Multiplicative Noise Removal Based on Unbiased Box-Cox Transformation

Yu-Mei Huang\textsuperscript{1}, Hui-Yin Yan\textsuperscript{1} and Tieyong Zeng\textsuperscript{2,3,}\textsuperscript{*}

\textsuperscript{1} School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China.
\textsuperscript{2} Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.
\textsuperscript{3} HKBU Institute of Research and Continuing Education, Shenzhen Virtual University Park, Shenzhen 518057, China.

Received 23 May 2016; Accepted (in revised version) 15 February 2017

Abstract. Multiplicative noise removal is a challenging problem in image restoration. In this paper, by applying Box-Cox transformation, we convert the multiplicative noise removal problem into the additive noise removal problem and the block matching three dimensional (BM3D) method is applied to get the final recovered image. Indeed, BM3D is an effective method to remove additive Gaussian white noise in images. A maximum likelihood method is designed to determine the parameter in the Box-Cox transformation. We also present the unbiased inverse transform for the Box-Cox transformation which is important. Both theoretical analysis and experimental results illustrate clearly that the proposed method can remove multiplicative noise very well especially when multiplicative noise is heavy. The proposed method is superior to the existing methods for multiplicative noise removal in the literature.

AMS subject classifications: 62E17, 62F10, 65H05, 68U10

Key words: Multiplicative noise removal, Box-Cox transformation, maximum likelihood estimation, block-matching three dimensional method (BM3D).

1 Introduction

In real image applications, most images are generated through image recording systems. During the digital image acquisition, images are commonly affected by noise. For instance, additive Gaussian white noise is the most frequent noise in image systems. In the past decades, various methods have been considered to reduce additive Gaussian white noise in images. For instance, besides the total variation-based methods [13, 15, 41, 46],

*Corresponding author. Email addresses: zeng@hkbu.edu.hk (T. Zeng), huangym@lzu.edu.cn (Y.-M. Huang), yanhy12@lzu.edu.cn (H.-Y. Yan)

there are wavelet-based methods [23], fourth-order method [33], nonlocal-based methods [9, 10], sparse representation methods such as dictionary-based methods [25], [31] and the block matching three dimensional (BM3D) method [19], of which BM3D method is very efficient to remove additive Gaussian white noise. Another classical degradation to real images is impulse noise which includes salt-and-pepper noise and random-valued impulse noise. For salt-and-pepper noise, pixel of the damaged image takes the smallest or biggest pixel value (e.g. 0 or 255 for an 8-bit image), and other pixel value keeps; for random-valued impulse noise, pixel of the damaged image takes random value, and other pixel remains. In literature, many approaches have been also proposed to impulse noise removal, see, for instance, [5, 11, 12, 28, 35, 47]. Moreover, Poisson noise often arises in radiographic imaging applications, and many efficient methods are devoted to removing this kind of noise, such as the methods proposed in [8, 17, 30, 34, 48].

In recent years, multiplicative noise removal problem has attracted much attention since it arises in many practical applications such as laser, microscope, and synthetic aperture radar (SAR) images. When monochromatic radiation is scattered from a surface, whose roughness is of the order of a wavelength, it causes wave interference which results in speckle or multiplicative noise in an image. Different from additive noises, a recorded image \( g \) contaminated by multiplicative noise is the multiplication of the original image \( u \) and a noise \( v \) pixel by pixel

\[
g = uv. \tag{1.1}
\]

Here \( u, g, \) and \( v \) are \( mn \)-by-1 vectors corresponding to the original clean \( m \)-by-\( n \) image, observed \( m \)-by-\( n \) image and \( m \)-by-\( n \) noise, respectively. Because of the special degraded mechanism, almost all information of the original image may disappear when it is distorted by multiplicative noise and the degraded image is very difficult to be recovered. Many methods have been proposed for multiplicative noise removal in the literature. For example, in [40], Rudin, Lions and Osher first considered the multiplicative noise image restoration and constructed constrained minimization methods for Gaussian multiplicative noise removal. In 2008, by utilizing Maximum A Posterior analysis, Aubert and Aujol proposed a variational method for multiplicative noise removal where the noise follows Gamma distribution [4]. In the same year, Shi and Osher applied the inverse scale space method to multiplicative noise degraded models by converting multiplicative noise into additive noise removal problem, see, for instance, [42]. Huang, Ng and Wen proposed a global convex model to remove multiplicative noise in logarithm domain in [27]. In recent years, some other efficient methods also have been proposed. Indeed, Chesneau, Fadili and Starck proposed a stein block thresholding method in [18]. An efficient hybrid approach was proposed by Durand, Fadili and Nikolova in [24]. Steidl and Teuber considered using the I-divergence as a data fitting term to reduce multiplicative Gamma noise in [43]. In [6], Bioucas-Dias and Figueiredothe handled the multiplicative noise removal problem by using variable splitting and constrained optimization. In addition, Teuber and Lang proposed efficient nonlocal filters for multiplicative noise removal in [44] and [45]. Huang et al. proposed a dictionary method for multiplicative noise
removal in [29]. A data driven filtering approach was proposed by Deledalle, Denis, and Tupin in [20]. Dong and Zeng utilized a quadratic penalty function technique and I-divergence term to obtain a strictly convex model that guarantees the uniqueness of the solution and the stability of the algorithm in [21, 22]. In [49], a new convex variational model for multiplicative noise and blur removal was proposed.

Because of the extensive research for removing additive noise and the excellent effects of the proposed methods in the literature, we are interested in finding better methods for multiplicative noise removal based on some effective new additive noise removal methods. In [37, 38], by applying Anscombe root transformation, the observed images corrupted by Poisson noise are stabilized into a new signal in which the noise can be treated as additive Gaussian noise with unitary variance. Then the noise is removed by using a conventional algorithm for additive Gaussian noise and an exact unbiased inverse transformation is performed to get the restored images. We know Box-Cox transformation [7] can effectively transform a random variable and force it to follow normal distribution exactly or approximately if a suitable transformation parameter is selected and BM3D is a rather novel method for additive Gaussian white noise removal proposed by Dabov et al. in [19]. Therefore, inspired by the work proposed in [37, 38], in this paper, we transform the multiplicative noise removal to additive Gaussian noise removal by applying Box-Cox transformation and the images are finally recovered by an unbiased denoising algorithm. The Box-Cox transformation parameter is determined through a maximum likelihood method. After Box-Cox transformation to the observed images, BM3D method is utilized to restore the transformed image and an unbiased improvement is performed so that the recovered image can finally be obtained.

