Mutually Guided Image Filtering

Xiaojie Guo, Yu Li, Jiayi Ma, and Haibin Ling

Abstract—Filtering images is required by numerous multimedia, computer vision and graphics tasks. Despite diverse goals of different tasks, making effective rules is key to the filtering performance. Linear translation-invariant filters with manually designed kernels have been widely used. However, their performance suffers from content-blindness. To mitigate the content-blindness, a family of filters, called joint/guided filters, have attracted a great amount of attention from the community. The main drawback of most joint/guided filters comes from the ignorance of structural inconsistency between the reference and target signals like color, infrared and depth images captured under different conditions. Simply adopting such guidelines very likely leads to unsatisfactory results. To address the above issues, this paper designs a simple yet effective filter, named mutually guided image filter (muGIF), which jointly preserves mutual structures, avoids misleading from inconsistent structures and smooths flat regions. The proposed muGIF is very flexible, which can work in various modes including dynamic only (self-guided), static/dynamic (reference-guided) and dynamic/dynamic (mutually guided) modes. Although the objective of muGIF is in nature non-convex, by subtly decomposing the objective, we can solve it effectively and efficiently. The advantages of muGIF in effectiveness and flexibility are demonstrated over other state-of-the-art alternatives on a variety of applications. Our code is publicly available at https://sites.google.com/view/xjguo/mugif.

Index Terms—Image filtering, joint image filtering, guided image filtering, mutually guided image filtering

1 Introduction

IMAGE filtering is a technique for modifying or enhancing images according to certain rules, which can be directly or indirectly written in the following general shape:

$$\min_{\mathbf{T}} \Psi(\mathbf{T}, \mathbf{T}_0) + \alpha\Phi(\mathbf{T}), \tag{1}$$

where $\mathbf{T}_0$ and $\mathbf{T}$ are the input and output signals respectively, $\Psi(\mathbf{T}, \mathbf{T}_0)$ denotes the fidelity term, $\Phi(\mathbf{T})$ stands for the regularizer on the expected output, and α is a non-negative coefficient balancing the two involved terms. Various multimedia, computer vision and graphics applications, such as image restoration [1], [2], image stylization [3], [4], stereo matching [5], [6], [7], optical flow [8], [9] and semantic flow [10], [11], require image filtering to help simultaneously suppress/eliminate unwanted information and preserve the desired one. For instance, texture removal is to extract structures under the complication of regular or irregular texture patterns (Fig. 1 (a)-(c)), while boundary detection seeks clear object boundaries from clutters. Ideally, if the indication (in this paper, the weight acts as the indication), *i.e.* which to discard and which to maintain, is set wisely, the filtering would naturally become much easier. However, it is difficult to construct the precise indication without the ground truth. Hence, in spite of different goals, *how to make effective rules of indication/weight construction from inputs is a core question regarding filtering performance*.

Manuscript received Aug. 30, 2017; revised Jun. 15, 2018 and Nov.20, 2018; X. Guo was supported by NSFC (grant no. 61772512) and CCF-Tencent Open Research Fund. J. Ma was supported by NSFC (grand no. 61773295). H. Ling was supported in part by US NSF (grants 1618398 and 1350521).

• *X. Guo (xj.max.guo@gmail.com) is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China.*
• *Y. Li (liyu@adsc.sg) is with the Advanced Digital Sciences Center, Singapore 138632, Singapore.*
• *J. Ma (jyma2010@gmail.com) is with Electronic Information School, Wuhan University, Wuhan 430072, China.*
• *H. Ling (hbling@temple.edu) is with Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA.*

In literature, *linear translation-invariant* (LTI) filters equipped with explicitly designed rules (also known as kernels, such as *mean*, *Gaussian* and *Laplacian* kernels [12]) are arguably the simplest ones. The spatial invariance, despite its simplicity, very often hurts the effectiveness of filtering in practical scenarios, as it treats noise, textures and structures identically. That is to say, LTIs are content blind. Different from LTIs, the mode and median filters [13], [14], [15], [16], [17] compute mode or median rather than average in local patches, which results in heavy computational loads. They are able to remove salt&pepper noise effectively, but frequently produce unsatisfactory results when facing oscillating signals (see Fig. 2 (c) for a 1D example) that contain frequent changes in certain dimensions. The oscillations with a period larger than the (pre-defined) window width persist, also commonly known as the oscillating effect. It is worth to note that, these methods degenerate the model (1) by disabling the regularizer and adopting $\|\mathbf{T} - f(\mathbf{T}_0)\|$ as the fidelity with $f(\cdot)$ and $\|\cdot\|$ representing a specific operator (*e.g.* Gaussian convolution and local median operators) and a certain norm (*e.g.* ℓ_1 and ℓ_2 norms), respectively.

To overcome the issue of content-blindness, it is natural to ask for some guidance information. As a consequence, a series of *guided filters* (GF) have been proposed. We call a GF relying on the input itself a self-guided filter. The *bilateral filter* (BF) [18], [19], as a classic GF, processes a pixel via averaging its neighbors, weighted by the Gaussian of both spatial and intensity/color distances. Though BF is successful in removing small textures while preserving edges, it may suffer from unexpected gradient reversal artifacts [20], [21]. Another self-guided filter, named *rolling guidance filter* (RGF) [22], builds upon the scale space theory, which shows that small structures can be completely removed by a properly scaled Gaussian filter while large-scale ones survive (though blurred). Different from BF, RGF iteratively recalls strong edges/structures, and employs the intensity information of the result obtained from the previous iteration as

