sensors

MDPI

Article

Decomposed Multilateral Filtering for Accelerating Filtering with Multiple Guidance Images

Haruki Nogami [1], Yamato Kanetaka [1], Yuki Naganawa [1], Yoshihiro Maeda [2] and Norishige Fukushima [1,*]

[1] Department of Computer Science, Faculty of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan; y.kanetaka.099@stn.nitech.ac.jp (Y.K.)

[2] Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, Tokyo 125-8585, Japan; ymaeda@rs.tus.ac.jp

* Correspondence: fukushima@nitech.ac.jp

Abstract: This paper proposes an efficient algorithm for edge-preserving filtering with multiple guidance images, so-called multilateral filtering. Multimodal signal processing for sensor fusion is increasingly important in image sensing. Edge-preserving filtering is available for various sensor fusion applications, such as estimating scene properties and refining inverse-rendered images. The main application is joint edge-preserving filtering, which can preferably reflect the edge information of a guidance image from an additional sensor. The drawback of edge-preserving filtering lies in its long computational time; thus, many acceleration methods have been proposed. However, most accelerated filtering cannot handle multiple guidance information well, although the multiple guidance information provides us with various benefits. Therefore, we extend the efficient edge-preserving filters so that they can use additional multiple guidance images. Our algorithm, named decomposes multilateral filtering (DMF), can extend the efficient filtering methods to the multilateral filtering method, which decomposes the filter into a set of constant-time filtering. Experimental results show that our algorithm performs efficiently and is sufficient for various applications.

Keywords: constant-time filtering; edge-preserving filtering; multilateral filtering

check for updates

Citation: Nogami, H.; Kanetaka, Y.; Naganawa, Y.; Maeda, Y.; Fukushima, N. Decomposed Multilateral Filtering for Accelerating Filtering with Multiple Guidance Images. *Sensors* **2024**, *24*, 633. https://doi.org/10.3390/s24020633

Academic Editor: Marcin Woźniak

Received: 13 November 2023
Revised: 8 January 2024
Accepted: 17 January 2024
Published: 19 January 2024

1. Introduction

Multimodal signal processing for sensor fusion is increasingly important in image sensing. Sensor fusion can combine beneficial information from different sensors to generate a richer single image. Image signal fusion approaches have various applications: RGB and infrared image fusion [1–3], RGB and multispectral image fusion [4], intercolor RGB signal fusion [5,6], RGB and depth fusion [7,8], RGB and light fusion [9], RGB and computed edge fusion [10], different focal image fusion [11,12], CT and MRI signal fusion for medical image processing [3], Retinex-based enhancement [13], SAR and multispectral image fusion [14], and general signal fusion [15].

Filtering is a basic tool for handling such multimodal signals. Multilateral filtering, which is one type of edge-preserving filtering, successfully handles multiple signal information. Edge-preserving filtering with additional guidance information, called joint edge-preserving filtering, recently attracted attention from image processing and computational photography researchers for sensor fusion. Joint edge-preserving filtering helps transfer major characteristics from guidance images, which are not filtering images themselves. Various applications use the filters, including flash/no-flash photography [16,17], up-sampling/super resolution [18], compression noise removal [19], alpha matting [20], haze removing [21], rain removing [22], depth refinement [23,24], stereo matching [25,26], and optical flow estimation [27].

Joint/cross bilateral filtering [16,17] is a seminal work of joint edge-preserving filtering. The filter is naturally derived from bilateral filtering [28] by computing the kernel weight from a guidance image instead of an input filtering image. This formulation enables us to reflect the edge information of the guidance image (e.g., RGB, infrared, and hyperspectral images) to the filtering target image (e.g., RGB image, alpha mask, depth map, and optical flow).

We can expect a higher edge-preserving effect using multiple guidance images (e.g., a set of multi-sensor signals), and recently, we have been able to capture not only RGB images but also infrared, hyperspectral, depth, and other images by new devices (e.g., infrared/hyperspectral cameras and depth sensors). The images have different edge information and signal characteristics from RGB images. The multiple guidance information is helpful for improving signal visibility and the signal-to-noise ratio [29,30]. In other cases, we can deal with a self-generated image and inversely rendered maps as an additional guidance signal [31,32].

There are two categories for using multiple guidance images in image filtering: high-dimensional filtering [30,33–37] and multilateral filtering [29,31,32,38,39]. The former is additive logic, and the latter is multiplicative logic for additional kernels. The main difference is the severity of the restriction to compute the kernel weight. The restriction of additive logic is looser than that of multiplicative logic; hence, high-dimensional filtering can robustly smooth out noise or rich textures. By contrast, since the restriction of the multiplicative logic is severe, multilateral filtering produces fewer blurred regions. Each filtering method has advantages and disadvantages, but multilateral filtering is preferred when we expect a sharply edge-preserving effect.

A critical issue of edge-preserving filtering for multimodel sensing is computational time. This is because sensing is the gateway to all processing, and signal processing during sensing is expected to operate in real time. Therefore, many researchers have proposed acceleration methods for edge-preserving filtering. In particular, the acceleration for bilateral filtering has been actively discussed. The bilateral grid [40,41] is the seminal approach, and Yang et al. [42,43] extend it to bilateral filtering in constant time. Yang's method [43] has adequate efficiency in grayscale images, and recent work further accelerates the bilateral filter [44–46]; however, they are inefficient in color cases. There are several proposals [34,47,48] to approximate and accelerate bilateral filtering in the case of higher-dimensional (color) images. Furthermore, hardware-friendly methods are proposed [49–51]. However, these approaches have limitations for kernel weight, whose kernels are defined by the Gaussian distribution. Other efficient edge-preserving filters, which do not limit the Gaussian distribution, are proposed in contrast to the bilateral filtering acceleration. Guided image filtering [20], domain transform filtering [33], and adaptive manifold filtering [30] are representative examples. These filters have assumptions different from Gaussian smoothing but have excellent edge-preserving effects and efficiency. Note that these filters can handle similar signals better than those with different modalities and characteristics.

Multiple guidance images provide richer information for various applications; however, these efficient methods cannot individually handle multiple guidance images. Therefore, we propose an efficient algorithm for accelerating multilateral filtering, which is developed for multiple-guidance image filtering. Furthermore, we extend the efficient edge-preserving filters so that they can exploit multiple guidance images.

Our algorithm is based on the fact that n-lateral filtering is represented by the summation of $(n-1)$-lateral filtering. Therefore, when multilateral filtering is expanded as an asymptotic expression, it becomes constant-time filtering since 1-lateral filtering is spatial filtering. Figure 1 denotes the overview of the proposed filter algorithm. The proposed filter—named DMF: decomposed multilateral filtering—recursively decomposes multilateral (n-lateral) filtering by splatting to $(n-1)$-lateral filtering until it is a constant-time filter. Then, the results of constant-time filtering for the decomposed components are merged into the result of multilateral filtering.

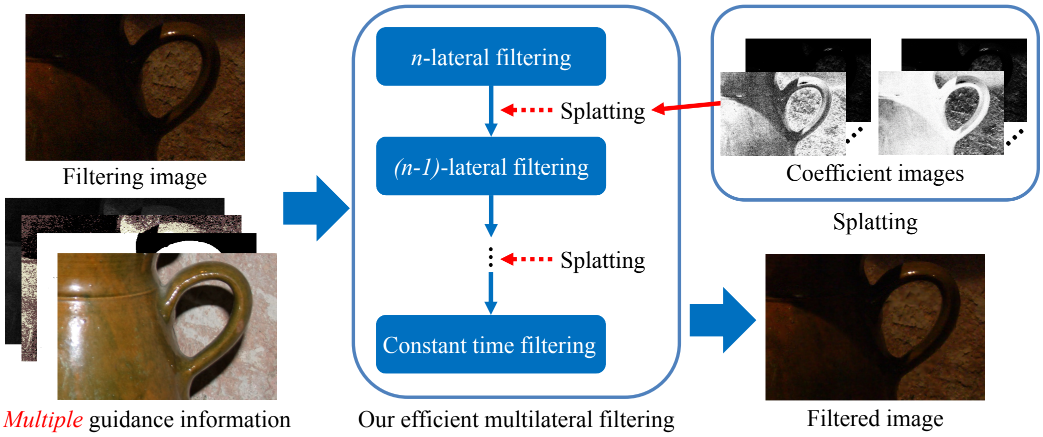


Figure 1. Algorithm overview. n-lateral filtering denotes multilateral filtering that multiplies a spatial filter and $n-1$ range filters. Examples of multiple guidance information are flash images, segmentation masks, and depth maps. The key point of the proposed algorithm is that it decomposes multilateral filtering into a set of constant-time filters. For more information on implementation, our code is available at https://fukushimalab.github.io/dmf/ (accessed on 16 January 2024).

The contributions of this paper are summarized as follows:

1. Introducing a constant-time algorithm for multilateral filtering (Section 5);
2. Extending various filters (e.g., guided image filtering [20] and domain transform filtering [33]) to deal with multiple guidance information (Section 6.1);
3. Proposing a multilateral extension to the filter that uses the filtering output as a guidance image, such as rolling guidance filters [52] (Section 6.2).

2. Related Work

Due to physical constraints, a single image sensor cannot simultaneously capture rich information such as resolution, wavelength range, focus, dynamic range, and scene features. Image fusion is one way to solve this problem. Research on image fusion is active, with the number of papers increasing each year, as well as many survey papers [53–62]. Image fusion involves smoothing, denoising, enhancement, sharpening, super-resolution, and blending for multiple signals to obtain the desired signal. Image fusion is mainly divided into digital photography image fusion and multimodal image fusion.

Digital photography image fusion combines images taken by the same sensor with different sensor settings and includes multi-focus image fusion, multi-exposure image fusion, multi-temporal image fusion, and multi-view image fusion. In multi-focus image fusion, an all-in-focus image is synthesized from images taken at different focus settings, and in multi-exposure image fusion, a wide dynamic range image is synthesized from images taken at different dynamic ranges. Multi-exposure image fusion also includes the use of different external flash environments. Multi-temporal image fusion synthesizes signals that vary along a time axis, while multi-view image fusion synthesizes signals from camera motion or multiple cameras capturing a scene.

Multimodal image fusion combines different characteristics of multiple sensors into one, including RGB-IR fusion, multi-hyperspectral-panchromatic image fusion, RGB-depth/LiDAR fusion, and medical image fusion (CT, PET, MRI, SPECT, X-ray), etc. In RGB-IR fusion, visible images are combined with IR images, taking advantage of the high contrast of IR and the good texture characteristics of RGB in the visible region. It also combines images using the different wavelength bands that can be captured by external flashes. In multi-hyperspectral-panchromatic image fusion, sensors that acquire images in different wavelength bands and resolutions are combined, and each sensor often has a different resolution and noise sensitivity. The objective is to improve the resolution and noise sensitivity of each sensor. RGB-depth/LiDAR fusion corrects depth sensor output from RGB images, including upsampling of depth information, missing interpolation, and contour correction noise reduction. Medical image fusion integrates the output of various medical sensors in the same dimension to assist in the diagnosis.

Among these image fusion methods, those that improve the acquisition signal are called sharpening fusion, which aims at signal denoising, sharpening, contrast improvement, and resolution improvement. In image fusion, various tools are used, such as weighted smoothing filtering, morphology filtering, principal component analysis (PCA), Laplacian pyramid, discrete cosine transformation (DCT), discrete Fourier transformation (DFT), discrete wavelet transform (DWT), etc. This paper is an extension of the weighted smoothing method. In particular, the proposed method extends existing smoothing/weighted smoothing methods to guided smoothing and has a wide range of applications.

