

Article

Infrared Image Deblurring Based on Lp-Pseudo-Norm and High-Order Overlapping Group Sparsity Regularization

Zhen Ye, Xiaoming Ou, Juhua Huang and Yingpin Chen *

College of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China

* Correspondence: cyp1707@mnnu.edu.cn; Tel.: +86-139-0696-8591

Abstract: A traditional total variation (TV) model for infrared image deblurring amid salt-and-pepper noise produces a severe staircase effect. A TV model with low-order overlapping group sparsity (LOGS) suppresses this effect; however, it considers only the prior information of the low-order gradient of the image. This study proposes an image-deblurring model (Lp_HOGS) based on the LOGS model to mine the high-order prior information of an infrared (IR) image amid salt-and-pepper noise. An Lp-pseudo-norm was used to model the salt-and-pepper noise and obtain a more accurate noise model. Simultaneously, the second-order total variation regular term with overlapping group sparsity was introduced into the proposed model to further mine the high-order prior information of the image and preserve the additional image details. The proposed model uses the alternating direction method of multipliers to solve the problem and obtains the optimal solution of the overall model by solving the optimal solution of several simple decoupled subproblems. Experimental results show that the model has better subjective and objective performance than Lp_LOGS and other advanced models, especially when eliminating motion blur.

Keywords: infrared image deblurring; overlapping group sparse total variation; Lp-pseudo-norm; salt-and-pepper noise



Citation: Ye, Z.; Ou, X.; Huang, J.; Chen, Y. Infrared Image Deblurring Based on Lp-Pseudo-Norm and High-Order Overlapping Group Sparsity Regularization. *Algorithms* **2022**, *15*, 327. <https://doi.org/10.3390/a15090327>

Academic Editor: Frank Werner

Received: 17 August 2022

Accepted: 9 September 2022

Published: 14 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Images are a primary source of information because they contain large volumes of information and are well-aligned with the cognitive functions of the brain. In the absence of visible light, infrared (IR) radiation generated by an object can be converted into visible IR images using an IR thermal imaging system. The brightness of each pixel in the image corresponds to the change in the intensity of the object's radiation energy [1]. IR thermal-imaging systems are widely used in biomedicine, military, industrial, and agricultural applications, owing to their strong environmental adaptability, concealment, anti-interference, and identification abilities. However, they have disadvantages such as high noise, low contrast, and blurring. Among the various types of noise, the salt-and-pepper and the Gaussian noises have the greatest impact on IR images [2].

Salt-and-pepper noise is a common random noise exhibiting sparsity in mathematics and statistics. The common methods for removing salt-and-pepper noise include median filter, partial differential equation (PDE) model-based methods, and total variation (TV) model-based methods. Although the median filter method can effectively remove salt-and-pepper noise, it produces deblurred images with incomplete image details [3]. In the last decade, PDE-based models have been developed for various physical applications in image restoration [4]. TV-based models can answer fundamental questions related to image restoration better than other models [5]. The main advantage of this method is that it preserves the image edge. Traditional TV models, including anisotropic TV (ATV) [6] and isotropic TV (ITV) [7], can only approximate the piecewise constant function, causing staircase effects in the restored images [8,9]. Scholars have proposed several variants of the TV model to alleviate the staircase effect. For example, Liu et al. used the L1

norm to describe the statistical characteristics of noise and introduced overlapping group sparsity TV into the salt-and-pepper noise model, achieving good results [10]. However, many non-convex models have achieved better sparse constraints in practical applications than those based on the L1 norm at low sampling rates. Therefore, Yuan and Ghanem proposed a new sparse optimization model (L0TVPADMM) that used an L0 norm as the fidelity term to solve the reconstruction problem based on the TV model [11]. Adam et al. proposed the HNHOTV-OGS method, which combined non-convex high-order total variation and overlapping group sparse regularization [12]. Chartrand proposed a non-convex optimization problem with the minimization of the Lp-pseudo-norm as the objective function [13,14]. Subsequently, Chartrand and Staneva provided a theoretical condition of the Lp-pseudo-norm to recover an arbitrary sparse signal [15]. Wu and Chen [16] and Wen et al. [17] theoretically demonstrated the superiority of methods based on the Lp-pseudo-norm. The Lp-pseudo-norm exhibited a stronger sparse representation ability than the L1 norm. Therefore, it has garnered extensive research interest in recent years [18–23]. For example, Lin et al. imposed sparse constraints on the high-order gradients of the image, combined the Lp-pseudo-norm with the total generalized variation model, and proposed an image restoration algorithm that achieved good performance [22]. Based on the mathematical model in [22], Wang et al. replaced the L1 norm with the Lp-pseudo-norm to describe the statistical characteristics of salt-and-pepper noise and proposed an image denoising method based on the Lp-pseudo-norm with low-order overlapping group sparsity (Lp_LOGS) [23].

Among these algorithms, Lp_LOGS performed the best in removing salt-and-pepper noise. However, this method only considers the prior information of the low-order gradient of an image. This study proposes an image-deblurring model based on the Lp_LOGS model to mine the high-order prior information of an IR image containing salt-and-pepper noise, called Lp_HOGS. The second-order TV regularization term was introduced with overlapping group sparsity into the LOGS model, and the Lp-pseudo-norm was retained in the salt-and-pepper noise model. Experimental results show that compared with Lp_LOGS and other advanced models, the proposed Lp_HOGS model demonstrated better peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and gradient magnitude similarity deviation (GMSD). Additionally, the proposed model retained more image details, making the visual effect greater in similarity to the original image. Finally, the proposed model facilitated the subsequent target recognition and tracking processing of the image.

2. Materials and Methods

2.1. The Background to Deblurring Algorithms

2.1.1. The Lp-Pseudo-Norm

The Lp-pseudo-norm represents the distance in the vector space and is a generalized concept of “distance.” The Lp norm of the matrix X is defined as $\|X\|_p = \left(\sum_{i=1}^M \sum_{j=1}^N |X_{ij}|^p \right)^{\frac{1}{p}}$, when p is 1 and 2, and corresponds to the L1 and L2 norms, respectively. The Lp-pseudo-norm is defined as $\|X\|_p^p = \sum_{i=1}^M \sum_{j=1}^N |X_{ij}|^p$. This study focused on the case where $0 < p < 1$. The contour maps of different norms are presented in Figure 1. The Lp-pseudo-norm has one more degree of freedom than the L1 and L2 norms, and the contour of the Lp-pseudo-norm was closest to the coordinate axis. Hence, the solution has a higher probability of being a point that is on or close to the axis. The Lp-pseudo-norm has a stronger sparse representation ability based on these reasons. Therefore, the Lp-pseudo-norm was introduced into the model for salt-and-pepper denoising in this study.

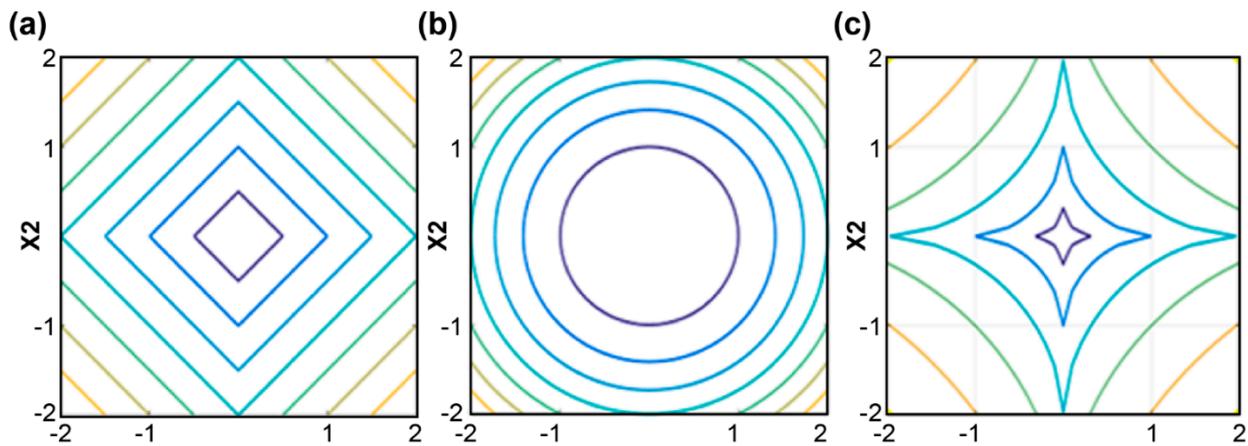


Figure 1. Contour maps of the different norms. (a) the L1 norm, (b) the L2 norm, and (c) the Lp-pseudo-norm.