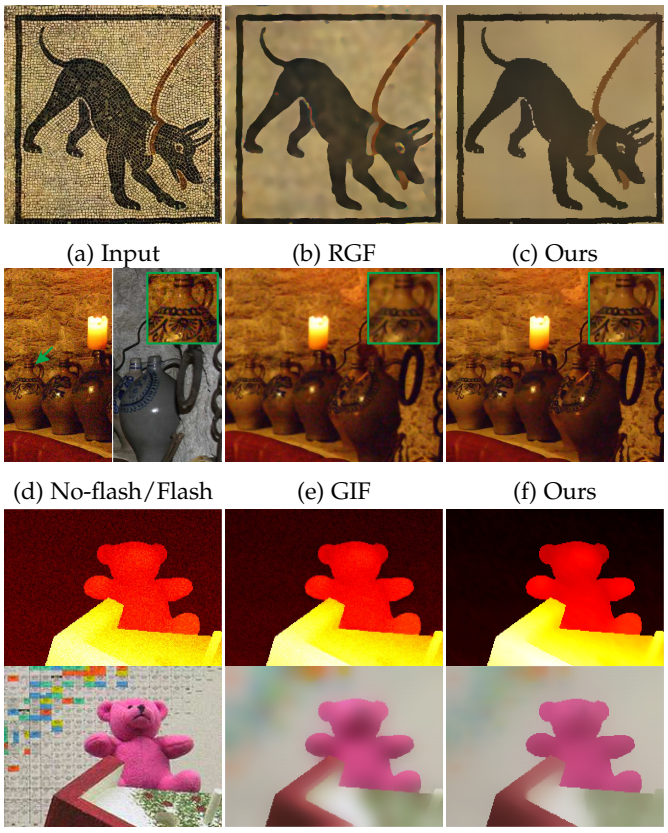


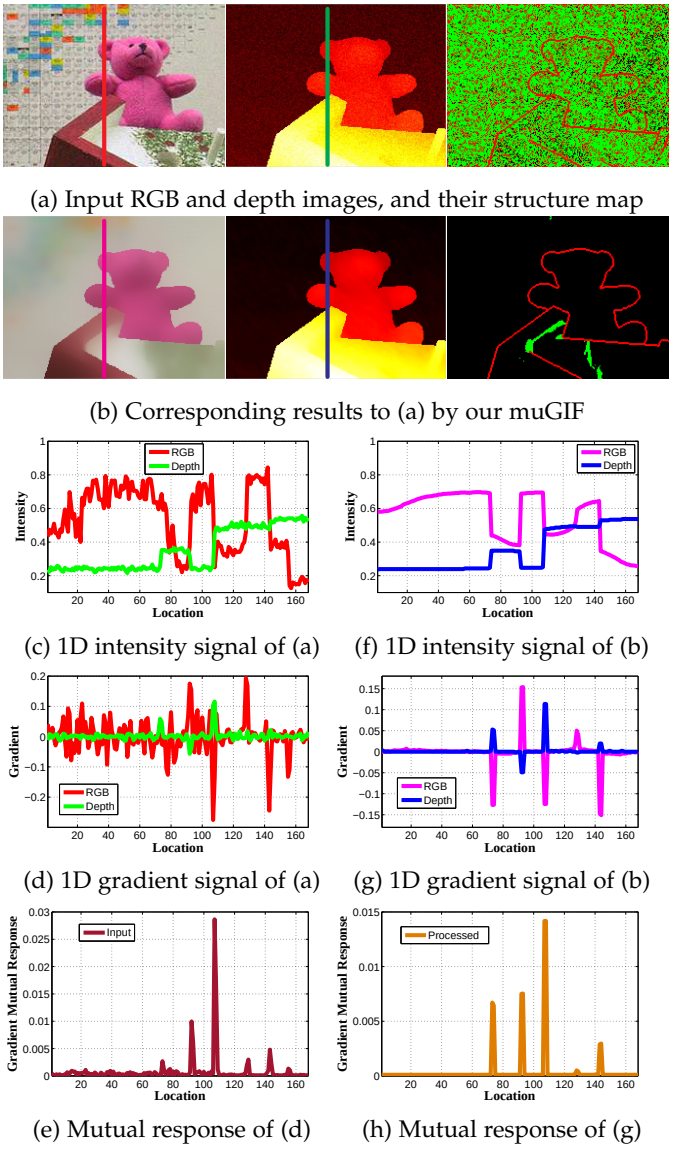
Fig. 1: *Texture removal*: RGF [22] (b) and our muGIF (c) remove rich textures from (a). *No-flash/flash image restoration*: Guided by the flash image, the restored no-flash results of GIF [27] and muGIF are in (e) and (f), respectively. *Mutual structure extraction*: From the noisy depth/RGB (g), the mutual structures extracted by JFMS [28] and muGIF are given in (h) and (i), respectively.

guidance, instead of the input itself. RGF is limited to scale-space filtering and may cause inaccurate edge localization. Although BF and RGF improve the Gaussian filter in terms of kernel construction according to local image content, they essentially follow the degenerated model of Eq. (1), *i.e.*, $\|\mathbf{T}-f(\mathbf{T}_0)\|$. Furthermore, the aforementioned methods are in nature local operators, thus still suffering from the oscillating effect. Other pioneering attempts in global manners, with *anisotropic diffusion* (AD) [23] and *weighted least squares* (WLS) [21] as representatives, utilize the gradients of the filtering image for the sake of structure-texture separation. The dominant assumption of these works is that gradients with large magnitudes should be preserved as they are of high probabilities to be on edges/boundaries and vice versa. A method adopting the ℓ_0 regularizer to constrain "intrinsic" boundaries [24], *a.k.a.* ℓ_0 *gradient minimization* (L0GM), has been developed, aiming to leverage the scale issue of [25] that makes use of the ℓ_1 regularizer. Xu *et al.* proposed a strategy based on *relative total variation* (RTV) [26], which enforces desired statistic properties to distinguish structure from rich textures.

Besides the filtering input, one may also consider another image to act as the guidance. The principle behind is transferring the structure in the reference image to the target one, so called reference-guided filtering. The *joint bilateral filter* (JBF) [1], generalized from BF, computes the weights from the guidance rather than the filtered image, which particularly favors the cases where the guidance image can provide more reliable edge information than the filtered image. He *et al.* proposed an approach, called *guided image filter* (GIF) [27], which is a locally linear transform of the guidance image. GIF has shown its promising performance in a number of applications, such as image smoothing, image enhancement and HDR compression. The aforementioned approaches including JBF and GIF, as the principle indicates, imply that the information of the guidance image is useful, which can be frequently violated. Because they ignore the structural inconsistency between the reference and target signals captured under different conditions, like color, infrared, depth and day/night images. Moreover, in practice, guidance images might be in trouble as well. Simply adopting such guidelines is at high risk of generating undesired results. Most recently, Ham *et al.* developed a *static/dynamic* (SD) filter [29]. The static part follows previous joint filters, say modulating the input image with a weight function depending on features of the guidance image. The dynamic component takes better care of the filtered signal, *i.e.* iteratively utilizing the target image as an additional (dynamic) guidance to constrain the output so as to mitigate the effect from structural differences. However, this strategy does not fully utilize the static guidance image and cannot jointly deal with the two inputs.