The contribution of our paper is clear. Indeed, we propose an unbiased Box-Cox transformation with some theoretical study to handle the task of multiplicative noise removal which leads to the leading performance; Base on previous works, we propose a rather effective scheme to estimate the noise level of the multiplicative noise which is largely ignored in the literature; Our framework is totally parameter free which is usual for multiplicative noise. Moreover, the proposed method can be easily extended to other type of non-Gaussian noises, which should have good impact along this direction.

We organize the paper as follows. In Section 2, the Box-Cox theory and BM3D method are introduced. In Section 3, we propose Box-Cox transformation method to remove multiplicative noises in images. In Section 4, we design a maximum likelihood method to determine Box-Cox transformation parameters in multiplicative noise image restoration. The unbiased inverse transform is then given. In Section 5, some numerical results are given to demonstrate the noise removal effectiveness of the proposed Unbiased Box-Cox transformation method. Finally, some concluding remarks are given in Section 6.

2 Box-Cox transformation and BM3D method

Box-Cox transformation is a kind of data transformation in statistical modeling and it is often used to transform a random variable to meet normal distribution. BM3D method is
a novel strategy based on an enhanced sparse representation for additive white noise removal. In this section, we briefly sketch the Box-Cox transformation and BM3D method.

2.1 Box-Cox transformation

In statistical analysis, many important results follow from the assumption that the population being sampled or investigated is normally distributed with a common variance and additive error structure. However, lots of data of a sample that we deal with in numerous applications do not conform to such assumption. Hence, data transformation tools are often used to improve normality of a distribution and equalize variance to meet assumptions and improve effect sizes so that important aspects of data cleaning and preparing can be constituted. The commonly discussed traditional transformations include: adding constants, square root, power transformation, converting to logarithmic scales, inverting and reflecting, applying trigonometric transformations such as sine wave transformations, etc. Of which the most practical and important transformation is the Box-Cox transformation proposed by Box and Cox in 1964, for details, see [7]. It is a family of power transformations that incorporates and extends the traditional options to help researchers easily find the optimal normalizing transformation for each variable. Box-Cox transformation has many different forms. One specific form is

\[
y^{(\lambda)} = \begin{cases} 
\frac{y^{\lambda}-1}{\lambda} & (\lambda \neq 0), \\
\log y & (\lambda = 0).
\end{cases}
\]

Indeed, (2.1) works with a parametric family of transformations from \( y \) to \( y^{(\lambda)} \), a particular parameter \( \lambda \) defines a particular transformation.

There are three key issues related to the application of Box-Cox transformation (2.1). First, we need to determine a good parameter \( \lambda \) because the proximity to normal distribution of the transformed data \( y^{(\lambda)} \) greatly depends on good selection of \( \lambda \). In [7], Box and Cox gave two methods for estimating the transformation parameters. One is to find the point estimates and confidence intervals based on maximum-likelihood theory; the other one is to estimate a suitable \( \lambda \) by constructing a posterior distribution for transformation parameter \( \lambda \) via Bayes’ theorem. Some other methods to estimate the Box-Cox transformation parameter for different practical applications can also be found in [2,3,39].

We note that the Box-Cox power transformation can not guarantee exact normality, because the transformation parameter is impossible to be calculated exactly in real applications as well as it is often obtained by data estimation, prediction and inference. Usually all Box-Cox transformations are with the assumption that \( \lambda \) value is between \(-5 \) and \( 5 \), see, for instance, [1]. Moreover, the normality is actually difficult to be checked in real applications. Second, one need carefully select a suitable denoising method for the nearly Gaussian noise case. For this, we use the leading BM3D method and it can be readily extended. Third, it is important to find a suitable unbiased inverse transform for the Box-Cox transformation. We discuss this issue in the coming Section.
2.2 BM3D method

BM3D method is an efficient method proposed by Dabov, Katkovnik and Egiazarian for additive Gaussian noise removal, for details, see [19]. This method restores the corrupted image based on image nonlocal self-similarity prior. For each extracted reference block from the corrupted image, the similar blocks to it in the corrupted image are found and then stacked to form a 3D array and which is called a group. Then, a collaborative filtering which consists of three successive steps is performed on the group: 3D transformation of the group, shrinkage of transform spectrum and inverse 3D transformation. After that, a 3D estimate of jointly grouped image blocks is obtained. In addition, the filtered blocks are returned to their original positions. The obtained block estimates can overlap and thus there are multiple estimates for each pixel. These estimates are aggregated to form an estimate of the whole image by taking advantage of the redundancy. Since the collaborative filtering can keep even the finest details in the group blocks and it also preserves the essential unique features of each block, the method can restore contaminated images very well. The experimental results demonstrate that BM3D method achieves state-of-art denoising performance.

3 Multiplicative noise removal by unbiased Box-Cox transformation

Considering multiplicative noise removal problem (1.1), we need to find an estimate of the original image $u$ from the observed image $g$. In this paper, we focus on multiplicative Gamma noise $v$ that follows the Gamma distribution of mean one with its probability density function given by

$$p(v; L) = \begin{cases} \frac{L^L v^{L-1}}{\Gamma(L)} e^{-Lv}, & v > 0, \\ 0, & v \leq 0 \end{cases}$$

(3.1)

where $L$ is an integer and $\Gamma(\cdot)$ is the classical Gamma function, see for instance [4, 27, 29]. Moreover, the variance of $v$ is given by $1/L$ and the noise level is controlled by its variance. Therefore, smaller the number $L$ is, more serious the noise level will be. Suppose that $u$ is fixed, (1.1) shows that each pixel of image $u$ is multiplied by random variable $v$, then each pixel of the observed image $g$ correspondingly can be considered as a new random variable. Applying Box-Cox transformation with parameter $\lambda$ to each pixel variable of $g = uv$, we get

$$g^{(\lambda)} = \frac{(uv)^\lambda - 1}{\lambda}.$$  

(3.2)

