Signal Processing 157 (2019) 62–72

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Local activity-driven structural-preserving filtering for noise removal and image smoothing

Lijun Zhao[a], Huihui Bai[a,*], Jie Liang[b], Anhong Wang[c], Bing Zeng[d], Yao Zhao[a]

[a] Beijing Key Laboratory of Advanced Information Science and Network Technology, and Institute of Information Science, Beijing Jiaotong University, No.3 Shangyuancun, China
[b] School of Engineering Science, Simon Fraser University, Burnaby, Canada
[c] Institute of Digital Media and Communication, Taiyuan University of Science and Technology, China
[d] Institute of Image Processing, University of Electronic Science and Technology of China, China

ARTICLE INFO

Article history:
Received 23 June 2018
Revised 12 November 2018
Accepted 14 November 2018
Available online 15 November 2018

Keywords:
Image filtering
Noise removal
Image smoothing
Scale representation

ABSTRACT

In this paper, a local activity measurement of the clipped and normalized variance or standard deviation is proposed to drive anisotropic diffusion and relative total variation (RTV) to work better for structural preservation. Firstly, two novel edge-stop functions are introduced for our local activity-driven anisotropic diffusion (LAD-AD) to efficiently remove severe artifacts and preserve the fine geometry structures in HEVC-compressed depth images. Secondly, we propose a simple yet effective local activity-driven RTV (LAD-RTV) with the way of the product between gradient and the local activity measurement for image smoothing and scale representation. Meanwhile, both color-sharing information and each-channel discriminative information are considered, which are significant to color image edge-preserving but not included in the RTV model. Besides, LAD-RTV leverages the form of the division of gradient and the local activity measurement to resolve the problem of general image de-noising by regarding the noises as the duplicate texture elements. Experimental results have validated that the proposed LAD-AD can greatly improve the precision of the HEVC-compressed depth image and the quality of its synthesized image. Additionally, large numbers of results have shown our LAD-RTV is superior to several existing methods.

1. Introduction

Image filtering is an effective way to improve the performance of many applications, such as edge detection and image editing [1–9]. Since different types of images have different characteristics and different applications have different requirements, image filtering algorithms should be designed for each case properly. For example, depth images having smooth regions divided by sharp boundaries represent scene's geometry structures. The high-quality boundaries should be preserved after depth image filtering, because they will strongly affect 3D video coding efficiency and the quality of view synthesis with depth image-based rendering (DIBR). Therefore, the quality of the virtual-view images should be enhanced after filtering contaminated depth images. Meanwhile, the precision of depth image should be kept at least or even be greatly improved. For natural images, when we want to remove image noises, we need to preserve both image's structures and textural details at the same time. If we want to apply image smoothing, we should remove texture details but keep major structures.

Although there are a large number of works for image filtering [5–14], most of these algorithms tend to be computationally complex, which are not well suitable for practical applications. Meanwhile, their algorithms are specifically designed for one model, which lose the sight of generalization, so we need to re-design a new algorithm for each new model. Based on the above observations, our motivation is whether a robust statistic measurement can be easily inserted into some models to adaptively control model's trade-off parameter between data term and regularization term. Meanwhile, this statistic measurement should not significantly increase computational complexity. Besides, this measurement can be easily put into most of low-level image processing model without complicated expert design. It is generally known that standard deviation is a good measurement on the degree of dispersion for a set of data. Because each image patch's standard deviation can be quickly computed through matrix operations, it will not significantly increase the complexity of the filtering. Consequently, we introduce a local activity measurement of variance or standard deviation to drive different models for better solutions.

[*] Corresponding author.
E-mail address: hhbai@bjtu.edu.cn (H. Bai).

https://doi.org/10.1016/j.sigpro.2018.11.012

In this paper, a clipped and normalized local variance or standard deviation is leveraged as the local activity measurement for image smoothing and denoising. In particular, the ratio between gradient and the clipped local activity could locate the noises and facilitate image denoising. In our first framework, we develop a robust local activity-driven anisotropic diffusion framework (LAD-AD) and apply it for HEVC-compression artifact removal of the contaminated piece-wise smooth images such as compressed depth images. More importantly, there are several issues to be noticed for LAD-AD. For example, the clip function plays a key role in controlling image diffusion of LAD-AD. Meanwhile, choosing local variance or standard deviation results in different edge-stop function formations. Furthermore, the updated way of the local activity measurement has great impacts on the algorithm's performances.

Our second framework is a local activity-driven relative total variation (LAD-RTV), which not only uses the local activity but also takes the color-sharing information and each-channel discriminative information into consideration. There are two schemes for our LAD-RTV. The first scheme is a local activity-driven RTV for image smoothing and image representation in different scale-spaces, where the RTV is divided by the clipped local activity, which emphasizes image salient contours. Additionally, the color-sharing information and each-channel discriminative information used in our LAD-RTV can provide more discriminative information than RTV. The second scheme of LAD-RTV is designed to remove additive white Gaussian noises by treating image noises as duplicate textures when using a ratio between gradient and the local activity to identify the location and the amplitude of the noises.

The rest of this paper is organized as follows. Firstly, some works on image denoising and image smoothing are reviewed in Section 2. Secondly, we introduce a robust local activity-driven anisotropic diffusion and a local activity-driven relative total variation in Section 3 and 4 respectively. After that, experimental results are presented in Section 5. Finally, we conclude our paper in Section 6.

2. Related works

Because our work involves two popular low-level image processing problems, we first give a review of general image denoising and image smoothing respectively. After that, we look back on a special class of image denoising for compressed depth image.

2.1. General image denoising

Image denoising is often separated into two categories: weighted filtering and optimization-based methods. For the first class, there are several well-known algorithms, such as bilateral filtering and its extensions. For example, bilateral filtering is a general image filtering technique [15], which can remove image noises and preserve sharp boundaries. Due to its high computational costs, fast bilateral filtering is developed to accelerate image filtering in [16]. For visual-pleasing image denoising, an optimally weighted bilateral filter is formed by minimizing the oracle mean-squared-error to get optimal weights [17], whose performance is competitive to the non-local means filter [18]. Recently, image local entropy is used to automatically direct filter's range parameter selections of bilateral filtering [19]. Based on the density of connected components, an image activity detector is built up for a faster noise removal filtering [20], as compared to a classic median filtering.

Anisotropic diffusion [21] belongs to the second class of image denoising algorithm. In [13], the relationship between anisotropic diffusion and robust statistics is analyzed. In [14], a new class of fractional-order anisotropic diffusion equations is introduced for noise removal. To preserve edges and fine details, both local gradient and variance are incorporated into the diffusion model to remove annoying noises effectively [10,11]. Most recently, a regularization model is leveraged to adaptively adjust the diffusivity based on the image gradient magnitude [22]. It is well-known that the total variation model can be viewed as a special case of the anisotropic diffusion with specific edge-stop function. Next, we review several works about total variation model. In a spatially adaptive total variation model, image denoising strength is differentially assigned to district regions and different bands [23]. In [24,25], spatially clustering-aware total variation is used, or total variation's weighted parameter is controlled by spatial difference curvature to resolve image super-resolution problem.

Next, we will look back to medical image denoising, which is always used as a pre-processing for computer-aided diagnosis system. Recently, CT image denoising and MRI image denoising have been reviewed in [26,27]. Here, several newest works will be discussed. Because there exist information redundancies for MR images in PCA domain, only noise principle component is removed, while other components are kept to improve the signal-to-noise ratio [28]. To accurately diagnose disease from MRI images, a group of filters is merged with an image segmentation technique to reduce image noises [29]. Meanwhile, image grid segmentation is utilized to automatically choose the filtering parameters for semi-classical signal analysis-based denoising [30]. Because low-dose CT images often incur annoying noises and artifacts, both generative adversarial loss and perceptual loss are introduced into the training of CNN-based denoising model [31].

2.2. Image smoothing

Image smoothing, also known as texture removal, is another important technique for many low-level and high-level applications [2,32,33], because this technique provides a lot of meaningful structural boundary clues. The family of weighted filtering methods is often achieved by a weighting method within small or large patches. For example, the guided image filter is a fast and non-approximate linear time algorithm [4]. Another efficient method is rolling guidance filtering [7], which is a fast iterative bilateral method. In order to achieve real-time tasks, domain transform is used for a high-quality edge preserving filtering [5].

Different from image weighted filtering for image smoothing, the family of optimization-based methods always faces a non-convex yet complex problem. In [8], both static guidance and dynamic guidance are jointly leveraged to achieve robust guided image filtering, which is formulated as a non-convex optimization problem. In [3], a multi-scale image decomposition method is conducted to form the edge-preserving smoothing operator in a weighted least square optimization framework. In [2], an L0 gradient minimization optimization framework globally controls how many non-zero gradients are kept in the smoothed image for textural removal. By taking advantage of a statistic diversity of gradient between texture patches and structure patches, a relative total variation (RTV) framework is presented to preserve image structures [1]. In this method, windowed inherent variation (WIV) and windowed total variation (WTV) are combined to discriminate structures from textures, while an optimization problem is formulated to extract image main structure. Afterward, another efficient approach tries to use region covariance for image smoothing [6]. Recently, mutually guided image filter (muGIF) [34] defines a new measurement for mutual response to manage structural similarity between two input images for image smoothing and scale-space filtering, etc. Although these methods achieve some excellent performances for structure-preserving smoothing, there are still some thorny problems, such as inefficient texture removal, severe boundary blurring, and the inaccuracy of edge localization after filtering.

2.3. Depth image denoising

Although depth image filtering is similar to image processing problems described above, there some differences between them. On the one hand, depth image denoising should keep surface smoothness within objects to satisfy depth image piece-wise smooth characteristics, whose ambition is partly similar to textural removal. On the other hand, depth denoising algorithms should also preserve depth image fine details like general image denoising. Many methods have been explored to remove severe artifacts in the compressed depth images so that the quality of the synthesized virtual image could be improved. In [35], a trilateral filtering method is treated as an in-loop filter to reduce depth coding artifacts. This method employs three filtering weights, which respectively come from the spatial domain, depth range domain, and color range domain. In [36], an adaptive depth truncation filter (ADTF) is presented to restore sharp object boundaries of depth images from blurring. In [37], a candidate-value-based depth boundary filtering is developed by selecting an appropriate candidate value to replace each unreliable pixel based on spatial correlation and statistical characteristics. In [38], two-stage filtering (TSF) scheme is presented to reduce depth coding artifacts by using binary segmentation-based depth filtering and Markov Random Field (MRF). Lately, an iterative range-domain weighted filtering (IRWF) [9] is used to improve the quality of compressed depth images without the use of spatial domain weights by iteratively filtering in the range domain. These methods greatly reduce image artifacts of synthesized virtual images caused by compressed depth image, but they often tend to change depth images too much. Up to now, simultaneously improving the accuracy of the depth image and its synthesized image quality by filtering compressed depth image is still a challenging problem and should be further studied.

