

# Non-local Image Smoothing with Objective Evaluation

Zhi-Ang Liu, Ying-Kun Hou, Jun Xu, Xian-Tong Zhen, Ling Shao, Ming-Ming Cheng

**Abstract**—With the rapid development of image processing techniques, image smoothing has gained increasing attention due to its important role in other image processing tasks, e.g., image editing and enhancement. However, the evaluation of image smoothing methods is subjectively performed on datasets without proper ground truth images. Therefore, an image smoothing benchmark with reasonable ground-truths is essential to prosper the image smoothing community. In this paper, we construct a new Nankai Smoothing (NKS) dataset containing 200 versatile images blended by natural textures and structure images. The structure images are inherent smooth and can be safely taken as ground truths. On our NKS dataset, we comprehensively evaluate 14 popular image smoothing algorithms. Moreover, we propose a novel Pixel-level Non-Local Smoothing (PNLS) method, exploiting better the non-local self-similarity of natural images to well preserve the structure of the smoothed images. Extensive experiments on several benchmark datasets demonstrate that our PNLS is very effective on the image smoothing task. Comprehensive ablation studies also reveal the work mechanism of our PNLS on image smoothing. To further show its effectiveness, we apply the proposed PNLS on semantic region smoothing, detail/edge enhancement, and image abstraction. The benchmark and code will be publicly released.

**Index Terms**—Image smoothing, benchmark dataset, performance evaluation, non-local similar pixels.

## I. INTRODUCTION

**I**MAGE smoothing is an important image processing problem. It aims to decompose an image into a piece-wise smooth base layer and a detail layer containing only textures [13]. The smooth base layer reflects the main structural information of the image, while the detail layer presents the residual small scale textures in the image. The decomposed layers can be manipulated separately and recomposed in different ways to fulfill specific applications such as image enhancement [22], [44] and image abstraction [7], [53].

In last decade, numerous image smoothing methods have been proposed from different perspectives, which can be roughly divided into three major categories: local filters [18], [29], [32], global filters [13], [31], [54], and deep filters [21], [25], [52]. Local filters smooth the input image by averaging pixel intensity in locally weighted manners [18], [29], [32]. These filters can be computed efficiently, but produce gradient reversals and halo artifacts (especially on edges) [18]. Global filters [13], [31], [54] attenuate this problem of local filters

ZA Liu, J Xu, and MM Cheng are with TKLNNDST, College of Computer Science, Nankai University, Tianjin, China. YK Hou is with School of Information Science and Technology, Taishan University, Tai'an, China. XT Zhen and L Shao are with Inception Institute of Artificial Intelligence (IIAI) and Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. J Xu (nankaimathxujun@gmail.com) is the corresponding author.



Fig. 1. Examples of our Nankai Smoothing (NKS) dataset.

by solving an optimization model in a principled manner on the whole image. However, most global filters are time and memory consuming [57]. Deep filters [21], [25], [52] train smoothing networks with pairs of smoothed and natural images. The smoothed “ground-truth” images are usually generated by other image smoothing methods [51], limiting these deep filters from better performance.

While the development of novel methods is in full swing, the lack of reasonable and objective benchmarks largely lag behind the image smoothing community. The reason is that evaluating novel image smoothing methods objectively requires a benchmark dataset with proper ground-truths and reasonable metrics. Although there are several datasets proposed for the image smoothing task [28], [57], the ground truths are usually generated by other image smoothing methods [57] or by blending cartoon images with synthetic textures [28]. On one hand, the “ground truths” generated by existing smoothing methods are highly biased, and suffer from the “Chicken or the egg” problem. When we use these “ground truths” to evaluate a novel smoothing method, the metric results are not the smoothing performance of that method, but the closeness of the smoothed results by this method to these smoothing methods. On the other hand, the networks learned on cartoon images and synthetic textures could not perform consistently well on smoothing out diverse natural textures from structures.

In this work, we construct a Nankai Smoothing (NKS) dataset with 200 versatile test images blended by structure ground truth images and natural textures. Some examples of our NKS dataset are illustrated in Figure 1. To contribute a baseline for the image smoothing community, we evaluate 14 popular image smoothing algorithms on our NKS dataset. We then present an extensive performance analysis with commonly used metrics. Furthermore, we propose a Pixel-level

Non-Local Smoothing (PNLS) method, which is based on the non-local self similarity (NSS) property of natural images [4], [10], [19] and outperforms previous image smoothing algorithms on several benchmark datasets (including our NKS). As far as we know, our PNLS is the first method to tackle the image smoothing problem from the perspective of pixel-level non-local self-similarity. Extensive experiments on our NKS dataset demonstrate that our PNLS method achieves better performance on objective metrics and subjective visual quality (by user study) than previous methods. To show its broad practicality, we also apply the proposed PNLS into various image processing tasks, such as salient region smoothing, image detail/edge enhancement, and image abstraction, *etc.*

In summary, our major contributions are manifold:

- **We contribute a new Nankai Smoothing (NKS) dataset with proper ground truths for the image smoothing community.** Our NKS dataset contains 200 images blended by structure and texture images, while the structure images can be safely taken as the ground truths.
- **We comprehensively evaluate 14 popular image smoothing algorithms** with our NKS dataset, on commonly used objective metrics and visual quality.
- **We propose a novel Pixel-level Non-Local Smoothing (PNLS) method** by utilizing the non-local self similarity (NSS) prior of natural images. Experiments shows that the proposed PNLS achieves promising image smoothing performance on several benchmark datasets.
- **We apply the proposed PNLS method on several image processing tasks** to show its broad practicality.

The remainder of this paper is organized as follows. In §II, we review the related work. In §III, we present the Nankai Smoothing (NKS) dataset for image smoothing, and benchmark 14 popular image smoothing algorithms on it. We then introduce our PNLS smoothing method in §IV. In §V, we perform extensive experiments on several datasets to demonstrate the advantages of our PNLS. We also provide more applications of our PNLS method on other image processing tasks in §VI. The Conclusion is given in §VII.

## II. RELATED WORK

### A. Image Smoothing Methods

**Local filters** explicitly filter each pixel as a weighted average of its neighborhood pixels in an one-step or iterative way. Bilateral filter (BF) [29] is a simple and intuitive method in this category, and widely applied in other image processing tasks [15], [24], [43]. However, since not all pixels have enough similar pixels around, the weighted average would be biased by outlier pixels, thus resulting gradient reversal artifacts [18]. It is also generalized to Joint Bilateral Filter (JBF) [32], in which the weights are computed upon another guidance image instead of the input image itself. The insights of resorting to a guidance image is later flourished in the Guided Filter (GF) [18]. GF has inspired numerous methods due to its  $\mathcal{O}(N)$  complexity for an image with  $N$  pixels. However, it cannot resolve the ambiguity regarding whether or not to smooth certain edges.

**Global filters** [13], [51], [53] solve an optimization function in a principled manner on the whole image. The predefined function is consisted of a fidelity term for data fitting and a prior regularization term for smoothness. They could attenuate the limitations of local filters such as gradient reversals and halo artifacts [13]. Among these methods, Weighted Least Square (WLS) [13] adjusts the matrix affinities according to the image gradient and produces halo-free smoothing results. Later, a semi-global extension of WLS [26] is proposed to solve the linear system in a time and memory efficient manner. The  $L_0$  gradient minimization (L0) [51] globally controls the number of non-zero gradients which are involved in approximating the prominent structure of input image. However, one unavoidable problem is that they are prone to over-sharpen the edges while smoothing the details [13], [26], [51]. The Rolling Guidance Filtering (RGF) [56] filters images with the complete control of detail smoothing under a scale measure, employing BF for filtering with a rolling guidance implemented in an iterative manner. Global filters usually need to solve large optimization systems, and therefore are computationally expensive. Besides, these global filters often sacrifice the local edge-preserving effects for better globally smoothing performance.

**Deep filters** mostly focus on accelerating while approximating state-of-the-art local or global filters such as BF [29] or WLS [13]. Deep Edge-Aware Filter (DEAF) [52] is a pioneering work in this category. It trains the network in the gradient domain, and the filtered output is reconstructed from the refined gradients produced by the deep network. In [25], a hybrid neural network is proposed based on the recursive filters whose coefficients can be learned by a deep network. This method utilizes convolutional neural networks (CNNs) for perceiving salient structures, and recurrent networks for producing smoothing output in a data-driven manner. Li *et al.* [21] proposed a learning-based approach to construct a joint filter based on CNNs. Shen *et al.* [35] introduces a convolutional neural pyramid to extract features of different scales, aiming at extracting larger receptive fields from input images. The work of [7] utilizes context aggregation networks to include more contextual information. Lu *et al.* [28] developed a structure and texture dataset and trained a texture and structure aware network, hence enabling their method the awareness. The work of [12] introduces an unsupervised learning CNN that facilitates generating flexible smoothing effects. One common issue is that all these approaches take the output of existing filters as “ground-truth”, and hence can hardly outperform these “teacher” filters.