Suppose that $u$ and $v$ are independent, the expectation of $g^{(\lambda)}$ follows as

$$\mathbb{E}(g^{(\lambda)}) = \mathbb{E}\left(\frac{(uv)^\lambda - 1}{\lambda}\right) = \frac{\Gamma(L+\lambda)}{\lambda L^\lambda \Gamma(L)} u^L - \frac{1}{\lambda}$$

(3.3)
If $\lambda$ is selected appropriately, $g^{(\lambda)}$ should follow or be close to Gaussian distribution and it can be expressed as

$$g^{(\lambda)} = \frac{\Gamma(L+\lambda)}{\lambda L^{\lambda} \Gamma(L)} u^\lambda - \frac{1}{\lambda} + \epsilon,$$

where the random variable $\epsilon \sim N(0, \sigma^2)$ based on the assumptions in the Box-Cox transformation. In (3.3), if we consider $\frac{\Gamma(L+\lambda)}{\lambda L^{\lambda} \Gamma(L)} u^\lambda - \frac{1}{\lambda}$ as the original image and $g^{(\lambda)}$ as the observed image, the additive noise removal methods can be applied to (3.3) and a denoised approximation $w$ of $\frac{\Gamma(L+\lambda)}{\lambda L^{\lambda} \Gamma(L)} u^\lambda - \frac{1}{\lambda}$ can be gotten and hence the recovered image

$$\tilde{u} = L \left( \frac{\Gamma(L)(\lambda w+1)}{\Gamma(L+\lambda)} \right)^{\frac{1}{\lambda}}$$

finally can be obtained.

In [37, 38], the authors proposed variance stabilizing Anscombe transformation for Poisson noise or Poisson-Gaussian noise removal. The unbiased inverse of Anscombe transformation was analyzed carefully there. As Box-Cox transformation with proper parameter $\lambda$ is another kind of variance stabilizing method, (3.4) is also an unbiased inverse of Box-Cox transformation. Indeed, following the unbiased inverse idea in [37], let function $f : g \rightarrow g_1^\lambda - \frac{1}{\lambda}$ denote Box-Cox transformation with parameter $\lambda$, then $u$ can be determined from an unbiased inverse Box-Cox transformation that maps $E\{f(g)|u\}$ to $E\{g|u\}$, where

$$E\{f(g)|u\} = \frac{1}{u} \int_0^{+\infty} \frac{(g^\lambda - 1) L_\lambda^{L-1} e^{-L \frac{g}{u}}}{\lambda \Gamma(L)} dg = \frac{\Gamma(L+\lambda)}{\lambda L^{\lambda} \Gamma(L)} u^\lambda - \frac{1}{\lambda}.$$

The same formula (3.4) can be obtained from $w = E\{f(g)|u\}$. Hence, (3.4) is the unbiased inverse of Box-Cox transformation. Box-Cox transformation is useful for both low-count regime and high-count regime, especially for low-count regime which can be demonstrated in numerical results section.

4 Determination of the Box-Cox parameter in multiplicative noise removal

In this section, we determine the Box-Cox transformation parameter in (3.2) for our actual multiplicative noise removal processing. We denote

$$a(\lambda) = \frac{\Gamma(L+\lambda)}{L^{\lambda} \Gamma(L)}.$$
Actually, $a(\lambda)$ is the expectation of the random variable $v^\lambda$, here $v$ specifically refers to the Gamma noise in (1.1). Suppose that we have value of $\lambda$ such that

$$
\epsilon = g^{(\lambda)} - \mathbb{E}(g^{(\lambda)}) = \frac{g^\lambda - a(\lambda)u^\lambda}{\lambda} \sim N(0, \sigma^2),
$$

where $g$ and $u$ are the observed image and clean image in (1.1) and $g^{(\lambda)}$ is the transformation result of $g$ in (3.2). Then the likelihood with respect to the observation $g$ is

$$
L_{\text{max}}(\lambda, \sigma^2) = (2\pi\sigma^2)^{-\frac{mn}{2}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^{mn} e_i^2 \right\} \prod_{i=1}^{mn} \frac{\partial \epsilon_i}{\partial g_i}
$$

$$
= (2\pi\sigma^2)^{-\frac{mn}{2}} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^{mn} \left(\frac{g_i^\lambda - a(\lambda)u_i^\lambda}{\lambda}\right)^2 \right\} \prod_{i=1}^{mn} g_i^{\lambda-1}.
$$

So the maximized log likelihood is

$$
\log L_{\text{max}} = -\frac{1}{2\sigma^2} \sum_{i=1}^{mn} \left(\frac{g_i^\lambda - a(\lambda)u_i^\lambda}{\lambda}\right)^2 - \frac{mn}{2} \log(2\pi)
$$

$$
- \frac{mn}{2} \log(\sigma^2) + (\lambda - 1) \sum_{i=1}^{mn} \log g_i.
$$

We can easily get the partial derivatives of $\log L_{\text{max}}$ with respect to $\sigma^2$ and $\lambda$, given by

$$
\frac{\partial \log L_{\text{max}}}{\partial \sigma^2} = -\frac{mn}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{mn} \left(\frac{g_i^\lambda - a(\lambda)u_i^\lambda}{\lambda}\right)^2,
$$

$$
\frac{\partial \log L_{\text{max}}}{\partial \lambda} = \sum_{i=1}^{mn} \log g_i - \frac{1}{\lambda^2\sigma^2} \sum_{i=1}^{mn} \left(\frac{g_i^\lambda - a(\lambda)u_i^\lambda}{\lambda}\right)^2
$$

$$
+ \lambda \left(\frac{g_i^\lambda \log g_i - a(\lambda)u_i^\lambda \log u_i - a(\lambda)u_i^\lambda \cdot (g_i^\lambda - a(\lambda)u_i^\lambda)}{\log L}\right),
$$

