Applied Mathematics and Computational Intelligence
Volume 14, No. 2, 2025 [95 − 118]

A Three-term PRP-DL Method Modification with Application in Image Denoising Problem

Muhammad Aqiil Iqmal Ishak[1], Nurin Athirah Azmi[2*], Siti Mahani Marjugi[3]

[1,2*,3]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Received: 18 October 2024
Revised: 10 December 2024
Accepted: 26 December 2024

ABSTRACT

Image denoising poses a critical challenge due to the impact of noise on image quality and the need to preserve essential details. This study introduces a hybrid Polak-Ribiére-Polyak (PRP)-Dai-Liao (DL) conjugate gradient method with a modified scalar to improve the performance of denoising algorithms on large-scale images. The proposed method involves modifying the scalar in the PRP-DL conjugate gradient method, thereby enhancing algorithmic efficiency, especially in handling large-scale problems. Convergence analysis under the standard Wolfe-Powell line search is established, and numerical results demonstrate that the proposed method is more efficient and robust than existing conjugate gradient methods. The application of the method to image denoising with various noise levels and window sizes confirms its capability to effectively remove noise while preserving image details. Overall, this modified conjugate gradient method shows promise for practical applications in image denoising problem.

Keywords: modified hybrid conjugate gradient method, image denoising, large-scale optimization, global convergence, line search, noise reduction.

1 INTRODUCTION

The conjugate gradient (CG) method is a well-established iterative algorithm, valued for its computational efficiency and adaptability to various problem settings. Introduced by Eduard Stiefel and Magnus [1], this method is celebrated for its memory efficiency and rapid convergence, making it applicable across numerous research domains [2]. In the CG method, each iteration improves an initial estimate, generating a sequence represented as x_k:

$$x_{k+1} = x_k + \alpha_k d_k, k = 0, 1, 2, \dots \tag{1}$$

where x_k is the current iteration point, d_k is the search direction, and $\alpha_k > 0$ is the step size, determined by a line search. The standard Wolfe-Powell line search, which extends the standard Wolfe-Powell method, includes additional conditions to ensure more reliable convergence:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \tag{2}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{3}$$

where scalar g_k is the derivative of $f(x)$ at the point of x_k, g_k^T is the transpose of g_k and δ is a very small positive value, $0 < \delta < \sigma < 1$.

For the line search algorithm to ensure convergence, the search direction d_k must satisfy the sufficient descent condition:

$$g_k^T d_k \leq -c||g_k||^2, \tag{4}$$

where $c > 0$ is a constant. The search direction, d_k is determined by :

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1. \end{cases} \tag{5}$$

where β_k is known as the CG coefficient.

There are several formulas available for β_k, such as the Fletcher-Reeves (FR) by [3], Polak-Ribière-Polyak (PRP) by [4], Hestenes-Stiefel (HS) by [5], Liu-Storey by [6] (LS), Dai-Yuan (DY) by [7], and Conjugate Descent (CD) by [8] as stated below:

$$\beta_k^{\text{HS}} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\text{PRP}} = \frac{g_k^T y_{k-1}}{g_{k-1}^T g_{k-1}}, \quad \beta_k^{\text{LS}} = \frac{g_k^T y_{k-1}}{-d_{k-1}^T g_{k-1}}$$

$$\beta_k^{\text{DY}} = \frac{g_k^T g_k}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{\text{FR}} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}, \quad \beta_k^{\text{CD}} = \frac{g_k^T g_k}{-d_{k-1}^T g_{k-1}}$$

where $y_{k-1} = g_k - g_{k-1}$.

The hybrid CG method integrates various classical CG approaches to leverage their respective strengths. It is widely regarded as being more efficient and robust compared to classical CG methods. Developed to address the dual challenges of computational efficiency and memory limitations in large-scale optimization, this hybrid approach aims to accelerate convergence while minimizing memory usage. The three-term CG method, an extension of the classical CG approach, introduces a novel three-term recursion formula. This enhanced recursion introduces greater complexity but also improves convergence speed and optimization efficiency compared to two-term methods. Although the three-term method involves more intricate calculations per iteration, its potential for faster convergence and superior performance on certain optimization problems makes it a valuable tool for optimization practitioners. In the study by [9], the numerical performance of a new hybrid three-term CG method incorporating a modified secant condition (HTTCGSC) was evaluated and compared with the modified three-term Hestenes–Stiefel (MTTHS) method [10]. Results demonstrated that HTTCGSC achieved superior performance, solving test problems with the lowest iteration count, fewer function and gradient evaluations, and minimal CPU time. Additionally, [11] examined a hybrid CG method (HCGM) combining the PRP and FR methods, which demonstrated sufficient descent under appropriate line search conditions. The MZZ method proposed by [12] also ensures the sufficient descent condition. The newly proposed modified hybrid three-term (MTT) CG method in this study will be compared with MTTHS and MZZ in the context of unconstrained optimization.

1.1 Image denoising problems

A two-phase approach was employed to eliminate salt and pepper noise from images which proposed in the study by Chan et al. [13]. The first phase involved identifying noisy pixels using an adaptive median filter. Meanwhile, the suggested NM algorithm was utilized to solve the objective function defined by equation (6) in the second phase, effectively removing noise points and restoring the image by minimizing the following function:

$$\min f_\alpha(\nu) := \sum_{(m,n)\in\mathbb{N}} \left(\lambda S^1_{m,n} + \frac{\lambda}{2} S^2_{m,n} + l_1 \right), \tag{6}$$

where

$$S^1_{m,n} = \sum_{(p,q)\in\mathbb{V}_{m,n}\setminus\mathbb{N}} \varphi_\alpha(\nu_{mn} - \eta_{pq}), \quad S^{m,n}_2 = \sum_{(p,q)\in\mathbb{V}_{mn}\cap\mathbb{N}} \varphi_\alpha(\nu_{mn} - \nu_{pq}).$$

$$\mathbb{N} = \{(m,n) \in \mathbb{A} : \overline{\eta}_{mn} \neq \eta_{mn}, \eta_{mn} = s_{\min} \text{ or } S_{\max}\}$$

$\mathbb{N}$ denotes as the candidate set of noisy pixels in the phase one. $(m,n) \in \mathbb{A} = \{1,2,\dots,P\} \times \{1,2,\dots,Q\}$ is the original image pixel set x of size $P \times Q$. Meanwhile, the range value of x is $[s_{\min}, S_{\max}]$, while $\nu = [\nu_{m,n}]_{(m,n)\in\mathbb{N}}$ is a column vector arrange in lexicographic order, in which $\eta_{m,n}$ is the observed pixel value at the coordinate (m,n). Furthermore, $\mathbb{V}_{m,n}$ denotes the set of four neighbors at the pixel point $(m,n) \in \mathbb{A}$, excluding (m,n), i.e., $\mathbb{V}_{m,n} = \{(m,n-1),(m,n+1),(m-1,n),(m+1,n)\}$. Moreover, l_1 represents the data-fitting component, $\phi_\alpha(a) = \sqrt{a^2 + a}$ where $a > 0$, serves as the edge-preserving function [14], while λ is an appropriate parameter.

Non-smooth optimization problems can be computationally expensive to solve. To address this, by eliminating the l_1 term in equation (6), the function is reformulated into a smooth UOP, making it more computationally efficient (refer to [15] for further information).

$$\min f_\alpha(\nu) := \sum_{(m,n)\in\mathbb{N}} \left(S^1_{m,n} + \frac{1}{2} S^2_{m,n} \right). \tag{7}$$

The experimental findings by Cai et al. [15] indicate that the presence or absence of a fitting term does not significantly impact the effectiveness of image restoration. Efficient removal of salt-and-pepper noise requires denoising algorithms capable of distinguishing corrupted pixels from uncorrupted ones while preserving the original features of the image. This study introduces a modified hybrid Polak-Ribière-Polyak (PRP)-Dai-Liao (DL) conjugate gradient method, which incorporates an enhanced scalar to improve denoising performance, especially for images affected by salt-and-pepper noise. The method is designed to address this specific type of noise, ensuring robust performance even for high noise levels.

This research introduces a new modified hybrid PRP-DL conjugate gradient algorithm specifically designed for unconstrained optimization. The three-term CG method serves as the foundation due to its efficacy and popularity in optimization. Inspired by [9], this study introduces scalar modifications based on [11], incorporating new strategies to enhance convergence speed, stability, and algorithmic performance.

The structure of this paper is as follows: Section 2 presents the underlying principles of the algorithm modification. Section 3 discusses the sufficient descent and global convergence properties under the standard Wolfe-Powell line search. Section 4 details the numerical results of the proposed method in image denoising applications. Finally, Section 5 summarizes the conclusions.

2 MODIFICATIONS

The primary sources of our motivation are the works of [9] and [11], wherein

$$\beta_k^{\mathrm{N}} = \frac{g_k^T(y_{k-1} - ts_{k-1})}{\max\{y_{k-1}^T s_{k-1}, ||g_{k-1}||^2\}}, \tag{8}$$

$$\delta_k^{\mathrm{N}} = \frac{g_k^T s_{k-1}}{\max\{y_{k-1}^T s_{k-1}, ||g_{k-1}||^2\}}. \tag{9}$$

The parameter t is defined as $t = \max\{\bar{t}, \frac{||y_k||^2 g_{k-1}^T s_k}{z_k}\}$ Some modification on (8) and (9) have been made. The parameter $s_{k-1}^T g_{k-1}$ has been inserted in both numerator and denominator of (8) and (9) while referring on the study of [11]. The new CG coefficient is defined as below,

$$\beta_k^{\mathrm{MTT}} = \frac{g_k^T(s_{k-1}^T y_{k-1} g_{k-1} - t||s_{k-1}||^2 g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}}, \tag{10}$$

$$\delta_k^{\mathrm{MTT}} = \frac{||s_{k-1}||^2 g_k^T g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}}. \tag{11}$$

According to [16], $\bar{t} = 0.1$ is an appropriate choice. Therefore, $t = \max\{0.1, \frac{||y_k||^2 g_{k-1}^T s_k}{z_k}\}$ is assigned.