For boosting the performance of joint processing in restoring shared structures, Shen *et al.* [28] explicitly defined the concept of mutual-structure. In [28], three kinds of structures are presented, including 1) *mutual structures*: simply explained as common edges existing in the corresponding two patches, which are not necessarily with the same magnitude and can be of different gradient directions; 2) *inconsistent structures*: different patterns between the two patches, *i.e.* when one edge appears in only one image but not in the other; and 3) *smooth/flat regions*: common low-variance smooth patches in both images, which often host visual artifacts. The three definitions suggest that mutual structures should be transferred to help filtering while inconsistent ones should not be transferred to avoid misleading. Based on the definitions, Shen *et al.* designed a normalized cross correlation (NCC)-based model, *joint filtering using mutual-structure* (JFMS) [28], which is in nature a locally linear transform model (local method). Compared with the SD filter, JFMS can be viewed as a *dynamic/dynamic* or mutually guided filter, having the ability of jointly processing the two inputs. However, due to the local filtering formulation of JFMS, it sometimes introduces halo artifacts to the results.

Contribution. This paper proposes a novel measure on structure similarity, and designs a general filtering model, termed as *mutually guided image filter* (muGIF). We define three kinds of structures similar to those in [28]. However, they are not identical: JFMS's definitions are on the patch level while ours are on the pixel level, which changes a local method to a global one. More concretely, the main contributions of this paper can be summarized as the following points: 1) We define a new measurement, *i.e. relative structure*, to manage the structure similarity between two



(a) Input RGB and depth images, and their structure map

(b) Corresponding results to (a) by our muGIF

(c) 1D intensity signal of (a)

(f) 1D intensity signal of (b)

(d) 1D gradient signal of (a)

(g) 1D gradient signal of (b)

(e) Mutual response of (d)

(h) Mutual response of (g)

Fig. 2: 1D and 2D illustration. Only y-direction derivatives are shown in (d-e) and (g-h). For structure maps, mutual structures are colored in red while inconsistent and smooth structures in green and black, respectively.

inputs. Based on the measurement, a global optimization objective is designed to achieve high-quality filtering; 2) The objective is in nature non-convex and hard to be directly optimized. By decomposing the objective, the target problem can be effectively and efficiently resolved; 3) Our muGIF is flexible in handling input signals captured by various sensors and under different shooting conditions. Compared with existing image filters, it can act as a dynamic/dynamic (mutually guided), a static/dynamic (reference-guided), or a dynamic only (self-guided) filter – please refer to Fig. 1 for examples; and 4) To demonstrate the efficacy of our muGIF and show its superiority over other state-of-the-art alternatives, experiments on a number of computer vision and multimedia tasks are conducted.

A preliminary version of this manuscript appeared in [30]. Compared with [30], this journal version presents the model design and the solver in more theoretical detail and gives deeper analysis on muGIF's properties and potentials. Extensive experimental comparisons are conducted to verify the advantages of our muGIF on more applications. To allow more comparisons from the community and encourage future work, we release our code at https://sites.google.com/view/xjguo/mugif.

2 Mutually Guided Image Filter

Our system takes as input signals captured from the same camera position by different sensors (*e.g.* RGB, infrared and depth), and/or under different shooting conditions. The inputs could have different number of channels. Pixel values in each channel are scaled to a fixed range*, and color channels are processed separately. Here we give the notations used in this paper. The target and reference images are denoted by $\mathbf{T}_0$ and $\mathbf{R}_0$, respectively. The filtering (intermediate) outputs are designated as $\mathbf{T}$ and $\mathbf{R}$, respectively. Further, we denote the pixel coordinates by $i = (x, y)^T$, *e.g.* $\mathbf{T}_i$ represents the pixel at $(x, y)^T$. In addition, $\mathbf{t}$, $\mathbf{t}_0$, $\mathbf{r}$ and $\mathbf{r}_0$ are vectorized versions of $\mathbf{T}$, $\mathbf{T}_0$, $\mathbf{R}$ and $\mathbf{R}_0$, respectively.

2.1 Problem Formulation and Illustration

This work aims to develop an image filter that can preserve mutual structures, prevent misleading from inconsistent structures, and smooth flat regions between two input images. Prior to detailing our objective, we first describe the definitions of mutual structure, inconsistent structure, and flat structure: 1) **mutual structure** – at a certain location, when the magnitudes of $\nabla\mathbf{R}_i$ and $\nabla\mathbf{T}_i$ are both strong enough; 2) **inconsistent structure** – when one of the magnitudes of $\nabla\mathbf{R}_i$ and $\nabla\mathbf{T}_i$ is strong (larger than a predefined threshold) and the other is weak; and 3) **smooth/flat structure** – when the magnitudes of $\nabla\mathbf{R}_i$ and $\nabla\mathbf{T}_i$ are both weak. ∇ is the first order derivative filter containing ∇_h (horizontal) and ∇_v (vertical). It is worth to clarify that, in the sense of common structure, both smooth and mutual structures are consistent. In order to explain our idea better, we also provide the definitions of mutual response and structure map. The mutual response represents the joint strength of derivatives of two signals in a certain dimension (*e.g.*, the horizontal mutual response of $\mathbf{R}$ and $\mathbf{T}$ at location i is $|\nabla_h\mathbf{R}_i \cdot \nabla_h\mathbf{T}_i|$), while the structure map indicates whether each region using the state is a mutual structure, inconsistent structure or a smooth region. Below, we introduce a key concept to our modeling, *i.e. relative structure*, as follows.