where

$$
a(\lambda)' = \frac{L^\lambda \Gamma(L+\lambda) - \Gamma(L+\lambda) \log L}{L^{2\lambda}},
$$

is the derivative of $a(\lambda)$ and

$$
\Gamma(L+\lambda)' = \int_0^\infty x^{L+\lambda-1} e^{-x} \log x dx.
$$

The parameters $\lambda$ and $\sigma^2$ can be obtained by solving the following system of equations

$$
\begin{cases}
\frac{\partial \log L_{\text{max}}}{\partial \sigma^2} = 0, \\
\frac{\partial \log L_{\text{max}}}{\partial \lambda} = 0.
\end{cases}
$$
By direct deduction from (4.2), we obtain that the relationship between \( \sigma^2 \) and \( \lambda \) is

\[
\sigma^2 = \frac{1}{mn} \sum_{i=1}^{mn} \left( \frac{g_i^\lambda - a(\lambda)u_i^\lambda}{\lambda} \right)^2.
\] (4.3)

By bringing (4.3) into (4.2), the equation with respect to variable \( \lambda \) can be easily expressed as

\[
-\frac{1}{\lambda^3 \sigma^2} \sum_{i=1}^{mn} \lambda (g_i^\lambda \log g_i^\lambda - a(\lambda)u_i^\lambda \log u_i^\lambda - a(\lambda)'u_i^\lambda \log u_i^\lambda) \cdot (g_i^\lambda - a(\lambda)u_i^\lambda) - (g_i^\lambda - a(\lambda)u_i^\lambda)^2 + \sum_{i=1}^{mn} \log g_i = 0.
\] (4.4)

Theoretically, \( \lambda \) can be solved exactly from this equation using Newton method with a proper initial guess. But this equation is too complicated and too nonlinear to calculate its derivatives required in Newton method. In practice, by bringing (4.3) into Eq. (4.1), we get the log likelihood function \( \log L_{\text{max}} \) with respect to the variable \( \lambda \). The function \( \log L_{\text{max}} \) is also complicated and it may have several maximal points for \( \lambda > 0 \). So we estimate the maximum point of \( \log L_{\text{max}} \) numerically. That is, we plot the curve of \( \log L_{\text{max}} \) in a certain range of \( \lambda \) by using such \( \log L_{\text{max}} \) values at equidistant nodes of \( \lambda \). In our multiplicative noise removal problem, we find when \( \lambda \) is negative or greater than 1, the multiplicative noise removal effect is not good. Therefore, in all our experiments, we try to find the suitable maximum point of \( \log L_{\text{max}} \) in the range of \([0,1]\). Here we note, in both (4.1) and (4.4), \( u \) is the unknown original image and the noise level \( L \) is unknown. In practical implementations, an estimated recovered image is used instead of the original image \( u \) and an estimated \( L \) with high precision replaces true \( L \) in \( \log L_{\text{max}} \). Indeed, taking logarithm transformation on both sides of (1.1), we get \( \log g = \log u + \log v \) \((u > 0, v > 0)\). We consider \( \log g \) as the observed image, \( \log u \) the original image and \( \log v \) the additive noise. Then the multiplicative noise removal is converted into an additive noise removal problem. Tight frame method proposed in [16] is applied to remove \( \log v \) and we obtain a recovered image \( \hat{u} \). Though \( \log v \) doesn’t follow Gaussian distribution, our experimental results show that \( \hat{u} \) is enough to make the maximum point of \( \log L_{\text{max}} \) be a good value of \( \lambda \). We remark that if \( u \) in \( \log L_{\text{max}} \) is replaced by a recovered image obtained through any other existing multiplicative noise removal method, we can also get a good Box-Cox parameter and finally obtain the good recovered images. For Log-Gamma distribution [14], we know the variance of \( \log v \) should \( \psi'(L) \), where \( \psi'(\cdot) \) is the first derivative of Digamma function. We apply the method proposed in [32] to estimate the variance of \( \log v \), that is, \( \log v \) is coarsely considered being Gaussian noise. By using Newton method, an estimate of \( L \) can be calculated. The resulted \( L \) is also used in (3.4) for final recovered image calculation.

In summary, the proposed multiplicative noise removal method is a two-step method. The first step is to estimate Box-Cox transformation parameter and then apply it in Box-Cox transformation to transform the observed image. The second step is to apply BM3D
method to denoise the transformed image obtained in the first step and then do inverse Box-Cox transformation to get the recovered image. In the first step, initial denoised image guess is taken as the denoised image using tight frame method proposed in [16]. For multiplicative noise level $L$, by using the method proposed in [14] and [32], we have $\psi'(L) = \eta^2$. Where $\eta^2$ denotes the variance of $\log v$ which is estimated using the method proposed in [32]. Then an estimate of $L$ is obtained through Newton iterations $L_{k+1} = L_k - \frac{\psi'(L_k) - \eta^2}{\psi''(L_k)}$ for some initial value $L_0$, the final integer $L$ is got as the roundoff of $L_k$. We set $L_0 = 4$ in our experiments. Where $\psi''(\cdot)$ is the second derivative of Digamma function. We find that the roundoff integer $L$ is the same as the true value of $L$ in our experiments. After having the initial denoised image guess $\tilde{u}$ and the estimated noise level $L$, the optimal $\lambda$ is searched in the interval $[0,1]$ with certain fixed stepsize. The stepsize is set as 0.001 in our experiments.