### B. Image Smoothing Benchmarks

**Datasets.** There are several datasets of BSDS500 [30], DIV2K [1], MIT5K [5] originally collected for the image segmentation [42], super-resolution [23], and image denoising [33] tasks, but are also used to present image smoothing performance. But these images do not have corresponding smooth ground-truths. A dataset of 500 images is published with the proposed image smoothing algorithm RTV [53], but similarly this dataset does not provide ground-truths. Zhu *et al.* [57] proposed a benchmark for image smoothing. Unfortunately, the ground-truth smoothed images are generated

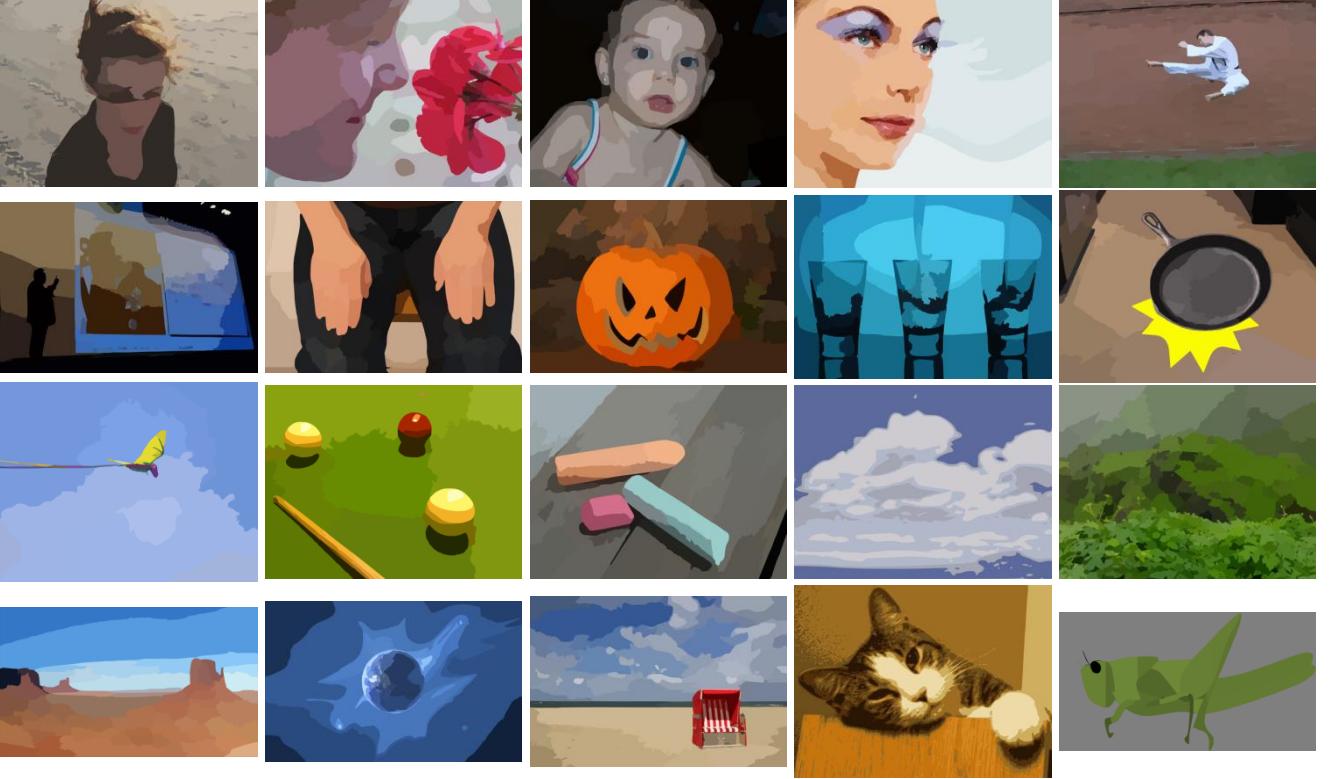


Fig. 2. The 20 structure images we used in our NKS dataset.



Fig. 3. The 10 natural texture images we used in our NKS dataset.

by existing smoothing algorithms. These “ground-truths” are prone to be subjective since in fact we could only evaluate the performance difference between a new algorithms and these handpicked existing algorithms. In our Nankai Smoothing (NKS) dataset, we collect the structure images as ground-truths and acquire samples by blending images.

**Evaluation metrics.** There are several meaningful evaluation metrics for the measurements of image quality. Peak-Signal-to-Noise-Ratio (PSNR) [2] is an objective metric computing the error between the original image and the distorted image. It is popular in image restoration tasks due to its intuitive meanings on measuring image quality and simple mathematical format. However, PSNR focuses on the pixel-level difference between two images, but ignores the visual characteristics of the human eye [40]. To solve this problem,

the Structural Similarity index (SSIM) [40] is designed to comprehensively measure image similarity from the aspects of brightness, contrast and structure. SSIM takes into account the correlation of structural patches instead of pixels, and hence is more in line with human eye’s judgment on image quality. Following SSIM, a series of variants are carried out [20], [34], [41]. By considering that not all pixels in an image are of the same importance, the Feature Similarity index (FSIM) [55] uses low-level features to evaluate the distance between the reference image and the distorted image. For image smoothing algorithms, one common evaluation manner is to subjectively evaluate the smoothed image through visual quality. However, this manner may lack accurate measurements on the visual quality. Although the objective metrics of Weighted Root Mean Squared Error (RMSE) and Weighted Mean Absolute

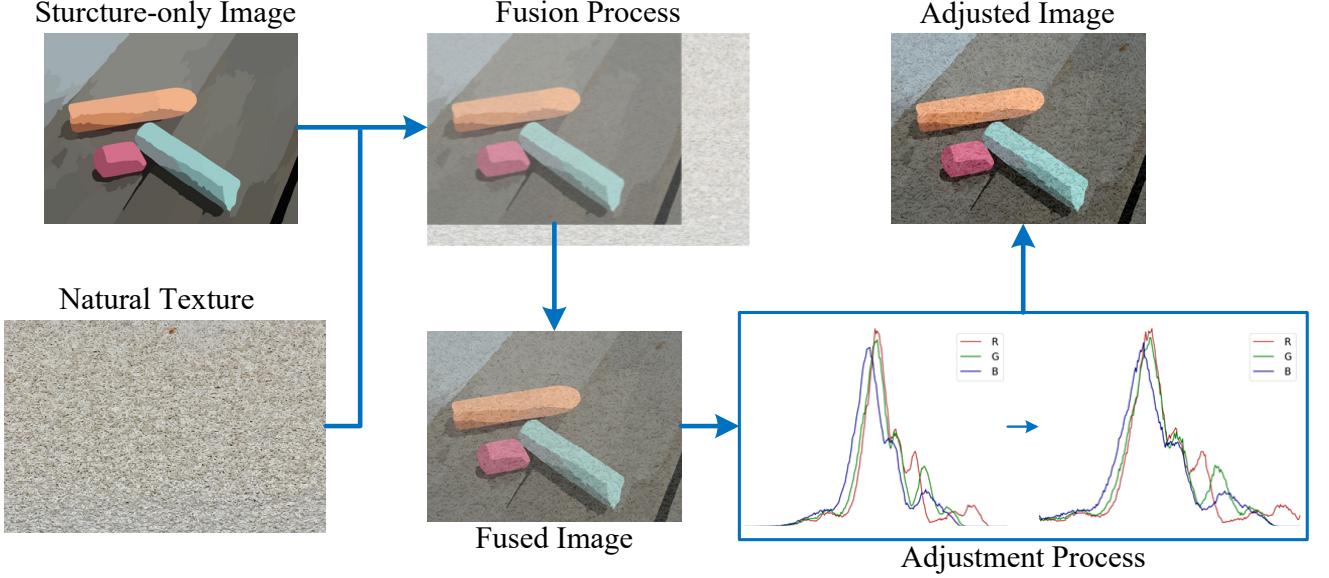


Fig. 4. **Flowchart of the image blending process.** We first directly blend the structure image and the natural texture in a certain ratio, and obtain the fused image. Then we adjust the brightness of three channels of the fused image separately to remove the color distortion caused by the texture, and get the final color adjusted image.

Error (WMAE) have been used in [57], they are stuck with the “ground truths” produced by existing smoothing methods. On our NKS benchmark, we employ PSNR, SSIM [40] and FSIM [55] as the evaluation metrics due to their consistent prediction with the visual perception of humans.

### III. PROPOSED NANKAI SMOOTHING DATASET

In this section, we develop a Nankai Smoothing (NKS) dataset for the image smoothing task, and evaluate 14 state-of-the-art algorithms on it with three commonly used metrics, i.e., PSNR, SSIM [40], and FSIM [55].

#### A. Constructing the NKS Image Smoothing Dataset

**Motivation.** Since manually annotating the structure of an image is subjective and costly, extracting structure ground truths directly from natural images is difficult. Note that image smoothing is very similar to the image denoising task [38], [45], [47]: both aim to filter out small scale components (textures or noise) from the image. Despite the collection of smoothing ground truths is ambiguous [28], [57], constructing image denoising datasets [37], [46] is usually explicit and feasible. That is, we add the noise to the clean image, and the synthetic noisy image is to be recovered by removing the noise [48]–[50]. In evaluation, the corresponding clean images are naturally taken as the ground truths to compute the objective metrics such as PSNR and SSIM [40]. Similarly, in image smoothing we can blend structure and texture images to generate the test images, and the structure images can be recovered by removing the texture from the blended image.

**Limitation of previous datasets.** Though previous work [28] is working towards this direction, it suffers from three limitations: 1) the cartoon images used as the ground truths limits its generality, since the structure images can be very versatile, and cartoon is only a specific type of them; 2) the synthetic

textures are not natural as real-world color images; and 3) their dataset is not publicly released, making it difficult for others to evaluate novel smoothing methods on this dataset.