3. Local activity-driven anisotropic diffusion

Generally, depth images are characterized with piece-wise smooth regions segmented by sharp boundaries. However, depth boundaries usually suffer from various compression artifacts after compression, which will badly affect the quality of view synthesis [35–38]. In this paper, we propose a local activity-driven anisotropic diffusion (LAD-AD) method to mitigate coding artifacts of depth images, which is written as:

$$\frac{\partial \boldsymbol{I}}{\partial t} = \nabla \cdot (c(||\nabla \boldsymbol{I}||, \boldsymbol{K})\nabla \boldsymbol{I}), \tag{1}$$

where $\nabla \boldsymbol{I}$ is the gradient of image, $\nabla \cdot$ is the divergence operator, and $\boldsymbol{K}$ is the local activity image got from a HEVC-compressed depth image $\boldsymbol{I}$. The discrete solution of Eq. (1) can be written as:

$$\boldsymbol{I}_i^{t+1} = \boldsymbol{I}_i^t + \lambda \sum_{j \in \mathbb{N}_i} c(||\nabla \boldsymbol{I}_{ij}^t||, \boldsymbol{K}_i^t)\nabla \boldsymbol{I}_{ij}^t, \tag{2}$$

where $\boldsymbol{I}_i^0 = \boldsymbol{I}_i$ in the first iteration, and $\boldsymbol{K}_i^t$ is a clipped and normalized local activity for pixel i at the t-th iteration, which will be defined later. Two novel local activity-based edge-stop functions are defined as follows:

$$c(||\nabla \boldsymbol{I}_{ij}^t||, \boldsymbol{K}_j^t) = exp\left(-\left(\frac{||\nabla \boldsymbol{I}_{ij}^t||}{\rho_1 \boldsymbol{K}_i^t}\right)^2\right), \tag{3}$$

$$c(||\nabla \boldsymbol{I}_{ij}^t||, \boldsymbol{K}_j^t) = exp\left(-\left(\frac{||\nabla \boldsymbol{I}_{ij}^t||^2}{(\rho_2)^2 \boldsymbol{K}_i^t}\right)\right), \tag{4}$$

where ρ_1 and ρ_2 are the diffusion parameters, e.g., $\rho_1 = 30$ and $\rho_2 = \sqrt{300}$ in the default setting. Note that $\boldsymbol{K}_i^t$ is squared in Eq. (3), but not in Eq. (4). The ratio of gradient and the local activity is used to capture where the coding artifacts exist in the compressed depth image. Moreover, the diffusion parameters are adaptively adjusted according to this ratio, i.e., larger diffusion parameters are assigned to more severely distorted pixels. Therefore, pixels with larger local activity would receive more anisotropic diffusion from neighboring pixels than pixels with smaller activity under the constraint of the edge-stop function. This contributes to removing noisy pixels and preventing blurry regions from being heavily diffused.

Next, we will introduce a clipped and normalized local activity measurement $\boldsymbol{K}_i^t$. First, we calculate a local mean value $\bar{\boldsymbol{I}}_i^t$ and standard deviation $\boldsymbol{v}_i^t$ of the 8-connected neighborhood around each pixel, which is written in Eq. (5). Here, $\mathbb{N}_i$ denotes the 8-connected neighborhood of pixel i. Note that our anisotropic diffusion uses 4-connected neighborhood in the regularization term for compressed depth image filtering, so only 3x3 window size is considered to get the local activity rather than other larger window size.

$$\boldsymbol{v}_i^t = [\frac{1}{9}((\boldsymbol{I}_i^t - \bar{\boldsymbol{I}}_i^t)^2 + \sum_{j \in \mathbb{N}_i}(\boldsymbol{I}_j^t - \bar{\boldsymbol{I}}_i^t)^2)]^{\frac{1}{2}}, \bar{\boldsymbol{I}}_i^t = \frac{1}{9}(\boldsymbol{I}_i^t + \sum_{j \in \mathbb{N}_i}\boldsymbol{I}_j^t) \tag{5}$$

A clipped version of $\boldsymbol{v}_i^t$, denoted as $\boldsymbol{V}_i^t$, is given as: $\boldsymbol{V}_i^t = t_l$, if $0 \leq \boldsymbol{v}_i^t < t_l$; $\boldsymbol{V}_i^t = \boldsymbol{v}_i^t$, if $t_l \leq \boldsymbol{v}_i^t < t_h$; $\boldsymbol{V}_i^t = t_h$, if $t_h \leq \boldsymbol{v}_i^t$, which is the clip function. Here, t_l and t_h are two pre-defined truncated parameters of the clip function to control the degree of diffusion by making anisotropic diffusion to work within a certain range of local activity, e.g., $t_l = 1$, $t_h = 30$. t_l is the minimum truncated value, while t_h is the maximum truncated value. This clip function restricts the local activity not to be large or small, which makes each pixel have a valid local activity measurement. If we use variance as local activity measurement to adjust image diffusion strength, too large distance between the largest variance ν_a^2 and the smallest variance ν_b^2 will make some pixels' diffusion not to work or heavily diffused. To see this, we first use a fact that $(\nu_a^2 - \nu_b^2)$ is $(\nu_a + \nu_b)$ times of $(\nu_a - \nu_b)$. When $\nu_a + \nu_b > 1$, $(\nu_a^2 - \nu_b^2)$ is more than one times of $(\nu_a - \nu_b)$. $(\nu_a^2 - \nu_b^2) = (\nu_a - \nu_b)$ if $\nu_a + \nu_b = 1$. $(\nu_a^2 - \nu_b^2)$ is less than $(\nu_a - \nu_b)$ if $\nu_a + \nu_b < 1$. Generally, the distance between ν_a and ν_b is far larger than 1, so $(\nu_a^2 - \nu_b^2)$ is immensely larger than $(\nu_a - \nu_b)$, so we choose the standard deviation as the local activity to driven image filtering. After calculating the local activity, $\boldsymbol{V}_i^t$ is normalized by $max(\boldsymbol{V}^t)$, which is the maximal value across the image in Eq. (6).

$$\bar{\boldsymbol{V}}_i^t = \boldsymbol{V}_i^t / max(\boldsymbol{V}^t), 0 \leq t \leq (m-1) \tag{6}$$

Finally, $\boldsymbol{K}_i^t$ is updated to make iterative results more stable from $\bar{\boldsymbol{V}}_i^t$ for every l iterations, which is defined as: $\boldsymbol{K}_i^t = \bar{\boldsymbol{V}}_i^t$, if $mod(t, l) = 0$; $\boldsymbol{K}_i^t = \bar{\boldsymbol{V}}_i^{t-mod(t,l)}$, if $mod(t, l) \neq 0$. Here, mod denotes the modulo operator, m be the maximal number of iterations, and the updating interval l is chosen as $l \in [1, m]$. In the following, the fixed local activity-driven anisotropic diffusion using Eq. (3) as edge-stop function is denoted as FLAD-AD. And the time-updated local activity-driven anisotropic diffusion with Eq. (3) is labelled as TLAD-AD. Meanwhile, the periodically local activity-driven anisotropic diffusion based on edge-stop function of Eq. (3) is referred to as PLAD-AD. Similarly, when Eq. (4) is used, three other methods are denoted respectively as FLAD-AD (I), TLAD-AD (I), and PLAD-AD (I). When l is set to be 1, the proposed method becomes TLAD-AD. If l is larger than 1, but small than m, it reduces to PLAD-AD. However, if l is set to be m, it is changed into FLAD-AD.

There are three previous works [10–12], which are similar to the proposed method, but they are different from each other in some aspects. Several discrepancies between our LAD-AD and [10,11] are listed as follows: we introduce a clipped function to restrict the value of local activity; the local activity is calculated by the interval-updated way; and our method uses the division between gradient and local activity, but the works of Chao and coworkers [10,11] employ the multiplication way. The discrepancies between the proposed LAD-AD and [12] are listed as follows:

1. In this paper, the clipped and normalized local activity in a periodically updated way is used to drive the anisotropic diffusion adaptively. The detailed operation of activity used in [12] is very complex, and the window for their activity is often set to be larger than 3×3. In this paper, we aim to achieve fast depth filtering for distorted image compressed by HEVC coder, so we use 3×3 window centered at pixel D_i to get the 8-connected standard deviation $\boldsymbol{v}_i$ instead of the variance. If the variance is used, a small variance can be easily dominated by the large variance, which leads to a little contribution to the diffusion. Another reason is that our LAD-AD uses the 4-connected neighborhood for anisotropic diffusion so that both standard deviation and the local gradient is calculated according to the same neighboring regions. But, the standard deviation within the 8-connected neighborhood is chosen to get the local activity rather than within 4-connected neighborhood in order to make LAD-AD robust to image noises.[1]
2. A clipped function is used for the local activity to make diffusion stable during anisotropic diffusion, because pixels with very large local activity always make local activity-driven anisotropic diffusion useless for pixels with small local activity measurements.
3. During the iterative diffusion, the updated activity is used to control the degree of diffusion. Generally, the fixed local activity often tends to blur image discontinuities. The time-updated local activity can always preserve the sharp boundaries, but it often requires extra calculation of the local activity in every iteration. The interval-updated activity is a good alternative, especially when some practical applications require fast filtering.

4. Local activity-driven relative total variation

Inspired by the literature of Xu et al. [1], we propose a local activity-driven relative total variation for image smoothing (denoted as LAD-RTVs) for color images. Given a color image $\boldsymbol{I}_0^c = [\boldsymbol{R}_0, \boldsymbol{G}_0, \boldsymbol{B}_0]$, our LAD-RTVs is written in Eq. (7), where $\boldsymbol{I}^c$ denote the smoothed color image, which has three channels $[\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}]$. The former term in Eq. (7) is data term, and the latter one is our LAD-RTVs regularization term. These two terms are balanced by λ, which controls the degree of smoothness for the solution $\boldsymbol{I}^c$ of our LAD-RTVs. The clipped and normalized local activity measurement $\boldsymbol{v}_p^r$ for red-channel is obtained according to Eq. (5) as well as Eq. (6). Then, $\boldsymbol{v}_p^g$ as well as $\boldsymbol{v}_p^b$ can be got like $\boldsymbol{v}_p^r$. The superscript of $\boldsymbol{v}_p^r$ indicates this local activity belongs to red channel, while its subscript denotes the position of p in the image. Other symbols can be labelled in this manner. $\boldsymbol{v}_p^c$ is the maximum value of the clipped and normalized local activity measurements along image channel dimension. The t_l and t_h for LAD-AD are denoted as th_l and th_h in the LAD-RTV model for the clear usage in the following. The default values of these two parameters for image smoothing are set to be $th_l = 1$ and $th_h = 10$.