3. Preliminaries

In this section, we review the previous work of constant-time bilateral filtering proposed by Yang et al. [42,43]. Bilateral filtering [28] is a representative edge-preserving smoothing filtering defined as a finite impulse response (FIR) manner. This filtering achieves edge-preserving effects by filtering in the range and spatial domains; thus, its filtering kernel weights are derived from a product of spatial and range weights based on a Gaussian distribution. Let input and output images be denoted as $I, O^{\mathrm{B}} : \mathcal{S} \to \mathcal{R}$, where $\mathcal{S} \subset \mathbb{Z}^D$ is the spatial domain, $\mathcal{R} = [0, n-1]^d \subset \mathbb{R}^d$ is the range domain, and d is the color range dimension (generally, $D = 2$, $N = 256$, and $d = 3$), respectively. Bilateral filtering is formulated as follows:

$$O^{\mathrm{B}}_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} f_{\mathrm{S}}(\boldsymbol{p}, \boldsymbol{q}) f_{\mathrm{R}}(I_{\boldsymbol{p}}, I_{\boldsymbol{q}}) I_{\boldsymbol{q}}}{\sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} f_{\mathrm{S}}(\boldsymbol{p}, \boldsymbol{q}) f_{\mathrm{R}}(I_{\boldsymbol{p}}, I_{\boldsymbol{q}})}, \tag{1}$$

where $\boldsymbol{p}, \boldsymbol{q} \in \mathcal{S}$ represents a target pixel and a neighboring pixel of $\boldsymbol{p}$, respectively. $I_{\boldsymbol{p}}, I_{\boldsymbol{q}} \in \mathcal{R}$ are pixel values at $\boldsymbol{p}, \boldsymbol{q}$. $\mathcal{N}_{\boldsymbol{p}} \subset \mathcal{S}$ is a set of neighboring pixels of $\boldsymbol{p}$. $f_{\mathrm{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, $f_{\mathrm{R}} : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ are weight functions based on the Gaussian distribution whose smoothing parameters are σ_S and σ_R, respectively. Here, we can formulate joint bilateral filtering [16,17] by replacing I in (1) with an arbitrary additional guidance image $J : \mathcal{S} \to \mathcal{R}$.

Naïve bilateral filtering is $\mathcal{O}(r^2)$ per pixel algorithm, where r is the filtering kernel radius; thus, the computational complexity increases exponentially when the kernel size is large. Several constant-time-per-pixel algorithms for bilateral filtering have been proposed to solve this problem. In particular, the algorithm proposed by Yang et al. [42,43] is the basis of the proposed method.

Yang et al. proposed a constant-time algorithm by extending the bilateral grid [40,41]. The algorithm decomposes bilateral filtering into a set of spatial filtering that can be computed in constant-time (e.g., box filter using integral image [63,64] and the recursive Gaussian filter [65–68]). The decomposition is conducted by computing principle bilateral filtered image components (PBFICs) [43] from the input or guidance image. Since arbitrary range filtering weights can generate PBFICs, the algorithm can compute the arbitrary bilateral filtering response in the range kernel.

Yang's algorithm [43] is further extended to apply to multichannel images in [42]. The extended algorithm computes multichannel images by preparing multichannel PBFICs with combinations of pixel values in each channel. However, this extension requires uniform processing for all channels. In other words, we cannot filter for each channel differently. This indicates that the algorithm is extendable when we compute multichannel or multiple guidance images with differential characteristics in each channel.

Our algorithm is inspired by Yang's algorithm [42,43], which represents bilateral filtering by a set of spatial filtering. In contrast, our algorithm decomposes a filter for multichannel images into arbitrary constant-time filters.

4. Relationship between Multilateral Filtering and Higher-Dimensional Filtering

In this section, we compare the filtering properties between multilateral filtering (MF) and high-dimensional filtering (HDF). The main difference between them is the logic to compute the filtering weight. The weight of multilateral filtering $f_M \in \mathbb{R}$ is computed by the multiplicative logic from spatial weight and range weights of multiple guidance images:

$$f_M(\boldsymbol{p},\boldsymbol{q}) = f_{\mathrm{S}}(\boldsymbol{p},\boldsymbol{q}) \prod_{i=1}^{m} f_{\mathrm{R}}^{i}(J_{\boldsymbol{p}}^{i}, J_{\boldsymbol{q}}^{i}), \tag{2}$$

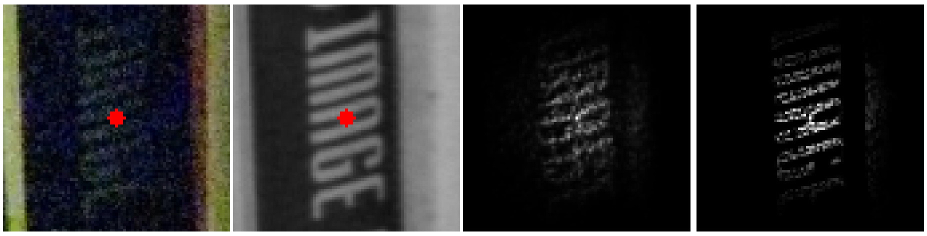
where $f_{\mathrm{R}}^{i} : \mathcal{R}^i \times \mathcal{R}^i \to \mathbb{R}$ is a filtering weight for the i-th guidance image $J^i : \mathcal{S} \to \mathcal{R}^i$, where $\mathcal{R}^i$ is the range domain of J^i. m is the number of guidance images. An early work on MF was proposed by Choudhury and Tumblin [32]. Each range weight f_{R}^{i} for the guidance image is individually defined to represent the characteristics of the image.

HDF's weight $f_H \in \mathbb{R}$ is computed by the additive logic:

$$f_H(\boldsymbol{p},\boldsymbol{q}) = \rho\left(\sum_{i=1}^{|\mathcal{V}_p|} \ell_\gamma(\mathcal{V}_{\boldsymbol{p}}(i) - \mathcal{V}_{\boldsymbol{p}}(i))\right), \tag{3}$$

where $\rho \in \mathbb{R}$ denotes an arbitrary weight function at the pixel $\boldsymbol{p}$; $\ell_\gamma \in \mathbb{R}$ denotes an arbitrary norm function; $\mathcal{V}_{\boldsymbol{p}}$ denotes higher-dimensional information consisting of spatial and range information, e.g., $\mathcal{V}_{\boldsymbol{p}} = (x_{\boldsymbol{p}}, y_{\boldsymbol{p}}, r_{\boldsymbol{p}}, g_{\boldsymbol{p}}, b_{\boldsymbol{p}})$ in RGB image, and $|\mathcal{V}_{\boldsymbol{p}}| \in \mathbb{Z}$ is the size of $\mathcal{V}_{\boldsymbol{p}}$. The work of Gastal and Oliveira [30] is a successful extension for HDF with multiple guidance information. They exploited additional guidance information to increase higher-dimensional information $\mathcal{V}$.

The two logics differ in terms of the severity of the restriction to compute the kernel weight; the multiplicative logic's restriction is more severe than the additive logic. The difference affects the edge-preserving performance. Figure 2 shows examples of HDF and MF weights. HDF assigns the low weights as a whole, even if the guidance pixel is hardly relevant to the target pixel. In contrast, MF assigns the low weights with the guidance pixel having a similar target pixel value. In this way, MF has a high edge preservation effect; hence, it is preferred when it is significant.



(**a**) RGB (**b**) Near-infrared (**c**) HDF weight (**d**) MF weight

Figure 2. Difference of filtering weights between HDF and MF. HDF weights are computed using the method in [30]. The red point represents the target pixel to compute the kernel weights.

5. Proposed Method: Decomposed Multilateral Filtering

The proposed filter of DMF first decomposes MF until constant-time filtering. This allows us to convert the computational complexity from $\mathcal{O}(r^2)$ to $\mathcal{O}(1)$ per pixel. This section defines MF and proves its decomposability. Algorithm 1 reviews the flow of DMF. Next, we discuss the extension of the algorithm.

5.1. Definition of Multilateral Filtering

MF assumes that the filtering weight is derived from the multiplicative logic discussed in Section 4. Furthermore, MF is equivalent to n-lateral filtering when $n-1$ guidance images are used for filtering. When $n = 1$ or 2, n-lateral filtering means spatial filtering or bilateral filtering, respectively. Therefore, we assume $n \geq 3$ in this section and compute n-lateral filtering output $O^n : \mathcal{S} \to \mathcal{R}$ as follows:

$$O_{\boldsymbol{p}}^n = \frac{\sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} f^n(\boldsymbol{p}, \boldsymbol{q}) I_{\boldsymbol{q}}}{\sum_{\boldsymbol{q} \in \mathcal{N}_{\boldsymbol{p}}} f^n(\boldsymbol{p}, \boldsymbol{q})}, \tag{4}$$

$$f^n(\boldsymbol{p}, \boldsymbol{q}) = f_{\mathrm{S}}(\boldsymbol{p}, \boldsymbol{q}) \prod_{i=1}^{n-1} f_{\mathrm{R}}^i(J_{\boldsymbol{p}}^i, J_{\boldsymbol{q}}^i) \qquad (n \geq 2), \tag{5}$$

$$f^1(\boldsymbol{p}, \boldsymbol{q}) = f_{\mathrm{S}}(\boldsymbol{p}, \boldsymbol{q}) \qquad (n = 1), \tag{6}$$

where f_{R}^i denotes the range filtering weight for i-th guidance image J^i. The n-lateral filtering weight is $f^n \in \mathbb{R}$. Equations (4) and (5) are the basic formulation of MF. Here, we basically define the first filter of f^1 as a spatial filter in (6), which is an arbitrary linear-time invariant (LTI) filter (e.g., box, circler, Gaussian, and Gabor filters) and linear-time variant (LTV) filters (e.g., spatially adaptive Gaussian filter [69]). LTI filters can be performed in $\mathcal{O}(1)$ by sliding DCT [68] filtering, but adaptive filtering has difficulty in converting $\mathcal{O}(1)$ filters.

