

## A GEOMETRY GUIDED IMAGE DENOISING SCHEME

WEIHONG GUO

10900 Euclid Avenue  
Cleveland, OH 44106-7058, USA

JING QIN

10900 Euclid Avenue  
Cleveland, OH 44106-7058, USA

**ABSTRACT.** During image denoising, it is often difficult to balance between the removal of noise and the preservation of contrast and fine features, especially when the noise is excessive. We propose to efficiently balance the two using segmentation and more general geometry extraction transforms. Explained in Nonlocal-means (NL-means) framework, we introduce mutual position function to ensure averaging is only taken over pixels in the same segmentation phase, and provide a convolution kernel and weight function selection scheme to further improve the performance. To address unreliable segmentation due to more excessive noise, we use feature extraction transform that is more general than segmentation and less sensitive to noise. Unlike most denoising approaches that only work for one type of noise and/or involve heuristic parameter tuning, the proposed method comes with an automatic parameter selection scheme, and can be easily adapted for various types of noise, ranging from Gaussian, Poisson, Rician to Ultrasound noise etc. Comparison with the original NL-means as well as ROF, BM3D, K-SVD on various simulated data, MRI and SEM images indicates potentials of our method.

**1. Introduction.** Image denoising is a crucial process to remove noise from images for direct visualization or further image analysis. The biggest challenge in image denoising is the balance between the removal of noise and the preservation of salient features, like edges and fine structures. Parameter selection is also an issue, especially when the noise is inhomogeneous. There have been some works aiming to remove noise while preserving edges. These include PDE/variational methods [1, 2, 3, 4], Bayesian approach [5], adaptive smoothing [6], bilateral filtering [7, 8, 9], hybrid methods [10], NL-means [11] and its improvements and variants [12, 13, 14, 15, 16, 17, 18, 19, 22, 23]. However, most existing methods assume that the noise is of a specific type and/or homogeneous (with fixed parameter(s)). Since real images usually contain various types of noise, the application scope of these methods is narrow. They fail to preserve features well when the noise is excessive, thereby leading to inefficient recovery for heavily contaminated images. In this paper, we propose a fully automatic scheme and its variants to remove inhomogeneous noise of various types while preserving image geometry. This scheme outperforms most of the existing methods, especially when there is excessive noise.

---

2000 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

*Key words and phrases.* Image denoising, segmentation, nonlocal means.

The proposed method is explained using NL-means framework. Instead of taking the local structures into account, NL-means method explores the similarity and repetitive information in the whole image, i.e., non-locally. Suppose the noise-free image  $u(x)$  is defined in the bounded domain  $\Omega \subset \mathbb{R}^2$ . Let the observed noisy image be  $v = u + n$  with  $n$  an additive white Gaussian noise. Then the NL-means algorithm states that the estimated intensity of pixel  $x$  is a Gaussian weighted average of the intensities of all the pixels in the image:

$$NL(v)(x) = \frac{1}{C(x)} \int_{\Omega} w(x, y) v(y) dy \quad (1)$$

where

$$w(x, y) = \exp\left(-\frac{\int_{\mathbb{R}^2} G_{\alpha}(t) |v(x+t) - v(y+t)|^2 dt}{h^2}\right) \quad \text{and} \quad C(x) = \int_{\Omega} w(x, y) dy.$$

Here  $G_{\alpha}(t)$  is a Gaussian *convolution kernel* with standard deviation  $\alpha$ , and  $h$  acts as a filtering parameter. If we denote  $D(z) = e^{-z^2}$  (a *kernel function* used to define weights  $w$ ), then the weight  $w(x, y)$  is computed as  $D\left(\frac{\sqrt{\int_{\mathbb{R}^2} G_{\alpha}(t) |v(x+t) - v(y+t)|^2 dt}}{h}\right)$ .

In practice, the integration is substituted by summation, and to save the computation cost, two local windows are employed in the implementation. One is the similarity window  $N(0)$ , a square neighborhood of the origin, that substitutes  $\mathbb{R}^2$  in the definition of weight  $w$ . The other one is search window  $S(x)$  that substitutes the whole image domain  $\Omega$  in the weighted average equation (1). Since enlarging the search domain beyond a certain size will not guarantee a dramatic improvement of performance but at the cost of computation, the search window is fixed as  $15 \times 15$  in practice [11].

To remove excessive noise, a high  $h$  value is usually used in NL-means, but it ends up with loss of fine features especially edges, leading to blurry boundaries or loss of contrast. To solve these problems, instead of averaging over the whole search window as in the original NL-means, we only average over the same phase. More specifically, as illustrated on the right of FIGURE 1, we call the blue object in the middle one phase (corresponding to level set function  $\phi < 0$  [24]) and the white background another phase (level set function  $\phi > 0$ ). Suppose  $A$  is a pixel in the background with excessive noise (corresponding to phase  $\phi > 0$ ) and is close to the border  $\phi = 0$ . Our proposed method removes noise at  $A$  by only using points like  $B$  located in the same phase as  $A$ , but not  $C$  or  $D$  which are in different phases and whose intensities are very different than that of  $A$ . Contributions of  $C, D$  to denoising at  $A$  lead to loss of contrast there. In comparison, in the original NL-means, even though the neighborhoods of  $A$  and  $C$  are not that close, but a larger  $h$  value brings  $\|v(A) - v(C)\|_{2,a}^2/h^2$  closer to 0, causing weight  $w(A, C)$  to be large. As a result, pixel  $C$  contributes significantly to the smoothing of  $A$ , causing decrease of contrast at  $A$ .

The paper has the following four main contributions:

- Present segmentation boosted NL-means (SNL-means) that utilized segmentation to remove excessive additive Gaussian noise while preserving contrast. Automatic selection of parameter is also addressed for piecewise constant and smooth images respectively. To reduce the computational burden, block-wise implementation of SNL-means is discussed.
- Develop transform based NL-means (TNL-means) that preserves more image geometry when segmentation is not reliable due to more excessive noise.

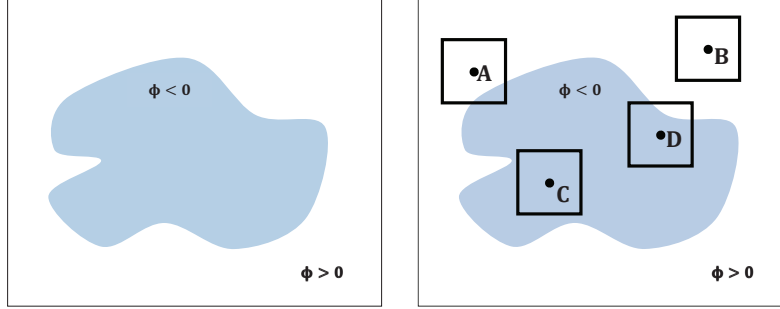


FIGURE 1. Illustration of the motivation of the proposed method, where a level set function  $\phi$  is used to represent phases.

- Provide mathematical validation to block-wise SNL-means from Bayesian perspective and pixel-wise SNL-means from nonparametric regression perspectives. Selection of convolution kernel and weight function is also addressed in detail.
- Generalize SNL-means to Poisson, Rician and Ultrasound noise removal.

The rest of the paper is organized as follows. Section 2 develops pixel-wise and block-wise SNL-means filter which takes full advantage of segmentation with automatic filtering parameter configuration. TNL-means is also presented to substitute SNL-means when the noise is too excessive to trust segmentation out of the noisy observation or its pre-smoothed variants. Section 3 focuses on mathematical justification of SNL-means in terms of Bayesian inference and nonparametric regression. To address other types of noise including Poisson noise, Rician noise and multiplicative noise, several variants of our SNL-means are stated in Section 4. The numerical implementation and more results are shown in Section 5, followed by conclusion in Section 6.

**2. Proposed Scheme.** In this section, as done similarly in the original NL-means, we assume that the observed noisy image  $v$  is related to the underlying clean image  $u$  by:  $v = u + n$  with  $n$  independent Gaussian noise. Sections 2.1 and 2.2 explain how to incorporate segmentation and geometry extraction transform respectively to remove excessive noise while preserving geometries. Generalization to removal of other noise types will be presented in Section 4.

### 2.1. Segmentation Boosted NL-means filter (SNL-means).

**2.1.1. Mutual Position Function.** We introduce the mutual position function  $m(x, y)$  to describe whether pixels  $x$  and  $y$  lie in the same phase as follows:

$$m(x, y) = \begin{cases} 1, & x, y \text{ lie in the same phase} \\ 0, & \text{o.w.} \end{cases}.$$

Phase information can be obtained from image segmentation. Since segmentation is not the focus of this paper, we just apply Mumford-Shah segmentation method [1]. We use Chan-Vese method [25], which is a level set implementation of Mumford-Shah and works well for piecewise constant images. If there are only two phases in an image, one level set function  $\phi$  is enough to describe them:  $\{x | \phi(x) > 0\}$  and

$\{x|\phi(x) < 0\}$  while  $\{x|\phi(x) = 0\}$  is the object contour. When there are four phases, two level set functions  $\phi_1, \phi_2$  are needed [37]. For instance,  $\{x|\phi_1(x) > 0, \phi_2(x) > 0\}$  represents one phase. By the four color theorem from graph theory, two level set functions suffice to distinguish all objects in an image. Let  $H(t)$  be the Heaviside function defined as

$$H(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases},$$

then we could define  $m(x, y)$  as below:

$$m(x, y) = \begin{cases} H(\phi(x)\phi(y)), & 2 \text{ phases} \\ H(\phi_1(x)\phi_1(y))H(\phi_2(x)\phi_2(y)), & \geq 3 \text{ phases} \end{cases}$$

where  $\phi(x)$  is the level set function obtained by the 2 phase Chan-Vese segmentation model and  $\phi_1, \phi_2$  are obtained by the multi-phase Chan-Vese segmentation model.