$$
\begin{aligned}
\underset{\boldsymbol{R},\boldsymbol{G},\boldsymbol{B}}{\arg\min} & \Big\{ \sum_p (\boldsymbol{R}(p) - \boldsymbol{R}_0(p))^2 + \sum_p (\boldsymbol{G}(p) - \boldsymbol{G}_0(p))^2 \\
& + \sum_p (\boldsymbol{B}(p) - \boldsymbol{B}_0(p))^2 \Big\} \\
& + \lambda \cdot \Bigg\{ \sqrt{\sum_p \frac{\frac{(\mathcal{D}r)_x(p)}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\big(\frac{\mathcal{D}r)_x^c(p)}{(\mathcal{L}r)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} + \sqrt{\sum_p \frac{\frac{(\mathcal{D}r)_y(p)}{(\mathcal{L}r)_y(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\frac{(\mathcal{D}r)_y^c(p)}{(\mathcal{L}r)_y^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} \\
& + \sqrt{\sum_p \frac{\frac{(\mathcal{D}g)_x(p)}{(\mathcal{L}g)_x(p)+\epsilon}}{\boldsymbol{v}_p^g} \cdot \frac{\frac{(\mathcal{D}g)_x^c(p)}{(\mathcal{L}g)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} + \sqrt{\sum_p \frac{\frac{(\mathcal{D}g)_y(p)}{(\mathcal{L}g)_y(p)+\epsilon}}{\boldsymbol{v}_p^g} \cdot \frac{\frac{(\mathcal{D}g)_y^c(p)}{(\mathcal{L}g)_y^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} \\
& + \sqrt{\sum_p \frac{\frac{(\mathcal{D}b)_x(p)}{(\mathcal{L}b)_x(p)+\epsilon}}{\boldsymbol{v}_p^b} \cdot \frac{\frac{(\mathcal{D}b)_x^c(p)}{(\mathcal{L}b)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} + \sqrt{\sum_p \frac{\frac{(\mathcal{D}b)_y(p)}{(\mathcal{L}b)_y(p)+\epsilon}}{\boldsymbol{v}_p^b} \cdot \frac{\frac{(\mathcal{D}b)_y^c(p)}{(\mathcal{L}b)_y^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} \Bigg\}
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
(\mathcal{D}r)_x(p) \cdot (\mathcal{D}r)_x^c(p) &= \sum_{q\in\mathbb{N}_p} g_{p,q} |(\partial_x \boldsymbol{R})_q| \cdot \sum_{q\in\mathbb{N}_p} g_{p,q} | \sum_{k\in 1,2,3} (\partial_x \boldsymbol{I}^c(k))_q / 3|, \\
(\mathcal{D}r)_y(p) \cdot (\mathcal{D}r)_y^c(p) &= \sum_{q\in\mathbb{N}_p} g_{p,q} |(\partial_y \boldsymbol{R})_q| \cdot \sum_{q\in\mathbb{N}_p} g_{p,q} | \sum_{k\in 1,2,3} (\partial_y \boldsymbol{I}^c(k))_q / 3|
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
(\mathcal{L}r)_x(p) \cdot (\mathcal{L}r)_x^c(p) &= | \sum_{q\in\mathbb{N}_p} g_{p,q} (\partial_x \boldsymbol{R})_q| \cdot | \sum_{q\in\mathbb{N}_p} g_{p,q} \sum_{k\in 1,2,3} (\partial_x \boldsymbol{I}^c(k))_q / 3|, \\
(\mathcal{L}r)_y(p) \cdot (\mathcal{L}r)_y^c(p) &= | \sum_{q\in\mathbb{N}_p} g_{p,q} (\partial_y \boldsymbol{R})_q| \cdot | \sum_{q\in\mathbb{N}_p} g_{p,q} \sum_{k\in 1,2,3} (\partial_y \boldsymbol{I}^c(k))_q / 3|
\end{aligned}
\tag{9}
$$

Our discriminatively color windowed total variation (DCWTV) measures along each axis for red channel are respectively written in Eq. (8). Our discriminatively color windowed inherent variation (DCWIV) measures along each axis for red channel are respectively presented in Eq. (9). In most cases, pixels around edges have a higher value of local activity than in relative other non-edge regions, such as fine detail regions. By dividing both $\boldsymbol{v}_p^r / \boldsymbol{v}_p^g / \boldsymbol{v}_p^b$ and $\boldsymbol{v}_p^c$ in Eq. (7), the LAD-RTVs regularization term for the edge pixels becomes smaller than others, so these pixels will have fewer contributions to the LAD-RTVs term in relative to the non-edge regions, which results in that more edges will be preserved. As compared to RTV [1], the proposed LAD-RTVs in Eq. (7) will further smoothen the detailed textures, but image salient structures are left. Due to the non-convexity of Eq. (7), its solution cannot be directly obtained. As described in [1,39], an objective function with a quadratic term as the penalty can be optimized linearly. The LAD-RTVs term for red-channel can be decomposed into a quadratic part and a non-linear part. Thus, the square of LAD-RTVs term for red-channel in the x-direction can be re-written as:

$$
\begin{aligned}
& \sum_p \frac{\frac{(\mathcal{D}r)_x(p)}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\frac{(\mathcal{D}r)_x^c(p)}{(\mathcal{L}r)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c} \\
&= \sum_p \frac{\frac{\sum_{q\in\mathbb{N}_p} g_{p,q} \cdot |(\partial_x \boldsymbol{R})_p|}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\frac{\sum_{q\in\mathbb{N}_p} g_{p,q} \cdot \frac{|\sum_{k\in 1,2,3} (\partial_x \boldsymbol{I}^C(k))_p|}{3}}{(\mathcal{L}r)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c} \\
&= \sum_p \sum_{q\in\mathbb{N}_p} \frac{\frac{g_{p,q} \cdot |(\partial_x \boldsymbol{R})_p|}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\frac{g_{p,q} \cdot \frac{|\sum_{k\in 1,2,3} (\partial_x \boldsymbol{I}^C(k))_p|}{3}}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^c} \\
&\approx \sum_p \sum_{q\in\mathbb{N}_p} (g_{p,q})^2 \cdot \frac{1}{(\mathcal{L}r)_x(p)+\epsilon} \\
&\quad \cdot \frac{1}{|(\partial_x \boldsymbol{R})_p| + \epsilon} \cdot \frac{1}{\boldsymbol{v}_p^r} \cdot (\partial_x \boldsymbol{R})_p^2 \cdot \frac{1}{(\mathcal{L}r)_x^c(p)+\epsilon} \\
&\quad \cdot \frac{1}{\frac{|\sum_{k\in 1,2,3} (\partial_x \boldsymbol{I}^c(k))_p|}{3} + \epsilon} \cdot \frac{1}{\boldsymbol{v}_p^c} \cdot (\partial_x \boldsymbol{R})_p^2,
\end{aligned}
\tag{10}
$$

$$
g_{p,q} = exp(-\frac{(x_p - x_q^2) + (y_p - y_q)^2}{2\sigma^2}),
\tag{11}
$$

where $g_{p,q}$ is a Gaussian weighting function with variance $\sigma = 3$, and (x_p, y_p) is the location of image pixel p. This equation can be

[1] More discussion can be found from https://github.com/mdcnn/Local-Activity-Driven-Filtering.

further rewritten as:

$$\sqrt{\sum_p \frac{\frac{(\mathcal{D}r)_x(p)}{(\mathcal{L}r)_x(p)+\epsilon}}{\boldsymbol{v}_p^r} \cdot \frac{\frac{(\mathcal{D}r)_x^c(p)}{(\mathcal{L}r)_x^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} \approx \sum_p \boldsymbol{s}_{x,p}^r \cdot \boldsymbol{c}_p^r \cdot (\partial_x \boldsymbol{R})_p^2, \tag{12}$$

where

$$\boldsymbol{c}_p^r = \frac{1}{\sqrt{\boldsymbol{v}_p^r * \boldsymbol{v}_p^c}}, \tag{13}$$

$$\boldsymbol{s}_{x,p}^r = \sum_{q \in \mathbb{N}_p} g_{p,q} \cdot \frac{1}{\sqrt{((\mathcal{L}r)_x(p)+\epsilon) \cdot (|(\partial_x \boldsymbol{R})_p| + \epsilon)}} \cdot \frac{1}{\sqrt{((\mathcal{L}r)_x^c(p)+\epsilon) \cdot (\frac{|\sum_{k \in 1,2,3} (\partial_x \boldsymbol{I}^c(k))_p|}{3} + \epsilon)}}. \tag{14}$$

Similarly, the LAD-RTVs term for red-channel in the y-direction can be written as:

$$\sqrt{\sum_p \frac{\frac{(\mathcal{D}r)_y(p)}{(\mathcal{L}r)_y(p)+\epsilon}}{\boldsymbol{v}_p} \cdot \frac{\frac{(\mathcal{D}r)_y^c(p)}{(\mathcal{L}r)_y^c(p)+\epsilon}}{\boldsymbol{v}_p^c}} \approx \sum_p \boldsymbol{s}_{y,p}^r \cdot \boldsymbol{c}_p^r \cdot (\partial_y \boldsymbol{R})_p^2, \tag{15}$$

where

$$\boldsymbol{s}_{y,p}^r = \sum_{q \in \mathbb{N}_p} g_{p,q} \cdot \frac{1}{\sqrt{((\mathcal{L}r)_y(p)+\epsilon) \cdot (|(\partial_y \boldsymbol{R})_p| + \epsilon)}} \cdot \frac{1}{\sqrt{((\mathcal{L}r)_y^c(p)+\epsilon) \cdot (\frac{|\sum_{k \in 1,2,3} (\partial_y \boldsymbol{I}^c(k))_p|}{3} + \epsilon)}}. \tag{16}$$

