

Denoising of Sparse Images in Impulsive Disturbance Environment

Isidora Stanković · Irena Orović ·
Miloš Daković · Srdjan Stanković

Received: date / Accepted: date

Abstract The paper presents a method for denoising and reconstruction of sparse images based on a gradient-descent algorithm. It is assumed that the original (non-noisy) image is sparse in the two-dimensional Discrete Cosine Transform (2D-DCT) domain. It is also assumed that a number of image pixels is corrupted by a salt and pepper noise. In addition, we assume that there are pixels corrupted by a noise of any value. In this paper we introduce a method to find the positions of the corrupted pixels when the noise is not of the salt and pepper form. The proposed algorithm for noisy pixels detection and reconstruction works blindly. It does not require the knowledge about the positions of corrupted pixels. The only assumption is that the image is sparse and that the noise degrades this property. The advantage of this reconstruction algorithm is that we do not change the uncorrupted pixels in the process of the reconstruction, unlike common reconstruction methods. Corrupted pixels are detected and removed iteratively using the gradient of sparsity measure as a criterion for detection. After the corrupted pixels are detected and removed, the gradient algorithm is employed to reconstruct the image. The algorithm is tested on both grayscale and color images. Additionally, the case when both salt and pepper noise and a random noise, within the pixel values range, are combined is considered. The proposed method can be used without explicitly imposing the image sparsity in a strict sense. Quality of the reconstructed image is measured for different sparsity and noise levels using the structural similarity index, the mean absolute error, mean-square

This work is supported by the Montenegrin Ministry of Science, project grant funded by the World Bank loan: CS-ICT “New ICT Compressive sensing based trends applied to: multimedia, biomedicine and communications”.

Isidora Stanković · Irena Orović · Miloš Daković · Srdjan Stanković
Faculty of Electrical Engineering, University of Montenegro
Dzordza Vasingtona bb, 20 000 Podgorica, Montenegro
E-mail: isidoras@ac.me · irenao@ac.me · milos@ac.me · srdjan@ac.me
Corresponding author is Isidora Stanković.

error and peak signal-to-noise ratio and compared to the traditional median filter and recent algorithms, one based on the total-variations reconstruction and a two-stage adaptive algorithm.

Keywords gradient algorithm · compressive sensing · denoising · image processing · reconstruction

1 Introduction

The analysis and reconstruction of corrupted/missing samples in a sparse signal using a reduced set of available samples has been introduced within the field of compressive sensing (CS). A sparse signal can be represented in certain transformation domain by a very few nonzero coefficients, comparing to the total signal length. Compressive sensing is based on the statement that if a signal is sparse in a transformation domain, it can be reconstructed with less samples than required by the Shannon-Nyquist theorem, if the conditions of the compressive sensing reconstruction are met, [1–13]. Many real-world signals are sparse in a certain transformation domain. This means that the theory of CS can be widely used in almost all areas of digital signal processing, such as multimedia (audio, speech, image, video), radars, remote sensing, biomedicine, communications, etc. Since the introduction of CS, many reconstruction algorithms have been developed. One of these algorithms, belonging to the large group of gradient-based algorithms, uses the gradient of the L_1 -norm as a sparsity measure in the minimization problem [14, 15]. In this algorithm the missing/corrupted pixels are considered as the minimization variables. The image is reconstructed in the spatial domain. This property makes the algorithm suitable for denoising of corrupted pixels in a noisy environment.

Common images can be considered as sparse in the two-dimensional discrete cosine transform. It means that they have only few nonzero coefficients in that domain. These images can be reconstructed from reduced set of pixels. The reduced set can occur for different reasons. If some pixels are corrupted, we can declare them as unavailable and try to reconstruct them using different CS reconstruction methods. The impulsive noise in an image could appear due to analog to digital conversion errors, communication errors, dead pixels in image acquisition equipment, etc. In this paper, we will assume that, in addition to the salt and pepper noise, there exists a noise whose values are within the range of the original image pixels. Since we assume that the noise exists in some image pixels only, we will consider this as an impulsive noise, although its amplitudes can be within the range of pixel values.

Algorithms for denoising sparse images with specific impulsive noise, whose positions can be detected based on the noise values (salt and pepper noise), were presented in [16–30]. The initial form of gradient-based algorithm for image reconstruction was introduced in [16]. In [17] a local median value was used for recovery of the noisy pixels. A technique for image denoising by using morphological filter and a training-based optimization scheme is presented in [18]. A method to reconstruct images corrupted by a salt-and-pepper noise

based on partially noise-free pixels is given in [19]. Algorithm in [20] is based on median filtering, patch-based sparse representation and weighted L_1 - L_1 regularization method. In [21, 22] noisy images are filtered using an adaptive discrete cosine transform filtering algorithm. Noisy signal and image recovery, assuming various constraints on noise positions and sparsity in a generally dictionary domain, is the topic of [23]. A decision-based median method that changes only the corrupted pixels by the median or the neighboring pixel values is presented [24]. The method in [25] uses a linear combination of uncorrupted pixel values and the median of the local window. Reconstruction of images using a primal-dual total variation method is presented in [26–28]. In [29] an average-filtering algorithm was applied to the image denoising. The algorithm presented in [30] introduces a combination of adaptive decision-based median technique with the block matching 3D filtering.

The difference in the approach considered here is that we cannot retrieve the positions of corrupted pixels based on their values. Noisy pixels will be marked down by detecting those that degrade the image sparsity in the transformation domain. Based on the sparsity measure and its gradient estimation, a criterion for iterative detection of the corrupted pixels will be defined. It uses the property that the corrupted pixels degrade the sparsity [31, 32], meaning that their variation causes larger finite difference estimates of the sparsity measure gradient. When the corrupted pixels are detected they are removed and considered as unavailable, and the potentially uncorrupted pixels will be used as the only available ones. At this point, the problem is reduced to the CS based image reconstruction. In contrast to the robust filtering algorithms that use corrupted pixels in the analysis (and minimize their influence), the presented algorithm removes these pixels and uses only the uncorrupted ones in the reconstruction. An additional sparsification step is proposed as well. Since the gradient-based reconstruction assumes sparsity in an implicit way (minimizing norm-one), the reconstructed image is only approximately sparse. Imposing the strict image sparsity constraint on the result, an additional improvement in the reconstruction is achieved.