Definition 1 (Relative Structure). *Given two input images* $\mathbf{T}$ *and* $\mathbf{R}$ *with the same size, the relative structure of* $\mathbf{T}$ *with respect to* $\mathbf{R}$ *is defined as:*

$$\mathcal{R}(\mathbf{T}, \mathbf{R}) \doteq \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d\mathbf{T}_i|}{|\nabla_d\mathbf{R}_i|}, \tag{2}$$

where $|\cdot|$ *is the absolute value operator.*

The relative structure $\mathcal{R}(\mathbf{T}, \mathbf{R})$ measures structure discrepancy of $\mathbf{T}$ with respect to $\mathbf{R}$. The sign reverse between $\nabla_d\mathbf{R}_i$

*. Information from different domains may have different value ranges. To avoid the scale issue, one can normalize the information of a certain domain from its raw range to a pre-defined one, *e.g.* [0,1].

and $\nabla_d \mathbf{T}_i$, which often exists when the inputs are captured from different sensors or under varying conditions, is ignored. For a location on an edge in $\mathbf{R}$, the penalty on $|\nabla_d \mathbf{T}_i|$, say $\frac{1}{|\nabla_d \mathbf{R}_i|}$, is small; while for a location in a flat region in $\mathbf{R}$, the penalty turns to be large.

Based on the defined relative structure, we give the following formulation:

$$\begin{aligned}\arg\min_{\mathbf{T},\mathbf{R}} \ & \alpha_t \mathcal{R}(\mathbf{T},\mathbf{R}) + \beta_t \|\mathbf{T}-\mathbf{T}_0\|_2^2 + \\ & \alpha_r \mathcal{R}(\mathbf{R},\mathbf{T}) + \beta_r \|\mathbf{R}-\mathbf{R}_0\|_2^2,\end{aligned} \tag{3}$$

where α_t, α_r, β_t and β_r are non-negative constants balancing the corresponding terms, and $\|\cdot\|_2$ stands for the ℓ_2 norm. We note that the fidelity terms $\|\mathbf{T}-\mathbf{T}_0\|_2^2$ and $\|\mathbf{R}-\mathbf{R}_0\|_2^2$ are introduced to avoid the trivial solution through constraining $\mathbf{T}$ and $\mathbf{R}$ not to wildly deviate from the inputs $\mathbf{T}_0$ and $\mathbf{R}_0$, respectively. We adopt the ℓ_2 loss on intensity due to its fast computation. We emphasize that the mutuality of guidance stems from $\mathcal{R}(\mathbf{T},\mathbf{R})$ and $\mathcal{R}(\mathbf{R},\mathbf{T})$.

Illustration. From Fig. 2 (a) and (c), we can hardly recognize obvious relations between the RGB and depth images in the intensity field, as the two images reflect different properties of the target scene. Specifically, the RGB image provides the appearance details, while the depth reveals the distances between the objects to the sensor. In addition, they have different intensity ranges. In Fig. 2 (b) and (f), despite the processed results showing less textures in (b) and less ups-and-downs in (f), the mentioned issues remain. Alternatively, transforming from the intensity field to the gradient field exhibits the correlation between the RGB and depth signals as shown in Fig. 2 (d-e) and (g-h). Although the curves in Fig. 2 (d) are frequently oscillating around 0, especially for the RGB image due to the rich textures and noise, the plot in Fig. 2 (e) demonstrates that the mutual responses tend to be sparse with the mutual structures (Case 1) giving powerful pulses. Please see the last picture in Fig. 2 (a) for the entire 2D structure map. Recall that the weighting strategy is important to indicate which structures to maintain and which ones to discard. The goal of mutually guided filtering is to jointly preserve mutual structures and suppress other structures between two inputs. Hence, it is natural to consider depressing the smoothing penalties on mutual structures and elevating those on the others for achieving the goal. By iteratively enhancing the sparsity of mutual response, our proposed muGIF produces the desired results as shown in Fig. 2 (b) and (f-h), *i.e.* textures and noise removed and common structures preserved. The sparsity of the structure map in Fig. 2 (b) and Fig. 2 (h) is significantly improved over that of the input images. In other words, the dominant mutual structures (in red) survive while the noisy ones are eliminated. Most of the inconsistent structures (the green regions) in (a) become flat (the black regions), and no reversed changes happen.

2.2 Numerical Solution

The muGIF model (3) is complex, and its solution is difficult to be obtained by directly optimization. We first introduce a surrogate function for the relative structure, then decompose the objective into several quadratic and non-linear terms, and customize an effective and efficient solver to solve the problem in an alternating manner.

Surrogate Function. First, to prevent against extreme situations such as division by zero, we introduce a small positive constant ϵ_r into the denominator of Eq. (2) as:

$$\mathcal{R}(\mathbf{T},\mathbf{R},\epsilon_r) \doteq \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d \mathbf{T}_i|}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)}. \tag{4}$$

Next, the relationship below holds true:

$$\frac{|\nabla_d \mathbf{T}_i|}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)} = \frac{|\nabla_d \mathbf{T}_i| \cdot \max(|\nabla_d \mathbf{T}_i|,\epsilon_t)}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)\cdot\max(|\nabla_d \mathbf{T}_i|,\epsilon_t)}, \tag{5}$$

where the denominator can be viewed as the pixel-wise mutual response in the gradient field. The introduction of ϵ_t, same as ϵ_r, is to avoid dividing by zero. Thus, we can distinguish different cases by treating the denominator as a weight. Please see the structure maps in Fig. 2 (a) and (b) for example. Furthermore, we have

$$\begin{aligned}&\tilde{\mathcal{R}}(\mathbf{T},\mathbf{R},\epsilon_t,\epsilon_r) \doteq \\ &\sum_i \sum_{d\in\{h,v\}} \frac{(\nabla_d \mathbf{T}_i)^2}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)\cdot\max(|\nabla_d \mathbf{T}_i|,\epsilon_t)} \leq \\ &\sum_i \sum_{d\in\{h,v\}} \frac{(\nabla_d \mathbf{T}_i)^2 + |\nabla_d \mathbf{T}_i|\cdot\max(\epsilon_t - |\nabla_d \mathbf{T}_i|, 0)}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)\cdot\max(|\nabla_d \mathbf{T}_i|,\epsilon_t)} \\ &= \mathcal{R}(\mathbf{T},\mathbf{R},\epsilon_r).\end{aligned} \tag{6}$$