2.1.2. *SNL-means.* With the mutual position function  $m(x, y)$ , we define a new weight as

$$w(x, y) = \frac{1}{C(x)} \exp\left(-\frac{1}{h^2} \|v(N(x)) - v(N(y))\|_{2,\alpha}^2\right) \cdot m(x, y)$$

where the normalizing constant

$$C(x) = \sum_{y \in S_K(x)} \exp\left(-\frac{1}{h^2} \|v(N(x)) - v(N(y))\|_{2,\alpha}^2\right).$$

Here  $v(N(x))$  represents the observed image patch in similarity window  $N(x)$  centered at pixel  $x$  and  $\|\cdot\|_{2,\alpha}$  represents the Euclidean distance weighted by Gaussian convolution kernel  $G_\alpha$ . The segmentation boosted nonlocal means filter is then defined as:

$$SNL(v)(x) = \sum_{y \in S_K(x)} w(x, y)v(y) \quad (2)$$

$S_K(x)$  is a spatial neighborhood of  $x$  that contains at least  $K$  pixels in the same phase as  $x$ . For a fixed pixel  $x$  near the boundary,  $m(x, y)$  tends to be 0 for some pixel  $y$ , leading to insufficient number of contributing pixels if the size of the search window  $S(x)$  is fixed (nonadaptive). To ensure similar number of contributing  $y$ 's in search window of every  $x$ , we look for an adaptive window  $S_K(x)$  as follows. At first we define the candidate based on the segmentation and square windows:

$$S^k(x) = \{y \in \Omega \mid m(x, y) = 1, \|x - y\|_\infty \leq k\}$$

where

$$\|x\|_\infty = \max\{|x_1|, |x_2|\}$$

and  $k$  corresponds to the radius of square windows. To ensure comparing at least  $K$  pixels, we set our adaptive search window  $S_K(x)$  as  $S^l(x)$  where

$$l = \operatorname{argmin}\{k > 0 \mid |S^k(x)| \geq K\}$$

and  $|S^k(x)|$  the cardinality of  $S^k(x)$ . If the phase containing  $x$  has fewer pixels than  $K$ , we define this whole phase as  $S_K(x)$ .

2.1.3. *Filtering Parameter  $h$ .* Selection of the filtering parameter  $h$  plays a vital role in the proposed SNL-means algorithm. Here we describe how to automatically select the parameter  $h$  for piecewise constant images and piecewise smooth images separately.

- Piecewise Constant Images

For the sake of simplicity, suppose the underlying clean image  $u : \Omega \rightarrow \mathbb{R}^2$  is piecewise constant with 2 phases such that one level set function  $\phi$  is enough to describe them. Let  $\Omega_1 = \{x \in \Omega \mid \phi(x) > 0\}$  and  $\Omega_2 = \{x \in \Omega \mid \phi(x) < 0\}$ ,

$$u(x) = \begin{cases} c_1, & x \in \Omega_1 \\ c_2, & x \in \Omega_2 \end{cases},$$

where  $c_1, c_2$  are constants. Under the assumption of additive noise, we have the observed image

$$v(x) = c_i + n_i(x), \quad x \in \Omega_i, \quad i = 1, 2$$

where  $n_i(x)$  are independent and identically distributed (i.i.d.) within  $\Omega_i$ ,  $i = 1, 2$ . Using the uniformly minimum variance unbiased (UMVU) estimator [26], we get the best estimator for noise variance in each phase as

$$\text{Var}(n_i) = \frac{1}{|\Omega_i| - 1} \sum_{x \in \Omega_i} (v(x) - c_i)^2, \quad i = 1, 2.$$

Then the filtering parameter  $h_i$  used in the  $i$ -th phase ( $i = 1, 2$ ) is chosen as  $h_i = \sqrt{\text{Var}(n_i)}$ .

In the multi-phase case, we still employ the unbiased estimation of the noise variance in each phase to obtain the phase-wise filtering parameter  $h$ 's.

- Piecewise Smooth Images

Let the underlying image  $u(x)$  be piecewise smooth with independent and spatially variant noise  $n(x)$  where  $n(x)$  has mean zero and variance  $\sigma^2(x)$ . By the assumption, the mean and variance for the conditional probability are as below:

$$E(v(x)|u(x)) = u(x), \quad \text{Var}(v(x)|u(x)) = \sigma^2(x).$$

In this case, there are no closed forms for the mean and variance and thereby we employ the nonparametric dual model to estimate the noise level locally [27]: at pixel  $x$ , we use the nearest four-pixel neighboring set, denoted by  $N_4(x)$ , as its sample space and obtain the Gasser-Sroka-Jennen-Steinmetz pseudo-residual

$$\epsilon(x) = \sqrt{\frac{4}{5}} \left( \frac{1}{4} \sum_{y \in N_4(x)} v(y) - v(x) \right)$$

which guarantees that  $E(\epsilon(x)^2) = \sigma^2(x)$ . By taking the average to correct the bias, we define the unbiased local variance estimator as:

$$V(x) = \frac{K}{K-1} \left\{ \left( \frac{1}{K} \sum_{y \in N(x)} \epsilon^2(y) \right) - \left( \frac{1}{K} \sum_{y \in N(x)} \epsilon(y) \right)^2 \right\}$$

where  $N(x)$  is the nearest  $K$ -pixel neighboring set of pixel  $x$ . Finally,  $h^2(x) = \beta V(x)$  where  $\beta$  is between 0 and 1 and is set as .5 in our experiments and the results are not sensitive to the choice of  $\beta$ .

The key ingredient in the proposed method SNL-means is the mutual position function defined from segmentation. The improvement over the original NL-means is significant when the segmentation is accurate. In FIGURE 2, we compare the results of NL-means and SNL-means (fed with ground truth segmentation) on a noisy MRI brain image with intensity range  $[0, 230]$  and additive zero-mean Gaussian noise with 10% ( $\sigma = 23$ ) noise. It is observed that the SNL-means leads to sharper reconstructed image or anatomically clearer separation of different phases (gray matter, white matter and cerebral spinal fluid).

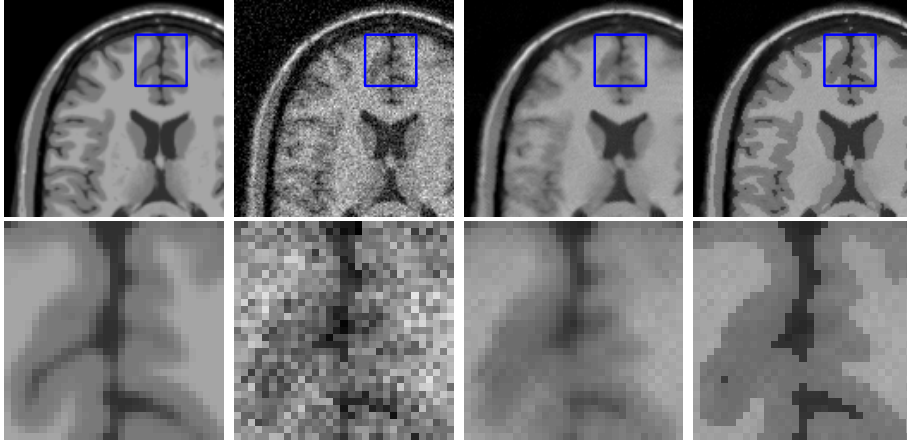


FIGURE 2. Comparison of NL-means and SNL-means. First row from left to right: clean image with intensity  $[0, 230]$ , noisy image with 10% noise  $\sigma = 23$ , result by NL-means filter, proposed method with the ideal partition. Second row is the amplified section corresponding to the first row.

**2.2. Transform based NL-means filter.** In practice, one does not have the exact segmentation of the ground truth but only the noisy observation. When the observed noisy image has excessive or spatially variant noise, it is more difficult to distinguish different phases. As the noise level grows, image segmentation gets more and more challenging: fine features are usually missed and some artificial edges are created. In this case, we have the following three options:

- **Pre-smooth.** Pre-smooth the noisy image to get a cleaner image, on which we apply segmentation. This approach is the easiest and works only when the noise level is low. But the segmentation depends on the selection of the pre-smooth approach and is not accurate when noise is excessive. Most of the existing denoising approaches tend to lose salient features while removing excessive noise. Loss of salient features in the pre-smoothed image causes segmentation errors.
- **Iterative denoising.** Alternately segment and smooth images using SNL-means in an iterative way: apply segmentation on the noisy observation, then apply segmentation boosted NL-means on the original noisy image to get one cleaner image, which is segmented again for better phase information that in the next round helps remove noise from the original noisy image. Repeat

the procedure until the adjacent smoothing results are close. This method is intuitive but expensive due to iterations.

