



HAL
open science

2-D Impulse Noise Suppression by Recursive Gaussian Maximum Likelihood Estimation

Yang Chen, Jian Yang, Huazhong Shu, Luyao Shi, Jiasong Wu, Limin Luo,
Jean-Louis Coatrieux, Christine Toumoulin

► **To cite this version:**

Yang Chen, Jian Yang, Huazhong Shu, Luyao Shi, Jiasong Wu, et al.. 2-D Impulse Noise Suppression by Recursive Gaussian Maximum Likelihood Estimation. PLoS ONE, 2014, 9 (5), pp.e96386. 10.1371/journal.pone.0096386 . hal-01119219

HAL Id: hal-01119219

<https://hal.science/hal-01119219>

Submitted on 9 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2-D Impulse Noise Suppression by Recursive Gaussian Maximum Likelihood Estimation

Yang Chen^{1,2,3,4}, Jian Yang⁵, Huazhong Shu^{1,2}, Luyao Shi^{1,2}, Jiasong Wu^{1,2,3,4}, Limin Luo^{1,2*}(*corresponding author*),
Jean-Louis Coatrieux^{1,2,3,4}, Christine Toumoulin^{1,2,3,4}

1. Laboratory of Image Science and Technology, Southeast University, Nanjing, China;
2. Centre de Recherche en Information Biomedicale Sino-Francais (LIA CRIBs), Rennes, France;
3. INSERM, U1099, Rennes, F-35000, France;
4. Université de Rennes 1, LTSI, Rennes, F-35000, France.
5. Research Center of Mixed Reality and Advanced Display, Beijing Institute of Technology, Beijing, China.

Abstract— An effective approach termed Recursive Gaussian Maximum Likelihood Estimation (RGMLE) is developed in this paper to suppress 2-D impulse noise. And two algorithms termed RGMLE-C and RGMLE-CS are derived by using spatially-adaptive variances, which are respectively estimated based on certainty and joint certainty & similarity information. To give reliable implementation of RGMLE-C and RGMLE-CS algorithms, a novel recursion stopping strategy is proposed by evaluating the estimation error of uncorrupted pixels. Numerical experiments on different noise densities show that the proposed two algorithms can lead to significantly better results than some typical median type filters. Efficient implementation is also realized via GPU (Graphic Processing Unit)-based parallelization techniques.

Index Terms—Median filter, Salt and pepper noise, Recursive Gaussian Maximum Likelihood Estimation (RGMLE), RGMLE-C, RGMLE-CS.

I. INTRODUCTION

Median type filters are known as the effective methods to suppress impulse noise which often arises from sensor damage, malfunctioning or timing errors in signal acquisition [1]. Suppressing impulse noise routinely includes two stages - noise detection and noise removal [2-4]. The Standard Median Filter (SMF) is ineffective in cases with high noise densities, and the SMF processed images are often seriously degraded by detail loss and new introduced artifacts when the noise density is over 50% [5].

Several methods have been developed to overcome the drawback of SMF. Adaptive Median Filter (AMF) was proposed in [6] to improve the performance of median filter by only replacing corrupted pixel values by median values while leaving uncorrupted pixels unchanged. Additionally, the AMF recursively enlarges the filter window until one uncorrupted median is identified. The Decision Based Algorithm (DBA) proposed in [7] is similar to the AMF method except that DBA replaces the corrupted pixel value by a neighborhood pixel value when the median is tagged as a corrupted pixel. In [8], a method named Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) was developed to improve DBA method by removing uncorrupted pixels in median calculation and setting the corrupted pixel value to mean of the selected window when all the window elements are corrupted. To achieve edge-preserving noise suppression, Chan and Nikolova put forward in [9] a two-phase algorithm which includes a SMF based noise detection and an edge-preserving term regularized optimization. In [10], a multi-scale adaptive median-based filtering was developed to suppress salt-and-pepper noise.

In suppressing impulse noise, median-type methods suffer from the image degradation of detail blurring and staircase artifacts. This problem might be more serious for high noise density cases [11], or the case combined with structure blurring [12]. The reason can be ascribed to the fact that only the median information from one single neighboring pixel is selected for median filtering and no structure continuity information is utilized. To overcome this, we propose in this paper an approach termed Recursive Gaussian Maximum Likelihood Estimation (RGMLE) for impulse noise removal. We focus this study on noise removal stage, so only salt-and-pepper type impulse noise with pre-known intensities is considered here. The proposed RGMLE approach solves the maximum likelihood estimation (MLE) of the corrupted points through recursive variance-weighted averaging. A robust strategy of controlling the recursion in RGMLE is also developed by analyzing the estimation error of the uncorrupted pixels. Under the framework of RGMLE, we derived two noise removal algorithms RGMLE-C and RGMLE-CS by respectively using certainty and joint certainty & similarity to estimate the variance. GPU (Graphic Processing Unit)-based technique is also applied to accelerate the proposed processing. Extensive experiments are performed to show that the RGMLE-C and RGMLE-CS algorithms can produce significantly improved results than the comparative median type filters. Among all the methods, the RGMLE-CS algorithm demonstrates the best restoration performance.

The structure of this paper is organized as follows: In section II, we explain the RGMLE approach together with the two derived algorithms RGMLE-C and RGMLE-CS. Experimental results are given and discussed in section III. Section IV concludes this paper with a brief description of its contributions and some open questions for future work.

II. RECURSIVE GAUSSIAN MAXIMUM LIKELIHOOD ESTIMATION

Based on the theory of statistical estimation, the sample median corresponds to the maximum likelihood estimators of locations for independent and identically distributed (i.i.d.) observations obeying the Laplacian distributions with probability density function (pdf) shown in Fig.1 [13-14]. Let x_1, \dots, x_N be N i.i.d. Laplacian distributed observations with unknown location parameter μ and the same variance $2\eta^2$, the MLE of μ is denoted by $\hat{\mu}$, which maximizes the likelihood function:

$$L(x_1, \dots, x_N; \mu) = \prod_{i=1}^N f(x_i - \mu) = \left(\frac{1}{2\eta}\right)^N \exp\left(-\sum_{i=1}^N |x_i - \mu|/\eta\right) \quad (1)$$

This is equivalent to minimizing (2):

$$\sum_{i=1}^N |x_i - \mu|/2\sigma^2 \quad (2)$$

Minimizing (2) with respect to μ leads to the simple median $\hat{\mu} = \text{MEDIAN}(x_1, \dots, x_N)$ [13]. Here for an ordered list, the MEDIAN operator is defined as producing the middle value for odd number of values or the arithmetic mean of the middle two observations for even number of values. Assuming the Laplacian distribution of the observations in the 3×3 window, the 2D median filtering of the center pixel value μ is in fact the maximizer of the likelihood function with respect to each location parameter μ [15-16].

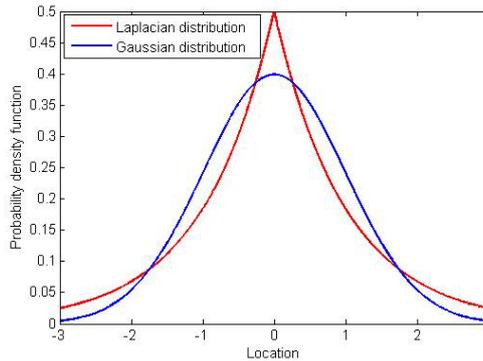


Fig. 1. The probability density functions for Laplacian and Gaussian distributions.

Likewise, with the Gaussian distribution shown in Fig. 1, we can build the likelihood function as follows[17-18]:

$$L(x_1, \dots, x_N; \mu) = \prod_{i=1}^N f(x_i - \mu) = \left(\frac{1}{2\pi\sigma^2}\right)^{N/2} \exp\left(-\sum_{i=1}^N (x_i - \mu)^2/2\sigma^2\right) \quad (3)$$

The value μ minimizing (3) leads to the average operator:

$$\bar{\mu} = \sum_{i=1}^N x_i / N \quad (4)$$

Then we can easily obtain the variance $\sigma_{\bar{\mu}}^2$ of $\bar{\mu}$:

$$\sigma_{\bar{\mu}}^2 = \sigma^2 / N \quad (5)$$

Here, equation (5) shows that the variance estimation for Gaussian distribution decreases linearly with the available observation numbers.

However, in 2-D impulse noise suppression, assuming an identical variance for all samples in (3), the averaging filtering often leads to a non-discriminative averaging operation and might tend to produce oversmoothing in the restored images (we would illustrate this in the following section on experiment). Improved performance can be expected if a discriminative estimation of variance is applied based on the different reliabilities of different observations. Furthermore, considering the fact that the estimation error (or variance) decreases as the available uncorrupted observations increase, the information from corrected intensities should be exploited in a recursive way to refine the final restoration. In this work, we develop a Recursive Gaussian Maximum Likelihood Estimation (RGMLE) for impulse noise suppression. In this 2-D noise suppression case, with y the noisy

image and μ the target restored image, to estimate each corrupted pixel μ_j we can rewrite the likelihood function with variances σ_{ij}^2 for each observation x_{ij} in the filter window N_j :

$$L(x_{i_1}, \dots, x_{i_j}; \mu_j) = \prod_{ij \in N_j} f(x_{ij} - \mu_j) = \left(\frac{1}{2\pi\sigma_{ij}^2} \right)^{|N_j|/2} \exp\left(-\sum_{ij \in N_j} (x_{ij} - \mu_j)^2 / 2\sigma_{ij}^2 \right) \quad (6)$$

Here $|N_j|$ denotes the point number in filter window N_j . In (6), spatially-adaptive variance is used based on joint certainty and similarity information of observations, which is to be elaborated in detail below. Minimizing (10), with respect to μ_j leads to a normalized variance-weighted average:

$$\mu_j = \sum_{ij \in N_j} x_{ij} / \sigma_{ij}^2 \bigg/ \sum_{ij \in N_j} 1 / \sigma_{ij}^2 = \sum_{ij \in N_j} w_{ij} x_{ij} \bigg/ \sum_{ij \in N_j} w_{ij} \quad (7)$$

Here, for each observation i , the weight w_{ij} is a positive value calculated as the direct reciprocal of the variance σ_{ij}^2 .