**Collecting structure and texture images.** We observe that vector images can be safely regarded as smooth structure images (Figure 2), and construct our NKS dataset by blending vector images and texture images. Specifically, we search the key word of “vector” on the Pixabay website [9] and select 20 highly realistic vector images from thousands of free structure vector images provided by the website. In addition, we manually select 10 natural texture images from the Pixabay website [9]. The selected structure images and natural textures are shown in Figures 2 and 3, respectively. We observe that the vector images are very smooth with clear structures.

**Generating blended images.** To generate mixed structure and texture images, we blend each of the 20 structure images and each of the 10 natural textures in a reasonable manner. Each structure image can be safely taken as the ground truth for the corresponding images blended by that structure image and the 10 natural textures. The image blending process is illustrated in Figure 4. We first mix the structure vector images and natural textures. Since directly adding structure and texture images together would result in overflow of pixel values, we mix the two components in proper proportions.

**Color adjustment.** Directly adding the natural color textures to the structure images would produce severe color distortion in the blended images, and existing image smoothing methods would suffer from poor performance on the directly blended images (as will be illustrated later). Therefore, we need to adjust separately the brightness of the Red, Green, and Blue channels of the blended images, to attenuate the color distortion from the corresponding structure images. In this way, we collect overall  $20 \times 10 = 200$  color adjusted images in our NKS dataset, with 20 structure images as the ground-truths.

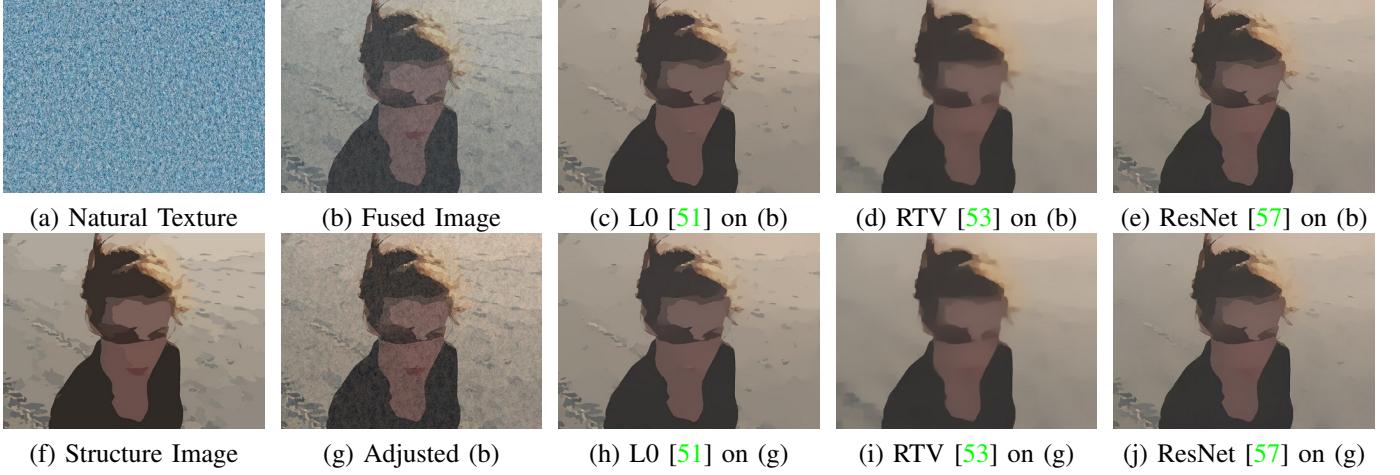


Fig. 5. **Importance of the color adjustment.** We get the fused image by directly blending the natural texture (a) and the structure image (f) (the ground truth). Comparing the images before and after color adjustment, we observe that the fused image (b) produces color distortion on the structure image. Our adjusted image (g) preserves the texture and eliminates this color distortion. Thus, the smoothed images (c)-(e) by three state-of-the-art methods on the adjusted image (g) are closer to the ground-truth than the corresponding images (h)-(j).

TABLE I  
SUMMARY OF OUR NKS SMOOTHING DATASET.

Class	Size		Number	Structures	Textures
	Width	Height			
Human	512	340 ~ 384	70		
Artifact	512	343 ~ 397	50		
Landscape	512	298 ~ 384	60	Vector	Carpet, Wood, ...
Animal	512	277 ~ 384	20		

**Dataset statistics.** Our NKS dataset contains 200 images of versatile scenarios. We present the statistics of our NKS dataset in Table I, from which one can see that our NKS dataset consists of diverse contents, such as humans (*e.g.*, children, women,...), artifacts (*e.g.*, pot, chalk,...), landscapes (*e.g.*, forest, beach,...), and animals (*e.g.*, cat, insect,...), *etc*. All images are resized into the width of 512 with proportionally heights, by the default Matlab function “imresize”.

**Color adjustment is essential to our NKS dataset.** To illustrate this point, we employ three state-of-the-art methods, L0 [51], RTV [53] and the ResNet in [57], to smooth the directly blended images and color adjusted ones, as shown in Figure 5. The smoothed images of these methods on the directly blended image Figure 5 (b) and adjusted image Figure 5 (g) are listed in Figures 5 (c)-(e) and Figures 5 (h)-(j), respectively. One can see that the smoothed images on the adjusted image (Figure 5 (g)) are with better visual quality than those on the directly blended image (Figure 5 (b)), when compared to the structure image (Figure 5 (f)).

In addition, we run seven popular smoothing methods L0 [51], RTV [53], TF [3], RGF [56], FIP [7], and the baselines ResNet and VDCNN in [57] on the directly blended images and the corresponding adjusted images. The results on PSNR and SSIM [40] are listed in Table II. One can see that the comparison methods achieve significantly higher PSNR and SSIM [40] results on the adjusted images than those on the directly blended images. The reason is that the directly blended images, though been smoothed, still suffer from severe color distortion from the original structure images.

TABLE II  
AVERAGE PSNR AND SSIM [40] RESULTS OF DIFFERENT METHODS ON THE 200 DIRECTLY BLENDED IMAGES AND COLOR ADJUSTED IMAGES, RESPECTIVELY. “↑” MEANS THAT HIGHER IS BETTER.

Methods	Directly Blended		Color Adjusted	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
L0 [51]	23.73	0.8923	<b>33.01</b>	<b>0.9249</b>
RTV [53]	23.51	0.8650	<b>31.81</b>	<b>0.9206</b>
TF [3]	24.16	0.8724	<b>33.23</b>	<b>0.9186</b>
RGF [56]	24.23	0.8720	<b>32.51</b>	<b>0.9135</b>
FIP [7]	23.50	0.8779	<b>32.03</b>	<b>0.8946</b>
VDCNN [57]	23.79	0.8855	<b>33.38</b>	<b>0.9349</b>
ResNet [57]	23.88	0.8907	<b>33.13</b>	<b>0.9354</b>

However, owing to the color adjustment, the smoothed results on the adjusted images by different methods can be very close to the structure images, with little color distortion. These results demonstrate that the color adjustment operations are essential for the smoothing methods to achieve reasonable visual perception and satisfactory quantitative results.

### B. Benchmarking Image Smoothing on our NKS dataset

**Comparison methods.** We evaluate 14 image smoothing algorithms on NKS dataset in total. These algorithms include 10 traditional filters: BF [29], WLS [13], EAW [14], GF [18], L0 [51], RTV [53], TF [3], FGS [31], RGF [56], fastABF [16] and 4 deep filters: LRNN [25], FIP [7], two baselines ResNet and VDCNN used in [57]. We employ the commonly used metrics of PSNR, SSIM [40], and FSIM [55] to quantitatively evaluate the performance of the comparison methods on our NKS dataset. These metrics measure the distance between the smoothed images and the corresponding ground-truths.

**Results.** The comparison results are listed in Table III. One can see that, on the three metrics, TF [3] performs best among the traditional (local and global) filters, while the baselines VDCNN and ResNet in [57] performs better than the other two deep filters. We also report the average running time (in seconds) of different methods on the 110 images of size 512 × 384 in our NKS dataset. Specifically, FIP [7], two baselines

TABLE III  
COMPARISON OF AVERAGE PSNR, SSIM [40], AND FSIM [55] BY 14 STATE-OF-THE-ART IMAGE SMOOTHING ALGORITHMS ON OUR NKS DATASET. THE AVERAGE RUNNING TIME (IN SECONDS) OF THESE METHODS (ON CPU OR GPU) IS REPORTED ON THE 110 IMAGES OF SIZE  $512 \times 384$  IN OUR NKS DATASET. OUR METHOD WILL BE INTRODUCED IN §IV. “PUB.” MEANS “PUBLICATION VENUES”.