Algorithm 2.1: Modified three-term(MTT) PRP-DL conjugate gradient method

Step 1: Let $k = 0$. Choose a starting point $x_0 \in \mathbb{R}^n$. Obtain $g(x_0)$ and assign $d_0 = -g_0$.
Step 2: If $||g_k|| \leq \varepsilon$, $\varepsilon = 10^{-6}$, then stop, otherwise proceed to the next step.
Step 3: Determine the step size α_k along the direction d_k by using the standard Wolfe-Powell line search stated in (2) and (3).
Step 4: Let $x_{k+1} = x_k + \alpha_k d_k$ to compute the new iterative point.
Step 5: Calculate the search direction d_k by using:

$$d_k^{\mathrm{MTT}} = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k^{\mathrm{MTT}} s_{k-1} - \delta_k^{\mathrm{MTT}} y_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where $s_{k-1} = \alpha_{k-1} d_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

Step 6: Set $k = k + 1$ and repeat Step 1.

Algorithm 2.2: Image Denoising Procedure

Step 1: Input the original images (e.g., Roro, Lung, and Hill).
Step 2: Adjust the noise ratio (e.g., 10%, 30%, and 50%) and maximum window size (e.g., 38, 41, and 43).

Step 3: Extract the image into a matrix format.
Step 4: Compute and record the results using the MTT method, evaluating CPU Time, PSNR, RE, and SSIM.
Step 5: Repeat Steps 1 to 4 for the MTTHS and MZZ methods.
Step 6: Compare the results of the MTT method with those of the MTTHS and MZZ methods.

3 CONVERGENCE ANALYSIS

The following discussion will clarify that, regardless of the line search strategy used, Algorithm 2 consistently exhibits the sufficient descent property.

Lemma 1. *Algorithm 2 generated the sequence $\{d_k^{MTT}\}$ independent on any line search, and it always holds that:*

$$g_k^T d_k^{MTT} \leq -||g_k||^2, \forall k \geq 0. \tag{12}$$

Proof. When $k = 0$, then $d_0 = -g_0$, and it holds that $g_0^T = -||g_0||^2$. For $k \geq 1$, the subsequent inequality obtained according to the definition of d_k^{MTT}:

$$\begin{aligned}
g_k^T d_k^{\mathrm{MTT}} &= -||g_k||^2 + \beta_k^{\mathrm{MTT}} g_k^T s_{k-1} - \delta_k^{\mathrm{MTT}} d_k^T y_{k-1} \\
&= -||g_k||^2 + \frac{g_k^T (s_{k-1}^T y_{k-1} g_{k-1} - t||s_{k-1}||^2 g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T s_{k-1} \\
&- \frac{||s_{k-1}||^2 g_k^T g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T y_{k-1} \\
&= -||g_k||^2 + \frac{g_k^T ||s_{k-1}||^2 y_{k-1} g_{k-1} g_k^T}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \\
&- \frac{t||s_{k-1}||^2 g_{k-1} g_k^T s_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \\
&- \frac{||s_{k-1}||^2 g_k^T g_{k-1} g_k^T y_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \\
&= -||g_k||^2 - \frac{t||s_{k-1}||^2 g_{k-1} g_k^T s_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \\
&\leq -||g_k||^2,
\end{aligned}$$

The final inequality is valid for $t \geq 0$. Consequently, (12) is satisfied, hence the proof completed.

□

Lemma (1) demonstrates that, independent of the line search method, the new direction maintains the sufficient descent property. Additionally, a conjugate condition is crucial for achieving optimal numerical performance. In the case of MTT, this is ensured by the design of the direction d_k^{MTT}.

$$
\begin{aligned}
(d_k^{\mathrm{MTT}})^T y_{k-1} &= -g_k^T y_{k-1} + \frac{g_k^T(s_{k-1}^T y_{k-1} g_{k-1} - t||s_{k-1}||^2 g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} y_{k-1}^T s_{k-1} \\
&\quad - \frac{||s_{k-1}||^2 g_k^T g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} ||y_{k-1}||^2 \\
&= -g_k^T y_{k-1} + \frac{||s_{k-1}||^2 y_{k-1} g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T y_{k-1}^T \\
&\quad - \frac{y_{k-1}^T (t||s_{k-1}||^2 g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T s_{k-1} \\
&\quad - \frac{||y_{k-1}||^2 (s_{k-1} g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T s_{k-1} \\
&\leq -\frac{t y_{k-1}^T ||s_{k-1}||^2 g_{k-1} + ||y_{k-1}||^2 s_{k-1} g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} g_k^T s_{k-1}.
\end{aligned} \tag{13}
$$

From (13), it holds that the new direction d_k^{MTT} satisfies DL conjugate condition, in an extent form in which $(d_k^{\mathrm{MTT}})^T y_{k-1} \leq -t_1 g_k^T s_{k-1}$ where $t_1 = -\frac{t y_{k-1}^T ||s_{k-1}||^2 g_{k-1} + ||y_{k-1}||^2 s_{k-1} g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}}$. In fact, if we adopt the line search technique which results in $||s_{k-1}||^2 g_{k-1}^T y_{k-1} \geq 0$, then it holds that $t_1 = -\frac{t y_{k-1}^T ||s_{k-1}||^2 g_{k-1} + ||y_{k-1}||^2 s_{k-1} g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} > 0$.

This section delves into the convergence properties of β_k^{MTT}. It is assumed that $g_k \neq 0$ for all k; a zero value for g_k would indicate a stationary point. The convergence of nonlinear conjugate gradient algorithms is typically proven under the following assumptions.

Assumption 1. *The level set $\mathbb{T} := \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded where x_0 is the initial point, then it means there exist a constant $X > 0$ in such a way that:*

$$||x|| \leq X, \quad \forall x \in \mathbb{T}. \tag{14}$$

Assumption 2. *In some neighborhood $\mathbb{N}$ of $\mathbb{T}$, the gradient of function $f(x)$ and $g(x)$ known as Lipschitz continuous, which means there exists a constant $L > 0$ such that:*

$$||g(x) - g(y)|| \leq L||x - y||, \quad \forall x, y \in \mathbb{N}. \tag{15}$$

It should be noted that, according to Assumption 1 and Assumption 2, there is a positive constant G that satisfies the following condition:

$$||g(x)|| \leq G, \quad \forall x \in \mathbb{T}. \tag{16}$$

In the following analysis, the sequence d_k^{MTT} produced by Algorithm 2 is bounded will be demonstrated.

Lemma 2. *Consider the condition $0 < t \leq T$, and assume that both Assumption 1 and Assumption 2 are satisfied. For any line search technique, consider the sequence $\{d_k^{MTT}\}$ generated by Algorithm 2. If the objective function f exhibits uniform convexity on the set $\mathbb{T}$, it can be concluded that $||d_k^{MTT}||$ is bounded.*

Proof. Given that the function f exhibits uniform convexity on the set $\mathbb{N}$, it follows that for any x, $y \in \mathbb{N}$, the following inequality holds:

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \tilde{u}||x - y||^2,$$

where $\tilde{u} > 0$ is the uniform convexity parameter. In particular, when assigning $x = x_k$ and $y = x_{k-1}$, the following equation is true:

$$||s_{k-1}||^2 g_{k-1}^T y_{k-1} \geq \tilde{u}||s_{k-1}||^2 > 0.$$

In the subsequent analysis, the boundedness of the parameters β_k^{MTT} and δ_k^{MTT} will be proved. According to their respective definitions, observe that:

$$\begin{aligned}
|\beta_k^{\text{MTT}}| &= \left| \frac{g_k^T (s_{k-1}^T y_{k-1} g_{k-1} - t||s_{k-1}||^2 g_{k-1})}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \right| \\
&\leq \frac{||g_k||(||s_{k-1}^T|| ||y_{k-1}|| ||g_{k-1}|| + t||s_{k-1}||^2 ||g_{k-1}||)}{|\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}|} \\
&\leq \frac{||g_k||(||s_{k-1}^T|| ||y_{k-1}|| ||g_{k-1}|| + t||s_{k-1}||^2 ||g_{k-1}||)}{||s_{k-1}||^2 g_{k-1}^T y_{k-1}} \\
&\leq \frac{(L+T)||s_{k-1}||}{\tilde{u}||s_{k-1}||^2} ||g_k|| \\
&= \frac{(L+T)}{\tilde{u}} \frac{||g_k||}{||s_{k-1}||}.
\end{aligned}$$

$$\begin{aligned}
|\delta_k^{\text{MTT}}| &= \left| \frac{||s_{k-1}||^2 g_k^T g_{k-1}}{\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}} \right| \\
&\leq \frac{||s_{k-1}||^2 ||g_k|| ||g_{k-1}||}{|\max\{||s_{k-1}||^2 g_{k-1}^T y_{k-1}, ||g_{k-1}||^2 g_{k-1}^T s_{k-1}\}|} \\
&\leq \frac{||s_{k-1}||^2 ||g_k|| ||g_{k-1}||}{||s_{k-1}||^2 g_{k-1}^T y_{k-1}} \\
&\leq \frac{||s_{k-1}||^2 ||g_k|| ||g_{k-1}||}{\tilde{u}||s_{k-1}||^2} \\
&= \frac{1}{\tilde{u}} ||g_k|| ||g_{k-1}||.
\end{aligned}$$

Hence, according to the definition of d_k^{MTT} :

$$\begin{aligned}
||d_k^{\mathrm{MTT}}|| &= \left|\left|-g_k + \beta_k^{\mathrm{MTT}} s_{k-1} + \delta_k^{\mathrm{MTT}} y_{k-1}\right|\right| \\
&\leq ||g_k|| + |\beta_k^{\mathrm{MTT}}|||s_{k-1}|| + |\delta_k^{\mathrm{MTT}}|||y_{k-1}|| \\
&\leq ||g_k|| + \frac{(L+T)}{\tilde{u}} \frac{||g_k||}{||s_{k-1}||}||s_{k-1}|| + \frac{1}{\tilde{u}}||g_k||||g_{k-1}||||y_{k-1}|| \\
&\leq ||g_k|| + \frac{(L+T)}{\tilde{u}}||g_k|| + \frac{L}{\tilde{u}}||g_k|| \\
&= \left(1 + \frac{2L+T}{\tilde{u}} + \frac{L}{\tilde{u}}\right)||g_k|| \\
&\leq \left(1 + \frac{2L+T}{\tilde{u}} + \frac{L}{\tilde{u}}\right)G.
\end{aligned}$$

where the last inequality is satisfied by (16). Then, this completes the proof. □

The subsequent Lemma presented serves as a crucial role in the global convergence theorem of the proposed method.