- **Geometrical structure extraction transform.** Substitute segmentation by some more general transform that is less sensitive to noise but detect more intrinsic geometric structures. Then replace mutual position function defined on segmentation by some new function (again part of the weight definition) describing geometric structure similarity. Segmentation only detects edges, while more general geometric structure extraction transforms detect more geometries including edges. In what follows, we explain this method in detail.

The idea is to apply certain transform catching the main features of the image and to reduce the impact of noise during the restoration step. We demonstrate the idea using contourlet transform, but any other general transform could be used. Contourlet is a pyramidal directional transform which represents  $C^2$  natural images efficiently using contour segments [28]. Due to the contour-like edges in many real world images, the contourlet transform is appropriate to be incorporated into the computation of the local weight. The proposed transform based NL-means (TNL-means) scheme is

$$TNL(v)(x) = \sum_y w(x, y)v(y), \quad (3)$$

with weight  $w(x, y)$  the product of weighted intensity similarity and geometry similarity defined as below

$$w(x, y) = \frac{1}{C(x)} \exp\left(-\frac{\eta_1}{h_1^2} \|v(N(x)) - v(N(y))\|_{2,\alpha}^2 - \frac{\eta_2}{h_2^2} \|\mathcal{T}(v)(N(x)) - \mathcal{T}(v)(N(y))\|_{2,\beta}^2\right)$$

where  $C(x)$  is the normalization factor,  $\eta_1, \eta_2$  are balancing constants and  $h_1, h_2$  are respective filtering parameters. In our later experiments,  $h_1$  and  $h_2$  are set as the standard deviation of the noise while  $\eta_1, \eta_2$  are adjustable values within  $(0, 1)$ . Here  $\mathcal{T}$  is designated as the combination of three steps: contourlet transform, thresholding in the transform domain, and inverse contourlet transform back to the image domain.

In FIGURE 3 and TABLE 1 we demonstrate that TNL-means outperforms thresholding based contourlet approach and NL-means respectively. We add the same additive Gaussian noise with  $\sigma = 23$  as in FIGURE 2 to the clean MRI brain image. We can see in FIGURE 3 that the contourlet restored image after thresholding in the transform domain captures certain geometries of the original image while having much noise remained (middle left). The proposed TNL-means method takes advantage of the geometric information to build the weight and removes the noise more efficiently (right two images). In TABLE 1, the comparisons with NL-means for three different noise levels 10% ( $\sigma = 24.6$ ), 15% ( $\sigma = 36.9$ ), 20% ( $\sigma = 49.3$ ) are listed. We use three quantitative measures: peak-to-noise ratio (PSNR), root of mean squared error (RMSE) and structural similarity (SSIM) [29]. It can be seen that the proposed method consistently gives the highest PSNR, lowest RMSE and the highest SSIM. The higher the noise level, the more the TNL-means outperforms.

In practice, depending on the structure of the image, contourlet transform based weight may not be the most appropriate candidate, and other filter bank based transforms may be more applicable, e.g. curvelet transform [20] and shearlet transform [21].

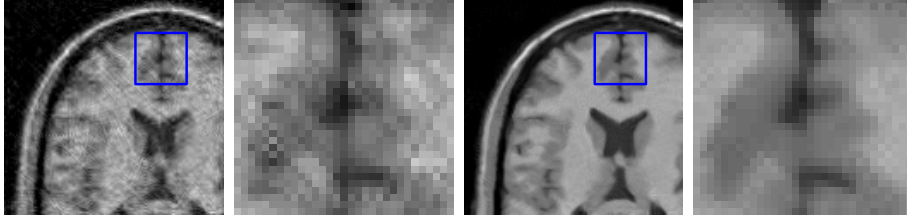


FIGURE 3. Comparison of thresholding based contourlet approach and contourlet based NL-means. From left to right:  $\mathcal{T}(v)$  based on contourlet transform with thresholding, the amplified section of  $\mathcal{T}(v)$ , recovered image by our proposed method  $\eta_1 = 0.8, \eta_2 = 2, h_1 = h_2 = \sigma$  and the corresponding amplified section.

TABLE 1. Quantitative comparison of NL-means and TNL-means on denoising of a T1 brain image

$\sigma$ of noise	24.6			36.9			49.3		
Method	Noisy	NLM	Ours	Noisy	NLM	Ours	Noisy	NLM	Ours
PSNR	20.29	29.07	29.68	16.80	26.70	27.45	14.42	24.67	25.64
RMSE	24.66	8.98	8.37	36.85	11.79	10.82	48.47	14.89	13.32
SSIM	0.43	0.84	0.85	0.30	0.74	0.77	0.22	0.66	0.71

**2.3. Block-wise SNL-means.** In this subsection, we present the block-wise SNL-means filter. For natural images with large size, pixel-wise SNL-means filtering will cost too much time on comparing the weighted patches. To lower the computation burden, we follow the idea in [22] to develop block-wise SNL-means. The idea is to use SNL-means to obtain estimate of the clean image in a *block* by taking the weighted average of the most similar blocks and then get the recovered intensity at a specific *pixel*  $x$  by averaging over all the estimators of intensities at  $x$  given by all the blocks containing  $x$ . Rather than choosing all the blocks in the search window, we requires the centers of the contributing blocks to lie in the same phase as that of the target block.

Specifically, we first partition the whole image domain  $\Omega$  into overlapping blocks  $\{B_i\}_{i=1}^k$  satisfying  $\Omega = \bigcup_{i=1}^k B_i$ . Then we combine the information from segmentation to the nonlocal means formula:

$$\hat{u}(B_i) = \sum_{j=1}^k w(B_i, B_j) v(B_j) \quad (4)$$

where

$$w(B_i, B_j) = \frac{1}{C(B_i)} M(B_i, B_j) \exp\left(-\frac{\|v(B_i) - v(B_j)\|_{2,\alpha}^2}{h^2}\right)$$

and  $C(B_i)$  is a normalizing constant. Here  $M(B_i, B_j)$  is a block-wise mutual position function and for simplicity we only apply the centers of the two blocks of interest:

$$M(B_i, B_j) = m(x_i, y_j)$$

where  $x_i$  and  $y_j$  are the center pixels of blocks  $B_i$  and  $B_j$ .



Let  $\mathcal{B}(x) = \{B_i | x \in B_i \subset \Omega\}$  be the set of all the blocks including  $x$ , the estimated intensity at each pixel is calculated as below:

$$\hat{u}(x) = \frac{\sum_{B(x)} \hat{u}(B_j(x))}{|\mathcal{B}(x)|}. \quad (5)$$

However, there is a trade-off between the denoising performance and the computation complexity. Block-wise implementation will lower the computation time tremendously but at the cost of the quality which will be shown in Section 5 TABLE 7.

**3. Mathematical validations.** In this section, we provide mathematical validations, in more general setup, of the proposed block-wise SNL-means from Bayesian perspective (Section 3.1) and the pixel-wise SNL-means version using non-parametric regression (Section 3.2). Based on the validations, we generalize the proposed scheme to deal with images with other noise types, e.g., Poisson, Rician and Ultra-sound noise in Section 4.

**3.1. Bayesian Interpretation of Block-wise SNL-means.** In [14], Kervrann et al. derived the NL-means filter in perspective of Bayesian framework which states the relationship between the weight function in NL-means filter and the probability density function (pdf) of the white Gaussian noise. We extend their derivation to general additive noise. Various local convolution kernels relating the structural prior of images are also considered.

As before, we assume  $v = u + n$  where  $n$  is any general independent noise and  $u$  is piecewisely constant or smooth. Let  $\{\Omega_i\}_i$  be a partition of the image domain  $\Omega$  such that

$$u(x) = \sum_i u_i(x), \quad x \in \Omega$$

where

$$u_i(x) = u(x)|_{\Omega_i}, \quad u_i(x)|_{\Omega_i^c} = 0.$$

For every  $x \in \Omega$ , we denote the vectorized  $u$  and  $v$  taken in a neighborhood of  $x$  (similarity window  $N(x)$  of size  $\sqrt{k} \times \sqrt{k}$ ) by

$$\mathbf{u}(x) = \begin{bmatrix} u(x_1) \\ \vdots \\ u(x_k) \end{bmatrix} \quad \mathbf{v}(x) = \begin{bmatrix} v(x_1) \\ \vdots \\ v(x_k) \end{bmatrix}$$

Let  $\hat{\mathbf{u}}$  be an estimator for  $\mathbf{u}$ , and define the expectation of loss associated with this estimator as

$$E(L(\mathbf{u}(x), \hat{\mathbf{u}}(x))) = \sum_{\mathbf{u}(x) \in \Lambda} L(\mathbf{u}(x), \hat{\mathbf{u}}(x)) p(\mathbf{u}(x) | \mathbf{v}(x)).$$

where  $\Lambda \subseteq \mathbb{R}^k$  is the range of  $\mathbf{u}(x)$ .