The performance of image denoising and reconstruction are analyzed by using the structural similarity (SSIM) index introduced in [33], as well as on the mean absolute error (MAE), mean squared error (MSE), and peak signal-to-noise ratio (PSNR). The proposed algorithm is also tested on a combination of salt and pepper and random noise (whose values are within the pixel intensity range). This method is applied on both grayscale and color images.

The reconstruction performance is analyzed for various noise levels and noise types, by using the proposed method, classical median filtering and two recent denoising and restoration methods [27, 29]. The two methods used for comparison are the total variation with primal-dual CS reconstruction algorithm [26–28] and the two-stage adaptive reconstruction method [29]. Since [29] contains comparison with other methods such as [24, 25] (proving its superiority), in this way we included an indirect comparison with those methods.

The paper is organized as follows. After the introduction in Section 1, the theoretical background about compressive sensing and image sparsification is

presented in Section 2. In Section 3, the reconstruction algorithm with the pixel selection criterion is presented. In Section 4, the experimental results are shown. The comparisons using different error parameters are presented in Section 5.

2 Theoretical Background

Let us consider an 8-bit $M \times N$ image $x(m, n)$. The 2D-DCT of this image and its inverse can be written as [10, 11]

$$\begin{aligned} X(k, l) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \varphi(k, l, m, n) \\ x(m, n) &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X(k, l) \psi(m, n, k, l) \end{aligned} \quad (1)$$

where $\varphi(k, l, m, n)$ is the 2D-DCT basis function and $\psi(m, n, k, l)$ is the 2D-DCT inverse basis function. They are defined as

$$\varphi(k, l, m, n) = \psi(m, n, k, l) = c_k c_l \cos\left(\frac{\pi(2m+1)k}{2M}\right) \cos\left(\frac{\pi(2n+1)l}{2N}\right). \quad (2)$$

The constants c_k and c_l are used for scaling and they are defined as

$$c_k = \begin{cases} 1/\sqrt{M}, & \text{for } k = 0 \\ \sqrt{2/M}, & \text{for } k \neq 0 \end{cases} \quad c_l = \begin{cases} 1/\sqrt{N}, & \text{for } l = 0 \\ \sqrt{2/N}, & \text{for } l \neq 0 \end{cases}.$$

The image can be rewritten into a vector form as

$$\mathbf{x} = [x(0, 0), x(0, 1), \dots, x(M-1, N-1)]^T. \quad (3)$$

The inverse transform in the vector/matrix form is defined as

$$\mathbf{x} = \mathbf{\Psi} \mathbf{X} \quad (4)$$

where $\mathbf{\Psi}$ is the rearranged inverse transformation matrix with rearranged elements defined in (2).

Assume that the image is sparse in the 2D-DCT domain with sparsity K such that $K \ll MN$. Assume that only the pixels at $(m, n) \in \mathbb{N}_{\mathbb{A}}$ are available, while the other pixels are unavailable or omitted as corrupted ones. In the initial calculation we can set the unavailable pixels to zeros. The initial image is then represented as

$$x_a(m, n) = \begin{cases} x(m, n) & \text{for } (m, n) \in \mathbb{N}_{\mathbb{A}} \\ 0 & \text{elsewhere} \end{cases} \quad (5)$$

with $\mathbb{N}_{\mathbb{A}} = \{(m_1, n_1), (m_2, n_2), \dots, (m_{N_A}, n_{N_A})\}$ being the set of N_A available (uncorrupted) samples.

The nonzero entries of (5) can be considered as measurements within the CS framework [1–3]. In vector notation these values are

$$\mathbf{y} = [x(m_1, n_1), x(m_2, n_2), \dots, x(m_{N_A}, n_{N_A})]^T.$$

Each measurement $x(m_i, n_i)$ is the linear combination of coefficients \mathbf{X} with sparsity K

$$x(m_i, n_i) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \psi(m_i, n_i, k, l) X(k, l) \quad (6)$$

or

$$\mathbf{y} = \mathbf{A}\mathbf{X} \quad (7)$$

where $(m_i, n_i) \in \mathbb{N}_A$ and \mathbf{A} is a measurement matrix of size $N_A \times MN$. It is obtained from the basis matrix Ψ , with the rows corresponding to the positions of the available samples \mathbb{N}_A . The goal of the CS is to reconstruct signal by minimizing the sparsity measure of \mathbf{X} subject to the available samples \mathbf{y} [1–3]. The most obvious sparsity measure is based on simple counting of nonzero values in the transformation domain. Counting can be done by using the so called L_0 -norm, $\|\mathbf{X}\|_0$. The problem formulation is then

$$\min \|\mathbf{X}\|_0 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{X}. \quad (8)$$

This is the easiest and most basic way to count nonzero coefficients and to minimize the sparsity. However, it is an NP-hard combinatorial problem. The closest convex form is the L_1 -norm, with the problem formulation

$$\min \|\mathbf{X}\|_1 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{X}. \quad (9)$$

Under certain conditions, the formulations (8) and (9) give the same solution [5]. In the gradient-descent based reconstruction, the problem is solved by varying the missing pixels until the minimum of sparsity measure is reached. The available samples remain unchanged in this method.

The image sparsity in the transformation domain is the basic condition for reconstruction, based on a reduced set of pixels. The image sparsification is done according to the quantization matrix of the JPEG standard. The standard quantization matrix is defined using 50% quality factor which is applied on the 8×8 DCT blocks of the image. For different quality factors (QF), which influences the number of DCT components in a block (i.e. the sparsity level of the block) [11], the quantization matrix is defined as

$$Q_{QF} = \text{round}(Q_{50} \cdot q), \quad (10)$$

where Q_{50} is the standard quantization matrix and q is calculated as

$$q = \begin{cases} 2 - 0.02QF, & \text{for } QF \geq 50 \\ \frac{50}{QF}, & \text{for } QF < 50 \end{cases}. \quad (11)$$

In this paper, the image is first split into blocks of size 8×8 and then processed and reconstructed. After the reconstruction of each block is finished, they are combined together to get the whole image back. Also, we assumed different quality factors to compare the performance of the algorithm with various sparsity levels. This will be examined in Section 5.

