Optimal Graph Edge Weights Driven NLMs with Multi-layer Residual Compensation

Fang Yang (yangfang.idif@wust.edu.cn)
Wuhan University of Science and Technology
Xin Chen
Wuhan University of Science and Technology https://orcid.org/0000-0002-7777-8182
Li Chai
Wuhan University of Science and Technology School of Medicine

Research

Keywords: Non Local Means, Graph Laplacian Regularization, Image Denoising, Graph Signal Processing, Residual Compensation

Posted Date: March 26th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-323646/v1

Noname manuscript No.
(will be inserted by the editor)

Optimal Graph Edge Weights Driven NLMs with Multi-layer Residual Compensation

Fang Yang , Xin Chen and Li Chai

the date of receipt and acceptance should be inserted later

Abstract Non-Local Means (NLMs) play important roles in image denoising, restoration, inpainting *etc.* due to its simple theory but effective performance. In this paper, in order to better remove noise without loss of image details, we further develop the NLMs based denoising method. It is realized by introducing a graph Laplacian regularization based weighting model and a multi-layer residual compensation strategy. Unlike the patch similarity based weights in classic NLMs filters, the graph Laplacian regularization defines the weights by considering 1) the distance between target pixel and the candidate pixel, 2) the local gradient and 3) the patch similarity. The proposed NLMs filter performs in a multi-layer framework to better remove the noise and smooth the result. The corresponding residuals at each layer are finally combined with the smooth image with learned weights to recover the image details. Experimental results show that our method is effective and robust, especially for piecewise-smooth images.

Keywords Non Local Means, Graph Laplacian Regularization, Image Denoising, Graph Signal Processing, Residual Compensation.

Fang Yang
Engineering Research Center of Metallurgical Automation and Measurement Technology, Wuhan University of Science and Technology,Wuhan, Hubei,430081, China E-mail: yangfang.idif@wust.edu.cn

Xin Chen
Engineering Research Center of Metallurgical Automation and Measurement Technology, Wuhan University of Science and Technology,Wuhan, Hubei,430081, China E-mail: xinchen0112@foxmail.com

Li Chai
Engineering Research Center of Metallurgical Automation and Measurement Technology, Wuhan University of Science and Technology,Wuhan, Hubei,430081, China E-mail: chaili@wust.edu.cn

1 Introduction

Image denoising is one of the most fundamental and important tasks in image processing and computer vision. Generally speaking, it aims at retrieving the original signal $\mathbf{u}_0$ given an observed signal $\mathbf{u} = \mathbf{u}_0 + \mathrm{n}$ with the additive Gaussian white noise n. In the past decades, numerous denoising methods have been proposed. In terms of the way to separate $\mathbf{u}_0$ from $\mathbf{u}$, the denoising methods can be implemented either in the spatial domain or in the transform domain.

In the spatial domain, the state-of-the-art methods average the neighbor pixels with different weights, *e.g.* equal weights of box-car filter, weights depending on the distance between pixels in Gaussian filter, weights computed from geometric and radiometric distances of bilateral filter [9,16,10]. Various extensions have been proposed to balance the smoothness and details, *e.g.*, averaging in local windows with adaptive size [19] or local regions with adaptive shape [28]. Contrast to these connected local regions, Buades *et al.* proposed a method to average pixels in non-local regions, which is named non local means (NLMs). The main idea of NLMs is to select similar pixels in a non-local region (even the whole image), then average them with different weights. Since the similarity between two pixels is computed by the corresponding non local patches around them, NLMs are very robust to noise and yield very effective performance. However, the NLMs filter removes some image details such as edges and rare textures during the process of denoising. Although total variation (TV) regularization models [14,5,22] have been combined with NLMs approachs to deal with rare patches (no similar pixels has selected), it still can not address this issue.

The basic idea of denoising in the transform domain is to separate noise from the observed signal in the transform domain. In general, the noise in images presents itself as rapid changes in smooth areas and it has uniform power across the whole frequency band. The noise-free signal changes slowly and its power is usually distributed only on lower frequencies. Various transforms have been used for denoising, e.g., Curvelet transform [21], Wavelet transform [4], graph Fourier transform [25,12,15]. Among all these methods, the block-match in 3D transform-domain filter (BM3D) is the most popular and widely used method [6]. BM3D filter is effective by combining the NLMs theory with the wavelet transform based denoising. Beyond transforms with fixed bases, data-driven transforms, including PCA [1], spares coding [11], dictionary learning [7] and compress sensing [8], have been widely used in image denoising tasks as well.

More recently, machine learning based denoising methods, especially deep learning based approaches, attract public attention. Deep network was first applied in image denoising in [13], in which the autoencoder network need not manually set parameters for removing the noise. Then Zhang *et al* proposed the DnCNN to deal with image denoising, super-resolution, and JPEG image deblocking [30]. The DnCNN is consisted of convolution neural network (CNN) and residual learning. A generative adversarial network (GAN)

based blind denoiser (GCBD) [2] used two phases to remove blind noise. The first phase was used to generate the ground truth, while the second phase utilized obtained ground truth into the GAN to train the denoiser. In addition, attention mechanism [26] and batch renormalization [27] theories have been introduced in denoising tasks, which achieve very good performance. In a word, the recent deep learning approaches can yield better results than the traditional filters, however, a huge amount of high-quality training data is required for the network training, which is not always available in reality.

In this paper, we are interested in developing the NLMs denoising method based on graphs. Each pixel on an image can be regarded as a node of a graph. By constructing proper links between nodes, one can interpret an image as a signal on a irregular graph. Graph signal processing (GSP) is a powerful and developed tool for the analysis of signals on graphs [20,17,3,29]. The graph Laplacian matrix describe the oscillatory behavior of graph signals, similar to the Fourier transform on regular grids. The optimal graph Laplacian regularization (OGLR) presented in [18] is applied here to remove noise. Unlike the patch similarity based weights in classic NLMs filters, the OGLR defines the weights by considering 1) the distance between target pixel and the candidate pixel, 2) the local gradient and 3) the patch similarity. The OGLR method yields good results for image denoising, however, it needs a large number of iterations achieve comparable performance. Hence in our paper, we proposed to take advantage of the optimal graph edge weights of OGLR algorithm and embed it into a multi-layer framework. The multi-layer representation is performed to better remove the noise while preserving the details. Obviously, the filtered image is smoother when the number of layer increases. For the sake of recovering image details, the residual images at each layer (the difference between the input and output of the NLMs filter) are combined with the smooth filtered image. The weights of each component, including the smooth filtered image and the residual images, are learned according to the least square method. Note that a similar idea has been proposed in [24]. However, the key difference is that our method adapts the NLMs filter parameters (the graph Laplacian regularization) to the input image, instead of a fixed filter presented in [24].