Algorithm 1 Decomposed Multilateral Filtering

```
function $n$-LATERAL_FILTERING($n, \boldsymbol{J}, I$)
    // $\boldsymbol{J} = \{J^1, J^2, \dots, J^{n-1}\}$
    for all values $k$ $(0 \leq k \leq T^{n-1} - 1)$ do
        // Splatting as Equations (11) and (12)
        $W_{L_k^{n-1}} \leftarrow f_R^{n-1}(k, J^{n-1}) I$
        $K_{L_k^{n-1}} \leftarrow f_R^{n-1}(k, J^{n-1})$
        //$(n-1)$-lateral filtering
        if $n \geq 2$ then
            $W_{L_k^{n-1}} \leftarrow$ $n$-lateral_filtering($n-1, \boldsymbol{J}, W_{L_k^{n-1}}$)
            $K_{L_k^{n-1}} \leftarrow$ $n$-lateral_filtering($n-1, \boldsymbol{J}, K_{L_k^{n-1}}$)
        else
            // Final filtering step
            $W_{L_k^{n-1}} \leftarrow f_{\mathrm{S}} * W_{L_k^{n-1}}$
            $K_{L_k^{n-1}} \leftarrow f_{\mathrm{S}} * K_{L_k^{n-1}}$
        end if
        // Normalization as Equation (13)
        $C_{L_k^{n-1}} \leftarrow K_{L_k^{n-1}} / W_{L_k^{n-1}}$
    end for
    // Interpolation as Equation (15)
    //$\boldsymbol{L}^{n-1} = \{L_1^{n-1}, \dots, L_{T^{n-1}-1}^{n-1}\}$
    $O^n \leftarrow$ Interpolation($J^{n-1}, \boldsymbol{L}^{n-1}, C_{\boldsymbol{L}^{n-1}}$)
    return $O^n$
end function
```

5.2. Recursive Representation for Decomposed Multilateral Filtering

We introduce DMF and prove the decomposability of MF in this subsection. In (5), the n-lateral filtering weight f^n can be replaced with the product of its one-dimensional lower weight f^{n-1} and the range weight f_R^{n-1}:

$$f^n(\boldsymbol{p}, \boldsymbol{q}) = f^{n-1}(\boldsymbol{p}, \boldsymbol{q}) f_{\mathrm{R}}^{n-1}(J_{\boldsymbol{p}}^{n-1}, J_{\boldsymbol{q}}^{n-1}). \tag{7}$$

We can re-formulate MF from (4) using (7) as follows:

$$O^n_{\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} f^{n-1}(\boldsymbol{p},\boldsymbol{q}) f^{n-1}_R(J^{n-1}_{\boldsymbol{p}}, J^{n-1}_{\boldsymbol{q}}) I_{\boldsymbol{q}}}{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} f^{n-1}(\boldsymbol{p},\boldsymbol{q}) f^{n-1}_R(J^{n-1}_{\boldsymbol{p}}, J^{n-1}_{\boldsymbol{q}})}. \tag{8}$$

This form shows that we can express n-lateral filtering using $(n-1)$-lateral filtering weight. Furthermore, we deform (8) by the additional assumption that the pixel values of the guidance images are discrete. Let $c \in \mathcal{R}^{n-1} = [0, N^{n-1} - 1]$ be a constant value, where N^{n-1} is the number of tones in the range of the $(n-1)$-th guidance image. When a pixel value in the n-th guidance image $J^{n-1}_{\boldsymbol{p}}$ in (8) is replaced by a constant value c, it is rewritten as follows:

$$C^{n-1}_{c,\boldsymbol{p}} = \frac{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} f^{n-1}(\boldsymbol{p},\boldsymbol{q}) f^{n-1}_R(c, J^{n-1}_{\boldsymbol{q}}) I_{\boldsymbol{q}}}{\sum_{\boldsymbol{q}\in\mathcal{N}_{\boldsymbol{p}}} f^{n-1}(\boldsymbol{p},\boldsymbol{q}) f^{n-1}_R(c, J^{n-1}_{\boldsymbol{q}})} \tag{9}$$

$$O^n_{\boldsymbol{p}} = C^{n-1}_{v,\boldsymbol{p}} \ \text{ s.t. } \ v = \arg\min_x \|x - J^{n-1}\|_1, \tag{10}$$

where $\|\cdot\|_1$ is ℓ_1 norm operator. We call $C^{n-1}_c : \mathcal{S} \to \mathcal{R}$ a component image of n-lateral filtering, and its pixel value at $\boldsymbol{p}$ is denoted by $C^{n-1}_{c,\boldsymbol{p}} \in \mathcal{R}$.

$f^{n-1}_R(c, J^{n-1}_{\boldsymbol{q}})$ and $f^{n-1}_R(c, J^{n-1}_{\boldsymbol{q}}) I_{\boldsymbol{q}}$ in (9) can be cached as images in constant-time; hence, we express these coefficients as the following images:

$$W^{n-1}_{c,\boldsymbol{q}} = f^{n-1}_R(c, J^{n-1}_{\boldsymbol{q}}) \tag{11}$$

$$K^{n-1}_{c,\boldsymbol{q}} = W^{n-1}_{c,\boldsymbol{q}} I_{\boldsymbol{q}}, \tag{12}$$

where $W^{n-1}_c : \mathcal{S} \to \mathcal{R}^{n-1}$ and $K^{n-1}_c : \mathcal{S} \to \mathcal{R}$ are the elements of the denominator and numerator in (9), respectively. We call the processes for Equations (11) and (12) as splatting followed by the paper [47,48], and we call the images as coefficient image. For simplification, we rewrite (9) using these coefficient images and the convolution operator $*$:

$$C^{n-1}_c = \frac{f^{n-1} * K^{n-1}_c}{f^{n-1} * W^{n-1}_c}, \tag{13}$$

where the pixel operator $\boldsymbol{p}$ can be dropped. Figure 3 shows the splatting procedure in DMF.

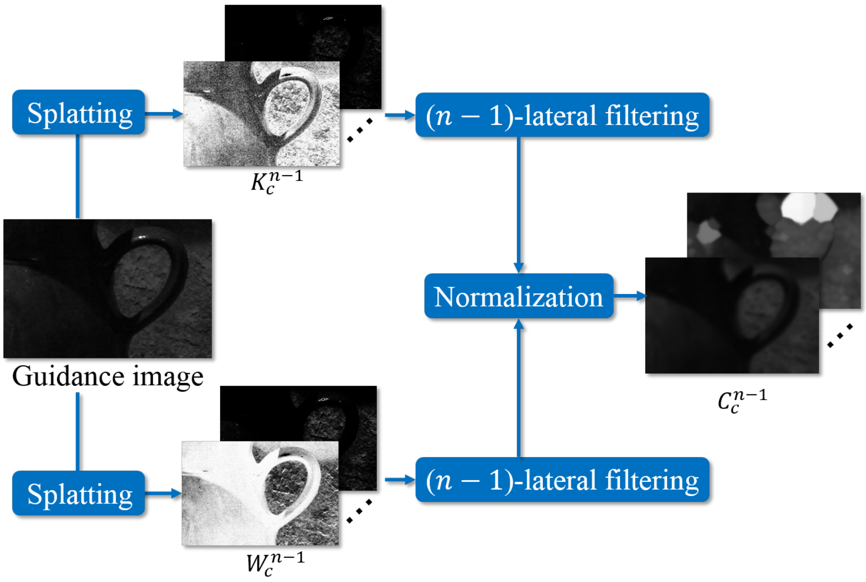


Figure 3. Procedure of splatting in DMF.

The denominator and numerator in (13) represent $(n-1)$-lateral filtering. This indicates that n-lateral filtering has been decomposed into $(n-1)$-lateral filtering. Therefore, MF can be decomposed recursively:

$$f^n = g^{n-1} \circ f^{n-1} = g^{n-1} \circ g^{n-2} \circ \cdots \circ g^1 \circ f^1, \tag{14}$$

where $g^x \circ$ denotes a decomposing operator, as described in Equations (11)–(13). Equation (14) summarizes the DMF formulation. Since f^1 (e.g., Gaussian filtering) can be computed in constant-time per pixel by recursive algorithms, and the decomposing operation is independent of kernel size, DMF can also be computed in constant-time.

5.3. Tonal Range Subsampling

The exact filtering result can be obtained by computing coefficient images for all values $c \in \mathcal{R}^{n-1}$ ranged in the guidance image J^{n-1}. Here, we increase efficiency by quantizing the guidance tonal ranges. Let $\mathcal{L}^{n-1}$ be a quantized set of $\mathcal{R}^{n-1}$, and $T^{n-1} = |\mathcal{L}^{n-1}|$ be the number of tones in a quantized tonal range of the $(n-1)$-guidance image, where $T^{n-1} \leq |\mathcal{R}^{n-1}|$. Furthermore, let $L^{n-1}_k \in \mathcal{L}^{n-1}$ be the k-th label's value, where $k \in \{0, 1, \ldots, |T^{n-1}|\}$ that the return value is in the quantized range domain. We can obtain the final output of DMF by linear interpolation of the current and next coefficient images (i.e., $C^{n-1}_{L^{n-1}_k}$ and $C^{n-1}_{L^{n-1}_{k+1}}$):

$$O^n_{\boldsymbol{p}} \approx (L^{n-1}_{k+1} - J^{n-1}_{\boldsymbol{p}}) C^{n-1}_{L^{n-1}_k, \boldsymbol{p}} + (J^{n-1}_{\boldsymbol{p}} - L^{n-1}_k) C^{n-1}_{L^{n-1}_{k+1}, \boldsymbol{p}} \tag{15}$$

$$\text{s.t. } \; k = \underset{x \in \mathcal{L}^{n-1}}{\arg\min} \, \| J^{n-1}_{\boldsymbol{p}} - L^{n-1}_x \|_1. \tag{16}$$

5.4. Spatial Domain Subsampling

Considering the sparsity of the coefficient images, we can also apply subsampling in the spatial domain for further increasing efficiency, as discussed in [41,42]. In the DMF case, we can apply spatial subsampling to the coefficient images in several steps: the first and the second decomposition. If we apply spatial subsampling to DMF in the n-lateral filtering splatting, the process is computed as:

$$\begin{aligned} K^{n-1\downarrow}_c &= \mathrm{downsample}(K^{n-1}_c) \\ W^{n-1\downarrow}_c &= \mathrm{downsample}(W^{n-1}_c) \qquad (17) \\ C^{n-1\downarrow}_c &= \frac{f^{n-1} * K^{n-1\downarrow}_c}{f^{n-1} * W^{n-1\downarrow}_c} \qquad (18) \\ C^{n-1}_c \approx C^{n-1\downarrow\uparrow}_c &= \mathrm{upsample}(C^{n-1\downarrow}_c), \qquad (19) \end{aligned}$$

where $X^{\downarrow}$ and $X^{\downarrow\uparrow}$ $(X = \{K, W, C\})$ are the downsampled and upsampled images, respectively. We use the average nearest-neighbor pixels and linear interpolation for subsampling and upsampling, respectively. Our method can apply different ratios of spatial subsampling to arbitrary guidance channels based on the sparsity of each channel (e.g., YUV image components of JPEG and MPEG format, RGB-D images). The flexibility is an advantage for Yang's algorithm [42].

6. Extension of Decomposed Multilateral Filtering

6.1. Beyond Gauss Transform

DMF can deal with any multilateral filtering responses and does not limit the Gauss transform [70], which is the combination of Gaussian filtering. DMF can select arbitrary ranges and spatial filters. Furthermore, we can select the filtering responses by changing the final filtering step.

We should have DMF until the spatial filtering in (6). In contrast, DMF does not always require decomposition until it is spatial filtering. Specifically, we can apply any joint edge-preserving filters for the final filtering step while decreasing the number of decompositions. Some edge-preserving filtering can handle multichannel signals in the designed weight function. For example, high-dimensional Gaussian filtering handles multichannel signals by the Gaussian distribution with the Euclid distance; instead, domain transform filtering uses the geodesic distance. Guided image filtering handles them by the local linear model with ℓ_2 norm between signals.