By choosing  $L(\mathbf{u}(x), \hat{\mathbf{u}}(x)) = |\mathbf{u}(x) - \hat{\mathbf{u}}(x)|^2$  and letting  $p$  be a probability, the optimal estimator is computed as below:

$$\begin{aligned} \mathbf{u}_{opt}(x) &= \operatorname{argmin}_{\hat{\mathbf{u}}} E(L(\mathbf{u}(x), \hat{\mathbf{u}}(x))) \\ &= \sum_{\mathbf{u}(x) \in \Lambda} \mathbf{u}(x) p(\mathbf{u}(x) | \mathbf{v}(x)) \\ &= \sum_{\mathbf{u}(x) \in \Lambda} \frac{\mathbf{u}(x) p(\mathbf{u}(x), \mathbf{v}(x))}{p(\mathbf{v}(x))} \\ &= \sum_{\mathbf{u}(x) \in \Lambda} \frac{\mathbf{u}(x) p(\mathbf{v}(x) | \mathbf{u}(x)) p(\mathbf{u}(x))}{\sum_{\mathbf{u}(x) \in \Lambda} p(\mathbf{v}(x) | \mathbf{u}(x)) p(\mathbf{u}(x))} \end{aligned}$$

Let  $S_K(x)$  be the neighborhood of  $x$  defined as in Section 2.1.2. By the assumption that patches with centers within the same phase follow the same distribution, the neighboring patches with centers in  $S_K(x)$  can be considered as the realizations of the patch  $\mathbf{u}(x)$ . Assuming  $\mathbf{u}(x)$  is uniformly distributed in  $\{\mathbf{u}(y) | y \in S_K(x)\}$  which replaces  $\Lambda$ , i.e., there is no preference to choose one neighboring patch from another, we have

$$p(\mathbf{u}(x)) = \frac{1}{|S_K(x)|}.$$

As  $K \rightarrow \infty$  and thereby the number of the neighboring patches increases, the law of large numbers asserts:

$$\begin{aligned} \sum_{y \in S_K(x)} \mathbf{u}(y) p(\mathbf{v}(x) | \mathbf{u}(y)) \frac{1}{|S_K(x)|} &\rightarrow \sum_{\mathbf{u}(x) \in \Lambda} \mathbf{u}(x) p(\mathbf{v}(x) | \mathbf{u}(x)) p(\mathbf{u}(x)) \\ \sum_{y \in S_K(x)} p(\mathbf{v}(x) | \mathbf{u}(y)) \frac{1}{|S_K(x)|} &\rightarrow \sum_{\mathbf{u}(x) \in \Lambda} p(\mathbf{v}(x) | \mathbf{u}(x)) p(\mathbf{u}(x)). \end{aligned}$$

Then one gets an approximated  $\mathbf{u}_{opt}$  as below:

$$\mathbf{u}_{opt}(x) = \sum_{y \in S_K(x)} \frac{\mathbf{u}(y) p(\mathbf{v}(x) | \mathbf{u}(y))}{\sum_{y \in S_K(x)} p(\mathbf{v}(x) | \mathbf{u}(y))}.$$

Considering  $\mathbf{v}(y)$  as the realization of  $\mathbf{u}(y)$ , we get an estimation of  $u$  in the similarity window  $N(x)$

$$\hat{\mathbf{u}}(x) = \sum_{y \in S_K(x)} \frac{p(\mathbf{v}(x) | \mathbf{v}(y))}{\sum_{y \in S_K(x)} p(\mathbf{v}(x) | \mathbf{v}(y))} \mathbf{v}(y). \quad (6)$$

Next, we assume  $\mathbf{n}$  is a Gaussian random field and  $\mathbf{n}(x)$  is a multivariate Gaussian random variable therein. Note that

$$\mathbf{v}(x) = \mathbf{v}(y) + \mathbf{n}(x, y), \quad \text{with} \quad \mathbf{n}(x, y) = \mathbf{u}(x) - \mathbf{u}(y) + \mathbf{n}(x) - \mathbf{n}(y), \quad (y \neq x).$$

If  $x$  and  $y$  are very close to each other within the same phase, then  $\mathbf{u}(x) \approx \mathbf{u}(y)$ . In addition, to embody certain regularity of the original image between  $\mathbf{u}(x)$  and  $\mathbf{u}(y)$ , we assume that  $\mathbf{n}(x, y) = R^{-1}E$  where  $E \sim \mathcal{N}(0, \sigma^2 I)$  and  $R \in \mathbb{R}^{k \times k}$  is symmetric positive definite. Thus we get

$$p(\mathbf{v}(x) | \mathbf{v}(y)) \propto \exp \left( -\frac{1}{2\sigma^2} (\mathbf{v}(x) - \mathbf{v}(y))^T R^T R (\mathbf{v}(x) - \mathbf{v}(y)) \right).$$

It's obvious that  $\text{Var}(\mathbf{n}(x, y)) = (R^T R)^{-1}$  and  $\|R(\mathbf{v}(x) - \mathbf{v}(y))\|_2$  is a semi-metric measuring the closeness of two image patches. In the original NL-means,  $R^T R$  is chosen as a square matrix

$$R^T R = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ & & \cdots & & \\ 0 & 0 & \cdots & 0 & d_k \end{bmatrix}$$

whose diagonal is the vectorized 2D Gaussian convolution kernel  $G_\alpha$  without taking phase information into account. In this case, equation (6) leads to the same formula as (5) in Section 2.3. The diagonal entries in  $R^T R$  could come from Gaussian convolution kernels or more generally ones like rectangular disk, circular disk, average (constant) as shown in FIGURE 4. We have tested some piecewise constant images and piecewise smooth images using the listed kernels and found that for piecewise constant images, flatter kernels enhance denoising performance. However for most images with more intensity variations, more exponential-like kernels are preferred. The denoising assessment for each kernel in terms of PSNR for a 2-phase toy image and a T1 MRI image are shown in TABLE 2.

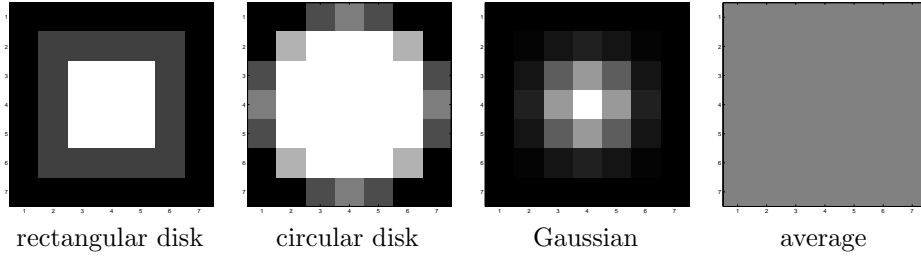
FIGURE 4.  $7 \times 7$  discretized convolution kernels

TABLE 2. PSNR comparison for convolution kernels

	rectangular disk	circular disk	Gaussian	average
Toy image	36.60	37.12	37.34	33.45
T1 MRI	29.14	28.93	28.52	27.89

### 3.2. Nonparametric Regression Understanding of Pixel-wise SNL-means.

Next we understand the proposed pixel-wise SNL-means filter in terms of local smooth regression. Define adaptive search window  $S_K(x_l)$  for a fixed pixel  $x_l$  as in Section 2.1.2. To get an estimate of the clean image intensity at  $x_l$ , we use sample pairs composed of intensities at each pixel in the search window and those in their similarity windows, i.e., we use sample pairs  $\{(v(x_i), Y_i)\}$  where  $v(x_i) \in \mathbb{R}$ ,  $1 \leq i \leq K$  with  $x_i \in S_K(x_l)$ , and  $Y_i \in \mathbb{R}^{k-1}$  the observed column-wise stacked intensity vector of the neighbors (similarity window of size  $\sqrt{k} \times \sqrt{k}$ ) of pixel  $x_i$  except at  $x_i$ . The Nadaraya-Watson kernel-weighted average [30] provides an estimate of  $u$  at  $x_l$  as

$$\hat{u}(x_l|Y_l) = \frac{\sum_{j: x_j \in S_K(x_l)} f(Y_l, Y_j) v(x_j)}{\sum_j f(Y_l, Y_j)}$$

where the *weight function*

$$f(Y_l, Y_j) = D\left(\frac{\|R(Y_l - Y_j)\|_2}{h}\right).$$

Here  $R$  is a diagonal matrix defined as in the previous section and  $D(\cdot)$  is a *kernel function* satisfying the conditions:

- (i) positivity:  $D(t) \geq 0$
- (ii) decay as distance increases:  $\lim_{t \rightarrow \infty} D(t) = 0$  and  $\lim_{t \rightarrow 0} D(t) = 1$ .
- (iii) unit volume:  $\int_0^\infty D(t) dt = 1$