This paper is organized as follows. Section 2 introduces the related work about graph construction, the OGLR algorithm and the multi-layer representation of filter images. The proposed denoising method is detailed in section 3. Experiments and results are presented in section 4 and section 5 respectively. And finally, conclusions are given in section 6.

2 Related Work

2.1 Graph Construction

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is composed by a set $\mathcal{V}$ of nodes and a set $\mathcal{E}$ of edges. A graph can be directed and undirected, in this paper, we are focusing on the

undirected graph. An edge connecting two points (i, j) contains an edge weight $w_{i,j}$, which is typically positive. The larger value the $w_{i,j}$ is, the more similar or correlated the nodes i and j are. Generally, the weight $w_{i,j}$ between i and j is defined as:

$$w_{i,j} = \begin{cases} \exp\left(-\frac{\phi_{ij}^2}{2\varepsilon^2}\right), \text{if} \, |\phi_{ij}| \leq r, \\ 0, \qquad \text{otherwise}, \end{cases} \tag{1}$$

where ϕ_{ij} measures the distance between two pixels i and j, and r is a threshold distance. Note that ϕ_{ij} does not necessarily correspond to the Euclidean distance between the nodes. Apparently, the larger the distance between two nodes, the smaller $w_{i,j}$ is. The weighted affinity matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is then formed by the weight $w_{i,j}$ and measures the similarity between nodes, where N is the number of nodes. The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with each entry the degree (sum of each row of $\mathbf{W}$) of each node.

A graph signal $\mathbf{u}$ is often defined as a discrete signal on the nodes of the graph $\mathbf{u} : \mathcal{V} \to \mathbb{R}$. The discrete signal $\mathbf{u}$ can be simply regarded as a vector $\mathbf{u} \in \mathbb{R}^N$, where the i-th entry represents the signal value at the i-th node in $\mathcal{V}$. In terms of 2D discrete images, each pixel represents a node, and the pixel intensity stands for the signal value. The edges weights are always defined according to the locations and intensities of the nodes.

2.2 The Optimal Graph Laplacian Regularization Algorithm

The OGLR algorithm seeks for the optimal graph construction of an image by defining a set of exemplar functions [18]. For each pixel location in a 2D image, a vector $\mathbf{v}_\mathrm{i}$ of length M is constructed by using a set of exemplar functions $\{\mathrm{f_m}\}_{m=1}^{M}$:

$$\mathbf{v}_i = [\mathrm{f}_1(i)\mathrm{f}_2(i) \ldots \mathrm{f}_M(i)]. \tag{2}$$

The set of vector $\{\mathbf{v}_i\}$ is used to build the weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ with N vertices, where N is the total number of pixels. The determination of the exemplar functions is induced from a continuous graph Laplacian regularizer, described by an anisotropic Dirichlet energy functional $E(\mathbf{u})$:

$$E(\mathbf{u}) = \int \nabla \mathbf{u}^T D^{-1} \nabla \mathbf{u} (\sqrt{\det D})^{2\gamma - 1} \mathrm{ds}, \tag{3}$$

where the metric space D can be viewed as the structure tensor constructed according to the gradients $\{\nabla f_m\}_{m=1}^{M}$:

$$D = \sum_{m=1}^{M} \nabla \mathrm{f_m} \nabla \mathrm{f_m}^{\mathrm{T}} \tag{4}$$

An optimal metric space G^* can be estimated by considering the ideal metric space and the noise model from patch gradients:

$$D^* = \tilde{g}\tilde{g}^T + \beta_g I, \tag{5}$$

where $\tilde{g}$ is the average gradient of a patch, and the constant β_g is denoted by β_g is mainly determined by the covariance of the patch. With the estimated D^*, the exemplar functions can be expressed in the following form:

$$\begin{array}{l} \mathrm{f}_1^*(i) = \sqrt{\beta_g} x_i \\ \mathrm{f}_2^*(i) = \sqrt{\beta_g} y_i \\ \mathrm{f}_3^*(i) = \frac{1}{L+\sigma_g^2/\sigma_p^2} \sum_{l=0}^{L-1} z_l \end{array} , \tag{6}$$

where (x_i, y_i) are the coordinates of pixel i, and z_l is a set of L non-local patches that are similar to z_0. Here σ_p is a given constant over the whole noisy image, and σ_g is an estimated variance of the gradient of the patch. f_1 and f_2 indicate the spatial relationship between pixels, f_3 represents the average pixel intensity of a target patch. Note that the coefficient of f_1, f_2 and f_3 can balance the contributions of the spatial and intensity factors. Hence, the three exemplar functions defined in Eq.(6) can be used to construct the optimal graph edge weight.

2.3 The Multi-layer Framework

The input image can be represented in a $K-$layer tree structure, all the leaf nodes sum up to the input image. In the multi-layer scheme, the output filtered image $\mathbf{u_{out}}$ can be described by a smooth term and several detail terms:

$$\mathbf{u_{out}} = \beta_0 \mathbf{u}_{smooth} + \beta_1 \mathbf{u}_{\mathrm{detail}_1} + \ldots + \beta_K \mathbf{u}_{\mathrm{detail}_K}, \tag{7}$$

where $\{\beta_0, \beta_1, ..., \beta_\mathrm{K}\}$ is a set coefficients that controls the smoothness and the detail preservation of the output image. More details on the multi-layer scheme can be found in [23,24].

3 Methods

3.1 optimal graph weights driven NLMs in a multi-layer framework

Although OGLR algorithm has ideal filtering performance, it needs numerous iterations to achieve a comparable result. Moreover, it involves in the inverse operation during the denoising process, thus leading to a very high computational cost. Hence in this paper, we would like to take advantage of the optimal edge weights defined in OGLR algorithm and apply it into the multi-layer scheme. The multi-layer scheme can decompose input image details from fine to coarse scale, where the fine scale is used to preserve image details and the coarse scale helps to smooth the image. The proposed pipeline is shown in Fig.1.