3 Reconstruction Algorithm

The approach considered here deals with images which have $MN - N_A$ noisy pixels with the noise amplitude being in the range of the available pixels values. The aim is to reconstruct the corrupted pixels without the knowledge of their number and the positions.

The algorithm is based on the minimization of the sparsity measure through iterations [14, 31, 32]. Each image pixel is considered as possibly corrupted. Its value is varied by adding $\pm\Delta$. For each pixel the gradient sparsity measure $\|\mathbf{X}\|_1$ is estimated based on its finite difference value. The pixel producing the largest gradient estimate is considered as corrupted and omitted. Then the iterative process is repeated until the sparsity measure does not change significantly. All corrupted pixels are set as missing/unavailable pixels. Then the CS-based reconstruction process is performed. The missing pixels are varied in the reconstruction step to produce the most sparse solution. The uncorrupted (available) pixels are not changed. Details of this method will be presented next.

3.1 Algorithm

In this subsection, the basic gradient-descent reconstruction algorithm, assuming known positions of the available (uncorrupted) pixels [14], is explained. It is described in Algorithm 1. We set the corrupted pixels to zero, and continue with the iterative reconstruction procedure.

The algorithm can be read as follows: the corrupted/missing pixels are considered as variables. We add and subtract a value Δ to the corrupted pixel at (m_i, n_i) . The images formed in this way are given by:

$$\begin{aligned} \mathbf{x}_a^+ &= x^{(p)}(m, n) + \Delta\delta(m - m_i, n - n_i) \\ \mathbf{x}_a^- &= x^{(p)}(m, n) - \Delta\delta(m - m_i, n - n_i) \end{aligned} \quad (12)$$

The 2D-DCT transforms of these images, in a vector form, are \mathbf{X}_a^+ and \mathbf{X}_a^- .

After that, the gradient of the sparsity measure is estimated using finite difference of the L_1 -norms of \mathbf{X}_a^+ and \mathbf{X}_a^- as

$$g(m_i, n_i) = \|\mathbf{X}_a^+\|_1 - \|\mathbf{X}_a^-\|_1.$$

According to the gradient value, the corrupted pixel is updated (step 16 in Algorithm 1). The calculation is continued until the stopping criterion is reached

Algorithm 1 Gradient-based image reconstruction**Input:**

- Set of the uncorrupted pixel positions \mathbb{N}_A
- Corrupted image \mathbf{x}

Output:

- Reconstructed image \mathbf{x}_R

```

1: function GRADREC( $\mathbf{x}, \mathbb{N}_A$ )
2:    $x_a^{(0)}(m, n) \leftarrow \begin{cases} x(m, n) & \text{for } (m, n) \in \mathbb{N}_A \\ 0 & \text{for } (m, n) \notin \mathbb{N}_A \end{cases}$ 
3:    $\Delta \leftarrow \max_{m, n} |x_a^{(0)}(m, n)|$ 
4:    $p \leftarrow 0$ 
5:   repeat
6:     repeat
7:        $\mathbf{x}_a^{(p+1)} \leftarrow \mathbf{x}_a^{(p)}$ 
8:       for all  $(m_i, n_i) \notin \mathbb{N}_A$  do
9:          $\mathbf{x}_a^+ \leftarrow \mathbf{x}_a^{(p)}$ 
10:         $x_a^+(m_i, n_i) \leftarrow x_a^+(m_i, n_i) + \Delta$ 
11:         $\mathbf{X}_a^+ \leftarrow \text{DCT2}\{\mathbf{x}_a^+\}$ 
12:         $\mathbf{x}_a^- \leftarrow \mathbf{x}_a^{(p)}$ 
13:         $x_a^-(m_i, n_i) \leftarrow x_a^-(m_i, n_i) - \Delta$ 
14:         $\mathbf{X}_a^- \leftarrow \text{DCT2}\{\mathbf{x}_a^-\}$ 
15:         $g(m_i, n_i) \leftarrow \|\mathbf{X}_a^+\|_1 - \|\mathbf{X}_a^-\|_1$ 
16:         $x_a^{(p+1)}(m_i, n_i) \leftarrow x_a^{(p)}(m_i, n_i) - \mu g(m_i, n_i)$ 
17:        $p \leftarrow p + 1$ 
18:       until stopping criterion is satisfied
19:        $\Delta \leftarrow \Delta/3$ 
20:   until required precision is achieved
21:    $\mathbf{x}_R \leftarrow \mathbf{x}_a^{(p)}$ 
22:   return  $\mathbf{x}_R$ 

```

for a given step Δ . As a stopping criterion we can use the angles between directions of the two successive gradients. If these are almost 180 degrees it means that the algorithm reached the solution with a precision defined by Δ . It oscillates around it and the step Δ should be decreased. The procedure is repeated until a required precision is achieved.

The normalization constant μ used in step 16 for image size $M \times N$ is

$$\mu = \frac{1}{\sum_{k=1}^M \sum_{m=1}^N |\text{DCT2}\{\delta(m - m_0, n - n_0)\}|} \approx \frac{\pi^2}{8\sqrt{MN}}$$

where $\delta(m, n)$ is 2D Dirac delta pulse.