Denoting

$$w_{lj} = \frac{f(Y_l, Y_j)}{\sum_{j: x_j \in S_K(x_l)} f(Y_l, Y_j)} = \frac{D(\|R(Y_l - Y_j)\|_2/h)}{\sum_{j: x_j \in S_K(x_l)} D(\|R(Y_l - Y_j)\|_2/h)},$$

the above local linear regression can be actually obtained by solving a weighted least squares problem:

$$\min_u \sum_l \left( u(x_l) - \sum_{j: x_j \in S_K(x_l)} w_{lj} v(x_j) \right)^2$$

which is the discretization of

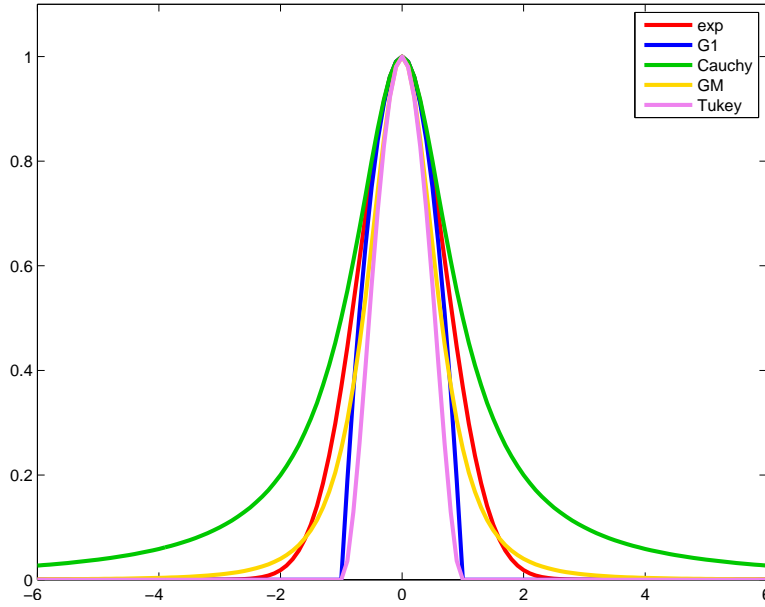
$$\min_u \int_\Omega \left( u(x) - \int_{S_K(x)} w_v(x, y) v(y) dy \right)^2 dx.$$

In the original NL-means algorithm, the weight function is defined through selecting  $D(t)$  as exponential function  $e^{-t^2}$  and  $R$  is based on discretized Gaussian convolution kernel. But different combinations of kernel function  $D$  and convolution kernel lead to different weight functions. Selection of convolution kernel has been addressed in the previous section. Next, we focus on selection of kernel function  $D$  on removing additive Gaussian noise. TABLE 3 lists five kernel functions  $D$  without normalizing constant and FIGURE 5 shows their respective graphs. Note  $D$  applied on  $\frac{\|R(Y_l - Y_j)\|_2}{h}$  gives weights before normalization. *For a fixed  $h$ , the principle of selecting  $D$  is as follows: for piecewise constant images, choose  $D$  that is flatter and more spread out, e.g. Cauchy; while for images with large intensity variation in a small neighborhood, choose  $D$  that is more steep in the range of  $\frac{\|R(Y_l - Y_j)\|_2}{h}$ , e.g., exponential one if the range is inside  $[1, 1.5]$ .* The reason is that for piecewise constant images, the intensity is close to one constant for all pixels in the same phase. Combining with the fact that larger number of contributing pixels leads to a more robust estimate (for i.i.d. random variables  $r_i$ 's, the variance of  $\sum_{i=1}^N r_i/N$  approaches to 0 as  $N$  goes to infinity), we expect most of the weights to be nonzero and close to each other, which can be done by choosing a kernel function  $D$  that is flat. From the graphs in FIGURE 5, it is obvious that Cauchy function is the flattest and is thus the best choice for piecewise constant images. On the other hand for images with various intensities, we only expect pixels with close intensities to contribute more. In terms of weights, we only expect those with smaller  $\|R(Y_l - Y_j)\|_2$  ( $h$  is fixed) to get larger values. Therefore, a kernel that is steep in the range of  $\frac{\|R(Y_l - Y_j)\|_2}{h}$  is preferred.

We test a piecewise constant toy image ( $h$  fixed as  $\sigma = 18.2$ ) and a T1 MRI image ( $h$  fixed as  $\sigma = 25.5$ ) with different weight functions (fixing the rectangular disk as  $R$ ), and we list the results as in TABLE 4. For the piecewise constant toy image, Cauchy function shows significantly better performance than the others. While for

TABLE 3. Kernel functions  $D(t)$ 

exponential	$\exp(-t^2)$
$G_1$	$\begin{cases} 1 - t^2 & \text{if }  t  \leq 1 \\ 0 & \text{if }  t  > 1 \end{cases}$
Cauchy	$1/(1 + t^2)$
Geman-McClure[15]	$1/((1 + t^2)^2)$
Tukey	$\begin{cases} (1 - t^2)^2 & \text{if }  t  \leq 1 \\ 0 & \text{if }  t  > 1 \end{cases}$

FIGURE 5. Comparison of various kernel function  $D$ .

the T1 MRI image that has more intensity variations, the range of  $\frac{\|R(Y_i - Y_j)\|_2}{h}$  is inside  $[1, 1.5]$ , so as analyzed above, the result using exponential function is the best. Furthermore, the similar conclusion will be obtained in the case of Rician noise with the same test images.

TABLE 4. PSNR comparison for various kernel functions.

	$h$	exponential	$G_1$	Cauchy	Geman-McClure	Tukey
Toy image	$\sigma$	36.71	23.37	<b>45.01</b>	41.94	23.33
T1 MRI	$\sigma$	<b>29.25</b>	20.30	24.63	26.65	20.28

**4. Generalization to other noise types.** Section 2 addresses in detail how SNL-means removes Gaussian noise automatically. To deal with non-Gaussian noise, we provide two options here. One is to insert a specific noise type into the generic

Bayesian formula (6) in Section 3.1 and see whether a closed form is computable. For some distributions (e.g., Poisson, binomial and negative binomial), another option is to transform the given noise to i.i.d. Gaussian noise, then apply the SNL-means scheme thoroughly discussed in Section 2 to remove the additive Gaussian noise, and finally transform the processed image back to the original image space. But special care must be taken to correct the bias resulting from the inverse transform. As some specific examples, we explain in detail how to use the proposed scheme to remove Poisson noise (Section 4.1) in computed tomography (CT), Rician noise (Section 4.2) in magnetic resonance imaging (MRI), and a speckle noise (Section 4.3) in ultrasound imaging.

**4.1. Poisson noise.** In this section, we apply our proposed method on removing Poisson noise. Suppose the observed image  $v(x)$  follows Poisson distribution with mean as the underlying image  $u(x)$ , i.e.

$$P(v(x) = k | u(x)) = \frac{e^{-u(x)} (u(x))^k}{k!}.$$

By virtue of Anscombe transform [31], i.e.

$$\mathcal{A} : u \rightarrow 2\sqrt{u + \frac{3}{8}},$$

the image contaminated by Poisson noise is transformed to an image with approximately homogeneous additive Gaussian noise. In the transformed domain, our proposed method (Section 2.1) with estimation of the unknown additive noise level (Section 2.1.3) can be implemented. To minimize the undesired bias generated by the transformation, the asymptotic unbiased inverse transform is used

$$\tilde{\mathcal{A}}^{-1} : u \rightarrow \left(\frac{u}{2}\right)^2 - \frac{1}{8}.$$

In a word, our proposed denoising scheme on Poisson noise removal is

$$\hat{u} = \tilde{\mathcal{A}}^{-1}(\text{SNL}(\mathcal{A}(v))). \quad (7)$$

**4.2. Rician noise.** In practice, images may be complex valued and medical technicians visualize the magnitude of the images. If the noise is well modeled by a Gaussian probability density function in the real and imaginary parts of the complex data respectively, then the noise in the associated magnitude image can be modeled as Rician noise. This noise model is used in a wide variety of data, such as MRI, SAR [32]. Let  $v \sim \mathcal{R}(u, \sigma)$  be the observed magnitude image obeying Rician distribution with parameters  $u \geq 0$  and  $\sigma \geq 0$ ,

$$p(v|u, \sigma) = \frac{v}{\sigma^2} e^{-\frac{v^2+u^2}{2\sigma^2}} J_0\left(\frac{zu}{\sigma^2}\right), \quad v \geq 0,$$

where  $J_n$  represents the modified Bessel function of order  $n$ ,

$$J_n(x) = \sum_{m=0}^{\infty} \frac{(\frac{x}{2})^{n+2m}}{m! \Gamma(n+m+1)}.$$

This distribution reduces to a Rayleigh distribution if  $u = 0$ . From the expression of  $J_n(x)$ , it's easy to see that

$$J_0(x) = \sum_{m=0}^{\infty} \frac{(x/2)^{2m}}{(m!)^2}, \quad J_1(x) = \sum_{m=0}^{\infty} \frac{(x/2)^{2m+1}}{m!(m+1)!}.$$

The mean and variance of  $v$  therefore are, respectively,

$$\begin{aligned} E(v|u, \sigma) &= \sigma \sqrt{\frac{\pi}{2}} F\left(-\frac{u^2}{2\sigma^2}\right) \\ \text{Var}(v|u, \sigma) &= 2\sigma^2 + u^2 - \frac{\pi\sigma^2}{2} F^2\left(-\frac{u^2}{2\sigma^2}\right). \end{aligned}$$

where

$$F(x) = e^{\frac{x}{2}} \left[ (1-x)J_0\left(-\frac{x}{2}\right) - xJ_1\left(-\frac{x}{2}\right) \right].$$