Algorithm RGMLE

1, Initialization: set μ , $\hat{\mu}$, x , \hat{x} to y ; set the size of the filtering window N for each point and limit recursion number T , and calculate the noise density ρ based on the ratio of the corrupted pixels; Γ denotes the increment of SNR estimation of the re-estimated uncorrupted pixels, and is initiated to $+\infty$.

2, Implementation:

Recursive Loop: iterate on recursion number $t=1,2,\dots, T$

While ($\Gamma > 0$ AND $\rho \leq 70\%$) OR $\rho > 70\%$

Pixel-wise Loop: iterate on pixel index $j=1,2,\dots, |M|$

If $M_j^o = 0$ (pixel j is an originally corrupted pixel):

Calculate w_{ij} using following (15) or (17) for the algorithms RGMLE-C or RGMLE-CS, respectively; Then calculate the restored image via above (7).

Else (pixel j is not an originally corrupted pixel) :

Set μ_j to x_j , and calculate \hat{w}_{ij} using following (12) or (14) for the algorithms RGMLE-C or RGMLE-CS, respectively, and then re-estimated pixel j :

$$\hat{\mu}_j = \sum_{ij \in N_j} \hat{w}_{ij} x_{ij} \bigg/ \sum_{ij \in N_j} \hat{w}_{ij} \quad (8);$$

End If

$$\text{PSNR_U}(t) = 10 \log_{10} \left(\left| M^o \right| \cdot 255^2 \bigg/ \sum_{j=1}^{|M|} M_j^o (\hat{\mu}_j - I_j)^2 \right) \quad (9);$$

End Pixel-wise Loop

$$\Gamma = \begin{cases} \text{PSNR_U}(t) - \text{PSNR_U}(t-1) & t > 1 \\ \text{PSNR_U}(t) & t = 1 \end{cases};$$

Update x to μ ;

End While

End Recursive Loop

Fig.2. The flowchart of the proposed RGMLE strategy.

The recursive update of image μ in (7) should be stopped at certain recursion number to avoid the deterioration caused by the successive involvement of those irrelevant uncorrupted pixels. But the original true image is not available to guide the recursion to stop at the point when the optimal restoration (the closest to the original true image) is reached. Here, we use the estimation error of the ground truth uncorrupted pixels to evaluate the restoration deterioration in recursion. After each recursion, we recalculate the intensities of the uncorrupted pixels using the restored image via (7), and analyze their peak signal-to-noise ratio with respect to the

original ground truth uncorrupted pixels (PSNR_U, calculated via (9) in the below algorithm outline). The recursion is stopped once the PSNR_U stops to increase, which represents the time when the contribution from corrected pixels starts to be outweighed by the deterioration from the irrelevant uncorrupted pixels. This strategy works well for the cases with noise densities ranging from 10% to 70%. For the cases with very high noise densities (>70%) this strategy does not work well because the re-estimation of the uncorrupted pixels mainly come from irrelevant pixel intensities, and can not reflect the recursion deterioration. In this case a large number of recursion is required to get a stable restoration. An illustration will be given in the below experiment to validate this. We also assume intensity value 255 and 0 for the corrupted pixels in the original image y , and let x denote the image to be restored. Two binary matrixes M^o and M^u represent the originally uncorrupted pixel mask M^o and the uncorrupted pixel mask M^u in recursion. We should note that the corrupted pixels will receive some estimation and become uncorrupted pixels when they receive some estimation in the processing. So M^u will be updated to an all-ones matrix when all corrupted pixels get estimated as the recursion proceeds. $|M|$, $|M^u|$ and $|M^o|$ denote the total pixel number in image, and the total non-zero pixel numbers in M^u and M^o , respectively. The flowchart of the RGMLE strategy is given in Fig.2. Here, a large recursion number T is set when the noise density is larger than 70%. We judge the PSNR increase for the estimation of the originally uncorrupted pixels by evaluating the error improvement Γ . The two algorithms RGMLE-C and RGMLE-CS are detailed as follows:

Algorithm RGMLE-C:

In RGMLE-C algorithm, we consider the certainty measure as whether the pixels are corrupted in the original image y . In estimating each μ_j via (7), the variance σ_{ij}^2 is estimated based on the certainty measure of each observation x_{ij} , and can be calculated as follows:

$$\sigma_{ij}^2 = \begin{cases} +\infty & M_{ij}^u = 0 \\ 1 & (M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0) \\ \lambda & M_{ij}^o = 1 \end{cases} \quad (10)$$

where the pixel x_{ij} is considered having no certainty if it is originally corrupted and has not yet been corrected ($M_{ij}^u = 0$), and in this case the variance σ_{ij}^2 is set to $+\infty$. We set σ_{ij}^2 to 1 if the originally corrupted pixel x_{ij} has obtained some certainty via the recursive estimation ($(M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0)$). As to those originally uncorrupted pixels ($M_{ij}^o = 1$), they should be assigned a high certainty, then a low variance is assigned to them with certainty parameter λ ($\lambda < 1$). The corresponding weight w_{ij} can then be calculated as the reciprocal of variance σ_{ij}^2 :

$$w_{ij} = \begin{cases} 0 & M_{ij}^u = 0 \\ 1 & (M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0) \\ 1/\lambda & M_{ij}^o = 1 \end{cases} \quad (11)$$

In re-estimating each uncorrupted pixel $\hat{\mu}_j$ via (8), to accurately reflecting the restoration deterioration in recursion, we exclude the original uncorrupted pixels and only use the restored ones, and the following weight \hat{w}_{ij} is used:

$$\hat{w}_{ij} = \begin{cases} 1 & M_{ij}^o = 0 \\ 0 & M_{ij}^o = 1 \end{cases} \quad (12)$$

Algorithm RGMLE-CS:

Within a filter window for one specific point, two pixels belonging to the same structure tend to come from the same class, and should be assigned low variances (or large weights) in estimating each other. We borrow the patch similarity idea in [19], [20], and quantify the variance by the exponential function of the ℓ^1 norm distance between the patches centered at the two pixels. Here we use ℓ^1 norm for its good edge-preserving property [20], [21]. By introducing this patch similarity information into the variance estimation in the above RGMLE-C algorithm, we calculate each variance σ_{ij}^2 :

$$\sigma_{ij}^2 = \begin{cases} +\infty & M_{ij}^u = 0 \\ \exp(d_{ij}/h) & (M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0) \\ \lambda \cdot \exp(d_{ij}/h) & M_{ij}^o = 1 \end{cases} \quad (13)$$

$$d_{ij} = \left\| \mathbf{P}_{ij}^u \cdot \mathbf{P}_{ij} - \mathbf{P}_j^u \cdot \mathbf{P}_j \right\|_1 \quad (14)$$

where the variance σ_{ij}^2 to estimate μ_j for pixel x_{ij} is set to $+\infty$ if the corrupted pixel has still not been corrected ($M_{ij}^u = 0$). Also if the originally corrupted pixel has obtained some certainty through the correction in recursion ($(M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0)$), we set the variance to the exponential value of the ℓ^1 norm of the distance d_{ij} between the uncorrupted pixels in the two 2D patches \mathbf{P}_j and \mathbf{P}_{ij} centered at point j and its neighboring point ij . In (14), two binary patch masks \mathbf{P}_{ij}^u and \mathbf{P}_j^u denote the positions of the uncorrupted points and are used to exclude the uncorrupted elements in \mathbf{P}_j and \mathbf{P}_{ij} in distance calculation. Practically, parameter h in (13) should be set to modulate the attenuation of the exponent function with respect to distance d_{ij} . As to the originally uncorrupted pixel ($M_{ij}^o = 1$), we also lower the value σ_{ij}^2 via multiplying it by a certainty parameter λ ($\lambda < 1$). Then, the weight w_{ij} is given as

$$w_{ij} = \begin{cases} 0 & M_{ij}^u = 0 \\ \exp(-d_{ij}/h) & (M_{ij}^u = 1) \text{ and } (M_{ij}^o = 0) \\ \exp(-d_{ij}/h)/\lambda & M_{ij}^o = 1 \end{cases} \quad (15)$$

Also, in re-estimating each uncorrupted pixel $\hat{\mu}_j$, we exclude the original uncorrupted pixels and only use the restored ones. In this case the following weight \hat{w}_{ij} is used:

$$\hat{w}_{ij} = \begin{cases} \exp(-\hat{d}_{ij}/h) & M_{ij}^o = 0 \\ 0 & M_{ij}^o = 1 \end{cases} \quad (16)$$

$$\hat{d}_{ij} = \left\| \bar{\mathbf{P}}_{ij}^o \cdot \mathbf{P}_{ij} - \bar{\mathbf{P}}_j^o \cdot \mathbf{P}_j \right\|_1 \quad (17)$$

where, two binary patch masks $\bar{\mathbf{P}}_{ij}^o$ ($\bar{\mathbf{P}}_{ij}^o = 1 - \mathbf{P}_{ij}^o$) and $\bar{\mathbf{P}}_j^o$ ($\bar{\mathbf{P}}_j^o = 1 - \mathbf{P}_j^o$) are used to exclude the original uncorrupted pixels in distance calculation.