The LAD-RTVs term for green-channel and blue-channel can be defined similarly. Finally, we re-write Eq. (7) in the form of matrix in the first iteration as follows:

$$\begin{aligned}
\mathop{\arg\min}_{\boldsymbol{R}(1),\boldsymbol{G}(1),\boldsymbol{B}(1)} \ & (\boldsymbol{V_R}(1) - \boldsymbol{V_R})^T (\boldsymbol{V_R}(1) - \boldsymbol{V_R}) \\
& + (\boldsymbol{V_G}(1) - \boldsymbol{V_G})^T (\boldsymbol{V_G}(1) - \boldsymbol{V_G}) + (\boldsymbol{V_B}(1) - \boldsymbol{V_B})^T (\boldsymbol{V_B}(1) - \boldsymbol{V_B}) \\
& + \lambda \Big[(\boldsymbol{V_R}(1))^T \big(\boldsymbol{G_x^r}\big)^T \boldsymbol{S_x^r} \boldsymbol{C^r} \boldsymbol{G_x^r} \boldsymbol{V_R}(1) + (\boldsymbol{V_R}(1))^T \big(\boldsymbol{G_y^r}\big)^T \boldsymbol{S_y^r} \boldsymbol{C^r} \boldsymbol{G_y^r} \boldsymbol{V_R}(1) \Big] \\
& + \lambda \Big[(\boldsymbol{V_G}(1))^T \big(\boldsymbol{G_x^g}\big)^T \boldsymbol{S_x^g} \boldsymbol{C^g} \boldsymbol{G_x^g} \boldsymbol{V_G}(1) + (\boldsymbol{V_G}(1))^T \big(\boldsymbol{G_y^g}\big)^T \boldsymbol{S_y^g} \boldsymbol{C^g} \boldsymbol{G_y^g} \boldsymbol{V_G}(1) \Big] \\
& + \lambda \Big[(\boldsymbol{V_B}(1))^T \big(\boldsymbol{G_x^b}\big)^T \boldsymbol{S_x^b} \boldsymbol{C^b} \boldsymbol{G_x^b} \boldsymbol{V_B}(1) + (\boldsymbol{V_B}(1))^T \big(\boldsymbol{G_y^b}\big)^T \boldsymbol{S_y^b} \boldsymbol{C^b} \boldsymbol{G_y^b} \boldsymbol{V_B}(1) \Big],
\end{aligned} \tag{17}$$

where $\boldsymbol{V_R}$, $\boldsymbol{V_G}$, and $\boldsymbol{V_B}$ are respectively the vectors of $\boldsymbol{R}_0$, $\boldsymbol{G}_0$, and $\boldsymbol{B}_0$. In addition, the symbols in Eq. (17) for red-channel will be introduced in the following, and the symbols of other channels can be labelled similarly. $\boldsymbol{G_x^r}$ and $\boldsymbol{G_y^r}$ are the Toeplitz matrices from the discrete gradient operators using forward difference for red-channel. $\boldsymbol{S_x^r}$, $\boldsymbol{S_y^r}$, and $\boldsymbol{C^r}$ are the diagonal matrices for red-channel, whose diagonal values are $\boldsymbol{S_x^r}[i,i] = \boldsymbol{s}_{x,i}^r$, $\boldsymbol{S_y^r}[i,i] = \boldsymbol{s}_{y,i}^r$, and $\boldsymbol{C^r}[i,i] = \boldsymbol{c}_i^r$.

To minimize Eq. (17), we take the derivative respectively w.r.t $\boldsymbol{V_R}(1)$, $\boldsymbol{V_G}(1)$, and $\boldsymbol{V_B}(1)$, and then set them to be zero. Finally, the solutions to the minimization of Eq. (17) can be written as:

$$\begin{aligned}
\boldsymbol{V_R} &= [\boldsymbol{E} + \lambda((\boldsymbol{G_x^r})^T \boldsymbol{S_x^r} \boldsymbol{C^r} \boldsymbol{G_x^r} + (\boldsymbol{G_y^r})^T \boldsymbol{S_y^r} \boldsymbol{C^r} \boldsymbol{G_y^r}] \boldsymbol{V_R}(1), \\
\boldsymbol{V_G} &= [\boldsymbol{E} + \lambda((\boldsymbol{G_x^g})^T \boldsymbol{S_x^g} \boldsymbol{C^g} \boldsymbol{G_x^g} + (\boldsymbol{G_y^g})^T \boldsymbol{S_y^g} \boldsymbol{C^g} \boldsymbol{G_y^g}] \boldsymbol{V_G}(1), \\
\boldsymbol{V_B} &= [\boldsymbol{E} + \lambda((\boldsymbol{G_x^b})^T \boldsymbol{S_x^b} \boldsymbol{C^b} \boldsymbol{G_x^b} + (\boldsymbol{G_y^b})^T \boldsymbol{S_y^b} \boldsymbol{C^b} \boldsymbol{G_y^b}] \boldsymbol{V_B}(1),
\end{aligned} \tag{18}$$

where $\boldsymbol{E}$ is the identity matrix.

Given the initial image $\boldsymbol{I}_0^c = [\boldsymbol{R}_0, \boldsymbol{G}_0, \boldsymbol{B}_0]$, the iterative optimization procedure of our LAD-RTVs is specifically presented as follows:

1. In each iteration, Eqs. (14) and (16) are used to calculate $\boldsymbol{s}_{x,p}^r$ and $\boldsymbol{s}_{y,p}^r$ in order to get matrices $\boldsymbol{S_x^r}(t-1)$ and $\boldsymbol{S_y^r}(t-1)$ for red-channel. Similarly, we can get matrices $\boldsymbol{S_x^g}(t-1)$, $\boldsymbol{S_y^g}(t-1)$, $\boldsymbol{S_x^b}(t-1)$, and $\boldsymbol{S_y^b}(t-1)$. According to Eq. (13), $\boldsymbol{C^r}(t-1)$ can be obtained. In the first iteration, $\boldsymbol{S_x^r}(0)$ and $\boldsymbol{S_y^r}(0)$ are calculated from $\boldsymbol{R}_0$. Otherwise, $\boldsymbol{S_x^r}(t-1)$ and $\boldsymbol{S_y^r}(t-1)$ are obtained from $\boldsymbol{R}(t-1)$, whose vector form is $\boldsymbol{V_R}(t-1)$. In this way, $\boldsymbol{S_x^g}(t-1)$, $\boldsymbol{S_y^g}(t-1)$, $\boldsymbol{S_x^b}(t-1)$ and $\boldsymbol{S_y^b}(t-1)$ can be calculated.
2. Given $\boldsymbol{S_x^r}(t-1)$, $\boldsymbol{S_y^r}(t-1)$, $\boldsymbol{G_x^r}(t-1)$, and $\boldsymbol{G_y^r}(t-1)$, the vector of $\boldsymbol{V_R}(t)$ can be calculated in each iteration according to Eq. (19). In the similar way, we can get $\boldsymbol{V_G}(t)$ and $\boldsymbol{V_B}(t)$. Note that $\boldsymbol{V_R}$, $\boldsymbol{V_G}$, and $\boldsymbol{V_B}$ are fixed during iteration.
3. After $\aleph$ times iterations with step (1–2), $\boldsymbol{V_R}(\aleph)$, $\boldsymbol{V_G}(\aleph)$, and $\boldsymbol{V_B}(\aleph)$ are re-arranged into a matrix $\boldsymbol{I}_\aleph = [\boldsymbol{R}_\aleph, \boldsymbol{G}_\aleph, \boldsymbol{B}_\aleph]$ with size $M \times N \times 3$, which is the final output image.

$$\begin{aligned}
\boldsymbol{V_R}(t) = [&\boldsymbol{E} + \lambda(\boldsymbol{G_x^r}(t-1)^T \boldsymbol{S_x^r}(t-1) \boldsymbol{C^r}(t-1) \boldsymbol{G_x^r}(t-1) \\
&+ \boldsymbol{G_y^r}(t-1)^T \boldsymbol{S_y^r}(t-1) \boldsymbol{C^r}(t-1) \boldsymbol{G_y^r}(t-1))]^{-1} \boldsymbol{V_R}
\end{aligned} \tag{19}$$

As compared with RTV [1], which is specifically designed for image smoothing, our LAD-RTV framework can be used for not only image smoothing but also image denoising. Next, we introduce our LAD-RTV denoising model (denoted as LAD-RTVd) in detail. Given a noisy image $\boldsymbol{I}_{noisy}^c = [\boldsymbol{R}_0, \boldsymbol{G}_0, \boldsymbol{B}_0]$, our LAD-RTVd is given in Eq. (20). The solution of LAD-RTVd in Eq. (20) can be obtained in the t-th iterative step similarly according to the derivation for LAD-RTV, which is presented in Eq. (21), where $\boldsymbol{W^r}$ is the diagonal matrix and its p-th diagonal value is $\sqrt{\boldsymbol{v}_p^r \cdot \boldsymbol{v}_p^c}$. In this way, $\boldsymbol{W^g}$ and $\boldsymbol{W^b}$ can be got similarly. The default values of th_l and th_h for denoising with LAD-RTVd are respectively set to be 4 and 30. Just as the denoising of LAD-AD, because the product of RTV and normalized and clipped standard variation could capture the locations of the noises in the contaminated image, the detected noisy pixels are always smoothed by LAD-RTVd to achieve image denoising. This comes from a fact that gradient information always contains noise gradient change except for boundary change, but a local standard deviation is usually a more stable statistic measure in relative.

$$\begin{aligned}
\mathop{\arg\min}_{\boldsymbol{R},\boldsymbol{G},\boldsymbol{B}} & \Big[\sum_p (\boldsymbol{R}(p) - \boldsymbol{R}_0(p))^2 + \sum_p (\boldsymbol{G}(p) - \boldsymbol{G}_0(p))^2 + \sum_p (\boldsymbol{B}(p) - \boldsymbol{B}_0(p))^2 \Big] \\
&+ \lambda \cdot \left\{ \sqrt{\sum_p \frac{(\mathcal{D}r)_x(p) \cdot \boldsymbol{v}_p^r}{(\mathcal{L}r)_x(p)+\epsilon} \cdot \frac{(\mathcal{D}r)_x^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}r)_x^c(p)+\epsilon}} + \sqrt{\sum_p \frac{(\mathcal{D}r)_y(p) \cdot \boldsymbol{v}_p^r}{(\mathcal{L}r)_y(p)+\epsilon} \cdot \frac{(\mathcal{D}r)_y^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}r)_y^c(p)+\epsilon}} \right. \\
&+ \sqrt{\sum_p \frac{(\mathcal{D}g)_x(p) \cdot \boldsymbol{v}_p^g}{(\mathcal{L}g)_x(p)+\epsilon} \cdot \frac{(\mathcal{D}g)_x^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}g)_x^c(p)+\epsilon}} + \sqrt{\sum_p \frac{((\mathcal{D}g)_y(p) \cdot \boldsymbol{v}_p^g}{(\mathcal{L}g)_y(p)+\epsilon} \cdot \frac{(\mathcal{D}g)_y^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}g)_y^c(p)+\epsilon}} \\
&+ \left. \sqrt{\sum_p \frac{(\mathcal{D}b)_x(p) \cdot \boldsymbol{v}_p^b}{(\mathcal{L}b)_x(p)+\epsilon} \cdot \frac{(\mathcal{D}b)_x^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}b)_x^c(p)+\epsilon}} + \sqrt{\sum_p \frac{(\mathcal{D}b)_y(p) \cdot \boldsymbol{v}_p^b}{(\mathcal{L}b)_y(p)+\epsilon} \cdot \frac{(\mathcal{D}b)_y^c(p) \cdot \boldsymbol{v}_p^c}{(\mathcal{L}b)_y^c(p)+\epsilon}} \right\}
\end{aligned} \tag{20}$$

$$\begin{aligned}
&\boldsymbol{V_R}(t) = [\boldsymbol{E} + \lambda[(\boldsymbol{G_x^r}(t-1))^T \boldsymbol{S_x^r}(t-1) \boldsymbol{W^r}(t-1) \boldsymbol{G_x^r}(t-1) \\
&+ (\boldsymbol{G_y^r}(t-1))^T \boldsymbol{S_y^r}(t-1) \boldsymbol{W^r}(t-1) \boldsymbol{G_y^r}(t-1)]^{-1} \boldsymbol{V_R}; \\
&\boldsymbol{V_G}(t) = [\boldsymbol{E} + \lambda[(\boldsymbol{G_x^g}(t-1))^T \boldsymbol{S_x^g}(t-1) \boldsymbol{W^g}(t-1) \boldsymbol{G_x^g}(t-1) \\
&+ (\boldsymbol{G_y^g}(t-1))^T \boldsymbol{S_y^g}(t-1) \boldsymbol{W^g}(t-1) \boldsymbol{G_y^g}(t-1)]^{-1} \boldsymbol{V_G}; \\
&\boldsymbol{V_B}(t) = [\boldsymbol{E} + \lambda[(\boldsymbol{G_x^b}(t-1))^T \boldsymbol{S_x^b}(t-1) \boldsymbol{W^b}(t-1) \boldsymbol{G_x^b}(t-1) \\
&+ (\boldsymbol{G_y^b}(t-1))^T \boldsymbol{S_y^b}(t-1) \boldsymbol{W^b}(t-1) \boldsymbol{G_y^b}(t-1)]^{-1} \boldsymbol{V_B}
\end{aligned} \tag{21}$$