The second moment is therefore

$$E(v^2|u, \sigma) = \text{Var}(v|u, \sigma) + (E(v|u, \sigma))^2 = 2\sigma^2 + u^2,$$

implying a denoising scheme for remove Rician noise. Based on the observation that

$$E\left(\sum_{y \in S_K(x)} w(x, y) v(y)^2 | u, \sigma\right) = 2\sigma^2 + u(x)^2, \quad \sum_{y \in S_K(x)} w(x, y) = 1$$

and the nonlocal estimator proposed by Wiest-Daesslé et al. [19], we present the proposed scheme on Rician noise removal

$$\hat{u}(x) = \sqrt{\max(SNL(v^2)(y) - 2\sigma^2, 0)}. \quad (8)$$

**4.3. Ultrasound noise.** In ultrasound imaging, the observed noisy image  $v$  is modeled as  $v = u + \sqrt{u} \cdot n$  [33, 34] where  $u$  is the underlying clean image, and noise is  $\sqrt{u} \cdot n$  with  $n \sim N(0, \sigma^2)$ , i.e., the noise depends on  $u$ . Then  $v|u \sim \mathcal{N}(u, u\sigma^2)$  i.e.,

$$p(v(x)|u(x)) \propto \exp\left(\frac{-\|v(x) - u(x)\|_2^2}{2u(x)\sigma^2}\right).$$

Following the reasoning in Section 3.1 and plugging the above  $p(v(x)|u(x))$  into the generic Bayesian formula (6), we get

$$\begin{aligned} \hat{u}(x) &= \frac{\sum_{y \in N(x)} m(x, y) p(v(x)|v(y)) v(y)}{\sum_{y \in N(x)} m(x, y) p(v(x)|v(y))} \\ &= \frac{1}{C(x)} \sum_{y \in N(x)} m(x, y) \exp\left(-\frac{\|R(v(N(x)) - v(N(y))) ./ 2v(N(y))\|_2^2}{h^2}\right) v(y) \end{aligned} \quad (9)$$

where  $./$  represents the entry-wise division and  $h$  can be chosen around  $\sigma$ .

**5. More Experimental Results.** Besides those results shown in the previous sections, here we further test our algorithm and its variants on a variety of scalar images and also compare these results with those of the standard NL-means algorithm and the other state-of-the-art denoising methods. Equipped with the geometric information obtained from segmentation, appropriate convolution kernel and weight function based on the image to be denoised, the proposed framework shows the prominence on edge preservation and noise removal. All tests were run under Windows 7 Professional and MATLAB v7.12 (R2011a) on a laptop with an Intel Core 2 Duo CPU at 2.8 GHz and 3 GB of memory.

Both qualitative and quantitative comparison are provided throughout this section. For quantitative comparison, among many other criteria, two assessment

measures are used here. One is peak signal-to-noise ratio (PSNR) which is defined in decibels (dB) as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}$$

where MSE is a mean squared error between the noisy image and the clean image. This is the main criterion used later to compare the denoising performance. Another one is signal to noise ratio (SNR) which quantifies the recovery by comparing between the “ground truth”  $u(x)$  and the reconstructed image  $\hat{u}(x)$ . We adopt the SNR formula in [38].

$$\text{SNR} = 10 \log_{10} \frac{\sum_{x \in \Omega} (u^2(x) + \hat{u}^2(x))}{\sum_{x \in \Omega} (u(x) - \hat{u}(x))^2}.$$

It’s used only for assessing the performance of the ultrasound denoising.

Four sets of tests are done to show the significant improvements of our proposed algorithms. Note that if not specified explicitly, pixel-wise implementation is always performed and the clean images for our tests range in  $[0, 255]$ .

- The first set is to show the advantage of the pixel-wise SNL-means on piecewise constant and smooth image denoising.

We start with a two-phase piecewise constant image containing three shapes contaminated by white Gaussian noise with different known variances. We apply our proposed SNL-means filter (2), the original NL-means algorithm, total variational method (ROF) [3], block-matching and 3D filtering (BM3D) [35] and K-SVD [36] to the same image. Optimal parameters are selected for all. In the proposed method, filtering parameter  $h$  is set as the given noise level, and since it is a piecewise constant image, as discussed in Section 3.2, we replace the exponential weight function used in the original NL-means by the Cauchy weight function.

TABLE 5 compares in terms of PSNR as noise variance changes. It can be seen that across all the noise levels, the proposed method consistently and significantly performs better than all the other methods. FIGURE 6 shows the visual comparisons when the noise level is 50.

TABLE 5. PSNR obtained with the compared filters for the toy image

Noise level $\sigma$	10	20	30	40	50
Noisy	28.06	22.10	18.50	16.15	14.16
NL-means	34.42	32.93	30.17	28.28	26.38
Proposed	<b>44.12</b>	<b>40.50</b>	<b>38.13</b>	<b>33.57</b>	<b>29.61</b>
ROF	35.77	29.95	26.64	24.61	22.92
K-SVD	40.24	34.98	30.96	28.90	27.37
BM3D	42.13	34.89	31.41	29.83	27.52

Next, we test a multi-phase piecewise constant Shepp-Logan phantom image. Multiple phases are obtained by using multi-phase Chan-Vese segmentation model [37]. To fetch more accurate phase information, a pre-smoothed version of the input image is favored for segmentation. Here BM3D is applied for pre-smoothing. We employ our 6-phase SNL-means filter (2) with phase-wisely defined  $h$  (Section 2.1.3). From FIGURE 7 and TABLE 6, one



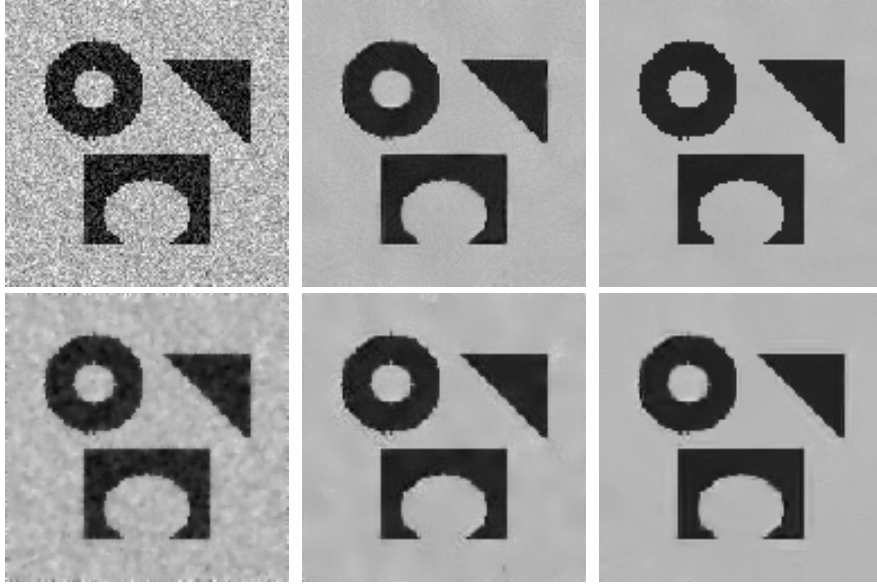


FIGURE 6. Comparison of denoising performance on a two-phase toy image. First row: noisy image with  $\sigma = 30$ ; NL-means; proposed. Second row: ROF; K-SVD; BM3D

can see that the proposed model significantly outperforms the others both quantitatively and qualitatively.

TABLE 6. PSNR of various filters on a multi-phase phantom image.

Noise level $\sigma$	20	30
Noisy	22.18	18.56
NL-means	33.56	31.31
Proposed	<b>38.57</b>	<b>35.59</b>
ROF	30.30	26.95
BM3D	36.34	33.06
K-SVD	33.67	30.83

In FIGURE 8, we test a brain MRI image corrupted by *inhomogeneous* additive Gaussian noise using iterative 3-phase piecewise smooth denoising scheme mentioned in the beginning of Section 2.2. Unlike the NL-means using one universal filtering parameter  $h$ , our proposed method mentioned in Section 2.1.3 estimates the inhomogeneous noise level automatically and defines the spatially variant  $h$  accordingly. The implementation starts from the NL-means filtering with low  $h$  and ends after two iterations of segmentation-smoothing. One can see the original NL-means removes some fine features while iterative denoising removes mostly random noise.

- The second test is to illustrate the quality/speed balance for pixel-wise and block-wise SNL-means. We test using the 2-phase toy image in FIGURE 6 with 10% noise ( $\sigma = 27.3$ ). There are two factors in block-wise SNL-means

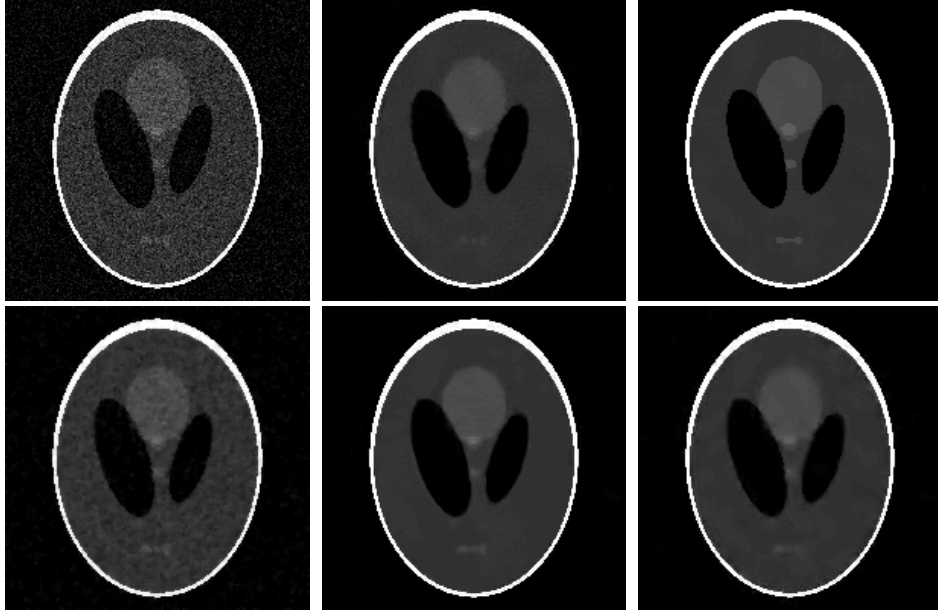


FIGURE 7. Multi-phase SNL-means denoising on a phantom image. First row: noisy image with  $\sigma = 20$ , NL-means, proposed. Second row: ROF, BM3D, K-SVD.