Therefore, when the final filter is performed in edge-preserving filtering, the DMF decomposition can be reduced by the number of dimensions handled by the edge-preserving filtering. Let the handling signal set be $\boldsymbol{G} = \{J_s, \ldots, J_1\}$, where s is the number of handling channels (i.e., $s = 3$ in the RGB image case). Using edge-preserving filtering, (13) of the final step is replaced as:

$$C_c^s = \frac{H_{\boldsymbol{G}} * K_c^s}{H_{\boldsymbol{G}} * W_c^s}, \tag{20}$$

where $H_{\boldsymbol{G}}*$ represents any joint edge-preserving filtering with the guidance signal set as $\boldsymbol{G}$. Examples of the final step filtering are high-dimensional filtering (high-dimensional Gaussian filtering [47,48], guided image filtering [20,71], domain transform filtering [33], adaptive manifold filtering [30]), frequency transform filtering (edge-avoiding wavelet [72,73], redundant frequency transform [74]), adaptive filtering (range parameter adaptive filtering [75]), enhancement filtering (local Laplacian filtering [76–78]), statical filtering (fast guided median filtering [79]), LUT-based filtering [80], optimization-based filtering (weighted least square optimization [81,82], and L0 smoothing optimization [83]).

The representation allows us to extend various edge-preserving filtering methods to handle multiple guidance images. This fact is helpful for various applications since the required filtering properties, e.g., local linearity [20] and geodesic distance [33], are different by application. Note that it has the potential to be faster because of the merged treatment of dimensions, but the filter may not work well if the characteristics of the combined set are not identical.

6.2. Multilateral Rolling Guidance Filtering

MF is also helpful in self-generating multiple guidance information from single guidance information, such as rolling guidance filtering [52]. Rolling guidance filtering is iteratively processed using the filtered image as the guidance image. In this regard, the filtering image is fixed as the input image and the guidance image varies. This iterative representation can be applied to multilateral filtering with some modifications. We call it multilateral rolling guidance filtering (MRGF) and show the actual processing in Figure 4. The main difference is the filter output as an additional guidance image.

MRGF is specifically practical when edge information is essential, such as image segmentation and feathering. Since we can reflect the smoothed or refined results to the filtering target image, it tends to refine the desirable features for the target image. Significantly, the first estimated maps of scene properties often contain noises and errors. MRGF has good performance in the refinement of maps. We verify the performance of MRGF in the following experimental part.

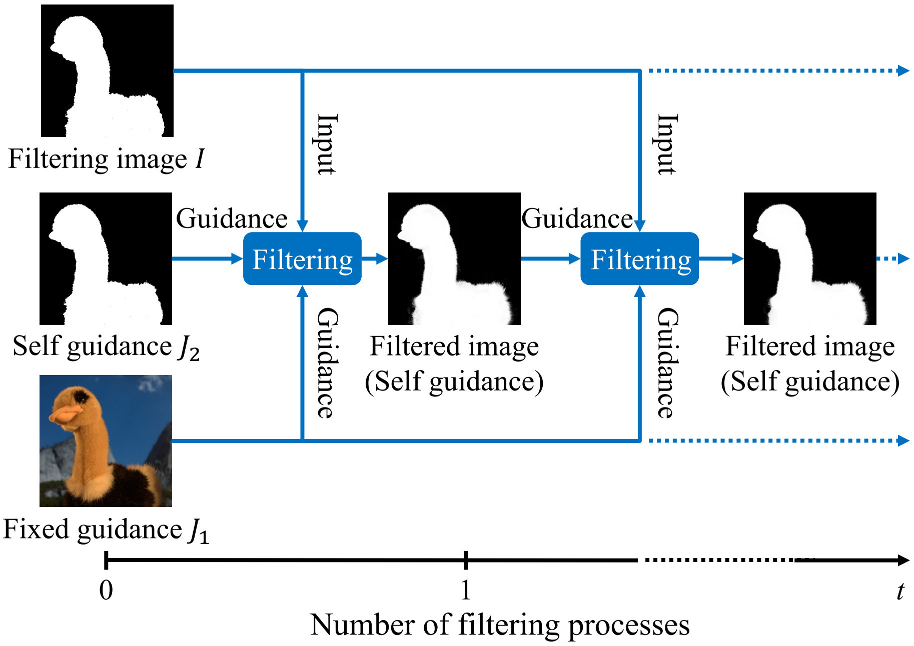


Figure 4. Multilateral rolling guidance filtering. Note that the constraining information J_2 is the same as the filtering image I when the filtering process is the first time.

7. Experimental Results

We evaluated the proposed filter of DMF in terms of accuracy and efficiency. The implementations for DMF and competitive methods are written in C++, and the codes are parallelized by OpenMP and vectorized by AVX2. We used Intel Core i7 7700K (four cores, eight threads) and Visual Studio2022 compiler for experiments.

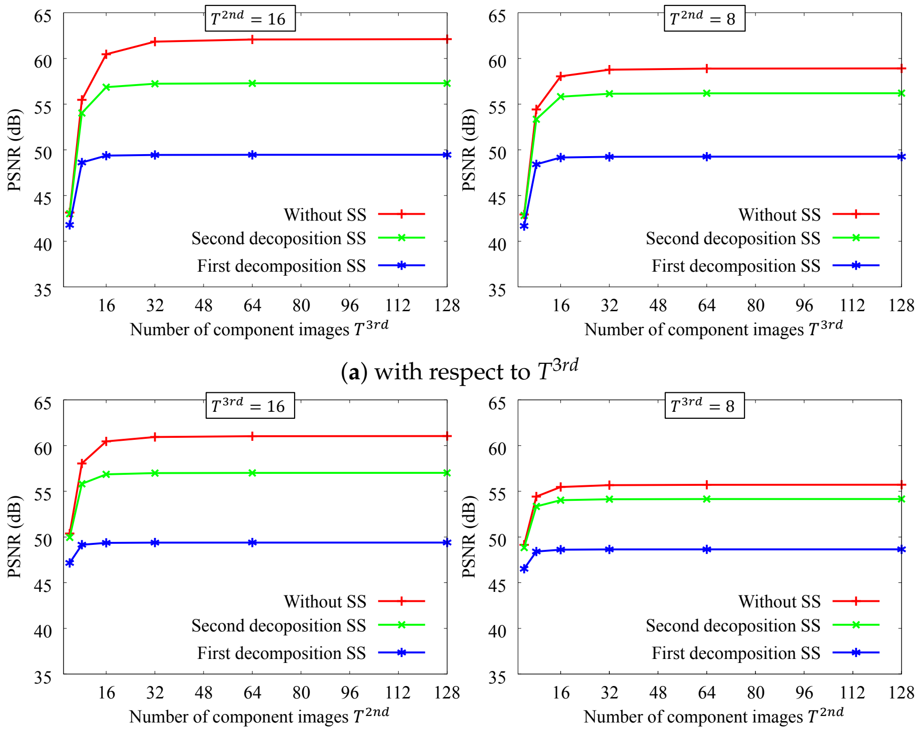
7.1. Accuracy Evaluation

First, we evaluated how much the DMF result corresponds to the naïve implementation of the MF result. In our experiments, we applied our algorithm to trilateral filtering [29]. Note that the Gauss transform is applied to the trilateral filtering weights for spatial and range weights, where the standard deviations are σ_s, σ_{r1}, and σ_{r2}. Here, σ_{r1} and σ_{r2} are the parameters for the tonal ranges of the guidance and filtering images, respectively. We apply recursive Gaussian filtering with sliding DCT [68,84] for spatial filtering. We evaluate the accuracy of our algorithm by flash/no-flash denoising [16,17]. The range kernels f_{R_1} and f_{R_2} for MF are computed from flash images and no-flash images, respectively. We use the peak signal-to-noise ratio (PSNR) [85,86] as the objective evaluation method of the approximation accuracy between naïve results and the approximation results. The evaluation formula used is as follows:

$$PSNR = 10 \log_{10} \frac{S \cdot 255^2}{\|\mathbf{A} - \mathbf{G}\|_2^2}, \tag{21}$$

where $\|\cdot\|_2$ is the ℓ_2 norm operator, $\mathbf{A}$ is approximated signals proceded by DMF, and $\mathbf{G}$ is ground truth signals produced by naïve MF. S is the number of the elements in the signal $\mathbf{A}$ and $\mathbf{G}$, $S = |\mathbf{A}| = |\mathbf{G}|$, where $|\cdot|$ returns the number of vector elements.

Figure 5 shows the results of the filtering accuracy in terms of the number of coefficient images. Note that T^{2nd} and T^{3rd} are the tonal-quantized numbers of coefficient images for the flash and no-flash images, respectively. The filtering accuracy in each case is high overall. For this result, eight coefficient images are enough. This trend is also the same in spatial subsampling, and spatial subsampling is practical because the PSNR accuracy is over 45 dB. It can be seen that the PSNR degradation is more significant when downsampling at the first decomposition. As shown in the next section, downsampling at the first stage has a greater speedup effect; thus, it is up to the application to decide which one to choose.



(**a**) with respect to T^{3rd}

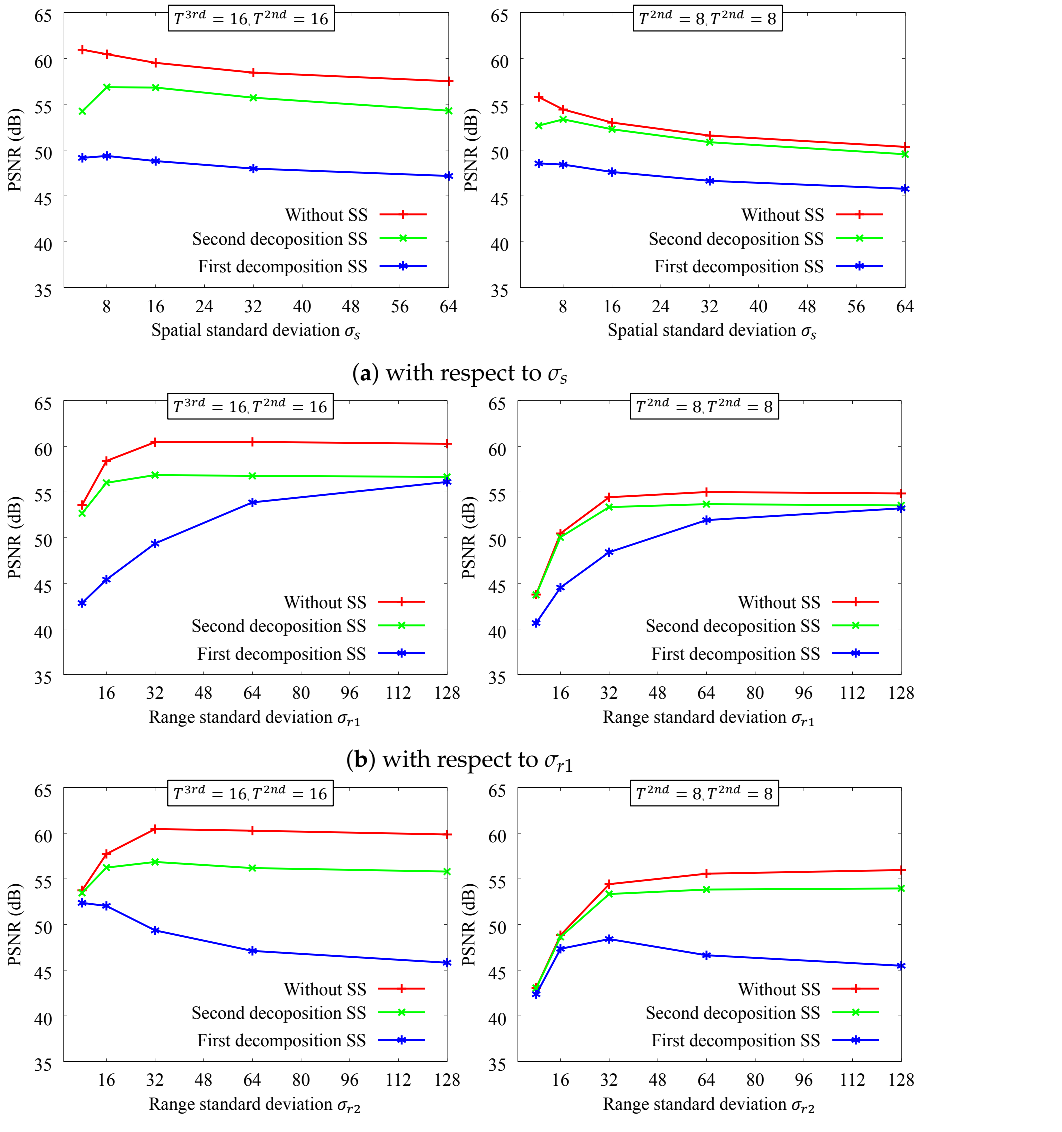
(**b**) with respect to T^{2nd}

Figure 5. PSNR accuracy with respect to the number of coefficient images. The parameters are $\sigma_{r1} = 32$, $\sigma_{r2} = 32$, $\sigma_s = 8$. SS denotes spatial subsampling rates, $\times\frac{1}{16}$. We tested 4 images for the input image.