In the RTV model [1], whether a pixel is judged as a texture pixel or a structural pixel depends on the measurements of the WTV and WIV within a patch. Thus, the RTV model smoothes all the textural pixels so as to extract image structures from textures. However, our LAD-RTVd judges whether and how much a pixel belongs to a noisy pixel based on a combination of local activity measurement and gradient, so LAD-RTVd prefers to smoothen noisy pixels detected by local activity and gradient, rather than all the

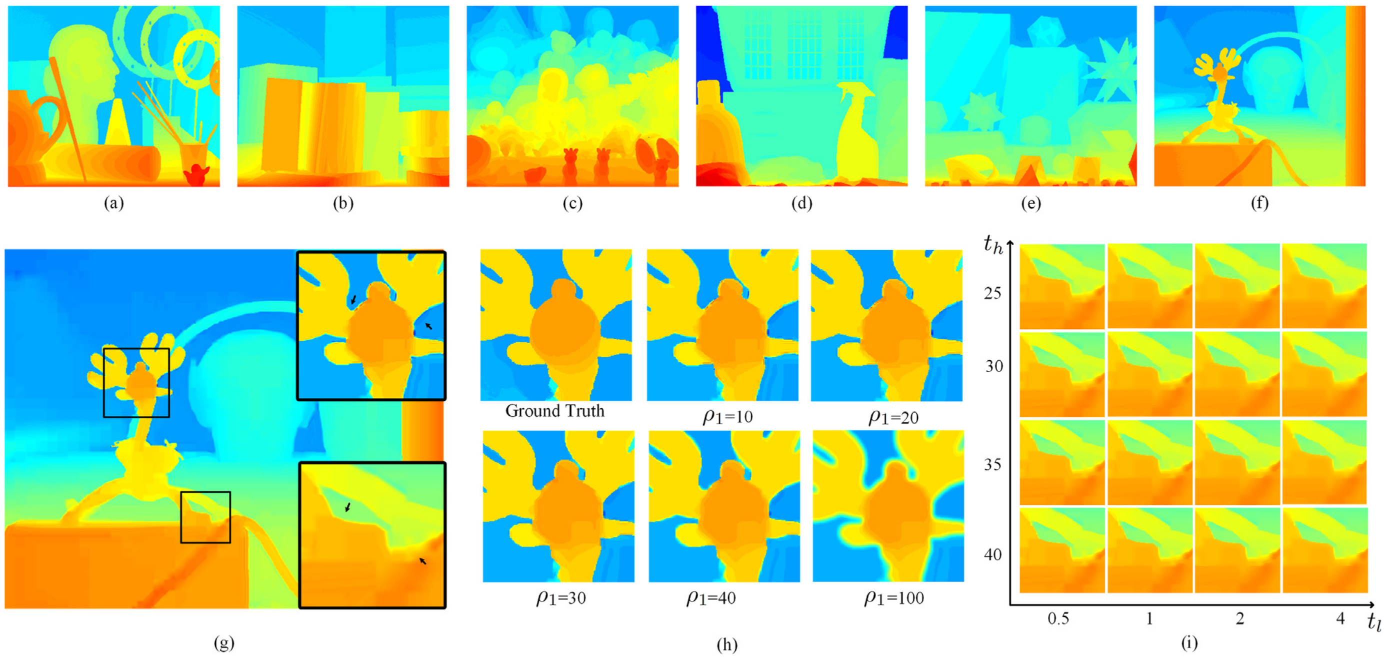

Fig. 1. (a-f) are several depth maps of different scenes from Middlebury, (g) HEVC-compressed depth image with QP=41, (h) the close-ups of the depth images filtered by TLAD-AD with different ρ_1 for (g) except for the ground truth, (i) the close-ups of the depths filtered TLAD-AD with different parameters of th_l and th_h for (g).

textural pixels. Therefore, our LAD-RTVd can maintain more detailed textures than RTV for image denoising.

Note that the LAD-RTVs reduces to discriminatively color RTV (denoted as DC-RTV), when $\sqrt{\boldsymbol{v}_i^r\cdot\boldsymbol{v}_i^c}=1$, $\sqrt{\boldsymbol{v}_i^g\cdot\boldsymbol{v}_i^c}=1$, and $\sqrt{\boldsymbol{v}_i^b\cdot\boldsymbol{v}_i^c}=1$ for each pixel i. Our DC-RTV could preserve more structures than RTV, which will be validated later, because both color-sharing information and single-channel discriminative information are used in our model. From Eq. (13), it can be clearly seen that LAD-RTV employs a multiplication way between local activity $\sqrt{\boldsymbol{v}_p^r\cdot\boldsymbol{v}_p^c}$ and gradient in the x-direction for red-channel. On the contrary, LAD-RTVd uses a division manner of local activity $\sqrt{\boldsymbol{v}_p^r\cdot\boldsymbol{v}_p^c}$ and gradient in the x-direction for red-channel. For the other channels, they share the similar expression.

5. Experimental results and analysis

In this section, we show extensive results to demonstrate the performance of the proposed frameworks. First, we first introduce the parameter setting,[1] after which our LAD-AD is applied to artifact removal of HEVC-compressed depth images. Then, we validate the efficiency of the proposed LAD-RTVs on image smoothing and scale representation. Finally, our LAD-RTVd is compared with several denoising methods to demonstrate its novelty.

5.1. Parameter setting

Several depth maps of different scenes from Middlebury[2] are used to analyze the parameter selection for HEVC-compressed depth image filtering, as displayed in Fig. 1 (a–f). Meanwhile, several different color images in Fig. 2 (a–f) from the McMaster dataset[3] are used to observe the performance of LAD-RTVd with different parameters. As well all know, the diffusion parameter plays a vital role in image denoising, so we use TLAD-AD as an example to see how depth images change after filtering with different diffusion parameters. From Fig. 1 (g–h), it can be observed that the filtered depth images tend to be blurry, when diffusion parameter ρ_1 is set too large, e.g., $\rho_1 = 40$ or 100. At the same time, coding artifacts can't be well removed from the HEVC-compressed depth image, if ρ_1 is too small, e.g., $\rho_1 = 10$ or 20. Moreover, the performance of TLAD-AD always is robust to noise and not to be over-smoothed but sharp when $\rho_1 = 30$. Based on these observations, we choose $\rho_1 = 30$ in the proposed TLAD-AD[1]. t_l and t_h are two parameters in the clip function to truncate the local activity in the proposed anisotropic diffusion. From Fig. 2 (g, i), we see that filtered depth images become to be more smooth or less sharp, when t_h is set to be lower, e.g., $t_h = 25$, and vice versa. If we fix the parameter, e.g., $t_h = 40$, TLAD-AD tends to make more discontinuities preserved, as t_l is set to be smaller, e.g, $t_h = 0.05$, and vice versa. From above, it can be known that both t_l and t_h can't be too large yet too small. In our simulation, t_l and t_h are chosen to be 1 and 30, because image smoothness and image structural preservation should be well balanced.

In the LAD-RTVd[1], the parameters of th_l and th_h have enormous impacts on image denoising. When we change th_l from low to high, as shown in Fig. 2 (h), non-boundary regions of the denoised image become more smooth and are less affected by noise. The LAD-RTVd-filtered images tend to have more details as th_h is adjusted from 20 to 40, but some structures of noisy are kept. Therefore, th_l in LAD-RTVd can be chosen from 1 to 4, and th_h are restricted to be less than 40 but larger than 20. To trade-off image smoothness and insensitivity to noise structure, we choose th_l to be 4 and th_h to be 30 for LAD-RTVd in default.

5.2. Compressed depth image filtering with LAD-AD

We use four standard multi-view-plus-depth sequences: Nokia's Undo_Dancer (U), NICT's Shark (S), Nagoya University's Champagne_Tower (C) (in which the first 250 frames of these three sequences are tested) and HHI's Book_Arrival (B) (in which the whole sequences with 100 frames are tested). The depth maps of these sequences are compressed by HEVC-v16.8 with quantization parameter (QP) chosen as 31, 33, 35, 37, 39 and 41, respectively. In the simulations, the 1D-fast mode of 3D-HEVC (HTM-DEV-2.0-dev3 version) is used to synthesize the virtual middle view using two

[2] http://vision.middlebury.edu/stereo/data/
[3] http://www4.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm

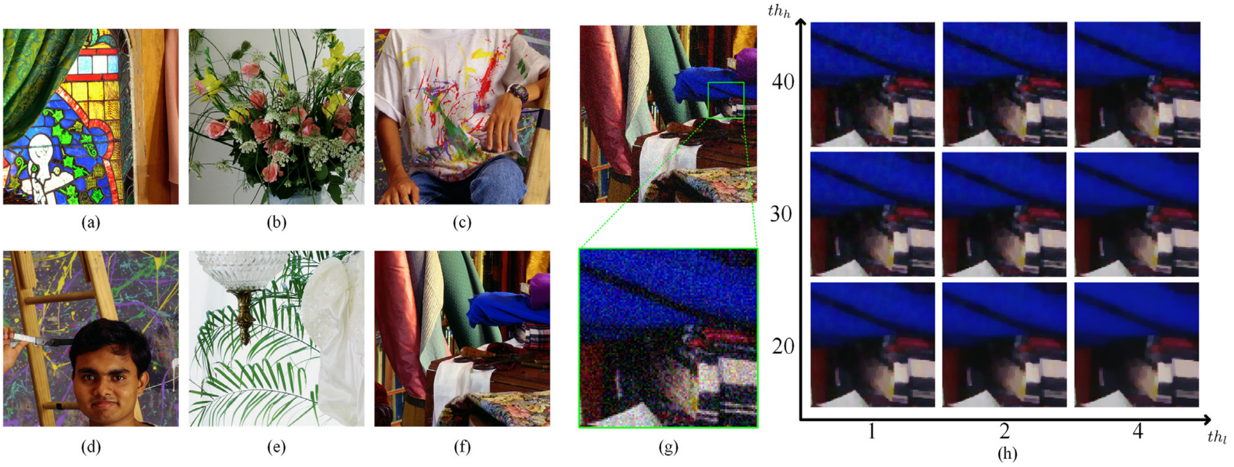

Fig. 2. (a-f) are different color images from the McMaster dataset, (g) the noisy color image and its close-up, (h) the close-ups of the denoised images filtered by LAT-RTVd with different parameters of th_l and th_h for (g) .

views of uncompressed texture images and compressed depth images (filtered or non-filtered).

