# A Novel Sparse Image Reconstruction Based on Iteratively Reweighted Least Squares Using Diagonal Regularization

Bamrung Tausiesakul [1] and Krissada Asavaskulkiet [2,*]

[1] Department of Electrical Engineering, Faculty of Engineering, Srinakharinwirot University, Nakhonnayok, Thailand; Email: bamrungt@g.swu.ac.th (B.T.)
[2] Department of Electrical Engineering, Faculty of Engineering, Mahidol University, Nakhon Pathom, Thailand
*Correspondence: krissada.asa@mahidol.edu (K.A.)

***Abstract*—In the information age, numerous data needs to be transferred from one point to another. The bigger the amount of the data, the more the consumption in computation and memory. Due to a limitation of the existing resource, the compression of the data and the reconstruction of the compressed data receive much attention in several research areas. A sparse signal reconstruction problem is considered in this work. The signal can be captured into a vector whose elements can be zeros. Iteratively Reweighted Least Squares (IRLS) is a technique that is designed for extracting the signal vector from the available observation data. In this paper, a new algorithm based on the iteratively reweighted least squares using diagonal regularization method are proposed for sparse image reconstruction. The explicit solution of the IRLS optimization problem is derived and then an alternative IRLS algorithm based on the available solution is proposed. Since the matrix inverse in the iterative computation can be subject to ill condition, a diagonal regularization is proposed to overcome such a problem. Numerical simulation is conducted to illustrate the performance of the new IRLS with the comparison to the former IRLS algorithm. Numerical results indicate that the new IRLS method provides lower signal recovery error than the conventional IRLS approach at the expense of more complexity in terms of more computational time.**

***Keywords*—compressed sensing, image reconstruction, iterative reweighted least squares**

## I. Introduction

There are many works that discussed a problem known as compressive sensing [1, 2]. The basic concept is that sparsity allows the signals that are sparse or compressible for being reconstructed from a small number of samples. This also allows for exploiting the additional information about the solutions, such as model-based compressive sensing. The methods in [3, 4] can enhance the recoverability and introduce a new class of the structured compressible signals along with a new sufficient condition of robust structured compressible signal recovery. In addition, based on structured sparsity theory [4–6], more benefits can be attained if more *a priori* information could be applied to the sparsity patterns, such as the components of the data may be clustered in groups. This technique is called group-sparse data in which the components within the same group tend to be either zeros or non-zeros.

Consider the problem of recovering a desired signal $\boldsymbol{x} \in \mathbb{R}^{N\times 1}$ from a set of observation data $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$ based on a modeling matrix or measurement matrix $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ which either depends on the model or can be chosen beforehand, where $M \in \mathbb{N}^{1\times 1}$ and $N \in \mathbb{N}^{1\times 1}$ are the lengths of real-valued input data and real-valued output data, respectively.

This problem has many applications in science and engineering, e.g., imaging and vision [7], photonic mixer device [8], electronic defense [9], security and cryptosystem [10], radar [11, 12], earth observation [13], wireless networks [14, 15], biometric watermarking [16], healthcare [17], etc.

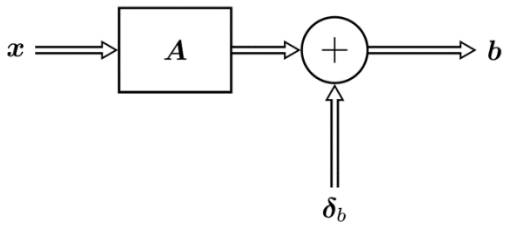


Figure 1. A noisy linear input-output system.

In Fig. 1, if the perturbation $\boldsymbol{\delta}_b$ is negligible, the linear complex system can approximately be written as

$$\boldsymbol{Ax} \approx \boldsymbol{b} \tag{1}$$

One of the possible ways to retrieve the unknown signal $\boldsymbol{x}$ such that Eq. (1) holds true.

Iteratively Reweighted Least Squares (IRLS) is a considerable criterion for extracting the signal vector $\boldsymbol{x}$ from the available observation data $\boldsymbol{b}$. This idea was proposed in [18, 19]. Other variations of the IRLS were examined in [20–24]. In this work, a closed-form solution of the IRLS criterion is derived. An alternative IRLS is

Manuscript received March 13, 2023; revised April 5, 2023; accepted May 8, 2023; published December 7, 2023.

later proposed based on the straightforward optimization. Due to a matrix inverse issue during the iterative computation, a regularization technique is adopted to overcome the ill-conditioned matrix. Numerical simulation is conducted to illustrate the performance of the new IRLS with the comparison to the former IRLS in [18, 19]. Numerical results indicate the lower signal recovery error at the expense of more complexity in terms of more computational time.

## II. Weighted Least Squares

Let $\boldsymbol{w} \in \mathbb{R}^{N\times 1}$ be a weighting vector, which can be element-wise expressed by

$$\boldsymbol{w} = [w_1, \quad w_2, \quad \ldots, \quad w_N]^{\mathrm{T}}, \tag{2}$$

where $(\cdot)^{\mathrm{T}}$is the transpose operator, which can be applied to a vector or a matrix. Since each element of $\boldsymbol{x}$ may be subject to outlier or meaningless information, e.g., $x_n = 0$, in some scenarios. A Weighted Least Squares (WLS) optimization problem can be formulated as

$$\hat{\boldsymbol{x}}_{\mathrm{WLS}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \textstyle\sum_{n=1}^{N} w_n\, x_n^2, \text{ s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{3}$$