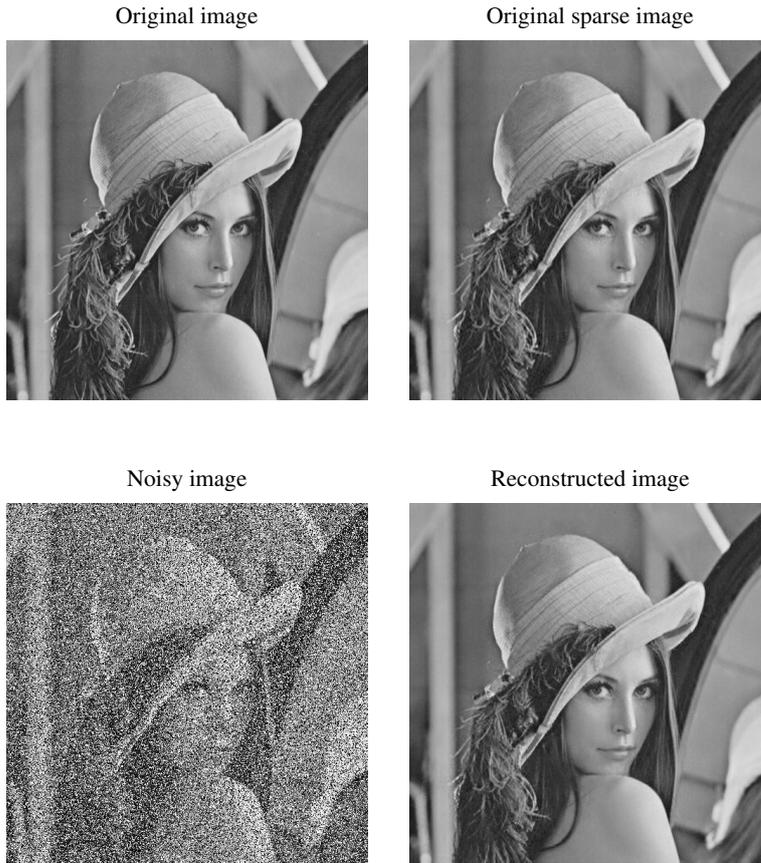


Fig. 1 Reconstruction of image corrupted with 50% salt and pepper noise: Original image (top left); Sparse image (top right); Noisy image (bottom left); Reconstructed image (bottom right)

The algorithm can also be used when the noise is much stronger than the signal itself, meaning that the corrupted pixels are distinguishable from the uncorrupted pixels (salt and pepper noise), so that their positions are easily found. When we have strong noise in the image, we will omit the noisy pixels from the calculations and continue with the reconstruction as described in Algorithm 1. Example with such a noise is given in Fig. 1. The original image, and the sparsified image (with $QF=25$) are shown in Fig. 1 (top). The image with salt and pepper noise is shown in Fig. 1 (bottom left). Half of the total number of pixels are corrupted. The positions of the pixels corrupted with salt and pepper noise are found using the α -trimming method. The reconstructed image is presented in Fig. 1 (bottom right).

3.2 Pixel Selection Motivation

Consider a sparse image $x(m, n)$. Assume that one pixel is corrupted at a position (m_0, n_0) . The noisy image will be defined as $x_a(m, n)$, and the noisy pixel can be defined as $x_a(m_0, n_0) = x(m_0, n_0) + z$ where z is some noise. The corrupted pixel will be changed for $\pm\Delta$ to form the images

$$\begin{aligned} x_a^+(m, n) &= x(m, n) + (z + \Delta) \delta(m - m_0, n - n_0) \\ x_a^-(m, n) &= x(m, n) + (z - \Delta) \delta(m - m_0, n - n_0) \end{aligned} \quad (13)$$

where Δ is the gradient parameter. The gradient is estimated as

$$g(m_0, n_0) = \|\mathbf{X}_a^+\|_1 - \|\mathbf{X}_a^-\|_1 \quad (14)$$

where \mathbf{X}_a^+ and \mathbf{X}_a^- are transformation coefficients of the images in (13). The transforms of images $x_a^+(m, n)$ and $x_a^-(m, n)$ are defined as

$$\begin{aligned} X_a^+(k, l) &= X(k, l) + (z + \Delta) \varphi(k, l, m_0, n_0) \\ X_a^-(k, l) &= X(k, l) + (z - \Delta) \varphi(k, l, m_0, n_0) \end{aligned} \quad (15)$$

Assume that the 2D-DCT of the original (uncorrupted) image pixels is $X(k, l)$ and $(z \pm \Delta) \varphi(k, l, m_0, n_0)$ is the 2D-DCT of the one missing (corrupted) pixel. The sparsity measures of $X_a^+(k, l)$ and $X_a^-(k, l)$ can be written as a sum of the original image measure and the measure of the noise (with the Δ shifts)

$$\begin{aligned} \|\mathbf{X}_a^+\|_1 &= \sum_{k,l=0}^{N-1} |X_a^+(k, l)| \cong \|\mathbf{X}\|_1 + |z + \Delta| C \\ \|\mathbf{X}_a^-\|_1 &= \sum_{k,l=0}^{N-1} |X_a^-(k, l)| \cong \|\mathbf{X}\|_1 + |z - \Delta| C \end{aligned} \quad (16)$$

where C is a constant dependent on (m_0, n_0) and the image size. The sparsity measure can be written as

$$g(m_0, n_0) = \|\mathbf{X}_a^+\|_1 - \|\mathbf{X}_a^-\|_1 \cong |z + \Delta| C - |z - \Delta| C. \quad (17)$$

For variations from the true image value smaller than the step $|z| < \Delta$ we get

$$g(m_0, n_0) \cong 2Cz \sim z \quad (18)$$

meaning that the gradient is proportional to the intensity of noise at the corrupted pixel.

3.3 Pixel Selection Algorithm

The image is split into blocks of size 8×8 and reconstructed for each block individually. The aim is to find the positions of missing (corrupted) pixels and

select the available (uncorrupted) pixels. The available pixels positions will be denoted by \mathbb{N}_A . According to the basic idea, this will be achieved by repeating steps 9-15 from Algorithm 1 for all pixels. That is, we form signals as in (12) for each pixel. Then, the 2D-DCT of these signals can be written as

$$\begin{aligned}\mathbf{X}_a^+ &= X_a^+(k, l) = \text{DCT2}\{x_a^+(m, n)\} \\ \mathbf{X}_a^- &= X_a^-(k, l) = \text{DCT2}\{x_a^-(m, n)\}\end{aligned}\quad (19)$$

and the gradient value at the missing pixels is calculated as in equation (14) for each (m, n) .

In the initial iteration all pixels are considered as possibly corrupted. The gradient of sparsity measure is estimated for each pixel. The pixel whose variation produce the largest gradient value is marked as corrupted. Note that the value Δ in this case is equal to the initial Δ value used for the reconstruction in the algorithm. For the pixel selection $\Delta = \max_{(m,n)} |x_a(m, n)|$ since we should provide that $\Delta \geq z$ for any possible z . After the highest corrupted pixel (or few of them) is removed in one iteration, the process is repeated with the remaining pixels. In each iteration the sparsity measure is calculated. When all corrupted pixels are removed, the sparsity of the image will be restored and it will not change anymore. This can be used as a stopping criterion for the corrupted pixels detection. By introducing the pixel selection criterion we get a set of available (uncorrupted) pixels. Algorithm 1 is then used for the image reconstruction at the positions of the removed pixels. In this way, we can generalize the algorithm to be completely independent on the number and the positions of corrupted pixels. The procedure is repeated until a required precision is achieved, and the algorithm is repeated until all blocks of the image are reconstructed.