2.1.2. The Overlapping Group Sparse TV Regularization Term

The overlapping group sparse TV regularization term is expressed as:

$$R_{OGSTV}(F) = \varphi(K_h * F) + \varphi(K_v * F) \tag{1}$$

where $*$ represents the convolution operator, $K_h = [1, -1]$ is the first-order horizontal difference convolution kernel, F is the original image, and $K_v = [1, -1]^T$ is the first-order vertical difference convolution kernel. $\varphi(V)$ is the function for calculating the combined gradient and is expressed as:

$$\varphi(V) = \sum_{i=1}^N \sum_{j=1}^N \|\tilde{V}_{i,j,K,K}\|_2 \quad (V \in \mathbb{R}^{N \times N}) \tag{2}$$

where $\tilde{V}_{i,j,K,K}$ represents the overlapping group sparsity matrix, which is defined as:

$$\tilde{V}_{i,j,K,K} = \begin{pmatrix} V_{i-K_l, j-K_l} & V_{i-K_l, j-K_l+1} & \dots & V_{i-K_l, j+K_r} \\ V_{i-K_l+1, j-K_l} & V_{i-K_l+1, j-K_l+1} & \dots & V_{i-K_l+1, j+K_r} \\ \vdots & \vdots & \ddots & \vdots \\ V_{i+K_r, j-K_l} & V_{i+K_r, j-K_l+1} & \dots & V_{i+K_r, j+K_r} \end{pmatrix} \in \mathbb{R}^{K \times K} \tag{3}$$

where K represents the length or width of the matrix of the combined gradient, $K_l = \lfloor \frac{K-1}{2} \rfloor$, $K_r = \lfloor \frac{K}{2} \rfloor$, and $\lfloor \cdot \rfloor$ means round down to the nearest integer.

As indicated in Equation (3), the overlapping group sparse TV regularization term uses the gradient of the pixel point (i, j) as the center, constructs a $K \times K$ matrix, combines it using the L2 norm, and replaces the independent gradient of the pixel point. Compared with the traditional anisotropic TV regularization term, the overlapping group sparse TV regularization term fully mines the gradient information of each pixel and considers the structural information of the image, thereby increasing the difference between the smooth and the edge areas of the image.

2.1.3. The Lp_LOGS Model

Blurred images are generally considered images corrupted by blur kernels and additive noise. They can be represented by the following linear mathematical model:

$$G = H * F + \eta \tag{4}$$

where G represents the degraded image to be deblurred, H is the blur kernel, and η is the additive noise, specifically the salt-and-pepper noise in this study.

The application of the Lp_LOGS model to solve F is an ill-posed inverse problem, which is expressed as:

$$F = \underset{F}{\operatorname{argmin}} \|H * F - G\|_p^p + \mu [R_{OGSTV}(F)], \tag{5}$$

where $\|H * F - G\|_p^p$ is the fidelity term, $R_{OGSTV}(F)$ is the prior term, and μ is the balance coefficient used to balance the prior and the fidelity terms.

The Lp_LOGS model leverages the Lp-pseudo-norm and overlapping group sparse TV regularization term, vastly outperforming the traditional ATV model in terms of deblurring. However, the Lp_LOGS model only considers the overlapping group sparse constraints of the low-order gradient information of the image.

To mine the prior information of the high-order gradient of an image, the second-order overlapping group sparse TV regularization term was introduced into the Lp_LOGS model. The proposed novel deblurring model is expressed as:

$$F = \underset{F}{\operatorname{argmin}} \|H * F - G\|_p^p + \mu_1[\varphi(K_h * F)] + \mu_2[\varphi(K_v * F)] + \mu_3[\varphi(K_h * K_h * F)] + \mu_4[\varphi(K_v * K_v * F)] + \mu_5[\varphi(K_v * K_h * F)], \tag{6}$$

where $\mu_i (i = 1, 2, 3, 4, 5)$ represents the balance coefficient, $K_h * K_h$ is the second-order horizontal difference convolution kernel, $K_v * K_v$ is the second-order vertical difference convolution kernel, and $K_v * K_h$ is the second-order mixed difference convolution kernel.

This model was named Lp_HOGS to represent a deblurring model based on the Lp-pseudo-norm with high-order overlapping group sparsity regularization.

2.2. The Solution of the Lp_HOGS Model

The alternating direction method of multipliers (ADMM) was used to solve the Lp_HOGS model. When solving this model, ADMM transformed the original complex problem into several relatively simple subproblems by introducing decoupling variables.

According to the ADMM solution framework, the intermediate variables $Z_0 = H * F - G, Z_1 = K_h * F, Z_2 = K_v * F, Z_3 = K_h * K_h * F, Z_4 = K_v * K_v * F, Z_5 = K_v * K_h * F$, were included and the original problem was transformed into an optimization problem with constraints, which is expressed as:

$$J = \underset{Z_0 \sim Z_5}{\min} \{ \|Z_0\|_p^p + \mu_1[\varphi(Z_1)] + \mu_2[\varphi(Z_2)] + \mu_3[\varphi(Z_3)] + \mu_4[\varphi(Z_4)] + \mu_5[\varphi(Z_5)] \}. \tag{7}$$

The corresponding Lagrange multiplier $\Lambda_i (i = 0, 1, 2, 3, 4, 5)$ and quadratic penalty coefficient $\beta_i (i = 0, 1, 2, 3, 4, 5)$ were used to transform Equation (7) into an unconstrained optimization problem, i.e., the augmented Lagrange objective function of the original problem, which is expressed as:

$$J = \max_{\Lambda_0 \sim \Lambda_5} \{ \underset{F, Z_0 \sim Z_5}{\min} \{ \|Z_0\|_p^p - \langle \beta_0 \Lambda_0, Z_0 - (H * F - G) \rangle + \frac{\beta_0}{2} \|Z_0 - (H * F - G)\|_2^2 + \mu_1[\varphi(Z_1)] - \langle \beta_1 \Lambda_1, Z_1 - K_h * F \rangle + \frac{\beta_1}{2} \|Z_1 - K_h * F\|_2^2 + \mu_2[\varphi(Z_2)] - \langle \beta_2 \Lambda_2, Z_2 - K_v * F \rangle + \frac{\beta_2}{2} \|Z_2 - K_v * F\|_2^2 + \mu_3[\varphi(Z_3)] - \langle \beta_3 \Lambda_3, Z_3 - K_h * K_h * F \rangle + \frac{\beta_3}{2} \|Z_3 - K_h * K_h * F\|_2^2 + \mu_4[\varphi(Z_4)] - \langle \beta_4 \Lambda_4, Z_4 - K_v * K_v * F \rangle + \frac{\beta_4}{2} \|Z_4 - K_v * K_v * F\|_2^2 + \mu_5[\varphi(Z_5)] - \langle \beta_5 \Lambda_5, Z_5 - K_v * K_h * F \rangle + \frac{\beta_5}{2} \|Z_5 - K_v * K_h * F\|_2^2 \} \}, \tag{8}$$

where $\langle X, Y \rangle$ represents the inner product of the X, Y matrices.

$$0 = \frac{\beta_i}{2} \|\Lambda_i\|_2^2 - \frac{\beta_i}{2} \|\Lambda_i\|_2^2 (i = 0, 1, 2, 3, 4, 5) \tag{9}$$

Subsequently, adding Equation (9) to the right side of Equation (8) obtains:

$$J = \max_{\Lambda_0 \sim \Lambda_5} \left\{ \min_{F, Z_0 \sim Z_5} \left\{ \|Z_0\|_p^p + \frac{\beta_0}{2} \|Z_0 - (H * F - G) - \Lambda_0\|_2^2 - \frac{\beta_0}{2} \|\Lambda_0\|_2^2 \right. \right. \\ + \mu_1 [\varphi(Z_1)] + \frac{\beta_1}{2} \|Z_1 - K_h * F - \Lambda_1\|_2^2 - \frac{\beta_1}{2} \|\Lambda_1\|_2^2 \\ + \mu_2 [\varphi(Z_2)] + \frac{\beta_2}{2} \|Z_2 - K_v * F - \Lambda_2\|_2^2 - \frac{\beta_2}{2} \|\Lambda_2\|_2^2 \\ + \mu_3 [\varphi(Z_3)] + \frac{\beta_3}{2} \|Z_3 - K_h * K_h * F - \Lambda_3\|_2^2 - \frac{\beta_3}{2} \|\Lambda_3\|_2^2 \\ + \mu_4 [\varphi(Z_4)] + \frac{\beta_4}{2} \|Z_4 - K_v * K_v * F - \Lambda_4\|_2^2 - \frac{\beta_4}{2} \|\Lambda_4\|_2^2 \\ \left. \left. + \mu_5 [\varphi(Z_5)] + \frac{\beta_5}{2} \|Z_5 - K_v * K_h * F - \Lambda_5\|_2^2 - \frac{\beta_5}{2} \|\Lambda_5\|_2^2 \right\} \right\}. \tag{10}$$

Each subproblem must first be solved to solve the objective function. As the introduced variables $Z_i (i = 0, 1, 2, 3, 4, 5)$, $\Lambda_i (i = 0, 1, 2, 3, 4, 5)$, and F were decoupled, the objective function corresponding to the subproblems of F became:

$$J_F = \min_F \left\{ \frac{\beta_0}{2} \|Z_0 - (H * F - G) - \Lambda_0\|_2^2 + \frac{\beta_1}{2} \|Z_1 - K_h * F - \Lambda_1\|_2^2 \right. \\ + \frac{\beta_2}{2} \|Z_2 - K_v * F - \Lambda_2\|_2^2 + \frac{\beta_3}{2} \|Z_3 - K_h * K_h * F - \Lambda_3\|_2^2 \\ \left. + \frac{\beta_4}{2} \|Z_4 - K_v * K_v * F - \Lambda_4\|_2^2 + \frac{\beta_5}{2} \|Z_5 - K_v * K_h * F - \Lambda_5\|_2^2 \right\}. \tag{11}$$