To solve the constraint optimization in Eq. (3), one has to consult the Lagrange multiplier technique [25]. The Lagrange function of Eq. (3) can be written as

$$f(\boldsymbol{x}, \boldsymbol{\lambda}) = \textstyle\sum_{n=1}^{N} w_n\, x_n^2 + \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}), \tag{4}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{N\times 1}$ is the Lagrange multiplier vector, which is also unknown. Based on few multivariate derivatives (see, e.g., [26, 27]), it can be shown that

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\lambda}} f(\boldsymbol{x}, \boldsymbol{\lambda}) &= \textstyle\sum_{n=1}^{N} \underbrace{\frac{\partial}{\partial \boldsymbol{\lambda}} w_n x_n^2}_{=\boldsymbol{0}} + \underbrace{\frac{\partial}{\partial \boldsymbol{\lambda}} \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})}_{=\boldsymbol{I}} \\ &= \boldsymbol{0} + \boldsymbol{I}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b},\end{aligned} \tag{5}$$

and

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\lambda}) &= \textstyle\sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{x}} w_n x_n^2 + \frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \textstyle\sum_{n=1}^{N} \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_N} \end{bmatrix} w_n x_n^2 + \underbrace{\frac{\partial}{\partial \boldsymbol{\lambda}} \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{A}\boldsymbol{x}}_{=\boldsymbol{I}} - \underbrace{\frac{\partial}{\partial \boldsymbol{\lambda}} \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{b}}_{=\boldsymbol{0}} \\ &= \begin{bmatrix} 2w_1 x_1 \\ 2w_2 x_2 \\ \vdots \\ 2w_N x_N \end{bmatrix} + \boldsymbol{I}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda} + \boldsymbol{0} \\ &= 2\boldsymbol{D}(\boldsymbol{w})\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda},\end{aligned} \tag{6}$$

where $\boldsymbol{D}(\cdot)$ is the diagonal matrix, whose diagonal is taken from the vector $\cdot$. Solving the critical points $\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{0}$and $\frac{\partial}{\partial \boldsymbol{\lambda}} f(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{0}$ for $\boldsymbol{x}$ and $\boldsymbol{\lambda}$, respectively, bring about

$$\begin{aligned} &2\boldsymbol{D}(\boldsymbol{w})\boldsymbol{x} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda} = \boldsymbol{0} \\ &\quad \Rightarrow 2\boldsymbol{D}(\boldsymbol{w})\boldsymbol{x} = -\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda} \\ &\quad \Rightarrow \hat{\boldsymbol{x}}_{\mathrm{WLS}} = -\tfrac{1}{2}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda}, \\ &\quad \underbrace{\boldsymbol{A}\hat{\boldsymbol{x}}}_{(7)} - \boldsymbol{b} = \boldsymbol{0} \end{aligned} \tag{7}$$

$$\Rightarrow \boldsymbol{A}\left(-\frac{1}{2}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda}\right) = \boldsymbol{b}$$

$$\Rightarrow \hat{\boldsymbol{\lambda}} = -2(\boldsymbol{A}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b}, \tag{8}$$

and

$$\begin{aligned}\hat{\boldsymbol{w}}_{\mathrm{WLS}} &= -\tfrac{1}{2}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\underbrace{\hat{\boldsymbol{\lambda}}}_{(8)} \\ &= -\tfrac{1}{2}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}(-2(\boldsymbol{A}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b}) \\ &= \boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{D}^{-1}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b},\end{aligned} \tag{9}$$

where $(\cdot)^{-1}$ is the inverse of a square matrix $\cdot$. The WLS estimate in Eq. (9) works perfectly if any element of $\boldsymbol{w}$ is non-zero, i.e., $w_n \neq 0$ for $\forall n \in \{1, 2, \ldots, N\}$. However, when some elements of $\boldsymbol{w}$ become zeros, i.e., $w_n = 0$ for $\exists n \in \{1, 2, \ldots, N\}$, the solution in Eq. (9) will diverge. Let us introduce the definite reciprocal operator $(\cdot)^{\mp}$as

$$z^{\mp} = \begin{cases} \frac{1}{z}, & z \neq 0, \\ 0, & z = 0, \end{cases} \tag{10}$$

for $z \in \mathbb{C}^{1\times 1}$. In compressed sensing, few or some elements of the unknown $\boldsymbol{x}$ possess zeros. To deal with the trivial case $w_n = 0$ for $\exists n \in \{1, 2, \ldots, N\}$, the WLS estimate in Eq. (9) has to be modified as

$$\hat{\boldsymbol{w}}_{\mathrm{WLS}} = \boldsymbol{D}^{\mp}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{D}^{\mp}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{b}, \tag{11}$$

Basic algebraic manipulation reveals that

$$\begin{aligned}\boldsymbol{A}\boldsymbol{D}^{\mp}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}} &= [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots\ \boldsymbol{a}_N] \begin{bmatrix} w_1^{\mp} & 0 & \cdots & 0 \\ 0 & w_2^{\mp} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_N^{\mp} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{a}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_N^{\mathrm{T}} \end{bmatrix} \\ &= \sum_{n=1}^{N} w_N^{\mp} \boldsymbol{a}_n\, \boldsymbol{a}_n^{\mathrm{T}} \\ &= \textstyle\sum_{n\in\mathcal{N}_{w\neq 0}} \frac{1}{w_n} \boldsymbol{a}_n \boldsymbol{a}_n^{\mathrm{T}},\end{aligned} \tag{12}$$

where $\mathcal{N}_{w\neq 0}$ is the set that consists of all indices of $n$ for which $w_n \neq 0$, i.e.,

$$\mathcal{N}_{w\neq 0} = \{n | w_n \neq 0; n = 1, 2, \ldots, N\}. \tag{13}$$

If the number of nonzero weights is less than $M$, the matrix $\boldsymbol{AD}^{\mp}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}$will be singular, i.e.,

$$|\mathcal{N}_{w\neq 0}| < M \Rightarrow \left|\boldsymbol{AD}^{\mp}(\boldsymbol{w})\boldsymbol{A}^{\mathrm{T}}\right| = 0, \quad (14)$$

where $|\cdot|$ is either the cardinality of a set or the determinant of a matrix. To alleviate the inverse issue, one might need to resort to a diagonal regularization or a truncated singular value decomposition.