5 Numerical results

In this section, we implement some experiments to demonstrate the efficiency of the proposed method to remove multiplicative noise especially when the noise level is high. Comparisons are also made with some efficient methods that have been proposed recently for multiplicative noise removal. These methods are: the “AA” method proposed by Aubert and Aujol in [4]; the “BS” method proposed by Chesneau, Fadili and Starck in [18]; the “DZ” method proposed by Dong and Zeng in [21]; the “DFN” method proposed by Durand, Fadili and Nikolova in [24]; the updating nonlocal method “NL” proposed by Teuber and Lang in [44]; the Dictionary method “HMNZ” proposed by Huang, Moisan, Ng and Zeng in [29]. The recovered results are assessed in terms of visual effects and quantities PSNR, MAE and computing times. For a mn-by-1 noise-free image $u$ and its denoised version $\tilde{u}$, the peak signal-to-noise ratio (PSNR, in decibels) is defined by

$$\text{PSNR} = 10\log_{10} \frac{mn \cdot |\max(u) - \min(u)|^2}{\|\tilde{u} - u\|_2^2} (\text{dB}),$$

where $|\max(u) - \min(u)|$ measures the gray-scale range of the original image $u$. The Mean Absolute-deviation Error (MAE) is defined by

$$\text{MAE} = \frac{1}{mn} \|\tilde{u} - u\|_1.$$

We adopt images Cameraman, Peppers with size 256-by-256 and Barbara, Nimes (a French city) with size 512-by-512 in our experiments. Each image is contaminated by multiplicative Gamma noise with three different levels $L = 10$, $L = 4$, $L = 1$ in (3.1), respectively. We know that smaller the number $L$ is, more serious the noise level will be.
5.1 Estimation of Box-Cox transformation parameters

Firstly, we determine the Box-Cox transformation parameter $\lambda$ in (2.1) using the method proposed in Section 4. In our experiments, we find that $\log L_{\text{max}}$ is negative and usually its absolute value is large. In order to determine the maximum point of $\log L_{\text{max}}$ clearly and easily, we plot the curve of $\log(-\log L_{\text{max}})$ for $\lambda$ in interval $[0,1]$ and afterwards try to find the minimum point of function $\log(-\log L_{\text{max}})$ which is used as the optimal value of Box-Cox transformation parameter. In all experiments, we first get a recovered image by applying tight frame method in [16] and then the clean image $u$ in $\log(-\log L_{\text{max}})$ is replaced with this recovered image to estimate the transformation parameter. Let the variance of $\log v$ obtained using the method in [32] be $\text{var}(\log v)$, then the estimate of $L$ can be got by solving $\text{var}(\log v) = \psi'(L)$ utilizing Newton method.

Figs. 1-3 show the curves of $\log(-\log L_{\text{max}})$ with $\lambda \in [0,1]$ for different observed images when $L = 10$, $L = 4$, $L = 1$, respectively. We easily see $\log(-\log L_{\text{max}})$ is convex for $\lambda \in [0,1]$. In each curve, we can visually see that the lowest part of $\log(-\log L_{\text{max}})$ is located in a small flat range, and the minima can not be determined clearly by eyes. So we take advantage of the computed value of $\log(-\log L_{\text{max}})$ at denser equidistant nodes in this small lowest area to determine the closest minima of $\log(-\log L_{\text{max}})$ and show them in Table 1. From this table, we can see that $\lambda$ is larger when noise level is higher. Here we note that if we set $\lambda$ as another value which is located in range with the optimal $\lambda$ as the center and equidistance step as the radius in our experiments, the recovered results are almost the same, including recovered visual qualities and quantities. Therefore, the estimation procedure to find the optimal Box-Cox transformation parameter is reasonable and robust.

<table>
<thead>
<tr>
<th>$L$</th>
<th>Pictures</th>
<th>Cameraman</th>
<th>Pepper</th>
<th>Barbara</th>
<th>Nimes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>0.021</td>
<td>0.027</td>
<td>0.061</td>
<td>0.012</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.054</td>
<td>0.070</td>
<td>0.111</td>
<td>0.084</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0.128</td>
<td>0.154</td>
<td>0.182</td>
<td>0.182</td>
</tr>
</tbody>
</table>

5.2 Image restoration experiments

We bring the estimated Box-Cox transformation parameter got from the above section into (2.1) for observed image $g$ and the recovered images are finally obtained through (3.4) based on (3.3) by using BM3D method. All experimental results are shown in Figs. 5-16. Fig. 4 shows the original “Cameraman”, “Peppers”, “Barbara”, “Nimes” images, respectively. Subplots (a) in each figure shows the noisy image in each experiment, respectively. And subplots (b), (c), (d), (e), (f), (g), (h) show the recovered images by using
Figure 1: Curves of $\log(-\log L_{\text{max}})$ for images (a) Cameraman; (b) Peppers; (c) Barbara; (d) Nimes when noise level $L = 10$.

Figure 2: Curves of $\log(-\log L_{\text{max}})$ for images (a) Cameraman; (b) Peppers; (c) Barbara; (d) Nimes when noise level $L = 4$.

Figure 3: Curves of $\log(-\log L_{\text{max}})$ for images (a) Cameraman; (b) Peppers; (c) Barbara; (d) Nimes when noise level $L = 1$.

"AA", "BS", "DZ", "DFN", "NL", "HMNZ" methods and our proposed Unbiased Box-Cox transformation method ("UBC"), respectively. Recovered PSNR values are shown in captions of each figure. Parameters in methods "AA", "BS", "DZ", "DFN", "NL" and "HMNZ" are selected to be those which make the recovered PSNRs and MAEs best in each experiment.

Tables 2-5 show the recovered PSNRs, MAEs and computing times in seconds (Time) of each experiment for the adopted four images using "AA", "BS", "DZ", "DFN", "NL", "HMNZ" methods and our proposed Box-Cox transformation method "UBC". We also show PSNR and MAE values of the observed noisy images in each experiment. "Noisy"
Table 2: The restored results corresponding to Figs. 5-7 for Cameraman image.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Recovered ($L = 10$)</th>
<th>Recovered ($L = 4$)</th>
<th>Recovered ($L = 1$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
</tr>
<tr>
<td>Noisy</td>
<td>15.65</td>
<td>29.39</td>
<td>11.69</td>
</tr>
<tr>
<td>BS</td>
<td>24.43</td>
<td>9.44</td>
<td>0.89</td>
</tr>
<tr>
<td>DZ</td>
<td>25.11</td>
<td>8.19</td>
<td>18.46</td>
</tr>
<tr>
<td>DFN</td>
<td>26.08</td>
<td>7.40</td>
<td>45.53</td>
</tr>
<tr>
<td>NL</td>
<td>26.55</td>
<td>6.82</td>
<td>1.57</td>
</tr>
<tr>
<td>HMNZ</td>
<td>27.32</td>
<td>6.63</td>
<td>23.52</td>
</tr>
<tr>
<td>UBC</td>
<td>27.91</td>
<td>6.19</td>
<td>11.91</td>
</tr>
</tbody>
</table>