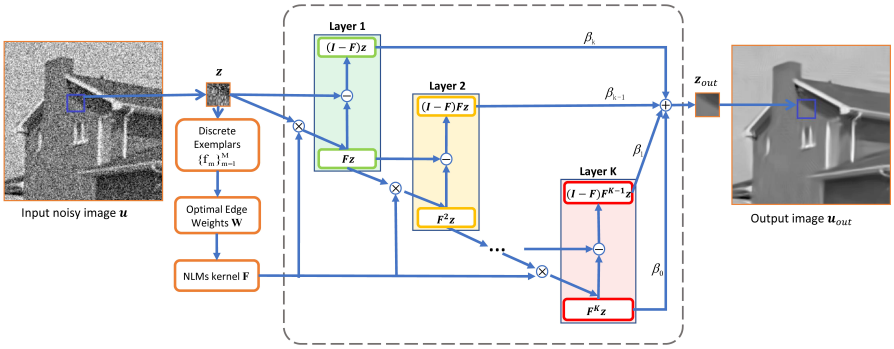


Fig. 1: Flowchart of the proposed algorithm

3.2 NLMs Kernel

The determination of the NLMs kernel is based on the graph weights. With the exemplar functions $\{\mathrm{f_m}\}_{m=1}^{M}$, the vector $\mathbf{v}_i$ on node i is as follows:

$$\mathbf{v}_i = [\sqrt{\beta_g}x_i, \sqrt{\beta_g}y_i, 1/(L+\sigma_g^2/\sigma_p^2)\sum_{l=0}^{L-1} z_l]. \tag{8}$$

Then $\mathbf{v}_i$ is used to calculate the distance ϕ in Eq.(1) between node i and j:

$$\phi_{ij} = \|\mathbf{v_i} - \mathbf{v_j}\|, \tag{9}$$

where $\|\cdot\|$ is the $\mathfrak{L}_2$ norm and ϕ_{ij} is used to determine the weighted affinity matrix $\mathbf{W}$ according to Eq.(1). The diagonal elements of the degree matrix $\mathbf{D}$ is defined as:

$$\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}. \tag{10}$$

The NLMs kernel $\mathbf{F}$ is a normalized version of the weight matrix and obtained by the product of $\mathbf{D}^{-1}$ and $\mathbf{W}$:

$$\mathbf{F} = \mathbf{D}^{-1}\mathbf{W} \tag{11}$$

The NLMs kernel $\mathbf{F}$ is similar to the graph-based bilateral filter [10] and the classical NLMs kernel. The difference lies in that the NLMs kernel considers the spatial relationship between pixels and the average intensity of patches. In addition, the relationship and the average intensity are weighted by the gradient estimates, which helps to improve to denoising performance. When the image is polluted by high-level noise, the spatial relationship between pixels dominates the denoising process (like a Gaussian filter). When the signal-noise-ratio is high, the average intensity plays a more important role.

The OGLR algorithm denoises the target patch z_0 by calculating the inverse of the Laplacian operator, *i.e.*, $\mathbf{u}^* = (\mathbf{I}+\tau\mathbf{L})^{-1}\mathbf{z}_0$. Although $\mathbf{L}$ is of

small size, the inverse operation still costs a lot of time. On the contrary, our NLMs kernel works forward and avoids the inverse operation (except for the inverse of the diagonal matrix $\mathbf{D}$, which is in fact a linear operation) as it is done in the OGLR algorithm, which makes our method much faster than the OGLR method.

3.3 Determine the Coefficients with Least Square

The set of coefficients $\{\beta_k\}$ in the multi-layer scheme plays a significant role in achieving good denoising performance. In this paper, instead of using parameters according to the s-curve functions proposed in [24], we regard the determination of $\{\beta_k\}$ as a regression problem and apply the least square algorithm to solve it. Our cost function is as follows:

$$C(\mathbf{u}) = \min_{\{\beta_0,\dots\beta_K\}} \sum_{p=1}^{P} \|(\beta_0 \mathbf{F}^K + \sum_{k=1}^{K} \beta_k (I - \mathbf{F})\mathbf{F}^{K-k})\mathbf{u}_p - \mathbf{u}_{0p}\|_2^2, \tag{12}$$

where K is the number of layers, P is the total number of training images, $\mathbf{u}_p$ represents the p-th noisy image, $\mathbf{u}_{0p}$ stands for the p-th noise-free image. The image is denoised patch by patch, so in the real training processing, the input images $\mathbf{u}_p$ and $\mathbf{u}_{0p}$ are in fact patches instead of the whole image.

Note that during the process of training, we distinguish the images with different noise level. In other words, each noise level will be assigned with a set of optimal coefficients. For each noise level, when the training process is finished, we will estimate the noise variance according to the newly-obtained $\{\beta_k\}$. If the estimated noise is higher than a given threshold σ_{th}, it is encouraged to train $\{\beta_k\}$ again with the newly-obtained $\{\beta_k\}$.

Additionally, the number of layers K is also an important parameter. Too fewer layers may lead to a incomplete representation of the image, *i.e.*, some details are not restored, or the homogeneous part of the filtered image is not smooth enough *etc.* However, too many layers would result in a large computation work, which consumes a lot of time, with only a little improvement in performance. The choice of K is discussed in the experiment section 4.2.

3.4 NLMs with K Residual Compensation

The NLMs filter can be embedded into the multi-layer scheme and the output filtered image is with one smooth term and K residuals:

$$\mathbf{u}_{out} = \beta_0 \mathbf{F}^K \mathbf{u} + \beta_1 (\mathbf{I} - \mathbf{F})\mathbf{F}^{K-1}\mathbf{u} + \beta_{K-1}(\mathbf{I} - \mathbf{F})\mathbf{F}\mathbf{u} + \beta_K (\mathbf{I} - \mathbf{F})\mathbf{u}, \tag{13}$$

where $\mathbf{F}^K\mathbf{u}$ represents the smooth term obtained after K iterations of $\mathbf{F}$. The residual K terms are the corresponding detail terms.

With the learned coefficients $\{\beta_k\}$ and the number of layers K, the proposed method is summarized in Algorithm 1. In addition, the flowchart of the proposed graph-based NLMs with multi-layer residual compensation is shown in Fig.1, where a noisy image with noise variance $\sigma = 50$ is used as an example.

Algorithm 1: Graph-based NLMs with Multi-Layer Residual Compensation

Input: Noisy image $\mathbf{u}(x, y)$, number of layers K, coefficients $\{\beta_k\}$ noise variance σ;
Output: Filtered image $\mathbf{u}_{out}$;

foreach Noisy patch z in **u do**
1. Find L similar patches of z, estimate the variance σ_g of the noisy gradient from $\{z_l\}_{l=0}^{L}$;
2: Compute the exemplar functions $\mathrm{f_m}$ and build the graph with the optimal edge weights $\mathbf{W}$ accordingly;
3: Determine the NLMs kernel $\mathbf{F}$ based on $\mathbf{W}$;
4: Denoise current patch z according to Eq.(13).
endfor
Integration of the denoised patches z to acquire $\mathbf{u}_{out}$
return $\mathbf{u}_{out}$