Figure 6 shows the filtering accuracy of the smoothing parameters. Although the accuracy of DMF varies depending on the parameters, ours has a high accuracy. We can see that it is not very sensitive to changes in spatial parameters σ_s. Each of the two guides has similar range parameters σ_{r1} and σ_{r2}. The smaller the range parameter, the lower the approximation accuracy tends to be, and the effect is more pronounced when the number of decompositions is small. Initial subsampling is also susceptible to this effect. However, the proposed method has an approximation accuracy that is generally better than 45 dB for all parameters, which is sufficient because it becomes difficult for a person to distinguish between two images at around 40 dB [87].

7.2. Efficiency Evaluation

We compare the computational time in two cases for efficiency evaluation. One is a comparison between naïve MF and DMF combined with some edge-preserving filters. Another is a comparison with/without subsampling. In this experiment, we apply real-time bilateral filtering (RTBF) [42,43], guided image filtering (GIF) [20] and domain transform filtering (DTF) [33] to (20). Note that we call DMF with these filters *DMF-Gauss*, *DMF-GF*, and *DMF-DTF*, respectively. Since RTBF can be interpreted as DMF with one guide image for Gaussian filtering, this is equivalent to DMF with two guide images for Gaussian filtering. Therefore, we refer to it as DMF-Gauss. For this experiment, flash and no-flash images were converted to grayscale, and the RGB no-flash image was filtered. In DMF-Gauss, the two-channel images were used as guides; in DMF-DFT and DMF-GIF, the no-flash images were used as guides for DMF, and the flash images were used as guides for DTF and GIF. Note that joint filtering is available for DTF and GIF. Cache-efficient filtering was computed using a one-pass version of Gaussian filtering for DMF-Gauss and box filtering for GIF [64].



(**a**) with respect to σ_s

(**b**) with respect to σ_{r1}

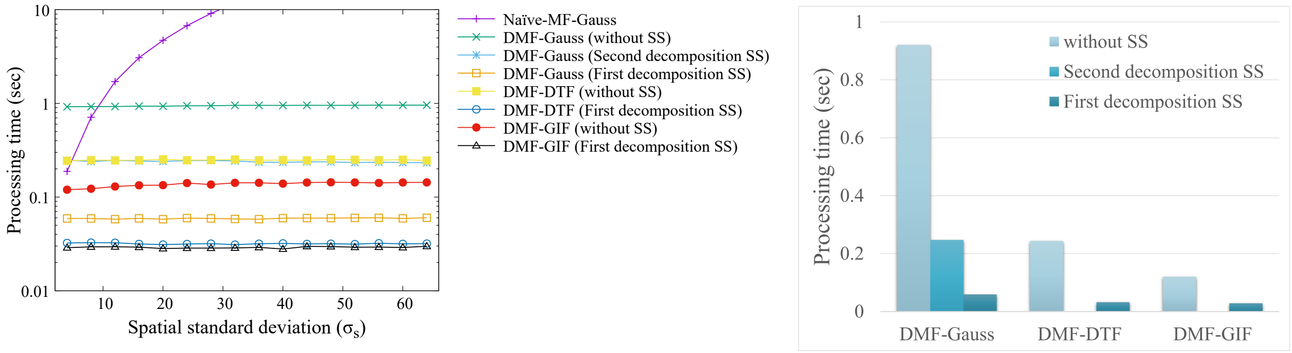
(**c**) with respect to σ_{r2}

Figure 6. PSNR accuracy with respect to smoothing parameters. (**a**) $\sigma_{r1} = 32$ and $\sigma_{r2} = 32$. (**b**) $\sigma_s = 8$ and $\sigma_{r2} = 32$. (**c**) $\sigma_s = 8$ and $\sigma_{r1} = 32$. SS denotes the spatial subsampling rate, which is $\times\frac{1}{16}$. The input images are the same as in Figure 5.

Figure 7 shows the processing time results. The processing time of the naïve ML-Guass increases exponentially as the filtering kernel size increases, whereas DMF can be computed in constant-time from Figure 7a. DMF is especially efficient when we use GIF or DTF for the filtering step described in Section 6.1. Furthermore, DMF becomes more efficient by subsampling the spatial domain as shown in Figure 7b. Since DMF and GIF are not decomposable by the proposed method, only a one-step decomposition is possible. Therefore, the computation time for the second decomposition subsampling has not been reported.

7.3. Denoising Performance Evaluation

Here, we evaluate the denoising performance of the proposed method; note that it is not the approximation accuracy evaluated by Section 7.1. In our experiments, we used RGB-IR images and simulated RGB-IR fusion by adding noise to the RGB images. Performance is evaluated in terms of PSNR for the noiseless RGB image and the de-noised image; the IR image is not evaluated in terms of PSNR because it is not noiseless and is not the final visible image.



(**a**) w.r.t. filtering kernel size (**b**) with/without subsampling

Figure 7. Processing time. The input image size is 1 megapixel (1024 × 1024). The parameters are $T^{3rd} = 8$, and $T^{2nd} = 8$. SS denotes spatial subsampling rate ($\frac{1}{16}$).

The comparison methods are redundant DCT denoising (DCT) [74], domain transform filtering (DTF) [33], guided image filtering (GIF) [20], cross-field joint image restoration (CFJIR) [88] and high-dimensional kernel filtering (HDKF) [37]. DCT, DTF, and GIF are extended by the proposed method to handle an additional guidance IR image, named DMF-DCT, DMF-DTF, and DMF-GIF. These methods were chosen for their high-speed performance. CGJIR and HDKF already use the characteristics of the guide image; thus, the proposed method extension is ineffective. For evaluation images, we used the RGB-IR dataset, which includes ten images [37] (https://norishigefukushima.github.io/TilingPCA4CHDGF/ (accessed on 16 January 2024)).

Tables 1–3 show PSNR results for each method in different noise levels. It can be seen that the classical method, DCT, has the best performance on average for all noise levels due to the DMF extension. CGJIR and HDKF are new dedicated methods for RGB-IR denoising, and performance comparable to these methods has been achieved by extending this method. All DMF extensions also show a steady improvement in performance.

7.4. Channel Perfomance Evaluation

Here, we evaluated the denoising effect of the number of channels for flash and no-flash images. In Section 7.1, guide images are grayscaled 2-channel, but here, we use two RGB images, 6-channel. In addition, the number of channels is controlled by using PCA dimensionality compression for guide images [37,89]. Note that the denoising performance is different from the approximation performance. We used a flash/no-flash image dataset [37], which contains ten images. Images are filtered by multilateral filtering. In addition to PSNR, we used structural similarity (SSIM) [90], which is a more robust quality metric. Noise was only added in no-flash images.

Table 1. PSNR accuracy of denoising (dB). The Gaussian noise parameter is $\sigma = 10$. * Filters that already use the characteristics of the guide image. Bold numbers mean the best results.

Image	DCT	DMF-DCT	DTF	DMF-DTF	GIF	DMF-GIF	CFJIR *	HDKF *
0	**36.73**	**36.73**	34.41	35.11	34.35	35.26	35.22	35.55
1	33.23	33.41	31.37	31.97	32.34	33.14	32.13	**33.53**
2	**36.16**	36.13	34.50	34.72	34.45	35.10	34.54	35.67
3	39.56	**40.54**	37.53	37.88	37.30	39.38	40.52	40.49
4	36.31	36.47	34.11	34.73	33.90	35.97	35.46	**36.77**
5	**35.51**	35.46	33.76	34.26	33.82	34.68	34.38	34.57
6	**34.06**	33.99	32.21	32.85	32.49	33.25	32.93	33.05
7	36.56	36.69	34.69	35.25	34.60	37.28	37.52	**37.70**
8	**34.50**	34.38	33.04	33.67	33.56	32.99	32.80	33.42
9	36.58	36.82	33.28	34.10	33.77	37.03	37.49	**37.62**
Average	35.92	**36.06**	33.89	34.45	34.06	35.41	35.30	35.84

Table 2. PSNR accuracy of denoising (dB). The Gaussian noise parameter is $\sigma = 20$. * Filters that already use the characteristics of the guide image. Bold numbers mean the best results.

Image	DCT	DMF-DCT	DTF	DMF-DTF	GIF	DMF-GIF	CFJIR *	HDKF *
0	32.81	**32.89**	29.90	31.42	29.98	31.91	32.26	32.09
1	29.50	29.92	27.76	28.82	28.28	29.75	29.50	**30.31**
2	31.64	**32.28**	29.47	30.34	29.66	31.44	30.86	31.60
3	36.24	37.34	34.18	34.58	33.45	36.04	**38.59**	36.54
4	32.45	32.85	29.18	31.05	29.09	32.31	32.42	**33.13**
5	31.67	**31.85**	29.56	30.49	29.63	31.02	31.05	31.07
6	30.03	**30.22**	27.50	29.07	27.77	29.90	29.84	29.77
7	32.92	33.27	31.30	32.24	31.03	34.03	**34.98**	34.31
8	30.77	**30.99**	28.56	29.97	29.07	30.08	29.99	30.32
9	32.61	33.21	29.15	30.74	29.32	33.51	**34.47**	34.02
Average	32.06	**32.48**	29.66	30.87	29.73	32.00	32.40	32.32

Table 3. PSNR accuracy of denoising (dB). The Gaussian noise parameter is $\sigma = 30$. * Filters that already use the characteristics of the guide image. Bold numbers mean the best results.

Image	DCT	DMF-DCT	DTF	DMF-DTF	GIF	DMF-GIF	CFJIR *	HDKF *
0	30.35	**30.55**	27.24	29.50	27.61	30.21	30.42	30.16
1	27.64	**28.28**	26.03	27.44	26.42	28.27	27.86	28.15
2	28.84	**29.83**	26.08	27.63	26.88	29.12	28.44	28.75
3	33.93	35.01	32.54	33.00	30.82	34.99	**36.41**	34.29
4	30.18	**30.88**	26.28	29.36	26.58	30.63	30.36	30.48
5	29.32	**29.71**	26.79	28.07	27.27	28.48	29.05	29.09
6	27.72	**28.18**	24.65	27.09	25.39	27.88	28.00	27.84
7	30.64	31.19	28.94	30.51	28.82	32.60	**32.85**	31.71
8	28.68	**29.16**	25.98	28.12	26.71	28.31	28.45	28.73
9	30.14	31.22	26.94	29.22	27.18	**32.17**	31.80	30.82
Average	29.74	**30.40**	27.15	28.99	27.37	30.27	30.36	30.00

Tables 4 and 5 show the results for each metric. On average, the optimal value is taken by four channels in every metric. The SSIM, which is said to have a high human subjective evaluation value, shows that the value is high enough even for two channels.