Generally, the direct way of iterative termination for our LAD-AD uses Mean Square Error (MSE) δ between two latest iterative images as the iterative stop criteria. For example, when δ is set to be very small as 0.0001, the corresponding TLAD-AD is denoted as TLAD-AD-1. If δ equals to 0.001, we denote it as TLAD-AD-2. However, this stop criteria doesn't work well for compressed depth images filtering with the LAD-AD. This will be verified in the following. The alternative way is to use the fixed iteration number as the termination condition for compressed depth images with fixed QP, because quantized depth image with fixed QP always has the same level coding artifacts and the iteration number always affects the filtering time. For FLAD-AD, TLAD-AD, PLAD-AD, FLAD-AD (I), and PLAD-AD (I), λ is 0.25, and the fixed number of iteration is 11 when QP is lower than 37, which are the experimental values. Otherwise, the number of iteration is 21. For PLAD-AD and PLAD-AD (I), the interval is 5 when QP =31, 34, 35, but the interval l is set to be 10 if QP=37, 39, 41. In our experiment, all the sequences are set with the fixed parameters ρ_1 and ρ_2 for HEVC-compressed depth filtering, as mentioned above. Note that we use the iteration number to control the strength of artifact removal, i.e., more artifacts would be removed, as more iterations are updated.

In order to validate the efficiency of the proposed LAD-AD, our filtering results are compared with the ones of BF [40], JTF [41], IRWF [9], ADTF [36], and TSF [38]. For both filtered depth images and corresponding synthesized virtual images (the middle view of two reference views), peak signal noise ratio (PSNR), structural similarity (SSIM), and image sharpness (ISS) [42], are taken as three objective quality evaluation metrics.

From Table 1, it can be clearly observed that FLAD-AD, TLAD-AD, PLAD-AD, FLAD-AD (I), and PLAD-AD (I) have better performance than TLAD-AD-1 and TLAD-AD-2. It indicates that using the fixed iteration number as the termination condition is superior to the MSE δ between two latest iterative images as the iterative stop criteria, when depth images are compressed with fixed QP. In addition, FLAD-AD (I) and PLAD-AD (I) have more stable results than FLAD-AD, TLAD-AD, and PLAD-AD. The performances of FLAD-AD and TLAD-AD as well as PLAD-AD are different, and the sharpness of TLAD-AD is more than PLAD-AD, but TLAD-AD requires to update the local activity in each step, so TLAD-AD has more complexity than PLAD-AD. The diffusion of FLAD-AD leads to the blurring of depth image discontinuities, so it has the worst performance on boundary regions, as compared to the other methods. In contrast to FLAD-AD, TLAD-AD, and PLAD-AD, the performances of FLAD-AD (I), TLAD-AD (I), and PLAD-AD (I) are very similar, because the form of $(\frac{||\nabla \boldsymbol{I}_{ij}^t||}{\rho_1 \boldsymbol{K}_i^t})^2$ leads to more diffusion for some artifact pixels than the form of $\frac{||\nabla \boldsymbol{I}_{ij}^t||^2}{\rho_2^2 \boldsymbol{K}_i^t}$ in each iteration. The stop-function in Eq. (3) is more efficient on preserving sharp boundaries, as compared to the stop-function in Eq. (4). But the stop-function of Eq. (4) in the proposed FLAD-AD (I), TLAD-AD (I), and PLAD-AD (I) does not change depth structures too much, and most of the detailed geometry structures are well preserved when removing severe coding artifacts.

As shown in Table 1, the proposed method can greatly improve the quality of both depth images and synthesized virtual image at the same time, as compared with several state-of-the-art approaches. From this table, it can also be seen that the proposed PLAD-AD (I) and FLAD-AD (I) greatly improve the accuracy of depth images and have the best performance except BF [40], while greatly enhancing synthesized images. Although depth image PSNR of BF [40] is high, it gets a little gains regarding PSNR, ISS, SSIM of the synthesized virtual image as well as ISS and SSIM of depth images, as compared with other methods. The synthesized images with filtered depth images are displayed in Fig. 3, from which we can see that the visual quality of the proposed method has superior performance to the other approaches.

From Fig. 3 (c), it can be observed that BF [40] can smoothen some artifacts, so it cannot restore image sharp boundaries, and the edges remain to be blurring. JTF [41] slightly improves the quality depth image and its synthesized color image, as displayed in Fig. 3 (d-n), but it can't compete with the proposed LAD-AD, IRWF [9], ADTF [36], and TSF [38] regarding the quality improvement of synthesized color image. As depicted in Fig. 3 and Table 1, IRWF [9], ADTF [36], and TSF [38] improve the objective and visual quality of synthesized images, but they do not greatly enhance the quality of depth images and even make them worse than the unfiltered distorted depth images. One fatal drawback of ADTF [36], and TSF [38] is that these methods always smoothen some small yet significant objects, and even may eliminate some small objects, as shown in Fig. 3 (c–e). It is obvious that the proposed method can avoid these drawbacks, as compared with these methods.

From Table 1, it can be seen that the BF [40], JTF [41], and IRWF [9] spends more filtering time than the proposed method, ADTF [36], and TSF [38], while the filtering time of the proposed TLAD-AD-2, TLAD-AD, FLAD-AD, PLAD-AD, FLAD-AD (I) and FLAD-AD (I) is less than TSF and TLAD-AD-1, but more than ADTF [36]. In addition, the TLAD-AD filtering time is more than FLAD-AD, PLAD-AD,

Table 1
The average objective quality comparison of HEVC-compressed depth images filtered by different methods and its synthesized images over four standard multi-view-plus-depth sequences with different quantization parameters. (The font of the top three ranking methods is bold).

Measurements	PSNR		ISS		SSIM		Time(s)
Methods	Depth	Sythesized	Depth	Sythesized	Depth	Sythesized	Depth
Coded	44.05	50.20	52.61	55.84	0.96426	0.99690	–
BF	**44.49**	50.88	53.06	56.05	**0.96755**	0.99721	16.15
JTF	44.25	50.45	52.91	56.02	0.96741	0.99706	41.82
IRWF	44.04	51.37	53.02	56.36	0.96646	0.99743	16.86
ADTF	43.93	51.31	52.84	56.27	0.96557	0.99739	**1.20**
TSF	43.39	**51.47**	52.91	**56.33**	0.96386	**0.99740**	3.45
TLAD-AD-1	44.09	51.20	53.10	56.28	0.96596	0.99737	4.18
TLAD-AD-2	44.34	50.89	52.97	56.12	0.96653	0.99729	**1.72**
FLAD-AD	44.16	50.86	**53.12**	56.30	0.96635	0.99739	**2.04**
TLAD-AD	44.18	50.80	53.10	56.29	0.96629	0.99737	4.35
PLAD-AD	44.19	50.83	53.10	56.29	0.96636	0.99737	2.47
FLAD-AD (I)	**44.39**	**51.43**	**53.18**	**56.33**	**0.96764**	**0.99740**	2.30
PLAD-AD (I)	**44.39**	**51.42**	**53.19**	**56.33**	**0.96767**	**0.99740**	2.57

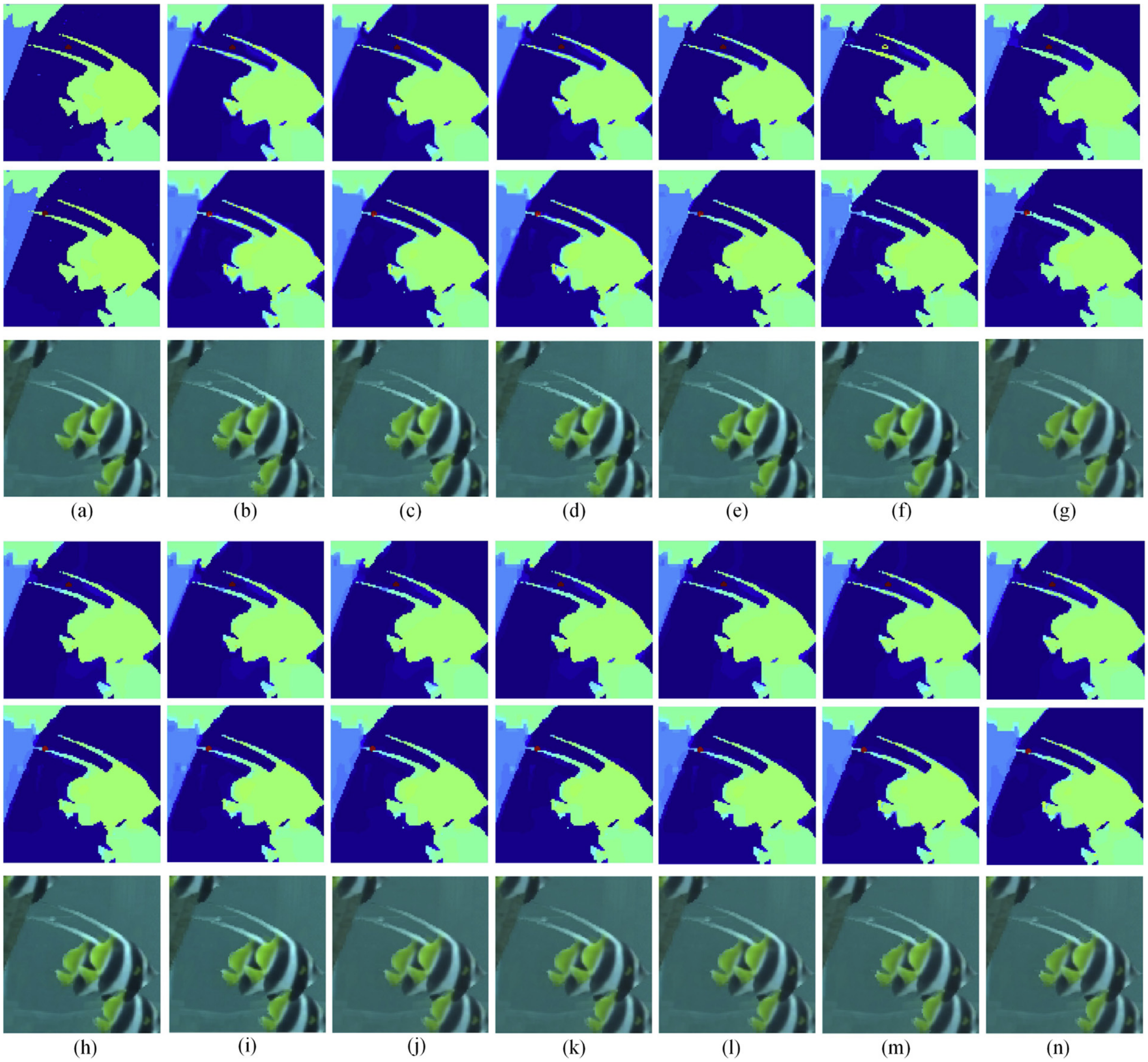

Fig. 3. The first row: (a) the close-up of the original depth map Shark in view 1, (b) HEVC (QP41), (c) BF [40], (d) JTF [41], (e) IRWF [9], (f) ADTF [36], (g) TSF [38], (h-n) TLAD-AD-1, TLAD-AD-2, FLAD-AD, TLAD-AD, PLAD-AD, FLAD-AD (I), PLAD-AD (I); the second row of (a-j) is corresponding depth image of view 5; the third row of (a-n) is middle virtual images synthesized by corresponding depth images in the first and second row.