4 Experimentation

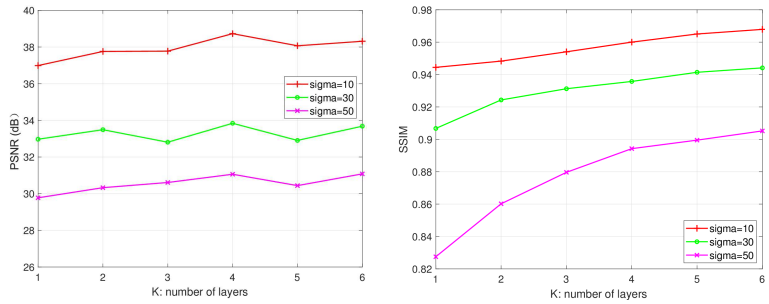
4.1 Experimental Setup

We testify the effectiveness of the proposed method both on natural images and depth images. Additive white Gaussian noise *(AWGN)* is added to these images, with standard deviations σ ranging from 10 to 50. According to different noise variances σ^2 , the patch size in our experiment is set to be from 10 to 22, the stepsize N_S ranges from 2 to 6. In the implementation, the normalization parameter γ in Eq.(3) was empirically set to be 0.6 for the natural images and $\gamma = 1$ for the depth images. We set the given constant σ_p in Eq.(6) as 10^6 and the patch cluster size L from 5 to 50. The noise variance threshold mentioned in Sec.3.3 is $\sigma_{th} = 5$.

We compare our method with the optimal graph laplacian regularizer (OGLR) [18] and two other state-of-the-art methods, *i.e.* Block-Matching 3D (BM3D) [6] and the ADNet method [26]. The peak signal noise ratio (PSNR) and the structural similarity (SSIM) are used to evaluate the performance of these methods.

4.2 Determination of Number of Layers K

To find the optimal number of layers K, we make experiments on four images, of which are two natural images and two depth images. Three levels of noise are tested separately, $\sigma = 10, 30, 50$. The average PSNR and SSIM of the four images under different noise level are computed with different K.



(a) (b)

Fig. 2: from above to bottom: (a) the average PSNR (dB) of different number of layers, (b) the average SSIM of different number of layers

In our experiments, K ranges from 1 to 6, where $K = 1$ means that the filtered image is composed of one smooth term and one residual term. The maximum of K is set to be 6 because when $K > 6$, the computation of the power of matrix costs a lot of time, which is contrary to our motivation.

Fig.2 presents the results of PSNR (a) and SSIM (b) according to K. From (a), we can see that for all three levels of noise, the average PSNR reaches the peak when $K = 4$. When K increases, the performance is not as good as when $K = 4$. In view of SSIM (b), the more the layer is, the higher the SSIM is. When $K \geq 4$, the SSIM is not improved too much. Hence, in order to balance the PSNR and SSIM, we make a compromise by setting $K = 4$ in our algorithm according to Fig.2.

4.3 Determination of Coefficients $\{\beta_k\}$

In the process of training the coefficient set $\{\beta_k\}$ in Eq.(13), six depth images and five natural images are used as the training data respectively, and each image contains 3000 to 6000 image patches under different noise level with different patch size. The value of $\{\beta_k\}$ is shown in Table.1 and 2. Both the two tables indicate that the smooth term $\mathbf{F}^K\mathbf{u}$ of Eq.(13) plays the major role. Although the coefficients of the residual terms are small values, even negative values, they also play important roles in retaining proper detailed information.

5 Results and Discussion

Fig.3 depicts the denoising performance of the four methods on two depth images (*wood* and *bowling*) and the corresponding zoom part with a noise variance $\sigma = 50$. From above to bottom are: the original image, the noisy image, the results obtained by OGLR, BM3D, ADnet and the proposed method respectively. From left to right, Column (b) and (d) are the zoom of the green box of (a) and (c) respectively.

Table 1: The learned $\{\beta_k\}$ for depth images with different levels of noise

	β_0	β_1	β_2	β_3	β_4
$\sigma < 5$	1.0223	0.0002	-0.0001	0.0001	0.0021
$\sigma = 10$	0.9992	0.0000	-0.0005	-0.0001	0.0002
$\sigma = 20$	0.9976	-0.0003	-0.0004	-0.0004	-0.0002
$\sigma = 30$	0.9766	-0.0006	-0.0007	-0.0009	-0.0001
$\sigma = 40$	0.9885	-0.0012	-0.0013	-0.0019	0.0011
$\sigma = 50$	0.9862	-0.0021	-0.0025	-0.0032	-0.0016

Table 2: The learned $\{\beta_k\}$ for real natural images with different levels of noise

	β_0	β_1	β_2	β_3	β_4
$\sigma < 5$	1.0058	0.0002	0.0003	-0.0003	0.0052
$\sigma = 10$	0.9996	0.0003	-0.0001	-0.0003	0.0069
$\sigma = 20$	0.9946	0.0001	0.0001	0.0001	0.0046
$\sigma = 30$	0.9940	0.0000	0.0001	0.0001	0.0070
$\sigma = 40$	0.9955	-0.0004	-0.0003	-0.0008	0.0153
$\sigma = 50$	0.9903	-0.0012	-0.0016	-0.0016	0.0085

For the *wood* image (a) and (b), the horizontal line in the center of the image is seriously blurred by OGLR and BM3D, moreover, the homogeneous parts are still corrupted and not well restored. ADnet can preserve edges very well visually, however, it generates some undesirable parts such as the black point in the lower-left corner and the black segment in the center. In our case, although the edges are not preserved as good as ADnet, the homogeneous parts are well smoothed. Visually speaking, our method provides a best denoising result. Our PSNR is the highest among the four methods.

For the *bowling* image (c) and (d), the PSNR and SSIM of the proposed method are superior to the other three methods. The edge between the bowling ball and pin is blurred, even distorted by BM3D. ADnet generates a deformation on the edge of the ball. The deformation may be due to that the training data does not include images with this kind of data and shape. Our method and OGLR provide better results, while our result is smoother in the homogeneous regions.

Table.3 and Table.4 illustrate the PSNR and SSIM of the proposed method and three other state-of-the-art methods on several depth images and real natural images. The highest indexes are in bold, the second-best are underlined.