## III. Iteratively Reweighted Least Squares

Let $\ell(\boldsymbol{x})$ be a non-increasing set of all elements in $\boldsymbol{x}$, which can be represented by

$$\ell(\boldsymbol{x}) = \{l_{n_1} | l_{n_1} = x_{n_2}; n_1, n_2 \in \{1, 2, \ldots, N\} \wedge |l_1| \geq |l_2| \geq \ldots \geq |l_N|\}. \quad (15)$$

Let $L_k(\boldsymbol{x})$ be the $k$-th element of the non-increasing set $\ell(\boldsymbol{x})$, i.e., $L_k(\boldsymbol{x}) = l_k$. An iterative computation was introduced earlier as follows.

**Algorithm 1.** Iteratively Reweighted Least Squares (IRLS) [18, 19]

**Input:** $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, $\boldsymbol{b} \in \mathbb{R}^{M\times 1}$, $p \in (0,1]$, $k \in \mathbb{Z}^{1\times 1}$
**Output:** $\hat{\boldsymbol{x}}_p \in \mathbb{R}^{N\times 1}$
$\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{1}$
$\hat{\boldsymbol{w}}[0] \leftarrow \boldsymbol{1}$
$\hat{\epsilon}_r[0] \leftarrow 1$
$i \leftarrow 1$
**While** $\hat{\epsilon}_r[0] \neq 1$ **do**
$i \leftarrow i + 1$
$\hat{\epsilon}_r[i] \leftarrow \min\left(\hat{\epsilon}_r[i-1], \frac{1}{N} L_{k+1}(\hat{\boldsymbol{x}}[i-1])\right)$
$\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{D}(\hat{\boldsymbol{w}}[i-1])\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{AD}(\hat{\boldsymbol{w}}[i-1])\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b}$
$\hat{\boldsymbol{w}}[i] \leftarrow (\hat{\boldsymbol{x}}^2[i] + \hat{\epsilon}_r^2[i]\boldsymbol{1})^{1-\frac{1}{2}p}$
**end while**
**return** $\hat{\boldsymbol{x}}[i]$

Algorithm 1 is different from [19] in two aspects.

First, the regularization parameter $\epsilon_r[i]$ in [19] is computed from the updated $\hat{x}[i]$ at the $i$-th iteration, which is unavailable. Second, the procedure addressed in [19] considers only $p = 1$.

Next, let the weight$w_n$ in Section II be $|\hat{x}_n[i]|^{p-2}$, i.e.,

$$w_n[i] = |\hat{x}_n[i]|^{p-2}, \quad (16)$$

where $i$ is the index of iteration, $p$ is the exponent of an $\ell_p$ norm, and $\hat{x}_n[i]$ is the $n$-th element of $\hat{\boldsymbol{x}}[i] \in \mathbb{R}^{N\times 1}$, i.e.,

$$\hat{\boldsymbol{x}}[\mathrm{i}] = \left[\hat{x}_1[i]\ \hat{x}_2[i] \ldots, \hat{x}_N[i]\right]^{\mathrm{T}}. \quad (17)$$

An alternative optimization for the $i$-th iteration can be written in the form of

$$\hat{\boldsymbol{x}}[i] = \underset{\boldsymbol{u}}{\operatorname{argmin}} \sum_{n=1}^{N} |\hat{x}_n[i-1]|^{p-2} u_n^2, \text{ s.t. } \boldsymbol{Au} = \boldsymbol{b}. \quad (18)$$

Following from the same way as Eq. (9) and Eq. (11) one would arrive at

$$\hat{\boldsymbol{x}}[\mathrm{i}] = \boldsymbol{D}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{AD}^{\mp}(|\hat{\boldsymbol{x}}[i-1]|^{p-2})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{b}. \quad (19)$$

In this work, an algorithm is proposed based on Eq. (19) as follows.

**Algorithm 2.** IRLS with diagonal regularization

**Input:** $\boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{b} \in \mathbb{R}^{M\times 1}, p \in (0,1]$, $N_{\max} \in \mathbb{Z}_+^{1\times 1}$, $\epsilon_{\min} \in \mathbb{R}_+^{1\times 1}$
**Output:** $\hat{\boldsymbol{x}}_p \in \mathbb{R}^{N\times 1}$
$\hat{\boldsymbol{x}}[0] \leftarrow \boldsymbol{1}$
$i \leftarrow 0$
$\epsilon_{\hat{\mathrm{x}}} \leftarrow \epsilon_{\min} + 1$
**While** $\epsilon_{\hat{\mathrm{x}}} > \epsilon_{\min} \wedge i \leq N_{\max}$ **do**
$i \leftarrow i + 1$
$\hat{\boldsymbol{x}}[i] \leftarrow \boldsymbol{D}^{\mp}(|\boldsymbol{x}|^{p-2})\boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{AD}^{\mp}(|\boldsymbol{x}|^{p-2})\boldsymbol{A}^{\mathrm{T}}\right)^{-1}\boldsymbol{b}|_{\boldsymbol{x}=\hat{\boldsymbol{x}}[i-1]}$
$\epsilon_{\hat{\mathrm{x}}} \leftarrow \dfrac{\|\hat{\boldsymbol{x}}[i] - \hat{\boldsymbol{x}}[i-1]\|_2}{\|\hat{\boldsymbol{x}}[i-1]\|_2}$
**end while**
**return** $\hat{\boldsymbol{x}}[i]$