Lemma 3. *Suppose that Assumption 1 and Assumption 2 are satisfied. Consider iterative method represented by equation 1, where d_k fulfils the sufficient descent condition and α_k is established using the standard Wolfe-Powell line search stated in (2) and (3). According to [9], if the aforementioned relationship holds:*

$$\sum_{k\geq 0} \frac{1}{||d_k||^2} = +\infty, \tag{17}$$

then, the method exhibits global convergence as such:

$$\lim_{k\to+\infty} \inf||g_k|| = 0. \tag{18}$$

A proof that Algorithm 2 is globally converge for uniformly convex objective functions will be presented in the next discussion.

Theorem 3.1. *Suppose that Assumption 1 and Assumption 2 are satisfied wherein α_k is established using the standard Wolfe-Powell line search stated in (2) and (3). If the objective function f exhibits uniform convexity on the set $\mathbb{N}$, it can be concluded that Algorithm 2 achieves global convergence in a way that:*

$$\lim_{k\to+\infty} ||g_k|| = 0.$$

Proof. According to Lemma 1, it follows that the direction d_k^{MTT} satisfies the sufficient descent property with a constant value of $c = 1$. From the inequality given in equation (2), it can be observed that the sequence $\{f(x_k)\}_{k\geq 0}$ is monotonically decreasing, and $\{x_k\}_{k\geq 0}$ belongs to the set of natural numbers, $\mathbb{N}$.

The validity of equation (17) can be established by applying the boundedness property of d_k^{MTT} as stated in Lemma 2. □

4 RESULTS AND DISCUSSIONS

Impulse noise arises from defects in sensors or communication paths, often characterized by the contamination of a certain number of pixels. Numerous image-related applications typically necessitate effective noise suppression techniques to restore the original image with stable outcomes. These problems are considered challenging in optimization due to their lack of smoothness. This section will discuss the capability of the MTT method in addressing image denoising challenges. Specifically, this section presents an evaluation of the MTT method's performance in solving image denoising problems, comparing it with two other methods, MTTHS and MZZ, based on CPU time, Peak Signal-to-Noise Ratio (PSNR), Relative Error (RE), and Structural Similarity Index (SSIM).

For the present study, we utilize the images Roro, Hill, and Lung, which are grayscale images of size 300×300. PSNR and SSIM are frequently used as objective measures of image quality to quantitatively evaluate the performance of denoising, assuming that the original image is completely available as a reference.

PSNR, which indicates the ratio of signal power to noise power, is defined as follows. A higher PSNR suggests better denoising quality as it indicates the denoised image is closer to the original:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \tag{19}$$

where the MSE (Mean Squared Error) is calculated by:

$$\text{MSE} = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{n=1}^{N} \left[I_1(m,n) - I_2(m,n) \right]^2, \tag{20}$$

in which M is the number of rows, N is the number of columns, $I_1(m,n)$ is the original image matrix data, and $I_2(m,n)$ is the denoised image matrix data. MSE represents the cumulative squared error between the original and the denoised image. A lower MSE value indicates improved denoising performance.

Relative Error (RE) measures the percentage error between the original and the denoised image. Lower RE values are desirable for better denoising performance. RE can be defined as:

$$\text{RE} = 100 \times \frac{\text{MSE}}{255^2/12}. \tag{21}$$

In the experiment, the "salt-and-pepper noise ratio" was considered 10%, 30%, and 50%. In this experiment, we consider Roro, Lung, and Hill as the test images with a resolution of 512×512 in gray level with various max window size such as 38, 41, and 43. The max window size in image denoising controls the balance between noise reduction and detail preservation. A larger window size provides better noise reduction by averaging over a broader area but can blur fine details, while a smaller window preserves details but may reduce noise less effectively. Optimal window size selection depends on noise type, image detail level, and computational limits.

The Structural Similarity Index (SSIM) measures similarity between two images in terms of luminance, contrast, and structural information. It refers to the ability of an image denoising

(a) Roro (b) Lung (c) Hill

Figure 1 : Original Images

algorithm to preserve the image's original features while reducing noise. Higher SSIM values indicate higher similarity between the original and denoised images, which is preferable.

Table 1 : Image denoising outputs based on tCPU, PSNR, RE, and SSIM.

Image	**Noise Ratio**	**Max Window Size**	**MTT**				**MTTHS**			
			tCPU	**PSNR**	**RE**	**SSIM**	**tCPU**	**PSNR**	**RE**	**SSIM**
Roro	10%	38	4.542	37.896	0.373	0.988	4.570	37.734	0.375	0.988
		41	4.627	37.767	0.377	0.988	4.629	37.683	0.378	0.987
		43	4.701	37.661	0.375	0.987	4.726	37.650	0.388	0.987
Roro	30%	38	18.270	31.595	1.049	0.949	19.150	31.554	1.049	0.949
		41	19.225	31.485	1.043	0.949	19.551	31.484	1.044	0.948
		43	19.901	31.368	1.052	0.948	19.987	31.331	1.054	0.948
Roro	50%	38	41.000	27.792	1.640	0.880	42.017	27.783	1.641	0.879
		41	42.494	27.854	1.628	0.880	42.883	27.724	1.628	0.878
		43	43.141	27.800	1.616	0.880	43.452	27.702	1.626	0.878
Lung	10%	38	4.365	38.707	0.461	0.990	6.394	38.164	0.464	0.983
		41	4.697	38.654	0.458	0.990	6.508	37.977	0.459	0.992
		43	4.891	38.587	0.461	0.989	6.953	37.028	0.464	0.983
Lung	30%	38	17.714	33.432	1.191	0.967	25.816	33.002	1.199	0.965
		41	17.721	33.442	1.190	0.967	24.061	33.552	1.195	0.965
		43	20.721	33.302	1.191	0.955	23.368	32.606	1.196	0.945
Lung	50%	38	42.160	30.802	1.613	0.886	55.441	30.363	1.752	0.853
		41	43.104	27.669	1.634	0.878	54.202	27.516	1.737	0.853
		43	44.490	26.198	1.624	0.878	52.814	26.142	1.731	0.853
Hill	10%	38	4.035	35.764	0.442	0.990	4.555	34.957	0.642	0.983
		41	4.991	36.108	0.602	0.991	5.174	35.359	0.642	0.983
		43	6.015	35.994	0.609	0.989	6.715	35.264	0.642	0.984
Hill	30%	38	18.307	29.720	1.558	0.953	20.190	29.777	1.561	0.944
		41	20.494	30.973	2.496	0.941	21.525	29.666	1.556	0.943

		43	24.812	26.720	1.558	0.943	27.416	25.644	1.670	0.942
Hill	50%	38	40.764	26.997	2.057	0.900	41.992	26.507	2.203	0.886
		41	41.173	27.052	2.097	0.890	42.008	27.877	2.010	0.900
		43	42.008	27.877	2.010	0.900	43.118	26.586	2.217	0.884

Table 1 and Table 2 show the image denoising outputs based on CPU Time, PSNR, RE, and SSIM for MTT along with MTTHS and MZZ. Various noisy rates have been used in this study, which are 10%, 30%, and 50%. In addition, different maximum window sizes, which are 38, 41, and 43, have been utilized in order to compare which will bring better results for denoising performance.

Table 2 : Image denoising outputs based on CPU Time, PSNR, RE, and SSIM.