From the results we can see that our proposed method is comparable with the state-of-the-art methods. In addition, it outperforms the OGLR method in nearly all cases, especially with a large noise variance σ. Furthermore, when σ is large, the performance of our method becomes more competitive. In addition, our method performs better for the piece-wise depth image compared to the performance on real natural images. This is due to that we use the multi-layer framework: the term $\mathbf{F}^K\mathbf{u}$ is obtained after K filtering, which results in a very

Table 3:
IMAGE DENOISING On Depth Images with Our Method BM3D ADNet and OGLR: peformance comparisons in PSNR (Left, in dB) and SSIM (Right)

Images	**noise**	**Our Method**	**BM3D**	**ADNet**	**OGLR**
teddy	$\sigma = 10$	38.24dB 0.9673	40.10dB 0.9818	**41.98dB0.9878**	39.98dB 0.9833
	$\sigma = 20$	34.69dB 0.9644	35.94dB 0.9674	**37.74dB0.9779**	35.94dB 0.9644
	$\sigma = 30$	33.02dB 0.9486	33.16dB 0.9481	**35.07dB0.9669**	33.49dB 0.9441
	$\sigma = 40$	30.91dB 0.9288	31.32dB 0.9279	**33.15dB0.9539**	31.78dB 0.9277
	$\sigma = 50$	30.38dB 0.9125	29.73dB 0.9190	**31.50dB0.9404**	30.46dB 0.9049
wood	$\sigma = 10$	42.50dB 0.9701	42.21dB 0.9889	43.46dB **0.9924**	**44.45dB** 0.9882
	$\sigma = 20$	**41.09dB** 0.9839	38.09dB 0.9727	39.79dB **0.9862**	40.56dB 0.9761
	$\sigma = 30$	**38.91dB** 0.9770	35.94dB 0.9573	37.82dB **0.9792**	38.15dB 0.9600
	$\sigma = 40$	**37.47dB** 0.9656	34.46dB 0.9420	36.58dB **0.9746**	36.57dB 0.9478
	$\sigma = 50$	**36.07dB** 0.9512	33.28dB 0.9375	35.41dB **0.9664**	34.55dB 0.9184
Sawtooth	$\sigma = 10$	43.40dB 0.9731	44.62dB 0.9884	**45.61dB0.9912**	45.35dB 0.9911
	$\sigma = 20$	41.11dB 0.9865	41.16dB 0.9787	**42.72dB0.9873**	40.96dB 0.9779
	$\sigma = 30$	39.56dB 0.9762	38.72dB 0.9655	**40.41dB0.9802**	38.34dB 0.9630
	$\sigma = 40$	37.29dB 0.9665	36.84dB 0.9497	**38.84dB0.9747**	36.51dB 0.9526
	$\sigma = 50$	35.84dB 0.9510	35.87dB 0.9500	**37.34dB0.9654**	34.94dB 0.9334
Tsukuba	$\sigma = 10$	40.67dB 0.9701	41.52dB 0.9798	41.41dB 0.9813	**41.64dB0.9850**
	$\sigma = 20$	37.28dB 0.9657	37.38dB 0.9577	**37.76dB0.9667**	37.56dB 0.9626
	$\sigma = 30$	35.23dB 0.9468	34.81dB 0.9319	**35.58dB0.9486**	34.91dB 0.9403
	$\sigma = 40$	33.41dB **0.9313**	33.05dB 0.9053	**34.00dB**0.9244	32.99dB 0.9245
	$\sigma = 50$	32.57dB 0.9133	31.67dB 0.8915	**32.78dB0.9157**	31.44dB 0.9003
Books	$\sigma = 10$	41.69dB 0.9827	40.89dB 0.9811	41.83dB **0.9838**	**42.37dB** 0.9828
	$\sigma = 20$	37.35dB **0.9666**	35.10dB 0.9519	37.01dB 0.9609	**38.26dB** 0.9625
	$\sigma = 30$	35.62dB **0.9491**	32.61dB 0.9235	34.45dB 0.9461	**35.88dB** 0.9420
	$\sigma = 40$	**34.28dB0.9334**	31.22dB 0.8970	32.44dB 0.9260	34.10dB 0.9256
	$\sigma = 50$	**33.31dB0.9154**	29.76dB 0.8812	31.11dB 0.9147	32.86dB 0.9050
Bowling	$\sigma = 10$	42.12dB 0.9725	41.62dB 0.9850	**43.09dB0.9899**	42.70dB 0.9867
	$\sigma = 20$	39.32dB **0.9816**	37.72dB 0.9682	**39.48dB** 0.9796	38.84dB 0.9723
	$\sigma = 30$	**38.22dB0.9732**	35.30dB 0.9484	37.46dB 0.9699	36.45dB 0.9573
	$\sigma = 40$	**36.27dB0.9654**	33.60dB 0.9379	36.03dB 0.9624	34.73dB 0.9481
	$\sigma = 50$	**35.56dB0.9531**	32.18dB 0.9193	34.37dB 0.9466	33.70dB 0.9330

smooth term nearly with no noise. The residual terms function as supplements, which helps to restore some details from the noise.

6 Conclusion

In this paper, we propose a graph-based NLMs algorithm for image denoising. The optimal edge weights defined in the OGLR algorithm is applied to