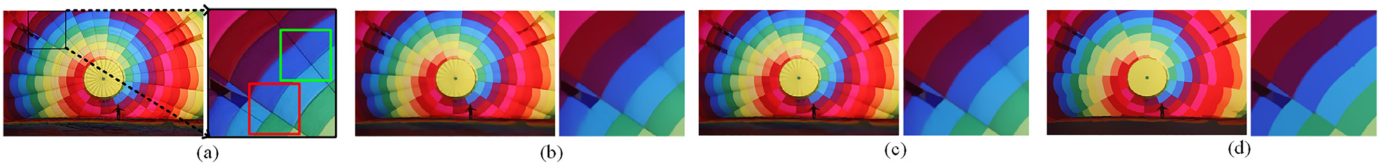

Fig. 4. The visual comparison of RTV, DC-RTV, and LAD-RTV. (a) Input image (Notice the red-line and green-line boxed regions for better comparison of different methods), (b) RTV [1], (c) DC-RTV, (d) our LAD-RTVs. For each image of (a-d), the left image is the full image, and the right image is the close-up. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

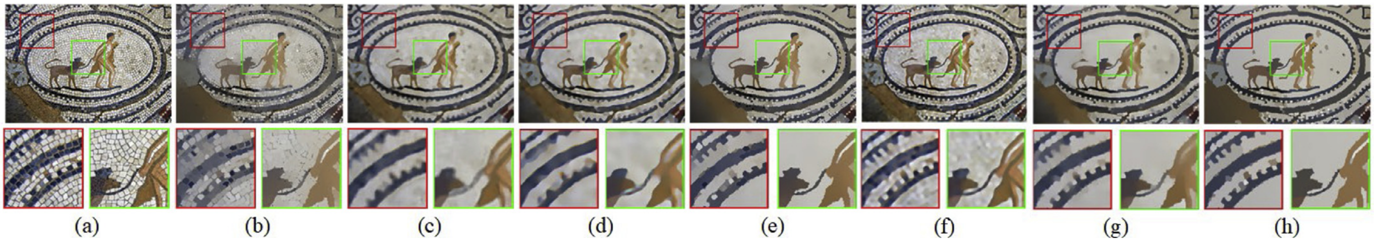

Fig. 5. The visual comparison of several image smoothing methods. (a) Input image, (b) WLS [3], (c) RC [6], (d) RGF [7], (e) RGIF [8], (f) DEAF(RGF) [43], (g) RTV [1], (h) our LAD-RTVs.

FLAD-AD (I) and PLAD-AD (I), because TLAD-AD requires to calculate the local activity in each iteration. TLAD-AD-1 takes more time than TLAD-AD-2 because it requires a large number of iteration to reach the stop condition.

In summary, our FLAD-AD (I) and PLAD-AD (I) have more stable performances than TLAD-AD-1, TLAD-AD-2, TLAD-AD, FLAD-AD, and PLAD-AD in terms of PSNR, ISS, SSIM, and image filtering time for filtered depth image and its synthesized image. Our FLAD-AD (I) and PLAD-AD (I) can greatly improve the quality of the depth image and its synthesized image at the same time, when compared with BF [40], JTF [41], and IRWF [9], TSF [38], and ADTF [36]. Furthermore, our FLAD-AD (I) and PLAD-AD (I) spend less filtering time than BF [40], JTF [41], IRWF [9], and TSF [38], apart from ADTF [36].

5.3. Image smoothing and scale representation with LAD-RTVs

For texture removal, we first compare the proposed LAD-RTVs with its reduced DC-RTV and RTV [1]. As shown in Fig. 4, our DC-RTV and LAD-RTVs are superior to RTV [1] on edge-preserving. In our LAD-RTVs model, by using both color-sharing information and each-channel discriminative information to form DCWTV and DCWIV measurements, our model can better maintain image salient contours than RTV [1]. Meanwhile, DC-RTV is inferior to the proposed model of LAD-RTVs, which can further remove more details. Secondly, we compare the proposed LAD-RTVs with several state-of-the-art image smoothing approaches, such as RTV [1], weighted least squares (WLS) [3], region covariance based method (RC) [6], rolling guidance filter (RGF) [7], robust guided image filtering (RGIF) [8], and DEAF(RGF) [43], as shown in Fig. 5. When the input image has complex textures with strong gradient, WLS could not remove these detail pixels efficiently, as displayed in Fig. 5 (b). RGF and DEAF(RGF) [43] could remove all kinds of textures, but this method has the difficulty on accurate edge localization, as displayed in Fig. 5 (d, f). In addition, RGF [7], RC [6] and DEAF(RGF) [43] may make most of edges to be blurred, as shown in Fig. 5 (d, e, f), although they have removed many details and textures. As displayed in Fig. 5 (h, g, e), the proposed LAD-RTVs, RTV [1], and RGIF [8] have some similar appearances for texture removal, but our methods could preserve more meaningful salient contour than RTV [1] and RGIF [8]. For image representation in three scales, we have compared our LAD-RTVs with WLS [3], L0GM [2], RTV [1], RGF [7], RGIF [8], and MuGIF [34]. Following [34], we set the comparative methods in the common smoothing level by tuning the parameters of each method to reach a similar difference, whose details can be found in [34]. As depicted in Fig. 6 (h), the proposed LAD-RTVs can preserve sharp edge and well locate the edge of main object contour, as compared with several existing methods. As displayed in Fig. 6 (b–c), WLS [3] and L0GM [2] can well remove the texture details, but they work poorly on the scale representation. RGF [7] can coarsely represent an image in different scales and preserve the contour, but it also changes edge localizations using the isotropic Gaussian kernel during the initialization. From Fig. 6 (d–f) and (g-h), it can be seen that LAD-RTVs, RTV [1], RGIF [8], and MuGIF [34] have a similar performance on the scale-space representation of images, but there are some differences on boundary preservation. The proposed LAD-RTVs could preserve more structures for scale-space representation than others. Although the proposed method LAD-RTVs, RGIF [8] and MuGIF [34] are achieved by optimization with similar appearances, they use different smoothing techniques. The proposed LAD-RTVs and RTV [1] use the features of texture and structure, but both RGIF [8] and MuGIF [34] consider the static and dynamic guidance's joint effects for image smoothing. Additionally, when compared with RTV [1], the proposed method has better performance on texture removal and edge localization for image scale-representation, because our LAD-RTVs leverages both color-sharing information and single-channel discriminative information.

5.4. Image denoising with LAD-RTVd

Ten image set consists of “Monarch”, “Barbara”, “Pepper”, “Lena”, “Man”, “Comic”, “Zebra”, “Flowers”, “Bird”, and “Boat”. This set is tested to evaluate the efficiency of different methods for image denoising. The zero mean Gaussian noises are added into the clean image with standard deviation of 13, 26, and 52 to get noisy images. We compare the proposed LAD-RTVd with several existing novel methods: BM3D [44], RBF [17], WBF [17], TV [45], and RTV [1], as shown in Table 2. From this table, it can be observed that the average objective quality of the proposed method for denoising has better performance than TV, RTV, RBF and WBF. Meanwhile, LAD-RTVd has better PSNR measurement than LAD-RTVd (1) and the LAD-RTVd filtered images are more smooth than those filtered by LAD-RTVd (1).

However, LAD-RTVd has lower quality than the approach of BM3D, which are built upon the block-matching and 3D filtering

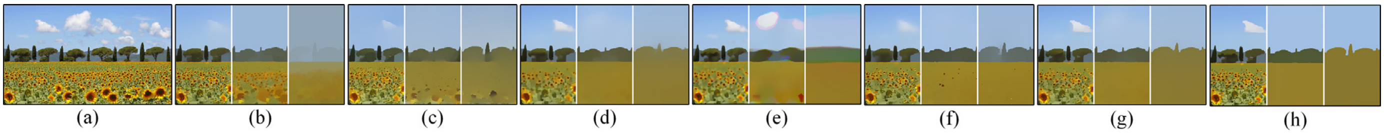

Fig. 6. The visual comparison of scale-space representation (a) Input image, (b) WLS [3], (c) L0GM [2], (d) RTV [1], (e) RGF [7], (f) RGIF [8], (g) MuGIF [34], (h) ours LAD-RTVs.

Table 2
The average objective PSNR comparison of noisy images filtered by different methods.

Sigma	Noisy	BM3D [44]	RBF [17]	WBF [17]	TV [45]	RTV [1]	LAD-RTVd (1)	LAD-RTVd
13	26.22	32.61	28.61	30.62	27.79	28.07	31.04	31.31
26	20.37	28.93	27.55	27.87	26.41	26.14	27.84	28.06
52	14.83	24.98	24.27	23.63	23.38	23.94	24.39	24.49
Ave.	20.47	**28.84**	26.81	27.38	25.86	26.05	27.76	**27.95**