No.	Method	PSNR	SSIM	FSIM	Device	Time
Traditional Filters	1 BF [29] <sub>IICCV'98</sub>	32.00	0.8478	0.8556	CPU	1.53
	2 WLS [13] <sub>TOG'08</sub>	28.59	0.9011	0.9107	CPU	0.91
	3 EAW [14] <sub>TOG'09</sub>	28.02	0.7953	0.8200	CPU	<b>0.02</b>
	4 GF [18] <sub>ECCV'10</sub>	32.09	0.8779	0.8672	CPU	<b>0.03</b>
	5 L0 [51] <sub>TOG'11</sub>	33.01	0.9249	0.9374	CPU	0.48
	6 RTV [53] <sub>TOG'12</sub>	31.81	0.9206	0.9234	CPU	0.68
	7 TF [3] <sub>TIP'13</sub>	33.23	0.9186	0.9149	CPU	0.33
	8 FGS [31] <sub>TIP'14</sub>	23.46	0.8368	0.7978	CPU	<b>0.03</b>
	9 RGF [56] <sub>ECCV'14</sub>	32.51	0.9135	0.9128	CPU	0.22
	10 fastABF [16] <sub>TIP'18</sub>	31.44	0.8977	0.8917	CPU	0.37
Deep Filters	11 LRNN [25] <sub>ECCV'16</sub>	30.61	0.8666	0.8600	CPU	0.43
	12 FIP [7] <sub>IICCV'17</sub>	32.03	0.8946	0.9061	GPU	0.45
	13 VDCNN [57] <sub>TIP'19</sub>	33.38	0.9349	0.9395	GPU	1.47
	14 ResNet [57] <sub>TIP'19</sub>	33.13	0.9354	<b>0.9434</b>	GPU	3.76
Ours	15 PNLS (Fast)	<b>33.45</b>	<b>0.9378</b>	0.9397	CPU	5.10
	16 PNLS (Slow)	<b>33.68</b>	<b>0.9420</b>	<b>0.9440</b>	CPU	78.68

ResNet and VDCNN used in [57] are tested on an NVIDIA GTX 1080 GPU and the other filters are tested on the Intel Core i7-6700K CPU. One can see that EAW [14], GF [18] and FGS [31] averagely take 0.02, 0.03, and 0.03 seconds to process a  $512 \times 384$  image, much faster than the other filters.

#### IV. PROPOSED PIXEL-LEVEL NON-LOCAL SMOOTHING

In this section, we present the proposed Pixel-level Non-Local Smoothing (PNLS) method, which is consisted of three steps: 1) searching non-local similar pixels (§IV-A); 2) estimating the smoothing threshold (§IV-B); and 3) smoothing by Haar transformation based thresholding (§IV-C). The flowchart of our PNLS is plotted in Figure 6.

We first transform an RGB image into the luminance-chrominance space [10], and get the corresponding YCbCr image. We search similar pixels and estimate the smoothing threshold in the Y channel. The similar pixels of the Cb and Cr channels are grouped according to results in the Y channel. Then, we perform image smoothing by threshold based Haar transformation on each channel. Finally, we transform the smoothed image in YCbCr space back to the RGB space.

##### A. Searching Non-local Similar Pixels

For the input image  $\mathbf{I} \in \mathbb{R}^{h \times w}$ , we extract reference patches of size  $m \times m$  with a step of  $s$  (horizontally and vertically) from its Y channel. For each reference patch, we first search

its similar patches in a window of size  $R \times R$ . The similarity is trivially measured by Euclidean distance. Then we reshape each similar patch into a vector  $\mathbf{v}_r \in \mathbb{R}^{m^2}$  ( $r = 1, \dots, R^2$ ,  $\mathbf{v}_1$  is the reference patch). We perform patch matching by selecting the  $q$  closest patches (including  $\mathbf{v}_1$  itself) to  $\mathbf{v}_1$ . By stacking the  $q$  vectors in columns, we get the similar patches matrix  $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_q] \in \mathbb{R}^{m^2 \times q}$ .

Since natural textures are usually repetitive, pixels are likely to be very similar in the same relative positions of different patches. This property of textures is very different from that of structures, in which adjacent pixels vary greatly on edges. To utilize this property for image smoothing, we take the  $i$ -th row  $\mathbf{v}^i \in \mathbb{R}^q$  ( $i = 1, \dots, m^2$ ) of  $\mathbf{P}$  as the reference row, and calculate the Euclidean distances between the reference row  $\mathbf{v}^i$  and the other rows  $\{\mathbf{v}^j \in \mathbb{R}^q, j = 1, \dots, m^2\}$ , as follows:

$$d^{ij} = \|\mathbf{v}^i - \mathbf{v}^j\|_2. \quad (1)$$

We select the  $p$  rows of pixels with the minimal distances to the reference row  $\mathbf{v}^i$ , and form the similar pixels matrix  $\mathbf{S} = [\mathbf{v}^{i_1}, \dots, \mathbf{v}^{i_p}] \in \mathbb{R}^{p \times q}$ . Note that we have  $\mathbf{v}^{i_1} = \mathbf{v}^i$  and  $d^{ii_1} = 0$ . The similar pixels matrices in Cb and Cr channels of the matrix  $\mathbf{P}$  are extracted corresponding to the Y channel.

##### B. Estimation of Smoothing Threshold

A threshold is essential to determine whether or not (and the extent to which) smooth each similar pixels matrix. Since the pixels in the  $p$  rows of  $\mathbf{S}$  are very close, we can consider the variance as the energy estimation due to texture changes and compute it as

$$\sigma = \frac{1}{m^2(p-1)\sqrt{q}} \sum_{t=2}^p \sum_{i=1}^{m^2} d^{it}. \quad (2)$$

To perform consistent image smoothing, we set a global threshold as the average  $\sigma$  of all similar pixels matrices.

##### C. Smoothing by Haar Transformation based Thresholding

In §IV-A, we have obtained a set of similar pixels matrices  $\mathbf{S} \in \mathbb{R}^{p \times q}$  and the threshold  $\sigma$ . We then employ the Haar transformation [17] for thresholding similar pixels matrices. Specifically, we utilize the Haar transformation with lifting scheme [11], [36], which includes vertical transformation matrix  $\mathbf{H}_l \in \mathbb{R}^{p \times p}$  and horizontal transformation matrix  $\mathbf{H}_r \in \mathbb{R}^{q \times q}$ . In order to perform Haar transformation, we set  $p, q$  to be powers of 2. The detailed transformation process is provided in the *Supplementary File*. Thus, we get the transformed representation matrix  $\mathbf{T} \in \mathbb{R}^{p \times q}$ :

$$\mathbf{T} = \mathbf{H}_l \mathbf{S} \mathbf{H}_r. \quad (3)$$

By using the smoothing threshold, we could restore the element in  $i$ -th ( $i = 1, \dots, q$ ) row,  $j$ -th ( $j = 1, \dots, m^2$ ) column of the transformed representation matrix  $\mathbf{T}$  via

$$\hat{\mathbf{T}} = \mathbf{T} \odot \mathbb{I}_{\{|T| \geq \lambda\sigma^2\}}, \quad (4)$$

where  $\odot$  means element-wise production,  $\mathbb{I}$  is the indicator function, and  $\lambda$  is the parameter controlling the extent to threshold. According to the wavelet theory [36], the elements in the last two rows of  $\mathbf{T}$  (except the 1-st column) belongs

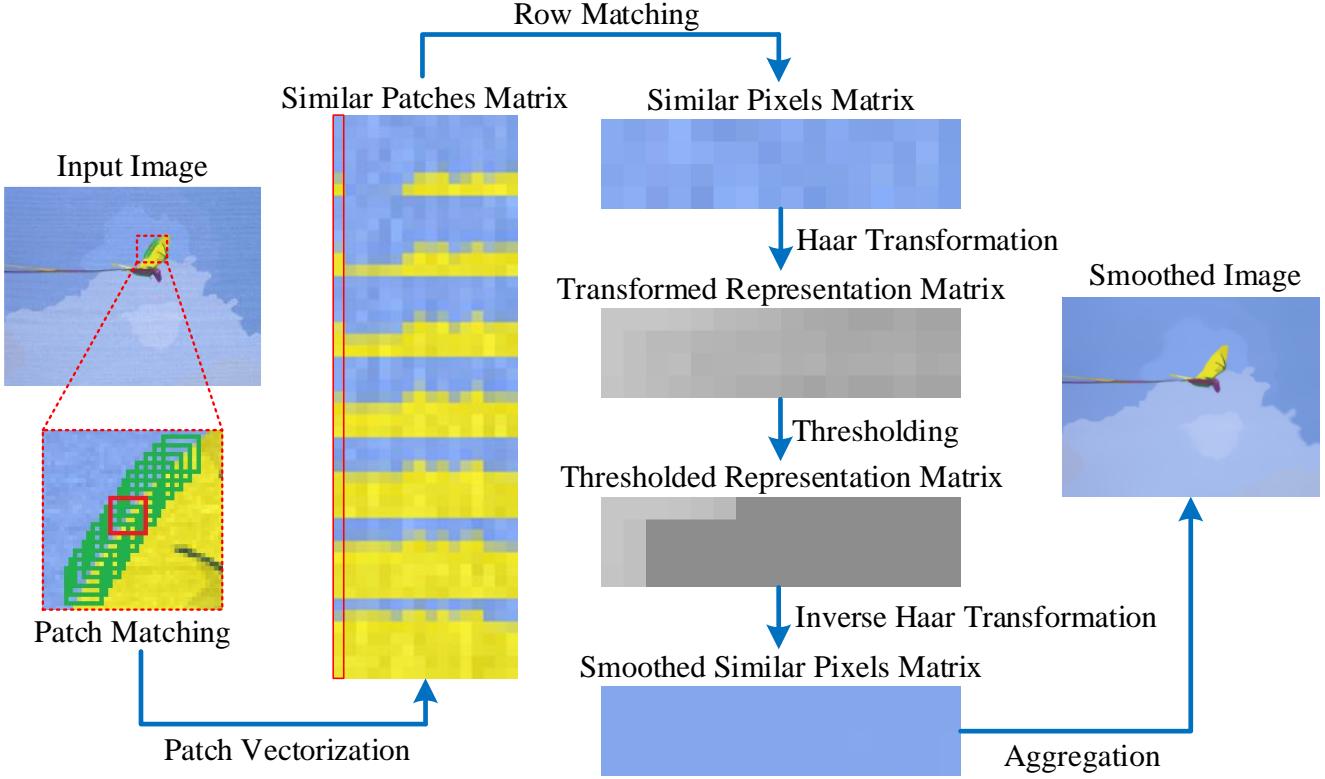


Fig. 6. **Flowchart of our proposed PNLS smoothing method.** First, we select multiple reference patches for smoothing in the input image. For a separate reference patch (shown in the red bounding box), the Euclidean distance is used to perform patch matching and row matching to obtain the Similar Pixels Matrix. Then, the three channels of the Similar Pixels Matrix are transformed by using the Haar transformation, and the threshold is used for smoothing. Then the Thresholded Representation Matrices of the three channels are to inverse Haar transformation to obtain the Smoothed Similar Pixels Matrix. Finally, we aggregate it back into the image to complete the smoothing process of a single reference patch. The above process is performed for all selected reference patches to complete one iteration of image smoothing. We perform our PNLS several iterations to improve its smoothing performance.