The pixel selection and image reconstruction procedure is illustrated in Algorithm 2. In steps 2 and 3 we initially assume that all pixels are available and there is no corrupted pixels. In the case of salt and pepper noise we can detect some corrupted pixels by using α -trimming or L-statistics prior to iterative selection procedure (steps 4–17). In the next step we test all individual pixels as possibly corrupted ones. The pixel with largest gradient of sparsity measure is marked as corrupted. It is reconstructed and the procedure is repeated. In the reconstruction we include all previously detected positions of corrupted pixels. For the image processed in blocks the procedure presented in Algorithm 2 is repeated for each image block.

Block edge effects can influence the quality of reconstruction. In some blocks, a few pixels from an object which dominantly belongs to the neighbouring blocks may appear at the edging pixels. Algorithm could recognize these pixels as disturbance and perform their “reconstruction” in order to minimize the sparsity of the considered block. To avoid this edge effect, image analysis is done with partially overlapping blocks. Only the central (non-overlapping) parts of the blocks are included for the final reconstruction. In this way, the edge effects are completely avoided.

Algorithm 2 Proposed method**Input:**

- Image \mathbf{x} of size $M \times N$ with possibly corrupted pixels
- Number of pixels to be selected in each iteration r

Output:

- Reconstructed image \mathbf{x}
- Set of the uncorrupted pixels \mathbb{N}_A

```

1:  $\Delta \leftarrow \max_{m,n} |x(m,n)|$ 
2:  $\mathbb{N}_A \leftarrow \{(m,n) : m = 1, 2, \dots, M, n = 1, 2, \dots, N\}$ 
3:  $\mathbb{N}_x \leftarrow \emptyset$ 
4: repeat
5:   for all  $(n,m) \in \mathbb{N}_A$  do
6:      $\mathbf{x}^+ \leftarrow \mathbf{x}$ 
7:      $x^+(m,n) \leftarrow x^+(m,n) + \Delta$ 
8:      $\mathbf{X}^+ \leftarrow \text{DCT2}\{\mathbf{x}^+\}$ 
9:      $\mathbf{x}^- \leftarrow \mathbf{x}$ 
10:     $x^-(m,n) \leftarrow x^-(m,n) - \Delta$ 
11:     $\mathbf{X}^- \leftarrow \text{DCT2}\{\mathbf{x}^-\}$ 
12:     $g(m,n) \leftarrow \|\mathbf{X}^+\|_1 - \|\mathbf{X}^-\|_1$ 
13:    Select  $r$  pixels  $(m,n) \in \mathbb{N}_A$  with highest  $|g(m,n)|$ 
14:    Add selected pixels to set  $\mathbb{N}_x$ 
15:    Remove selected pixels from set  $\mathbb{N}_A$ 
16:     $\mathbf{x} \leftarrow \text{GRADREC}(\mathbf{x}, \mathbb{N}_A)$  ▷ Algorithm 1
17: until the sparsity is not significantly changed
18: Optionally, perform sparsification of the reconstructed image
19: return  $\mathbf{x}, \mathbb{N}_A$ 

```

4 Experimental Results

Application of the algorithm is explained on the corrupted image “Lena”. The image is of size $M \times N = 512 \times 512$. The quality factor (QF) used is $QF = 25$ because it results in a sparser image than the standard 50% and the visual degradation will not be significant. Reconstruction of images “Lena” and “Peppers” affected with a combination of salt and pepper and uniform noise is demonstrated in the last subsection, as well. A detailed performance analysis of the proposed algorithm is done in the next section using eight common images.

4.1 Grayscale images

The original sparse image is presented in Fig. 2 (top left), together with the noisy image with 12.5% uniform noise in Fig. 2 (top right). The noisy pixels are

uniform and in the range between 20 and 230. The image reconstructed using the presented algorithm is shown in Fig. 2 (middle left). Since the original image is sparsified, the reconstructed image is also sparsified at the end of the process. Note that the presented gradient-based algorithm minimizes L_1 -norm of the signal transform coefficients and, in real cases, it does not preserve sparsity in a strict sense. Sparsification of the resulting image is done in the same way as the sparsification of the input image using equations (10) and (11). The image reconstructed using the additional step of sparsification is shown in Fig. 2 (middle right). The same image was corrupted with 12.5% of a Gaussian noise with mean 100 and standard deviation 50. The noisy and reconstructed image are shown in Fig. 2 (bottom). It can be seen that the reconstruction was successfully done for both images, corrupted either with a uniform or a Gaussian noise.

4.2 Color images

The reconstruction will be done using the RGB version of image “Lena” of the same size 512×512 . The original sparse image and the noisy image are shown in Fig. 3 (top). The image has 12.5% of noisy pixels in the range between 20 and 230. The reconstructed image and the sparsified reconstructed image are shown in Fig. 3 (bottom).

4.3 Combination of noises

The reconstruction of the image “Lena” with the combination of salt and pepper and uniform noise is presented. The noisy image is shown in Fig. 4 (top left). The image contains 50% of noisy pixels, with 10% of them being the uniform noise. The image is not strictly sparse, which means that it was not sparsified (as in previous sections). When no sparsification step is used, the 32×32 gives a faster reconstruction. Using larger blocks, the sparsity is more emphasized and the reconstruction is still computationally feasible on ordinary computers. The reconstruction using proposed method with 32×32 block size is shown in Fig. 4 (top right). For comparison, the reconstruction using median filter of size 3×3 and 5×5 are shown in Fig. 4 (middle). The algorithm is compared with a total variation L1 (TV-L1) model with a primal-dual algorithm [26,27]. The code used for the TV-L1 reconstruction algorithm can be found in [28]. The proposed algorithm was also compared with a two-stage adaptive algorithm presented in [29]. The reconstruction results using these two algorithms are shown in Fig. 4 (bottom).