The convolution theorem was used to apply the Fourier transform to both sides of Equation (11), which is expressed as:

$$J_{\bar{F}} = \min_{\bar{F}} \left\{ \frac{\beta_0}{2} \|(\bar{H} \circ \bar{F} - \bar{G}) + \bar{\Lambda}_0 - \bar{Z}_0\|_2^2 + \frac{\beta_1}{2} \|\bar{K}_h \circ \bar{F} + \bar{\Lambda}_1 - \bar{Z}_1\|_2^2 \right. \\ + \frac{\beta_2}{2} \|\bar{K}_v \circ \bar{F} + \bar{\Lambda}_2 - \bar{Z}_2\|_2^2 + \frac{\beta_3}{2} \|\bar{K}_h \circ \bar{K}_h \circ \bar{F} + \bar{\Lambda}_3 - \bar{Z}_3\|_2^2 \\ \left. + \frac{\beta_4}{2} \|\bar{K}_v \circ \bar{K}_v \circ \bar{F} + \bar{\Lambda}_4 - \bar{Z}_4\|_2^2 + \frac{\beta_5}{2} \|\bar{K}_v \circ \bar{K}_h \circ \bar{F} + \bar{\Lambda}_5 - \bar{Z}_5\|_2^2 \right\}, \tag{12}$$

where \circ denotes the dot product operator and \bar{X} is the Fourier transform of the X matrix. The partial derivative of \bar{F} can be calculated using:

$$\frac{\partial J_{\bar{F}}}{\partial \bar{F}} = \beta_0 \bar{H}^* \circ ((\bar{H} \circ \bar{F} - \bar{G}) + \bar{\Lambda}_0 - \bar{Z}_0) + \beta_1 \bar{K}_h^* \circ (\bar{K}_h \circ \bar{F} + \bar{\Lambda}_1 - \bar{Z}_1) \\ + \beta_2 \bar{K}_v^* \circ (\bar{K}_v \circ \bar{F} + \bar{\Lambda}_2 - \bar{Z}_2) + \beta_3 (\bar{K}_h \circ \bar{K}_h)^* \circ (\bar{K}_h \circ \bar{K}_h \circ \bar{F} + \bar{\Lambda}_3 - \bar{Z}_3) \\ + \beta_4 (\bar{K}_v \circ \bar{K}_v)^* \circ (\bar{K}_v \circ \bar{K}_v \circ \bar{F} + \bar{\Lambda}_4 - \bar{Z}_4) \\ + \beta_5 (\bar{K}_v \circ \bar{K}_h)^* \circ (\bar{K}_v \circ \bar{K}_h \circ \bar{F} + \bar{\Lambda}_5 - \bar{Z}_5), \tag{13}$$

where X^* represents the conjugate matrix of the X matrix.

Here, Equation (13) is assumed to be equal to a zero matrix and can be rearranged as:

$$LHS \circ \bar{F} = RHS \tag{14}$$

where

$$LHS = \beta_0 \bar{H}^* \circ \bar{H} + \beta_1 \bar{K}_h^* \circ \bar{K}_h + \beta_2 \bar{K}_v^* \circ \bar{K}_v + \beta_3 (\bar{K}_h \circ \bar{K}_h)^* \circ (\bar{K}_h \circ \bar{K}_h) \\ + \beta_4 (\bar{K}_v \circ \bar{K}_v)^* \circ (\bar{K}_v \circ \bar{K}_v) + \beta_5 (\bar{K}_v \circ \bar{K}_h)^* \circ (\bar{K}_v \circ \bar{K}_h) \tag{15}$$

and

$$RHS = \beta_0 \bar{H}^* \circ (\bar{Z}_0 + \bar{G} - \bar{\Lambda}_0) + \beta_1 \bar{K}_h^* \circ (\bar{Z}_1 - \bar{\Lambda}_1) \\ + \beta_2 \bar{K}_v^* \circ (\bar{Z}_2 - \bar{\Lambda}_2) + \beta_3 (\bar{K}_h \circ \bar{K}_h)^* \circ (\bar{Z}_3 - \bar{\Lambda}_3) \\ + \beta_4 (\bar{K}_v \circ \bar{K}_v)^* \circ (\bar{Z}_4 - \bar{\Lambda}_4) + \beta_5 (\bar{K}_v \circ \bar{K}_h)^* \circ (\bar{Z}_5 - \bar{\Lambda}_5). \tag{16}$$

The updated formula of F is expressed as:

$$F = \mathcal{F}^{-1}(RHS./LHS) \tag{17}$$

where $\cdot /$ represents the dot division operator and $\mathcal{F}^{-1}(X)$ is the inverse Fourier transform of X .

The objective function to solve the subproblem Z_0 is expressed as:

$$J_{Z_0} = \min_{Z_0} \left\{ \|Z_0\|_p^p + \frac{\beta_0}{2} \|Z_0 - (H * F - G) - \Lambda_0\|_2^2 \right\} \tag{18}$$

According to the Lp shrinkage operator:

$$\text{shrink}_p(\xi, \tau) = \max\{|\xi| - \tau^{2-p}|\xi|^{p-1}, 0\} \frac{\xi}{|\xi|} \tag{19}$$

The updated formula of Z_0 is expressed as:

$$\begin{aligned} Z_0 &= \text{shrink}_p\left((H * F - G) + \Lambda_0, \frac{1}{\beta_0}\right) \\ &= \max\left\{ |(H * F - G) + \Lambda_0| - \beta_0^{p-2} |(H * F - G) + \Lambda_0|^{p-1}, 0 \right\} \frac{(H * F - G) + \Lambda_0}{|(H * F - G) + \Lambda_0|}. \end{aligned} \tag{20}$$

Additionally, the objective function of the subproblem Z_1 is expressed as:

$$J_{Z_1} = \min_{Z_1} \left\{ \mu_1 [\varphi(Z_1)] + \frac{\beta_1}{2} \|Z_1 - K_h * F - \Lambda_1\|_2^2 \right\} \tag{21}$$

According to the ADMM algorithm, the updated formula of Z_1 is expressed as:

$$Z_{1(n+1)}^{(k+1)} = \mathbf{mat}\left\{ \left[I + \frac{\mu_1}{\beta_1} D^2(Z_{1(n)}^{(k+1)}) \right]^{-1} \text{vec}\left(Z_{1(0)}^{(k+1)}\right) \right\} \quad (Z_{1(0)}^{(k+1)} = K_h * F + \Lambda_1^{(k)}) \tag{22}$$

where \mathbf{mat} is the vector matricization operator, vec is the matrix vectorization operator, $I \in \mathbb{R}^{N^2 \times N^2}$ is the identity matrix, and $D \in \mathbb{R}^{N^2 \times N^2}$ is the diagonal matrix with diagonal elements.

$$[D(U)]_{m,m} = \sqrt{\sum_{i=-K_l}^{K_r} \sum_{j=-K_l}^{K_r} \left\{ \sum_{k_1=-K_l}^{K_r} \sum_{k_2=-K_l}^{K_r} |U_{m-i+k_1, m-j+k_2}|^2 \right\}^{-\frac{1}{2}}} \tag{23}$$

where U represents an arbitrary matrix.

Similarly, the updated formulas of $Z_2 \sim Z_5$ can be obtained, which are expressed as:

$$\left\{ \begin{aligned} Z_{2(n+1)}^{(k+1)} &= \mathbf{mat}\left\{ \left[I + \frac{\mu_2}{\beta_2} D^2(Z_{2(n)}^{(k+1)}) \right]^{-1} \text{vec}\left(Z_{2(0)}^{(k+1)}\right) \right\} & (Z_{2(0)}^{(k+1)} &= K_v * F + \Lambda_2^{(k)}) \\ Z_{3(n+1)}^{(k+1)} &= \mathbf{mat}\left\{ \left[I + \frac{\mu_3}{\beta_3} D^2(Z_{3(n)}^{(k+1)}) \right]^{-1} \text{vec}\left(Z_{3(0)}^{(k+1)}\right) \right\} & (Z_{3(0)}^{(k+1)} &= K_h * K_h * F + \Lambda_3^{(k)}) \\ Z_{4(n+1)}^{(k+1)} &= \mathbf{mat}\left\{ \left[I + \frac{\mu_4}{\beta_4} D^2(Z_{4(n)}^{(k+1)}) \right]^{-1} \text{vec}\left(Z_{4(0)}^{(k+1)}\right) \right\} & (Z_{4(0)}^{(k+1)} &= K_v * K_v * F + \Lambda_4^{(k)}) \\ Z_{5(n+1)}^{(k+1)} &= \mathbf{mat}\left\{ \left[I + \frac{\mu_5}{\beta_5} D^2(Z_{5(n)}^{(k+1)}) \right]^{-1} \text{vec}\left(Z_{5(0)}^{(k+1)}\right) \right\} & (Z_{5(0)}^{(k+1)} &= K_v * K_h * F + \Lambda_5^{(k)}) \end{aligned} \right. \tag{24}$$

The objective function to solve subproblem Λ_0 is expressed as:

$$J_{\Lambda_0} = \max_{\Lambda_0} \left\{ - \langle \beta_0 \Lambda_0, Z_0 - (H * F - G) \rangle \right\} \tag{25}$$

According to the gradient ascent method, the updated formula of Λ_0 is expressed as:

$$J_{\Lambda_0} = \max_{\Lambda_0} \left\{ - \langle \beta_0 \Lambda_0, Z_0 - (H * F - G) \rangle \right\} \tag{26}$$

where γ is the learning rate.