Since $\hat{x}_n[i]$ has a high probability to be nonzero during the iteration, the definite inverse from Eq. (10) for the matrix $\boldsymbol{D}(|\boldsymbol{x}|^{p-2})$ can be replaced by

$$\begin{aligned}\boldsymbol{D}^{\mp}(|\boldsymbol{x}|^{p-2}) &= \boldsymbol{D}^{-1}(|\boldsymbol{x}|^{p-2}) \\ &= \boldsymbol{D}(|\boldsymbol{x}|^{2-p}). \end{aligned} \quad (20)$$

Since the matrix $\boldsymbol{AD}^{\mp}(|\hat{\boldsymbol{x}}_n[i-1]|^{p-2})\boldsymbol{A}^{\mathrm{T}}$ can be ill-conditioned, it can be approximated by

$$\begin{aligned}&\boldsymbol{AD}^{\mp}(|\hat{\boldsymbol{x}}[\mathrm{i}-1]|^{p-2})\boldsymbol{A}^{\mathrm{T}} \\ &\approx \lim_{\varepsilon\to 0}(\boldsymbol{AD}^{-\boldsymbol{1}}(|\hat{\boldsymbol{x}}[\mathrm{i}-1]|^{p-2})\boldsymbol{A}^{\mathrm{T}} + \varepsilon\boldsymbol{I})^{-1}, \end{aligned} \quad (21)$$

where $\varepsilon \in \mathbb{R}_+^{1\times 1}$ is a small positive constant close to zero.

## IV. Numerical Examples

All numerical simulation in this work is conducted using Python language.

### A. Random Number Generation

Details of random number generation are setup as same as in [28, 29].

### B. Algorithmic Comparison

The methods intended to numerical comparison include

- linear least squares (LLS); the estimate given by $\hat{\boldsymbol{x}}_{\mathrm{LLS}} = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{AA}^{\mathrm{T}})^{-1}\boldsymbol{b}$,
- iterative hard thresholding (IHT); the estimate $\hat{\boldsymbol{x}}[i]$ given by Blumensath and Davies [30, 31],
- Iteratively reweighted $\ell_1$ (IRL1): the estimate $\hat{\boldsymbol{x}}[i]$ given by Candès *et al.* [32] whose convex optimization problem is solved by an open-source Python-embedded modeling language for convex optimization (known as CVXPY) [33],
- IRLS: the estimate $\hat{\boldsymbol{x}}[i]$ given by Algorithm 1.
- IRLS-DR: the estimate $\hat{\boldsymbol{x}}[i]$given by Algorithm 2.

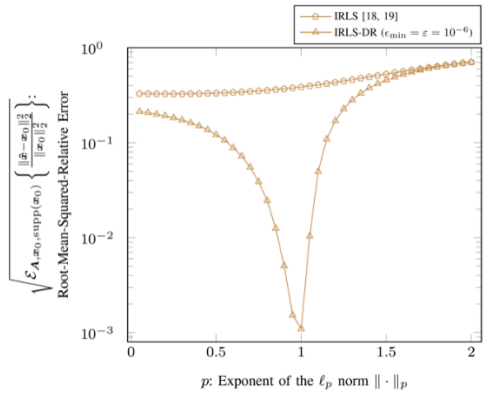


Figure 2. RMSRE as a function of $p$ from $N_{\mathrm{R}}$=100,000 independent runs for $K$ = 32, $M$ = 128, and $N$ = 256.

In Fig. 2, the RMSRE is shown as a function of the norm exponent $p$. The new IRLS has a sharp decrease to the lowest error around $p = 1$. For $p = 2$, the new IRLS provides the same error as the conventional one.

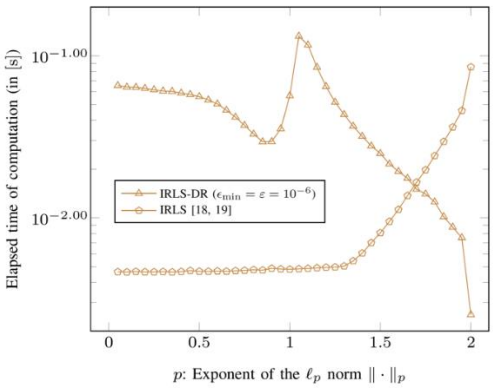


Figure 3. Elapsed time of computational as a function of $p$ from $N_{\mathrm{R}}$=100,000 independent runs for $K$ = 32, $M$ = 128, and $N$ = 256.

In Fig. 3, the computational time is shown as a function of the norm exponent $p$. One can see that the IRLS-DR generally requires more computational time than the conventional IRLS, except for $p > 1.7$.

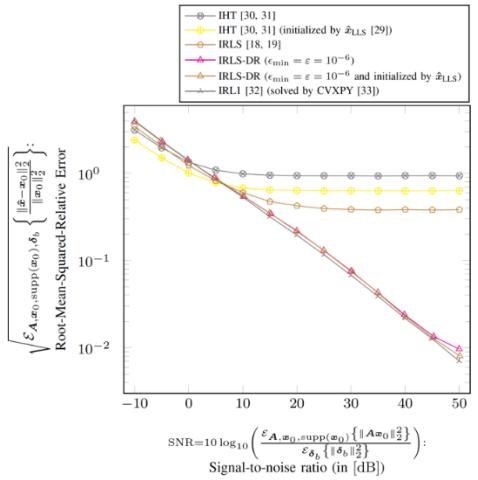


Figure 4. RMSRE as a function of SNR from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0}$=32, $N_{\mathrm{supp}(\boldsymbol{x}_0)}$ = 32, $N_{\boldsymbol{\delta}_b}$ = 100, and $N_{\mathrm{R}} = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b}$ = 102,400 independent runs for $K$ = 32, $M$ = 128, and $N$ = 256, $p$ = 0.9, and $\epsilon_{\min} = \epsilon = 10^{-6}$.