From these figures and tables, we can see that “BS”, “NL” methods and the proposed method are among the fastest ones. Moreover, the results obtained from the proposed Box-Cox transformation method have the best recovered visual quality and the best PSNRs, MAEs. For example, in Figs. 5-7 of Cameraman restoration results, the constant blocks of the images are more flat and textures are more clear in (h) than the corresponding parts in (b)-(g) of each figure. Especially when $L = 1$, the PSNR of the recovered image obtained by proposed method is 2.06dB, 1.71dB, 2.28dB, 1.32dB, 1.51dB, 0.66dB greater than those obtained by “AA”, “BS”, “DZ”, “DFN”, “NL”, “HMNZ” methods, respectively. Recovered results of Peppers image in Figs. 8-10 have the same recovered effects as those for Cameraman image. And the recovered effect is most apparent for serious noise when $L = 1$ where both structures and textures are well recovered. Barbara image is usually used in image restoration to test the texture recovering effect of an algorithm. From Figs. 11-13, we can clearly see that the proposed Box-Cox transformation can effectively restore the textures in the Barbara image than the compared methods for multiplicative noise removal. Even when $L = 1$, most textures in tablecloth, scarf and legs can still be...
Table 3: The restored results corresponding to Figs. 8-10 for Peppers image.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Recovered ((L = 10))</th>
<th></th>
<th></th>
<th>Recovered ((L = 4))</th>
<th></th>
<th></th>
<th>Recovered ((L = 1))</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
</tr>
<tr>
<td>Noisy</td>
<td>15.66</td>
<td>30.50</td>
<td></td>
<td>11.68</td>
<td>47.82</td>
<td></td>
<td>5.58</td>
<td>90.69</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>24.40</td>
<td>10.91</td>
<td>18.44</td>
<td>22.68</td>
<td>13.67</td>
<td>18.35</td>
<td>18.93</td>
<td>21.82</td>
<td>18.22</td>
</tr>
<tr>
<td>BS</td>
<td>25.09</td>
<td>9.90</td>
<td>0.86</td>
<td>22.82</td>
<td>13.08</td>
<td>0.75</td>
<td>19.19</td>
<td>20.40</td>
<td>0.75</td>
</tr>
<tr>
<td>DZ</td>
<td>26.19</td>
<td>8.27</td>
<td>17.04</td>
<td>23.43</td>
<td>11.65</td>
<td>16.78</td>
<td>17.91</td>
<td>23.13</td>
<td>16.91</td>
</tr>
<tr>
<td>DFN</td>
<td>26.40</td>
<td>8.49</td>
<td>34.99</td>
<td>23.81</td>
<td>11.52</td>
<td>38.41</td>
<td>19.42</td>
<td>19.33</td>
<td>37.45</td>
</tr>
<tr>
<td>NL</td>
<td>27.07</td>
<td>7.62</td>
<td>1.24</td>
<td>24.32</td>
<td>10.15</td>
<td>1.18</td>
<td>19.90</td>
<td>17.74</td>
<td>1.10</td>
</tr>
<tr>
<td>HMNZ</td>
<td>27.30</td>
<td>7.57</td>
<td>24.73</td>
<td>24.78</td>
<td>10.14</td>
<td>22.84</td>
<td>20.89</td>
<td>16.72</td>
<td>19.33</td>
</tr>
<tr>
<td>UBC</td>
<td>27.97</td>
<td>6.95</td>
<td>11.71</td>
<td>25.31</td>
<td>9.53</td>
<td>11.97</td>
<td>21.28</td>
<td>15.50</td>
<td>11.59</td>
</tr>
</tbody>
</table>

Table 4: The restored results corresponding to Figs. 11-13 for Barbara image.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Recovered ((L = 10))</th>
<th></th>
<th></th>
<th>Recovered ((L = 4))</th>
<th></th>
<th></th>
<th>Recovered ((L = 1))</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
</tr>
<tr>
<td>Noisy</td>
<td>15.90</td>
<td>29.32</td>
<td></td>
<td>11.91</td>
<td>45.78</td>
<td></td>
<td>5.91</td>
<td>86.19</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>23.10</td>
<td>12.35</td>
<td>81.68</td>
<td>20.49</td>
<td>17.88</td>
<td>81.62</td>
<td>17.96</td>
<td>24.73</td>
<td>81.38</td>
</tr>
<tr>
<td>BS</td>
<td>24.60</td>
<td>10.39</td>
<td>3.19</td>
<td>22.35</td>
<td>13.34</td>
<td>3.04</td>
<td>19.91</td>
<td>18.45</td>
<td>2.86</td>
</tr>
<tr>
<td>DZ</td>
<td>22.87</td>
<td>11.69</td>
<td>60.55</td>
<td>21.63</td>
<td>14.14</td>
<td>62.81</td>
<td>17.53</td>
<td>18.84</td>
<td>61.20</td>
</tr>
<tr>
<td>DFN</td>
<td>25.46</td>
<td>9.45</td>
<td>180.84</td>
<td>23.46</td>
<td>11.99</td>
<td>180.99</td>
<td>20.34</td>
<td>17.87</td>
<td>179.57</td>
</tr>
<tr>
<td>NL</td>
<td>26.09</td>
<td>9.17</td>
<td>9.05</td>
<td>22.95</td>
<td>11.91</td>
<td>10.47</td>
<td>20.18</td>
<td>17.86</td>
<td>9.39</td>
</tr>
<tr>
<td>HMNZ</td>
<td>26.47</td>
<td>8.43</td>
<td>109.45</td>
<td>23.63</td>
<td>11.62</td>
<td>108.34</td>
<td>20.31</td>
<td>17.82</td>
<td>107.93</td>
</tr>
<tr>
<td>UBC</td>
<td>28.24</td>
<td>6.98</td>
<td>49.77</td>
<td>25.64</td>
<td>9.41</td>
<td>49.59</td>
<td>21.44</td>
<td>15.60</td>
<td>49.79</td>
</tr>
</tbody>
</table>

recovered by the proposed method. For Nimes image, the recovered results especially for \(L = 1\) are still the best in Figs. 14-16 by the proposed Box-Cox transformation method. Heavy noise removal is a difficult task in image restoration. The experimental results tell us that the proposed Box-Cox transformation method can remove serious multiplicative noise better than the existing methods.