III. EXPERIMENTS

A. Experiment configuration

Three 512×512 grayscale images (Lena, Pepper and MRI in the below Fig.3-Fig.8) were chosen in the experiments. The Lena and Pepper images were downloaded from <http://links.uwaterloo.ca/Repository.html>. The intensities of Lena and Peppers images are 8 bit and within the interval $[0, 255]$. The MRI image is in DICOM (Digital Imaging and Communications in Medicine) format and was collected from a head scanning of a 57 years old patient in the first hospital of Nanjing. The intensity range of the MRI image is $[0, 1332]$. The patient in this manuscript has given the written informed consent of using the MRI image data. A wide range of salt-and-pepper noise with density from 10% to 90% (10% increments) is considered. The same simulation as the analysis in above Section II is used. The AMF, DBA and MDBUTMF methods are implemented based on [6-8] for comparison. The MDBUTMF method is actually the standard median filter when any pixel in window is uncorrupted, and switches to averaging filtering otherwise [8]. Replacing the median operator in MDBUTMF by the average operator leads to an average based method and we termed it MDBUTAF. We implement this MDBUTAF method to provide a fair comparison with median and average filtering. We also compare our method with the method based on steering kernel regression (SKR method) in [23], which has demonstrated good performance in denoising, interpolation and super-resolution. The parameters involved in different methods are all set to give stable results with the highest peak signal-to-noise ratio (PSNR, calculated via (18)). We use a 3×3 filtering window N in the AMF, DBA, MDBUTMF and MDBUTAF methods. For the SKR method, the global smoothing parameter and the kernel size were manually tuned to give the results with the highest PSNR. For the RGMLE-C and RGMLE-CS algorithms, several parameters need to be set, namely the size of filtering window W , the certainty parameter λ and the recursion limit T . Here 3×3 and 5×5 windows are respectively used in the RGMLE-C and RGMLE-CS algorithms. For RGMLE-CS algorithm, the

5×5 window allows a utilization of more distant pixels via the above similarity-based weighting strategy. Also two additional parameters (the smoothing parameter h and the size of patch P) need to be set for the RGMLE-CS method. In RGMLE-CS algorithm, a 7×7 patch P is set to give an effective characterization of local structures. And the smoothing parameter h is set to 3.5 for Lena and Peppers images, and to 10 for the MRI image with a larger intensity range. Specially, for RGMLE-CS algorithm, we set P and N to 1×1 and 3×3 in the first recursion to overcome the lack of structure information in the original corrupted noisy image. We set the certainty parameter λ in (11) and (15) to 0.2 to increase the weights of the uncorrupted pixels. The recursion number limit T is set to 50 to include a long recursion in restoration using RGMLE-C and RGMLE-CS algorithms.

The pixel-wise operation (7) in RGMLE-C and RGMLE-CS algorithms allows for easy parallelization using GPU technique [24], [25]. Under Compute Unified Device Architecture (CUDA) framework, we set the total number of blocks in grid to the row size of the image, and the total number of threads in each block to the column size of the image. In implementing the RGMLE-C and RGMLE-CS algorithms, all threads in the block-grid structure execute simultaneously to perform all the pixel-wise operations, and only half of the distance d is calculated because of the symmetry property of distance calculation ($d_{ij} = d_{ji}$). All the images were processed in a PC workstation (Intel Core™ 2 Quad CPU and 4096 Mb RAM, GPU (NVIDIA GTX465)) with Visual C++ as the developing language (Visual Studio 2008 software; Microsoft).

Restoration performances are quantitatively measured by PSNR and the structural similarity index comparisons (SSIM) proposed in [26]:

$$\text{PSNR}(\mu, I) = 10 \log_{10} \left(|M| \cdot 255^2 / \sum_{j=1}^{|M|} (\mu_j - I_j)^2 \right) \quad (18)$$

$$\text{SSIM}(\mu, I) = \frac{(2m_\mu m_I)(2\sigma_{\mu I} + c_2)}{(m_\mu^2 + m_I^2 + c_1)(\sigma_\mu^2 + \sigma_I^2 + c_2)} \quad (19)$$

where, μ and I denote the restored image and the original true image. $|M|$ denotes the total pixel number in image. m_μ and m_I denote the mean intensities of images μ and I , σ_μ and σ_I are the standard deviation of images μ and I , $\sigma_{\mu I}$ is the covariance of images μ and I , $c_1 = (K_1 L)^2$ and $c_2 = (K_2 L)^2$ with L the dynamic range of the pixel values (255 for 8-bit grayscale images), K_1 and K_2 are set to 0.01 and 0.03 as suggested in [25].

B. Restoration Result

Fig. 3 and Fig. 4 illustrate the restoration results of Lena image with 80% and 90% noise densities for different methods. Likewise, Fig. 5 and Fig. 6 present the results for Pepper image, and Fig. 7 and Fig. 8 present the results for MRI image. The calculated PSNR and SSIM with respect to the original true images are also given in the captions below the figures. The original true images and the corrupted images are illustrated in (a) and (b) for reference. In Fig.3-Fig.8, the denoised images from different methods are given in (c)-(i), and (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i). We can see that the traditional median-type filters (AMF, DBA and MDBUTMF methods with results (c)-(e) in Fig.3-Fig.8) can not effectively estimate corrupted points, and tend to introduce many artifacts in restoration. Comparing the images (e) and (f) in Fig.3-Fig.8, we find that under high noise densities the MDBUTAF method works much better than MDBUTMF method (over 2.5dB PSNR enhancement). But we also find the MDBUTMF method with non-discriminative averaging operation can not effectively restore image details and overcome oversmoothing. Comparing the images (f) and (g) in Fig.3-Fig.8, we can see that, by recursively incorporating the certainty measure into estimation, the RGMLE-C algorithm produces notable improvement with more than 2.5dB PSNR enhancement over the MDBUTAF method.

Obviously, the proposed RGMLE-CS algorithms give good performance in terms of both noise suppression and detail preservation (see images (i) in Fig.3-Fig.8). With patch similarity information used, some tiny features (e. g. the hair structures in Lena image and the brain structures in MRI image) can be successfully restored by the RGMLE-CS algorithm. Among all the methods, the RGMLE-CS algorithm leads to the results with significantly higher PSNR values than other methods except the SKR method. We note that SKR method can give higher PSNR values than the RGMLE-CS algorithm in Lena image (80% noise density), Peppers image (90% noise density) and MRI image (80% and 90% noise density). But in terms of SSIM, which is deemed a more robust quality metric than PSNR, the proposed RGMLE-CS algorithm provides the best results. Compared to the results from SKR method, the processed results from RGMLE-CS algorithm present structures with higher contrast in the case of 80% noise density (see the arrows in images (iz) in Fig.3, Fig.5 and Fig.7), and perceptively comparable results under 90% noise density.



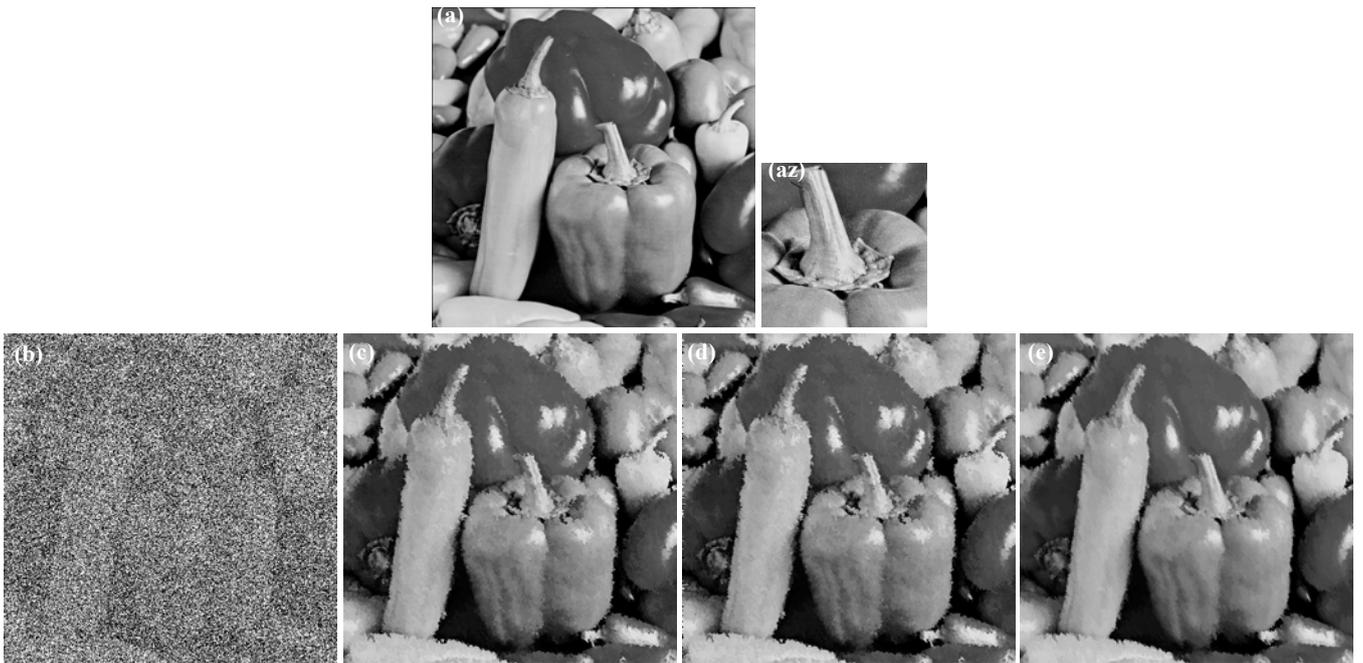


Fig. 3. Restoration results of different methods for Lena image (80% salt-and-pepper noise). (a) Original image (b) Corrupted image (6.43 dB, SSIM=0.0091). (c) AMF (23.44 dB, SSIM=0.7113) [6]. (d) DBA filter (23.18 dB, SSIM=0.7112) [7]. (e) MDBUTMF filter (24.51 dB, SSIM=0.7682) [8]. (f) MDBUTAF filter (27.28 dB, SSIM=0.8286). (g) SKR method (31.59 dB, SSIM=0.8793) [23]. (h) RGMLE-C algorithm (29.84 dB, SSIM=0.8674). (i) RGMLE-CS algorithm (31.54 dB, SSIM=0.8872). (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i).