In Fig. 4, the RMSRE is shown as a function SNR. This is the complete scenario, where the additive noise in Fig. 1 takes place. The IRLS-DR and the IRL1 decreases with the SNR continuously, while the IHTs and the former IRLS do not. The IRL1 outperforms the new IRLS by a little amount of margin. However, as seen in Fig. 5, the IRL1 needs more computational time than the new IRLS. In addition, one can hardly see any benefit of the LLS initialization to the new IRLS in Fig. 4.

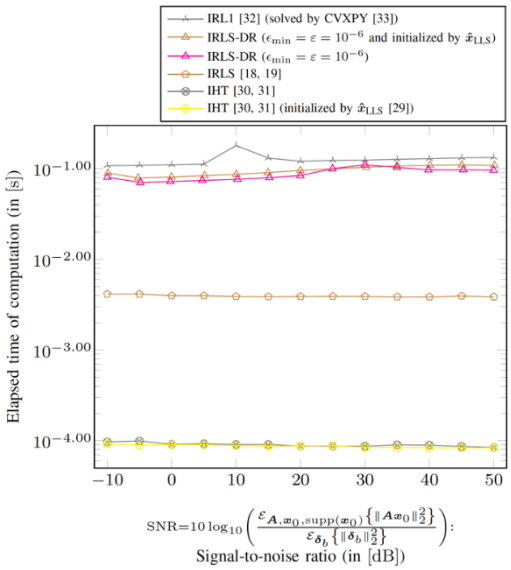


Figure 5. Elapsed time of computation as a function of SNR from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0}$ = 32, $N_{\mathrm{supp}(\boldsymbol{x}_0)}$ = 32, $N_{\boldsymbol{\delta}_b}$ = 100, and $N_{\mathrm{R}} = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b}$ = 102,400 independent runs for $K$ = 32, $M$ = 128, and $N$ = 256, $p$ = 0.9, and $\epsilon_{\min} = \epsilon = 10^{-6}$.

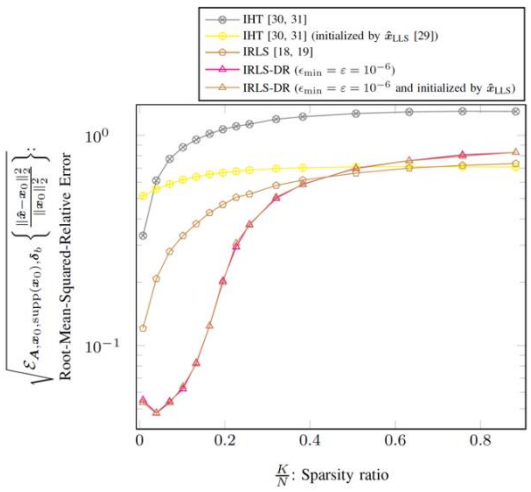


Figure 6. RMSRE as a function of sparsity ratio from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = \left\lceil \frac{10^3}{K} \right\rceil$, $N_{\mathrm{supp}(\boldsymbol{x}_0)} = K$, $N_{\boldsymbol{\delta}_b}$ = 100, and $N_{\mathrm{R}} = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b}$ = independent runs for $M$ = 128, $N$ = 256, $p$ = 0.9, SNR ≈ 30[dB], and $\epsilon_{\min} = \epsilon = 10^{-6}$.

In Fig. 6, the RMSRE is shown as a function of the sparsity ratio $\frac{K}{N}$. The new IRLS provides lower RMSRE than the conventional IRLS, yet at the expense of more computational time as seen in Fig. 7.

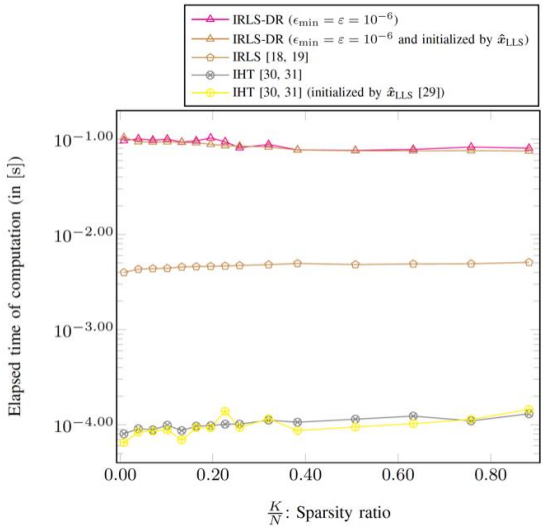


Figure 7. Elapsed time of computation as a function of sparsity ratio from $N_{\boldsymbol{A}} = N_{\boldsymbol{x}_0} = \left\lceil \frac{10^3}{K} \right\rceil$, $N_{\mathrm{supp}(\boldsymbol{x}_0)} = K$, $N_{\boldsymbol{\delta}_b} = 100$, and $N_{\mathrm{R}} = N_{\boldsymbol{A}} N_{\mathrm{supp}(\boldsymbol{x}_0)} N_{\boldsymbol{\delta}_b}$ = independent runs for $M = 128$, $N = 256$, $p = 0.9$, SNR ≈ 30[dB], and $\epsilon_{\min} = \epsilon = 10^{-6}$.

