Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

doi:10.3934/xx.xx.xx.xx

pp. **X–XX**

SPATIAL-FREQUENCY DOMAIN NONLOCAL TOTAL VARIATION FOR IMAGE DENOISING

Haijuan Hu*

Northeastern University at Qinhuangdao, School of Mathematics and Statistics
Hebei, 066004, China

Jacques Froment

Univ Bretagne-Sud, CNRS UMR 6205 LMBA, Campus de Tohannic
Vannes, F-56000 France

Baoyan Wang

Guangdong University of Petrochemical Technology, College of Computer Science
Guangdong, 525000, China

Xiequan Fan

Tianjin University, Center for Applied Mathematics
Tianjin, 300072, China

(Communicated by the associate editor name)

Abstract. Following the pioneering works of Rudin, Osher and Fatemi on total variation (TV) and of Buades, Coll and Morel on non-local means (NL-means), the last decade has seen a large number of denoising methods mixing these two approaches, starting with the nonlocal total variation (NLTV) model. The present article proposes an analysis of the NLTV model for image denoising as well as a number of improvements, the most important of which being to apply the denoising both in the space domain and in the Fourier domain, in order to exploit the complementarity of the representation of image data. A local version obtained by a regionwise implementation followed by an aggregation process, called Local Spatial-Frequency NLTV (L-SFNLTV) model, is finally proposed as a new reference algorithm for image denoising among the family of approaches mixing TV and NL operators. The experiments show the great performance of L-SFNLTV in terms of image quality and of computational speed, comparing with other recently proposed NLTV-related methods.

1. Introduction.

Gaussian noise removal is a fundamental and simple inverse problem of image processing, and it provides a good platform for theoretical research. Therefore it is still a hot topic in recent years, despite that there have been numerous works in the past 60 years or so. Among the numerous works, two examples of well-known algorithms are total variational model and non-local means. The total variation model [24] is firstly proposed by Rudin, Osher, and Fatemi (ROF), which introduces the total variation (TV) to replace the more traditional Euclidean norm. Since TV allows for discontinuity, the ROF model can recover

2010 *Mathematics Subject Classification.* Primary: 35Q68, 65T50, 68U10, 62H12.

Key words and phrases. Nonlocal total variation, Discrete Fourier transform, Image denoising, SURE, Regionwise implementation.

* Corresponding author: Haijuan Hu.

image edges and piecewise constant regions very well. But at the same time the recovered images look piecewise constant. Another shortage is the loss of texture due to the fact that total variation disfavors oscillations. Non-local means (NL-means) method is proposed by Buades, Coll, and Morel in [1], which exploits the fact that each patch in natural images generally has many similar patches. NL-means is a weighted average filter, where the weight between two pixels is determined by the Euclidean distance of two patches centered at the corresponding two pixels, which is more reliable than the weight determined by pixel distances.

In [7, 8], inspired by non-local means, a regularization functional called nonlocal total variation (NLTV) is proposed to generalize the total variation. In [19], the nonlocal total variation term is combined with a fidelity term in L^2 norm for image deconvolution and tomographic reconstruction. The applications of nonlocal operators include image denoising [13, 6, 7], image deblurring [13, 33, 19, 26], image inpainting [8, 34], motion estimation [29], etc. Some examples of recently proposed image denoising methods which improve the nonlocal total variation model significantly are [15, 17, 18]. In [15], the authors propose a regularizer which employs a novel non-local version of the structure tensor. In [17], the authors propose a regularization term on the weight function, combined with the nonlocal total variation term to model the image prior. In [18], a block-based nonlocal total variation is proposed to extend the original point-based model, and the weighting function can also be adaptively determined by the cost function itself.

In the past years, many other hybrid methods appeared based on the ideas of ROF model, NL-means or transform domain methods [4, 2]. For instance, the approaches [5, 3, 21, 9, 11] are based on the fact that similar patches in images can be sparsely represented or the matrix composed of similar patches has low rank; the paper [20] proposes a local total variation filter combining the idea of NL-means; the researches [14, 22] consider dual domain filters. Recent years have seen the emergence of a new paradigm for image denoising, where the modeling of natural images is left to computers: the deep learning framework [28, 30, 31, 32, 16]. This very promising domain will not be considered here and, on the contrary, we will make use of very old fashioned tools such as Fourier analysis.

To estimate the denoising performance without reference to original true image, an effective method is Stein's Unbiased Risk Estimation (SURE) [25], which provides an unbiased estimation of the true MSE for the denoised image. When the analytical expression is difficult to obtain, SURE can be estimated using Monte-Carlo method [23]. In [27], the authors give the explicit analytical expression for NL-means method and use it for parameter selection. However, we do not find the computation of SURE for NLTV in the literature.

In this paper, we will study the NLTV model in [19] for image denoising and improve it using Fourier transform. Note that the paper [7] also considers image denoising with NLTV model, but the weight is different from [19]. We first study NLTV: we establish that NLTV is essentially a neighborhood filter as ROF model in a simpler way than [20]; we find that NLTV is better than ROF and comparable to non-local means in most of cases by experiments. In addition, we derive SURE for NLTV, and show that it is better to choose the regularization parameter by it than randomly. Secondly, in order to benefit from the fact that the Fourier transform structures the image data in a totally different way from the standard basis, we consider frequency domain nonlocal total variation, called FNLTV. We empirically find that the regularization coefficient of FNLTV is not very sensitive and FNLTV is good at retaining image texture and details at the cost of leaving some noise evident in homogeneous regions. Then we propose a spatial-frequency domain nonlocal total variation model (SFNLTV) to combine NLTV in spatial domain and frequency domain. To fully utilize the frequency features, we finally take into account the local application of SFNLTV, abbreviated as L-SFNLTV. To understand well the roles played by the local application, we also consider the local versions of NLTV and FNLTV respectively. Experiments demonstrate that the local application of FNLTV greatly improves FNLTV, and local application of NLTV is similar to NLTV. Finally experiments show that L-SFNLTV is better than recently proposed NLTV related algorithms.

This paper is organized as follows: in Section 2, we introduce NLTV and SURE; in Section 3, FNLTV, SFNLTV and L-SFNLTV models are proposed; in Section 4, we show the relations of NLTV with neighborhood filters, discuss parameter choices, compute SURE for NLTV, and analyze the character of FNLTV and L-FNLTV model; simulations are provided in Section 5, and the conclusion is made in Section 6.

This paper is an extended and improved version of the paper [10]. The main new contributions are: the proposition of the local version of SFNLTV in Section 3.2, which greatly improves SFNLTV; the derivation of SURE for NLTV in Section 4.3 for the choice of the regularization parameter; the analyses of the characters of FNLTV and L-FNLTV models in Section 4.4 to better understand L-SFNLTV; the addition of more experimental results in Section 5.

2. **Preliminaries.**

As usual, a digital image is denoted by a $M \times N$ matrix $u = \{u(i) : i \in I\}$, where $I = \{0, 1, ..., M-1\} \times \{0, 1, ..., N-1\}$, and $0 \leq u(i) \leq 255$. The additive Gaussian noise model is:

$$v(i) = u_0(i) + \eta(i), \tag{1}$$

where u_0 and v are the original and noisy images respectively, and η is the Gaussian noise: $\eta(i)$ are independent and identically distributed Gaussian random variables with mean 0 and positive standard deviation σ. A denoised image is denoted as $\bar{v}$. For simplicity, we assume symmetric boundary conditions in this paper.

2.1. **Nonlocal variation model.**

We consider the following variational model introduced in [8, 19]:

$$E(u) := \lambda \sum_{i \in I} |\nabla_w u(i)| + \frac{1}{2} \sum_{i \in I} (u(i) - v(i))^2, \tag{2}$$

where

$$|\nabla_w u(i)| = \sqrt{\sum_{j \in I} (u(i) - u(j))^2 w(i,j)} \tag{3}$$

is called the nonlocal total variation of u. The variational model (2) is denoted as NLTV. The denoised image $\bar{v}$ is the minimizer of the functional (2). For $i \in I$ and d an odd integer, let $\mathcal{N}_i(d) = \{j \in I : \|j - i\|_\infty \leq (d-1)/2\}$ be the window with center i and size $d \times d$, simply written as $\mathcal{N}_i$. Similarly, denote $U_i(D)$ the search window with center i and size $D \times D$, simply written as U_i. We consider $w(i,j)$ used in [19]: $w(i,j)$ is taken as the one used in NL-means[1] without normalization, which is different from the weight in [7]. That is, the following weight is used

$$w(i,j) = \begin{cases} e^{-||v(\mathcal{N}_i)-v(\mathcal{N}_j)||_a^2/(2\sigma_r^2)} & \text{if} \quad j \in U_i(D) \\ 0 & \text{else} \end{cases}, \tag{4}$$

where $\sigma_r > 0$ is a control parameter, and

$$||v(\mathcal{N}_i) - v(\mathcal{N}_j)||_a^2 = \frac{\sum_{k\in\mathcal{N}_i(d)} a(i,k)|v(k) - v(\mathcal{T}(k))|^2}{\sum_{k\in\mathcal{N}_i(d)} a(i,k)}, \tag{5}$$

with $\mathcal{T} = \mathcal{T}_{ij}$ being the translation mapping of $\mathcal{N}_i$ onto $\mathcal{N}_j$: $\mathcal{T}(k) = k-i+j, k \in \mathcal{N}_i$, and $a(i,k) = e^{-\|i-k\|^2/2\sigma_s^2}$ ($\sigma_s = (d-1)/4$ is a good choice).

The weight $w(i,j)$ is used to estimate the similarity between two pixels i and j. When the two pixels i and j are similar, $w(i,j)$ is large, which makes the recovered values $\bar{v}(i)$ and $\bar{v}(j)$ close. Note that for $i = (i_1, i_2)$, if we use

$$w(i,j) = \begin{cases} 1 & \text{if} \quad j \in \{(i_1+1, i_2), (i_1, i_2+1)\} \\ 0 & \text{else} \end{cases}, \tag{6}$$

then (2) reduces to the classical total variation model [24] denoted as ROF. Thus NLTV is more general and more adaptive to image content than ROF.

For any fixed i, $|\nabla_w u(i)|$ is a convex functional of u. Therefore $\sum_{i\in I} |\nabla_w u(i)|$ is convex. Since the fidelity term is strictly convex, so is the energy function $E(u)$. Thus the gradient descent method can be used to find the minimizer. Write

$$W_u(i,j) = \frac{w(i,j)}{|\nabla_w u(i)|} + \frac{w(j,i)}{|\nabla_w u(j)|}, \tag{7}$$

then the gradient of $E(u)$ [19] is $\nabla E(u) = \{\nabla E(u)(i)\}_{i\in I}$, with

$$\nabla E(u)(i) = \lambda \sum_{j\in I} (u(i) - u(j))\, W_u(i,j) + u(i) - v(i). \tag{8}$$

Notice that the equation (7) is not well defined at points $|\nabla_w u(i)| = 0$. The common technique is to replace $|\nabla_w u(i)|$ by $\sqrt{|\nabla_w u(i)|^2 + \beta}$ with β a small positive constant. The gradient descent algorithm can be expressed as follows,

$$u^{k+1}(i) = u^k(i) - t_k \nabla E(u^k)(i), \quad k = 0, 1, 2, \cdots \tag{9}$$

where $u^0 = v$ and t_k is chosen such that $E(u^{k+1}) < E(u^k)$.

