast enhancement using generalizations of histogram equalization,” *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 889–896, May 2000.

[22] H. Cheng and X. Shi, “A simple and effective histogram equalization approach to image enhancement,” *Digital Signal Processing*, vol. 14, no. 2, pp. 158–170, 2004.

[23] J. M. DiCarlo and B. A. Wandell, “Rendering high dynamic range images,” in *Electronic Imaging.* International Society for Optics and Photonics, 2000, pp. 392–401.

[24] J. Tumblin and G. Turk, “Lcis: A boundary hierarchy for detail-preserving contrast reduction,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 83–90. [Online]. Available: http://dx.doi.org/10.1145/311535.311544

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

[26] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications.* Springer, 1990, vol. 84.

[27] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, “Image denoising using scale mixtures of gaussians in the wavelet domain,” *Image Processing, IEEE Transactions on*, vol. 12, no. 11, pp. 1338–1351, Nov 2003.

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

[29] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[30] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *Image Processing, IEEE Transactions on*, vol. 9, no. 9, pp. 1532–1546, 2000.

[31] B. Zhang and J. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, 2008.

[32] M. Zhang and B. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, 2008.

[33] Z. Li, J. Zheng, Z. Zhu, S. Wu, and S. Rahardja, "A bilateral filter in gradient domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 1113–1116.

[34] Q. Yang, "Recursive bilateral filtering," in *Proceedings of the 12th European conference on Computer Vision (ECCV)*, 2012, pp. 399–413.

[35] A. Wong and J. Orchard, "A nonlocal-means approach to exemplar-based inpainting," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.* IEEE, 2008, pp. 2600–2603.

[36] M. Protter, M. Elad, H. Takeda, and P. Milanfar, “Generalizing the nonlocal-means to super-resolution reconstruction,” *Image Processing, IEEE Transactions on*, vol. 18, no. 1, pp. 36–51, Jan 2009.

[37] W. Zuo, L. Zhang, C. Song, and D. Zhang, “Texture enhanced image denoising via gradient histogram preservation,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1203–1210.

[38] O. K. Al-Shaykh and R. M. Mersereau, “Lossy compression of noisy images,” *Image Processing, IEEE Transactions on*, vol. 7, no. 12, pp. 1641–1652, 1998.

[39] B. T. Oh, Y. Su, C. Segall, and C.-C. Kuo, “Synthesis-based texture video coding with side information,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 647–659, May 2011.

[40] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on.* IEEE, 2007, pp. 1–8.

[41] J. Zhang and S. Sclaroff, “Saliency detection: a boolean map approach,” in *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2013, pp. 153–160.

[42] M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards, and Practice.* springer, 2002, vol. 1.