Next, an image reconstruction is considered. The $(m,n)$-th pixel of the image $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ can be found from the two-dimensional inverse discrete cosine transform (IDCT) of the image spectrum matrix $\boldsymbol{S} \in \mathbb{R}^{M \times N}$ [34], i.e.,

$$x_{m,n} = [\mathcal{C}^{-1}\{\boldsymbol{S}\}]_{m,n} = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{M-1} \alpha_{k_1} \alpha_{k_2} s_{k_1,k_2} \cos\left(\frac{1}{2M}(2m+1)k_1\right) \cos\left(\frac{1}{2N}(2n+1)k_2\right), \quad (22)$$

where $\mathcal{C}^{-1}\{\cdot\}$ is the IDCT operator and $s_{k_1,k_2}$ is the $(k_1, k_2)$-element of $\boldsymbol{S}$. The sensing matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \boldsymbol{\Phi}(\boldsymbol{C}_N \otimes \boldsymbol{C}_M), \quad (23)$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{M \times (MN)}$ is the sampling matrix and $\boldsymbol{C}_M \in \mathbb{R}^{M \times N}$ is the IDCT matrix, which is given by

$$\boldsymbol{C}_M = \mathcal{C}^{-1}\{\boldsymbol{I}_M\}, \quad (24)$$

and $\boldsymbol{C}_N \in \mathbb{R}^{M \times N}$ is the IDCT matrix, which is given by

$$\boldsymbol{C}_N = \mathcal{C}^{-1}\{\boldsymbol{I}_N\}. \quad (25)$$

Peak Signal-to-Noise Ratio (PSNR) is defined by

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{255}{\sqrt{\frac{1}{MN}\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_2^2}}\right). \quad (26)$$

The original image is shown in Fig. 8 which contains the data of size $N = 1534 \times 1433 \times 3 = 6594666$. The data of this size requires so much computational time in a computer simulation. This image is converted to a grayscale image. Furthermore, to reduce the computational burden, the grayscale image is down sampled to $153 \times 143$ pixels in Fig. 9, which results in $N = 153 \times 143 = 21879$.

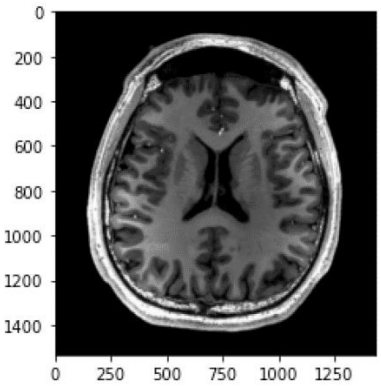


Figure 8. An original MRI image of a human brain with the size of 1534×1433 pixels and 3-bit color depth in .jpg format.

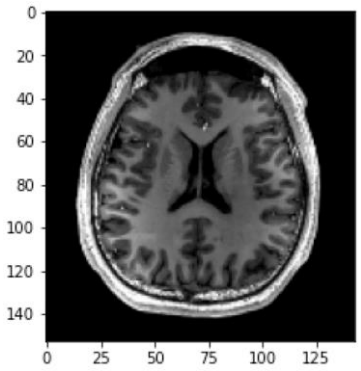


Figure 9. The MRI image of the human brain converted to the grayscale and down sampled to 153×143 pixels.

Assume that the length of the compressed data to be of the size $M = 10940$, which is the same value for $K$. The compression ratio thus becomes $\frac{1}{N}(N - M) = 0.49997714703597057$, which is still less than the Nyquist's sampling rate.

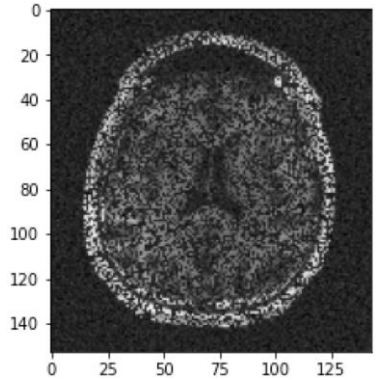


Figure 10. IHT with $N_{\max} = 100$ iterations, $\epsilon_{\min} = 10^{-6}$, $c = 0.1$, $\kappa = \frac{1}{1-c} + 0.1$, $\gamma = \frac{1}{\kappa(1-c)}$, and $\hat{\boldsymbol{x}}[0] = \boldsymbol{1}$.

The IHT algorithm reconstructs the compressed image, which is in Fig. 10, with the performance in Table I.

Initialized by the LLS estimate, the IHT algorithm again recovers the compressed image, which is in Fig. 11, with the performance in Table II. One can see that the LLS

initialization gives better reconstruction error performance at the expense of the computational time.