The objective function to solve subproblem Λ_1 is expressed as:

$$J_{\Lambda_1} = \max_{\Lambda_1} \{ - \langle \beta_1 \Lambda_1, Z_1 - K_h * F \rangle \} \tag{27}$$

According to the gradient ascent method, the updated formula of Λ_1 is expressed as:

$$\Lambda_1^{(k+1)} = \Lambda_1^{(k)} + \gamma \beta_1 (K_h * F - Z_1^{(k+1)}) \tag{28}$$

Similarly, the updated formulas of $\Lambda_2 \sim \Lambda_5$ can be obtained, which are expressed as:

$$\begin{cases} \Lambda_2^{(k+1)} = \Lambda_2^{(k)} + \gamma \beta_2 (K_v * F - Z_2^{(k+1)}) \\ \Lambda_3^{(k+1)} = \Lambda_3^{(k)} + \gamma \beta_3 (K_h * K_h * F - Z_3^{(k+1)}) \\ \Lambda_4^{(k+1)} = \Lambda_4^{(k)} + \gamma \beta_4 (K_v * K_v * F - Z_4^{(k+1)}) \\ \Lambda_5^{(k+1)} = \Lambda_5^{(k)} + \gamma \beta_5 (K_v * K_h * F - Z_5^{(k+1)}) \end{cases} \tag{29}$$

Hence, the subproblems are solved.

Given the above descriptions, the specific description of the Lp_HOGS algorithm is shown in Algorithm 1.

Algorithm 1. Lp_ HOGS

Input Observed image G .

Output Deblurred image F .

Initialize:

$$k = 0, tol = 10^{-4}, err = 1, p, \gamma, F = 0, Z_i = 0, \Lambda_i = 0, \mu_j, \beta_i \\ (i = 0, 1, \dots, 5; j = 1, 2, \dots, 5).$$

- 1: **If** $err > tol$ **do**
 - 2: **Use** Equations (15)–(17) to **update** F ;
 - 3: **Use** Equation (20) to **update** Z_0 ;
 - 4: **Use** Equations (22) and (24) to **update** $Z_i (i = 1, 2, \dots, 5)$;
 - 5: **Use** Equation (26) to **update** Λ_0 ;
 - 6: **Use** Equations (28) and (29) to **update** $\Lambda_i (i = 1, 2, \dots, 5)$;
 - 7: $k = k + 1$;
 - 8: $err = \|F^{(k+1)} - F^{(k)}\|_2 / \|F^{(k)}\|_2$;
 - 9: **End if**
 - 10: **Return** $F^{(k)}$ as F .
-

where tol represents the threshold.

3. Results and Discussion

The IR test images used in the experiment were downloaded from the publicly available datasets found at <http://adas.cvc.uab.es/elektra/datasets/far-infra-red/> (accessed on 5 January 2022) and <http://www.dgp.toronto.edu/~\{ \}nmorris/data/IRData/> (accessed on 5 January 2022). This study evaluated the quality of the denoised images from subjective and objective aspects. The objective evaluation metrics used in this study are PSNR, SSIM, and GMSD, which are defined using Equations (30)–(32), respectively:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \|x_{ij} - y_{ij}\|_1^2} \right) \quad (\text{dB}), \tag{30}$$

$$SSIM = \frac{(2u_X u_Y + (255k_1)^2) (2\sigma_{XY} + (255k_2)^2)}{(u_X^2 + u_Y^2 + (255k_1)^2) (\sigma_X^2 + \sigma_Y^2 + (255k_2)^2)}, \tag{31}$$

where X and Y are the original and the restored image, respectively. u_X and u_Y represent the mean values of X and Y , respectively. σ_X^2 and σ_Y^2 represent the variances of X and Y , respectively. σ_{XY} represents the covariance of X and Y . k_1 and k_2 are constants to ensure that SSIM is not zero.

$$\text{GMSD} = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \left\{ [\text{GMS}]_{i,j} - \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [\text{GMS}]_{i,j} \right\}^2}, \quad (32)$$

where

$$[\text{GMS}]_{i,j} = \frac{2[m_X]_{i,j}[m_Y]_{i,j} + c}{[m_X]_{i,j}^2 + [m_Y]_{i,j}^2 + c}. \quad (33)$$

where c is a constant to ensure that the denominator is a non-zero number. $[m_X]_{i,j}$ and $[m_Y]_{i,j}$ refer to the gradient amplitude of the image in the horizontal and vertical directions, respectively.

Larger PSNR, closer-to-1 SSIM, and smaller GMSD values indicate better deblurring performance.

The variable parameters were preset before the experiment to focus on PSNR optimization before optimizing SSIM and GMSD. The variable parameters of the Lp_HOGS model were set as follows:

The maximum number of iterations and the learning rate γ were set to 500 and 0.618, respectively.

For each model, the balance coefficient $\mu_j (j = 1, 2, \dots, 5)$, the quadratic penalty coefficient $\beta_i (i = 0, 1, \dots, 5)$, and the Lp-pseudo-norm p were manually optimized to achieve the best deblurring effect on the IR image and ensure a fair experiment.

A sensitivity experiment analysis revealed that K was essential and the image quality indicators were optimal when $K = 3$ was the size of the matrix. The structured information of the image could not be fully mined if the value of K was insignificant. Conversely, unstructured information could be introduced if the value was excessively large.

3.1. The Comparison of Lp_HOGS with Lp_LOGS

Lp_HOGS and Lp_LOGS were compared to verify the effect of adding the second-order overlapping group sparse TV regularization term. We added 30%, 40%, and 50% salt-and-pepper noise to the Gaussian, box, and motion blurs to obtain nine degradation combinations and compared the quality of the deblurred images. Nine distinct IR images were selected as the test images. Images of a passerby, a station, a truck, a car, and some buildings were 384×288 pixels each, and images of a garden, some stairs, a corridor, and a zebra crossing were 506×408 pixels each.

3.1.1. The Gaussian Blur

The noise was generated by the MATLAB built-in function “noise (I, type, parameters)”. For example, 30% of the salt-and-pepper noise was set to noise (I, ‘salt & Pepper’, 0.3). We included 30%, 40%, and 50% of the salt-and-pepper noise in the test images with a 7×7 Gaussian blur. The experimental results are summarized in Table 1.

The three performance indicators obtained by the Lp_HOGS model were higher than those of the Lp_LOGS model; thereby demonstrating that the proposed method achieved better deblurring and denoising effects. The Lp_HOGS model achieved average PSNR values that were 0.304, 0.784, and 1.287 dB higher than those of the Lp_LOGS model when the salt-and-pepper noise was 30%, 40%, and 50%, respectively. Therefore, the Lp_HOGS model had a greater advantage when the noise levels increased.

The passerby images with different degrees of degradation are shown in Figure 2. The deblurring effects using Lp_LOGS and Lp_HOGS are compared in Figure 3.

Table 1. The Lp_HOGS and Lp_LOGS deblurring effects on images with the 7×7 Gaussian blur. The optimal indicators for each condition are denoted in bold to facilitate data observation.

Image	Noise Level (%)	Lp_LOGS			Lp_HOGS		
		PSNR (dB)	SSIM	GMSD	PSNR (dB)	SSIM	GMSD
Passerby	30	40.8630	0.9455	0.0165	41.1052	0.9511	0.0118
	40	39.8587	0.9397	0.0197	40.6369	0.9505	0.0139
	50	38.3621	0.9247	0.0310	39.7808	0.9422	0.0178
Station	30	39.7670	0.9466	0.0126	39.9526	0.9504	0.0083
	40	38.8663	0.9413	0.0146	39.6986	0.9497	0.0104
	50	37.7629	0.9323	0.0177	38.5159	0.9428	0.0131
Truck	30	39.6943	0.9472	0.0096	39.8380	0.9502	0.0080
	40	38.4550	0.9391	0.0122	39.2490	0.9491	0.0092
	50	36.8715	0.9253	0.0254	38.4970	0.9420	0.0119
Garden	30	41.0244	0.9804	0.0115	41.2420	0.9879	0.0091
	40	40.5242	0.9781	0.0135	40.8009	0.9871	0.0099
	50	39.2265	0.9729	0.0214	40.0002	0.9850	0.0119
Stairs	30	41.0574	0.9826	0.0124	41.4211	0.9898	0.0084
	40	40.1333	0.9801	0.0135	40.9604	0.9892	0.0096
	50	38.6847	0.9709	0.0273	40.0652	0.9868	0.0114
Corridor	30	40.3201	0.9813	0.0128	40.7789	0.9904	0.0092
	40	39.4102	0.9775	0.0156	40.4021	0.9897	0.0105
	50	38.2169	0.9719	0.0300	39.6962	0.9872	0.0127
Car	30	39.5951	0.9456	0.0129	39.9887	0.9502	0.0088
	40	38.6298	0.9397	0.0183	39.7841	0.9486	0.0103
	50	36.7113	0.9101	0.0229	38.5019	0.9409	0.0163
Buildings	30	39.3587	0.9547	0.0171	39.8475	0.9606	0.0120
	40	38.6073	0.9487	0.0190	39.2298	0.9589	0.0145
	50	37.3181	0.9249	0.0345	38.6869	0.9531	0.0160
Zebra crossing	30	39.9135	0.9868	0.0112	40.1558	0.9906	0.0099
	40	38.9183	0.9843	0.0137	39.7001	0.9900	0.0111
	50	37.7372	0.9755	0.0264	38.7267	0.9879	0.0133



Figure 2. The passerby image with different degrees of degradation. (a) The original image, (b) the 7×7 Gaussian blur, and (c) the 7×7 Gaussian blur + 30% salt-and-pepper noise.