Fig. 4. Restoration results of different methods for Lena image (with 90% salt-and-pepper noise). (a) Original image (b) Corrupted image (PSNR=5.90 dB, SSIM=0.0052). (c) AMF (17.72 dB, SSIM=0.5762) [6]. (d) DBA filter (19.52 dB, SSIM=0.5751) [7]. (e) MDBUTMF filter (20.18 dB, SSIM=0.6233) [8]. (f) MDBUTAF filter (23.82 dB, SSIM=0.7373). (g) SKR method (28.52 dB, SSIM=0.8297) [23]. (h) RGMLE-C algorithm (27.25 dB, SSIM=0.8084). (i) RGMLE-CS algorithm (28.52 dB, SSIM=0.8318). (az)-(jz) denote the zoomed local parts for the corresponding images in (a)-(j).



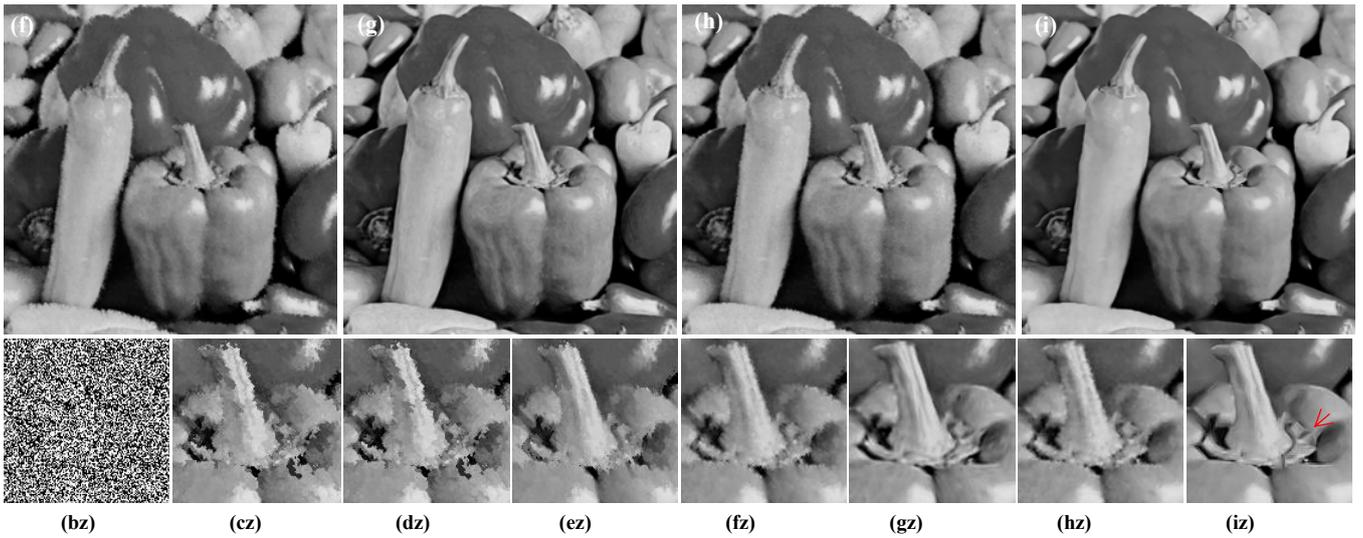
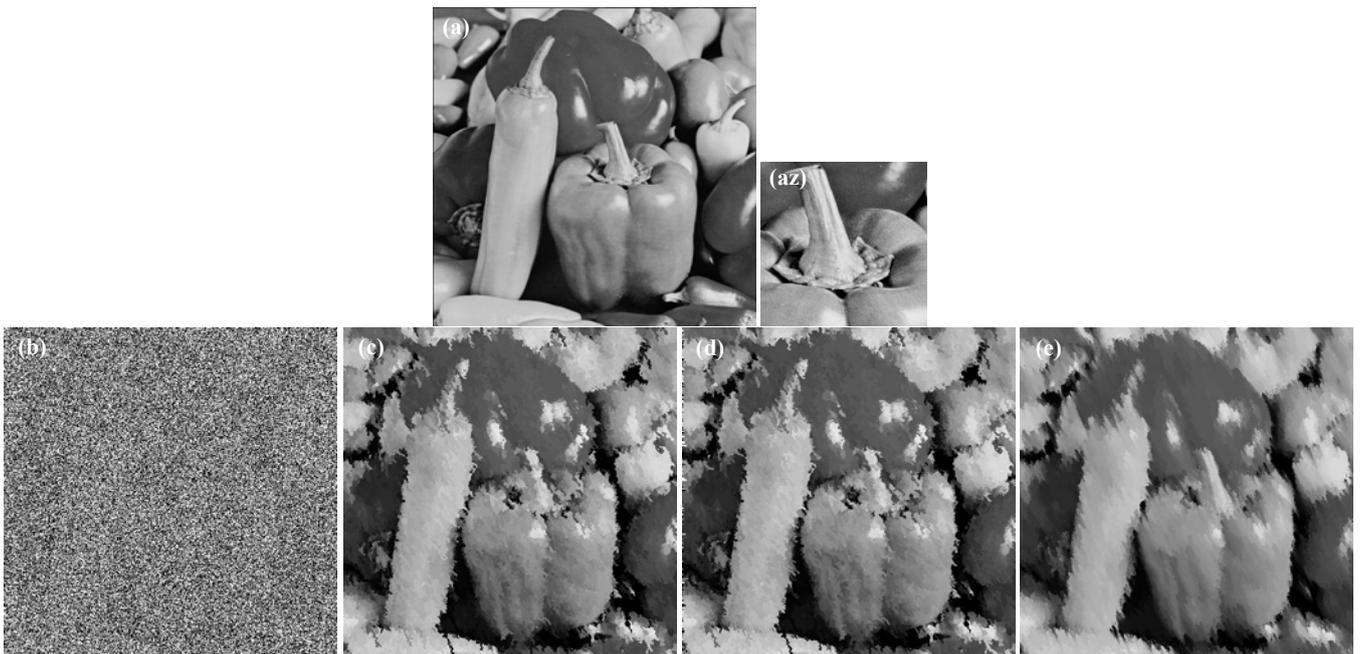


Fig. 5. Restoration results of different methods for Peppers image (80% salt-and-pepper noise). (a) Original image (b) Corrupted image (6.26 dB, SSIM=0.0093). (c) AMF (22.85 dB, SSIM=0.6581) [6]. (d) DBA filter (22.65 dB, SSIM=0.6592) [7]. (e) MDBUTMF filter (23.78 dB, SSIM=0.7174) [8]. (f) MDBUTAF filter (26.72 dB, SSIM=0.7805). (g) SKR method (31.16 dB, SSIM=0.8094) [23]. (h) RGMLE-C algorithm (29.97 dB, SSIM=0.8194). (i) RGMLE-CS algorithm (32.01 dB, SSIM=0.8381). (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i).