TABLE I. RECONSTRUCTION PERFORMANCE OF THE IHT IN FIG. 10

| Performance | Value |
|---|---|
| normalized error $\frac{\|\hat{\boldsymbol{X}}_{\mathrm{IHT}}-\boldsymbol{X}_0\|_2}{\|\boldsymbol{X}_0\|_2}$ | 0.546294822456923 |
| Peak signal-to-noise ratio | 13.419542261080572 [dB] |
| computational time | 0.1734757089870982 [s] |

TABLE II. RECONSTRUCTION PERFORMANCE OF THE IHT IN FIG. 11

| Performance | Value |
|---|---|
| normalized error $\frac{\|\hat{\boldsymbol{X}}_{\mathrm{IHT}}-\boldsymbol{X}_0\|_2}{\|\boldsymbol{X}_0\|_2}$ | 0.5013291943650706 |
| Peak signal-to-noise ratio | 13.49846419956603 [dB] |
| computational time | 0.3722585839859 6756 [s] |

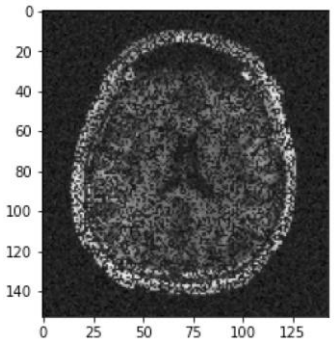

Figure 11. IHT with $N_{\max}$ = 100 iterations, $\epsilon_{\min} = 10^{-6}$, $c$ = 0.1, $\kappa = \frac{1}{1-c} + 0.1$, $\gamma = \frac{1}{\kappa(1-c)}$, and $\hat{\boldsymbol{x}}[0] = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}\boldsymbol{b}$.

The conventional IRLS method provides the reconstructed image in Fig. 12 and the associating performance in Table III.

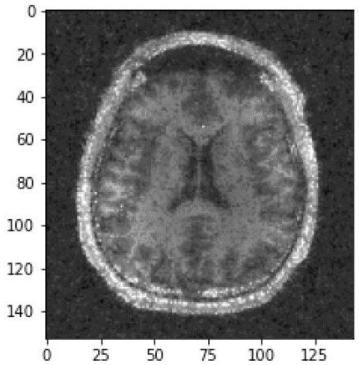

Figure 12. IRLS with $N_{\max}$ = 100 iterations, $\epsilon_{\min} = 10^{-6}$, $p$ = 0.9, $\hat{\boldsymbol{w}}[0] = \mathbf{1}$, and $\hat{\boldsymbol{x}}[0] = \mathbf{1}$.

TABLE III. RECONSTRUCTION PERFORMANCE OF THE IRLS IN FIG. 12

| Performance | Value |
|---|---|
| normalized error $\frac{\|\hat{\boldsymbol{X}}_{\mathrm{IRLS}}-\boldsymbol{X}_0\|_2}{\|\boldsymbol{X}_0\|_2}$ | 0.13054135154075275 |
| Peak signal-to-noise ratio | 20.261949757237478 [dB] |
| computational time | 882.3728725830151 [s] |

Comparing Table II to Table III, one can find that the IRLS approach provides better reconstruction performance than the IHT algorithm. However, it takes more time in the computation than the IHT approach.

Fig. 13 is the acquired image given by the IRLS-DR technique with the corresponding performance in Table IV. One can see that the IRLS-DR method gives less normalized error and more PSNR than the conventional IRLS algorithm. However, the computational time is required more in the IRLS- DR method.

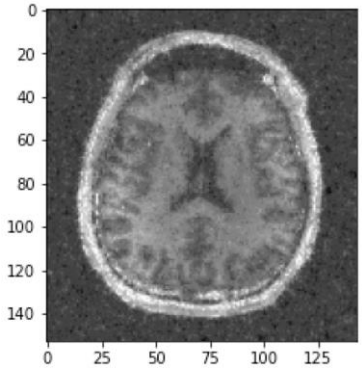

Figure 13. IRLS-DR with $N_{\max}$ = 100 iterations, $\epsilon_{\min} = 10^{-6}$, $p$ = 0.9, $\epsilon = 10^{-6}$, and $\hat{\boldsymbol{x}}[0] = 10^{-5}\mathbf{1}$.

TABLE IV. RECONSTRUCTION PERFORMANCE OF THE IRLS-DR IN FIG. 13

| Performance | Value |
|---|---|
| normalized error $\frac{\|\hat{\boldsymbol{X}}_{\mathrm{IRLS-DR}}-\boldsymbol{X}_0\|_2}{\|\boldsymbol{X}_0\|_2}$ | 0.07993784956037091 |
| Peak signal-to-noise ratio | 21.194630017158865 [dB] |
| computational time | 22889.644447542003 [s] |

TABLE V. RECONSTRUCTION PERFORMANCE OF THE IRL1 IN FIG. 14

| Performance | Value |
|---|---|
| normalized error $\frac{\|\hat{\boldsymbol{X}}_{\mathrm{IRL1}}-\boldsymbol{X}_0\|_2}{\|\boldsymbol{X}_0\|_2}$ | 0.07596854561346658 |
| Peak signal-to-noise ratio | 21.64997462101646 [dB] |
| computational time | 73293.26560041701 [s] |

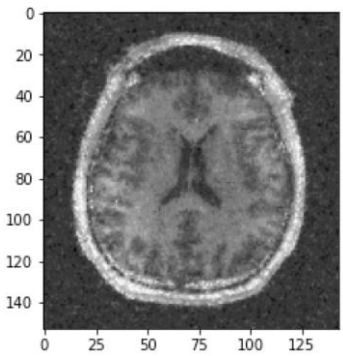

Figure 14. IRL1 with $N_{\max}$ = 100 iterations, $\epsilon_{\min} = 10^{-6}$, $p$ = 0.9, $\epsilon = 10^{-6}$, and $\hat{\boldsymbol{x}}[0] = 10^{-5}\mathbf{1}$.

In Fig. 14, the IRL1 algorithms gives a recovered image with the performance in Table V. One can see that the IRL1 algorithm provides better image than the IRLS method, but the computational time is approximately three times more than that of the IRLS-DR approach. Our companion works relating to compressive sensing in other applications can be seen in [35, 36].