For a certain pixel, the equality breaks only when $|\nabla \mathbf{T}_i| < \epsilon_t$, and the gap is upper-bounded by

$$\frac{|\nabla_d \mathbf{T}_i|\cdot(\epsilon_t - |\nabla_d \mathbf{T}_i|)}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)\cdot\epsilon_t} \leq \frac{\epsilon_t}{4\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)}. \tag{7}$$

As can be seen, the biggest gap $\epsilon_t/(4\epsilon_r)$ (for all the experiments, we empirically set $\epsilon_t = \epsilon_r = 0.01$) is reached when the two corresponding regions are both flat. Even the biggest gap is trivial. An immediate consequence is the suitability of employing $\tilde{\mathcal{R}}(\mathbf{T},\mathbf{R},\epsilon_t,\epsilon_r)$ as a tight surrogate of $\mathcal{R}(\mathbf{T},\mathbf{R},\epsilon_r)$. Analogous analysis and replacement serve $\mathcal{R}(\mathbf{R},\mathbf{T},\epsilon_t)$ and $\tilde{\mathcal{R}}(\mathbf{R},\mathbf{T},\epsilon_r,\epsilon_t)$. Please see Fig. 3 for the visualized shapes of the functions. We will see the benefit of the replacement to the fast numerical solution later. The final objective to solve is:

$$\begin{aligned}\arg\min_{\mathbf{T},\mathbf{R}} \ & \alpha_t \tilde{\mathcal{R}}(\mathbf{T},\mathbf{R},\epsilon_t,\epsilon_r) + \beta_t \|\mathbf{t}-\mathbf{t}_0\|_2^2 + \\ & \alpha_r \tilde{\mathcal{R}}(\mathbf{R},\mathbf{T},\epsilon_r,\epsilon_t) + \beta_r \|\mathbf{r}-\mathbf{r}_0\|_2^2.\end{aligned} \tag{8}$$

Solver. Let $\mathbf{Q}_d$ and $\mathbf{P}_d$ ($d \in \{h,v\}$) denote the diagonal matrices with the ith diagonal entries being $\frac{1}{\max(|\nabla_d \mathbf{T}_i|,\epsilon_t)}$ and $\frac{1}{\max(|\nabla_d \mathbf{R}_i|,\epsilon_r)}$, respectively. Consequently, the objective (8) can be decomposed as follows:

$$\begin{aligned}\arg\min_{\mathbf{t},\mathbf{r}} \ & \alpha_t \mathbf{t}^T \Big(\sum_{d\in\{h,v\}} \mathbf{D}_d^T \mathbf{Q}_d \mathbf{P}_d \mathbf{D}_d \Big) \mathbf{t} + \beta_t \|\mathbf{t}-\mathbf{t}_0\|_2^2 + \\ & \alpha_r \mathbf{r}^T \Big(\sum_{d\in\{h,v\}} \mathbf{D}_d^T \mathbf{Q}_d \mathbf{P}_d \mathbf{D}_d \Big) \mathbf{r} + \beta_r \|\mathbf{r}-\mathbf{r}_0\|_2^2,\end{aligned} \tag{9}$$

where $\mathbf{D}_d$ is the Toeplitz matrix from the discrete gradient operator in the d direction with forward difference.

Thanks to the decomposition, the objective in the shape of (9) makes an *alternating least squares* (ALS) solver possible. To solve (9), we propose the following procedure:

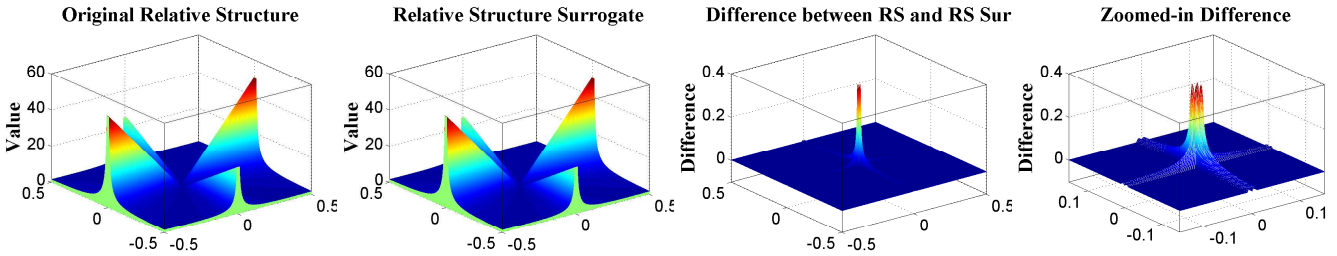


Fig. 3: Gap between relative structure and relative structure surrogate (6). The leftmost picture draws the shape of relative structure (RS): $\frac{|u|}{\max(|v|,0.01)} + \frac{1}{2}\frac{|v|}{\max(|u|,0.01)}$. The middle left shows the shape of relative structure surrogate (RS Sur.): $\frac{|u|^2}{\max(|u|,0.01)\cdot\max(|v|,0.01)} + \frac{1}{2}\frac{|v|^2}{\max(|u|,0.01)\cdot\max(|v|,0.01)}$. The rest two depict the gap and the zoomed-in gap between RS and RS Sur., respectively. Please notice the scale of z-axis, the largest value is $0.25 + 0.5 \times 0.25 = 0.375$.