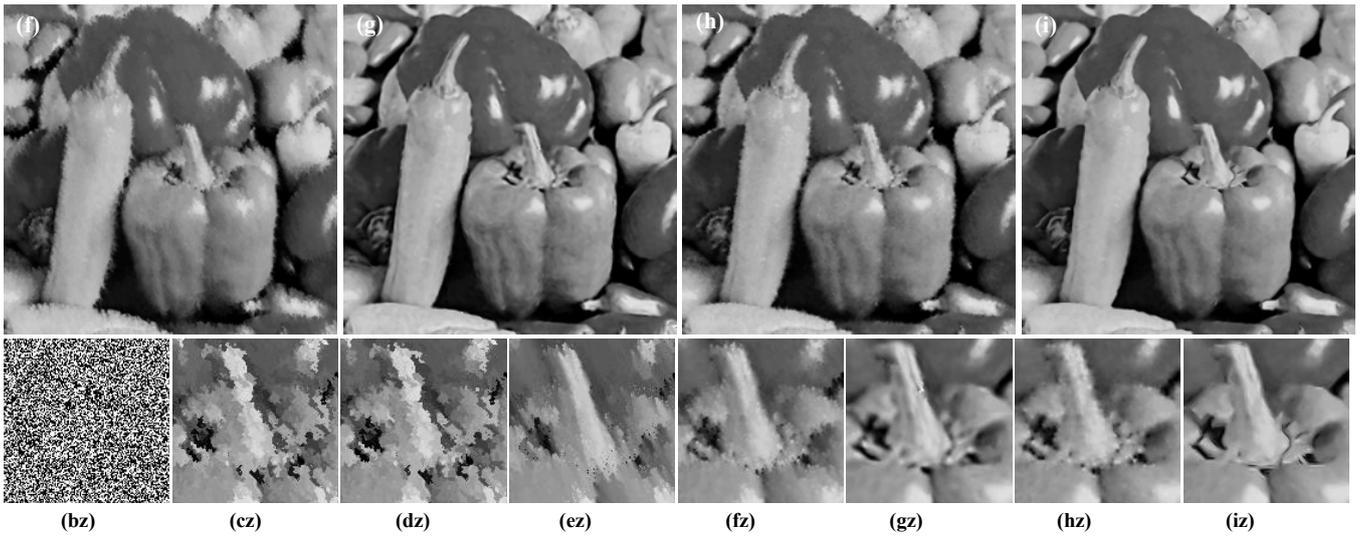
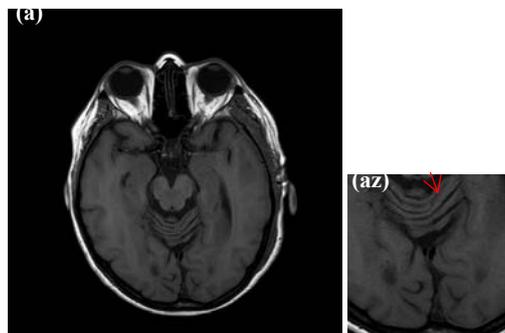


Fig. 6. Restoration results of different methods for Peppers image (90% salt-and-pepper noise). (a) Original image (b) Corrupted image (PSNR=5.84 dB, SSIM=0.0061). (c) AMF (19.02 dB, SSIM=0.5205) [6]. (d) DBA filter (18.94 dB, SSIM=0.5232) [7]. (e) MDBUTMF filter (19.01 dB, SSIM=0.5657) [8]. (f) MDBUTAF filter (22.92dB, SSIM=0.6824). (g) SKR method (29.11 dB, SSIM=0.7765) [23]. (h) RGMLE-C algorithm (27.56 dB, SSIM=0.7649). (i) RGMLE-CS algorithm (28.83 dB, SSIM=0.7915). (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i).



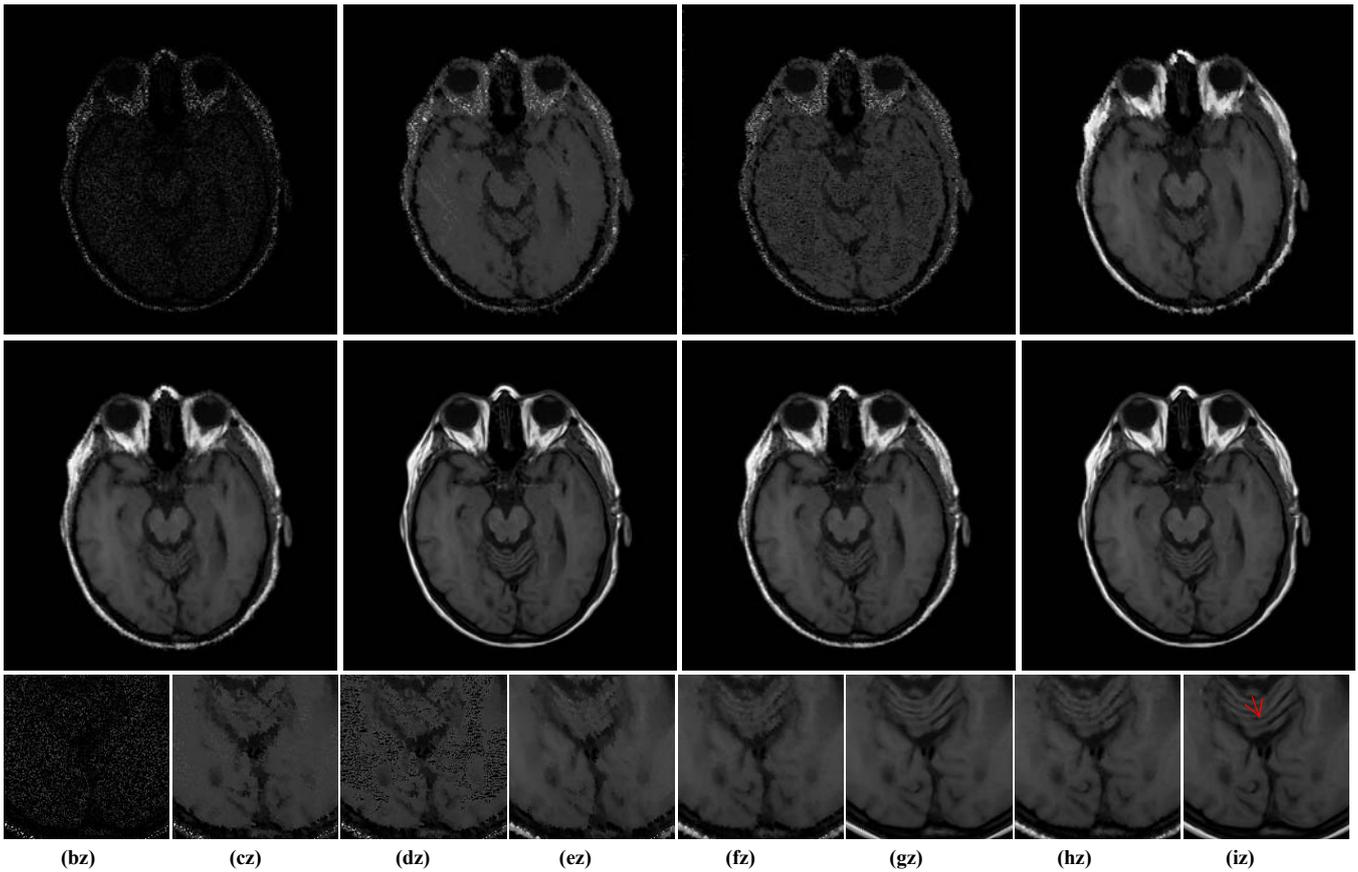
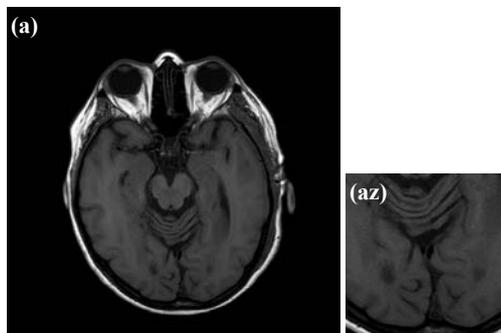


Fig. 7. Restoration results for MRI image (80% salt-and-pepper noise). (a) Original image (b) Corrupted image (2.54 dB, SSIM=0.0195). (c) AMF (8.53 dB, SSIM=0.6889) [6]. (d) DBA filter (6.26 dB, SSIM=0.6191) [7]. (e) MDBUTMF filter (13.68 dB, SSIM=0.7981) [8]. (f) MDBUTAF filter (16.54 dB, SSIM=0.8524). (g) SKR method (24.22 dB, SSIM=0.9203) [23]. (h) RGMLE-C algorithm (19.74 dB, SSIM=0.9012). (i) RGMLE-CS algorithm (23.76 dB, SSIM=0.9226). (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i).