A comparison of Figure 2a,c with Figure 3b,c revealed that the image deblurred by Lp_LOGS had more speckle noise than the one deblurred by Lp_HOGS. The images in the red boxes in Figure 3b,c were enlarged to further visualize the difference between the two and the results are displayed in Figure 3e,f, respectively.

The right half was a largely smooth area in Figure 3e,f. Figure 3f had less salt-and-pepper noise compared with Figure 3e and its visual effect was closer to that of the original image. The left half was mostly the edge area, therefore it was difficult to see the denoising

effect. However, Figure 3f retained more details. For example, the lines of the iron fence are clearer and more continuous. Overall, the visual effect of Figure 3f was closer to that of the original image, indicating that the deblurring performance of Lp_HOGS under the Gaussian blur was better than that of Lp_LOGS. Lp_HOGS also preserved more details while suppressing the staircase effect of deblurred images.

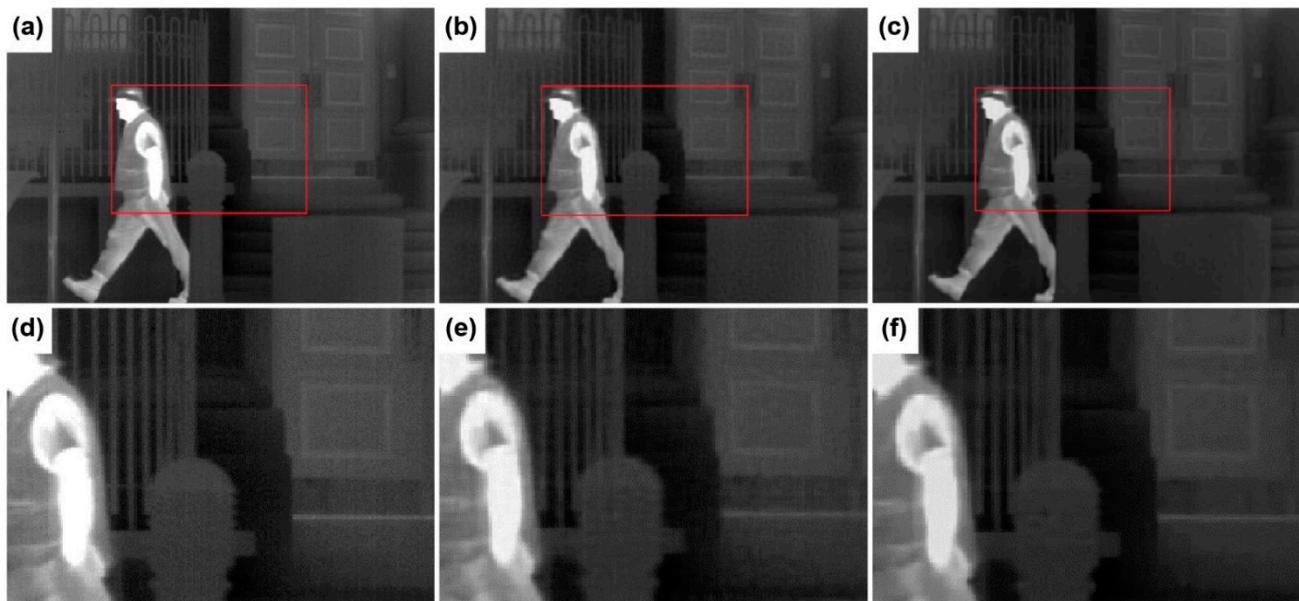


Figure 3. A comparison of the deblurring effects on a passerby image. (a) The original passerby image, (b) the image deblurred using Lp_LOGS, and (c) the image deblurred using Lp_HOGS. (d–f) The local enlarged images from the red boxes in (a–c), respectively.

3.1.2. The Box Blur

We added 30%, 40%, and 50% salt-and-pepper noise to the test images with a 7×7 box blur. The experimental results are listed in Table 2.

The three performance indicators obtained by the Lp_HOGS model were higher than those obtained by the Lp_LOGS model. The Lp_HOGS model achieved average PSNR values that were 0.362, 0.805, and 1.356 dB higher than those of the Lp_LOGS model when the salt-and-pepper noise was 30%, 40%, and 50%, respectively. In terms of the difference in the PSNR value, Lp_HOGS had a marginally greater advantage over Lp_LOGS in the 7×7 box blur than in the 7×7 Gaussian blur.

A station image with different degrees of degradation is shown in Figure 4. The images that were deblurred by Lp_LOGS and Lp_HOGS are compared in Figure 5.

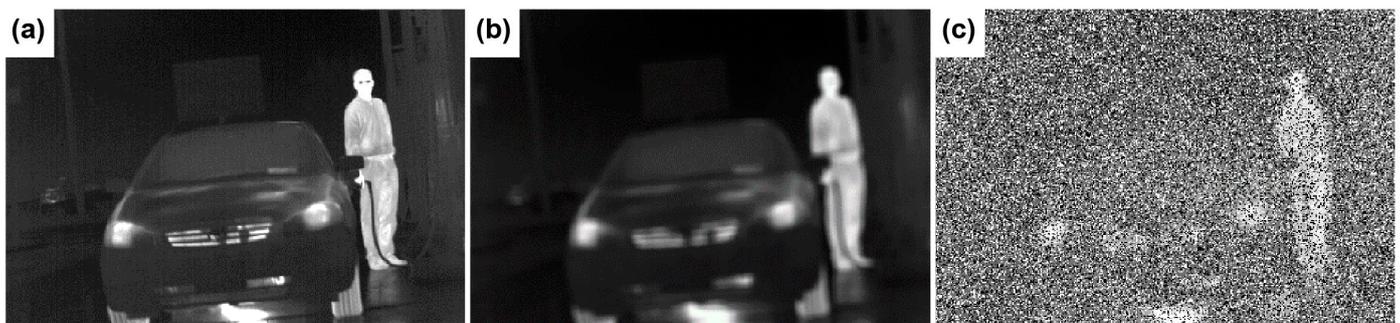


Figure 4. The station image with different degrees of degradation. (a) The original image, (b) the 7×7 box blur, and (c) the 7×7 box blur + 40% salt-and-pepper noise.

Table 2. The Lp_HOGS and Lp_LOGS deblurring effects on images with a 7×7 box blur. The optimal indicators for each condition are denoted in bold.

Image	Noise Level (%)	Lp_LOGS			Lp_HOGS		
		PSNR (dB)	SSIM	GMSD	PSNR (dB)	SSIM	GMSD
Passerby	30	40.8217	0.9457	0.0180	41.0760	0.9535	0.0124
	40	40.1691	0.9402	0.0194	40.7809	0.9518	0.0144
	50	38.7155	0.9303	0.0281	40.2154	0.9497	0.0159
Station	30	39.8743	0.9474	0.0094	40.1122	0.9532	0.0080
	40	39.1676	0.9415	0.0114	39.7945	0.9521	0.0085
	50	37.9245	0.9348	0.0268	39.1034	0.9487	0.0147
Truck	30	39.9195	0.9480	0.0091	40.1125	0.9534	0.0070
	40	39.1053	0.9415	0.0108	39.7635	0.9520	0.0080
	50	37.4036	0.9304	0.0252	38.7291	0.9481	0.0108
Garden	30	41.1108	0.9821	0.0107	41.3335	0.9872	0.0100
	40	40.3230	0.9766	0.0167	40.8003	0.9863	0.0114
	50	39.0822	0.9658	0.0279	39.8915	0.9853	0.0126
Stairs	30	41.0179	0.9832	0.0126	41.5490	0.9895	0.0102
	40	40.0870	0.9786	0.0173	41.1270	0.9887	0.0117
	50	38.6422	0.9701	0.0255	40.0578	0.9872	0.0133
Corridor	30	40.4221	0.9818	0.0162	41.0429	0.9905	0.0093
	40	39.3766	0.9758	0.0247	40.5461	0.9897	0.0106
	50	37.7766	0.9578	0.0424	39.4056	0.9882	0.0123
Car	30	39.6822	0.9454	0.0106	40.0593	0.9528	0.0090
	40	38.8954	0.9362	0.0136	39.8088	0.9511	0.0106
	50	37.5309	0.9199	0.0340	38.9581	0.9466	0.0136
Buildings	30	39.6155	0.9534	0.0129	40.2602	0.9642	0.0099
	40	38.9742	0.9496	0.0154	39.7577	0.9618	0.0114
	50	37.5042	0.9338	0.0258	39.3106	0.9594	0.0125
Zebra crossing	30	40.0891	0.9851	0.0127	40.2653	0.9904	0.0104
	40	38.9207	0.9816	0.0209	39.8872	0.9896	0.0118
	50	37.6985	0.9721	0.0308	38.8089	0.9884	0.0136

A comparison of Figure 4a,c with Figure 5b,c revealed satisfactory overall deblurring effects of Lp_LOGS and Lp_HOGS on images with the box blur. However, the image deblurred by Lp_LOGS contained more speckle noise than the image deblurred by Lp_HOGS. The images in the red boxes in Figure 5b,c are enlarged in Figure 5e,f, respectively.