Update $\mathbf{T}^{(k+1)}$: Given $\mathbf{R}^{(k)}$, $\mathbf{Q}_d^{(k)}$ and $\mathbf{P}_d^{(k)}$ estimated from the previous iteration, by dropping the terms unrelated to $\mathbf{T}$, the $\mathbf{T}$ subproblem boils down to the following:

$$\arg\min_{\mathbf{t}} \ \frac{\alpha_t}{\beta_t}\mathbf{t}^T\Big(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{Q}_d^{(k)}\mathbf{P}_d^{(k)}\mathbf{D}_d\Big)\mathbf{t} + \|\mathbf{t}-\mathbf{t}_0\|_2^2. \tag{10}$$

As can be observed, problem (10) only involves quadratic terms. Thus, its solution in closed form can be easily obtained by solving the equation system:

$$\Big(\mathbf{I} + \frac{\alpha_t}{\beta_t}\Big(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{Q}_d^{(k)}\mathbf{P}_d^{(k)}\mathbf{D}_d\Big)\Big)\mathbf{t} = \mathbf{t}_0, \tag{11}$$

where $\mathbf{I}$ is the identity matrix with proper size. Directly calculating the inverse of the target matrix $\mathbf{I} + \frac{\alpha_t}{\beta_t}\Big(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{Q}_d^{(k)}\mathbf{P}_d^{(k)}\mathbf{D}_d\Big)$ is a straightforward way to accomplish this job. However, the matrix inverse is computationally expensive, especially for large matrices like the involved one. Fortunately, since the target is a symmetric positive definite Laplacian matrix, there are many efficient techniques available for solving it, for example, [21], [31], [32], [33], [34].

Update $\mathbf{Q}_d^{(k+1)}$: Having $\mathbf{T}^{(k+1)}$ refreshed, the update of $\mathbf{Q}_d^{(k+1)}$ can be simply done by following its definition.

Update $\mathbf{R}^{(k+1)}$: With $\mathbf{T}^{(k+1)}$, $\mathbf{Q}_d^{(k+1)}$ and $\mathbf{P}_d^{(k)}$ fixed, picking out the terms relevant to $\mathbf{R}$ yields:

$$\arg\min_{\mathbf{r}} \ \frac{\alpha_r}{\beta_r}\mathbf{r}^T\Big(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{Q}_d^{(k+1)}\mathbf{P}_d^{(k)}\mathbf{D}_d\Big)\mathbf{r} + \|\mathbf{r}-\mathbf{r}_0\|_2^2. \tag{12}$$

Its solution can be obtained in a similar way with (10).

Update $\mathbf{P}_d^{(k+1)}$: It is easy to construct $\mathbf{P}_d^{(k+1)}$ based on $\mathbf{R}^{(k+1)}$ according to its definition.

Iteratively processing the above steps is found efficient to converge with promising performance. For clarity and completeness, we sketch the whole scheme of muGIF in Algorithm 1. We notice that the initialization of $\mathbf{Q}_d^{(0)}$ and $\mathbf{P}_d^{(0)}$ is finished based on $\mathbf{T}^{(0)}$ and $\mathbf{R}^{(0)}$ at the beginning of the procedure. Please refer to Algorithm 1 for details.

Algorithm 1: muGIF

Input: $\mathbf{T}_0$, $\mathbf{R}_0$, K, α_r, α_t, β_r, β_t, ϵ_r, ϵ_t, $\mathbf{T}^{(0)} \leftarrow \mathbf{T}_0$, $\mathbf{R}^{(0)} \leftarrow \mathbf{R}_0$.

Initialization: $\mathbf{Q}_d^{(0)}$ and $\mathbf{P}_d^{(0)}$ based on $\mathbf{T}_0$ and $\mathbf{R}_0$

for *k from* 0 *to* $K-1$ **do**

  Update $\mathbf{T}^{(k+1)}$ via solving Eq. (10);

  Update $\mathbf{Q}_d^{(k+1)}$ based on $\mathbf{T}^{(k+1)}$;

  Update $\mathbf{R}^{(k+1)}$ via solving Eq. (12);

  Update $\mathbf{P}_d^{(k+1)}$ based on $\mathbf{R}^{(k+1)}$;

end

Output: $(\mathbf{T}^{(K)}, \mathbf{R}^{(K)})$.

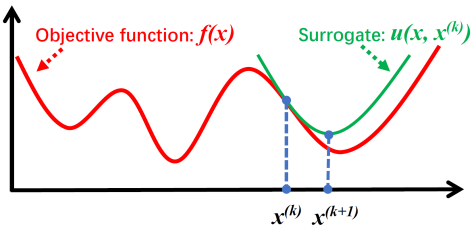


Fig. 4: Sketch of MM. At the k-th iteration, the surrogate function $u(x, x^{(k)})$ is constructed based on the current $x^{(k)}$, the curve of which is above that of the objective function $f(x)$. At the point $x^{(k)}$, $f(x) = u(x, x^{(k)})$. Through minimizing $u(x, x^{(k)})$, the next estimate $x^{(k+1)}$ is obtained. The values of objective function $f(x)$ at $\{x^{(k)}\}$ are non-increasing. The procedure is terminated until convergence.

3 Property Analysis

3.1 Optimization with Majorization-Minimization

Technically, the muGIF algorithm follows the framework of *iteratively re-weighted least squares* (IRLS) [35], which can be viewed as an advanced version of WLS [21] by dynamically adjusting weights, and as a specific case of *majorization-minimization* (MM) [36], [37].

For completeness, we briefly review the optimization with MM. The MM technique is preferred to handle an objective function $f(x)$ that is difficult to manipulate directly. The fundamental idea of MM is to successively minimize an easy-to-tackle majorizing surrogate $u(x, x^{(k)})$ associated

with the current estimate $x^{(k)}$. Besides, the constructed surrogate function should satisfy the following properties:

$$\forall x\ f(x) \leq u(x, x^{(k)}) \ \text{ and } \ x^{(k)} = \text{argmin}_x u(x, x^{(k)}) - f(x). \tag{13}$$

Very often, the second property is satisfied by choosing $u(x^{(k)}, x^{(k)}) = f(x^{(k)})$. Then, by minimizing $u(x, x^{(k)})$, the next estimate $x^{(k+1)}$ is obtained. The values of objective function $f(x)$ at $\{x^{(k)}\}$ are non-increasing. The two steps are repeated until convergence. Figure 4 pictorially shows the idea of MM. Specifically, we provide two surrogate construction rules. **Rule #1**: if $f(x) \doteq |x|$ then it has a quadratic majorizer $u(x, x^{(k)}) \doteq \frac{1}{2}\left(|x^{(k)}| + \frac{x^2}{|x^{(k)}|}\right)$, satisfying the properties in Eq. (13); and **Rule #2**: if $f(x)$ is concave and differentiable then $f(x) \leq u(x, x^{(k)}) \doteq f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)})$.