2.2. **Mean squared error and Stein's unbiased risk estimate.** The mean squared error (MSE) of the denoised image $\bar{v}$ with respect to the original image u_0 is

$$\text{MSE}(\bar{v}) = \frac{1}{MN} \sum_{i\in I} (\bar{v}(i) - u_0(i))^2. \tag{10}$$

SURE provides an unbiased estimation of the true MSE for the denoised image of an algorithm. It is expressed by the following analytical equation [25, 27]:

$$\text{SURE} = \frac{1}{MN} \sum_{i\in I} (\bar{v}(i) - v(i))^2 - \sigma^2 + 2\sigma^2 \frac{\text{div}_v\{\bar{v}\}}{MN}, \tag{11}$$

where $\text{div}_v\{\bar{v}\}$ is the divergence of the denoised image $\bar{v}$ of the algorithm with respect to the noisy image v

$$\text{div}_v\{\bar{v}\} = \sum_{i\in I} \frac{\partial \bar{v}_i}{\partial v_i}, \tag{12}$$

and the variance of the noise σ^2 is supposed to be known.

3. Proposed models based on Frequency domain nonlocal total variation.

The Fourier transform structures the image data in a totally different way from the standard basis. For example, uniform areas generate low frequency values only and microtextures may be localized in some narrow bands, whereas edges spread frequencies over a wide spectral band. In contrast to these natural structures, Gaussian noise remains Gaussian noise. These remarks indicate that denoising in the Fourier domain is likely to produce an effect complementary to that of denoising in the direct domain. As a result, additional denoising without artifact accumulation can be expected by combining denoising in both domains. Let us start by describing denoising applied in the Fourier domain only.

3.1. FNLTV model.

For an image $u = \{u(i), i = (i_1, i_2) \in I\}$, denote its discrete Fourier transform as $\hat{u} = \{\hat{u}(\omega), \omega = (\omega_1, \omega_2) \in I\}$, where we still use I to represent the index set in the frequency domain for simplicity and we use the normalized discrete Fourier transform. That is, the Fourier transform of u is defined as[1]

$$\hat{u}(\omega_1, \omega_2) = \frac{1}{\sqrt{MN}} \sum_{i_0=1}^{M-1} \sum_{i_0=1}^{N-1} u(i_1, i_2) e^{-i2\pi(i_1\omega_1/M + i_2\omega_2/N)}. \tag{13}$$

In the frequency domain, the noise model becomes

$$\hat{v} = \hat{u_0} + \hat{\eta}.$$

By the orthogonality of the discrete Fourier transform, $\hat{\eta}$ is also Gaussian noise with variance σ^2. Taking into account the following energy functional, we get the frequency domain nonlocal total variation (FNLTV) model, which is written in the discrete form:

$$E(u) = \lambda_f \sum_{\omega \in I} |\nabla_{w_f} \hat{u}(\omega)| + \frac{1}{2} \sum_{\omega \in I} |\hat{u}(\omega) - \hat{v}(\omega)|^2, \tag{14}$$

where

$$|\nabla_{w_f} \hat{u}(\omega)| = \sqrt{\sum_{\xi \in I} |\hat{u}(\omega) - \hat{u}(\xi)|^2 w_f(\omega, \xi)},$$

noting that $\sum_{\omega \in I} |\hat{u}(\omega) - \hat{v}(\omega)|^2 = \sum_{i \in I} |u(i) - v(i)|^2$ by the definition (13). The weight $w_f(\omega, \xi)$ is taken as (4) with $\hat{v}$ replacing v, and the parameters d_f, D_f, σ_{rf} replacing d, D, σ_r respectively. That is,

$$w_f(\omega, \xi) = \begin{cases} e^{-||\hat{v}(\mathcal{N}_\omega) - \hat{v}(\mathcal{N}_\xi)||_a^2 / (2\sigma_{rf}^2)} & \text{if} \quad \xi \in U_\omega(D_f) \\ 0 & \text{else} \end{cases}, \tag{15}$$

where

$$||\hat{v}(\mathcal{N}_\omega) - \hat{v}(\mathcal{N}_\xi)||_a^2 = \frac{\sum_{k \in \mathcal{N}_\omega(d_f)} a(\omega, k) |\hat{v}(k) - \hat{v}(\mathcal{T}(k))|^2}{\sum_{k \in \mathcal{N}_\omega(d_f)} a(\omega, k)}. \tag{16}$$

Since the Fourier transform is linear, the functional (14) is still strictly convex. Thus we can also use the gradient descent algorithm to find the minimizer.

Let $J_f(u) = \sum_{\omega \in I} |\nabla_{w_f} \hat{u}(\omega)|$ be the frequency domain regularizer. We will show that the gradient of $J_f(u)$ can be expressed as follows:

$$\nabla J_f(u) = \Re f, \text{with } \hat{f}(w) = \sum_{\xi \in I} (\hat{u}(\omega) - \hat{u}(\xi)) \left(\frac{w_f(\omega, \xi)}{|\nabla_{w_f} \hat{u}(\omega)|} + \frac{w_f(\xi, \omega)}{|\nabla_{w_f} \hat{u}(\xi)|} \right), \tag{17}$$

[1] Here, i is the square root of -1, while i represent an index for a pixel in other places in this paper.

where $\Re$ represents the real part of some complex number. In practice, as NLTV, $|\nabla_{w_f}\hat{u}(\omega)|$ is replaced by $\sqrt{|\nabla_{w_f}\hat{u}(\omega)|^2+\beta}$ with β a small positive constant to avoid the division by 0. It is easy to see that

$$\begin{aligned}\frac{d}{dt}J_f(u+tv)|_{t=0} &= \sum_{\omega\in I}\sum_{\xi\in I}[(\hat{u}(\omega)-\hat{u}(\xi))(\bar{\hat{v}}(\omega)-\bar{\hat{v}}(\xi)) \\ &+ (\hat{v}(\omega)-\hat{v}(\xi))(\bar{\hat{u}}(\omega)-\bar{\hat{u}}(\xi))]w_f(\omega,\xi)\frac{1}{2|\nabla_{w_f}\hat{u}(\omega)|}. \end{aligned} \tag{18}$$

With a similar proof to (8), we obtain

$$\sum_{\omega\in I}\sum_{\xi\in I}(\hat{u}(\omega)-\hat{u}(\xi))(\bar{\hat{v}}(\omega)-\bar{\hat{v}}(\xi))w_f(\omega,\xi)\frac{1}{2|\nabla_{w_f}\hat{u}(\omega)|} = \sum_{\omega\in I}\frac{1}{2}\hat{f}(\omega)\bar{\hat{v}}(\omega). \tag{19}$$

Then by (18) and (19), it follows that

$$\begin{aligned}\frac{d}{dt}J_f(u+tv)|_{t=0} &= \sum_{\omega\in I}\frac{1}{2}\hat{f}(\omega)\bar{\hat{v}}(\omega)+\overline{\sum_{\omega\in I}\frac{1}{2}\hat{f}(\omega)\bar{\hat{v}}(\omega)} \\ &= \sum_{i\in I}\frac{1}{2}f(i)\bar{v}(i)+\sum_{i\in I}\frac{1}{2}\bar{f}(i)\bar{v}(i) \\ &= \sum_{i\in I}\Re f(i)v(i).\end{aligned}$$

Thus, we have the conclusion (17).

3.2. **SFNLTV and Local SFNLTV.** Now we propose a model to combine the regularization terms of NLTV and FNLTV to exploit the advantages of both models. It is called spatial-frequency domain nonlocal total variation (SFNLTV) model, which involves minimizing the following energy functional:

$$E(u)=\lambda\sum_{i\in I}|\nabla_w u(i)|+\lambda_f\sum_{\omega\in I}|\nabla_{w_f}\hat{u}(\omega)|+\frac{1}{2}\sum_{i\in I}(u(i)-v(i))^2, \tag{20}$$

recalling that

$$|\nabla_w u(i)| = \sqrt{\sum_{j\in I}(u(i)-u(j))^2 w(i,j)}, \tag{21}$$

$$|\nabla_{w_f}\hat{u}(\omega)| = \sqrt{\sum_{\xi\in I}|\hat{u}(\omega)-\hat{u}(\xi)|^2 w_f(\omega,\xi)}. \tag{22}$$

Since the two regularization terms are both convex with respect to u, SFNLTV energy functional is also strictly convex, and can be minimized by the gradient descent algorithm. By (7), (8) and (17),

$$\begin{aligned}\nabla E(u)(i) &= \lambda\sum_{j\in I}(u(i)-u(j))\left(\frac{w(i,j)}{|\nabla_w u(i)|}+\frac{w(j,i)}{|\nabla_w u(j)|}\right) \\ &+\lambda_f\Re\left\{\mathfrak{F}^{-1}\left[\sum_{\xi\in I}(\hat{u}(\omega)-\hat{u}(\xi))\left(\frac{w_f(\omega,\xi)}{|\nabla_{w_f}\hat{u}(\omega)|}+\frac{w_f(\xi,\omega)}{|\nabla_{w_f}\hat{u}(\xi)|}\right)\right]\right\} \\ &+u(i)-v(i),\end{aligned} \tag{23}$$

where $\Re$ and $\mathfrak{F}^{-1}$ represent the real part and the inverse discrete Fourier transform respectively. The algorithm is summarized in Algorithm 1.

Since Fourier transform is well suited for dealing with regular image patterns, and local image regions are more regular than entire images, we now take account of applying SFNLTV locally. Concretely, an image is divided into overlapping regions of size $S_r \times S_r$. The horizontal and vertical steps of moving regions are equal and denoted as n_s. SFNLTV is then applied to each region. Since regions are overlapping, some pixels belong to several regions, and thus have several estimations. The final estimation of each pixel is the average of all the estimations. In addition, to alleviate the edge effects caused by the Fourier transform, for each local region, the edges with the size of one pixel are removed from the denoised region. The algorithm is called local SFNLTV (L-SFNLTV). Note that there are two differences between L-SFNLTV and SFNLTV, which make L-SFNLTV better: firstly, local implementation replaces global implementation; secondly the average of several estimations is taken as the final estimation. Moreover, the local application methods can be easily implemented in parallel, which makes L-SFNLTV a fast method. The algorithm is summarized in Algorithm 2.

Algorithm 1 Image denoising by SFNLTV

```
Require: $v$.
Initialization $Iters$, $u^0 = v$, $t = 0.5, d_t = 0.8, \epsilon = 1e - 20,$ and $k = 0$.
Calculate $E(u^0)$ with (20) and $\nabla E(u^0)$ with (23).
Set $E(u^{-1}) = 1$.
while $|E(u^k) - E(u^{k-1})| > \epsilon$ and $t > \epsilon$ and $k < Iters$ do
    $u_{new} = u^k - t\nabla E(u^k)$.
    Calculate $E(u_{new})$ with (20).
    if $E(u_{new}) < E(u^k)$ then
        $u^{k+1} = u_{new}$.
        Calculate $\nabla E(u^{k+1})$ according to (23).
        $k = k + 1$.
    else $t = t \times d_t$.
    end if
end while
Ensure: $u^k$.
```

Algorithm 2 Image denoising by L-SFNLTV

```
Require: $v$.
Divide $v$ into overlapping regions of size $S_r \times S_r$. The horizontal and vertical
  steps of moving regions are equal and denoted as $n_s$.
for each region do
    Apply SFNLTV on this region.
end for
Return each region to the corresponding original location and take averages for
  repeated estimates.
Ensure: $\bar{v}$.
```

4. **Analyses of NLTV and FNLTV.** To better understand the proposed models, let us make some theoretical and empirical analysis. For empirical analysis, we use

the Peak Signal-to-Noise Ratio (PSNR) to measure the quality of restored images:

$$\text{PSNR }(\bar{v}) = 10\log_{10}\frac{255^2}{\text{MSE}(\bar{v})},$$

where MSE is defined in (10). In addition, in our experiments we use (512×512) images, Lena, Barbara, Bridge, Boats and (256×256) images, Peppers, House, Cameraman, which were all downloaded online. In this section and the following section, the level of noise is supposed to be known, otherwise there are methods to estimate it; see e.g. [12].

4.1. **Relations with neighborhood filters.** We will establish that NLTV is essentially a neighborhood filter by analyzing the gradient descent algorithm. Thus the image can be divided into local regions and NLTV is applied to each region, which can be implemented in parallel and speed up the algorithm.