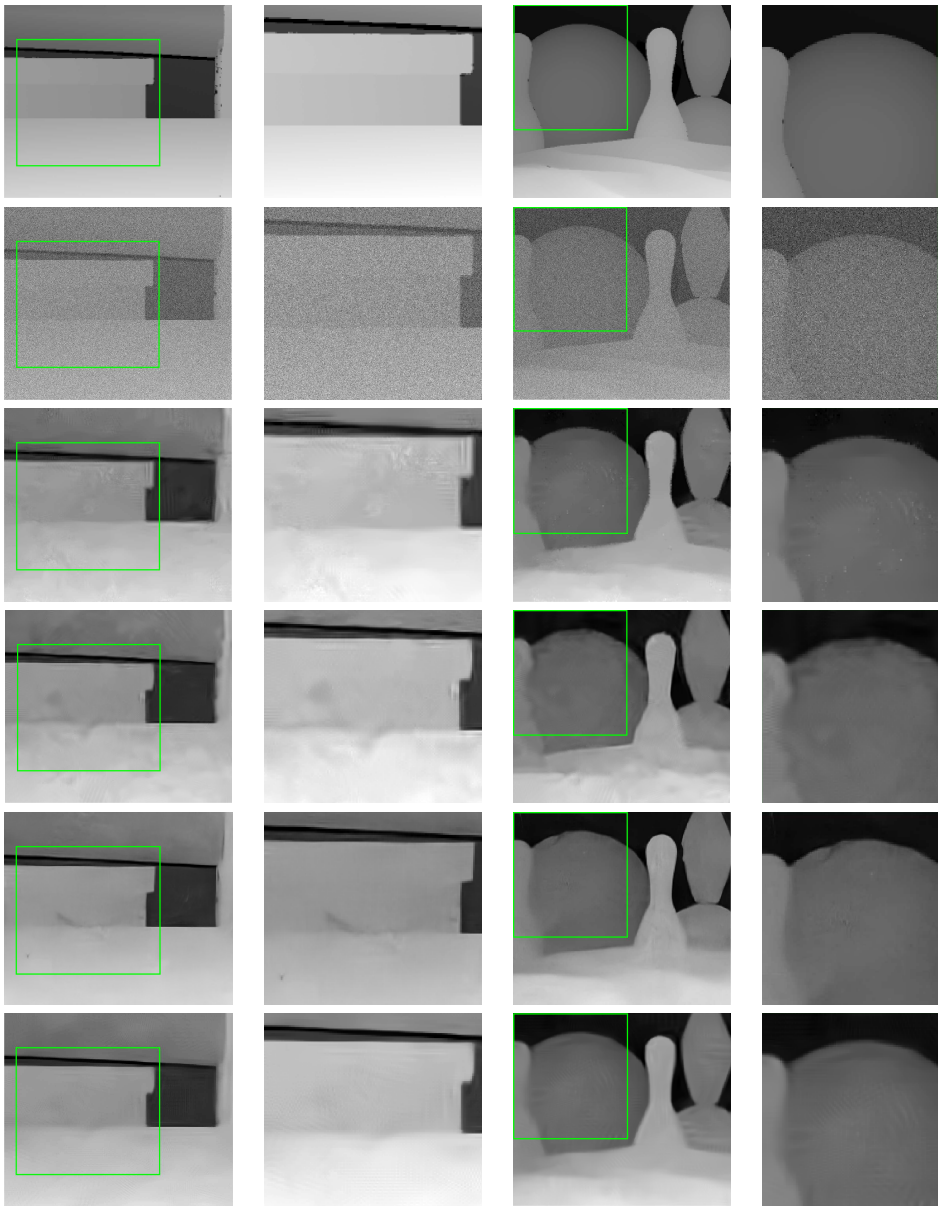


Fig. 3: Denoising results of two depth images with the noise variance $\sigma = 50$. From the above to bottom are: the original image, the noisy image, the results obtained by OGLR, BM3D, ADnet and the proposed method respectively. Column (b) and (d) are the zoom of the green box of their left (a) and (c).

Table 4:
IMAGE DENOISING On Nature Images with Our Method BM3D ADNet and OGLR: performance comparisons in PSNR (Left, in dB) and SSIM (Right)

Images	**noise**	**Our Method**	**BM3D**	**ADNet**	**OGLR**
house	$\sigma = 10$	38.50dB 0.9533	36.71dB 0.9212	36.57dB 0.9077	**38.88dB0.9622**
	$\sigma = 20$	35.68dB **0.9448**	33.77dB 0.8721	34.12dB 0.8713	**35.87dB** 0.9405
	$\sigma = 30$	33.79dB **0.9279**	32.09dB 0.8473	32.62dB 0.8556	**33.86dB** 0.9183
	$\sigma = 40$	31.33dB **0.9085**	30.65dB 0.8249	31.26dB 0.8387	**32.49dB** 0.9024
	$\sigma = 50$	**31.13dB0.8969**	29.69dB 0.8116	30.28dB 0.8230	30.67dB 0.8631
church	$\sigma = 10$	39.18dB 0.9629	39.53dB 0.9670	**40.38dB** 0.9710	39.59dB **0.9744**
	$\sigma = 20$	35.68dB 0.9564	35.99dB 0.9455	**37.23dB0.9580**	36.04dB 0.9527
	$\sigma = 30$	33.59dB 0.9378	33.92dB 0.9254	**35.06dB0.9413**	33.71dB 0.9292
	$\sigma = 40$	30.80dB 0.9101	32.42dB 0.9056	**33.64dB0.9398**	32.13dB 0.9048
	$\sigma = 50$	30.66dB 0.8945	31.33dB 0.8974	**32.37dB0.9142**	30.60dB 0.8636
flower	$\sigma = 10$	37.78dB 0.9620	38.15dB 0.9667	**38.70dB0.9712**	37.77dB 0.9655
	$\sigma = 20$	34.15dB 0.9362	34.29dB 0.9314	**35.22dB** 0.9476	34.33dB **0.9342**
	$\sigma = 30$	31.94dB 0.9049	32.19dB 0.8988	**33.17dB** 0.9225	31.95dB **0.8998**
	$\sigma = 40$	29.73dB 0.8661	30.67dB 0.8679	**31.78dB** 0.9046	30.51dB **0.8713**
	$\sigma = 50$	29.37dB 0.8544	29.72dB 0.8529	**30.44dB** 0.8813	29.09dB **0.8231**
jar	$\sigma = 10$	38.01dB 0.9333	**38.63dB0.9459**	37.52dB 0.9349	37.91dB 0.9354
	$\sigma = 20$	34.93dB 0.8884	**35.26dB0.8986**	34.58dB 0.8887	34.96dB 0.8911
	$\sigma = 30$	33.40dB 0.8615	**33.41dB0.8637**	32.96dB 0.8506	33.29dB 0.8598
	$\sigma = 40$	32.02dB **0.8403**	**32.04dB** 0.8350	32.02dB 0.8222	32.09dB 0.8375
	$\sigma = 50$	31.08dB **0.8204**	**31.17dB** 0.8191	31.13dB 0.8002	31.00dB 0.8128
bird	$\sigma = 10$	36.41dB 0.9679	38.28dB **0.9829**	**38.55dB** 0.9754	37.04dB 0.9796
	$\sigma = 20$	32.79dB 0.9635	34.30dB **0.9651**	**35.20dB** 0.9615	33.62dB 0.9621
	$\sigma = 30$	31.38dB 0.9503	31.99dB 0.9499	**33.21dB0.9506**	31.58dB 0.9421
	$\sigma = 40$	28.47dB 0.9261	30.09dB 0.9217	**31.63dB0.9377**	30.01dB 0.9201
	$\sigma = 50$	29.02dB **0.9213**	28.95dB 0.9101	**30.62dB** 0.9189	28.62dB 0.8830

construct the graph structure. A multi-layer residual compensation strategy is then used to recover the details. The coefficients of the smooth term and the residual terms of the multi-layer representation are learned according the least mean square method. We testify the effectiveness of our method both on natural images and depth images. Our proposed method outperforms the original OGLR method in PSNR/SSIM. Compared with the other state-of-the-art methods, including the OGLR, BM3D and the AD-Net method, our proposed method provides comparable results. Especially, our method has excellent denoising performance on the piece-wise depth images.

Abbreviations

NLMs:Non-Local Means;TV:Total variation; BM3D: 3D transform-domain filter; PCA:Principal component analysis; GAN:Generative adversarial network; GSP:Graph signal processing; OGLR:The optimal graph Laplacian regularization; AWGN: Additive white Gaussian noise; PSNR:The peak signal noise ratio; SSIM:the structural similarity

Acknowledgements

Not applicable.

Authors' contributions

Conceptualization, Methodology, Writing—original draft, F.Y.; Software, Validation, X.C.; Supervision, Funding acquisition, L.C. All authors approved the final, submitted version of the manuscript.

Funding

This research was funded by National Natural Science Foundation of China (61625305).