to high frequency bands of the Haar transformation, and these elements are texture information. We directly set these elements as zero in  $\hat{T}$ :

$$\tilde{T}(i, j) = \hat{T}(i, j) \odot \mathbb{I}_{\{\text{if } i=1, \dots, q-2 \text{ or } j=1\}}, \quad (5)$$

where  $\tilde{T}(i, j)$  and  $\hat{T}(i, j)$  are the elements in  $i$ -th ( $i = 1, \dots, q$ ) row,  $j$ -th ( $j = 1, \dots, m$ ) column of matrices  $\tilde{T}$  and  $\hat{T}$ , respectively. We then use the vertical inverse Haar transformation matrix  $H_{il} \in \mathbb{R}^{p \times p}$  and horizontal inverse Haar transformation matrix  $H_{ir} \in \mathbb{R}^{q \times q}$  on thresholded representation matrix  $\tilde{T}$ . The detailed inverse transformation process is also provided in the *Supplementary File*. Then we could get the smoothed similar pixels matrix  $\tilde{S}$  without texture via

$$\tilde{S} = H_{il} \tilde{T} H_{ir}. \quad (6)$$

Finally, the smoothed similar pixels matrices are aggregated to the corresponding positions in the original image. The above is the detailed process of image smoothing based on Haar transformation techniques [17].

#### D. Iterative Smoothing Scheme

For better performance, we apply the above smoothing process for  $N = 10$  iterations. Experiments show that our PNLS method with  $N = 10$  achieves satisfactory smoothing results (please refer to §V for more details).

#### E. Complexity Analysis

The proposed PNLS method contains three parts: 1) in §IV-A, the complexity of patch matching is  $\mathcal{O}(whR^2m^2/s^2)$ , while the complexity of row matching is  $\mathcal{O}(whqm^4/s^2)$ ; 2) in §IV-B, the complexity for smoothing threshold estimation is  $\mathcal{O}(whm^2p/s^2)$ ; 3) in §IV-C, the complexity for Haar transformation based thresholding is  $\mathcal{O}(whpqm^2/s^2)$ . Since the above process iterates  $N$  times, the complexity of our PNLS is  $\mathcal{O}(whm^2/s^2 \cdot \max\{R^2, qm^2, pq\})$ .

## V. EXPERIMENTS

In this section, we first compare the proposed Pixel-level Non-Local Smoothing (PNLS) method with competitive methods on several image smoothing benchmark datasets. We also perform comprehensive ablation studies to gain deeper insights into the proposed PNLS method. More comparison results of visual quality can be found in the *Supplementary File*.

#### A. Implementation Details

**Parameter settings.** As shown in Table III, we have two versions of PNLS: fast PNLS (No. 15) and slow PNLS (No. 16). The parameters of our fast PNLS include the size  $R = 15$  of searching window, the step  $s = 4$  for extracting neighbor reference patches, the patch size  $m = 4$ , the iteration number  $N = 10$ , the threshold parameter  $\lambda = 0.4$ . For our slow PNLS,

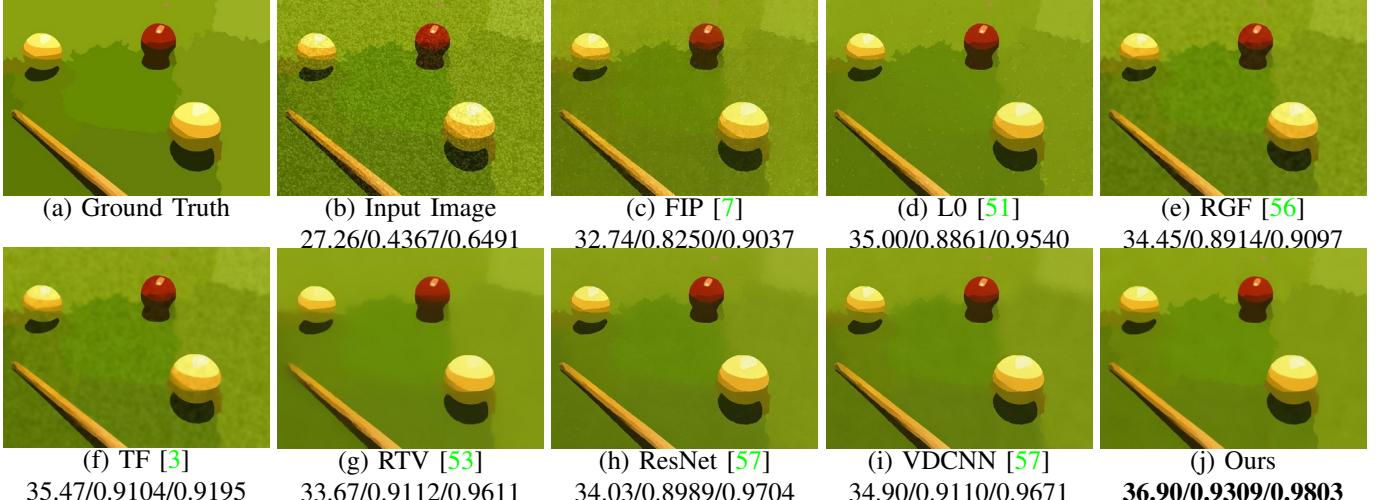


Fig. 7. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S15\_T01” from our NKS dataset. The best results are highlighted in **bold**.

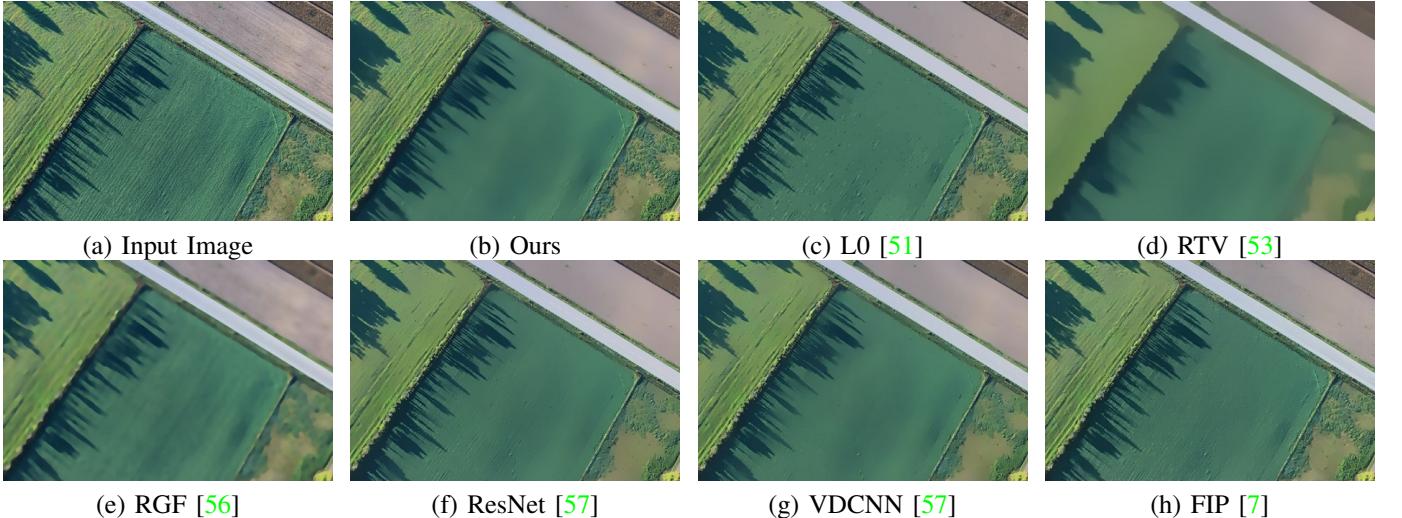


Fig. 8. Comparison of smoothed images by different methods on the image “0261” from the DIV2K dataset [1].

the parameters are almost the same with those of fast PNLS, but the step is set as  $s = 1$  (rather than  $s = 4$  in fast PNLS) to extract more reference patches.

**Comparison methods.** We compare the proposed PNLS method with 14 state-of-the-art image smoothing methods: BF [29], WLS [13], EAW [14], GF [18], L0 [51], RTV [53], TF [3], FGS [31], RGF [56], fastABF [16], LRNN [25], FIP [7], the two baselines ResNet and VDCNN in [57]. For every comparison method, we download its original code from the corresponding authors’ website, and perform experiments with its default parameter settings. The comparisons are evaluated on PSNR, SSIM [40], FSIM [55], and visual quality.