Notice that by (4), the summation of $j \in I$ in (8) can be replaced by $j \in U_i^0 = U_i \backslash \{i\}$. By (8) and (9), we have

$$u^{k+1}(i) = \sum_{j \in U_i} u^k(j) W^k(i,j) + t_k v(i), \tag{24}$$

where

$$W^k(i,j) = \begin{cases} t_k \lambda W_{u^k}(i,j) & \text{if} \quad i \neq j \\ 1 - t_k - t_k \lambda \sum_{j \in U_i^0} W_{u^k}(i,j) & \text{if} \quad i = j. \end{cases} \tag{25}$$

Note that

$$\sum_{j \in U_i} W^k(i,j) + t_k = 1.$$

By (24), we can obtain

$$\begin{aligned} u^{k+1}(i) \quad = \quad & \sum_{(j^0, j^1, \cdots, j^k) \in \mathcal{U}_i^k} W^k(i,j^0) W^{k-1}(j^0,j^1) \cdots W^0(j^{k-1},j^k) u^0(j^k) \\ & + t_0 \sum_{(j^0, j^1, \cdots, j^{k-1}) \in \mathcal{U}_i^{k-1}} W^k(i,j^0) \cdots W^1(j^{k-2},j^{k-1}) v(j^{k-1}) \\ & + \cdots \\ & + t_{k-1} \sum_{j^0 \in \mathcal{U}_i^0 = U_i} W^k(i,j^0) v(j^0) \\ & + t_k v(i). \end{aligned}$$

with

$$\mathcal{U}_i^k = \{(j^0, j^1, \cdots, j^k) : j^0 \in U_i, j^1 \in U_{j^0}, \cdots, j^{k-1} \in U_{j^{k-2}}, j^k \in U_{j^{k-1}}\}.$$

Recalling that $U_i = U_i(D)$, in our experiments, $D = 3$ is used. An illustration of the set $\mathcal{U}_i^2$ with $D = 3$ is shown in Figure 1. Thus we can see that $u^{k+1}(i)$ is the weighted average of its neighbors $v(j), j \in U_i(Dk + D - k)$ if we do not consider that $W^l(s,t)$ depends on the neighbors of s and t. In Figure 2, we plot the PSNR values versus iterations for different images. After 15 iterations, the iterative process for all the image approximately converges. Thus NLTV model is also an approximately local operator as ROF model stated in [20], which means that the denoised value at a given pixel has little dependence on the pixels which are far away from this given pixel. This can be verified by the Figures 12 and 13 of Section 4.4, from which we can see that local application of NLTV yields similar results to original NLTV. Thus, despite the fact that $W^l(s,t)$ depends on the neighbors of s and t, which

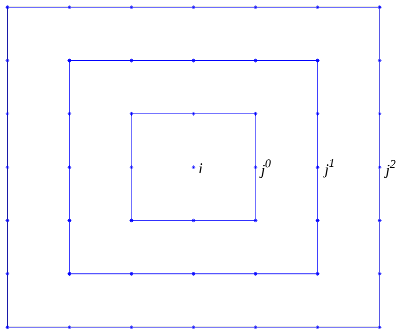


FIGURE 1. An illustration of the set $\mathcal{U}_i^2$ with $D = 3$.

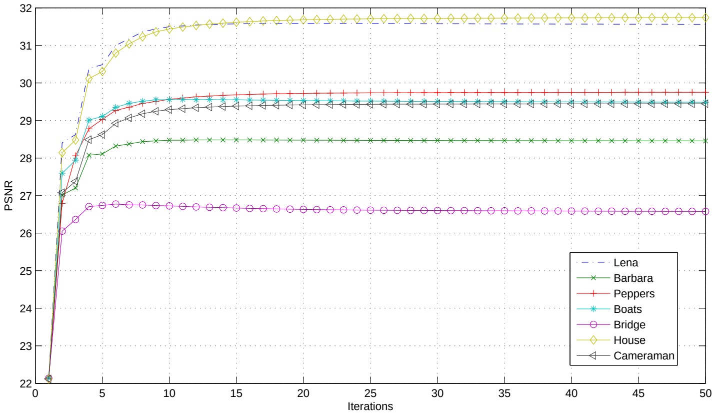


FIGURE 2. PSNR values versus iterations for different images for NLTV.

makes NLTV nonlocal, the dependence is negligible. Therefore, we can speed up NLTV by running it locally in parallel with multi-cores computers.

In addition, by (24) and (25), when i is an isolated point, there are no or few similar points in its neighbors, so $\sum_{j\in U_i^0(D)} W(i,j)$ is small. Thus the recovered value is close to the noisy one. This explains that NLTV is good at preserving isolated values unlike NL-means (visual results can be seen in Sec. 5.1).

4.2. **Parameter Choices.** We analyze the choices of the parameters empirically in the following. The influence of the search window size D is firstly studied with $d = 9$ fixed. In Figure 3, the PSNR values for different λ and D are plotted; in Figure 4, we show the denoised images for different values of D with the optimal λ obtained by Figure 3; and the denoised images with different λ are shown in Figure 5. From the Figures 4 and 5, we can see that when D increases, the staircase artifact is less obvious, since by the analysis in Section 4.1, the local neighborhood is larger. But the adjustment of λ can not produce better images with larger D. That is, if λ

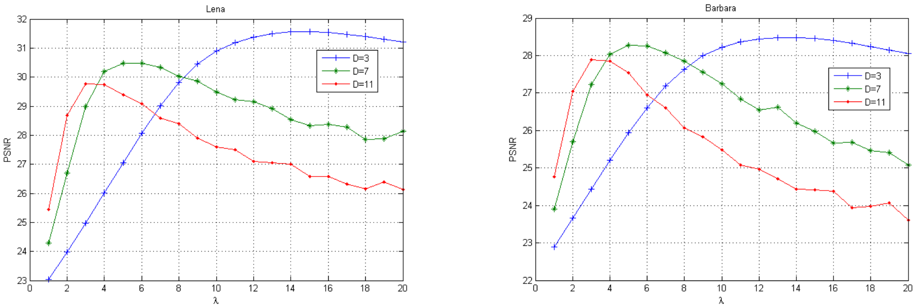
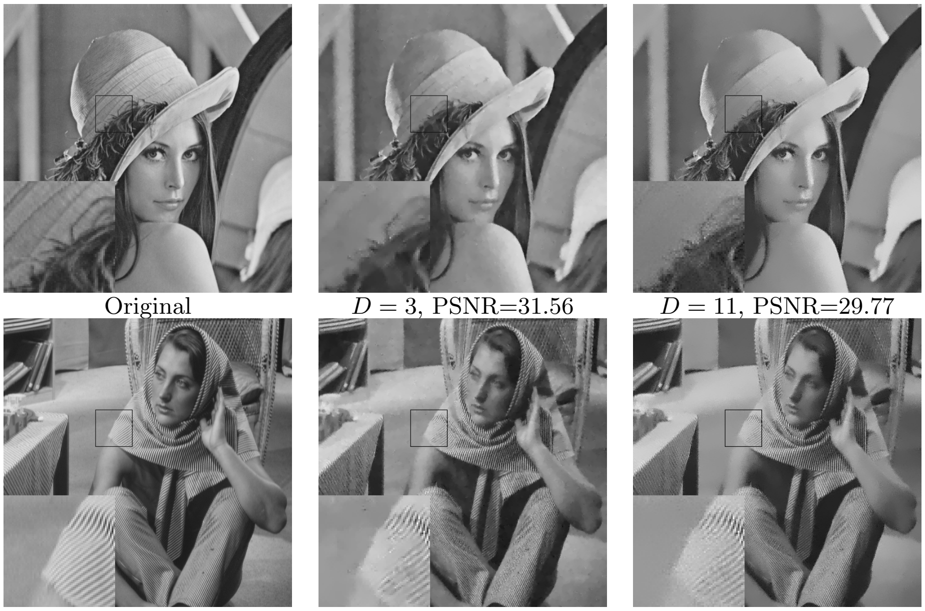


FIGURE 3. PSNR values versus λ for different search window sizes D with $\sigma = 20$.

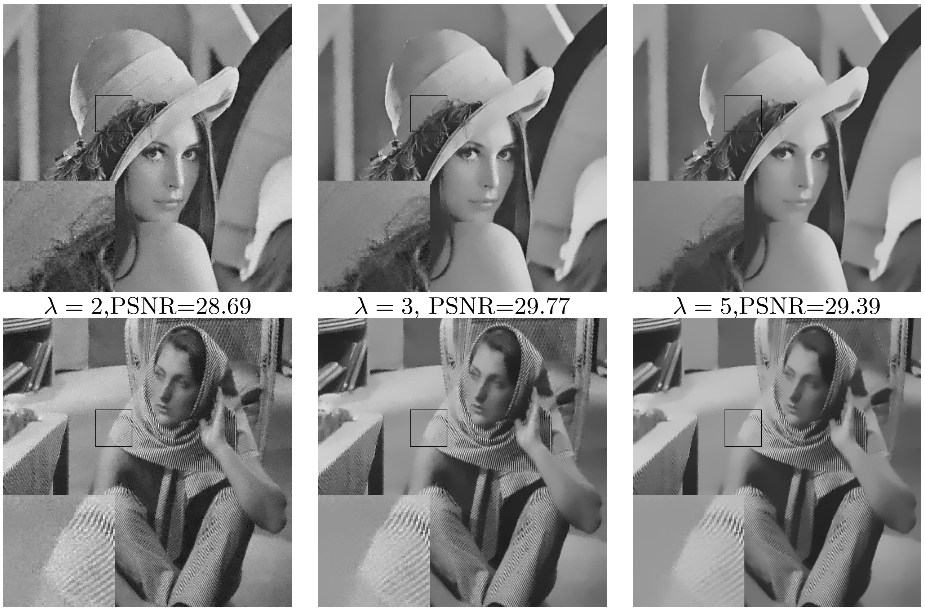


Original $D = 3$, PSNR=31.56 $D = 11$, PSNR=29.77

Original $D = 3$, PSNR=28.48 $D = 11$, PSNR=27.89

FIGURE 4. Denoised image by NLTV with $D = 3, 11$ with $\sigma = 20$.

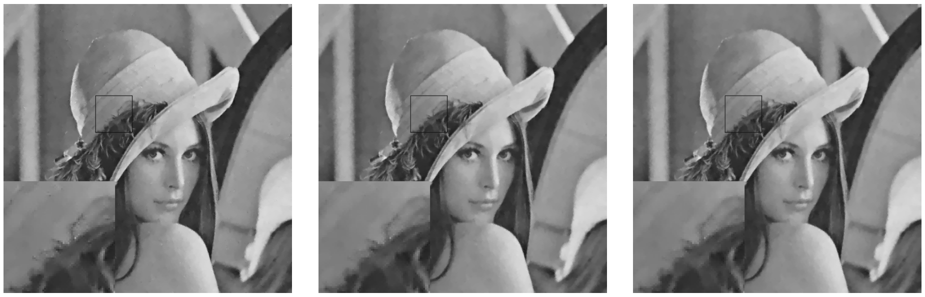
is large, then many fine details disappear; otherwise much noise remains, especially near edges. Thus the proposed choice $D = 21$ in [19] in accordance with the choice in NL-means in [1] is not an optimal choice for image denoising. Since a better choice for D is a smaller value, the denoising process is faster. In Figure 6, we compare the results for different d with $D = 3, \lambda = 15$. We can see that there are not significant differences among the three images. In fact, as the patch size d increases, in the limit case, all the nonzero weights $w(i, j)$ tend to be identical, thus close to ROF model if $D = 3$. Therefore, when d is larger, the images are more like ROF denoised images.



$\lambda = 2$,PSNR=28.69 $\lambda = 3$, PSNR=29.77 $\lambda = 5$,PSNR=29.39

$\lambda = 2$, PSNR=27.05 $\lambda = 3$, PSNR=27.89 $\lambda = 5$, PSNR=27.53

FIGURE 5. Denoised image by NLTV with $D = 11$ for different values of λ with $\sigma = 20$.