The noisy color image, and the reconstructed one using the gradient-descent algorithm are presented in Fig. 5 (top). Note that the impulsive noise in this case is randomly positioned in each of the three channels separately. For comparison, the image was reconstructed using the marginal median filter of size 3×3 and 5×5 and the reconstruction images are shown in Fig. 5



Fig. 2 Reconstruction of sparse images corrupted by different noise types: Sparse image (top left); Noisy image corrupted with uniform noise (top right); Reconstructed image corrupted with uniform noise (middle left); Reconstructed sparsified (middle right); Noisy image corrupted with Gaussian noise (bottom left); Reconstructed image corrupted with Gaussian noise (bottom right)



Fig. 3 Reconstruction of color image corrupted with 12.5% uniform noise: Sparse image (top left); Noisy image (top right); Reconstructed image (bottom left); Reconstructed sparse image (bottom right)

(middle). The marginal median filters each color channel separately. The reconstruction using the TV-L1 and the two-stage adaptive algorithms is shown in Fig. 5 (bottom). The algorithm was also tested on the image “Peppers”, with the same characteristics. The original and noisy images are shown in Fig. 6 (top). The reconstruction using the gradient algorithm and the reconstruction of the image using the 5×5 marginal median filter are presented in Fig. 6 (middle). In Fig. 6 (bottom), the reconstruction with the other two algorithms is presented.

5 Performance Analysis

The performance of the algorithm will be examined using the SSIM index as well as MAE, MSE, and PSNR with respect to the original image. We



Fig. 4 Reconstruction of image corrupted with 50% combined noise: Noisy image (top left); Reconstructed image using the proposed method (top right); Reconstructed image using median filters (middle); Reconstructed image using TV-L1 algorithm (bottom left); Reconstructed image using two-stage adaptive method (bottom right)



Fig. 5 Reconstruction of color image corrupted with 50% combined noise: Noisy image (top left); Reconstructed image using the proposed method (top right); Reconstructed image using the 3×3 and 5×5 marginal median filters (middle); Reconstructed image using the TV-L1 algorithm (bottom left); Reconstructed image using the two-stage adaptive method (bottom right)

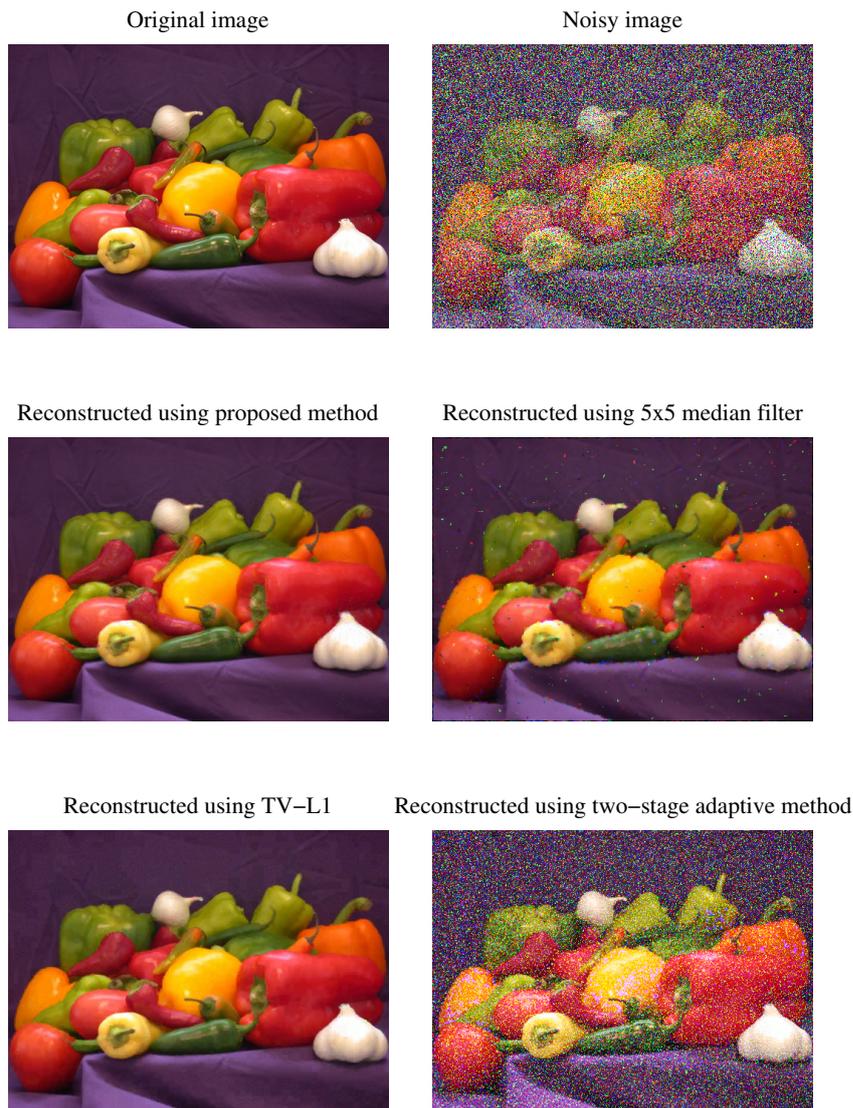


Fig. 6 Reconstruction of color image corrupted with 50% combined noise: Original image (top left); Noisy image (top right); Reconstructed image using the proposed method (middle left); Reconstructed image using the 5×5 marginal median filter (middle right); Reconstructed image using the TV-L1 algorithm (bottom left); Reconstructed image using the two-stage adaptive method (bottom right)

Table 1 SSIM index between original and reconstructed image for various quality factor QF and percentage of corrupted pixels

QF	after reconstruction				after additional sparsification			
	12.5%	25%	37.5%	50%	12.5%	25%	37.5%	50%
5	0.9958	0.9857	0.9123	0.6441	0.9991	0.9938	0.9252	0.6713
10	0.9965	0.9860	0.9196	0.6357	0.9985	0.9905	0.9217	0.6431
25	0.9970	0.9871	0.9225	0.6314	0.9981	0.9892	0.9207	0.6402
50	0.9962	0.9851	0.9175	0.6209	0.9970	0.9860	0.9152	0.6239
75	0.9955	0.9820	0.9159	0.6120	0.9959	0.9822	0.9136	0.6138
90	0.9929	0.9777	0.9114	0.5992	0.9925	0.9770	0.9104	0.5987

will declare the original image $x(m, n) \rightarrow \mathbf{x}_o$ and the reconstructed image as $x_R(m, n) \rightarrow \mathbf{x}_r$. The SSIM index is a function of luminance, contrast and structure comparison between two images. It is introduced in [33]. The SSIM index is defined as

$$\text{SSIM}(\mathbf{x}_o, \mathbf{x}_r) = \frac{(2\mu_{x_o}\mu_{x_r} + c_1)(2\sigma_{x_o x_r} + c_2)}{(\mu_{x_o}^2 + \mu_{x_r}^2 + c_1)(\sigma_{x_o}^2 + \sigma_{x_r}^2 + c_2)} \quad (20)$$

where the values μ_{x_o} , μ_{x_r} are the mean values of the images, $\sigma_{x_o x_r}$ is the covariance between the two images, $\sigma_{x_o}^2$, $\sigma_{x_r}^2$ are the variances of the two images, and c_1, c_2 are used for stabilization. The SSIM value is in range between 0 and 1, with 1 being very similar and 0 is for not similar.