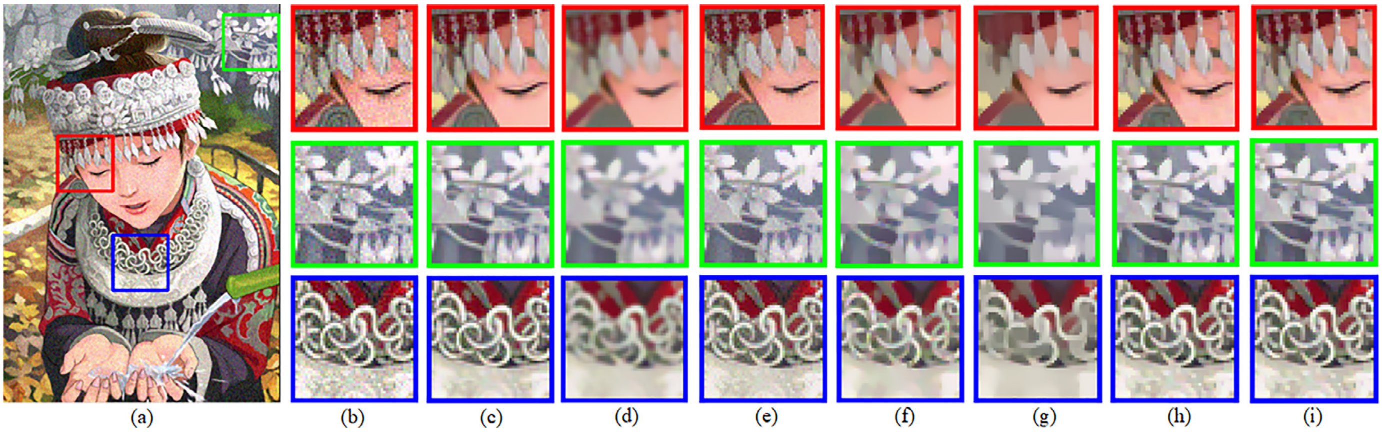

Fig. 7. The visual comparison of several denoising methods: (a) "Comic" containing zero mean Gaussian noise with standard deviation to be 13; (b) is the close-ups of the line-boxed regions in (a); (c) BM3D [44], (d) RBF [17], (e) WBF [17], (f) TV [45], (g) RTV [1], (h) LAD-RTVd (1), (i) our LAD-RTVd. (Note that LAD-RTVd (1) refers to LAD-RTVd with $th_l = 1$).

with very high computational complexity. Our LAD-RTVd comes from image smoothing by regarding the noises as the duplicate texture elements. Thus, the inherent drawbacks of LAD-RTVd limit its denoising efficiency, especially for textural images with noise-like details. As compared to BM3D, our LAD-RTVd could produce sharper results, as displayed in Fig. 7 (c, i). However, the BM3D-filtered image tends to be blurry, which may often lead to higher PSNR improvement than LAD-RTVd. The visual performance of TV [45] is inferior to the one of RTV [1], as displayed in Fig. 7 (f, g). RBF [17] and WBF [17] have high objective performance than RTV [1], as shown in Fig. 7 (d, e, g). Although RBF [17] can eliminate the noisy pixels by bilateral filtering, it may remove some fine details. Built on RBF [17], as shown in Fig. 7 (d, e), WBF [17] has better visual performance than RBF [17] by adding the corresponding BF-filtered noisy image back to the filtered image. From Fig. 7 (g, h), it can be seen that RTV only tends to smoothen the textures for image structures preservation, but the LAD-RTVd could well preserve details by catching the location and amplitude of noises according to a combination of the local activity and gradient. Besides, the filtered image by LAD-RTVd is more smooth than that filtered by LAD-RTVd (1), as displayed in Fig. 7 (h, i).

6. Conclusion

In this paper, we propose to use a local activity measurement of the clipped and normalized variance or standard deviation to drive anisotropic diffusion and relative total variation for better structural-preserving filtering. Meanwhile, two novel edge-stop functions are introduced for our LAD-AD to efficiently remove severe artifacts and preserve the fine geometry structures in HEVC-compressed depth images. Furthermore, our LAD-RTV can be not only used for image denoising but also image smoothing as well scale-representation. Through a large number of experimental results, it has been demonstrated that our methods are superior to several state-of-the-art approaches. Additionally, the parameter setting for our frameworks has been discussed and analyzed. Our future works will be put on the exploration of more powerful local activity measurements for image filtering, and we will generalize the proposed method for many other filtering frameworks.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (No. 61672373), Key Innovation Team of Shanxi 1331 Project (KITSX1331) and the Fundamental Research Funds for the Central Universities (No. 2017YJS053).

References

[1] L. Xu, Q. Yan, Y. Xia, J. Jia, Structure extraction from texture via relative total variation, ACM Trans. Graph. 31 (6) (2012) 1–10.
[2] L. Xu, C. Lu., Y. Xu, J. Jia, Image smoothing via l0 gradient minimization, ACM Trans. Graph. 30 (6) (2011) 1–12.
[3] Z. Farbman, R. Fattal, D. Lischinski, Edge-preserving decompositions for multi--scale tone and detail manipulation, ACM Trans. Graph. 27 (3) (2008) 1–10.
[4] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2013) 1397–1409.
[5] E. Gastal, M. Oliveira, Domain transform for edge-aware image and video processing, ACM Trans. Graph. 30 (4) (2011) 1244–1259.
[6] L. Karacan, E. Erdem, A. Erdem, Structure-preserving image smoothing via region covariances, ACM Trans. Graph. 32 (6) (2003) 1–11.

[7] Q. Zhang, L. Xu, J. Jia, Rolling guidance filter, in: European Conference on Computer Vision, Zurich, 2014.

[8] B. Ham, M. Cho, J. Ponce, Robust guided image filtering using nonconvex potentials, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2017). 1–1

[9] L. Zhao, H. Bai, A. Wang, Y. Zhao, Iterative range-domain weighted filter for structural preserving image smoothing and de-noising, Multimed Tools Appl. (2017) 1–28.

[10] S. Chao, D. Tsai, An improved anisotropic diffusion model for detail-and edge-preserving smoothing, Pattern Recognit. Lett. 31 (13) (2010) 2012–2023.

[11] S. Chao, D. Tsa, Anisotropic diffusion-based detail-preserving smoothing for image restoration, in: IEEE International Conference on Image Processing, Hong Kong, 2010.

[12] Y. Zaz, L. Masmoudi, K. Bouzouba, L. Radouane, A new adaptive anisotropic diffusion using the local intensity variance, in: International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Susa, Tunisia, 2005.

[13] M. Black, G. Sapiro, D. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. Image Process. 7 (3) (1998) 421–432.

[14] J. Bai, X. Feng, Fractional-order anisotropic diffusion for image de-noising, IEEE Trans. Image Process. 16 (10) (2007) 2492–2502.

[15] K. Chaudhury, S. Dabhade, Fast and provably accurate bilateral filtering, IEEE Trans. Image Process. 25 (6) (2016) 2519–2528.

[16] K. Chaudhury, D. Sage, M. Unser, Fast O(1) bilateral filtering using trigonometric range kernels, IEEE Trans. Image Process. 20 (12) (2011) 3376–3382.

[17] K. Chaudhury, K. Rithwik, Image de-noising using optimally weighted bilateral filters: a sure and fast approach, in: IEEE International Conference on Image Processing, Quebec, Canada, 2015.

[18] A. Buades, B. Coll, J. Morel, A non-local algorithm for image de-noising, in: IEEE Conference on Computer Vision and Pattern Recognition, Boston, 2015.

[19] T. Dai, W. Lu, W. Wang, J. Wang, S.T. Xia, Entropy-based bilateral filtering with a new range kernel, Signal Process. 137 (2017) 223–234.

[20] P. Simard, H. Malvar, An efficient binary image activity detector based on connected components, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Quebec, 2004.

[21] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 1629–1639.

[22] P.W. Hsieh, S.P. C., S.Y. Yang, A regularization model with adaptive diffusivity for variational image denoising, Signal Process. 149 (2017) 214–228.

[23] Q. Yuan, L. Zhang, H. Shen, Hyperspectral image denoising employing a spectralspatial adaptive total variation model, IEEE Trans. Geosci. Remote Sens. 50 (10) (2012) 3660–3677.

[24] Q. Yuan, L. Zhang, H. Shen, Regional spatially adaptive total variation super-resolution with spatial information filtering and clustering, IEEE Trans. Image Process. 22 (6) (2013) 2327–2342.

[25] Q. Yuan, L. Zhang, H. Shen, Multiframe super-resolution employing a spatially weighted total variation model, IEEE Trans. Circuits Syst. Video Technol. 22 (3) (2012) 379–392.

[26] M. Diwakar, M. Kumar, A review on ct image noise and its denoising, Biomed. Signal Process. Control 42 (2018) 73–88.

[27] H.V. Bhujle, B.H. Vadavadagi, Nlm based mri denoisinga review, Biomed. Signal Process. Control 47 (2019) 252–261.

[28] J. Veraart, D.S. Novikov, D. Christiaens, B. Adesaron, J. Sijbers, E. Fieremans, Denoising of diffusion mri using random matrix theory, Neuroimage 142 (2016) 394–406.

[29] N. Joshi, S. Jain, A. Agarwal, Segmentation based non local means filter for denoising mri, in: International Conference on Reliability, Infocom Technologies and Optimization, Noida, India, 2017, pp. 640–644.

[30] A. Chahid, H. Serrai, E. Achten, T.M. Laleg-Kirati, Adaptive method for mri enhancement using squared eigenfunctions of the schrηdinger operator, in: IEEE Biomedical Circuits and Systems Conference (BioCAS), Torino, Italy, 2017, pp. 1–4.

[31] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss, IEEE Trans. Med. Imaging 37 (6) (2018) 1348–1357.

[32] S. Hao, D. Pan, Y. Guo, R. Hong, M. Wang, Image detail enhancement with spatially guided filters, Signal Process. 120 (2014) 789–796.

[33] L. Zhao, H. Bai, J. Liang, A. Wang, Y. Zhao, Simultaneous color-depth super-resolution with conditional generative adversarial network, in: arXiv:1708.09105v1, 2017.

[34] X. Guo, Y. Li, J. Ma, Mutually guided image filtering, in: ACM Multimedia, Mountain View, 2017, pp. 1283–1290.

[35] S. Liu, P. Lai, D. Tian, C. Chen, New depth coding techniques with utilization of corresponding video, IEEE Trans. Broadcast. 57 (2) (2011) 551–561.

[36] X. Xu, L. Po, T. Cheung, K. Cheung, L. Feng, C. Ting, K. Ng, Adaptive depth truncation filter for MVC based compressed depth image, Signal Process. Image Commun. 29 (3) (2014) 316–331.

[37] L. Zhao, A. Wang, B. Zeng, Y. Wu, Candidate value-based boundary filtering for compressed depth images, Electron. Lett. 51 (3) (2015) 224–226.

[38] L. Zhao, H. Bai, A. Wang, Y. Zhao, B. Zeng, Two-stage filtering of compressed depth images with markov random field, Signal Process. Image Commun. 54 (2017) 11–22.

[39] D. Krishnan, R. zeliski, Multigrid and multilevel preconditioners for computational photography, ACM Trans. Graph 30 (6) (2011) 177.

[40] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: International Conference on Computer Vision, 1998, pp. 839–846.

[41] S. Liu, P. Lai, D. Tian, C. Chen, Joint trilateral filtering for depth map compression, in: Proc. SPIE 7744, Visual Communications and Image Processing 2010, 2010.

[42] M. M., C. C., L. Y., Deep multi-scale video prediction beyond mean square error, in: arXiv:1511.05440v1, 2015.

[43] L. Xu, J. Ren, Q. Yan, R. Liao, J. Jia, Deep edge-aware filters, in: International Conference on Machine Learning, Lille, France, 2015, pp. 1669–1678.

[44] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image de-noising by sparse 3D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[45] L. Rudin, O. Stanley, F. Emad, Nonlinear total variation based noise removal algorithms, Physica D 60 (1) (1992) 259–268.