$d = 9$, PSNR=31.56 $d = 25$, PSNR=31.56 $d = 41$, PSNR=31.49

FIGURE 6. Denoised image by NLTV for different values of d with $\sigma = 20$.

4.3. **SURE based NLTV.** The choice of regularization parameter is important for NLTV. For this, in this section, we compute SURE for NLTV according to (11) in order to get an estimation of MSE. As the denoised image is not the result of an analytical computation but of an iterated sequence, we calculate SURE iteratively. The core of SURE is the calculation of the divergence. For the proposed algorithms, the calculation of divergence of the gradient of $E(u)$ is crucial. Since we use symmetric weights for NLTV, by (7) and (8), the gradient can be rewritten as

$$\nabla E(u)(i) = \lambda \sum_{j \in I} (u(i) - u(j))\, w(i,j)(\frac{1}{|\nabla_w u(i)|} + \frac{1}{|\nabla_w u(j)|}) + u(i) - v(i). \quad (26)$$

Note that the value at each pixel i depends on its neighbors, so we calculate the partial derivative of $\nabla E(u)(i)$ with respect to each pixel l in the image. It can be easily obtained that

$$\frac{\partial \nabla E(u)(i)}{\partial v(l)} = \lambda(P_1(i,l) + P_2(i,l) + P_3(i,l)) + \frac{\partial u(i)}{\partial v(l)} - \delta(i,l), \tag{27}$$

with

$$P_1(i,l) = \sum_{j\in I}(\frac{\partial u(i)}{\partial v(l)} - \frac{\partial u(j)}{\partial v(l)})w(i,j)(\frac{1}{|\nabla_w u(i)|} + \frac{1}{|\nabla_w u(j)|}), \tag{28}$$

$$P_2(i,l) = \sum_{j\in I}(u(i) - u(j))\frac{\partial w(i,j)}{\partial v(l)}(\frac{1}{|\nabla_w u(i)|} + \frac{1}{|\nabla_w u(j)|}), \tag{29}$$

$$P_3(i,l) = \sum_{j\in I}(u(i) - u(j))\, w(i,j)\left(\frac{\partial \frac{1}{|\nabla_w u(i)|}}{\partial v(l)} + \frac{\partial \frac{1}{|\nabla_w u(j)|}}{\partial v(l)}\right), \tag{30}$$

and

$$\delta(i,l) = \begin{cases} 1 & i = l \\ 0 & i \neq l \end{cases}, \tag{31}$$

in which

$$\begin{aligned}\frac{\partial \frac{1}{|\nabla_w u(i)|}}{\partial v(l)} = \quad & \frac{-1}{2|\nabla_w u(i)|^3}\sum_{j\in I}[2(u(i) - u(j))(\frac{\partial u(i)}{\partial v(l)} - \frac{\partial u(j)}{\partial v(l)})w(i,j) \\ & +(u(i) - u(j))^2\frac{\partial w(i,j)}{\partial v(l)}],\end{aligned} \tag{32}$$

and by (3), when $j \in U_i(D)$,

$$\begin{aligned}\frac{\partial w(i,j)}{\partial v(l)} = \quad & w(i,j)\frac{\sum_{k\in\mathcal{N}_i(d)} a(i,k)2(v(k) - v(\mathcal{T}(k)))(\frac{\partial v(k)}{\partial v(l)} - \frac{\partial v(\mathcal{T}(k))}{\partial v(l)})}{-2\sigma_r^2\sum_{k\in\mathcal{N}_i(d)} a(i,k)} \\ = \quad & w(i,j)\frac{\sum_{k\in\mathcal{N}_i(d)} a(i,k)(v(k) - v(\mathcal{T}(k)))(\delta(k,l) - \delta(\mathcal{T}(k),l))}{-\sigma_r^2\sum_{k\in\mathcal{N}_i(d)} a(i,k)}.\end{aligned}$$

By (9), it follows that

$$\frac{\partial u^{k+1}(i)}{\partial v(l)} = \frac{\partial u^k(i)}{\partial v(l)} - t_k\frac{\partial \nabla E(u^k)(i)}{\partial v(l)}, \qquad k = 0, 1, 2, \cdots \tag{33}$$

In our experiments, $u^0 = v$, so $\frac{\partial u^0(i)}{\partial v(l)} = \delta(i,l)$. Hence, by iteration, we can get the value of $\frac{\partial u^{k+1}(i)}{\partial v(l)}$ for $k = 0, 1, 2, \cdots$ by (27-33). Finally, we can obtain $\mathrm{div}_v\{\bar{v}\}$, and then the value of SURE by (12) and (11).

We compare the estimated MSE with the true MSE for different parts of Lena image shown in Figure 7 with different $\lambda \in \{1, 4, 7, \cdots, 49\}$, when the noise level $\sigma = 20$. Since the estimated MSE values fluctuate around the true MSE values, it is possible that the estimated MSE values are negative when the true MSE values are low. The comparisons demonstrate that the optimal λ obtained by estimated MSE and the true MSE roughly achieve at the same location. To show the performance of estimated MSE, we consider the application to the choice of optimal λ. We test the Barbara image and divide it into regions of size 16×16 or of size 32×32, then denoise the image for each region. The noise level $\sigma = 20$ is utilized. We take into account the comparisons between two cases: choosing the optimal λ from the set $\{1, 4, 7, \cdots, 49\}$ by estimated MSE, and choosing λ randomly in the same set. The

Table 1. PSNR values by choosing λ randomly (the first and third lines) and choosing λ according to the estimated MSE (the second and the fourth lines). The first two lines are with region size 16× 16; the last two lines are with region size 32× 32.

Lena	Barbara	Peppers	Boats	Bridge	House	Cameraman
28.65	26.50	28.14	27.36	25.08	28.92	27.49
30.59	28.02	29.55	29.02	26.54	30.73	29.10
28.96	26.50	28.41	27.53	25.08	28.52	27.80
30.89	28.22	29.82	29.19	26.65	30.98	29.27

results are displayed in Table 1, which shows that choosing λ according to estimated MSE works much better than choosing λ at random. The denoised Barbara images are displayed in Figure 8. Since the regions are disjoint, the region boundaries are discernable in the images.

4.4. **Parameter Analyses by Comparing NLTV and FNLTV.** Note that applying FNLTV is equivalent to denoise the discrete Fourier transform of the noisy image and then apply the inverse Fourier transform. In the Fourier domain, for the structures with high values, e.g. low frequencies, the weights w_f are very small. Thus the low frequencies and other structures with high values can be hardly influenced. In Figure 9, the Fourier transforms of noisy image, denoised image, noise and method noise (the difference of noisy image and denoised image $v-\bar{v}$) are shown. Like NLTV, the denoised Fourier transform has stair-casing effects. Comparing the Fourier transform of denoised image with that of noisy image, we find that FNLTV has really removed Gaussian noise in the Fourier domain, and the low frequencies and structures with high values stay unchanged. Moreover, we can see that the method noise has no low-pass Fourier components and FNLTV performs thus like a lowpass filter. Note that FNLTV is better than traditional lowpass filters since high frequency structures with high values stay unchanged too.

To understand well the SFNLTV model, we analyze the choice of regularization parameters by comparing the denoising performance and method noise of NLTV and FNLTV. Firstly, the regularization parameters λ and λ_f are both taken as 10, 16, 50. The visual results are shown in Figure 10, where the six method noise images are scaled in the same way to show their differences. The comparisons indicate that FNLTV model removes less image detail than NLTV model. Even when λ_f is large, i.e. $\lambda_f = 50$, the detail loss is not very evident. Note that when λ is large, much image details are lost for NLTV. Secondly, in Figure 11, we plot the root mean square of method noise and PSNR values[2] for more values of λ and λ_f: $\lambda, \lambda_f = 1, 2, \cdots, 50$. From the figures we can observe that when the regularization parameters λ or λ_f increase, the removing amount of noise are similar for the two models, while the PSNR values vary less for FNLTV model than for NLTV model, especially when the regularization parameters are large. Therefore, the choice of the regularization parameter λ_f is not very sensitive. Similar conclusions can be obtained for the search window size D_f. The comparisons show that NLTV and FNLTV have different properties. FNLTV demonstrates its advantage for retaining

[2] which is defined in Section 5.

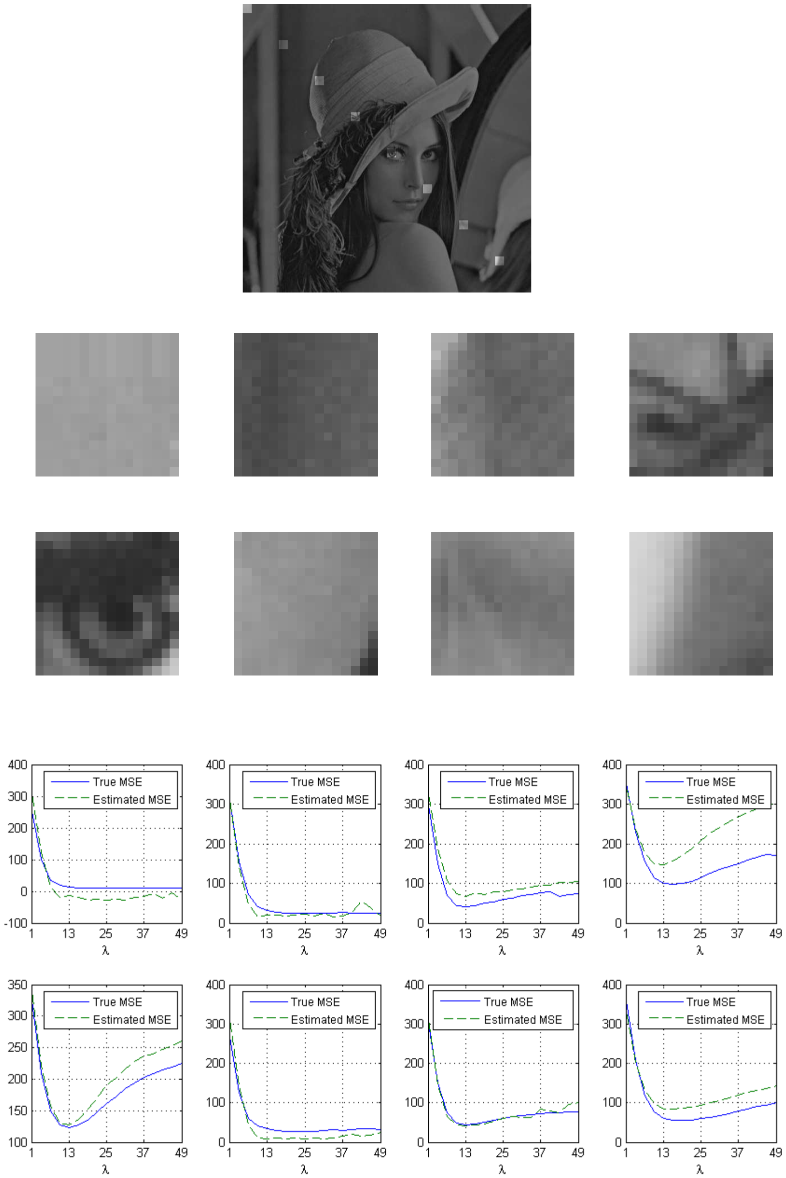
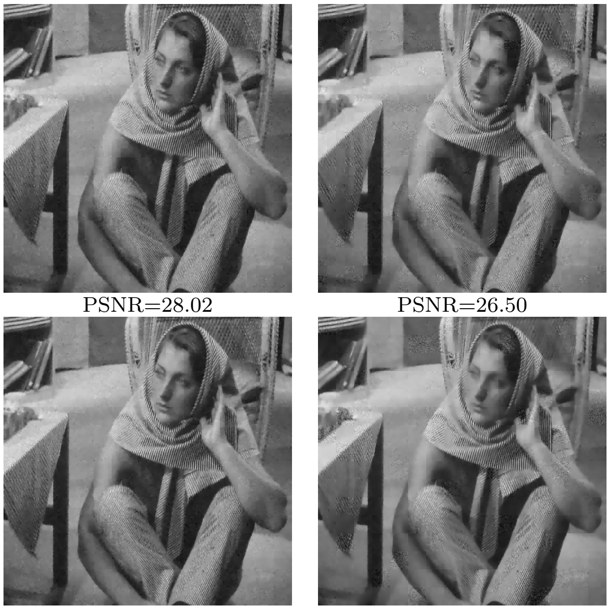


FIGURE 7. The top row is Lena image, with regions of size 16×16 highlighted; the following two rows are enlarged regions of the top row; the last two rows are the corresponding estimated MSE and true MSE for the corresponding regions.

image detail as shown in Figure 10. At the other hand, note that FNLTV can not remove noise as clearly as NLTV in homogenous regions. So the combination of them SFNLTV has the possibility to improve both of them.