Table 6 shows the computational time. The computation time increases exponentially with the number of channels. This indicates that we are suffering from the curse of dimensionality. Therefore, it is better to have as small a number of channels as possible.

Table 4. PSNR accuracy metrics where **higher** values indicate better flash/no-flash denoising with PCA (dB). Gaussian noise parameter for no-flash images is $\sigma = 10$. Bold numbers mean the best results.

	Noise	1	2	3	4	5	6
0	28.82	30.92	30.96	31.84	**33.33**	**33.33**	33.32
1	28.47	33.95	34.55	36.34	**36.35**	36.34	36.33
2	28.16	39.91	40.91	**41.32**	41.30	41.29	41.28
3	28.27	37.66	39.56	**39.80**	**39.80**	39.79	39.79
4	28.20	36.34	38.01	38.70	**38.80**	38.76	38.74
5	28.16	38.35	40.43	**40.73**	40.70	40.67	40.66
6	28.18	40.85	42.20	**42.31**	42.29	42.27	42.27
7	28.26	37.44	39.48	39.84	**39.85**	**39.85**	39.84
8	28.28	35.19	37.38	37.52	**37.60**	37.59	37.59
9	28.15	39.35	**41.42**	41.40	41.38	41.36	41.35
Average	28.30	37.00	38.49	38.98	**39.14**	39.12	39.12

Table 5. SSIM accuracy metrics where **higher** values indicate better flash/no-flash denoising with PCA. Gaussian noise parameter for no-flash images is $\sigma = 10$. Bold numbers mean the best results.

	Noise	1	2	3	4	5	6
0	0.932	0.965	0.965	0.967	**0.975**	0.974	0.974
1	0.863	0.970	0.974	**0.984**	**0.984**	0.983	0.983
2	0.723	0.983	0.984	**0.987**	**0.987**	**0.987**	**0.987**
3	0.751	0.973	**0.983**	**0.983**	**0.983**	**0.983**	**0.983**
4	0.769	0.977	**0.985**	**0.985**	**0.985**	0.984	0.984
5	0.693	0.975	**0.985**	**0.985**	**0.985**	0.984	0.984
6	0.650	0.974	**0.984**	0.983	0.983	0.982	0.982
7	0.762	0.976	0.984	**0.985**	**0.985**	**0.985**	**0.985**
8	0.796	0.961	**0.981**	**0.981**	0.980	0.980	0.980
9	0.676	0.973	**0.984**	**0.984**	**0.984**	0.983	0.983
Average	0.762	0.973	0.981	0.982	**0.983**	**0.983**	**0.983**

Table 6. Processing time of filtering with multi-channel guide images.

Channels	Time (msec)
1	29.7
2	79.1
3	455.9
4	3380.4
5	27,224.7
6	224,146.0

7.5. Memory Usage Analysis

The memory requirement has linear relations in the number of pixels N_p. The number of tones N_t has exponential relations in the number of channels N_c and multiple guidance images N_j. Consequently, the amount of memory required is $\mathcal{O}(N_p N_t^{N_c N_j})$, according to Algorithm 1.

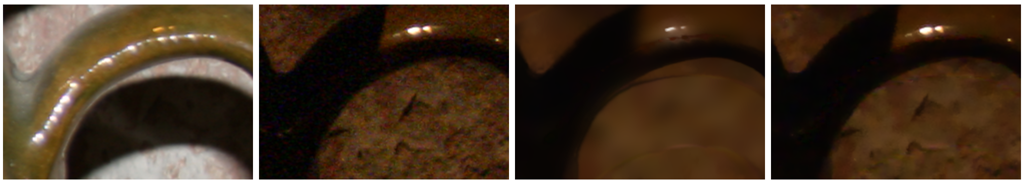
The vast memory requirement is one of the limitations, and the limitation is inherited from previous work [42,43]. However, tonal range and spatial domain subsampling can moderate the memory requirement. We can also make memory requirements independent of the number of channels by processing DMF, as discussed in [42,43]. The implementation, however, loses parallelizability and computational efficiency somewhat.

8. Multilateral Filtering for Computational Photography

We verify the effectiveness of MF and DMF by applying several applications of sensor fusion in computational photography.

8.1. Flash/No-Flash Denoising

Flash/no-flash denoising [16,17] is the representative application for edge-preserving filtering with multiple guidance images. Flashing sometimes causes false edges (e.g., appearance/disappearance of shadow edges), as shown in Figure 8a. Joint bilateral filtering can remove noise in the no-flash image, but it simultaneously preserves the false edges of the flash image (Figure 8c). The conventional method requires multiple steps [17] to solve the problem, while MF requires only one step. As shown in Figure 8d, MF can remove noise while preventing false edges from being transferred. Note that we used the value information in the HSV color space of the no-flash image and the color-flash image as the guidance images. In addition, our algorithm can be efficiently computed by applying efficient edge-preserving filtering, such as domain transform filtering.

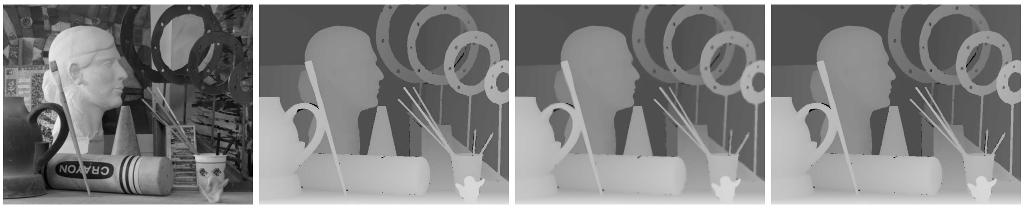


(**a**) Flash image (**b**) No-flash image (**c**) Joint bilateral filter (**d**) DMF-DTF

Figure 8. Flash/no-flash denoising without the false edge in the flash image. The parameters are $\sigma_{r1} = 64$ (for joint bilateral filtering and DTF), $\sigma_{r2} = 16$, $\sigma_s = 8$, $T^{3rd} = 16$.

8.2. Depth Map Refining

Trilateral filtering is effective for refining degraded depth maps by lossy compressing [29]. In the case of a single guidance image, the object boundaries in the depth maps are blurred even if the artifacts are removed; hence, there is a trade-off between denoising performance and the edge preservation effect (Figure 9c). MF can improve this problem by considering both the edges in the depth map and the guidance image (Figure 9d). Although this depth-refinery experiment targets removing coded artifacts, MF can also be applied to noise removal for a depth sensor.



(**a**) Guidance image (**b**) Coded depth map (**c**) Joint bilateral filter (**d**) DMF-RTBF

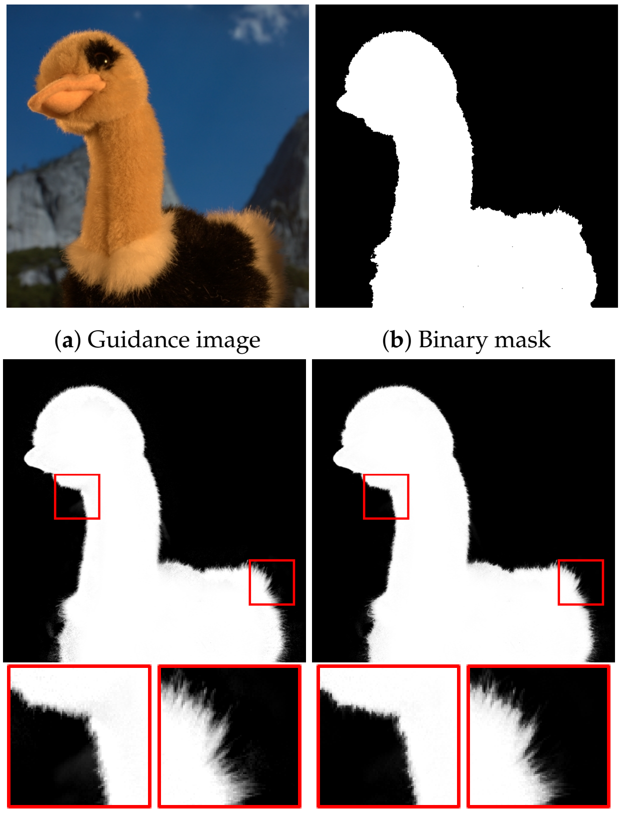
Figure 9. Depth map refining. (**b**) is coded by JPEG (quality factor = 50). The parameters are $\sigma_{r1} = 16$, $\sigma_{r2} = 16$, $\sigma_s = 2$, $T^{3rd} = 16$ and $T^{2nd} = 16$. The values of the ratio of bad pixels [25] (error threshold is 1.0) in (**b**–**d**) are 12.47, 9.24, and 5.38, respectively.

8.3. Feathering

We demonstrate the property of our algorithm beyond the Gauss transforms. Guided feathering [20] refines a binary mask for alpha mating near the boundary of the object. For guided feathering, guided image filtering has excellent performance in terms of efficiency and accuracy [20]. The result of naïve guided image filter is shown in Figure 10c. We can confirm that the feather can be computed in detail; however, several noises are simultaneously caused near the object boundary regions. This is because the local linear model of the guided image filter is violated. By contrast, MRGF results hardly include such noises, as shown in Figure 10d, while the detailed feather is computed. This result indicates that MRGF prevents the violation.

8.4. Haze Removing

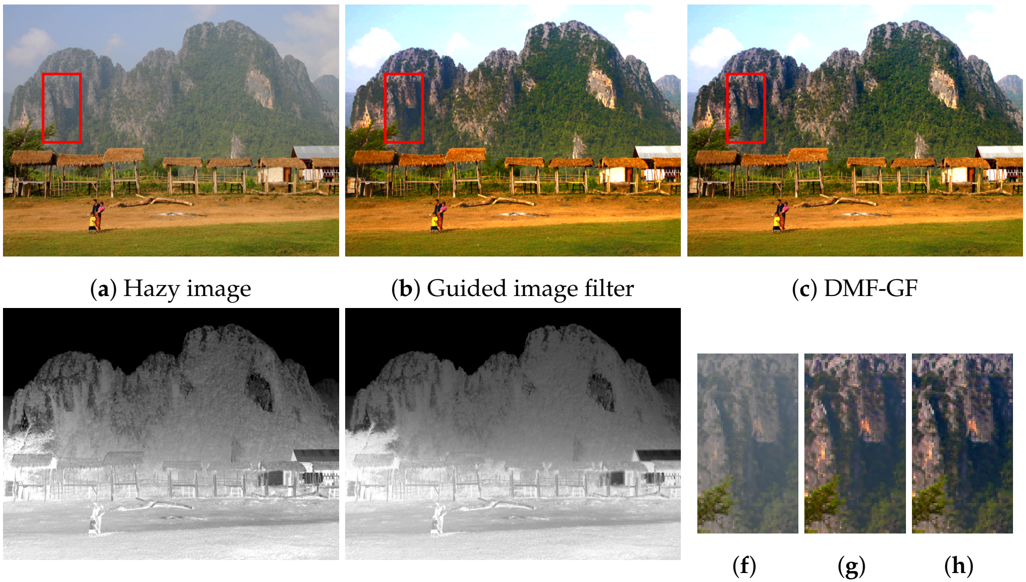
Furthermore, our algorithm with guided image filtering is also effective for haze removal [21]. The large kernel size for filtering is required in haze removal; thus, guided image filtering violates the local linear model, as well as guided feathering. Consequently, some haze remains in Figure 11b. On the contrary, MRGF with guided image filtering can suppress expansion in different objects, as shown in Figure 11c.