Image	**Noise Ratio**	**Max Window Size**	**MTT**				**MZZ**			
			tCPU	**PSNR**	**RE**	**SSIM**	**tCPU**	**PSNR**	**RE**	**SSIM**
Roro	10%	38	4.542	37.896	0.373	0.988	4.689	37.840	0.379	0.986
		41	4.627	37.767	0.377	0.988	4.715	37.685	0.380	0.986
		43	4.701	37.661	0.375	0.987	4.794	37.636	0.375	0.987
Roro	30%	38	18.270	31.595	1.049	0.949	18.827	31.568	1.053	0.946
		41	19.225	31.485	1.043	0.949	19.274	31.479	1.043	0.946
		43	19.901	31.368	1.052	0.948	19.968	31.360	1.060	0.947
Roro	50%	38	41.000	27.792	1.640	0.880	47.748	27.777	1.622	0.875
		41	42.494	27.854	1.628	0.880	47.534	27.690	1.628	0.874
		43	43.141	27.800	1.616	0.880	43.842	27.755	1.632	0.876
Lung	10%	38	4.365	38.707	0.461	0.990	4.800	38.506	0.460	0.979
		41	4.697	38.654	0.458	0.990	4.479	38.466	0.458	0.979
		43	4.891	38.587	0.461	0.989	4.498	38.530	0.461	0.979
Lung	30%	38	17.714	33.432	1.191	0.967	18.120	33.153	1.194	0.945
		41	17.721	33.442	1.190	0.967	18.607	32.127	1.198	0.945
		43	20.721	33.302	1.191	0.955	21.331	32.636	1.200	0.944
Lung	50%	38	42.160	30.802	1.613	0.886	42.830	30.766	1.739	0.875
		41	43.104	27.669	1.634	0.878	45.236	27.566	1.754	0.814
		43	44.490	26.198	1.624	0.878	46.003	28.977	1.740	0.855
Hill	10%	38	4.035	35.764	0.442	0.990	4.739	35.242	0.644	0.978
		41	4.991	36.108	0.602	0.991	5.623	35.278	0.649	0.979
		43	6.015	35.994	0.609	0.989	6.832	35.533	0.644	0.979
Hill	30%	38	18.307	29.720	1.558	0.953	21.879	29.608	1.580	0.930
		41	20.494	30.973	2.496	0.941	21.324	29.783	1.573	0.932
		43	24.812	26.720	1.558	0.943	25.650	25.665	1.563	0.932
Hill	50%	38	40.764	26.997	2.057	0.900	45.892	26.547	2.212	0.872
		41	41.173	27.052	2.097	0.890	42.402	26.530	2.210	0.871
		43	42.008	27.877	2.010	0.900	41.701	26.647	2.204	0.873

From both tables, it is noticeable that MTT has lower values when comparing CPU time, relative error, and mean square error while it has higher values when comparing metrics that measure the quality of denoised images: PSNR and SSIM in most cases. Moreover, it can be seen in the tables that MTT performs very well in solving the image denoising problem for a maximum window size of 38. Thus, it can be concluded that MTT is superior to the other two algorithms.

A problem has arisen while trying to solve the image denoising problem. This problem occurred as the algorithm was unable to denoise the image. Fortunately, the problem has been solved by setting the window size equal to 500 and step size equal to 800.

Figures 2 through 10 show the denoised images for MTT, MTTHS, and MZZ after being corrupted by various noisy rates (10%, 30%, and 50%) and using different maximum window sizes (38, 41, and 43).

Figure 11a illustrates the graph of PSNR values for MTT, MTTHS, and MZZ in image denoising problems. The higher the PSNR values, the better the quality of the denoised image. From Figure 13, it can be seen that MTT performed very well at every stage compared to MTTHS and MZZ. At a noisy rate of 10%, there is fierce competition among all the maximum window sizes. However, a maximum window size of 38 is slightly better compared to the others. Meanwhile, at a noisy rate of 30%, a maximum window size of 41 is more preferable. For a noisy rate of 50%, a maximum window size of 38 is preferable. Thus, it can be concluded that MTT is superior to the other two methods, as it performed well and showed promising results.

Figure 11b displays the results of MTT, MTTHS, and MZZ for SSIM values in solving image denoising problems. Higher SSIM values indicate a higher similarity between the two images, which is desirable. From Figure 14, it is observable that MTT showed good performance during every stage. At every noisy rate, it can be seen that a maximum window size of 38 is more preferable.

Figure 12a shows the CPU time resulting from solving image denoising problems using MTT, MTTHS, and MZZ. CPU time should be lower to indicate that the method required less time to solve the image denoising problems. From Figure 15, it is evident that the MTT method took the shortest CPU time among all methods across all noisy rates and maximum window sizes. For every noisy rate, the maximum window size of 38 is suitable, as a smaller window size requires fewer calculations compared to larger ones. This translates to faster processing times, especially for larger images or real-time applications. Thus, MTT took the shortest CPU time in solving the image denoising problems and showed promising results for a maximum window size of 38 across all noisy rates.

The relative error results for MTT, MTTHS, and MZZ are displayed in Figure 12b. Lower relative error indicates better denoising performance. From Figure 16, MTT performs better at most stages but shows a higher spike at a 30% noisy rate for a maximum window size of 41. This is due to some outliers in the data that disproportionately affect the relative error. These outliers could result from regions of the image where the denoising algorithm performs poorly. At 10% and 30% noisy rates, a maximum window size of 38 is more suitable. However, a maximum window size of 41 is better suited for a 50% noisy rate. Despite this, MTT still performs better compared to the other methods.