that directly affect its quality and speed. One is the block size described by its radius  $r$  and the other one is the distance between two adjacent blocks denoted by  $d$ . We vary both  $r$  and  $d$  to get different block-wise SNL-means results and compare their PSNR and computation time (in seconds) with that of SNL-means. The comparison is listed in TABLE 7, in which pixel-wise and block-wise SNL-means are abbreviated as SNL and BSNL respectively. The two numbers behind “BSNL” represent  $r, d$  respectively. It is not surprising to see that pixel-wise SNL-means takes more time than all the block-wise SNL-means but gives the best quality. And as  $r$  or  $d$  increases, block-wise SNL-means is faster but pays the price of lower recovery quality.

TABLE 7. Comparison of pixel-wise and block-wise SNL-means denoising efficiency on a toy image.

	Noisy	SNL	BSNL(3,1)	BSNL(3,2)	BSNL(4,2)
PSNR(dB)	19.41	34.46	28.49	27.45	26.27
Time(s)		27.24	22.96	6.01	5.76

- The third set of tests is to show the performance of the extensions of our proposed method in removing Poisson, Rician and Ultrasound noise.

First, we compare our proposed filter (7) with NL-means based filter (i.e.,  $\hat{u} = \tilde{\mathcal{A}}^{-1}(NL(\mathcal{A}(v)))$ ) in FIGURE 9. Even though the PSNR improvement is not much, a close visual comparison shows that the proposed method provides better contrast, especially the buildings in the background.

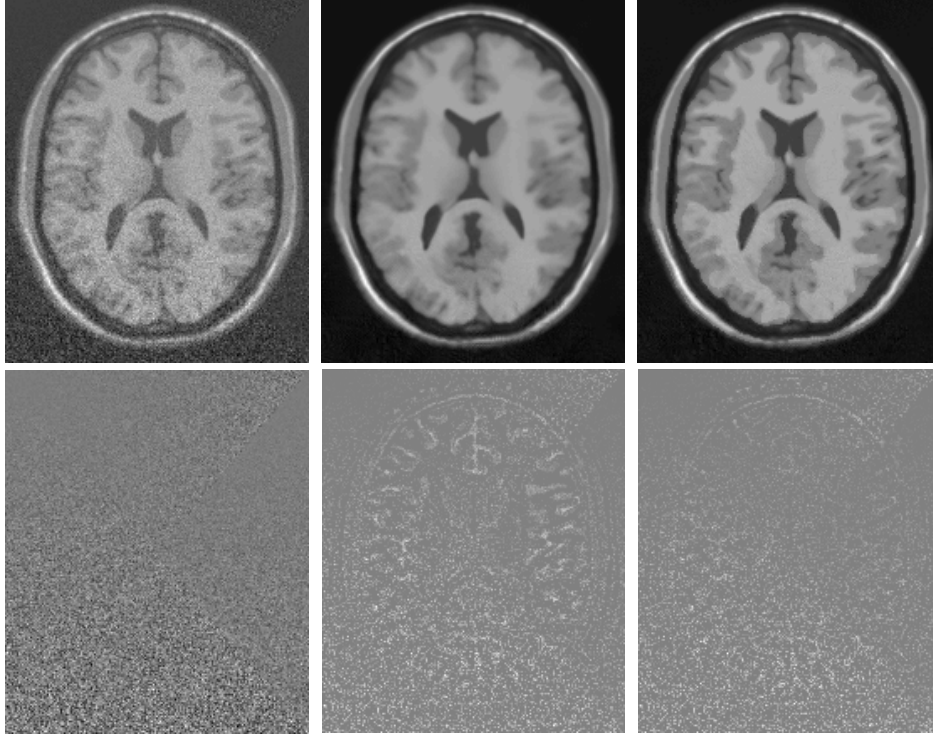


FIGURE 8. Comparison of NL-means and the proposed iterative denoising method in inhomogeneous Gaussian noise removal. First row: noisy MRI brain image with PSNR = 23.66; NL-means result with PSNR = 30.28; iterative SNL result with PSNR = 31.00. Second row: inhomogeneous noise added; difference from the noisy image by NL-means and our method.

Next we add Rician noise on an eagle image as described in Section 4.2 and test our proposed method (8). After segmenting the image into two phases, we applied exponential and Cauchy weight functions for the phases inside and outside of the eagle respectively. As seen from FIGURE 10 our proposed method results in more noise suppression in the background and clearer outline of the eagle.

For removing speckle noise in ultrasound images, we simulate the noisy image in the way as described in Section 4.3. Results of our proposed method (9) and NL-means based ultrasound denoising filter are shown in FIGURE 11. Since Bayesian-based NL-means filter shows better denoising performance than some competitive state-of-the-art methods: Lee's filter, Kuan's filter, SPF and SRAD [38], we restrict the comparison to in between NL-means filter and our method. Consistently, the proposed method leads to a reconstruction closer to the ground truth.

- Finally, we apply the SNL-means filter on a real clinical data. FIGURE 12 shows the result of SNL-means on a scanning electron microscopy (SEM) image of mouse axons showing myelin sheath, mitochondria and microtubules.



FIGURE 9. Comparison of NL-means based Poisson noise removal and our proposed method. First row: clean image ‘cameraman’ of size  $256 \times 256$ , noisy image with Poisson noise and  $\text{PSNR} = 27.41$ . Second row: result by NL-means based denoising with  $\text{PSNR} = 30.19$ , our proposed method (7) with  $\text{PSNR} = 30.60$  where the buildings in the background are preserved better.

The sharp denoising result makes it easier to investigate the mitochondria (amount and structure) and microtubules which are affected by multiple sclerosis (MS).

**6. Conclusion.** The proposed method extracts reliable geometric information and incorporates it to guide the denoising. In the framework of NL-means, we described how to remove excessive inhomogeneous noise while preserving fine features. To speed up computation, the blockwise implementation of our proposed filter is described as well. We also adapted the idea to remove excessive noise of various types. It appears that coupling geometry extraction and nonlocal means leads to better denoising from both quantitative and qualitative aspects. In spite of getting state-of-the-art recoveries, we believe that there is large room for further improvements by exploiting the structures of edges, as well as other image geometry, in a more effective way.

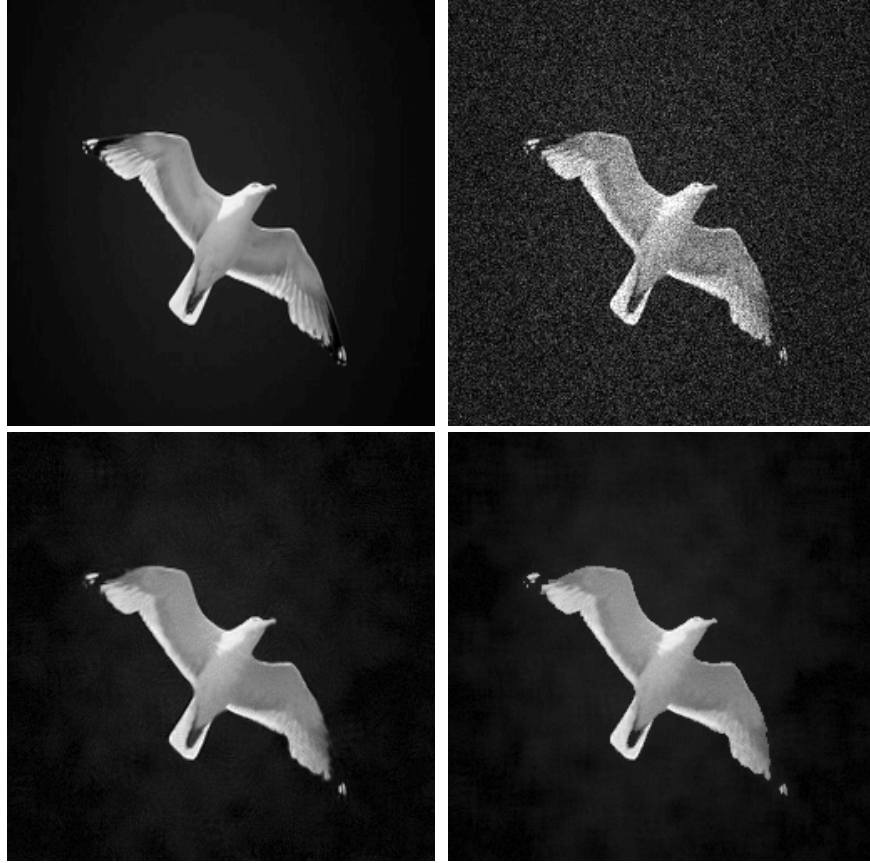


FIGURE 10. Comparison of NL-means based Rician noise removal and our proposed method. First row: clean image ‘eagle’ of size  $256 \times 256$ , noisy image with Rician noise  $\sigma = 25.5$  and PSNR = 19.79. Second row: NL-means based denoising with PSNR = 32.33, our proposed method (8) with PSNR = 32.99.