(**a**) Guidance image (**b**) Binary mask

(**c**) Guided filter (**d**) DMF-GF

Figure 10. Guided feathering. MRGF's results are in Figure 4. The parameters are $r = 20$, $\epsilon = 10^{-6}$, $\sigma_r = 160$ and $T^{3rd} = 4$. Red boxes indicate magnified areas.



(**a**) Hazy image (**b**) Guided image filter (**c**) DMF-GF

(**d**) Transition map of (b) (**e**) Transition map of (c) (**f**) (**g**) (**h**)

Figure 11. Haze removing. (**f**), (**g**), and (**h**) are the details of (**a**), (**b**), and (**c**) with red boxes, respectively. Our result has been computed by MRGF in Figure 4. The parameters are $r = 60$, $\epsilon = 10^{-6}$, $\sigma_r = 60$ and $T^{3rd} = 16$.

9. Conclusions

This paper presents an efficient algorithm of edge-preserving filtering with multiple guidance images for sensor fusion signals. Our algorithm, named decomposed multilateral filtering (DMF), can accelerate general multilateral filtering with the Gauss transform and extend various edge-preserving filtering methods to exploit multiple guidance images. In addition, we introduced a method to apply multilateral filtering for the output of multilateral filtering, such as rolling guidance filters [52], named multilateral rolling guidance filtering (MRGF). The experimental results showed that our algorithm has high accuracy and high efficiency. Furthermore, the proposed method is verified by various applications: flash/no-flash denoising, depth map refining, feathering, and Haze removal.

The limitations of our algorithm are that the computational time depends on the image dimensionality and the number of guidance images. However, this problem can be solved by clustering [37,91]. In addition, automatic adjustment of the downsampling amount is also an issue. These issues can be resolved by extending Gaussian KD-trees [47] and permutohedral lattice [48].

Author Contributions: Conceptualization, N.F.; methodology, N.F.; software, H.N.; validation, Y.N. and Y.K.; formal analysis, H.N.; investigation, H.N.; resources, N.F.; data curation, H.N., Y.N. and Y.K.; writing—original draft preparation, H.N.; writing—review and editing, Y.M. and N.F.; visualization, Y.M. and N.F.; supervision, N.F.; project administration, Y.M. and N.F.; funding acquisition, Y.M. and N.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by JSPS KAKENHI (21H03465, 21K17768) and the Environment Research and Technology Development Fund (JPMEERF20222M01) of the Environmental Restoration and Conservation Agency of Japan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our code is available at https://fukushimalab.github.io/dmf/ (accessed on 16 January 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CFJIR	cross-field joint image restoration
CT	computed tomography
DCT	discrete cosine transform
DMF	decomposed multilateral filtering
DTF	domain transform filtering
FIR	finite impulse response
GIF	guided image filtering
HDF	high-dimensional filtering
HDKF	high-dimensional kernel filtering
HSV	hue, saturation and value
JPEG	joint photographic experts group
LTI	linear-time invariant
LTV	linear-time variant
MF	multilateral filtering
MPEG	moving picture experts group
MRI	magnetic resonance imaging
MRGF	multilateral rolling guidance filtering
PBFICs	principle bilateral filtered image components
PSNR	peak signal-to-noise ratio
RGB	Red, green and blue
RGB-D	red, green, blue and depth
RTBF	real-time bilateral filtering
SIMD	single instruction, multiple data

References

1. Jia, W.; Song, Z.; Li, Z. Multi-scale Fusion of Stretched Infrared and Visible Images. *Sensors* **2022**, *22*, 6660. [CrossRef]
2. Li, H.; Xiao, Y.; Cheng, C.; Song, X. SFPFusion: An Improved Vision Transformer Combining Super Feature Attention and Wavelet-Guided Pooling for Infrared and Visible Images Fusion. *Sensors* **2023**, *23*, 7870. [CrossRef]
3. Chen, H.; Deng, L.; Zhu, L.; Dong, M. ECFuse: Edge-Consistent and Correlation-Driven Fusion Framework for Infrared and Visible Image Fusion. *Sensors* **2023**, *23*, 8071. [CrossRef] [PubMed]
4. Monno, Y.; Kiku, D.; Tanaka, M.; Okutomi, M. Adaptive Residual Interpolation for Color and Multispectral Image Demosaicking. *Sensors* **2017**, *17*, 2787. [CrossRef] [PubMed]
5. Morillas, S.; Gregori, V.; Sapena, A. Adaptive Marginal Median Filter for Colour Images. *Sensors* **2011**, *11*, 3205–3213. [CrossRef]
6. Morillas, S.; Gregori, V. Robustifying Vector Median Filter. *Sensors* **2011**, *11*, 8115–8126. [CrossRef]
7. Le, A.V.; Jung, S.W.; Won, C.S. Directional Joint Bilateral Filter for Depth Images. *Sensors* **2014**, *14*, 11362–11378. [CrossRef] [PubMed]
8. Chen, L.; Li, Q. An Adaptive Fusion Algorithm for Depth Completion. *Sensors* **2022**, *22*, 4603. [CrossRef]
9. Takeda, J.; Fukushima, N. Poisson disk sampling with randomized satellite points for projected texture stereo. *Opt. Contin.* **2022**, *1*, 974–988. [CrossRef]
10. Cheong, H.; Chae, E.; Lee, E.; Jo, G.; Paik, J. Fast Image Restoration for Spatially Varying Defocus Blur of Imaging Sensor. *Sensors* **2015**, *15*, 880–898. [CrossRef]
11. Yang, Y.; Tong, S.; Huang, S.; Lin, P. Dual-Tree Complex Wavelet Transform and Image Block Residual-Based Multi-Focus Image Fusion in Visual Sensor Networks. *Sensors* **2014**, *14*, 22408–22430. [CrossRef]
12. Li, Q.; Yang, X.; Wu, W.; Liu, K.; Jeon, G. Multi-Focus Image Fusion Method for Vision Sensor Systems via Dictionary Learning with Guided Filter. *Sensors* **2018**, *18*, 2143. [CrossRef]
13. Oishi, S.; Fukushima, N. Retinex-Based Relighting for Night Photography. *Appl. Sci.* **2023**, *13*, 1719. [CrossRef]
14. Huang, D.; Tang, Y.; Wang, Q. An Image Fusion Method of SAR and Multispectral Images Based on Non-Subsampled Shearlet Transform and Activity Measure. *Sensors* **2022**, *22*, 7055. [CrossRef]
15. Xiao, Y.; Guo, Z.; Veelaert, P.; Philips, W. General Image Fusion for an Arbitrary Number of Inputs Using Convolutional Neural Networks. *Sensors* **2022**, *22*, 2457. [CrossRef] [PubMed]
16. Eisemann, E.; Durand, F. Flash Photography Enhancement via Intrinsic Relighting. *ACM Trans. Graph.* **2004**, *23*, 673–678. [CrossRef]
17. Petschnigg, G.; Agrawala, M.; Hoppe, H.; Szeliski, R.; Cohen, M.; Toyama, K. Digital Photography with Flash and No-flash Image Pairs. *ACM Trans. Graph.* **2004**, *23*, 664–672. [CrossRef]
18. Kopf, J.; Cohen, M.; Lischinski, D.; Uyttendaele, M. Joint Bilateral Upsampling. *ACM Trans. Graph.* **2007**, *26*, 6497. [CrossRef]
19. Wada, N.; Kazui, M.; Haseyama, M. Extended Joint Bilateral Filter for the Reduction of Color Bleeding in Compressed Image and Video. *ITE Trans. Media Technol. Appl.* **2015**, *3*, 95–106. [CrossRef]
20. He, K.; Shun, J.; Tang, X. Guided Image Filtering. In Proceedings of the European Conference on Computer Vision (ECCV), Crete, Greece, 5–11 September 2010; pp. 1–14. [CrossRef]
21. He, K.; Sun, J.; Tang, X. Single Image Haze Removal using Dark Channel Prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 2341–2353.
22. Shi, Z.; Li, Y.; Zhang, C.; Zhao, M.; Feng, Y.; Jiang, B. Weighted median guided filtering method for single image rain removal. *EURASIP J. Image Video Process.* **2018**, *2018*, 35. [CrossRef]
23. Eichhardt, I.; Chetverikov, D.; Janko, Z. Image-guided ToF depth upsampling: A survey. *Mach. Vis. Appl.* **2017**, *28*, 267–282. [CrossRef]
24. Matsuo, T.; Fukushima, N.; Ishibashi, Y. Weighted Joint Bilateral Filter with Slope Depth Compensation Filter for Depth Map Refinement. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Barcelona, Spain, 21–24 February 2013; pp. 300–309.
25. Scharstein, D.; Szeliski, R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [CrossRef]
26. Hosni, A.; Rhemann, C.; Bleyer, M.; Rother, C.; Gelautz, M. Fast Cost-Volume Filtering for Visual Vorrespondence and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 504–511. [CrossRef] [PubMed]
27. Baker, S.; Scharstein, D.; Lewis, J.P.; Roth, S.; Black, M.J.; Szeliski, R. A Database and Evaluation Methodology for Optical Flow. *Int. J. Comput. Vis.* **2011**, *92*, 1–31. [CrossRef]
28. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Bombay, India, 4–7 January 1998; pp. 839–846. [CrossRef]
29. Lai, P.; Tian, D.; Lopez, P. Depth Map Processing with Iterative Joint Multilateral Filtering. In Proceedings of the Picture Coding Symposium (PCS), Nagoya, Japan, 8–10 December 2010; pp. 9–12. [CrossRef]
30. Gastal, E.S.L.; Oliveira, M.M. Adaptive Manifolds for Real-Time High-Dimensional Filtering. *ACM Trans. Graph.* **2012**, *31*, 2185529. [CrossRef]
31. Butt, I.T.; Rajpoot, N.M. Multilateral Filtering: A Novel Framework for Generic Similarity-Based Image Denoising. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 2981–2984. [CrossRef]