To understand well L-SFNLTV, we first test the performance of local application of FNLTV. That is, like L-SFNLTV, we divide the image into local square regions,



PSNR=28.02 PSNR=26.50

PSNR=28.22 PSNR=26.50

Figure 8. Left: image denoised by choosing λ with smallest estimated MSE; right: choosing λ randomly. Top: region size 16× 16; bottom region size 32× 32.

and then apply FNLTV for each local region respectively. The results are shown in Figure 12. Note that the local version of FNLTV yields images which are more visually pleasant than global FNLTV does. When the local regions are not overlapping, the region boundaries are evident. While the regions are overlapping the boundaries can disappear. In fact, as DFT is a global operator, the estimations from different local regions for same pixels are generally different.

Furthermore, we apply NLTV locally in the same manner as FNLTV for comparison. The results in Figure 12 and Figure 13 show that the local application of NLTV does not make great difference both visually and in terms of PSNR. This verifies that NLTV is essentially a neighborhood filter as stated in Section 4.1. When the local regions are overlapping, the resulting images are also similar, since all the estimations for same pixels generally have similar values as long as the local region is large enough.

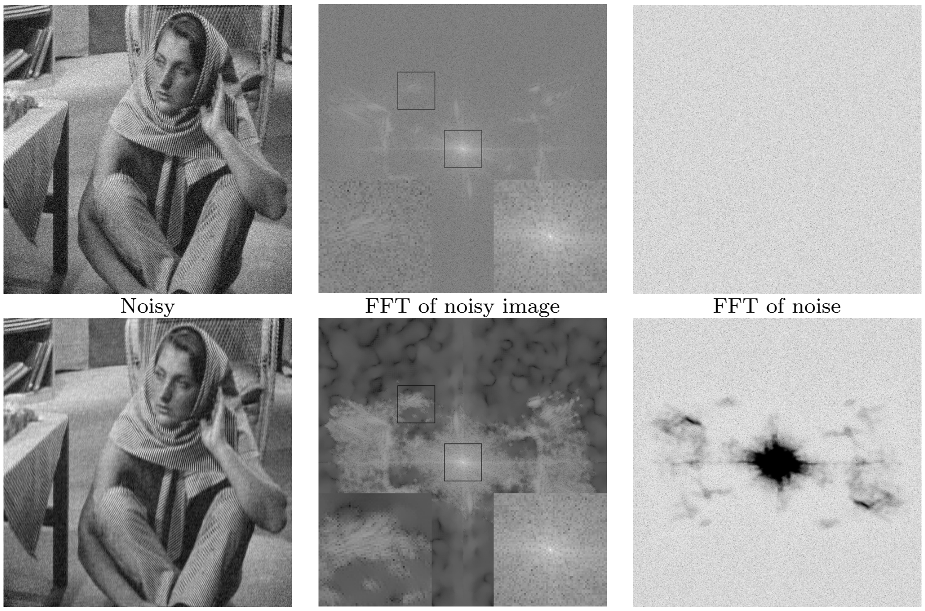
5. **Simulations.**

5.1. **Performance of NLTV and SFNLTV.**

For NLTV, we test our images in the case $\sigma = 20$, and search for the best PSNR value in average for the tested images with $\sigma_r = 20$ fixed and other parameters varying in some ranges. The parameter values $D = 3, d = 9, \lambda = 14$ are then utilized.

NLTV model is based on the idea of ROF model [24] and NL-means [1]. We now compare NLTV with them. Recall that the denoised image by NL-means is

$$\bar{v}(i) = \frac{\sum_{j \in U_i(D)} w(i,j) v(j)}{\sum_{j \in U_i(D)} w(i,j)},$$



Noisy

FFT of noisy image

FFT of noise

FNLTV

FFT of denoised image

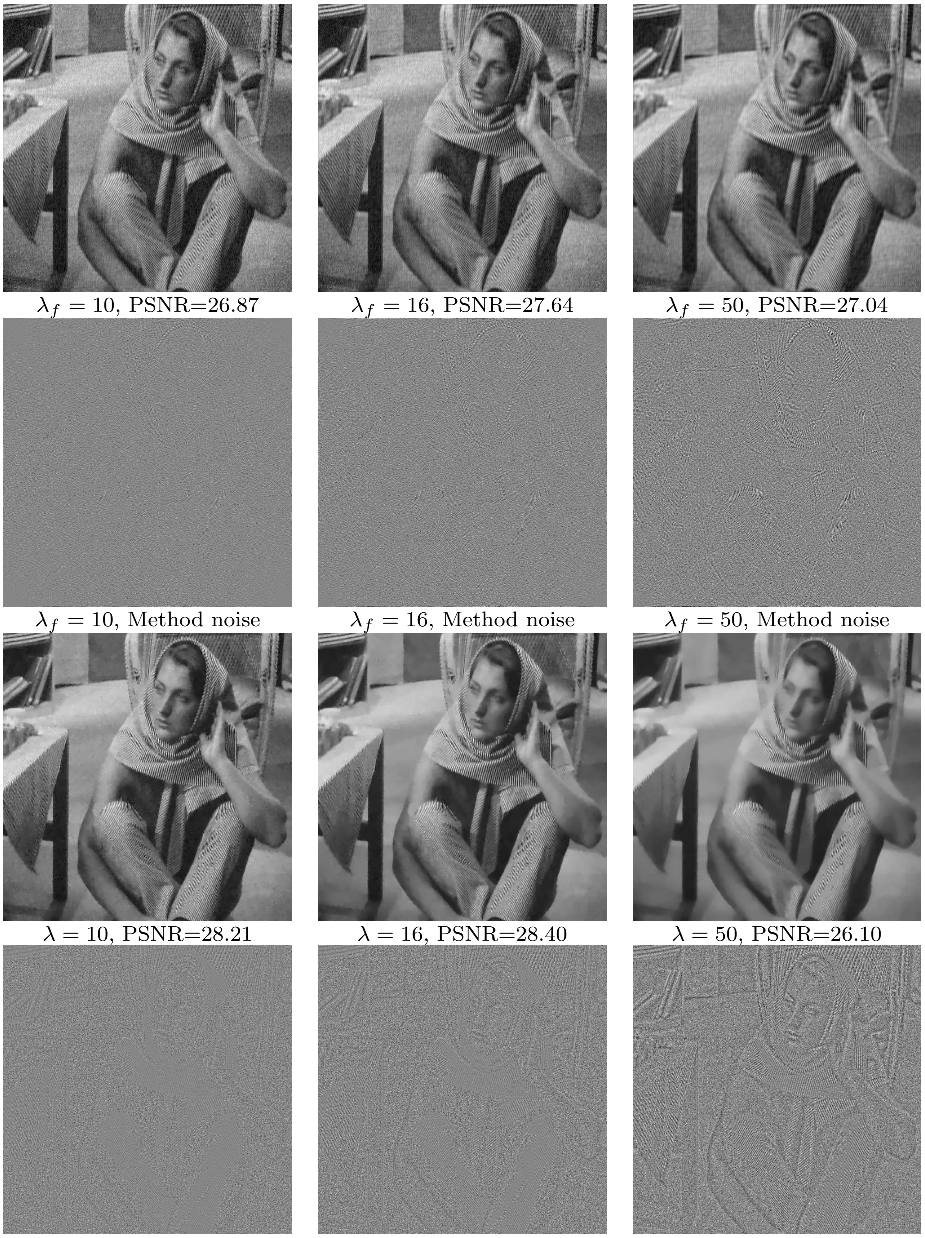
FFT of method noise

FIGURE 9. Left: noisy image and denoised image by FNLTV. Middle: the corresponding Fourier transforms of left column, where the bottom one can also be considered as denoised Fourier transform of noisy image. Right: the Fourier transforms of noise and method noise.

where $w(i,j)$ is defined in (4), and ROF model is obtained by (2) and (3) with $w(i,j)$ defined in (6). For NL-means, we use $D = 9, d = 7, \sigma_r = 18$. For ROF model, $\lambda = 15$ is used.

For SFNLTV model, $w(i,j)$ is taken as (4) in NLTV model, and $w_f(\omega,\xi)$ is taken as (15) in FNLTV model. Note that we have now eight parameters λ, d, D, σ_r, λ_f, d_f, D_f, σ_{rf}. We use the same choices of parameters in $w(i,j)$ in NLTV model for general natural images, i.e. $d = 9, D = 3, \sigma_r = 20$, and test the other parameters in the case $\sigma = 20$. We find that $\lambda = 11, \lambda_f = 2, d_f = 9, D_f = 5, \sigma_{rf} = 16$ is a good choice for all the images. Therefore, despite that SFNLTV has more parameters than NLTV model, the choice of parameters for SFNLTV model is less sensitive to images, which happens because the choice of the frequency domain regularizer coefficient λ_f is not very sensitive.

The comparisons of PSNR values are shown in Table 2, which shows that NLTV model is better than ROF model for all the images, and is similar to NL-means except for the image Barbara. SFNLTV model is better than NLTV model for all the images, even for Bridge image, and it is the best for almost all the tested images among all the methods. Examples of denoised images are shown in Figures 14 and 15. We can see that NLTV model is better than NL-means for isolated pixels (rare patches); the denoised images with NLTV model have less staircase artifacts than ROF model, and have less noise left. SFNLTV model is better than NLTV model for fine details, but is a little noisier in homogeneous regions.



$\lambda_f = 10$, PSNR=26.87

$\lambda_f = 16$, PSNR=27.64

$\lambda_f = 50$, PSNR=27.04

$\lambda_f = 10$, Method noise

$\lambda_f = 16$, Method noise

$\lambda_f = 50$, Method noise

$\lambda = 10$, PSNR=28.21

$\lambda = 16$, PSNR=28.40

$\lambda = 50$, PSNR=26.10

$\lambda = 10$, Method noise

$\lambda = 16$, Method noise

$\lambda = 50$, Method noise

FIGURE 10. Top row: images denoised by FNLTV with different λ_f; the third row: images denoised by NLTV with different λ; the second row and bottom row: the corresponding method noise images of the top row and the third row.