**Datasets.** We evaluate our PNLS with the comparison image smoothing methods on the DIV2K dataset [1] (1000 high-resolution RGB images with diverse contents), the images used in RTV [53], and the Edge-Preserving image Smoothing (EPS) dataset [57] (500 images). Note that the images in the DIV2K [1] and RTV [53] datasets do not have corresponding ground truth images, while the ground truths of the EPS dataset [57] are constructed by manually selecting the smoothed images generated by five existing state-of-the-art image smoothing algorithms.

**Evaluation.** On the three datasets of [1], [53], [57], we evaluate the performance of different smoothing methods by subjectively comparing the visual quality of smoothed images. On our NKS dataset with structure images as ground truths. Thus we evaluate the performance of different smoothing algorithms qualitatively on visual quality, and quantitatively on PSNR, SSIM [40] and FSIM [55].

## B. Comparison Results

As shown in Table III, the slow version of our PNLS achieves the best results of PSNR/SSIM/FSIM on the NKS dataset. Our fast PNLS achieves comparable objective performance when compared to the slow PNLS. Besides, it greatly improves the running time, and needs averagely 5.10 seconds to process a  $512 \times 384$  image. In Figures 7, 8, 9, and 10, we compare the visual results of our PNLS method with the other state-of-the-art image smoothing methods. We observe that our PNLS method preserves the structures of the image contents while smoothing out the textures of the images.

In Figure 10, RTV [53] and RGF [56] well erase the wall textures, but destroy the structure of the human face, making

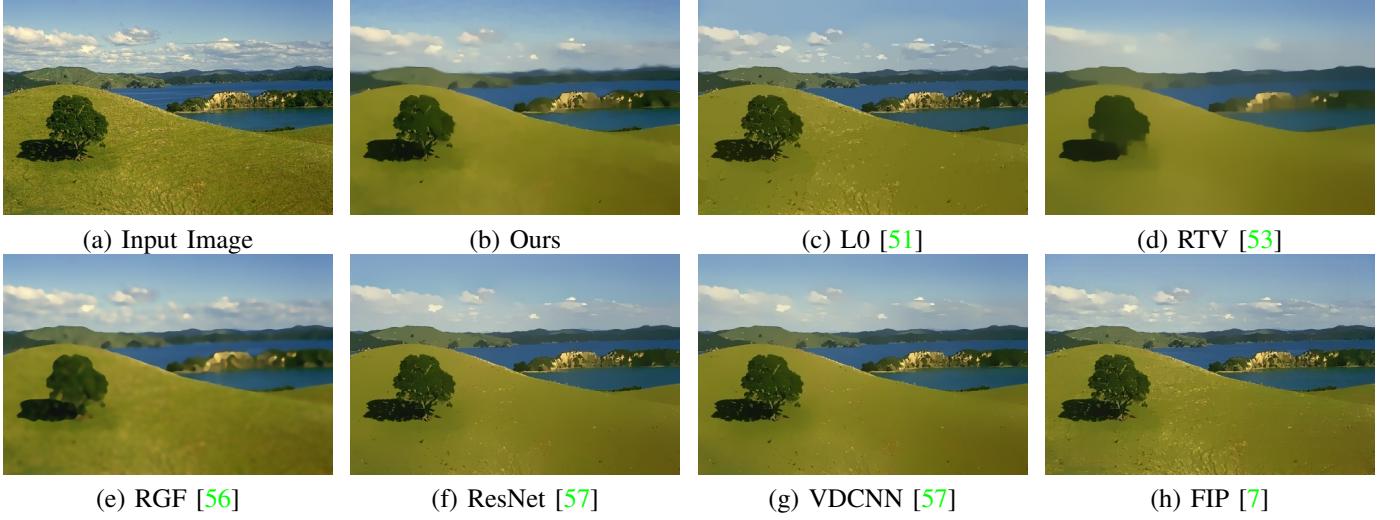


Fig. 9. Comparison of smoothed images by different methods on the image “0251” from the dataset [57].



Fig. 10. Comparison of smoothed images by different methods on the image “03\_11” from the dataset in [53].

it blurry. The other methods do not even fully remove the wall textures. But our PNLS ensures the integrity and sharpness of the human face while well removing the wall textures. Note that our PNLS outperforms the deep learning based smoothing networks in [57], even though they are trained by plenty of images with the ground truths. In Figure 7, we compare both the visual quality (subjective metric) and PSNR/SSIM/FSIM (objective metrics) results of different image smoothing methods on the image “S15\_T01” (“15” is the index of structure image, while “01” is the index of texture one).

### C. User Study on Our NKS dataset

Though our NKS dataset has ground truths, it is more convincing to subjectively evaluate the comparison methods by highly controlled user study, as did by previous work [57]. To perform user study together with our algorithm, we select seven state-of-the-art methods: L0 [51], RTV [53], RGF [56], FIP [7], TF [3] and the two baselines ResNet and VDCNN in [57]. We perform user study with the help of 80 randomly selected undergraduate and postgraduate students in Nankai University, and uniformly use our NKS dataset for evaluation. As shown in the Figure 11, we have given the input image and the ground truth in every page. What the user needs to do is

TABLE IV

AVERAGE PSNR (DB), SSIM AND FSIM OF OUR PNLS WITH DIFFERENT PARAMETERS OVER THE NKS DATASET. WE CHANGE ONE PARAMETER AT A TIME TO ASSESS ITS INDIVIDUAL INFLUENCE ON OUR PNLS. “↑” MEANS THAT HIGHER IS BETTER.

	Value	11	13	15	17	Margin
<i>R</i>	PSNR↑	33.30	33.39	33.45	33.48	0.18
	SSIM↑	0.9358	0.9371	0.9378	0.9379	0.0021
	FSIM↑	0.9378	0.9392	0.9397	0.9396	0.0019
<i>s</i>	Value	1	2	3	4	Margin
	PSNR↑	33.68	33.65	33.59	33.45	0.23
	SSIM↑	0.9420	0.9415	0.9403	0.9378	0.0042
<i>m</i>	Value	4	5	6	7	Margin
	PSNR↑	33.45	33.41	33.32	33.27	0.18
	SSIM↑	0.9378	0.9391	0.9388	0.9382	0.0013
<i>N</i>	Value	5	10	15	20	Margin
	PSNR↑	33.31	33.45	33.08	32.68	0.14
	SSIM↑	0.9206	0.9378	0.9377	0.9355	0.0172
$\lambda$	Value	0.3	0.4	0.5	0.6	Margin
	PSNR↑	33.49	33.45	33.31	33.13	0.36
	SSIM↑	0.9361	0.9378	0.9378	0.9370	0.0017
	FSIM↑	0.9365	0.9397	0.9402	0.9396	0.0037

only to choose the smoothed image that, in his or her opinion, is closest to the ground truth (top right image). The voting results in the Figure 12 show that our PNLS has won the most choices (206 votes), better than the second one ResNet [57] (186 votes) and the third one L0 [51] (163 votes).

#### D. Ablation Study

Here, we conduct deeper examinations on how the parameters influence our PNLS. All experiments are performed on our NKS smoothing dataset. Our PNLS has 5 major parameters, the searching window size  $R = 15$ , the step  $s = 4$  of extracting reference patches, the patch size  $m = 4$ , the iteration number  $N = 10$ , the threshold  $\lambda = 0.4$ . We study the individual influence on our PNLS of each parameter while fixing the others. The PSNR, SSIM [40], and FSIM [55] results are summarized in Table III. We observe that, by increasing the searching field size  $\sqrt{r}$ , the performance of our PNLS increases because more patches could be matched in one time. Increasing the step  $s$  in our PNLS will decrease the number of reference patches to be processed, and thus naturally decrease the quantitative performance of our PNLS (but also speed up the running time). As the iteration number  $N$  increases, our PNLS performs better. Our PNLS is very robust to the separate change of  $m$  and  $\lambda$ .

#### VI. APPLICATIONS

We apply the proposed Pixel-level Non-Local Smoothing (PNLS) method on 4 image processing tasks: semantic region smoothing, image detail enhancement, image edge enhancement, and image abstraction.

**Semantic region smoothing** is mainly to smooth only the foreground or background region of an image while leaving the other region as is. In this task, we first predict the foreground mask that separates the foreground and background of the

image, and then smooth the specific area we are interested in. As shown in Figure 13, the foreground (upper row) or background (lower row) is smoothed while the background region is left as it is. Here, we use the salient ground truths as the corresponding foreground masks. In practice, we can employ the popular methods [8] to predict the salient region. **Image detail enhancement** aims at enhancing the details of an image while avoiding producing artifacts (gradient reversals or halos). For the input image, our PNLS decomposes it into a base layer (the smoothed image) and a detail layer (the removed textures). Then we enlarge the detail layer by 3 times while leaving the base layer as it is. The enhanced image is obtained by adding the enlarged detail layer back to the base layer. In Figure 14, we present the input images, the smoothed images by our PNLS, and the enhanced images on the images “0347” and “0484” from [57]. We observe that the images with enhanced details looks very natural when compared to the input images. This demonstrates that our PNLS well preserves the structure of input images while removing the details.