32. Choudhury, P.; Tumblin, J. The Trilateral Filter for High Contrast Images and Meshes. In Proceedings of the Eurographics Workshop on Rendering, Leuven, Belgium, 25–27 June 2003; pp. 186–196.
33. Gastal, E.S.L.; Oliveira, M.M. Domain Transform for Edge-Aware Image and Video Processing. *ACM Trans. Graph.* **2011**, *30*, 1–12. [CrossRef]
34. Sugimoto, K.; Fukushima, N.; Kamata, S. Fast bilateral filter for multichannel images via soft-assignment coding. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, Republic of Korea, 13–15 December 2016; pp. 1–4. [CrossRef]
35. Nair, P.; Chaudhury, K.N. Fast High-Dimensional Kernel Filtering. *IEEE Signal Process. Lett.* **2019**, *26*, 377–381. [CrossRef]
36. Miyamura, T.; Fukushima, N.; Waqas, M.; Sugimoto, K.; Kamata, S. Image Tiling for Clustering to Improve Stability of Constant-time Color Bilateral Filtering. In Proceedings of the International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 1038–1042. [CrossRef]
37. Oishi, S.; Fukushima, N. Tiling and PCA strategy for Clustering-based High-Dimensional Gaussian Filtering. *SN Comput. Sci.* **2023**, *5*, 40. [CrossRef]
38. Lin, G.S.; Chen, C.Y.; Kuo, C.T.; Lie, W.N. A Computing Framework of Adaptive Support-Window Multi-Lateral Filter for Image and Depth Processing. *IEEE Trans. Broadcast.* **2014**, *60*, 452–463. [CrossRef]
39. Yang, Y.; Liu, Q.; He, X.; Liu, Z. Cross-View Multi-Lateral Filter for Compressed Multi-View Depth Video. *IEEE Trans. Image Process.* **2019**, *28*, 302–315. [CrossRef]
40. Durand, F.; Dorsey, J. Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. *ACM Trans. Graph.* **2002**, *21*, 257–266. [CrossRef]
41. Paris, S.; Durand, F. A Fast Approximation of the Bilateral Filter using A Signal Processing Approach. *Int. J. Comput. Vis.* **2009**, *81*, 24–52. [CrossRef]
42. Yang, Q.; Ahuja, N.; Tan, K.H. Constant Time Median and Bilateral Filtering. *Int. J. Comput. Vis.* **2014**, *112*, 307–318. [CrossRef]
43. Yang, Q.; Tan, K.H.; Ahuja, N. Real-Time O(1) Bilateral Filtering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 557–564. [CrossRef]
44. Ghosh, S.; Chaudhury, K.N. On Fast Bilateral Filtering Using Fourier Kernels. *IEEE Signal Process. Lett.* **2016**, *23*, 570–573. [CrossRef]
45. Sugimoto, K.; Fukushima, N.; Kamata, S. 200 FPS Constant-time Bilateral Filter Using SVD and Tiling Strategy. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 190–194. [CrossRef]
46. Sumiya, Y.; Fukushima, N.; Sugimoto, K.; Kamata, S. Extending Compressive Bilateral Filtering for Arbitrary Range Kernel. In Proceedings of the International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 1018–1022. [CrossRef]
47. Adams, A.; Gelfand, N.; Dolson, J.; Levoy, M. Gaussian KD-Trees for Fast High-Dimensional Filtering. *ACM Trans. Graph.* **2009**, *28*, 1531327. [CrossRef]
48. Adams, A.; Baek, J.; Davis, M.A. Fast High-Dimensional Filtering Using the Permutohedral Lattice. *Comput. Graph. Forum* **2010**, *29*, 753–762. [CrossRef]
49. Maeda, Y.; Fukushima, N.; Matsuo, H. Taxonomy of Vectorization Patterns of Programming for FIR Image Filters Using Kernel Subsampling and New One. *Appl. Sci.* **2018**, *8*, 1985. [CrossRef]
50. Maeda, Y.; Fukushima, N.; Matsuo, H. Effective Implementation of Edge-Preserving Filtering on CPU Microarchitectures. *Appl. Sci.* **2018**, *8*, 1235. [CrossRef]
51. Naganawa, Y.; Kamei, H.; Kanetaka, Y.; Nogami, H.; Maeda, Y.; Fukushima, N. SIMD-Constrained Lookup Table for Accelerating Variable-Weighted Convolution on x86/64 CPUs. *IEEE Access* **2024**. . [CrossRef]
52. Zang, Q.; Shen, X.; Xu, L.; Jia, J. Rolling Guidance Filter. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014 [CrossRef]
53. Pajares, G.; De La Cruz, J.M. A wavelet-based image fusion tutorial. *Pattern Recognit.* **2004**, *37*, 1855–1872. [CrossRef]
54. Wang, Z.; Ziou, D.; Armenakis, C.; Li, D.; Li, Q. A comparative analysis of image fusion methods. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1391–1402. [CrossRef]
55. James, A.P.; Dasarathy, B.V. Medical image fusion: A survey of the state of the art. *Inf. Fusion* **2014**, *19*, 4–19. [CrossRef]
56. Li, S.; Kang, X.; Fang, L.; Hu, J.; Yin, H. Pixel-level image fusion: A survey of the state of the art. *Inf. Fusion* **2017**, *33*, 100–112. [CrossRef]
57. Zhang, H.; Xu, H.; Tian, X.; Jiang, J.; Ma, J. Image fusion meets deep learning: A survey and perspective. *Inf. Fusion* **2021**, *76*, 323–336. [CrossRef]
58. Kaur, H.; Koundal, D.; Kadyan, V. Image fusion techniques: A survey. *Arch. Comput. Methods Eng.* **2021**, *28*, 4425–4447. [CrossRef] [PubMed]
59. Hermessi, H.; Mourali, O.; Zagrouba, E. Multimodal medical image fusion review: Theoretical background and recent advances. *Signal Process.* **2021**, *183*, 108036. [CrossRef]
60. Azam, M.A.; Khan, K.B.; Salahuddin, S.; Rehman, E.; Khan, S.A.; Khan, M.A.; Kadry, S.; Gandomi, A.H. A review on multimodal medical image fusion: Compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics. *Comput. Biol. Med.* **2022**, *144*, 105253. [CrossRef]
61. Kalamkar, S. Multimodal image fusion: A systematic review. *Decis. Anal. J.* **2023**, *9*, 100327. [CrossRef]

62. Singh, S.; Singh, H.; Bueno, G.; Deniz, O.; Singh, S.; Monga, H.; Hrisheekesha, P.; Pedraza, A. A review of image fusion: Methods, applications and performance metrics. *Digit. Signal Process.* **2023**, *137*, 104020. [CrossRef]
63. Crow, F.C. Summed-Area Tables for Texture Mapping. In Proceedings of the ACM SIGGRAPH, Minneapolis, MN, USA, 23–27 July 1984; pp. 207–212. [CrossRef]
64. Fukushima, N.; Maeda, Y.; Kawasaki, Y.; Nakamura, M.; Tsumura, T.; Sugimoto, K.; Kamata, S. Efficient Computational Scheduling of Box and Gaussian FIR Filtering for CPU Microarchitecture. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018.
65. Deriche, R. Recursively Implementating the Gaussian and its Derivatives. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Singapore, 7–11 September 1992; pp. 263–267.
66. Sugimoto, K.; Kamata, S. Fast Gaussian Filter with Second-Order Shift Property of DCT-5. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Melbourne, Australia, 15–18 September 2013; pp. 514–518. [CrossRef]
67. Takagi, H.; Fukushima, N. An Efficient Description with Halide for IIR Gaussian Filter. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Auckland, New Zealand, 7–10 December 2020.
68. Otsuka, T.; Fukushima, N.; Maeda, Y.; Sugimoto, K.; Kamata, S. Optimization of Sliding-DCT based Gaussian Filtering for Hardware Accelerator. In Proceedings of the International Conference on Visual Communications and Image Processing (VCIP), Macau, China, 1–4 December 2020; pp. 423–426. [CrossRef]
69. Sayed, A.H. *Fundamentals of Adaptive Filtering*; John Wiley & Sons: Hoboken, NJ, USA, 2003.
70. Yoshizawa, S.; Belyaev, A.; Yokota, H. Fast Gauss Bilateral Filtering. *Comput. Graph. Forum* **2010**, *29*, 60–74. [CrossRef]
71. Fujita, S.; Fukushima, N. Extending Guided Image Filtering for High-Dimensional Signals. In *Communications in Computer and Information Science Book Series, Revised Selected Papers in 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016), Rome, Italy, 27–29 February 2016*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; Volume 693, pp. 439–453. [CrossRef]
72. Fattal, R.; Carroll, R.; Agrawala, M. Edge-Based Image Coarsening. *ACM Trans. Graph.* **2009**, *29*, 1640449. [CrossRef]
73. Fattal, R. Edge-Avoiding Wavelets and Their Applications. *ACM Trans. Graph.* **2009**, *28*, 1531328. [CrossRef]
74. Fukushima, N.; Kawasaki, Y.; Maeda, Y. Accelerating Redundant DCT Filtering for Deblurring and Denoising. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 4175–4179. [CrossRef]
75. Gavaskar, R.G.; Chaudhury, K.N. Fast Adaptive Bilateral Filtering. *IEEE Trans. Image Process.* **2019**, *28*, 779–790. [CrossRef] [PubMed]
76. Aubry, M.; Paris, S.; Hasinoff, S.W.; Kautz, J.; Durand, F. Fast Local Laplacian Filters: Theory and Applications. *ACM Trans. Graph.* **2014**, *33*, 2629645. [CrossRef]
77. Sumiya, Y.; Otsuka, T.; Maeda, Y.; Fukushima, N. Gaussian Fourier Pyramid for Local Laplacian Filter. *IEEE Signal Process. Lett.* **2022**, *29*, 11–15. [CrossRef]
78. Hayashi, K.; Maeda, Y.; Fukushima, N. Local Contrast Enhancement with Multiscale Filtering. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Taipei, Taiwan, 31 October–3 November 2023.
79. Mishiba, K. Fast Guided Median Filter. *IEEE Trans. Image Process.* **2023**, *32*, 737–749. [CrossRef] [PubMed]
80. Tsubokawa, T.; Tajima, H.; Maeda, Y.; Fukushima, N. Local look-up table upsampling for accelerating image processing. *Multimed. Tools Appl.* **2023**. [CrossRef]
81. Farbman, Z.; Fattal, R.; Lischinski, D.; Szeliski, R. Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation. *ACM Trans. Graph.* **2008**, *27*, 1360666. [CrossRef]
82. Min, D.; Choi, S.; Lu, J.; Ham, B.; Sohn, K.; Do, M. Fast Global Image Smoothing Based on Weighted Least Squares. *IEEE Trans. Image Process.* **2014**, *23*, 5638–5653. [CrossRef]
83. Xu, L.; Lu, C.; Xu, Y.; Jia, J. Image Smoothing via L0 Gradient Minimization. *ACM Trans. Graph.* **2011**, *30*, 2024208. [CrossRef]
84. Kanetaka, Y.; Takagi, H.; Maeda, Y.; Fukushima, N. SlidingConv: Domain-Specific Description of Sliding Discrete Cosine Transform Convolution for Halide. *IEEE Access* **2024**, *12*, 7563–7583 [CrossRef]
85. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801. [CrossRef]
86. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369. [CrossRef]
87. Honda, S.; Maeda, Y.; Fukushima, N. Dataset of Subjective Assessment for Visually Near-Lossless Image Coding based on Just Noticeable Difference. In Proceedings of the International Conference on Quality of Multimedia Experience (QoMEX), Ghent, Belgium, 20–22 June 2023; pp. 236–239. [CrossRef]
88. Yan, Q.; Shen, X.; Xu, L.; Zhuo, S.; Zhang, X.; Shen, L.; Jia, J. Cross-Field Joint Image Restoration via Scale Map. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013. [CrossRef]
89. Ishikawa, K.; Oishi, S.; Fukushima, N. Principal Component Analysis for Accelerating Color Bilateral Filtering. In Proceedings of the International Workshop on Advanced Imaging Technology (IWAIT), Jeju, Republic of Korea, 9–11 January 2023; Volume 12592, p. 125921F. [CrossRef]

90. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
91. Otsuka, T.; Fukushima, N. Vectorized Implementation of K-means. In Proceedings of the International Workshop on Advanced Image Technology (IWAIT), Kagoshima, Japan, 5–6 January 2021. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.