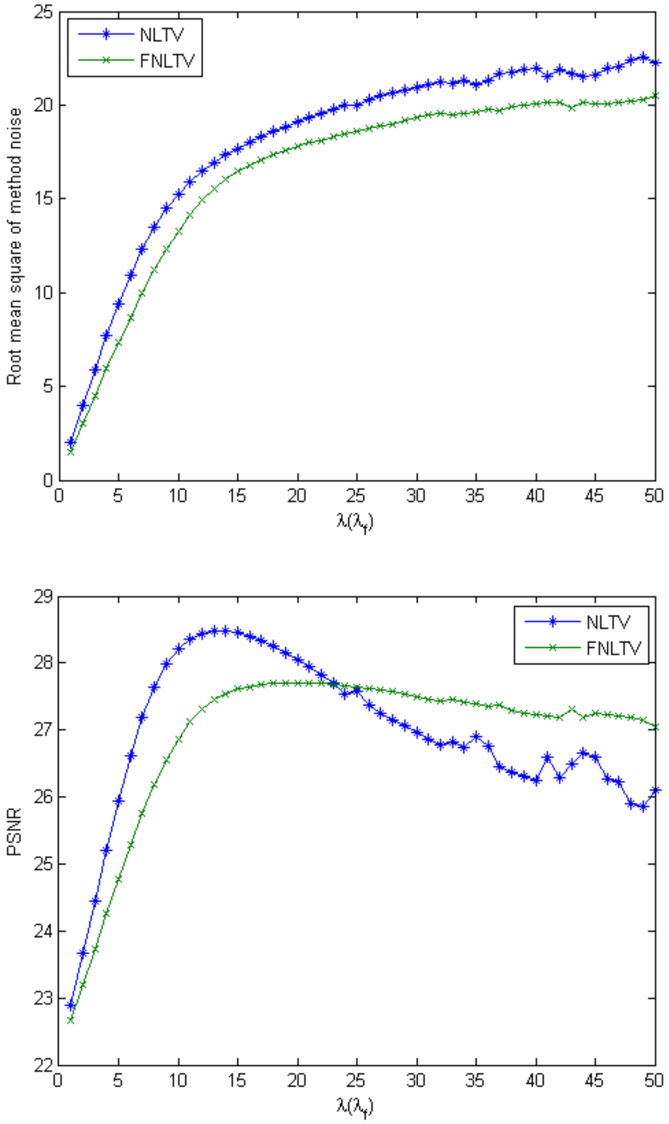
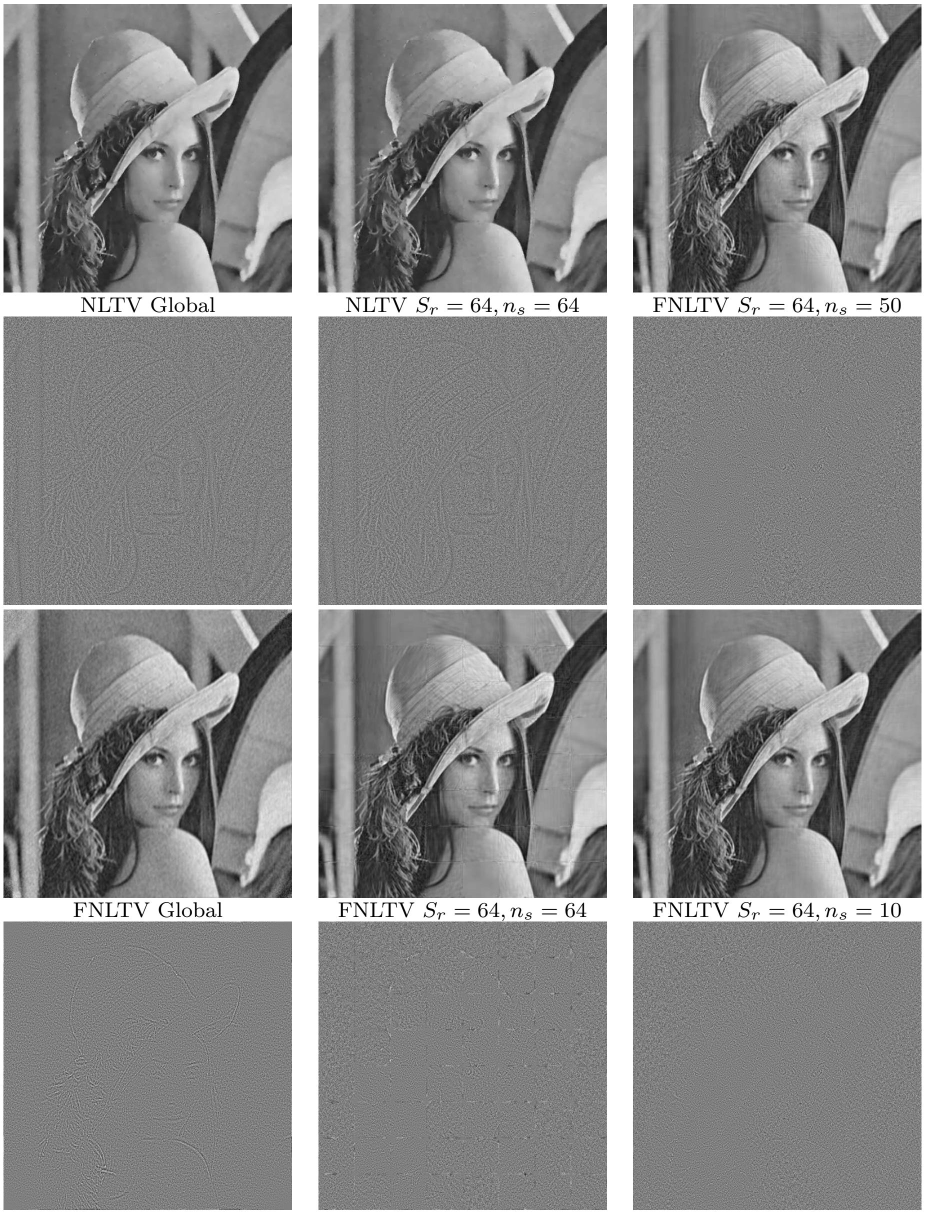


Figure 11. Top: Root mean square of method noise versus different λ (for NLTV) or λ_f (for FNLTV); Bottom: PSNR values versus different λ (for NLTV) or λ_f (for FNLTV).

5.2. **Performance of L-SFNLTV.** To test the performance of our algorithm L-SFNLTV, we have done experiments with $\sigma = 10, 20, 30, 50$, to compare with NLTV, SFNLTV and other NLTV related methods, NLSTV[15], RNLTV [17], BNLTV [18]. In addition, for the comparisons we use more images which were also downloaded online. For L-SFNLTV, the region size 16×16 and the moving step $n_s = 6$ are used. To be simple, we use fixed numbers of iterations for the gradient descent algorithm with L-SFNLTV, NLTV and SFNLTV. For L-SFNLTV, with each local



NLTV Global

NLTV $S_r = 64, n_s = 64$

FNLTV $S_r = 64, n_s = 50$

FNLTV Global

FNLTV $S_r = 64, n_s = 64$

FNLTV $S_r = 64, n_s = 10$

Figure 12. Left: image denoised by NLTV and FNLTV globally; Middle: image denoised by NLTV and FNLTV locally with no-overlapping regions of size 64×64; Right: image denoised by NLTV and FNLTV locally with overlapping regions of size 64×64 for moving step $n_s = 50$ (top) and $n_s = 10$ (third row). The second and bottom rows are the method noise images of the corresponding images of the top and third rows.

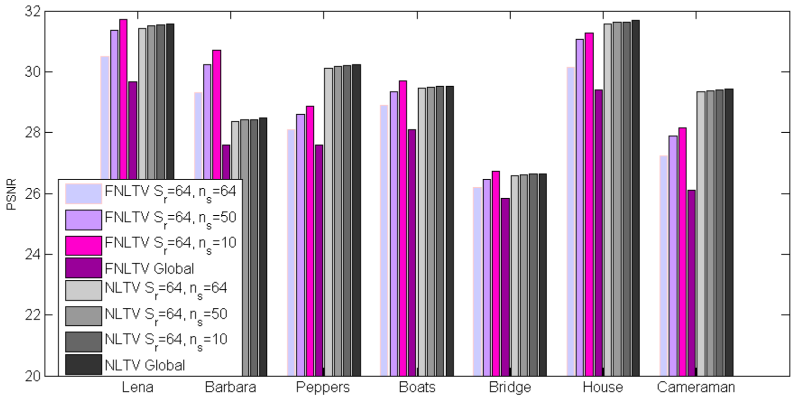


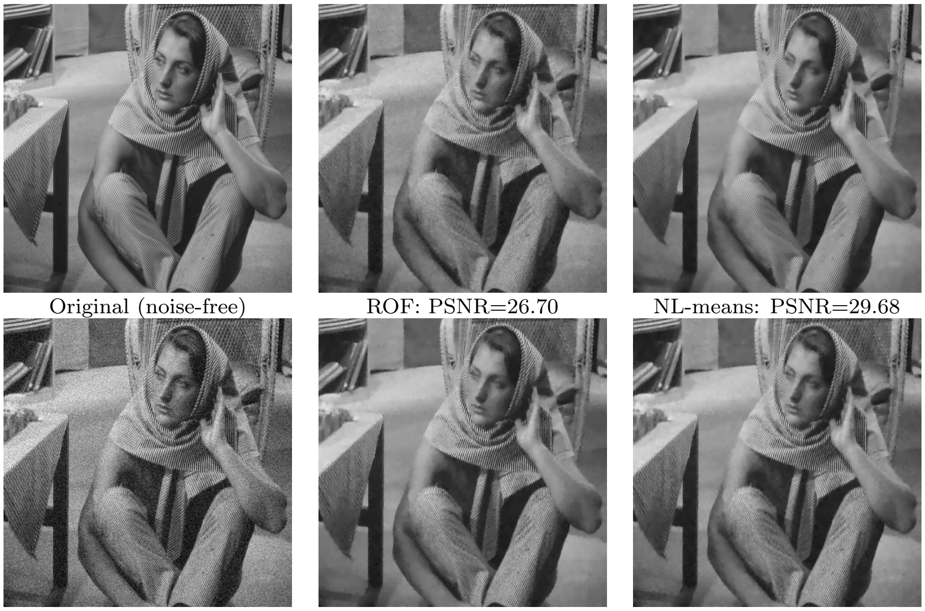
FIGURE 13. PSNR values for different images with different versions of FNLTV as in Figure 12 and the corresponding versions of NLTV.

TABLE 2. PSNR values for different images with NLTV, NL-means, ROF, and SFNLTV in the case $\sigma = 20$.

Image	Lena	Barbara	Peppers	Boats	Bridge	House	Cameraman
NLTV	31.56	28.48	30.16	29.51	26.66	31.68	29.41
NL-means	31.61	**29.68**	30.28	29.47	26.41	31.78	29.27
ROF	31.00	26.70	29.65	29.19	26.43	31.09	28.77
SFNLTV	**31.77**	29.19	**30.29**	**29.89**	**26.92**	**32.14**	**29.64**

region, we use 20 iterations for the gradient descent algorithm of SFNLTV. For NLTV and SFNLTV, 50 iterations are used. Other parameters of NLTV, SFNLTV and L-SFNLTV are chosen as in Table 3. For NLSTV and BNLTV, the authors do not report concrete values for the regularization parameter τ or λ. We choose the parameter which results in the best PSNR values in average for all our tested images, that is, $\tau = 0.006, 0.013, 0.029, 0.071$ and $\lambda = 27, 11, 7.5, 4.3$ for $\sigma = 10, 20, 30, 50$ respectively. For RNLTV, when $\sigma = 10, 20$, we use the proposed parameter values of the paper [17](the corresponding parameter values for $\sigma = 0.04, 0.08$, noting that the original images are normalized to the interval [0,1] in [17]); for $\sigma = 30, 50$, we also choose the parameter γ which yields the best PSNR values in average for the tested images Boats and Barbara from the authors' code, that is, $\gamma = 0.6, 2.4$. Other parameters are chosen according to the corresponding papers.