3.2 Flexibility

The model of muGIF is flexible to handle various cases, including dynamic/dynamic (mutually guided), static/dynamic (reference-guided) and dynamic only (self-guided) cases.

Dynamic/Dynamic (mutually guided) When one intends to seek the mutual structure between two inputs (see Fig. 1 (g)-(i)), *e.g.* depth/RGB and day/night, muGIF can handle these scenarios using the general model of (9), which approximately solves the problem:

$$\begin{aligned}\min_{\mathbf{T},\mathbf{R}}\ & \alpha_t \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d \mathbf{T}_i|}{\max(|\nabla_d \mathbf{R}_i|, \epsilon_r)} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2 + \\ & \alpha_r \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d \mathbf{R}_i|}{\max(|\nabla_d \mathbf{T}_i|, \epsilon_t)} + \beta_r \|\mathbf{r} - \mathbf{r}_0\|_2^2.\end{aligned} \tag{14}$$

Static/Dynamic (reference-guided) When the reference is reliable, we can fix it during the process. Subsequently, the model (9) degenerates to the following:

$$\arg\min_{\mathbf{t}}\ \alpha_t \mathbf{t}^T \left(\sum_{d\in\{h,v\}} \mathbf{D}_d^T \mathbf{Q}_d \bar{\mathbf{P}}_d \mathbf{D}_d \right) \mathbf{t} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2, \tag{15}$$

where $\bar{\mathbf{P}}_d$ is the fixed weighting matrix according to the reference $\mathbf{R}_0$. The applications such as no-flash/flash and RGB/depth image restoration (see Fig. 1 (d)-(f) for example) fit this situation. Actually, muGIF in static/dynamic mode attempts to resolve the following problem:

$$\min_{\mathbf{T}}\ 2\alpha_t \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d \mathbf{T}_i|}{\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r)} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2, \tag{16}$$

which can be viewed as a *weighted anisotropic total variation minimization* problem [38], [39], [40]. In this case, muGIF gradually decreases the energy of (16), please see Proposition 1.

Proposition 1. *Optimizing the static/dynamic image filtering problem* (15) *by muGIF is convergent, which gradually decreases the energy of the objective* (16).

Proof. The proof can be done from an MM [36] perspective. Let us consider the objective function $\mathcal{E}(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ defined as Eq. (16). We first put attention on the first term. In this case, the following relationship can be obtained by employing Rule #1, *i.e.*:

$$\begin{aligned}\frac{|\nabla_d \mathbf{T}_i|}{\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r)} \leq \frac{|\nabla_d \mathbf{T}_i^{(k)}|}{2\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r)} + \\ \frac{(\nabla_d \mathbf{T}_i)^2}{2\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r) \cdot \max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+)},\end{aligned} \tag{17}$$

in which, the equality occurs at $|\nabla_d \mathbf{T}_i| = |\nabla_d \mathbf{T}_i^{(k)}|$. In addition, the notation $\epsilon_t \to 0^+$ means that ϵ_t is a positive constant and sufficiently close to 0.

With the above (17), it is immediate to give a surrogate function of (16) as $\mathcal{Q}^k(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0) \doteq$

$$\begin{aligned}&\alpha_t \sum_i \sum_{d\in\{h,v\}} \frac{(\nabla_d \mathbf{T}_i)^2}{\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r) \cdot \max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+)} \\ &+ \alpha_t \sum_i \sum_{d\in\{h,v\}} \frac{|\nabla_d \mathbf{T}_i^{(k)}|}{\max(|\nabla_d \mathbf{R}_{0i}|, \epsilon_r)} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2.\end{aligned} \tag{18}$$

The function $\mathcal{Q}^k(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ majorizes $\mathcal{E}(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ at the point $\mathbf{T}^{(k)}$. Then, minimizing $\mathcal{Q}^k(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ equals to minimizing (15) by discarding the constant terms. Optimizing (16) using MM ensures the non-incremental property of the objective function. In addition, based on the nature of the (16), the energy has a lower-bounded value. This is to say, the muGIF in static/dynamic is convergent. □

Dynamic Only (self-guided) When no other references are available, the target itself is employed as the guidance. The tasks like texture removal and scale-space filtering (see Fig. 1 (a)-(c)) belong to the dynamic only category. In this situation, the muGIF turns out to be:

$$\arg\min_{\mathbf{t}}\ \alpha_t \mathbf{t}^T \left(\sum_{d\in\{h,v\}} \mathbf{D}_d^T \mathbf{Q}_d \mathbf{Q}_d \mathbf{D}_d \right) \mathbf{t} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2. \tag{19}$$

Optimizing (19) via Alg. 1 implicitly minimizes the following non-convex problem:

$$\min_{\mathbf{T}}\ 2\alpha_t \sum_i \sum_{d\in\{h,v\}} \log(|\nabla_d \mathbf{T}_i|) + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2. \tag{20}$$

Please see Proposition 2 for explanation. The log term $\log(|\nabla_d \mathbf{T}_i|)$ can better approximate the sparsity (ℓ_0) [41], [42] than $|\nabla_d \mathbf{T}_i|$ (ℓ_1, *total variation regularized minimization*) [25], [43].

Proposition 2. *Optimizing the dynamic only image filtering problem* (19) *by muGIF is convergent, which gradually decreases the energy of the objective* (20).