**Image edge enhancement** is a typical image processing task. As shown in Figure 15 (b), the boundary of the horse and the lawn are still very clear when compared to the input image (Figure 15 (a)). Here, we use a Laplacian operator and a Canny edge detector [6] to compute the gradient maps and edge maps of Figures 15 (a) and (b), respectively. The boundary in the gradient map (c) of the input image (a) is difficult to distinguish due to the interference of the textures in the surrounding area. The edge map (e) extracted by the Canny edge detector [6] is also seriously affected by the texture in (a). As shown in Figures 15 (d) and (f), our PNLS method smooths out unimportant textures (please refer to the textures in horse and the lawn in Figure 15 (b)). Thus, the Laplacian operator and the Canny edge detector [6] are enabled to extract clear gradient map (Figure 15 (e)) and reliable edge map (Figure 15 (f)), respectively, of the smoothed image (Figure 15 (b)).

**Image abstraction.** Our PNLS method can also be applied into the image abstraction task. The visual results are shown in Figure 16. As suggested by [43], we perform image abstraction by replacing bilateral filtering [29] with our PNLS, and extract cartoon-style abstractions (c) and (d) of the input images (a) and (b), respectively. In addition, we also used the method of [27] to generate pencil sketching results (e) and (f) of the abstract images (c) and (d), respectively. One can see that our PNLS can help obtain promising image abstraction and pencil sketching results, due to its capability of capturing accurate structure of images.

#### VII. CONCLUSION

Despite the progress of algorithms, proper benchmark datasets in image smoothing community are in urgent requirements. In this paper, we constructed a Nankai Smoothing (NKS) dataset to affiliate the comparison of image smoothing algorithms. With our NKS dataset, we benchmark 14 popular image smoothing algorithms and found that deep filters are not necessarily more effective than traditional (local or global) filters. For fine-grained image smoothing, we also proposed a Pixel-level Non-Local Smoothing (PNLS) method, by utilizing

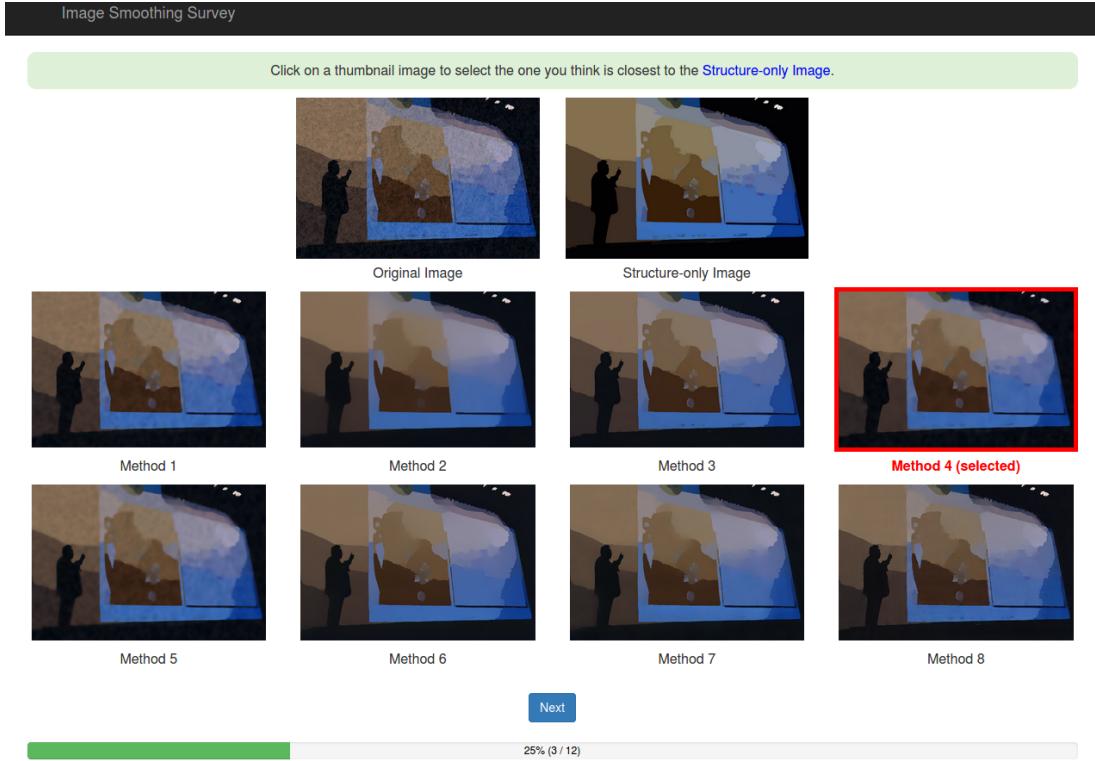


Fig. 11. **Interface of the system used in our user study experiments.** Each user has 12 rounds of voting. During each round of voting, the user selects one of the images smoothed by eight different methods that he/she thinks is closest to the structure image. The user only needs to click the selected image and then click the next button to proceed to the next round of voting.

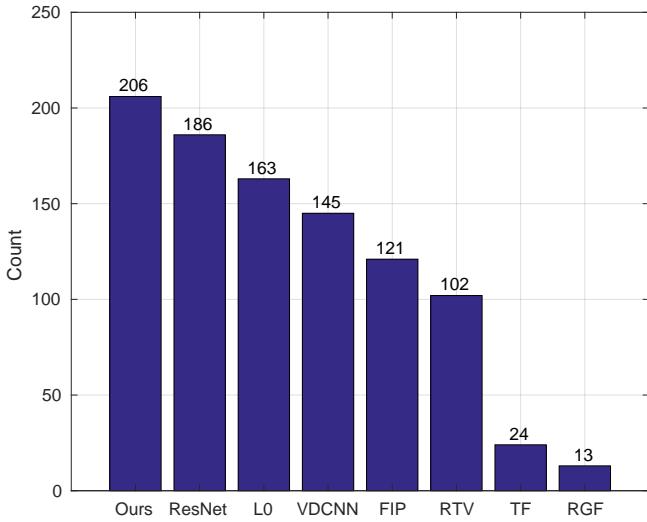


Fig. 12. Histogram of votes by 80 users for different methods.

the pixel-level non-local self similarity prior of natural images. Our PNLS achieved better qualitative and (or) quantitative performance than the other competing methods on several benchmark datasets (including our NKS dataset). Extensive parameter analysis validated the robustness of our PNLS on image smoothing. We further validated the broad practicality of the proposed PNLS method on the tasks of semantic region smoothing, detail/edge enhancement, and image abstraction.

This work can be extended in at least two directions.

First, we can further speed up our PNLS method to fulfill practical applications. Second, we can construct a larger image smoothing dataset, consisting of training and test images, to better benchmark deep learning based smoothing networks.

## REFERENCES

- [1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. [2](#), [8](#)
- [2] J. Antkowiak, T. Jamal Baina, F. V. Baroncini, N. Chateau, F. France-Telecom, A. C. F. Pessoa, F. Stephanie Colonnese, I. L. Contin, J. Caviedes, and F. Philips. Final report from the video quality experts group on the validation of objective models of video quality assessment march 2000, 2000. [3](#)
- [3] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. *IEEE Transactions on Image Processing*, 23(2):555–569, 2013. [5](#), [6](#), [8](#), [9](#)
- [4] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 60–65, 2005. [2](#)
- [5] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. [2](#)
- [6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. [10](#)
- [7] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017. [1](#), [2](#), [5](#), [6](#), [8](#), [9](#)
- [8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015. [10](#)
- [9] Clker-Free-Vector-Images. Pixabay. <https://pixabay.com/users/clker-free-vector-images-3736/>. Accessed: 2019-11-02. [4](#)



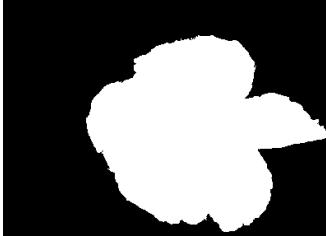
(a) Input Image



(a) Input Image



(b) Input Image



(c) Mask



(d) Mask



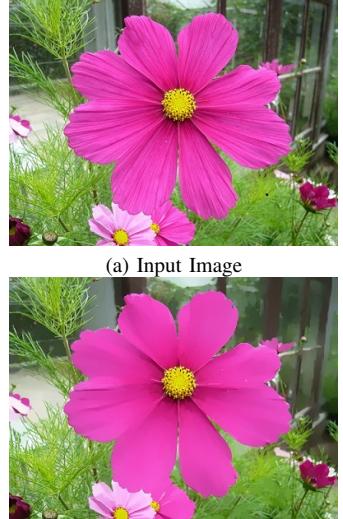
(e) Foreground Smoothing



(f) Background Smoothing

Fig. 13. Semantic region smoothing by our PNLS method of two images from the MSRA-B SOD dataset [39].