The upper halves of Figure 5e,f were mostly smooth areas. The model used to create Figure 5f removed salt-and-pepper noise more thoroughly than that used on Figure 5e and its visual effect was closer to the original image. The lower half mostly contained the edge area. Thus, the denoising effect was not evident. However, Figure 5f retained more detail from the original image. For example, the car logo is clearer. Overall, the visual effect of Figure 5f was closer to that of the original image, indicating that Lp_HOGS exhibited better deblurring performance than Lp_LOGS under the box blur. It was also reconfirmed that Lp_HOGS preserved more detail while suppressing the staircase effect of the deblurred images.

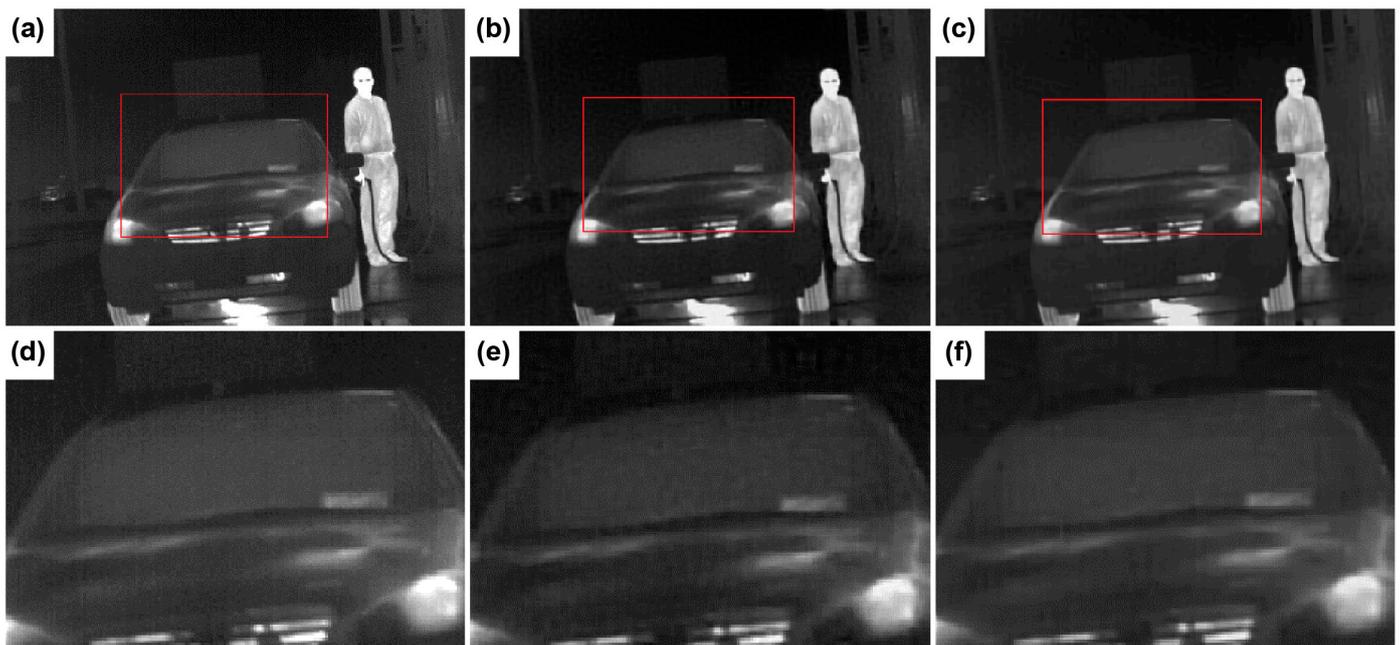


Figure 5. A comparison of deblurring effects of the two models on a station image. (a) The original station image, (b) the image deblurred using Lp_LOGS, and (c) the image deblurred using Lp_HOGS. (d–f) The Local enlarged images from the red boxes in (a–c), respectively.

3.1.3. The Motion Blur

Finally, 30%, 40%, and 50% salt-and-pepper noise were added to the test images with a 7×7 motion blur. The experimental results are listed in Table 3.

The three performance indicators obtained by the Lp_HOGS model were higher than those obtained by the Lp_LOGS model. The Lp_HOGS model achieved average PSNR values that were 1.387, 1.774, and 2.372 dB higher than those of the Lp_LOGS model when the salt-and-pepper noise was 30%, 40%, and 50%, respectively. The difference in the PSNR values showed that Lp_HOGS had more obvious advantages over Lp_LOGS in the motion blur than in the Gaussian and box blurs.

The images of a truck with different degrees of degradation are shown in Figure 6. The deblurring effects of Lp_LOGS and Lp_HOGS are compared in Figure 7.



Figure 6. The truck image with different degrees of degradation. (a) The original image, (b) the 7×7 motion blur, and (c) the 7×7 motion blur + 50% salt-and-pepper noise.

A comparison of Figure 6a,c with Figure 7b,c revealed that Lp_LOGS and Lp_HOGS achieved satisfactory deblurring effects in terms of motion blur. Furthermore, the performance edge of the proposed model was similar to that under the Gaussian and the box blurs. The deblurring effects are depicted in Figure 7b,c,e,f.

Table 3. The Lp_HOGS and Lp_LOGS deblurring effects on images with the 7×7 motion blur. The optimal indicators under each condition are denoted in bold.

Image	Noise Level (%)	Lp_LOGS			Lp_HOGS		
		PSNR (dB)	SSIM	GMSD	PSNR (dB)	SSIM	GMSD
Passerby	30	41.4154	0.9473	0.0224	42.7863	0.9643	0.0109
	40	40.0856	0.9324	0.0384	41.4734	0.9594	0.0176
	50	38.0234	0.9293	0.0487	40.0456	0.9535	0.0311
Station	30	41.0730	0.9534	0.0134	42.2171	0.9660	0.0048
	40	39.3011	0.9415	0.0253	40.7997	0.9629	0.0088
	50	36.5096	0.9263	0.0349	38.5807	0.9573	0.0139
Truck	30	41.1586	0.9441	0.0132	42.1004	0.9665	0.0039
	40	39.1459	0.9408	0.0178	41.0177	0.9639	0.0058
	50	36.0788	0.9372	0.0291	39.1823	0.9590	0.0117
Garden	30	43.1018	0.9877	0.0116	44.4994	0.9929	0.0063
	40	41.6052	0.9857	0.0144	43.3407	0.9913	0.0096
	50	38.3654	0.9733	0.0356	40.5691	0.9876	0.0269
Stairs	30	43.2679	0.9900	0.0133	44.7027	0.9947	0.0102
	40	41.3666	0.9861	0.0221	42.9469	0.9939	0.0119
	50	38.2970	0.9779	0.0330	40.0135	0.9884	0.0343
Corridor	30	41.8418	0.9888	0.0138	43.5307	0.9939	0.0092
	40	40.0657	0.9841	0.0281	41.8881	0.9920	0.0132
	50	37.3735	0.9703	0.0404	39.4419	0.9854	0.0368
Car	30	41.1081	0.9562	0.0084	42.2795	0.9664	0.0048
	40	38.9419	0.9439	0.0155	40.9768	0.9635	0.0074
	50	35.9764	0.9350	0.0406	38.7575	0.9583	0.0157
Buildings	30	39.8062	0.9577	0.0203	41.3484	0.9716	0.0080
	40	37.7007	0.9430	0.0359	39.9433	0.9673	0.0137
	50	34.1941	0.9151	0.0686	37.6026	0.9592	0.0243
Zebra crossing	30	41.8608	0.9909	0.0110	43.6546	0.9952	0.0061
	40	40.2476	0.9880	0.0167	42.0393	0.9934	0.0135
	50	37.2561	0.9760	0.0414	39.2335	0.9891	0.0300