We give numerical results of image Barbara in Figs. 17-19 from model \(\log g = \log u + \log v\) by using BM3D in log domain and the proposed method, respectively. The figures clearly show that the proposed Box-Cox transformation method has a great improvement comparing with the BM3D in log domain method. Therefore, the proposed Box-Cox transformation converts multiplicative noise removal into Gaussian additive noise removal effectively.

During the Box-Cox transformation parameter estimation step, tight frame method proposed in [16] is applied to obtain an initial denoised image guess and which is used in the computation of \(\log (−\log L_{\text{max}})\). We perform some experiments for Barbara im-
Figure 5: (a) The noisy image with noise level $L=10$; The recovered images by methods (b) “AA” (24.48); (c) “BS” (24.43); (d) “DZ” (25.11); (e) “DFN” (26.08); (f) “NL” (26.55); (g) “HMNZ” (27.32); (h) proposed “UBC” (27.91).

Figure 6: (a) The noisy Cameraman image with noise level $L=4$; The recovered images by methods (b) “AA” (21.97); (c) “BS” (22.36); (d) “DZ” (22.72); (e) “DFN” (22.98); (f) “NL” (24.20); (g) “HMNZ” (24.91); (h) proposed “UBC” (25.67).
Figure 7: (a) The noisy Cameraman image with noise level $L = 1$; The recovered images by methods (b) “AA” (19.07); (c) “BS” (19.42); (d) “DZ” (18.85); (e) “DFN” (19.81); (f) “NL” (19.62); (g) “HMNZ” (20.47); (h) proposed “UBC” (21.13).

Figure 8: (a) The noised Peppers image with noise level $L = 10$; The recovered images by methods (b) “AA” (24.40); (c) “BS” (25.09); (d) “DZ” (26.19); (e) “DFN” (26.40); (f) “NL” (27.07); (g) “HMNZ” (27.30); (h) proposed “UBC” (27.97).
Figure 9: (a) The noisy Peppers image with noise level $L = 4$; The recovered images by methods (b) “AA” (22.68); (c) “BS” (22.82); (d) “DZ” (23.43); (e) “DFN” (23.81); (f) “NL” (24.32); (g) “HMNZ” (24.78); (h) proposed “UBC” (25.31).

Figure 10: (a) The noisy Peppers image with noise level $L = 1$; The recovered images by methods (b) “AA” (18.93); (c) “BS” (19.19); (d) “DZ” (17.91); (e) “DFN” (19.42); (f) “NL” (19.90); (g) “HMNZ” (20.89); (h) proposed “UBC” (21.28).
Figure 11: (a) The noisy Barbara image with noise level $L = 10$; The recovered images by methods (b) “AA” (23.10); (c) “BS” (24.60); (d) “DZ” (22.87); (e) “DFN” (25.46); (f) “NL” (26.09); (g) “HMNZ” (26.47); (h) proposed “UBC” (28.24).

Figure 12: (a) The noisy Barbara image with noise level $L = 4$; The recovered images by methods (b) “AA” (20.49); (c) “BS” (22.35); (d) “DZ” (21.63); (e) “DFN” (23.46); (f) “NL” (22.95); (g) “HMNZ” (23.63); (h) proposed “UBC” (25.64).
Figure 13: (a) The noisy Barbara image with noise level $L=1$; The recovered images by methods (b) “AA” (17.96); (c) “BS” (19.91); (d) “DZ” (17.53); (e) “DFN” (20.34); (f) “NL” (20.18); (g) “HMNZ” (20.31); (h) proposed “UBC” (21.44).

Table 5: The restored results corresponding to Figs. 14-16 for Nimes image.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Recovered($L=10$)</th>
<th></th>
<th></th>
<th>Recovered($L=4$)</th>
<th></th>
<th></th>
<th>Recovered($L=1$)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
<td>PSNR</td>
<td>MAE</td>
<td>TIME(s)</td>
</tr>
<tr>
<td>Noisy</td>
<td>22.46</td>
<td>13.67</td>
<td>18.49</td>
<td>21.33</td>
<td>40.06</td>
<td></td>
<td>22.29</td>
<td>13.38</td>
<td>80.91</td>
</tr>
<tr>
<td>AA</td>
<td>25.26</td>
<td>8.33</td>
<td>80.92</td>
<td>24.55</td>
<td>10.06</td>
<td>80.46</td>
<td>22.17</td>
<td>13.93</td>
<td>2.84</td>
</tr>
<tr>
<td>BS</td>
<td>27.00</td>
<td>7.69</td>
<td>2.97</td>
<td>24.92</td>
<td>9.87</td>
<td>2.84</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DZ</td>
<td>27.01</td>
<td>8.07</td>
<td>62.94</td>
<td>25.04</td>
<td>10.81</td>
<td>64.37</td>
<td>21.41</td>
<td>16.01</td>
<td>64.05</td>
</tr>
<tr>
<td>DFN</td>
<td>27.80</td>
<td>7.21</td>
<td>176.44</td>
<td>25.84</td>
<td>9.09</td>
<td>172.82</td>
<td>22.87</td>
<td>13.14</td>
<td>187</td>
</tr>
<tr>
<td>HMNZ</td>
<td>28.85</td>
<td>6.33</td>
<td>110.83</td>
<td>26.34</td>
<td>8.60</td>
<td>107.06</td>
<td>23.19</td>
<td>12.59</td>
<td>111.27</td>
</tr>
<tr>
<td>UBC</td>
<td>28.96</td>
<td>6.10</td>
<td>49.22</td>
<td>26.76</td>
<td>7.95</td>
<td>49.56</td>
<td>23.65</td>
<td>11.61</td>
<td>50.50</td>
</tr>
</tbody>
</table>