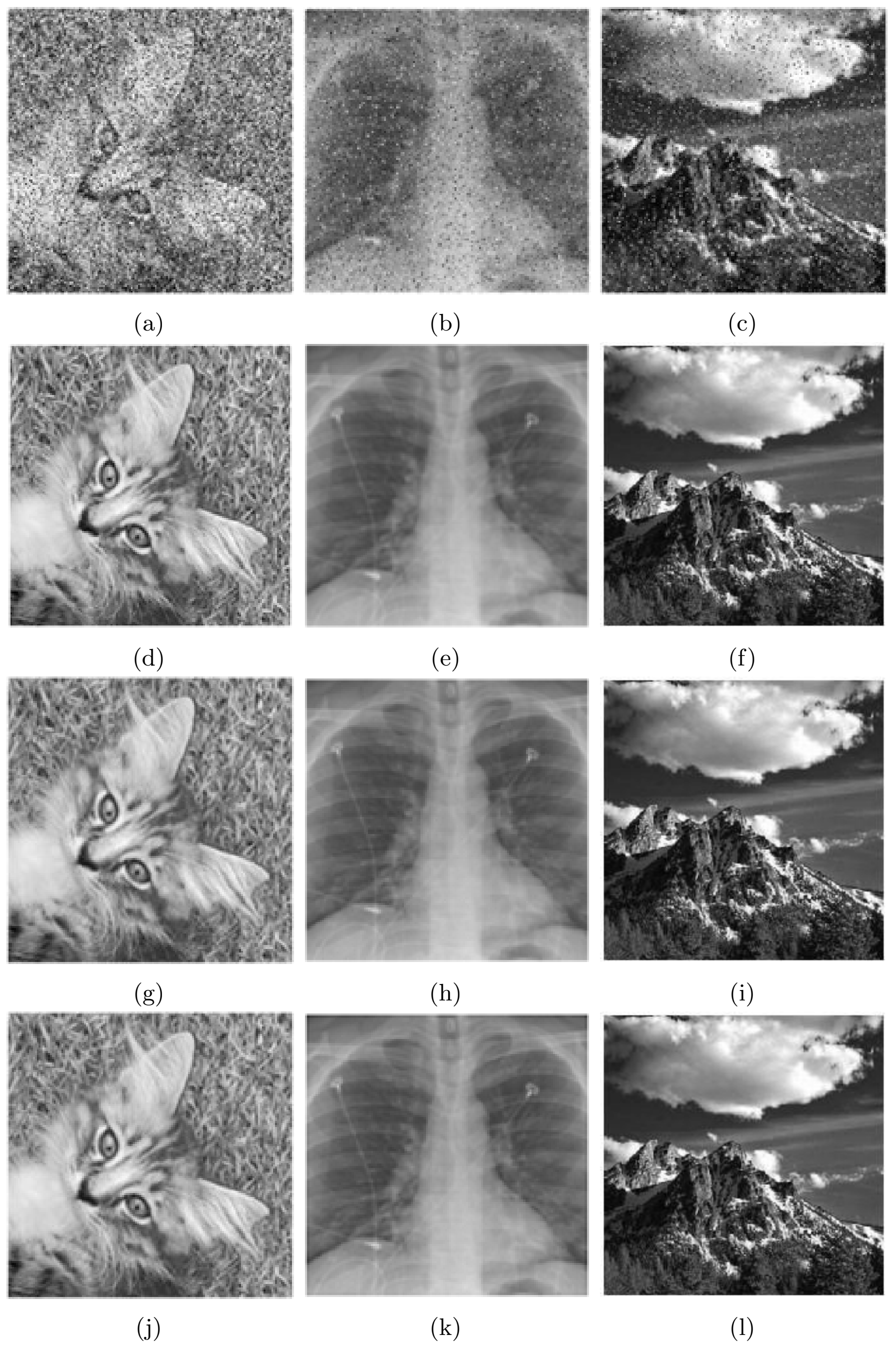
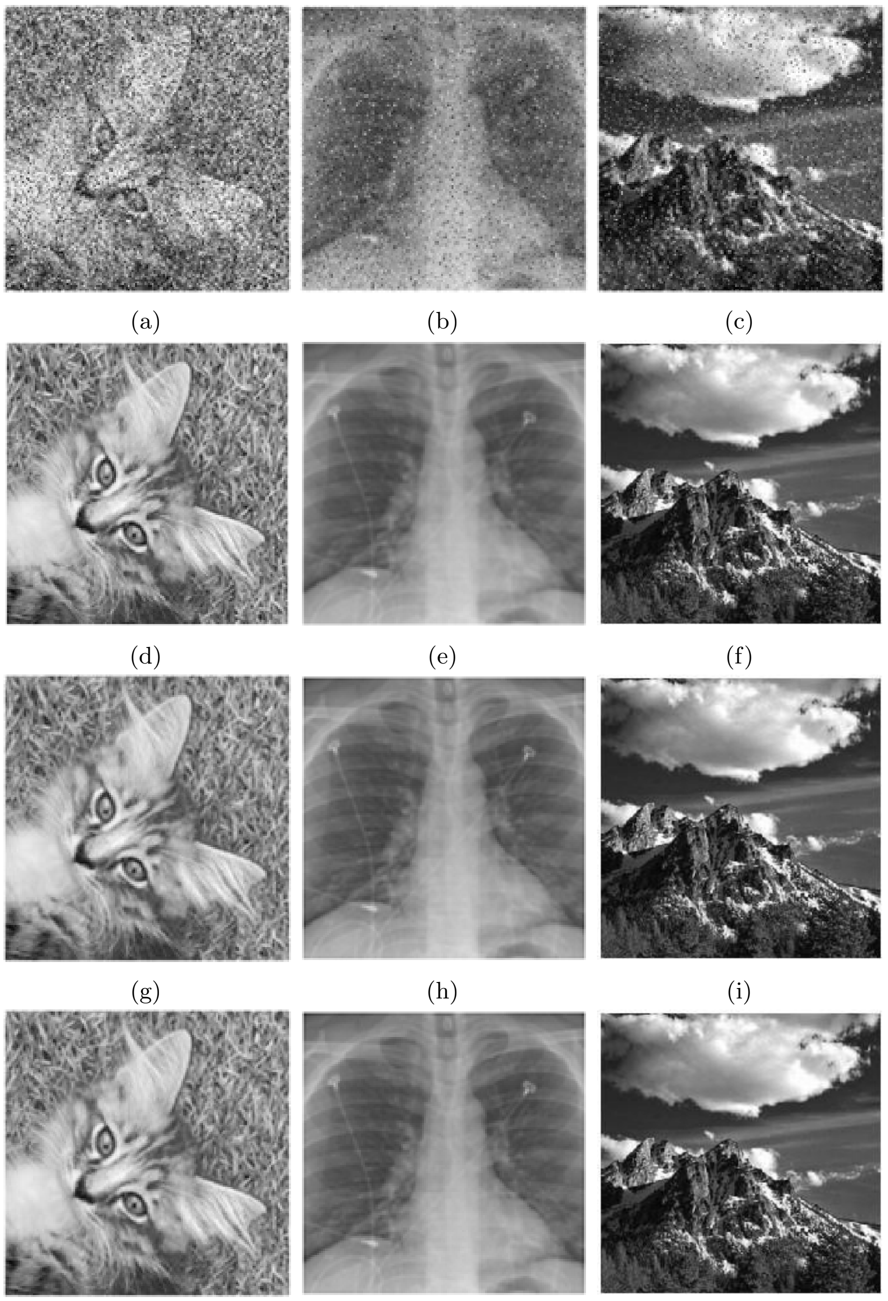
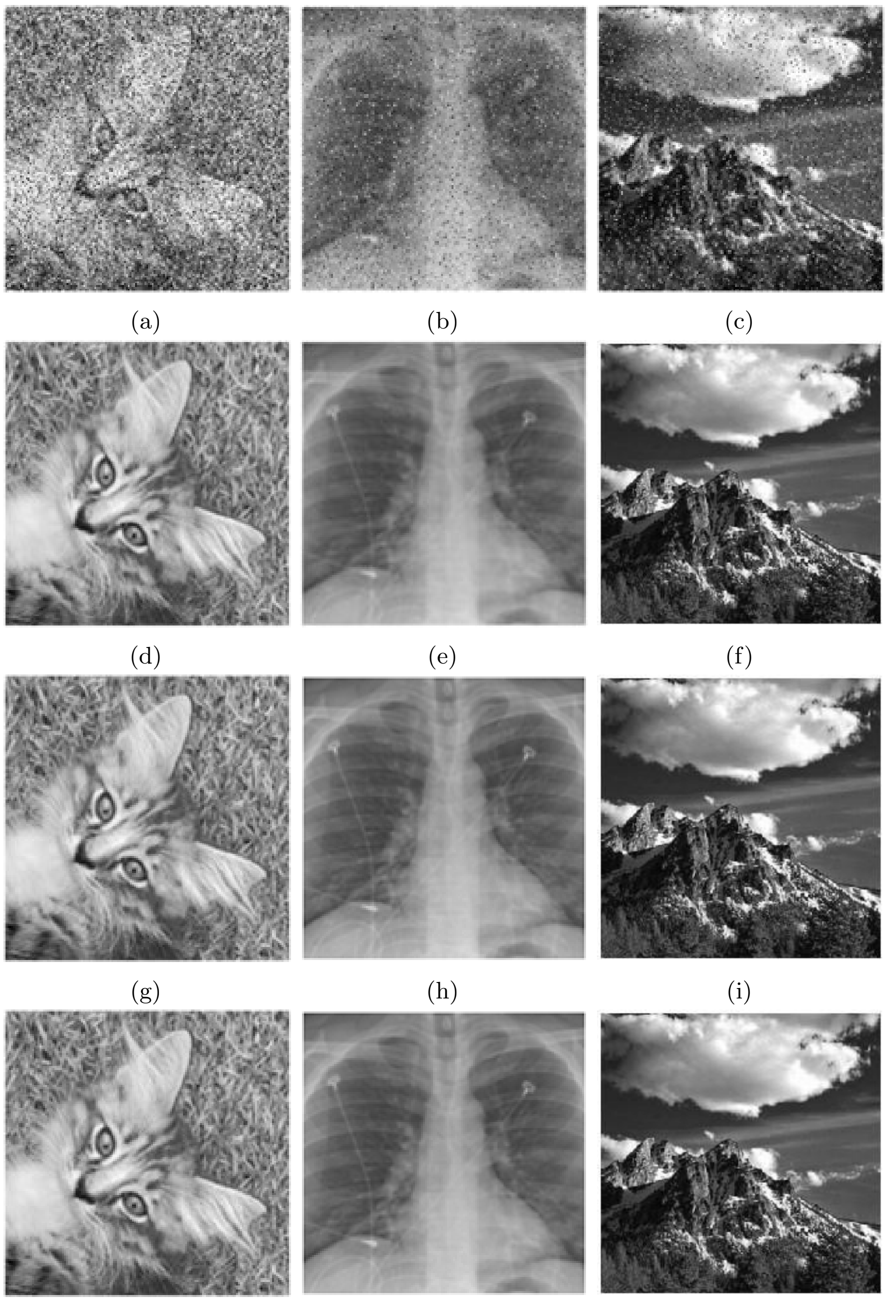


Figure 2 : Noisy Image with 10% Noisy Rate (a-c), Denoised Image with Maximum Window Size 38 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 3 : Noisy Image with 10% Noisy Rate (a-c), Denoised Image with Maximum Window Size 41 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)



(a) (b) (c)
(d) (e) (f)
(g) (h) (i)
(j) (k) (l)

Figure 4 : Noisy Image with 10% Noisy Rate (a-c), Denoised Image with Maximum Window Size 43 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

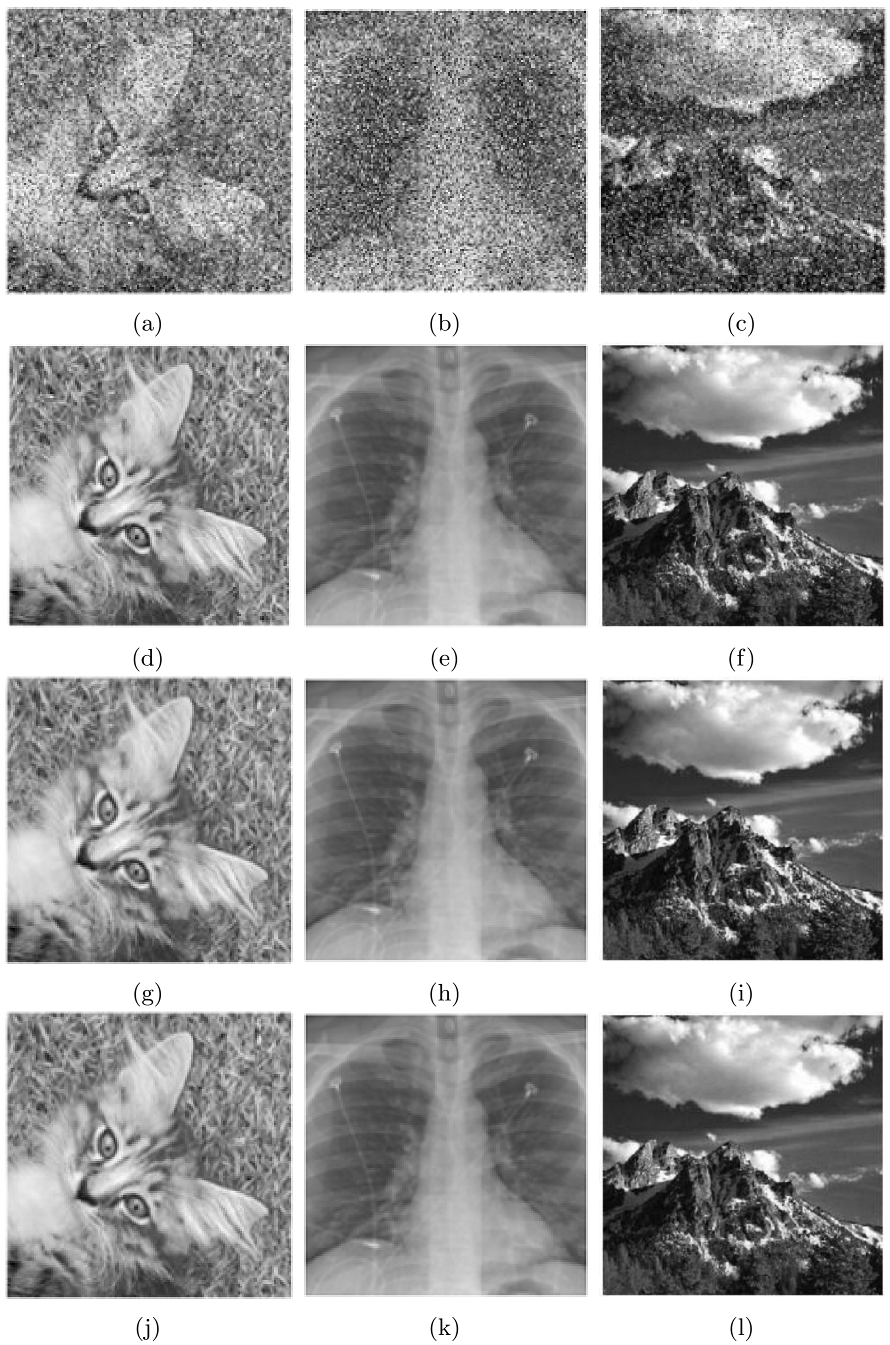


Figure 5 : Noisy Image with 30% Noisy Rate (a-c), Denoised Image with Maximum Window Size 38 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