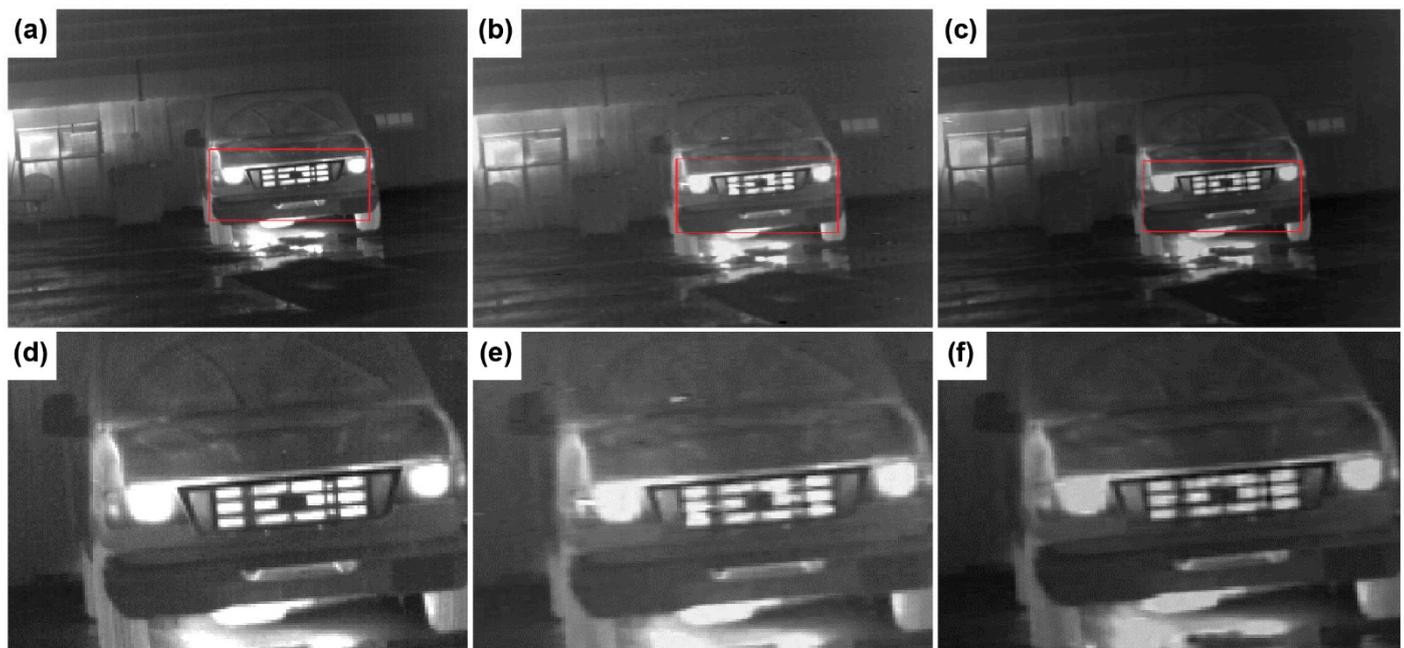


Figure 7. A comparison of deblurring effects of the two models on a truck image. (a) The original truck image, (b) the image deblurred using Lp_LOGS, and (c) the image deblurred using Lp_HOGS. (d–f) The local enlarged images of the red boxes from (a–c), respectively.

3.2. The Comparison with Other Methods

This section compares the proposed model with existing models, including ATV [4], ITV [5], L0TVPADMM [9], and HNHOTV-OGS [10]. Nine distinct IR images were selected as test images. The truck, the buildings, the car, and the figure images were 384×288 pixels, and the garden, the stairs, the corridor, the road, and the zebra crossing images were 506×408 pixels. The experimental results are listed in Table 4 and the unit of PSNR as dB.

Table 4. The deblurring effects of Lp_HOGS and the other methods on the images with a 7×7 Gaussian blur. The optimal indicators under each condition are denoted in bold.

Image	Method	Noise Level								
		30%			40%			50%		
		PSNR	SSIM	GMSD	PSNR	SSIM	GMSD	PSNR	SSIM	GMSD
Truck	ITV	30.0291	0.8884	0.0616	29.5562	0.8893	0.0651	29.1880	0.8837	0.0691
	ATV	35.3620	0.9228	0.0331	33.4723	0.9087	0.0412	31.4261	0.8976	0.0529
	HNHOTV-OGS	36.9788	0.9246	0.0280	34.6626	0.9037	0.0365	32.1012	0.8763	0.0490
	L0TVPADMM	37.1920	0.9033	0.0271	36.5800	0.8964	0.0291	35.5870	0.8682	0.0326
	Lp_HOGS	39.8380	0.9502	0.0080	39.2490	0.9491	0.0092	38.4970	0.9420	0.0119
Garden	ITV	36.4479	0.9357	0.0379	36.0066	0.9318	0.0400	34.5629	0.9233	0.0456
	ATV	37.1587	0.9376	0.0352	36.6090	0.9321	0.0385	35.2160	0.9221	0.0432
	HNHOTV-OGS	38.8667	0.9429	0.0290	37.7810	0.9328	0.0329	36.5325	0.9163	0.0380
	L0TVPADMM	39.2100	0.9290	0.0274	39.0490	0.9347	0.0287	38.7830	0.9260	0.0293
	Lp_HOGS	41.2420	0.9879	0.0091	40.8009	0.9871	0.0099	40.0002	0.9850	0.0119
Stairs	ITV	36.1477	0.9353	0.0480	35.5521	0.9303	0.0517	34.4880	0.9225	0.0562
	ATV	37.2454	0.9394	0.0426	36.3834	0.9312	0.0469	35.4070	0.9222	0.0517
	HNHOTV-OGS	38.2716	0.9351	0.0375	37.1927	0.9226	0.0425	35.8911	0.9027	0.0494
	L0TVPADMM	38.8690	0.9273	0.0350	38.6270	0.9251	0.0360	38.3940	0.9227	0.0370
	Lp_HOGS	41.4211	0.9898	0.0084	40.9604	0.9892	0.0096	40.0652	0.9872	0.0127
Corridor	ITV	35.1721	0.9273	0.0544	34.5450	0.9227	0.0581	33.8964	0.9122	0.0676
	ATV	35.7739	0.9299	0.0519	34.8115	0.9202	0.0565	33.9243	0.9078	0.0628
	HNHOTV-OGS	37.4461	0.9281	0.0420	36.4630	0.9151	0.0470	34.9757	0.8995	0.0558
	L0TVPADMM	38.5150	0.9404	0.0373	38.4240	0.9395	0.0375	38.0550	0.9343	0.0391
	Lp_HOGS	40.7789	0.9904	0.0092	40.4021	0.9897	0.0105	39.6962	0.9872	0.0127
Road	ITV	34.1196	0.9351	0.0443	33.5094	0.9300	0.0472	32.6307	0.9202	0.0520
	ATV	34.7909	0.9377	0.0411	33.8026	0.9289	0.0456	32.7069	0.9155	0.0515
	HNHOTV-OGS	37.3621	0.9451	0.0302	36.3655	0.9355	0.0338	34.8160	0.9167	0.0404
	L0TVPADMM	38.5640	0.9462	0.0262	38.0270	0.9378	0.0276	37.7320	0.9401	0.0289
	Lp_HOGS	40.6179	0.9926	0.0077	40.1633	0.9921	0.0088	39.0861	0.9901	0.0119
Buildings	ITV	30.1056	0.8555	0.0613	29.7495	0.8454	0.0645	29.2963	0.8365	0.0676
	ATV	34.8630	0.9071	0.0374	33.1033	0.8860	0.0443	30.4774	0.8579	0.0585
	HNHOTV-OGS	36.8107	0.9336	0.0286	34.9021	0.9140	0.0356	32.4871	0.8845	0.0470
	L0TVPADMM	37.0520	0.9258	0.0276	36.1780	0.9121	0.0305	35.5460	0.8971	0.0328
	Lp_HOGS	39.8475	0.9606	0.0120	39.2298	0.9589	0.0145	38.6869	0.9531	0.0160
Car	ITV	30.9921	0.8883	0.0555	30.6337	0.8809	0.0578	30.2899	0.8763	0.0602
	ATV	36.0531	0.9175	0.0310	34.6872	0.9013	0.0363	32.1603	0.8813	0.0485
	HNHOTV-OGS	37.1645	0.9217	0.0273	35.1484	0.8987	0.0344	32.7652	0.8685	0.0453
	L0TVPADMM	37.6620	0.9099	0.0255	37.0140	0.8986	0.0275	36.0630	0.8790	0.0307
	Lp_HOGS	39.9887	0.9502	0.0088	39.7841	0.9486	0.0103	38.5019	0.9409	0.0163
Figure	ITV	29.1305	0.8744	0.0861	28.4494	0.8759	0.0931	27.5542	0.8740	0.1032
	ATV	35.3557	0.9040	0.0420	33.2249	0.8911	0.0537	29.7096	0.8736	0.0806
	HNHOTV-OGS	35.2386	0.8793	0.0426	32.6695	0.8424	0.0573	30.0911	0.7974	0.0771
	L0TVPADMM	37.3870	0.8962	0.0318	36.7670	0.8834	0.0357	36.2720	0.8739	0.0378
	Lp_HOGS	40.9767	0.9431	0.0094	39.7917	0.9420	0.0147	38.7635	0.9378	0.0190
ZebraCrossing	ITV	33.9752	0.9193	0.0454	33.4185	0.9134	0.0484	32.4327	0.9023	0.0542
	ATV	34.8510	0.9267	0.0410	33.8304	0.9170	0.0462	32.9694	0.9032	0.0510
	HNHOTV-OGS	37.3081	0.9374	0.0309	36.1585	0.9271	0.0353	34.5367	0.9018	0.0426
	L0TVPADMM	38.3140	0.9391	0.0324	38.0470	0.9377	0.0284	37.1780	0.9284	0.0360
	Lp_HOGS	40.1558	0.9906	0.0099	39.7001	0.9900	0.0111	38.7267	0.9879	0.0133

ITV performed the worst, whereas Lp_HOGS outperformed the other four methods in terms of PSNR, SSIM, and GMSD. The PSNR of Lp_HOGS under 30%, 40%, and 50% salt-and-pepper noise was at least 1.2 dB higher than that of L0TVPADMM, which had the second-best PSNR. Therefore, the proposed Lp_HOGS model achieved better performance for removing salt-and-pepper noise than the other state-of-the-art methods. In addition, the

IR image after deblurring obtained a better visual effect, which is conducive to subsequent image analysis and processing.