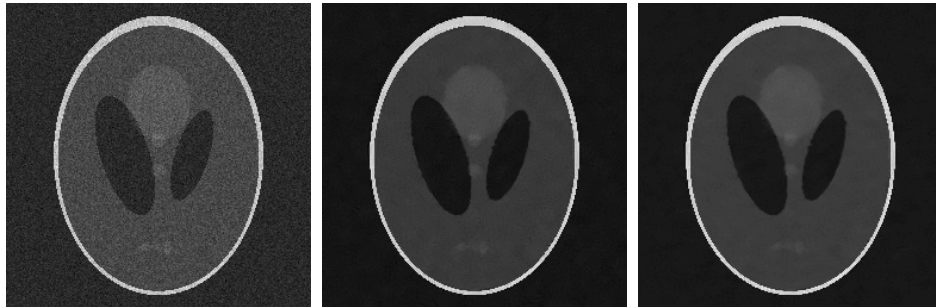


FIGURE 11. Denoising phantom image of intensity range  $[10, 30]$  with simulated ultrasound noise  $\sigma = 0.4$ . From left to right: noisy image with SNR = 22.46, Bayesian based NL-means result with SNR = 32.67 and our proposed with SNR = 33.22.



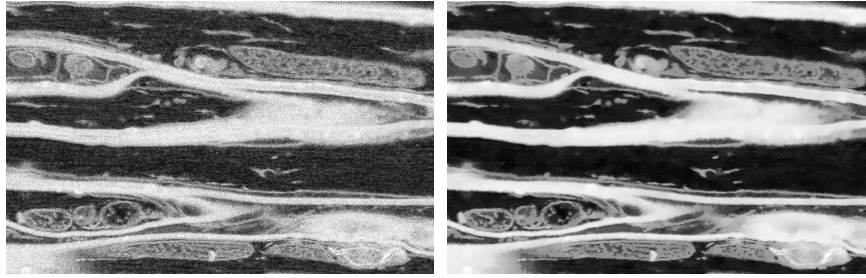


FIGURE 12. Noisy observation and result obtained by SNL-means.

**Acknowledgments.** We would like to thank Drs. Arthur Heuer, Amir Avishai, Nanthawan Avishai from Material science of Case Western Reserve Univeristy and Dr. Grahame Kidd from Cleveland Clinic Foundation for providing the SEM data and evaluation of the denoising results. Thanks also go to the referees who help improve the presentation of this paper.

#### REFERENCES

- [1] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., **42** (1989), 577–685.
- [2] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **12** (1990), 629–639.
- [3] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithm*, Physica D: Nonlinear Phenomena, **60** (1992), 259–268.
- [4] M. Bertalmio, V. Caselles, B. Rougé, and A. Solé, *TV Based Image Restoration with Local Constraints*, Journal of Scientific Computing, **19** (2003), 95–122.
- [5] S. Geman and D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **6** (1984), 721–741.
- [6] P. Saint-Marc, J. S. Chen, and G. Medioni, *Adaptive smoothing: A general tool for early vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **13** (1991), 514–529.
- [7] S. M. Smith and J. M. Brady, *SUSAN - A New Approach to Low Level Image Processing*, International Journal of Computer Vision, **23** (1995), 45–78.
- [8] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, in “Proceedings of Sixth International Conference on Computer Vision”, (2008), 13–24.
- [9] M. Elad, *On the origin of the bilateral filter and ways to improve it*, IEEE Transactions on Image Processing, **11** (2002), 1141–1151.
- [10] S. Durand and J. Froment, *Reconstruction of wavelet coefficients using total variation minimization*, SIAM J. Sci. Comput., **24** (2003), 1754–1767.
- [11] A. Buades, B. Coll, and J. M. Morel, *A non-local algorithm for image denoising*, IEEE Computer Society, **2** (2005), 60–65.
- [12] S. Kindermann, S. Osher and P. W. Jones, *Deblurring and Denoising of Images by Nonlocal Functionals*, Multiscale Modelling and Simulation, **4** (2005), 1091–1115.
- [13] T. Brox and D. Cremers, *Iterated Nonlocal Means for Texture Restoration*, Scale Space and Variational Methods in Computer Vision, (2008), 13–24.
- [14] C. Kervrann, J. Boulanger and P. Coupé, *Bayesian Non-local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal*, Scale Space and Variational Methods in Computer Vision, (2007), 520–532.
- [15] B. Goossens, Q. Luong, A. Pizurica and W. Philips, *An improved non-local denoising algorithm*, In “Local and Non-Local Approximation in Image Processing, International Workshop, Proceedings”, (2008).
- [16] G. Gilboa and S. Osher, *Nonlocal Operators with Applications to Image Processing*, Multiscale Modeling and Simulation, **7** (2008), 1005–1028.

- [17] CA. Deledalle, L. Denis and F. Tupin, *Iterative weighted maximum likelihood denoising with probabilistic patch-based weights*, Transactions on Image Processing, **18** (2009), 2661–2672.
- [18] D. Peter, V. Govindan and A. Mathew, *Nonlocal-Means Image Denoising Technology Using Robust M-Estimator*, Journal of Computer Science and Technology, **25** (2010), 623–631.
- [19] N. Wiest-Daesslé, S. Prima, P. Coupé, S. P. Morrissey and C. Barillot, *Rician Noise Removal by Non-Local Means Filtering for Low Signal-to-Noise Ratio MRI: Applications to DT-MRI*, In “Proceedings of the 11th International Conference on Medical Image Computing and Computer-Assisted Intervention”, (2008), 171–179.
- [20] E. Candés, D. Donoho, *Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges, Curves and Surfaces*, L. L. Schumaker et al. (eds), Vanderbilt University Press, Nashville, TN.
- [21] K. Guo, G. Kutyniok, and D. Labate, *Sparse Multidimensional Representations using Anisotropic Dilation and Shear Operators*, Wavelets and Splines (Athens, GA, 2005), Nashboro Press, Nashville, TN.
- [22] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann and C. Barillot, *An Optimized Blockwise Nonlocal Means Denoising Filter for 3-D Magnetic Resonance Images*, Medical Imaging, IEEE Transactions on, **27** (2008), 425–441.
- [23] CA. Deledalle, F. Tupin and L. Denis, *Poisson NL means: Unsupervised non local means for Poisson noise*, In “Proc. of ICIP”, Hongkong, (2010).
- [24] S. Osher and J. A. Sethian, *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, Journal of Computational Physics, **79** (1988), 12–49.
- [25] T. F. Chan and L. A. Vese, *Active contours without edges*, Image Processing, IEEE Transactions on, **10** (2001), 266–277.
- [26] K. Krishnamoorthy, “Handbook of Statistical Distributions with Applications”, Chapman and Hall/CRC Press, London/Boca Raton, 2006.
- [27] T. Gasser, L. Sroka and C. Jennen-Steinmetz, *Residual variance and residual pattern in nonlinear regression*, Biometrika, **73** (1986), 625–633.
- [28] M. N. Do and M. Vetterli, *The contourlet transform: an efficient directional multiresolution image representation*, IEEE Transactions on Image Processing, **14** (2005), 2091–2106.
- [29] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, Image Processing, IEEE Transactions on, **13** (2004), 600–612.
- [30] E. A. Nadaraya, *On estimating regression*, Theory Probab. Appl., **9** (1964), 141–142.
- [31] F. J. Anscombe, *The transformation of Poisson, binomial and negative-binomial data*, Biometrika, **35** (1948), 246–254.
- [32] M. D. DeVore, A. D. Lanterman and J. A. O’Sullivan, *ATR Performance of a Rician Model for SAR Images*, In “Automatic Target Recognition X, Proc. SPIE 4050”, (2000), 34–45.
- [33] T. Loupas, W. N. McDicken and P. L. Allan, *An adaptive weighted median filter for speckle suppression in medical ultrasonic images*, Circuits and Systems, IEEE Transactions on, **36** (1989), 129–135.
- [34] K. Krissian, *Speckle-constrained anisotropic diffusion for ultrasound images*, In “Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition”, (2005), 547–552.
- [35] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, *Image denoising with block-matching and 3D filtering*, In “Electronic Imaging06, Proc. SPIE 6064”, (2006), no. 6064A-30.
- [36] M. Elad and M. Aharon, *Image Denoising via Sparse and Redundant Representations over Learned Dictionaries*, Image Processing, IEEE Transactions on, **15** (2006), 3736–3745.
- [37] L. A. Vese and T. F. Chan, *A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model*, International Journal of Computer Vision, **50** (2002), 271–293.
- [38] P. Coupé, P. Hellier, C. Kervrann and C. Barillot, *Bayesian non local means-based speckle filtering*, In “Proceedings of ISBI”, (2008), 1291–1294.

E-mail address: wxg49@case.edu

E-mail address: jxq26@case.edu