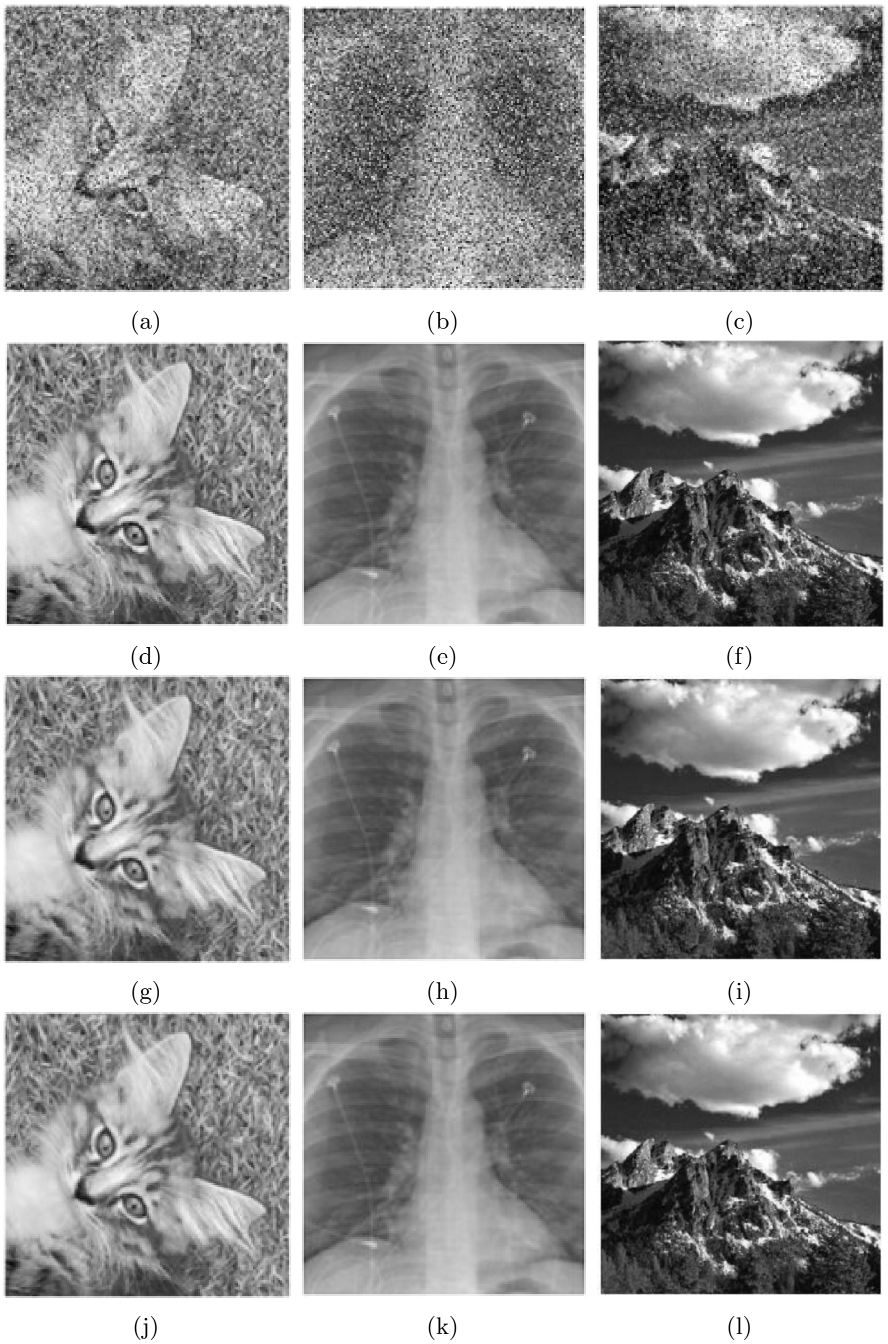


Figure 6 : Noisy Image with 30% Noisy Rate (a-c), Denoised Image with Maximum Window Size 41 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

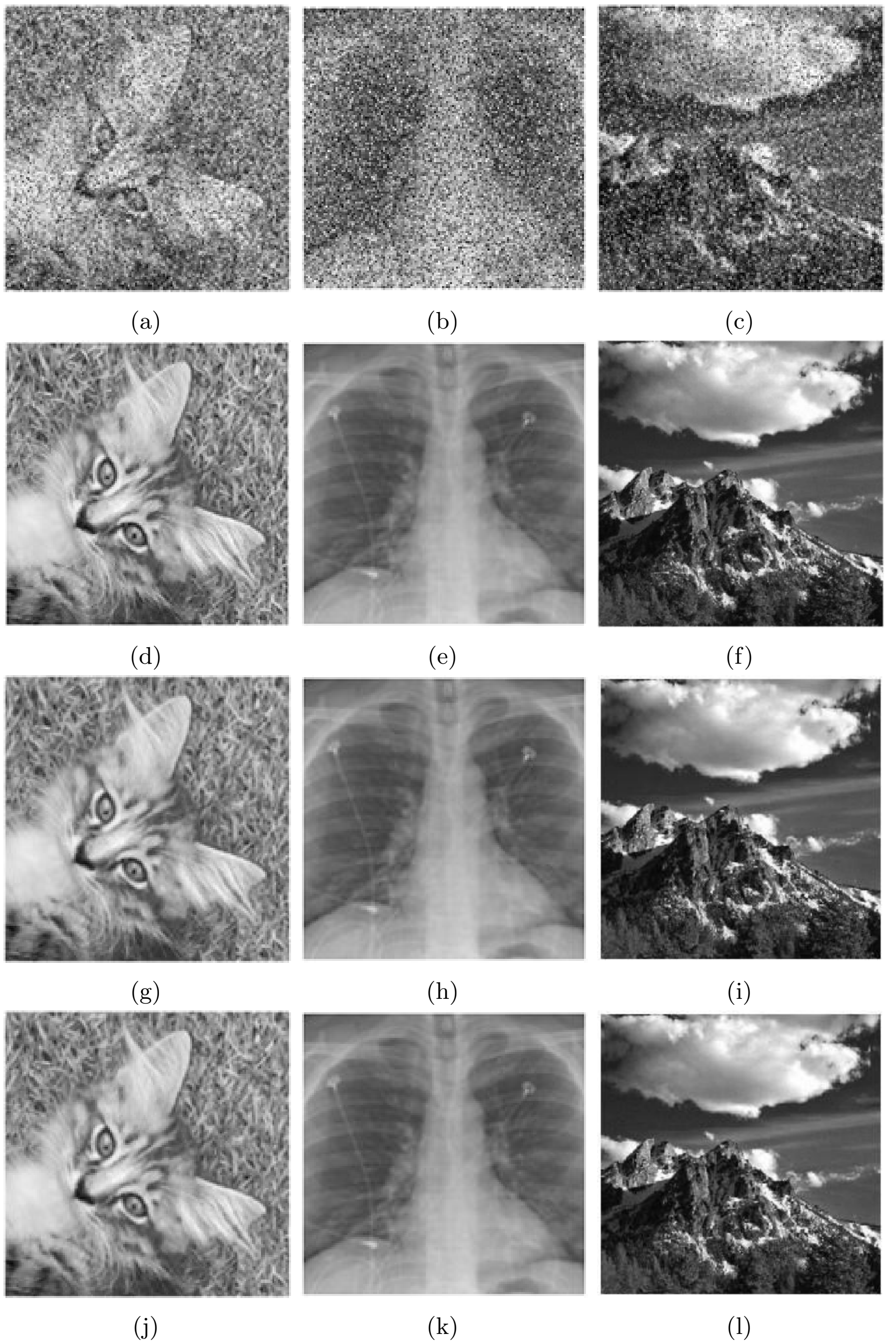


Figure 7 : Noisy Image with 30% Noisy Rate (a-c), Denoised Image with Maximum Window Size 43 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