4. Conclusions

This study proposed an image-deblurring model based on the LOGS model to mine the high-order prior information of an IR image containing salt-and-pepper noise. The LOGS regularization term was investigated, combining the advantage of the Lp-pseudo-norm in describing salt-and-pepper noise with replacing the low-order term with a high-order term. The proposed IR image-deblurring model (Lp_HOGS) successfully deblurred an IR image under salt-and-pepper noise. Lp_HOGS achieved average PSNR values that were 0.304, 0.784, and 1.287 dB higher for salt-and-pepper noise at 30%, 40% and 50%, respectively, than those of the Lp_LOGS model for a Gaussian blur. Similarly, Lp_HOGS was 0.362, 0.805, and 1.356 dB higher for the box blur and 1.387, 1.774, and 2.372 dB higher for the motion blur. The findings of this study resulted in the following conclusions:

1. Upon adding the Gaussian blur and different levels of salt-and-pepper noise to a given test image, the Lp_HOGS model exhibited a better deblurring effect than the existing models. This result implies that the Lp-pseudo-norm had a stronger sparse representation ability and the overlapping group sparsity regularization term increased the difference between the smooth and the edge areas of an image.
2. Upon adding the different types of blur and levels of salt-and-pepper noise to a given test image, Lp_HOGS yielded stronger indicators than Lp_LOGS. This advantage became greater as the noise level increased. Therefore, the high-order prior information of the image improved the quality of the deblurred IR images and the stability of salt-and-pepper noise removal.
3. The advantage of Lp_HOGS over Lp_LOGS was most obvious in terms of the motion blur, indicating that adding the prior constraints of the high-order gradient to the model could significantly improve the IR image deblurring effect amid the motion blur.

A limitation of this approach is that the application of the Lp_HOGS model is time-consuming. In our future work, we will accelerate the process by introducing accelerated ADMM to improve the performance and efficiency of the proposed method. Nevertheless, the proposed Lp_HOGS model provides a new approach to reduce the salt-and-pepper noise impact on IR images.

Author Contributions: Conceptualization, Z.Y. and Y.C.; Methodology, Z.Y.; Software, Z.Y. and J.H.; Validation, Z.Y. and X.O.; Formal analysis, Z.Y.; Investigation, Z.Y. and J.H.; Resources, Y.C.; Data curation, Z.Y. and X.O.; Writing—original draft preparation, Z.Y. and X.O.; Writing—review and editing, Z.Y. and Y.C.; Visualization, Z.Y.; Supervision, Y.C.; Project administration, Y.C.; Funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation Project of Fujian Province, grant numbers 2020J05169 and 2020J01816; Principal Foundation of Minnan Normal University, grant number KJ19019; and High-Level Science Research Project of Minnan Normal University, grant number GJ19019.

Data Availability Statement: The data presented in this study are openly available at <http://adas.cvc.uab.es/elektra/datasets/far-infra-red/> (accessed on 5 January 2022), <http://www.dgp.toronto.edu/~\{n\}nmorris/data/IRData/> (accessed on 5 January 2022), and <http://adas.cvc.uab.es/elektra/datasets/far-infra-red/> (accessed on 5 January 2022). Additionally, the perfect code data for the proposed method in this paper are available upon request from the corresponding author.

Acknowledgments: Thank you to X.L. of the University of Electronic Science and Technology of China for sharing the FTV4Lp code.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Li, J.; Yang, W.; Zhang, X. *Infrared Image Processing, Analysis, and Fusion*; Science Press: Beijing, China, 2009; pp. 7–12. Available online: <https://max.book118.com/html/2019/0105/5210100330001344.shtml> (accessed on 15 January 2022).
2. Fan, Z.; Bi, D.; He, L.; Ma, S. Noise suppression and details enhancement for infrared image via novel prior. *Infrared Phys. Technol.* **2016**, *74*, 44–52. [[CrossRef](#)]
3. Faragallah, O.S.; Ibrahim, H.M. Adaptive switching weighted median filter framework for suppressing salt-and-pepper noise. *Int. J. Electron. Commun.* **2016**, *70*, 1034–1040. [[CrossRef](#)]
4. Afraites, L.; Hadri, A.; Laghrib, A. A denoising model adapted for impulse and Gaussian noises using a constrained-PDE. *Inverse Probl.* **2020**, *36*, 025006. [[CrossRef](#)]
5. Lim, H.; Williams, T.N. A Non-standard Anisotropic Diffusion for Speckle Noise Removal. *Syst. Cybern. Inform.* **2007**, *5*, 12–17. Available online: <http://www.iiisci.org/Journal/pdv/sci/pdfs/P610066.pdf> (accessed on 16 August 2022).
6. Osher, S.; Burger, M.; Goldfarb, D.; Jinjun, X.; Wotao, Y. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.* **2005**, *4*, 460–489. [[CrossRef](#)]
7. Peng, Z.; Chen, Y.; Pu, T.; Wang, Y.; He, Y. Image denoising based on sparse representation and regularization constraint: A Review. *J. Data Acquis. Process.* **2018**, *33*, 1–11. Available online: <http://www.cnki.com.cn/Article/CJFDTotal-SJCJ201801001.htm> (accessed on 16 December 2021).
8. Li, S.; He, Y.; Chen, Y.; Liu, W.; Yang, X.; Peng, Z. Fast multi-trace impedance inversion using anisotropic total p-variation regularization in the frequency domain. *J. Geophys. Eng.* **2018**, *15*, 2171–2182. [[CrossRef](#)]
9. Wu, H.; He, Y.; Chen, Y.; Shu, L.; Peng, Z. Seismic acoustic impedance inversion using mixed second-order fractional ATPV regularization. *IEEE Access* **2020**, *8*, 3442–3452. [[CrossRef](#)]
10. Liu, G.; Huang, T.Z.; Liu, J.; Lv, X.G. Total variation with overlapping group sparsity for image deblurring under impulse noise. *PLoS ONE* **2015**, *10*, e0122562. [[CrossRef](#)]
11. Yuan, G.; Ghanem, B. l0tv: A new method for image restoration in the presence of impulse noise. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5369–5377.
12. Adam, T.; Paramesran, R. Image denoising using combined higher order non-convex total variation with overlapping group sparsity. *J. Multidimens. Syst. Signal Process.* **2019**, *30*, 503–527. [[CrossRef](#)]
13. Chartrand, R. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.* **2007**, *14*, 707–710. [[CrossRef](#)]
14. Chartrand, R. Shrinkage mappings and their induced penalty functions. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1026–1029. [[CrossRef](#)]
15. Chartrand, R.; Staneva, V. Restricted isometry properties and nonconvex compressive sensing. *Inverse Probl.* **2008**, *24*, 035020. [[CrossRef](#)]
16. Wu, R.; Chen, D.R. The improved bounds of restricted isometry constant for recovery via l(p)-minimization. *IEEE Trans. Inf. Theory* **2013**, *59*, 6142–6147. [[CrossRef](#)]
17. Wen, J.; Li, D.; Zhu, F. Stable recovery of sparse signals via lp-minimization. *Appl. Comput. Harmon. Anal.* **2015**, *38*, 161–176. [[CrossRef](#)]
18. Chen, Y.; Peng, Z.; Gholami, A.; Yan, J.; Li, S. Seismic signal sparse time-frequency representation by Lp-quasinorm constraint. *Digital Signal Process.* **2019**, *87*, 43–59. [[CrossRef](#)]
19. Liu, X.; Chen, Y.; Peng, Z.; Wu, J. Total variation with overlapping group sparsity and Lp quasinorm for infrared image deblurring under salt-and-pepper noise. *J. Electron. Imaging* **2019**, *28*, 043031. [[CrossRef](#)]
20. Liu, X.; Chen, Y.; Peng, Z.; Wu, J.; Wang, Z. Infrared image super-resolution reconstruction based on quaternion fractional order total variation with Lp quasinorm. *Appl. Sci.* **2018**, *8*, 1864. [[CrossRef](#)]
21. Xu, J.; Chen, Y. Method of removing salt and pepper noise based on total variation technique and Lp pseudo-norm. *J. Data Acquis. Process.* **2020**, *35*, 89–99. Available online: <https://qikan.cqvip.com/Qikan/Article/Detail?id=7100955546> (accessed on 16 August 2022).
22. Lin, F.; Chen, Y.; Chen, Y.; Yu, F. Image deblurring under impulse noise via total generalized variation and non-convex shrinkage. *Algorithms* **2019**, *12*, 221. [[CrossRef](#)]
23. Wang, L.; Chen, Y.; Lin, F.; Chen, Y.; Yu, F.; Cai, Z. Impulse noise denoising using total variation with overlapping group sparsity and Lp-pseudo-norm shrinkage. *Appl. Sci.* **2018**, *8*, 2317. [[CrossRef](#)]