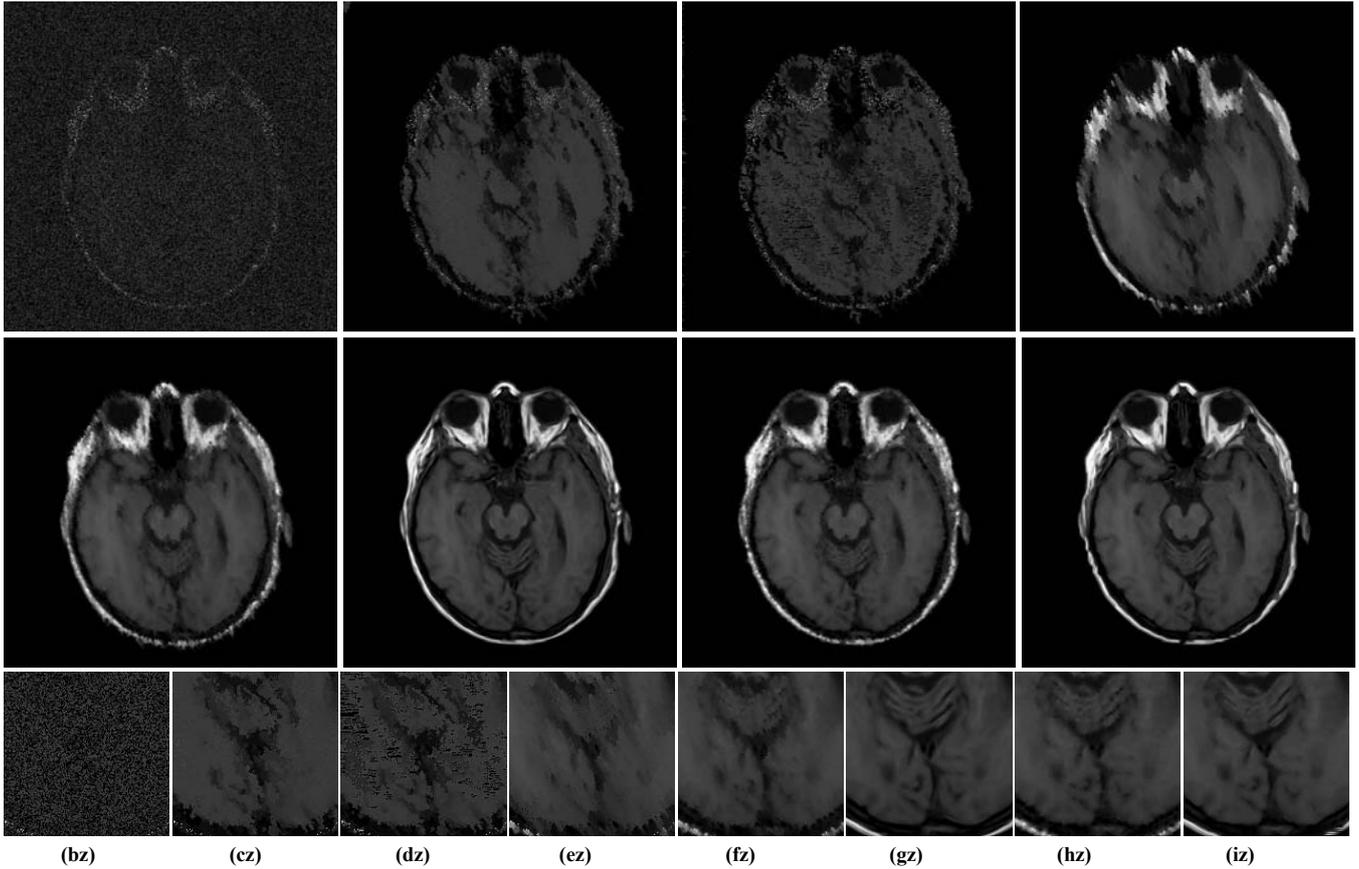
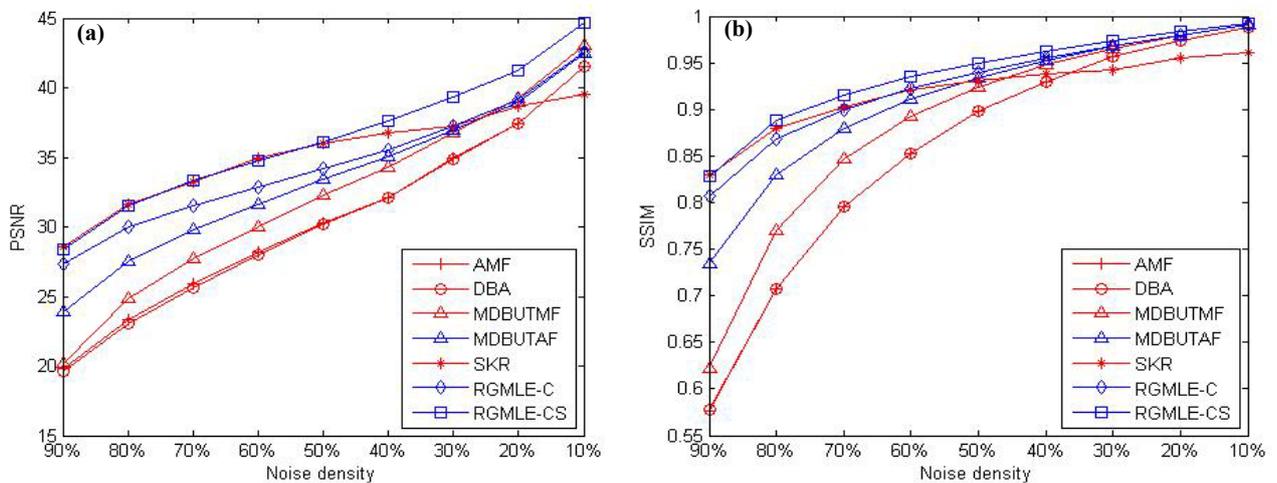


Fig. 8. Restoration results for MRI image (90% salt-and-pepper noise). (a) Original image (b) Corrupted image (2.03 dB, SSIM=0.0096). (c) AMF (6.91 dB, SSIM=0.6051) [6]. (d) DBA filter (5.91 dB, SSIM=0.5614) [7]. (e) MDBUTMF filter (9.30 dB, SSIM=0.6623) [8]. (f) MDBUTAF filter (12.87 dB, SSIM=0.7532). (g) SKR method (19.40 dB, SSIM=0.8706) [23]. (h) RGMLE-C algorithm (16.81 dB, SSIM=0.8426). (i) RGMLE-CS algorithm (18.92 dB, SSIM=0.8717). (az)-(iz) denote the zoomed local parts for the corresponding images in (a)-(i).

In Fig. 9, we summarize the PSNR and SSIM values of the restored images for noise densities ranging from 10% to 90%. We can observe in the plots that the obtained PSNR and SSIM values decrease as noise level rises for all the methods, and our proposed RGMLE-C and RGMLE-CS algorithms obtain significantly higher PSNR and SSIM than the other methods, except the SKR method in almost all noise densities. This advantage becomes more prominent as noise density increases. The SKR method can sometimes provide higher PSNR values than RGMLE-CS algorithm under high noise densities (80% and 90%), but has poor performance in the cases of low noise densities. We can see the proposed RGMLE-CS algorithm can give the best results in terms of SSIM for all the noise densities.