## V. Conclusion

The numerous data is stringent to the flow rate in the information era. Under the limited amount of the available resource, there is a demand in compressing the data and the recovery of the compressed data. The second step, which is the reconstruction of the compressed signal, is crucial in compressed sensing. A method that can provide high accuracy in the signal recovery remains as an open problem. In this work, a closed-form solution of the IRLS optimization problem is derived. It is found however that a matrix for an inverse might be subject to an ill condition. This issue can be resolved by adding the diagonal regularization to that matrix. Numerical results illustrate that the error given by the new proposed IRLS method is obviously lower than that by the conventional IRLS algorithm. Unfortunately, the IRLS-DR algorithm requires more computational time than the conventional IRLS approach. The implication of this work is that the use of exponent norm can increase the accuracy of the signal reconstruction.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

B. Tausiesakul conducted the research. K. Asavaskulkiet analyzed the data. Both of them wrote the paper. All authors had approved the final version.

## References


[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 1289–1306, Apr. 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, no. 1, pp. 49–67, 2006.

[4] R. G. Baraniuk, V. Cevher, and M. B. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 959–971, Jun. 2010.

[5] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *J. Mach. Learn. Res.*, vol. 12, no. 103, pp. 3371–3412, 2011.

[6] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proc. 26th Int. Conf. Mach. Learn. (ICML 2009)*, Montreal, Canada, Jun. 2009, pp. 433–440.

[7] V. M. Patel and R. Chellappa, *Sparse Representations and Compressive Sensing for Imaging and Vision*, New York, NY: Springer Science, 2013.

[8] M. H. Conde, *Compressive Sensing for the Photonic Mixer Device: Fundamentals, Methods and Results*, Wiesbaden, Germany: Springer Fachmedien, 2017.

[9] A. K. Mishra and R. S. Verster, *Compressive Sensing Based Algorithms for Electronic Defence*, Switzerland: Springer International, 2017.

[10] M. Testa, D. Valsesia, T. Bianchi, and E. Magli, *Compressed Sensing for Privacy-Preserving Data Processing*, Singapore: Springer Nature, 2019.

[11] M. Amin, *Compressive Sensing for Urban Radar*, Boca Raton, FL: CRC Press, 2015.

[12] A. D. Maio, Y. C. Eldar, and A. M. Haimovich, *Compressed Sensing in Radar Signal Processing*, Cambridge, UK: Cambridge University Press, 2020.

[13] C. Chen, *Compressive Sensing of Earth Observations, ser. Signal and Image Processing of Earth Observations*, Boca Raton, FL: CRC Press, 2017.

[14] Z. Han, H. Li, and W. Yin, *Compressive Sensing for Wireless Networks*, Cambridge, UK: Cambridge University Press, 2013.

[15] L. Kong, B. Wang, and G. Chen, *When Compressive Sensing Meets Mobile Crowdsensing*, Singapore: Springer Nature, 2019.

[16] R. M. Thanki, V. J. Dwivedi, and K. R. Borisagar, *Multibiometric Watermarking with Compressive Sensing Theory: Techniques and Applications*, Switzerland: Springer International, 2018

[17] M. Khosravy, N. Dey, and C. A. Duque, *Compressive Sensing in Healthcare*, London, UK: Academic Press, 2020, vol. 11.

[18] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. 2008 (ICASSP 2008)*, Las Vegas, NV, Mar./Apr. 2008, pp. 3869–3872.

[19] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Comm. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.

[20] M.-J. Lai, Y. Xu, and W. Yin, "Improved iteratively reweighted least squares for unconstrained smoothed q minimization," *SIAM J. Numer. Anal.*, vol. 51, no. 2, pp. 927–957, Nov. 2013.

[21] N. Bi and K. Liang, "Iteratively reweighted algorithm for signals recovery with coherent tight frame," *Math. Methods Appl. Sci.*, vol. 41, no. 14, pp. 5481–5492, Sep. 2018.

[22] Z. Zhou and J. Yu, "A new nonconvex sparse recovery method for compressive sensing," *Front. Appl. Math. Stat.*, vol. 5, pp. 1–11, Mar. 2019.

[23] K. Liang and M. J. Clay, "Iterative re-weighted least squares algorithm for $l_p$-minimization with tight frame and $0 < p \leq 1$," *Linear Algebra Its Appl.*, vol. 581, no. 15, pp. 413–434, Nov. 2019.

[24] Y. Liu, Z. Zhu, and B. Zhang, "Improved iteratively reweighted least squares algorithms for sparse recovery problem," *IET Image Process.*, vol. 16, pp. 1324–1340, Nov. 2022.

[25] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York, NY: John Wiley & Sons, 2000.

[26] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[27] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing*, Norwood, MA: Artech House, 2005.

[28] B. Tausiesakul, "Method of Lagrange multipliers for normalized zero norm minimization," *Math. Prob. Eng.*, vol. 2022, pp. 1–10, Jan. 2022.

[29] B. Tausiesakul, "Iterative hard thresholding using least squares initialization," in *Proc. IEEE Int. Conf. Cummun. Syst. Netw. Technol. 2022 (CSNT 2022)*, Indore, India, Apr. 2022, pp. 612–615.

[30] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 629–654, Dec. 2008.

[31] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[32] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted $\ell_1$ minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.

[33] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 2909–2913, Jan. 2016.

[34] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2008.

[35] B. Tausiesakul and K. Asavaskulkiet, "Fractional norm regularization using inverse perturbation," *Mech. Syst. Signal Process.*, vol. 199, Sep. 2023.

[36] B. Tausiesakul and K. Asavaskulkiet, "Soft thresholding using Moore–Penrose inverse," *IEEE. Trans. Instrum. Meas.*, vol. 72, Jun. 2023.