- [10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. [2](#), [6](#)
- [11] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998. [6](#)
- [12] Q. Fan, J. Yang, D. Wipf, B. Chen, and X. Tong. Image smoothing via unsupervised learning. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA 2018)*, 37(6), 2018. [2](#)
- [13] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics*, 27(3):67, 2008. [1](#), [2](#), [5](#), [6](#), [8](#)
- [14] R. Fattal. Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics*, 28(3):22, 2009. [5](#), [6](#), [8](#)
- [15] R. Fattal, M. Agrawala, and S. Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph.*, 26(3), July 2007. [2](#)
- [16] R. G. Gavaskar and K. N. Chaudhury. Fast adaptive bilateral filtering. *IEEE Transactions on Image Processing*, 28(2):779–790, 2018. [5](#), [6](#), [8](#)
- [17] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, Sep 1910. [6](#), [7](#)
- [18] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. [1](#), [2](#), [5](#), [6](#), [8](#)
- [19] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu, and L. Shao. Nlh: A blind pixel-level non-local method for real-world image denoising, 2019. [2](#)
- [20] C. Li and A. C. Bovik. Three-component weighted structural similarity index. In *Image quality and system performance VI*, volume 7242, page 7242Q. International Society for Optics and Photonics, 2009. [3](#)
- [21] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep joint image filtering. In *The European Conference on Computer Vision (ECCV)*, pages 154–169. Springer, 2016. [1](#), [2](#)
- [22] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang. A hybrid l1-l0 layer decomposition model for tone mapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [1](#)
- [23] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. [2](#)



(a) Input Image



(b) Input Image



(c) Smoothed Image by Our PNLS



(d) Smoothed Image by Our PNLS

Fig. 14. Illustration of image detail enhancement implemented by our PNLS smoothing method.

- [24] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 901–908. IEEE, 2006. [2](#)
- [25] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *The European Conference on Computer Vision (ECCV)*, pages 560–576. Springer, 2016. [1](#), [2](#), [5](#), [6](#), [8](#)
- [26] W. Liu, X. Chen, C. Shen, Z. Liu, and J. Yang. Semi-global weighted least squares in image filtering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5861–5869, 2017. [2](#)
- [27] C. Lu, L. Xu, and J. Jia. Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, pages 65–73. Eurographics Association, 2012. [10](#)
- [28] K. Lu, S. You, and N. Barnes. Deep texture and structure aware filtering network for image smoothing. In *The European Conference on Computer Vision (ECCV)*, September 2018. [1](#), [2](#), [4](#)
- [29] R. Manduchi and C. Tomasi. Bilateral filtering for gray and color images. In *The IEEE International Conference on Computer Vision (ICCV)*, 1998. [1](#), [2](#), [5](#), [6](#), [8](#), [10](#)
- [30] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. [2](#)
- [31] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing*, 23(12):5638–5653, 2014. [1](#), [5](#), [6](#), [8](#)
- [32] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004. [1](#), [2](#)
- [33] D. Ren, W. Zuo, D. Zhang, J. Xu, and L. Zhang. Partial deconvolution with inaccurate blur kernel. *IEEE Transactions on Image Processing*, 27(1):511–524, 2018. [2](#)
- [34] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE transactions on image processing*, 18(11):2385–2401, 2009. [3](#)
- [35] X. Shen, Y.-C. Chen, X. Tao, and J. Jia. Convolutional neural pyramid for image processing. *arXiv preprint arXiv:1704.02071*, 2017. [2](#)
- [36] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*,

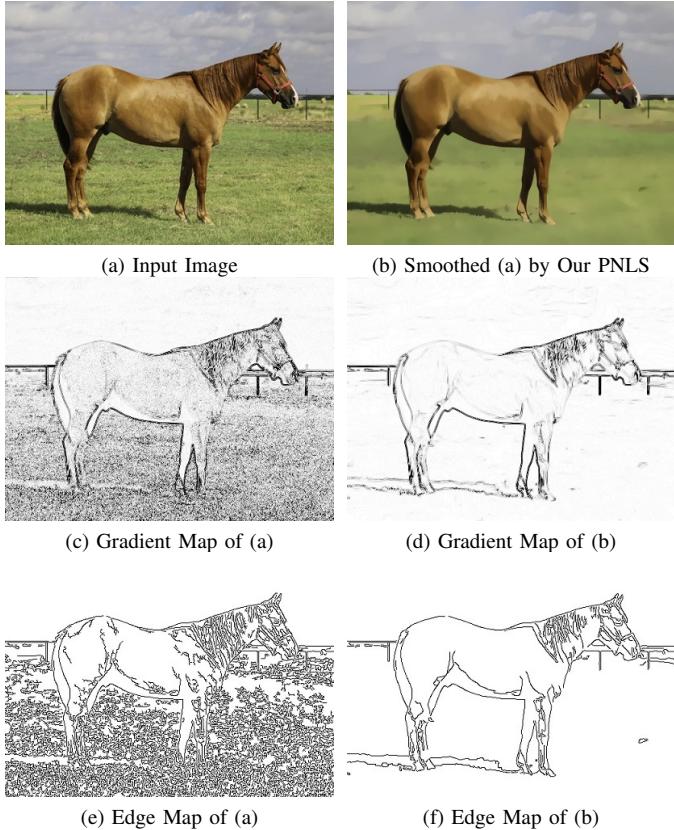


Fig. 15. Illustration of edge enhancement and extraction results. Our PNLS smoothing method suppresses the texture details, and strengthen structural edges of the input image (a).

- 3(2):186 – 200, 1996. 6
- [37] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin. Deep learning on image denoising: An overview. *arXiv preprint arXiv:1912.13171*, 2019. 4
- [38] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu. Attention-guided cnn for image denoising. *Neural Networks*, 2020. 4
- [39] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng. Salient object detection: A discriminative regional feature integration approach. *International Journal of Computer Vision*, 123(2):251–268, 2017. 12
- [40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 3, 4, 5, 6, 8, 10
- [41] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirtieth Asilomar Conference on Signals, Systems & Computers*, 2003, volume 2, pages 1398–1402. Ieee, 2003. 3
- [42] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao. Ranet: Ranking attention network for fast video object segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 2
- [43] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Trans. Graph.*, 25(3):1221–1226, July 2006. 2, 10
- [44] J. Xu, Y. Hou, D. Ren, L. Liu, F. Zhu, M. Yu, H. Wang, and L. Shao. Star: A structure and texture aware retinex model. *IEEE Transactions on Image Processing*, 2020. 1
- [45] J. Xu, Y. Huang, M.-M. Cheng, L. Liu, F. Zhu, X. Hou, and L. Shao. Noisy-as-clean: Learning unsupervised denoising from the corrupted image, 2019. 4
- [46] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv:1804.02603*, 2018. 4
- [47] J. Xu, L. Zhang, and D. Zhang. External prior guided internal prior learning for real-world noisy image denoising. *IEEE Transactions on Image Processing*, 27(6):2996–3010, June 2018. 4
- [48] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *The European Conference on Computer Vision (ECCV)*, 2018. 4
- [49] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *The IEEE*

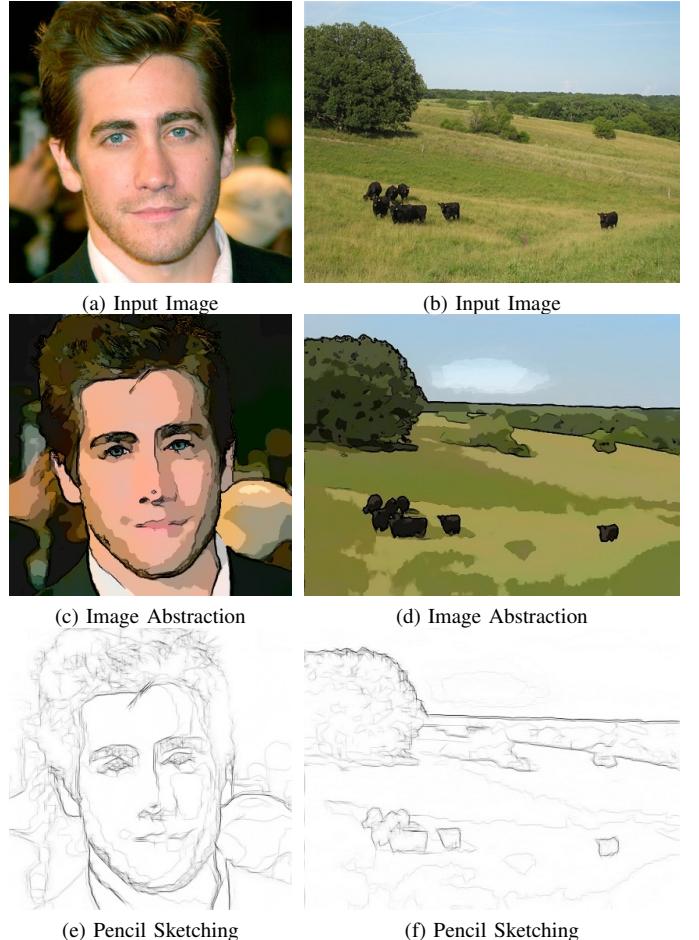


Fig. 16. Illustration of image abstraction and pencil sketching results, in which our PNLS method removes the texture details.

- International Conference on Computer Vision (ICCV)*, 2017. 4
- [50] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 244–252, 2015. 4
- [51] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via  $l_0$  gradient minimization. *ACM Transactions on Graphics*, 30(6):174, 2011. 1, 2, 5, 6, 8, 9, 10
- [52] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *International Conference on Machine Learning*, pages 1669–1678, 2015. 1, 2
- [53] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Transaction on Graphics*, 31(6):139:1–139:10, 2012. 1, 2, 5, 6, 8, 9
- [54] F. Zhang, L. Dai, S. Xiang, and X. Zhang. Segment graph based image filtering: Fast structure-preserving smoothing. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 361–369, 2015. 1
- [55] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 3, 4, 5, 6, 8, 10
- [56] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *ECCV*, pages 815–830, 2014. 2, 5, 6, 8, 9
- [57] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu. A benchmark for edge-preserving image smoothing. *IEEE Transactions on Image Processing*, 28(7):3556–3570, 2019. 1, 2, 4, 5, 6, 8, 9, 10