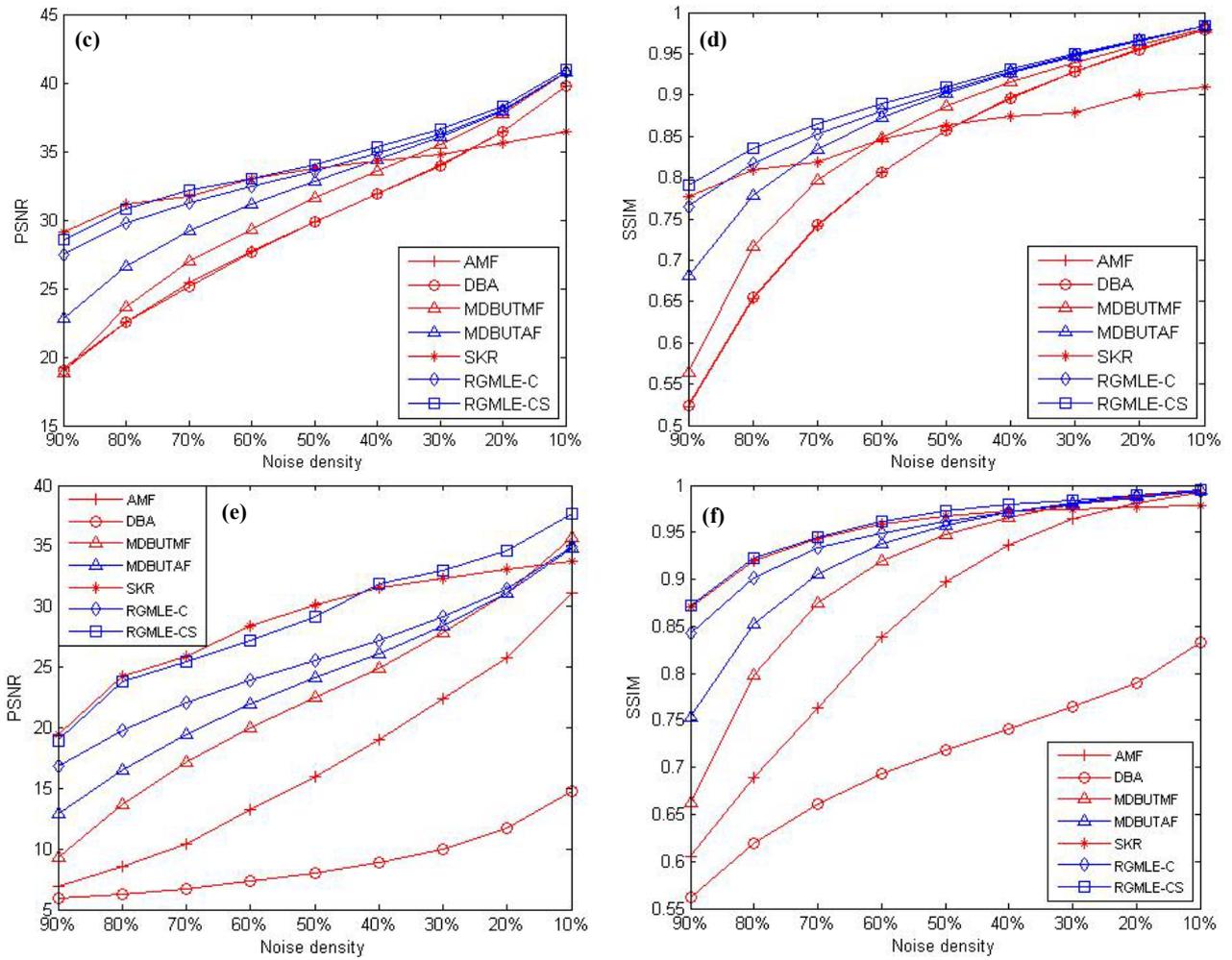
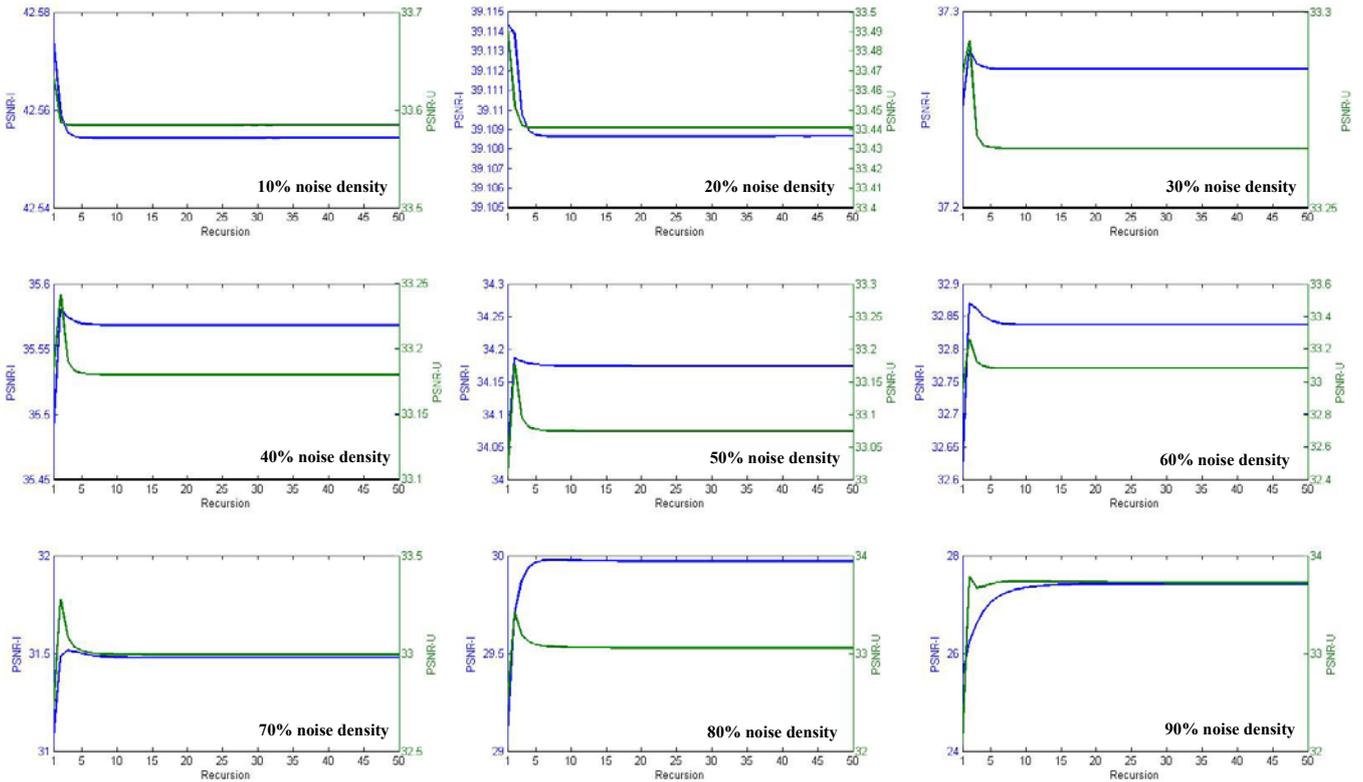
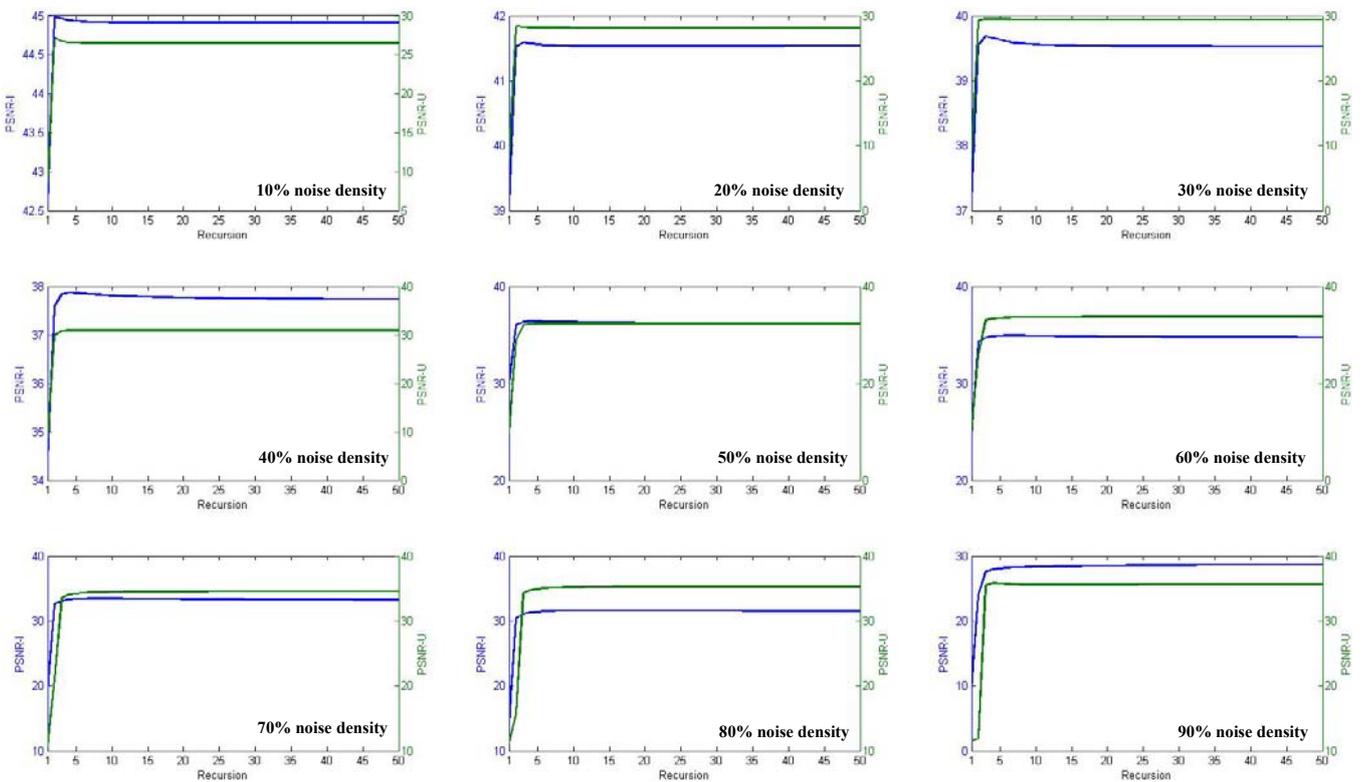


Fig. 9. PSNR and SSIM values for various noise densities for different algorithms. (a) and (b): the PSNR and SSIM for Lena image; (c) and (d): the PSNR and SSIM for Pepper image; (e) and (f): the PSNR and SSIM for MRI image.



(a) RGMLE-C algorithm



(b) RGMLE-CS algorithm

Fig. 10. For Lena image test, the PSNR of the whole restored image (PSNR-I) and the re-estimated uncorrupted pixels (PSNR-U) as the recursion proceeds for noise densities from 10% to 90%. (a) the result of RGMLE-C algorithm; (b) the result of RGMLE-CS algorithm.

TABLE I
COMPUTATION TIME (IN CPU SECONDS) FOR DIFFERENT METHODS WITH DIFFERENT NOISE DENSITIES

Images and noise levels	AMF	DBA	MDBUTMF	MDBUTAF	SKR	Serial RGMLE-C	Serial RGMLE-CS	Parallelized RGMLE-C	Parallelized RGMLE-CS
Lena (10% noise density)	3.35	3.04	1.99	1.99	35.58	13.63	107.73	0.14	0.85
Lena (30% noise density)	4.61	3.16	3.26	3.44	94.93	21.69	197.45	0.23	1.27
Lena (50% noise density)	8.86	3.13	4.39	4.61	106.53	22.72	400.73	0.24	2.65
Lena (70% noise density)	27.22	3.18	5.56	5.95	86.86	23.61	998.64	0.26	11.99
Lena (90% noise density)	85.53	3.22	7.25	7.20	69.76	78.53	1109.16	0.83	12.88
Peppers (10% noise density)	30.07	3.05	2.09	1.96	36.41	13.75	104.67	0.15	0.85
Peppers (30% noise density)	48.54	3.12	3.18	3.52	86.89	21.69	203.78	0.23	1.35
Peppers (50% noise density)	48.23	3.19	4.52	4.69	103.01	22.62	225.26	0.27	1.46
Peppers (70% noise density)	53.25	3.19	5.61	5.99	87.72	23.50	233.47	0.29	1.64
Peppers (90% noise density)	94.87	3.24	7.26	7.28	72.23	80.85	1218.09	0.91	14.13
MRI (10% noise density)	324.60	3.05	1.99	1.96	82.10	13.68	104.89	0.15	0.89
MRI (30% noise density)	377.83	3.22	3.24	3.44	74.05	22.07	198.37	0.24	1.32
MRI (50% noise density)	307.07	3.22	4.50	4.82	88.65	23.18	215.40	0.25	1.41
MRI (70% noise density)	222.87	3.21	5.70	6.00	82.70	23.61	238.47	0.28	1.66
MRI (90% noise density)	167.03	3.33	7.39	7.39	71.93	79.44	1201.26	0.89	13.52

C. Validation of Recursion Stopping

Fig. 10 plots the PSNR (via (18)) with respect to the recursion numbers ($T=50$) for the whole restored image (denoted by PSNR_I, blue lines) and the PSNR_U (via above (9)) of the re-estimated uncorrupted pixels (green lines) under different noise densities. Lena image with the same parameter setting as above is used in this validation. Fig. 10 shows that the PSNR-I decreases when the recursion proceeds across one specific point, which shows the deterioration will appear in recursion for the proposed RGMLE-C and RGMLE-CS methods. In Fig. 10, we can note a good match between the highest obtained PSNR of the restored images and that of the re-estimated uncorrupted pixels, but such match degree tends to obviate as the noise density increases to over 70%. The reason of the match deviation for noise density higher than 70% is that under very high noise densities the re-estimation of the uncorrupted pixels mainly relies on irrelevant pixel intensities, which can not accurately reflect the overall deterioration in restoration. This observation validates the above proposed stopping strategy for the RGMLE algorithm.