SSIM index is calculated when the noisy grayscale image "Lena" from Fig. 2 (top right) is reconstructed. It is tested on different number of DCT components obtained as a result of applying different quantization matrices with $QF \in \{5, 10, 25, 50, 75, 90\}$, and percent of corrupted samples $\{12.5\%, 25\%, 37.5\%, 50\%\}$. The algorithm does not assume the knowledge of the number and positions of the corrupted pixels.

Table 1 presents the SSIM index values for different quality factors in terms of the percent of corrupted pixels. The SSIM is calculated after reconstruction and after additional sparsification step. Sparsification leads to improved smoothness in the reconstructed image.

The MAE is calculated as

$$MAE(\mathbf{x}_o, \mathbf{x}_r) = \text{mean}(\text{mean}(|\mathbf{x}_o - \mathbf{x}_r|)). \quad (21)$$

The MAE is used to compare the performances of the reconstruction of noisy images using the algorithm presented. Table 2 presents the MAE values calculated after the reconstruction and after sparsification of the reconstructed image for various quality factor and percent of corrupted pixels.

Two more parameters will be used for the reconstruction comparison. The MSE is calculated as

$$MSE(\mathbf{x}_o, \mathbf{x}_r) = \text{mean}(\text{mean}(|\mathbf{x}_o - \mathbf{x}_r|^2)), \quad (22)$$

Table 2 MAE values for varying percent of corrupted pixels and quality factor

QF	after reconstruction				after additional sparsification			
	12.5%	25%	37.5%	50%	12.5%	25%	37.5%	50%
5	0.4055	1.0847	3.2756	10.8029	0.1376	0.4559	2.7673	10.4414
10	0.3831	1.0540	3.1820	11.6284	0.2645	0.7985	3.2111	11.5774
25	0.3693	1.0759	3.2701	12.2769	0.4273	1.1505	3.6485	12.3009
50	0.4158	1.1712	3.4486	12.7903	0.6273	1.4565	3.9675	12.9114
75	0.4727	1.3113	3.6015	13.0952	0.8242	1.7840	4.2034	13.2539
90	0.6072	1.5378	3.8524	13.7471	1.1595	2.2244	4.4407	14.0689

Table 3 Performance measures for different noise types, percent of corrupted pixels and optional sparsification after reconstruction

Noise type	Level	Sparsification	MAE	SSIM	MSE	PSNR
Salt & pepper	50.0%	no	0.5728	0.9911	3.4448	42.7592
Uniform	12.5%	no	0.3803	0.9965	5.9053	40.4184
Uniform	12.5%	yes	0.4394	0.9978	3.1721	43.1173
Gaussian	12.5%	no	0.3557	0.9952	6.7221	39.8557
Gaussian	12.5%	yes	0.6816	0.9953	4.8952	41.2331

Table 4 Error measures after reconstruction of the noisy image with 50% corrupted pixels by using the proposed method, median filtering, total variation L1, and two-stage adaptive restoration method

Measure	Proposed	Median 3×3	Median 5×5	TV-L1	Two-stage
SSIM	0.9835	0.4215	0.8831	0.9127	0.7554
MAE	0.7657	5.9514	3.1872	2.2110	2.4174
MSE	4.9124	22.9010	19.7015	16.5719	16.8124
PSNR	41.2179	34.5323	35.1858	35.9371	35.8745

while the PSNR for an 8-bit image is defined as

$$PSNR(\mathbf{x}_o, \mathbf{x}_r) = 10 \log_{10} \left(\frac{255^2}{MSE(\mathbf{x}_o, \mathbf{x}_r)} \right). \quad (23)$$

Table 3 presents the error values for the reconstructed images using different noise types. The comparison among the algorithms using 50% combined noise is shown in Table 4. Grayscale image ‘‘Lena’’ was used for the comparison. The comparison among the algorithms for eight test images with 50% pixels corrupted by combined noise, is presented in Table 5.

The proposed method is based on the detection and compressive sensing reconstruction of the corrupted pixels. In theory, if the compressive sensing conditions are met, the reconstruction is exact. Other methods are based on filtering or interpolation, and in general they produce approximations of the original pixels.

Table 5 PSNR and SSIM for 8 test images. The results are obtained by the proposed, two-stage (2-stage) adaptive [29] and total variation L1 [26,27] methods

Test image	PSNR			SSIM		
	Proposed	2-stage	TV-L1	Proposed	2-stage	TV-L1
Lena	41.22	35.87	35.94	0.9835	0.7544	0.9127
Lifting body	43.92	35.90	40.15	0.9860	0.7295	0.9438
Boat	39.33	34.15	34.41	0.9728	0.7312	0.8524
Butterfly	39.22	36.20	35.04	0.9768	0.8100	0.8840
Camera	36.54	36.36	33.01	0.9408	0.8102	0.7884
Pout	45.87	39.59	39.46	0.9802	0.6272	0.9189
Peppers	42.74	39.84	38.58	0.9877	0.6229	0.9529
Tissue	32.44	30.92	29.35	0.9101	0.8566	0.7303

6 Conclusions

The gradient-based algorithm for reconstruction of noisy images with noise being within the range of the available pixels is considered in this paper. The basic algorithm form implies that we know the number and positions of the uncorrupted pixels (or the noise is very high so that the corrupted pixels are distinguishable from the uncorrupted ones). In this paper we proposed an algorithm for detection of corrupted pixel positions and their reconstruction. It has been shown that the image can be reconstructed successfully without any knowledge of the corrupted pixels, except that these pixels degrade image sparsity. We also have used modifications of the initial algorithm by adding a sparsification step at the end of the reconstruction process and by introducing partially overlapping blocks, to avoid edge effects.