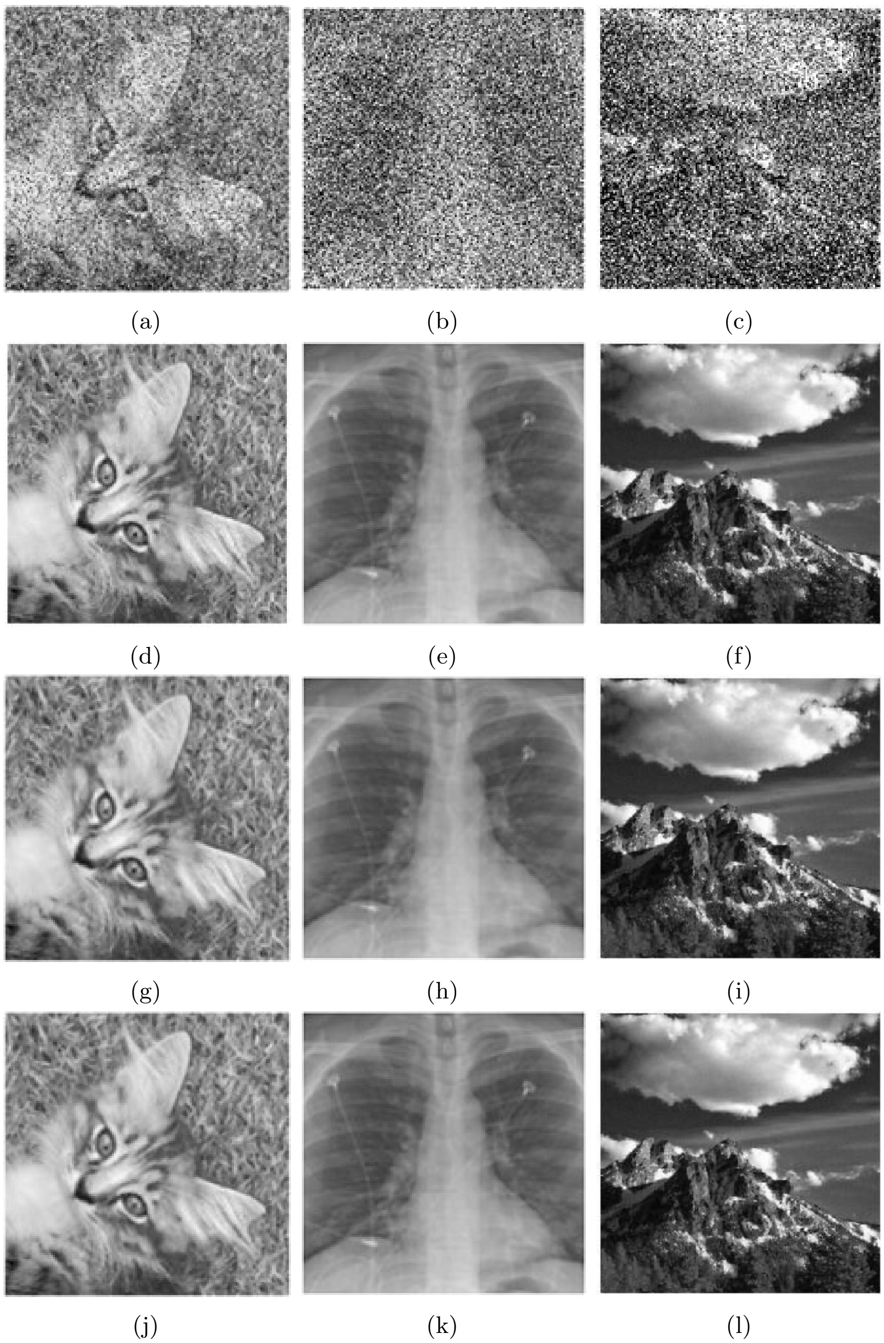


Figure 8 : Noisy Image with 50% Noisy Rate (a-c), Denoised Image with Maximum Window Size 38 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

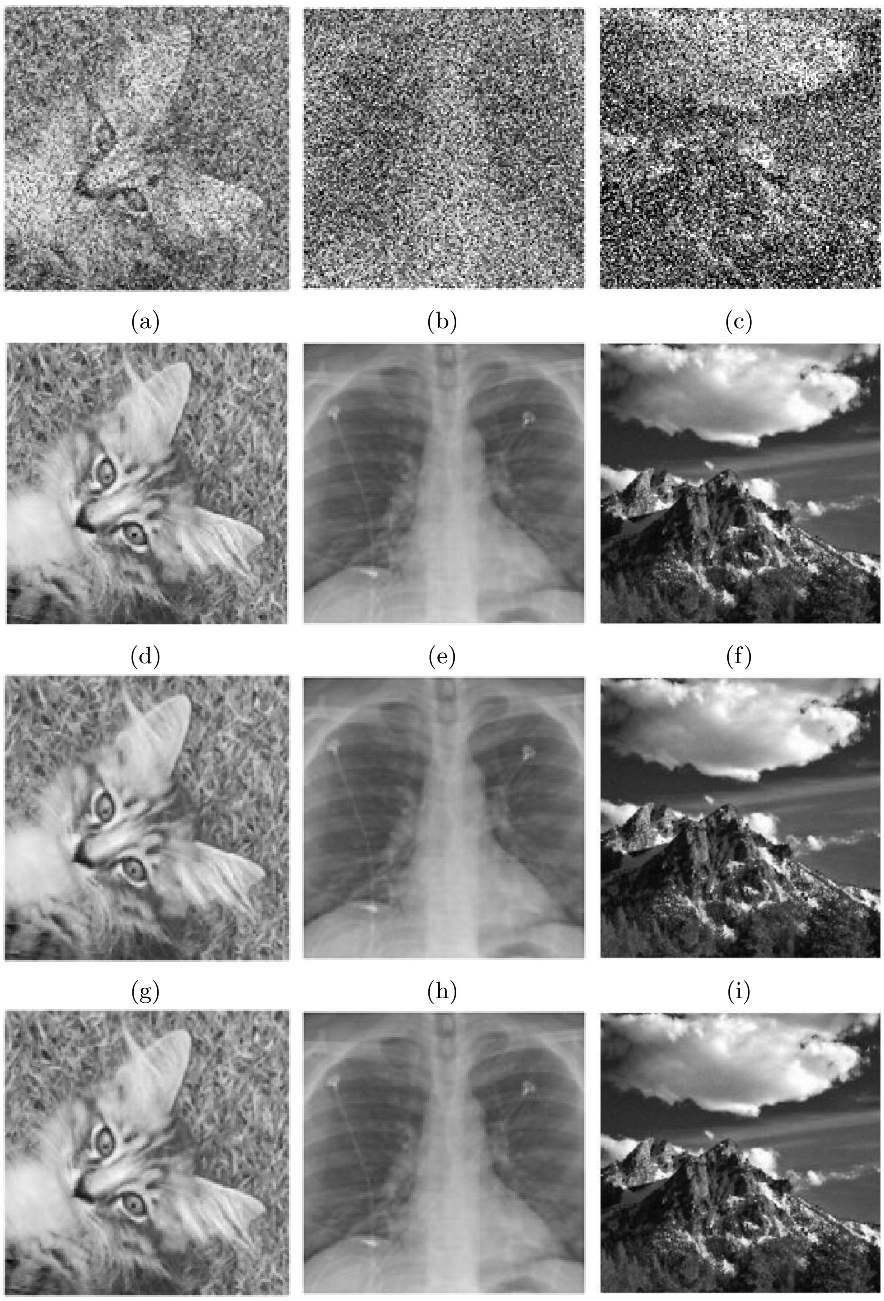


Figure 9 : Noisy Image with 50% Noisy Rate (a-c), Denoised Image with Maximum Window Size 41 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)

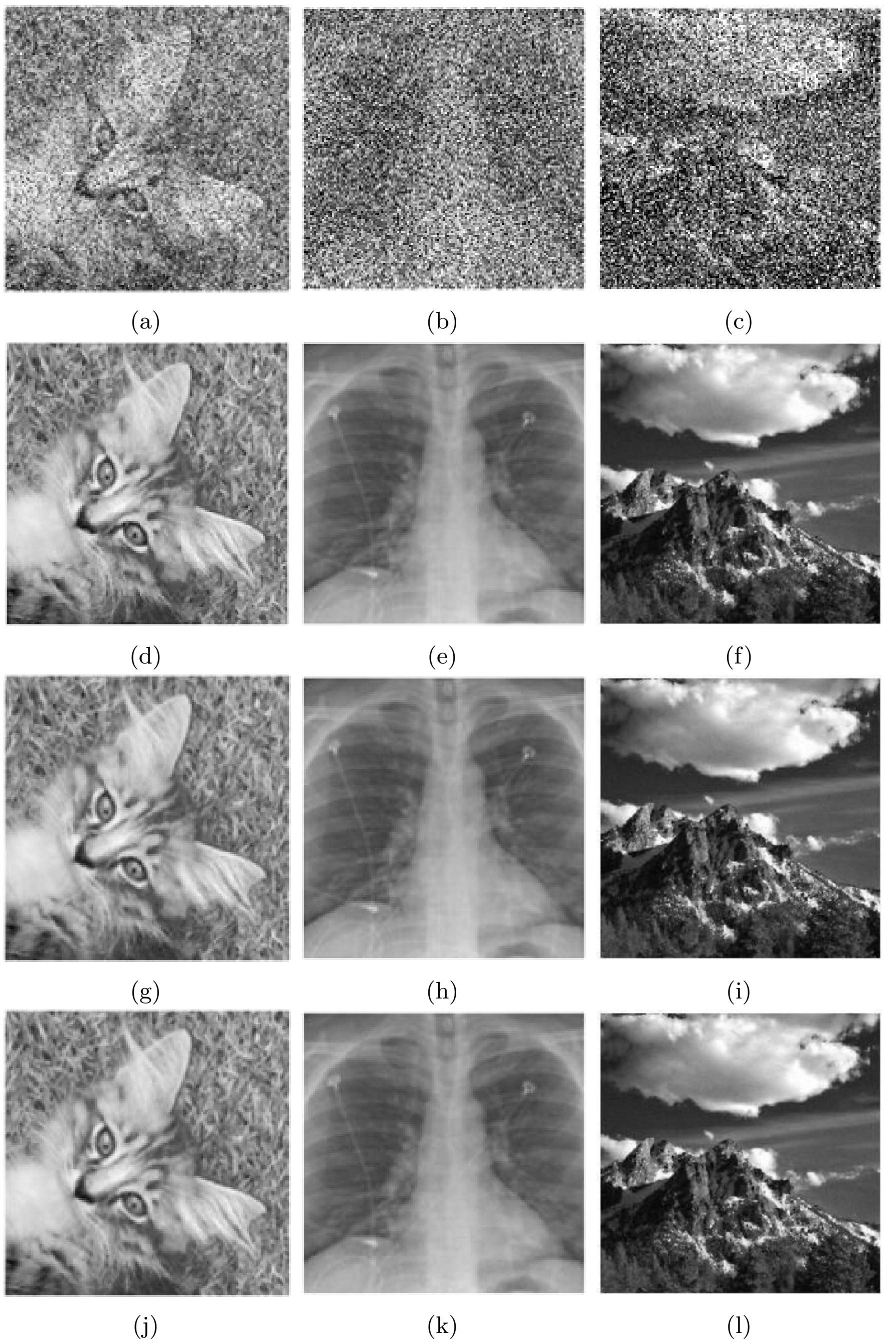
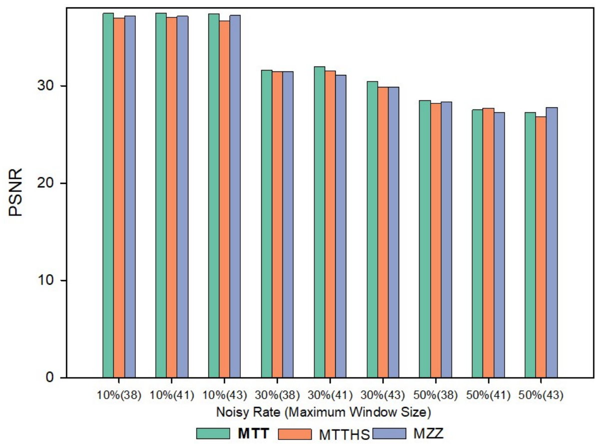
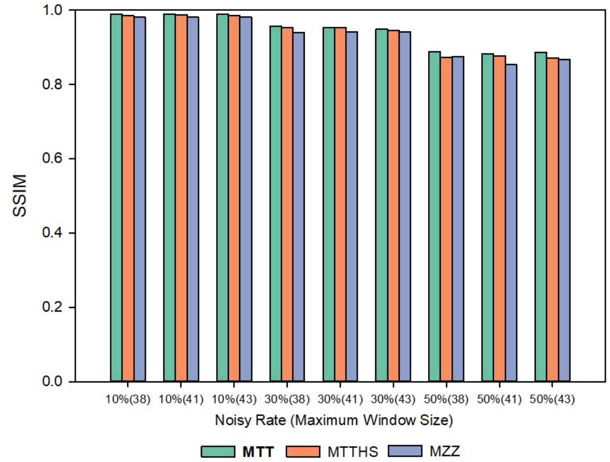