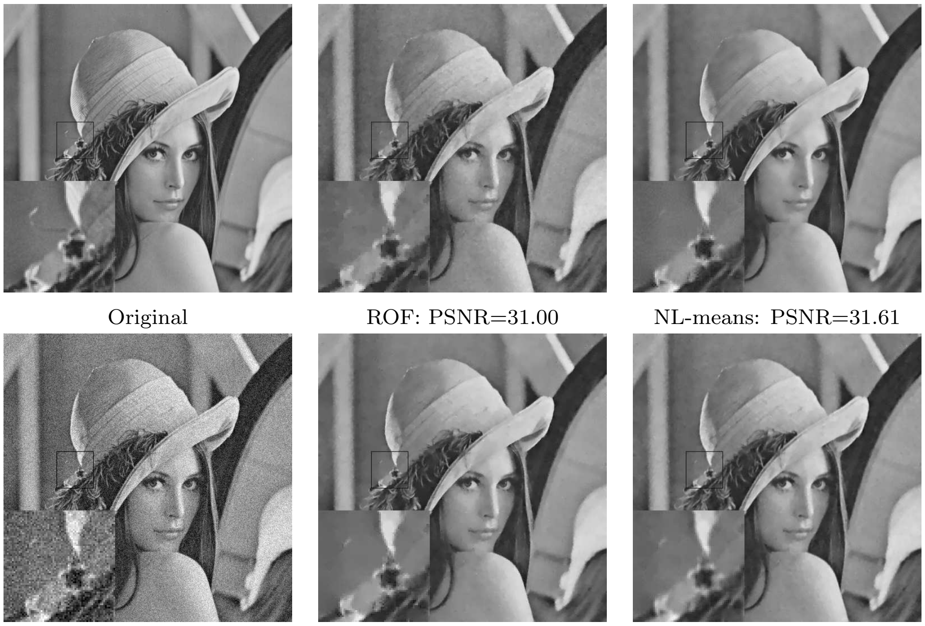
The comparison results of PSNR values are shown in Tables 4 and 5, which show that L-SFNLTV performs better than all the other NLTV related methods, NLSTV, RNLTV and BNLTV; and improves SFNLTV and NLTV. The visual comparisons can be seen in Figures 16, 17, and 18. NLSTV and SFNLTV models still contain stair-casing effects; RNLTV recovers thin structure very well, but homogenous regions are not smooth enough; BNLTV model recovers thin structures and homogenous regions well, but texture-like artifact exists. The artifacts of these



Original (noise-free) ROF: PSNR=26.70 NL-means: PSNR=29.68

Noisy image NLTV: PSNR=28.48 SFNLTV: PSNR=29.19

Figure 14. Denoised images by ROF model, NL-means, NLTV model and SFNLTV model for Barbara.



Original ROF: PSNR=31.00 NL-means: PSNR=31.61

Noisy ($\sigma = 20$) NLTV: PSNR=31.56 SFNLTV: PSNR=31.77

Figure 15. Denoised images by ROF model, NL-means, NLTV model and SFNLTV model for Lena.

Table 3. Choice of parameters of NLTV, SF(SFNLTVL) and L-SF(L-SFNLTV).

	NLTV/SF/L-SF			L-SF
σ	d	D	σ_r	λ_f
10	9	3	σ	6
20	9			14
30	11			25
50	15			49

NLTV	$\lambda = 2 + 0.6\sigma$	
SF	$D_f = 5$	$d_f = 9$
	$\sigma_{rf} = 0.8\sigma$	
	$\lambda = 0.55\sigma$	$\lambda_f = 1.6 + 0.02\sigma$
L-SF	$D_f = 3$	$d_f = 5$
	$\sigma_{rf} = \sigma$	$\lambda = 4$

methods are more evident when the noise level is high, e.g. $\sigma = 50$. Compared with these methods, our method has less artifact, while recovers thin structure and homogenous regions well.

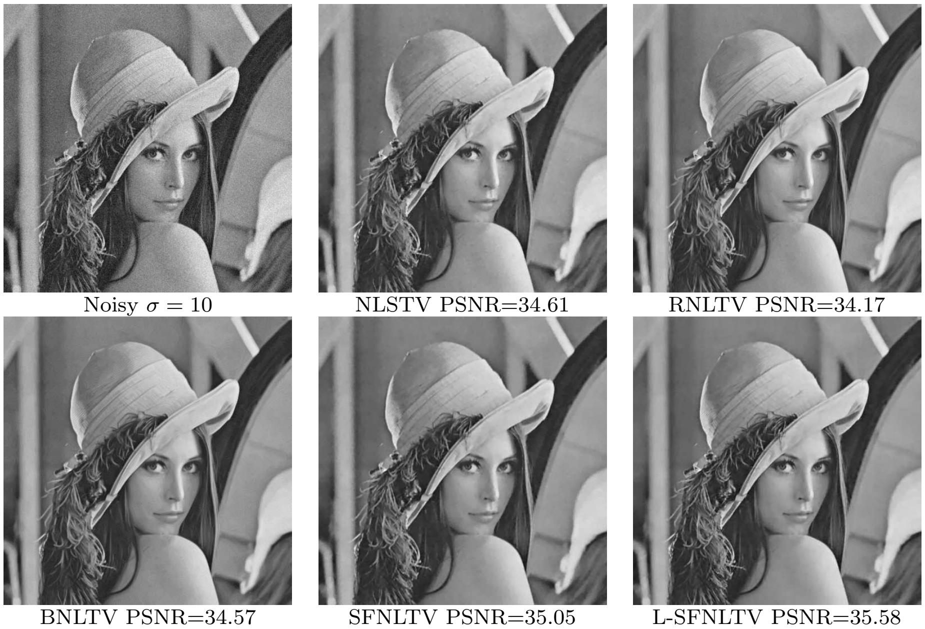
In addition, note also that L-SFNLTV is better than L-FNLTV; see Figure 19. In fact, the spatial domain nonlocal total variation helps to attenuate the artifact created by the algorithms based on Fourier transform. For L-FNLTV, we use the regions of size 16×16, the moving step $n_s = 6$ and $\lambda_f = \sigma$.

Finally, we compare the running time of L-SFNLTV with other methods. All the codes except NLSTV are run in the platform of MATLAB R2014a on a computer with 2.40 GHz Intel Core i7 CPU under Windows system. NLSTV[3] is run in the platform of MATLAB R2011a on another computer with 2.13 GHz Intel Core i3 CPU under Linux system. From Table 6, we can see that L-SFNLTV also has advantage in running time comparing with the other NLTV related algorithms except SFNLTV. Note that for L-SFNLTV, the local regions to implement SFNLTV are overlapping, and the CPU used in the experiments has only two cores, which explains why the benefit of multi-core parallelization seems to be low.

6. Conclusion.

In this paper, we first studied the nonlocal total variation (NLTV) model for image denoising, which was initially introduced for deconvolution in [19]. We established the relation between this model and neighborhood filters and derived the iterative formula of SURE (Steins Unbiased Risk Estimation) for NLTV with gradient descent algorithm to estimate the denoising performance without reference to original true image. Then, we extended the NLTV model to a spatial-frequency domain nonlocal total variation model (SFNLTV), which was showed to be better than NLTV for texture and fine details in images, and also better than the NL-means algorithm in most cases. Finally, we proposed a local version of SFNLTV, abbreviated as L-SFNLTV, to make full use of the advantage of Fourier transform and we showed that L-SFNLTV improved SFNLTV greatly. Since L-SFNLTV can be implemented in parallel, it leaded to a fast denoising algorithm. Experiments showed that L-SFNLTV had evident advantages both in denoising performance and implementation speed, comparing with other recently proposed NLTV-related methods.

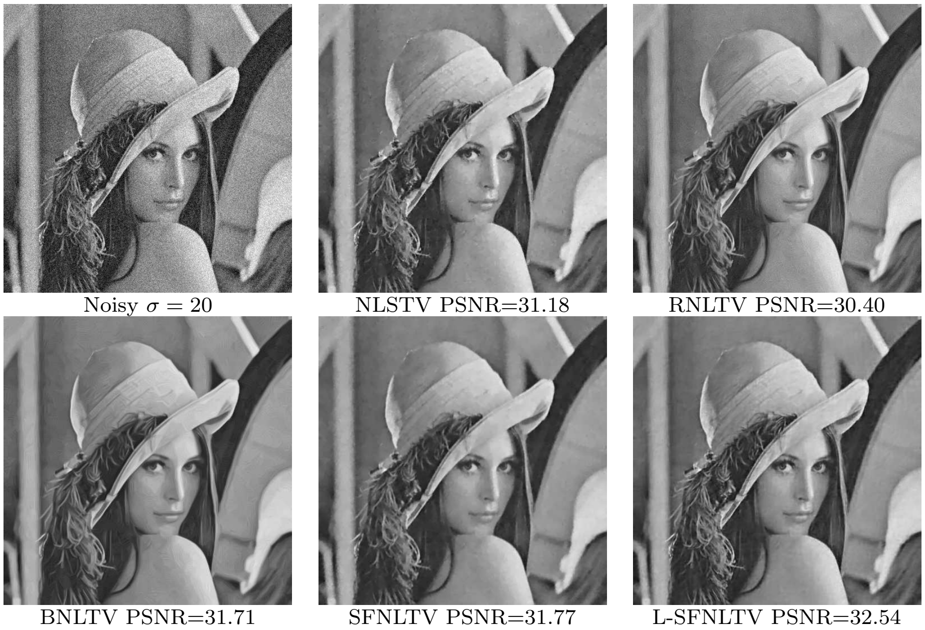
[3] The original code can not be implemented easily under Windows system, so we use a different computer installing Linux system.



Noisy $\sigma = 10$ NLSTV PSNR=34.61 RNLTV PSNR=34.17

BNLTV PSNR=34.57 SFNLTV PSNR=35.05 L-SFNLTV PSNR=35.58

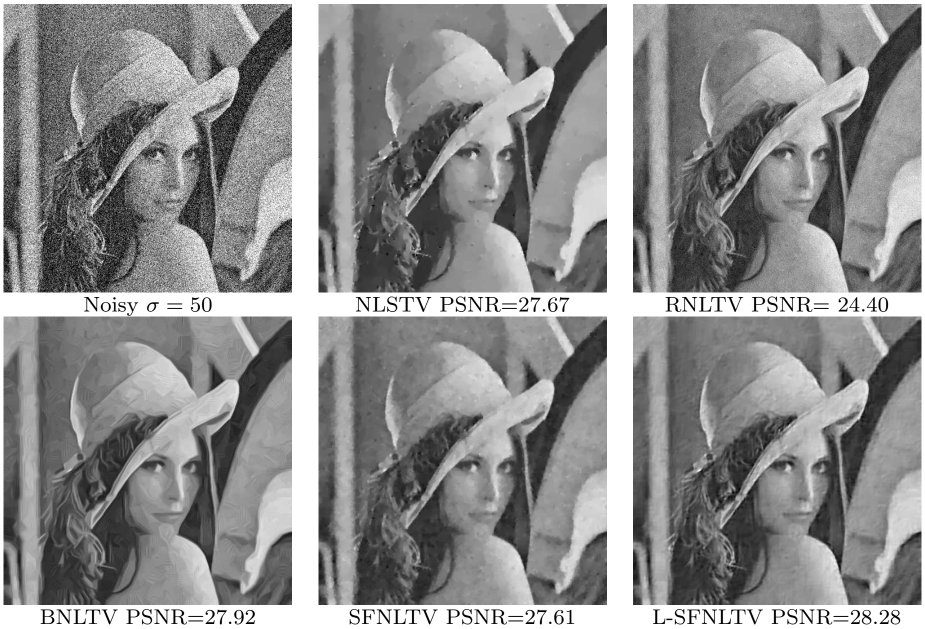
FIGURE 16. Denoised Lena images by L-SFNLTV, NLSTV [15], RNLTV [17], BNLTV [18] and SFNLTV in the case $\sigma = 10$.