Denosing of sparse images using the presented algorithm, when the image is corrupted with combined noise, is presented as well. The considered noises were impulsive with values within the pixel range, also combined with a salt and pepper noise. It is shown that the algorithm can produce successful reconstructions of these images, both grayscale and color ones. The reconstruction was successful even in the cases when the original image is not strictly sparse. The results are compared with the standard median-based filtering, a CS-based algorithm that uses total variations L1 (dual prime approach), and a two-stage adaptive algorithm. The proposed gradient-based algorithm overperformed the considered algorithms for a set of standard images that we used in the reconstruction.

Future work could include analysis of reconstruction accuracy when images are approximately sparse as well as possible parallelization of the presented detection and reconstruction algorithm. Application of the presented algorithm to radar, sonar and medical imaging will be considered as well, taking into account their specific forms.

References

1. Baraniuk RG (2007) Compressive sensing. *IEEE Signal Processing Magazine* 24(4):118–121
2. Candes EJ, Wakin M (2008) An introduction to compressive sampling. *IEEE Signal Processing Magazine* 25(2):21–30
3. Donoho DL (2006) Compressive sensing. *IEEE Transactions on Information Theory* 52(4):1289–1306
4. Candes EJ, Romberg J, Tao T (2006) Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52(2):489–509
5. Candes EJ (2008) The restricted isometry property and its implications for compressive sensing. *Comptes Rendus Mathematique* 346(9-10):589–592
6. Stanković S, Orović I, Stanković L (2014) An automated signal reconstruction method based on analysis of compressive sensed signals in noisy environment. *Signal Processing* 104:43–50
7. Stanković L, Stanković S, Amin M (2014) Missing samples analysis in signals for applications to L-estimation and compressive sensing. *IEEE Signal Processing Letters* 94:401–408
8. Rauhut H, Schnass K, Vandergheynst P (2008) Compressed sensing and redundant dictionaries. *IEEE Transactions on Information Theory* 54(5):2210–2219
9. Fornasier M, Rauhut H (2008) Iterative thresholding algorithms. *Applied and Computational Harmonic Analysis* 25(2):187–208
10. Stanković L (2015) Digital signal processing with selected topics. CreateSpace Independent Publishing Platform, An Amazon.com Company
11. Stanković S, Orović I, Sejdić E (2012) Multimedia signals and systems. Springer-Verlag, New York
12. Elad M (2010) Sparse and redundant representations: From theory to applications in signal and image processing. Springer, New York
13. Donoho D, Elad M, Temlyakov V (2006) Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory* 52:6–18
14. Stanković L, Daković M, Vujović S (2014) Adaptive variable step algorithm for missing samples recovery in sparse signals. *IET Signal Processing* 8(3):246–256
15. Figueiredo MAT, Nowak RD, Wright SJ (2007) Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing* 1(4):586–597
16. Stanković I, Orović I, Stanković S (2014) Image reconstruction from a reduced set of pixels using a simplified gradient algorithm. 22nd Telecommunications Forum TELFOR, Belgrade, Serbia
17. Wang X-L, Wang C-L, Zhu J-B, Liang D-N (2011) Salt-and-pepper noise removal based on image sparse representation. *Optical Engineering* 50(9), article ID 097007
18. Koivisto P, Astola J, Lukin V, Melnik V, Tsymbal O (2003) Removing impulse bursts from images by training-based filtering. *EURASIP Journal in Advances in Signal Processing*
19. Huang S, Zhu J (2010) Removal of salt-and-pepper noise based on compressed sensing. *Electronics Letters* 46(17):1198–1199
20. Guo D, Qu X, Du X, Wu K, Chen X (2014) Salt and pepper noise removal with noise detection and a patch-based sparse representation. *Advances in Multimedia 2014*, article ID 682747
21. Lukin VV, Oktem R, Ponomarenko N, Egiazarian K (2007) Image filtering based on discrete cosine transform. *Telecommunications and Radio Engineering* 66(18):1685–1701
22. Lukin VV, Ponomarenko N, Egiazarian K, Astola J (2008) Adaptive DCT-based filtering of images corrupted by spatially correlated noise. *Proc. SPIE Conference Image Processing: Algorithms and Systems VI* 6812
23. Studer C, Kuppinger P, Pope G, Bolcskei H (2012) Recovery of sparsely corrupted signals. *IEEE Transactions of Information Theory* 58(5):3115–3130
24. Srinivasan KS, Ebenezer, D (2007) A new fast and efficient decision based algorithm for removal of high-density impulse noises. *IEEE Signal Process. Lett.* doi:10.1109/LSP.2006.884018

25. Luo W (2006), An efficient detail-preserving approach for removing impulse noise in images. *IEEE Signal Processing Letters*, doi:10.1109/LSP.2006.873144
26. Caselles V, Chambolle A, Novaga M (2011) Total variation in imaging. *Handbook of Mathematical Methods in Imaging*, Springer, New York, 1016–1057
27. Zhang B, Zhu Z, Wang S (2016) A simple primal-dual method for total variation image restoration. *Journal of Visual Communication and Image Representation* 38(C):814–823
28. Lourakis M (2016), TV-L1 Image Denoising Algorithm. <https://www.mathworks.com/matlabcentral/fileexchange/57604-tv-l1-image-denoising-algorithm>
29. Ramadan Z (2012), Efficient Restoration Method for Images Corrupted with Impulse Noise. *Circuits, Systems, and Signal Processing*, 31(4):1397–1406
30. Djurovic I (2016), BM3D filter in salt-and-pepper noise removal. *EURASIP Journal on Image and Video Processing* 2016(13), doi: 10.1186/s13640-016-0113-x
31. Stanković L, Daković M, Vujović S (2016) Reconstruction of sparse signals in impulsive disturbance environment. *Circuits, Systems and Signal Processing* 35, in print.
32. Stanković I, Orović I, Stanković S, Daković M (2016) Iterative denoising of sparse images. 39th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO, Opatija, Croatia
33. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4):600–612