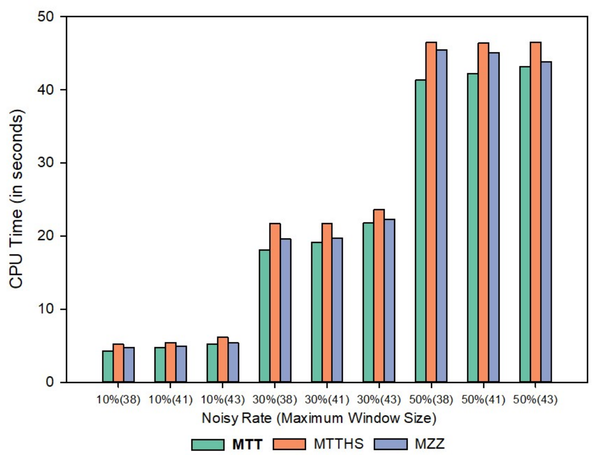
Figure 10 : Noisy Image with 50% Noisy Rate (a-c), Denoised Image with Maximum Window Size 43 by MTT (d-f), MTTHS (g-i), and MZZ (j-l)



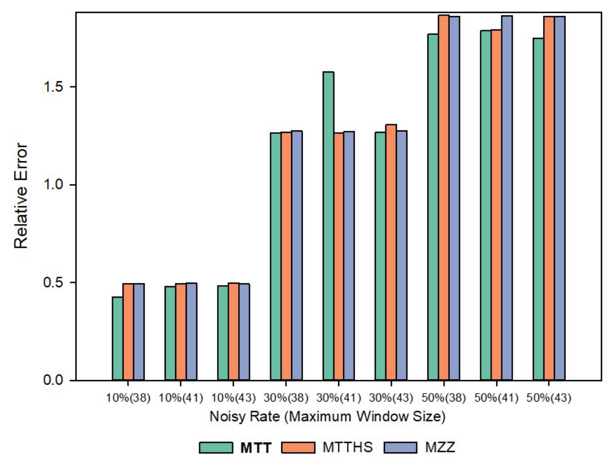
(a) PSNR Values for MTT, MTTHS and MZZ



(b) SSIM Values for MTT, MTTHS and MZZ



(a) CPU Time for MTT, MTTHS and MZZ



(b) Relative Error for MTT, MTTHS and MZZ

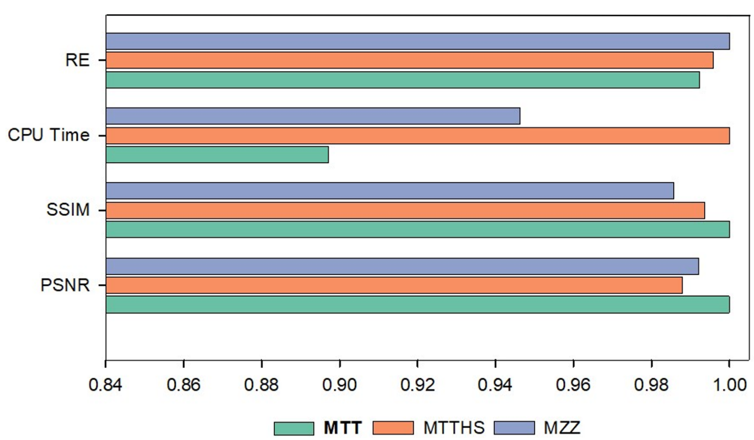


Figure 13 : Overall Results for MTT, MTTHS and MZZ

Figure 13 represents the overall results for MTT, MTTHS, and MZZ in solving image denoising problems with respect to PSNR, SSIM, CPU Time, Relative Error, and Mean Square Error. From Figure 13, it is apparent that, in terms of metrics that measure and evaluate how well an image preserves the information and appearance of the original image (PSNR and SSIM), the MTT method outmatches both the MTTHS and MZZ methods. The MTT method again outperforms the other two methods by having the lowest values of CPU time and relative error. Thus, from the overall results, it can be concluded that the MTT method provides the best results among all three methods and is effective in solving image denoising problems across various noisy rates and maximum window sizes.

5 CONCLUSION

In this study, a new modification of the hybrid three-term PRP-DL CG method, MTT, was developed and applied to large-scale unconstrained optimization problems, specifically in the context of image denoising. The MTT method, equipped with a modified search direction and enhanced line search techniques, demonstrated significant improvements over existing methods like MTTHS and MZZ. The experimental results showed that MTT consistently provided superior performance, as indicated by higher PSNR and SSIM values, and reduced CPU time and relative error, across varying noise levels and window sizes. Notably, MTT performed best with a window size of 38, which balanced denoising effectiveness and computational efficiency. Overall, this study underscores the robustness and adaptability of the MTT method in handling complex image denoising tasks. The proposed algorithm shows promise for further applications in real-time and large-scale optimization challenges, reinforcing its value as a reliable tool in the field of image processing.

ACKNOWLEDGMENTS

The authors are grateful to the editor and the anonymous reviewers for their comments and suggestions which improved this paper substantially. This paper was also presented at the 2^{nd} Mathematical Sciences and Statistics Seminar ($2^{\mathrm{nd}}\mathrm{MS}^3$). Special thanks to the Department of Mathematics and Statistics, Faculty of Science, UPM.

REFERENCES

[1] M. R. Hestenes, "The conjugate-gradient method for solving," *Numerical analysis*, no. 6, p. 83, 1956.

[2] J. Wang, W. Wu, and J. M. Zurada, "Deterministic convergence of conjugate gradient method for feedforward neural networks," *Neurocomputing*, vol. 74, no. 14-15, pp. 2368–2376, 2011.

[3] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.

[4] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[5] M. R. Hestenes, E. Stiefel *et al.*, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[6] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of optimization theory and applications*, vol. 69, pp. 129–137, 1991.

[7] Y.-H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[8] R. Fletcher, *Practical methods of optimization.* John Wiley & Sons, 2000.

[9] Q. Tian, X. Wang, L. Pang, M. Zhang, and F. Meng, "A new hybrid three-term conjugate gradient algorithm for large-scale unconstrained problems," *Mathematics*, vol. 9, no. 12, p. 1353, 2021.

[10] L. Zhang, W. Zhou, and D. Li, "Some descent three-term conjugate gradient methods and their global convergence," *Optimisation Methods and Software*, vol. 22, no. 4, pp. 697–711, 2007.

[11] H. Wasi and M. A. Shiker, "A new hybrid cgm for unconstrained optimization problems," in *Journal of Physics: Conference Series*, vol. 1664, no. 1. IOP Publishing, 2020, p. 012077.

[12] S. Babaie-Kafaki, "On optimality of the parameters of self-scaling memoryless quasi-newton updating formulae," *Journal of Optimization Theory and Applications*, vol. 167, pp. 91–101, 2015.

[13] R. H. Chan, C. W. Ho, and M. Nikolova, "Salt and pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.

[14] P. Charbonnier, L. Blanc-Féraud, and G. Aubert, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.

[15] J. F. Cai, R. Chan, and B. Morini, "Minimization of an edge-preserving regularization functional by conjugate gradient type methods," in *Image Processing Based on Partial Differential Equations.* Berlin, Heidelberg: Springer, 2007, pp. 109–122.

[16] Y. H. Dai and L. Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Applied Mathematics and optimization*, vol. 43, pp. 87–101, 2001.