D. Computation Complexity

Table I lists the CPU time costs in implementing all the methods in the above experiments with different noise densities. The CPU time costs for implementing both the parallelized RGMLE methods (parallelized RGMLE-C and RGMLE-CS) and the original serial versions (serial RGMLE-C and RGMLE-CS) are included. The AMF and SKR methods require much more intensive computation than other median-type filters because the AMF method recursively enlarges the filter window to obtain an uncorrupted median and the SKR method requires a computationally costly operation of matrix inversion. Table I also shows that the proposed RGMLE-C and RGMLE-CS methods require more computation than other methods, and about 150 times acceleration is achieved after GPU based parallelization. We can also notice that the required computation cost of RGMLE-C and RGMLE-CS methods increases as the noise density increases because more recursions need to be run based on the recursion stopping rule.

IV. CONCLUSION

This paper proposed a Recursive Gaussian Maximum Likelihood Estimation (RGMLE) for 2-D impulse noise removal. Assuming Gaussian distribution for neighboring intensities, the proposed RGMLE approach solves the maximum likelihood estimation (MLE) of corrupted points through recursive variance-weighted averaging. Simulation in Section II reveals that Gaussian distribution assumption is theoretically preferable to Laplacian distribution assumption in estimating corrupted points under high noise densities. In validation we use salt-and-pepper type impulse noise with pre-known intensities. Two noise removal algorithms RGMLE-C and RGMLE-CS are respectively derived by using certainty based or certainty&similarity based variance estimations. An effective stopping strategy for recursion is also developed based on the estimation error of uncorrupted pixels using restored image information. Extensive experiments have been performed to show the RGMLE-C and RGMLE-CS algorithm can obtain significantly better results than the comparative median type filters. Among all the methods, the RGMLE-CS algorithm achieves good restoration performance through the incorporation of patch similarity information. The application of the proposed RGMLE methods can be easily extended to suppress general impulse noise with random values when combined with the detection

methods proposed in [3], [4], [11], [27]. An extension to Gaussian noise suppression would also be performed by considering a spatially varying noise corruption degree for each pixel.

With GPU parallelization, the RGMLE-CS algorithm can be efficiently applied and should be suggested for impulse noise suppression. A constant smoothing parameter h is currently used in the RGMLE-CS algorithm, and we can improve the algorithm by optimizing the estimation of parameter h [28], [29]. The denoising performance of the RGMLE-CS algorithm can also be further enhanced via the combination with multiscale or compress sensing techniques [30], [31]. Current recursion stopping method is based on the knowledge of estimation error of uncorrupted pixels, and from above we know that this strategy does not apply to the case of very high noise density. More effective method is to be studied to control recursion to obtain the exactly optimal restoration for different noise levels. Computation efficiency still holds a future issue to be considered. Though significant acceleration has been obtained by GPU parallelization technique, further computation optimization is still required to realize real-time processing of the RGMLE-CS algorithm.

ACKNOWLEDGEMENT

This research was supported by National Basic Research Program of China under grant (2010CB732503), National Natural Science Foundation under grants (81000636, 81370040), and the Project supported by Natural Science Foundations of Jiangsu Province (BK2011593). This work was also supported by the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education.

REFERENCES

- [1] A. Bovik, Handbook of Image and Video Processing. New York: Academic, 2000.
- [2] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*. Boca Raton, FL: CRC, 1997.
- [3] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 360–363, 2002.
- [4] X.-M. Zhang and Y.-L. Xiong, "Impulse noise removal using directional difference based noise detector and adaptive weighted mean filter," *IEEE Signal Process. Lett.*, vol. 16, no. 4, pp. 295–298, 2009.
- [5] P. E. Ng and K. K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1506–1516, 2006.
- [6] H. Hwang and R. A. Haddad, "Adaptive median filter: New algorithms and results," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 499–502, 1995.
- [7] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision based algorithm for removal of high density impulse noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, 2007.
- [8] S. Esakkirajan, T. Veerakumar, Adabala N. Subramanyam, and C. H. Prem Chand, "Removal of high density salt and pepper noise through modified decision based unsymmetric Trimmed Median Filter," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 287–290, 2011.
- [9] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt and pepper noise removal by median type noise detectors and detail preserving regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [10] P. Sree, P. Kumar, R. Siddavatam, R. Verma, "Salt-and-pepper noise removal by adaptive median-based lifting filter using second-generation wavelets", *Signal, Image and Video Processing*, vol. 7, no. 1, pp. 111-118, 2013.
- [11] Y. Q. Dong, R. H. Chan and S. F. Xu, "A detection statistic for random-valued impulse noise," *IEEE Trans. Image Process.*, vol. 16, pp. 1112-1120, 2007.
- [12] J. F. Cai, R. H. Chan and M. Nikolova, "Two-phase Approach for Deblurring Images Corrupted by Impulse Plus Gaussian Noise," *Inverse Problems and Imaging.*, vol 2, no. 2, pp. 187-204, 2011.
- [13] G. R. Arce, "A general weighted median filter structure admitting negative weights," *IEEE Trans. Signal Process.*, vol. 46, pp. 3195–3205, 1998.
- [14] J. W. Harris, and H. Stocker, "Maximum Likelihood Method." (§21.10.4) in *Handbook of Mathematics and Computational Science*. New York: Springer-Verlag, pp. 824, 1998.
- [15] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II, Transforms and Median Filters*, Huang, T. S., Ed. New York: Springer, 1981.
- [16] Y. H. Lee and S. Tantarata, "A simple formula for the variance of the median filter output with Laplacian input," *IEEE Trans. Signal Process.*, vol. 39, pp. 984-986, 1991.
- [17] D. Andrews and C. Mallows, "Scale mixtures of normal distributions," *J. Royal Stat. Soc.*, vol. 36, pp. 99–102, 1974.
- [18] M. J. Wainwright and E. P. Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images," in *Adv. Neural Information Processing Systems (NIPS*99)*, S. A. Solla, T. K.
- [19] A. Buades, B. Coll, J. M. Morel. "A non-local algorithm for image denoising," In: *Proc IEEE Computer Vision and Pattern Recognition*, pp. 60–65, 2005.
- [20] Y. Chen, J. H. Ma, Q. J. Feng, L. M. Luo, W. F. Chen and P.C. Shi, "Nonlocal Prior Bayesian Tomographic Reconstruction," *Journal of Mathematical Imaging and Vision*, 30(2), pp.133-146, 2008.
- [21] Y. Chen, L.M Luo, W.F Chen et al. "Joint-MAP tomographic reconstruction with patch similarity based mixture prior model," *SIAM, Multiscale Modeling and Simulation*, 9, pp. 1399-1419, 2011.
- [22] K. N. Chaudhury, A. Singer, "Nonlocal Euclidean medians," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 745 - 748, 2012.
- [23] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing*, vol. 16, pp. 349-366, 2007.
- [24] NVIDIA CUDATM Programming Guide (Version 3.0), http://developer.download.nvidia.com/compute/cuda/3.0/toolkit/docs/NVIDIA_CUDA_Programming_Guide.pdf.
- [25] Accelerating MATLAB with CUDA Using MEX Files (White Paper), http://developer.nvidia.com/object/matlab_cuda.html.
- [26] Z. Wang, Alan C. Bovik, Hamid R. Sheikh and Eero P. Simoncelli "Image quality assessment: from error visibility to structural similarity" *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.
- [27] C.-H. Lin, J.-S. Tsai, and C.-T. Chiu, "Switching bilateral filter with at Texture/noise detector for universal noise removal," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2307–2320, 2010.
- [28] C. -A. Deledalle, L. Denis, F. Tupin, "Iterative weighted maximum likelihood denoising with probabilistic patch-based weights," *IEEE Trans. Image Process.*, vol.18, no.12, pp.2661-2672, 2009

- [29] D. Van De Ville and M. Kocher, "Nonlocal means with dimensionality reduction and SURE-based parameter selection," *IEEE Trans. Image Process.*, vol. 20, pp. 2683–2690, 2011.
- [30] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, pp. 3736–3745, 2006.
- [31] Y. Chen, Z. Yang, Y. Hu, G. Yang, L. Luo, W. Chen et al. "Thoracic low-dose CT image processing using an artifact suppressed large-scale nonlocal means," *Phys. Med. Biol.*, vol. 57, pp. 2667–2688, 2012.