To demonstrate the effectiveness of such initials. In these experiments, we use the original clean image $u$ or the observed noisy image $g$ instead as the initial guess in the computation of $\log(-\log L_{\text{max}})$. When clean image $u$ is applied, for $L = 10, L = 4, L = 1$, the estimated parameters $\lambda$ are 0.073, 0.130, 0.209, respectively, and the corresponding recovered PSNRs are 28.28, 25.63, 22.11, respectively. Both the estimated $\lambda$ values and recovered PSNRs are close to the corresponding results when the proposed initial image guesses are applied and the differences between corresponding $\lambda$ are $-0.012$, $-0.019$, $-0.019$, $-0.019$, $-0.019$.
Figure 14: (a) The noisy Nimes image with noise level $L=10$; The recovered images by methods (b) “AA” (25.26); (c) “BS” (27.00); (d) “DZ” (27.01); (e) “DFN” (27.80); (f) “NL” (28.42); (g) “HMNZ” (28.85); (h) proposed “UBC” (28.96).

Figure 15: (a) The noisy Nimes image with noise level $L=4$; The recovered images by methods (b) “AA” (24.55); (c) “BS” (24.92); (d) “DZ” (25.04); (e) “DFN” (25.84); (f) “NL” (26.07); (g) “HMNZ” (26.34); (h) proposed “UBC” (26.76).
Figure 16: (a) The noisy Nimes image with noise level $L = 1$; The recovered images by methods (b) “AA” (22.29); (c) “BS” (22.17); (d) “DZ” (21.41); (e) “DFN” (22.87); (f) “NL” (23.07); (g) “HMNZ” (23.19); (h) proposed “UBC” (23.65).

PSNRs are 0.04, 0.01 and 0.67 for $L = 10, L = 4, L = 1$, respectively. However, when observed image $g$ is applied, for $L = 10, L = 4, L = 1$, the estimated parameters $\lambda$ are 0.963, 0.986, 0.997, respectively, and the recovered PSNRs are 24.92, 20.24, 12.38, respectively. Both the estimated $\lambda$ values and recovered PSNRs have big differences between the corresponding results when the proposed initial denoised image guesses are applied and the differences between corresponding $\lambda$ are $-0.902, -0.875, -0.815$, PSNRs are 3.32, 4.40 and 9.06 for $L = 10, L = 4, L = 1$, respectively. We also perform some tests for Barbara image when we set $\lambda = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0$, respectively. The corresponding recovered PSNRs are 24.73, 23.24, 22.39, 21.82, 21.35, 20.91, respectively, which is much lower than the recovered PSNR 25.64 obtained by the proposed method.

In addition, by using the computed $L$ and $\lambda$, we plot the histogram in Fig. 20 of $\epsilon$ in (3.3) for image Barbara recovery. The figure shows that $\epsilon$ is very close to the histogram of Gaussian distribution variable whether the proposed initial denoised image guess or the original image is applied. However, when the observed image is used as the initial image guess, the histograms are not close to the Gaussian distribution especially when the noise level is high.

We performed a denoising test on a real noisy Synthetic Aperture Radar (SAR) image shown in subplot (a) of Fig. 21. The recovered results by different methods are shown in other subplots of Fig. 21. Since the original clean image is unknown, we adopt the blind
image quality analyzer Natural Image Quality Evaluator (NIQE) constructed in [36] to judge the quality of the recovered images. NIQE is a quantity calculated in terms of the construction of a quality aware collection of statistical features based on a simple and suc-
cessful space domain natural scene statistic model, for details, see [36]. The smaller the NIQE is, the better recovering quality will be. The estimated noise level of the noisy SAR image and the computed Box-Cox transformation parameter $\lambda$ in the proposed method are $L = 40$ and $\lambda = 0.205$, respectively. The noise is not serious. Subplots (b)-(h) in Fig. 21 are the recovered images of (a) by methods “AA”, “BS”, “DZ”, “DFN”, “NL”, “HMMNZ” and the proposed “UBC”, respectively. The recovered NIQE are 7.02, 6.65, 6.47, 5.98, 5.94, 5.92 and 4.95, respectively. Both the visual effect and recovered NIQE by the proposed method are the best. In order to see the denoising effects clearly, we enlarge one same patch of images in Fig. 21 and show them in Fig. 22. We can see from Fig. 22 that the proposed method can remove noise effectively especially in image edges. Fig. 23 is the histogram of $\varepsilon$ in (3.3) for this real SAR image test and it shows that the transformed noise after Box-Cox transformation extremely obeys normal distribution.
Figure 21: (a) The original noisy SAR image (NIQE: 9.83); The recovered images by methods (b) “AA” (NIQE: 7.02); (c) “BS” (NIQE: 6.65); (d) “DZ” (NIQE: 6.47); (e) “DFN” (NIQE: 5.98); (f) “NL” (NIQE: 5.94); (g) “HMNZ” (NIQE: 5.92); (h) proposed “UBC” (NIQE: 4.95).

Figure 22: (a) One patch of the original noisy SAR image; The corresponding patches in the recovered images by methods (b) “AA”; (c) “BS”; (d) “DZ”; (e) “DFN”; (f) “NL”; (g) “HMNZ”; (h) proposed “UBC”.
6 Concluding remarks

In this paper, we have considered the multiplicative Gamma noise removal problem by taking advantage of the efficient additive noise removal method BM3D. The multiplicative Gamma noise problem is converted to the additive noise removal problem by applying the Box-Cox transformation. The optimal Box-Cox transformation parameter is estimated by the maximum likelihood estimation theory and we present an unbiased inverse transform for this Box-Cox transformation. The BM3D method is used for the transformed additive noise removal and we can readily extended to other leading approaches such as [26]. The numerical results demonstrate that the proposed unbiased Box-Cox transformation method can efficiently remove the multiplicative noise in different images degraded by multiplicative noise with different levels.

Acknowledgments

Research supported in part by NSFC Grant 11571156, NSFC 11271049, RGC grants 211911, 12302714 and HKBU FRGs.

References