Noisy $\sigma = 20$ NLSTV PSNR=31.18 RNLTV PSNR=30.40

BNLTV PSNR=31.71 SFNLTV PSNR=31.77 L-SFNLTV PSNR=32.54

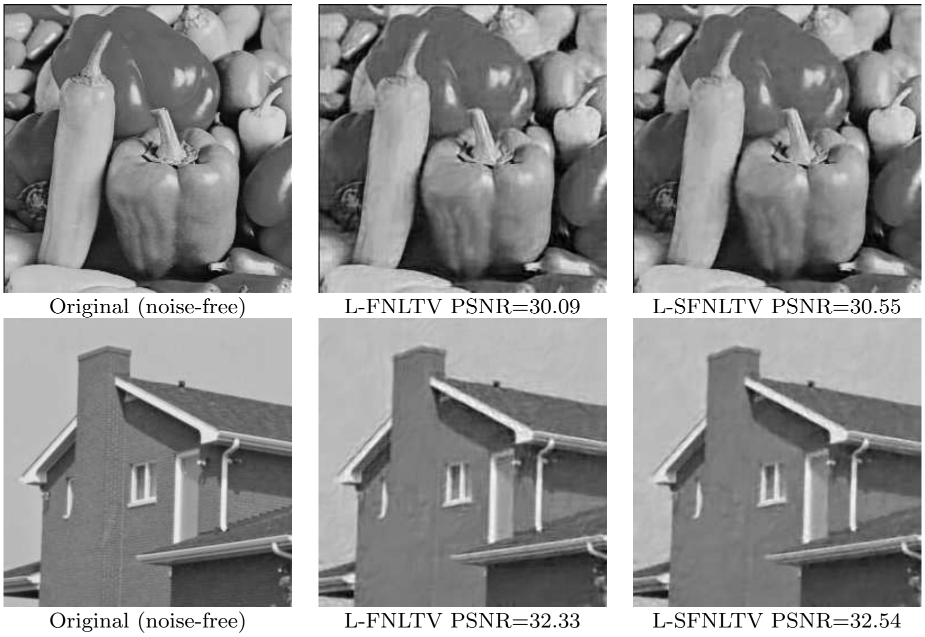
FIGURE 17. Denoised Lena images by L-SFNLTV, NLSTV [15], RNLTV [17], BNLTV [18] and SFNLTV in the case $\sigma = 20$.



Noisy $\sigma = 50$ NLSTV PSNR=27.67 RNLTV PSNR= 24.40

BNLTV PSNR=27.92 SFNLTV PSNR=27.61 L-SFNLTV PSNR=28.28

FIGURE 18. Denoised Lena images by L-SFNLTV, NLSTV [15], RNLTV [17], BNLTV [18] and SFNLTV in the case $\sigma = 50$.



Original (noise-free) L-FNLTV PSNR=30.09 L-SFNLTV PSNR=30.55

Original (noise-free) L-FNLTV PSNR=32.33 L-SFNLTV PSNR=32.54

FIGURE 19. Denoised images Peppers and House by L-FNLTV and L-SFNLTV in the case $\sigma = 20$.

TABLE 4. Comparisons of PSNR values for $\sigma = 10$ and $\sigma = 20$.

	NLTV	NLSTV	RNLTV	BNLTV	SFNLTV	L-SFNLTV
$\sigma = 10$						
Lena	34.74	34.61	34.17	34.57	35.05	**35.58**
Barbara	32.79	31.29	32.79	33.77	33.93	**34.46**
Peppers	33.80	34.06	33.11	32.31	33.82	**34.28**
Boats	32.80	33.15	32.68	32.83	33.42	**33.57**
Bridge	30.56	30.62	29.14	29.96	30.86	**30.97**
House	34.94	34.52	34.43	34.97	35.49	**35.62**
Cameraman	33.25	33.30	31.97	32.52	33.45	**33.65**
Monarch	32.98	33.00	31.49	32.41	33.51	**33.73**
Couple	32.73	33.07	32.78	32.96	33.21	**33.57**
Fingerprint	30.82	31.17	29.44	30.09	32.05	**32.34**
Hill	32.66	32.41	32.15	32.89	33.11	**33.29**
Man	33.18	33.00	32.19	32.96	33.40	**33.75**
$\sigma = 20$						
Lena	31.56	31.18	30.40	31.71	31.77	**32.54**
Barbara	28.48	27.23	29.19	30.40	29.19	**30.75**
Peppers	30.16	30.16	29.64	28.38	30.29	**30.55**
Boats	29.51	29.80	29.50	29.55	29.89	**30.42**
Bridge	26.66	27.03	26.63	26.06	26.92	**27.06**
House	31.68	30.93	30.21	32.17	32.14	**32.54**
Cameraman	29.41	29.41	28.54	28.85	**29.64**	29.63
Monarch	29.30	28.56	27.82	28.78	**29.66**	29.61
Couple	29.02	29.50	29.36	29.58	29.36	**30.19**
Fingerprint	26.71	26.83	26.94	26.22	27.50	**28.55**
Hill	29.58	29.13	28.90	29.92	29.84	**30.38**
Man	29.77	29.42	28.93	29.66	29.88	**30.31**

REFERENCES

[1] A. Buades, B. Coll and J. Morel, A review of image denoising algorithms, with a new one, *Multiscale Model. Simul.*, **4** (2005), 490–530.

[2] S. G. Chang, B. Yu and M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, *IEEE Trans. Image Process.*, **9** (2000), 1532–1546.

[3] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.*, **16** (2007), 2080–2095.

[4] D. Donoho and J. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, **81** (1994), 425–455.

[5] M. Elad and M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Trans. Image Process.*, **15** (2006), 3736–3745.

[6] G. Gilboa, J. Darbon, S. Osher and T. Chan, Nonlocal convex functionals for image regularization, *UCLA CAM Report*, 06–57.

[7] G. Gilboa and S. Osher, Nonlocal linear image regularization and supervised segmentation, *Multiscale Model. Simul.*, **6** (2007), 595–630.

[8] G. Gilboa and S. Osher, Nonlocal operators with applications to image processing, *Multiscale Model. Simul*, **7** (2008), 1005–1028.

[9] S. Gu, L. Zhang, W. Zuo and X. Feng, Weighted nuclear norm minimization with application to image denoising, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, 2862–2869.

TABLE 5. Comparisons of PSNR values for $\sigma = 30$ and $\sigma = 50$.

	NLTV	NLSTV	RNLTV	BNLTV	SFNLTV	L-SFNLTV
$\sigma = 30$						
Lena	29.67	29.86	27.89	29.98	29.82	**30.64**
Barbara	26.16	24.84	26.74	28.59	26.55	**28.63**
Peppers	27.96	**28.51**	27.26	26.58	28.13	28.44
Boats	27.73	28.14	27.18	27.94	27.93	**28.54**
Bridge	24.86	25.04	24.93	24.69	25.01	**25.23**
House	29.69	29.89	27.78	30.36	29.97	**30.65**
Cameraman	27.48	**27.76**	26.55	27.21	27.58	27.65
Monarch	27.09	26.89	25.95	27.00	**27.29**	**27.29**
Couple	27.11	27.62	27.01	27.88	27.31	**28.26**
Fingerprint	24.37	25.05	25.14	24.84	24.97	**26.48**
Hill	28.06	27.79	26.68	28.44	28.22	**28.76**
Man	28.03	27.93	26.74	28.10	28.08	**28.47**
$\sigma = 50$						
Lena	27.51	27.67	24.40	27.92	27.61	**28.28**
Barbara	24.00	23.17	23.30	**26.21**	24.11	26.00
Peppers	25.31	26.00	23.91	24.39	25.48	**26.03**
Boats	25.62	25.96	23.94	25.92	25.69	**26.28**
Bridge	23.09	23.12	22.50	23.03	23.18	**23.41**
House	27.23	27.57	24.12	28.10	27.40	**28.22**
Cameraman	24.87	**25.42**	23.41	25.08	24.83	25.20
Monarch	24.39	24.33	22.97	**24.69**	24.47	24.64
Couple	25.12	25.36	23.72	25.75	25.21	**26.00**
Fingerprint	21.71	22.36	22.48	22.96	22.16	**23.96**
Hill	26.35	25.87	23.50	26.60	26.46	**26.86**
Man	26.11	25.89	23.59	26.18	26.13	**26.41**

TABLE 6. Running time in second with grayscale images of size 256×256, where NLSTV is run under Linux system, and other algorithms are run under Windows system on another computer with a slightly faster processor.

NLSTV	RNLTV	BNLTV	SFNLTV	L-SFNLTV
22	3344	11.7	2.4	8.3

[10] H. Hu and J. Froment, Nonlocal total variation for image denoising, in *Symposium on Photonics and Optoelectronics (SOPO), 2012*, IEEE, 2012, 1–4.

[11] H. Hu, J. Froment and Q. Liu, A note on patch-based low-rank minimization for fast image denoising, *J. Visual Commun. Image Representation*, **50** (2018), 100–110.

[12] I. M. Johnstone and B. W. Silverman, Wavelet threshold estimators for data with correlated noise, *Journal of the royal statistical society: series B (statistical methodology)*, **59** (1997), 319–351.

[13] S. Kindermann, S. Osher and P. W. Jones, Deblurring and denoising of images by nonlocal functionals, *Multiscale Model. Simul.*, **4** (2006), 1091–1115.

[14] C. Knaus and M. Zwicker, Dual-domain image denoising, in *2013 IEEE International Conference on Image Processing*, IEEE, 2013, 440–444.

[15] S. Lefkimmiatis and S. Osher, Non-local structure tensor functionals for image regularization, *IEEE Transactions on Computational Imaging*, **1** (2015), 16–29.

[16] S. Lefkimmiatis, Universal denoising networks: a novel CNN architecture for image denoising, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 3204–3213.

[17] F. Li Z.and Malgouyres and T. Zeng, Regularized non-local total variation and application in image restoration, *J. Math . Imaging Vis.*, **59** (2017), 1–22.

[18] J. Liu and X. Zheng, A block nonlocal TV method for image restoration, *SIAM J. Imaging Sciences*, **10** (2017), 920–941.

[19] Y. Lou, X. Zhang, S. Osher and A. Bertozzi, Image recovery via nonlocal operators, *J. Sci. Comput.*, **42** (2010), 185–197.

[20] C. Louchet and L. Moisan, Total variation as a local filter, *SIAM J. Imaging Sci.*, **4** (2011), 651–694.

[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, Non-local sparse models for image restoration, in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, 2009, 2272–2279.

[22] N. Pierazzo, M. Lebrun, M. E. Rais, J.-M. Morel and G. Facciolo, Non-local dual image denoising, in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, 813–817.

[23] S. Ramani, T. Blu and M. Unser, Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms, *IEEE Trans. Image Process.*, **17** (2008), 1540–1554.

[24] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.

[25] C. M. Stein, Estimation of the mean of a multivariate normal distribution, *Ann. Statist.*, **9** (1981), 1135–1151.

[26] S. Tang, W. Gong, W. Li and W. Wang, Non-blind image deblurring method by local and nonlocal total variation models, *Signal Processing*, **94** (2014), 339 – 349.

[27] D. Van De Ville and M. Kocher, Sure-based non-local means, *IEEE Signal Process. Lett.*, **16** (2009), 973–976.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, *J. Mach. Learn. Res.*, **11** (2010), 3371–3408.

[29] M. Werlberger, T. Pock and H. Bischof, Motion estimation with non-local total variation regularization, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, 2464–2471.

[30] J. Xie, L. Xu and E. Chen, Image denoising and inpainting with deep neural networks, in *Advances in neural information processing systems*, 2012, 341–349.

[31] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising, *IEEE Trans. Image Process.*, **26** (2017), 3142–3155.

[32] K. Zhang, W. Zuo and L. Zhang, Ffdnet: Toward a fast and flexible solution for CNN-based image denoising, *IEEE Trans. Image Process.*, **27** (2018), 4608–4622.

[33] X. Zhang, M. Burger, X. Bresson and S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, *SIAM J. Imaging Sci.*, **3** (2010), 253–276.

[34] X. Zhang and T. Chan, Wavelet inpainting by nonlocal total variation, *Inverse Probl. Imaging*, **4** (2010), 191–210.

E-mail address: huhaijuan61@126.com
E-mail address: Jacques.Froment@univ-ubs.fr
E-mail address: wangbaoyan2005@163.com
E-mail address: fanxiequan@hotmail.com