Availability of data and materials

The data that support the findings of this study are available on request from the corresponding author F.Y.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

1. Thierry Bouwmans, Sajid Javed, Hongyang Zhang, Zhouchen Lin, and Ricardo Otazo. On the applications of robust pca in image and video processing. *Proceedings of the IEEE*, 106(8):1427–1457, 2018.
2. Jingwen Chen, Jiawei Chen, Hongyang Chao, and Yang Ming. Image blind denoising with generative adversarial network based noise modeling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
3. G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng. Graph spectral image processing. *Proceedings of the IEEE*, 106(5):907–930, 2018.
4. Dongwook Cho and Tien D Bui. Multivariate statistical modeling for image denoising using wavelet transforms. *Signal Processing: Image Communication*, 20(1):77–89, 2005.
5. Camille Couprie, Leo Grady, Laurent Najman, Jean-Christophe Pesquet, and Hugues Talbot. Dual constrained tv-based regularization on graphs. *SIAM Journal on Imaging Sciences*, 6(3):1246–1273, 2013.
6. Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
7. Weisheng Dong, Xin Li, Lei Zhang, and Guangming Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *CVPR 2011*, pages 457–464. IEEE, 2011.
8. Karen Egiazarian, Alessandro Foi, and Vladimir Katkovnik. Compressed sensing image reconstruction via recursive spatially adaptive filtering. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I–549. IEEE, 2007.
9. Michael Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on image processing*, 11(10):1141–1151, 2002.
10. Akshay Gadde, Sunil K Narang, and Antonio Ortega. Bilateral filter: Graph spectral interpretation and extensions. In *2013 IEEE International Conference on Image Processing*, pages 1222–1226. IEEE, 2013.
11. Aapo Hyvärinen, Patrik Hoyer, and Erkki Oja. Image denoising by sparse code shrinkage. In *Intelligent Signal Processing*. Citeseer, 1999.
12. Amin Kheradmand and Peyman Milanfar. A general framework for regularized, similarity-based image restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, 2014.
13. Jianglin Liang and Ruifang Liu. Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In *2015 8th International Congress on Image and Signal Processing (CISP)*, 2015.
14. Cécile Louchet and Lionel Moisan. Total variation as a local filter. *SIAM Journal on Imaging Sciences*, 4(2):651–694, 2011.
15. François G Meyer and Xilin Shen. Perturbation of the eigenvectors of the graph laplacian: Application to image denoising. *Applied and Computational Harmonic Analysis*, 36(2):326–334, 2014.
16. Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128, 2012.
17. M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka. Graph signal denoising via trilateral filter on graph spectral domain. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):137–148, 2016.
18. Jiahao Pang and Gene Cheung. Graph laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Transactions on Image Processing*, 26(4):1770–1785, 2017.
19. J. M. Park, W. J. Song, and W. A. Pearlman. Speckle filtering of sar images based on adaptive windowing. *IEE Proceedings-Vision, Image and Signal Processing*, 146(4):191–0, 1999.
20. David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

21. Jean-Luc Starck, Emmanuel J Candès, and David L Donoho. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684, 2002.
22. Camille Sutour, Charles Alban Deledalle, and Jean Francois Aujol. Adaptive regularization of the nl-means: Application to image and video denoising. *IEEE Transactions on Image Processing*, 23(8):3506–3521, 2014.
23. Hossein Talebi and Peyman Milanfar. Nonlocal image editing. *Image Processing IEEE Transactions on*, 23(10):4460–4473, 2014.
24. Hossein Talebi and Peyman Milanfar. Fast multilayer laplacian enhancement. *IEEE Transactions on Computational Imaging*, 2(4):496–509, 2016.
25. Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.
26. Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided cnn for image denoising. *Neural Networks*, 124:177–129, 2020.
27. Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020.
28. G. Vasile, E. Trouve, Jong Sen Lee, and V. Buzuloiu. Intensity-driven adaptive-neighborhood technique for polarimetric and interferometric sar parameters estimation. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1609–1621, 2006.
29. Jin Zeng, Gene Cheung, Michael Ng, Jiahao Pang, and Cheng Yang. 3d point cloud denoising using graph laplacian regularization of a low dimensional manifold model. *IEEE Transactions on Image Processing*, 2019.
30. Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2016.

Figures

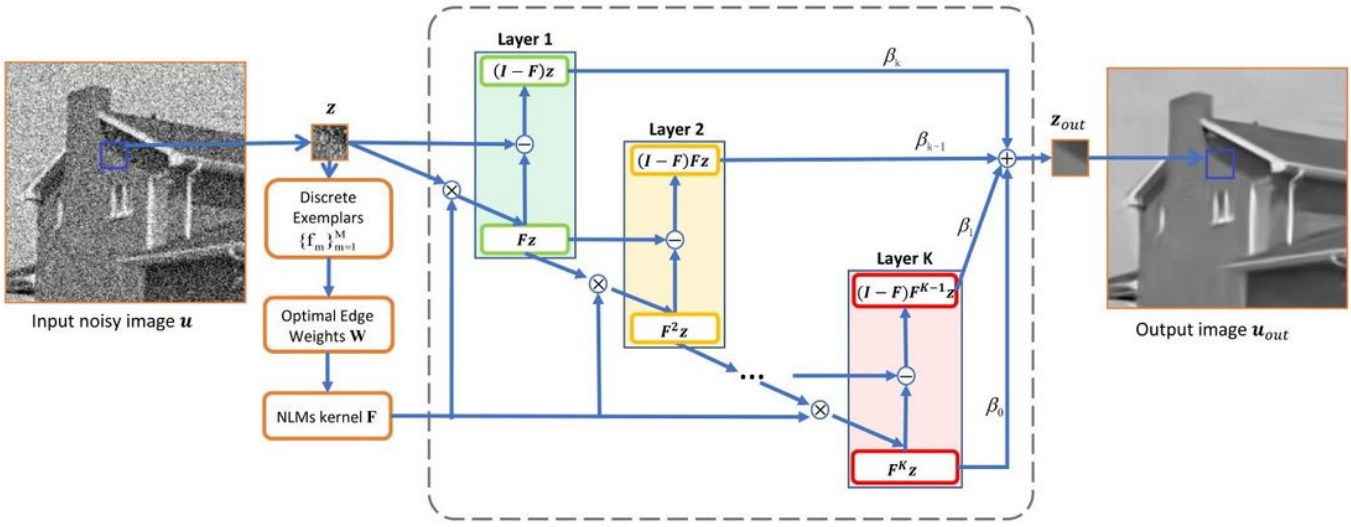


Figure 1

Flowchart of the proposed algorithm

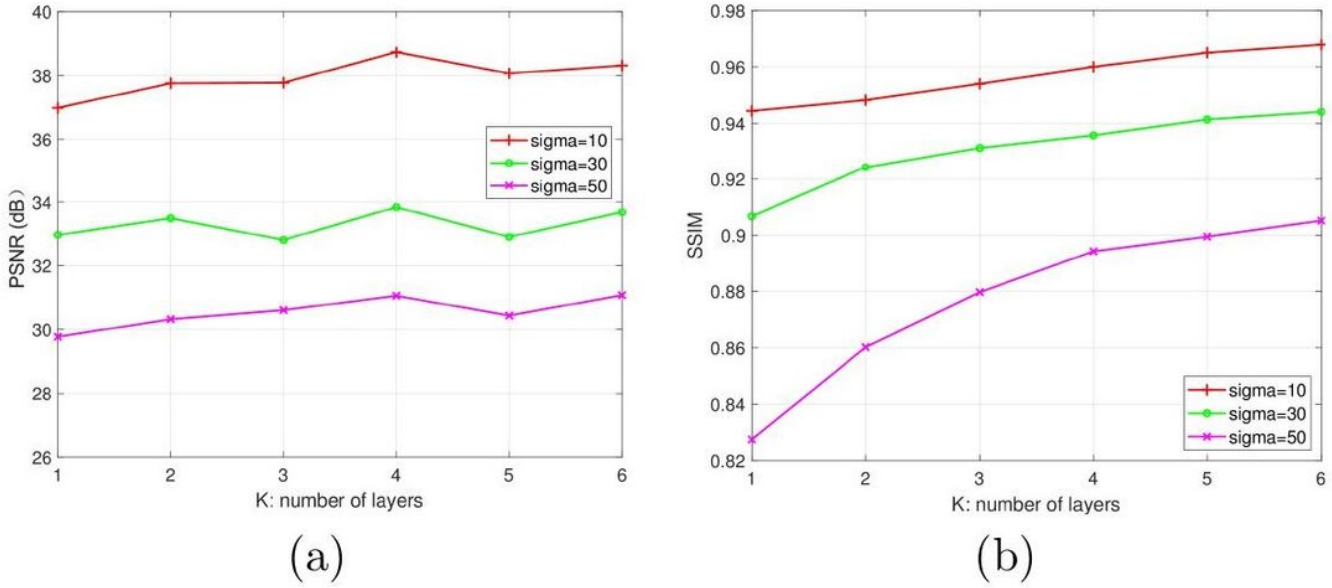


Figure 2

from above to bottom: (a) the average PSNR (dB) of different number of layers, (b) the average SSIM of different number of layers

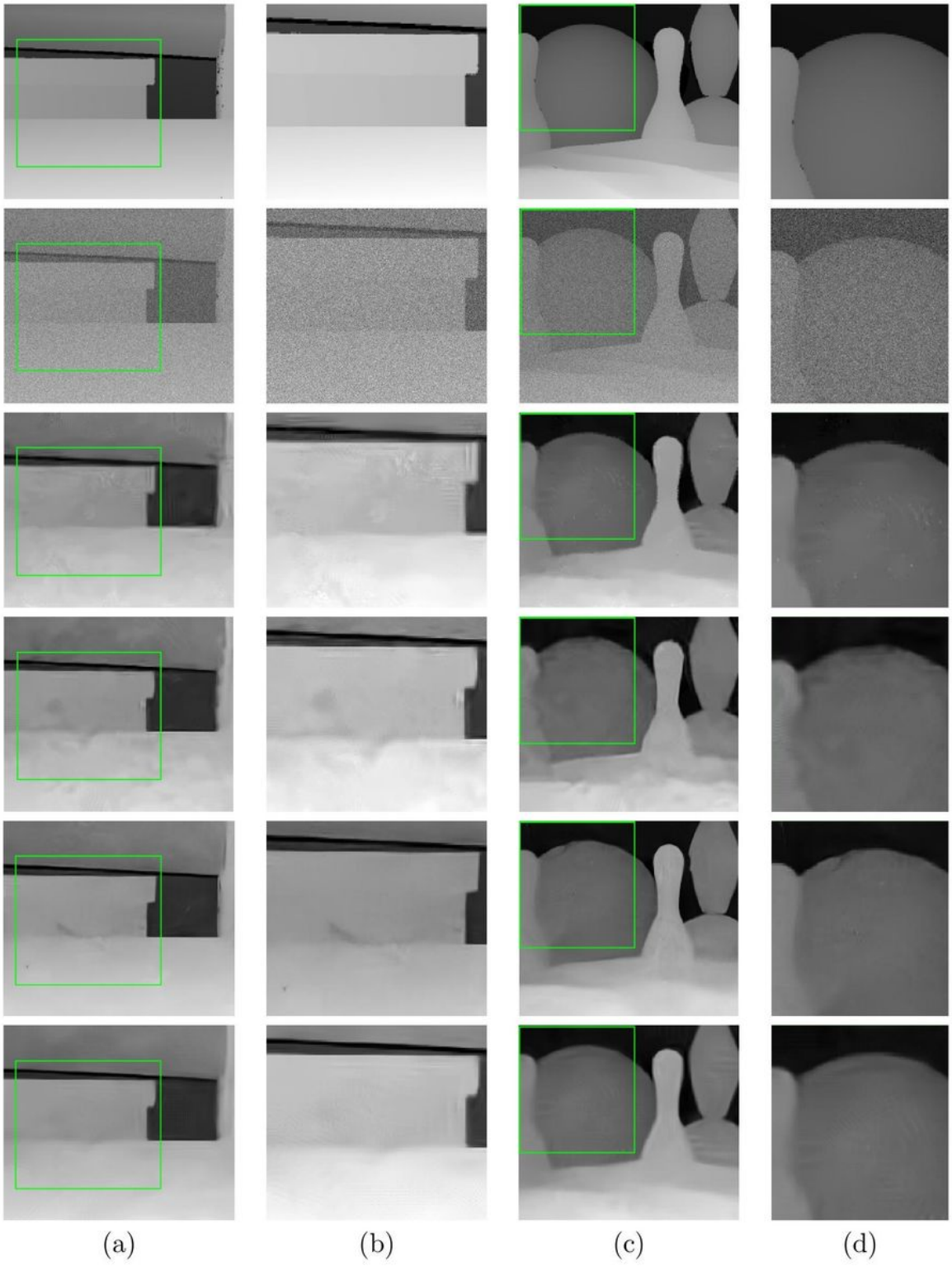


Figure 3

Denoising results of two depth images with the noise variance σ = 50. From the above to bottom are: the original image, the noisy image, the results obtained by OGLR, BM3D, ADnet and the proposed method respectively. Column (b) and (d) are the zoom of the green box of their left (a) and (c).