Proof. In this case, the objective function $\mathcal{E}(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ is defined as Eq. (20). The log term satisfies the following:

$$\begin{aligned}\log(|\nabla_d \mathbf{T}_i|) &\leq \log(|\nabla_d \mathbf{T}_i^{(k)}|) + \frac{|\nabla_d \mathbf{T}_i| - |\nabla_d \mathbf{T}_i^{(k)}|}{\max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+)} \\ &\leq \log(|\nabla_d \mathbf{T}_i^{(k)}|) + \frac{(\nabla_d \mathbf{T}_i)^2}{2(\max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+))^2} \\ &\quad - \frac{|\nabla_d \mathbf{T}_i^{(k)}|}{2\max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+)}.\end{aligned} \tag{21}$$

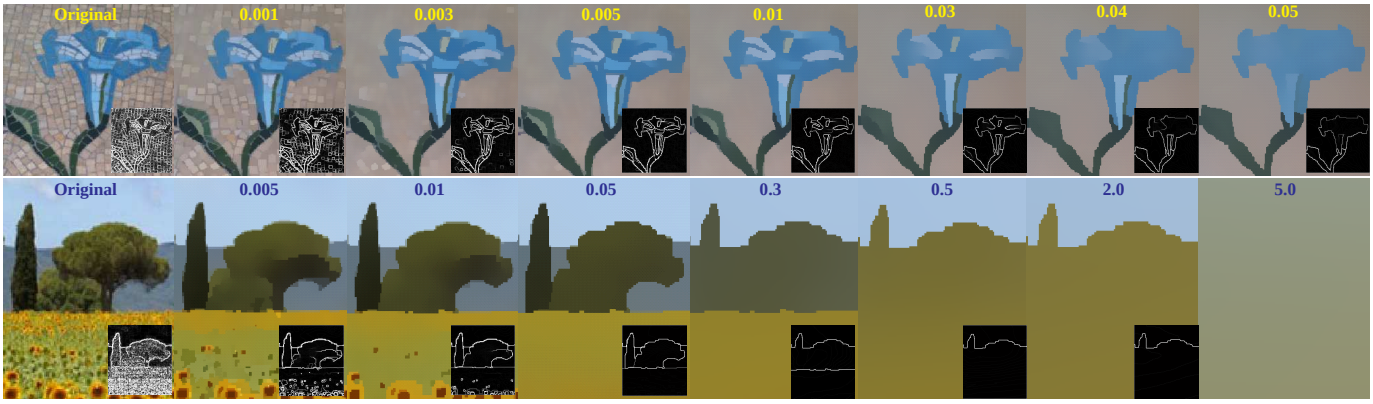


Fig. 5: Effect of α_t on dynamic only cases. The 1st case corresponds to texture removal while the 2nd scale-space filtering.

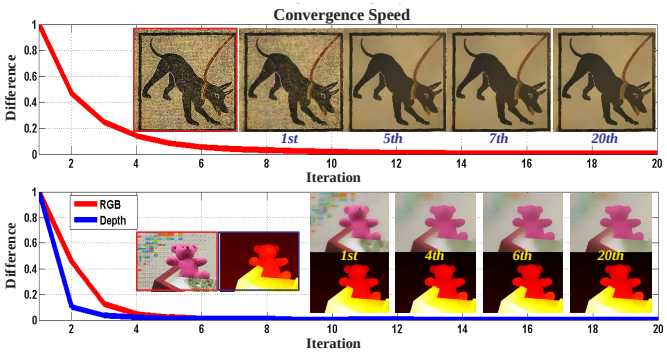


Fig. 6: Convergence behavior. *Top row*: A dynamic only example (texture removal, $\alpha_t = 0.05$). *Bottom row*: A dynamic/dynamic example (mutual structure extraction, $\alpha_t = 0.02$ for RGB and $\alpha_r = 0.01$ for depth).

The relation in the first row is satisfied due to Rule #2, while the second relation is satisfied by Rule #1. Both the equalities hold when $|\nabla_d \mathbf{T}_i| = |\nabla_d \mathbf{T}_i^{(k)}|$. Similarly, the majorizer of $\mathcal{E}(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0)$ at the point $\mathbf{T}^{(k)}$ is given by $\mathcal{Q}^k(\mathbf{T}|\mathbf{T}_0, \mathbf{R}_0) \doteq$

$$\sum_i \sum_{d\in\{h,v\}} \left(2\log\left(|\nabla_d \mathbf{T}_i^{(k)}|\right) - \frac{|\nabla_d \mathbf{T}_i^{(k)}|}{\max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+)} \right) + \sum_i \sum_{d\in\{h,v\}} \frac{\alpha_t (\nabla_d \mathbf{T}_i)^2}{(\max(|\nabla_d \mathbf{T}_i^{(k)}|, \epsilon_t \to 0^+))^2} + \beta_t \|\mathbf{t} - \mathbf{t}_0\|_2^2. \tag{22}$$

It is easy to recognize that optimizing (22) is actually minimizing (19). The convergence property of MM for non-convex problems [37], like (20), establishes the claim. □

3.3 Convergence Speed & Complexity

We first discuss the convergence speed of our algorithm. It would be intuitive to see how quickly the algorithm converges with the number of iterations (K). Two cases including a dynamic only one and a dynamic/dynamic one are provided in Fig. 6. From the first row of Fig. 6, we can observe that, the curve of difference (defined as $\|\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}\|_2 / \|\mathbf{U}^{(0)}\|_2$) *versus* iteration rapidly drops and converges within 10 iterations. Further, the result at the 7th iteration is very close to that at the 20th iteration. The second case corresponds to a dynamic/dynamic case. It behaves similarly to the first test. The difference between the 6th and 20th iterations is unnoticeable quantitatively and qualitatively in both depth and RGB images. Please note that, for a better view of different settings, the difference plots are normalized into the range $[0, 1]$. For all the experiments shown in the paper, we set $K = 10$ according to the results reported in Fig. 6.

TABLE 1: Runtime (in sec) of muGIF (dynamic/dynamic) on inputs with different resolutions.

Resolution	$256 \times 256 \times 3$	$512 \times 512 \times 3$	$1024 \times 1024 \times 3$	$2048 \times 2048 \times 3$
PCG (10 iter.)	2.5	11.2	46.7	191.6
SA (3 iter.)	0.13	0.59	2.6	17.0
JFMS [28] (10|20 iter.)	0.42|0.85	3.69|7.4	16.87|35.22	68.15|138.68

For the complexity, as summarized in Alg. 1, our muGIF mainly iterates four steps. Among the four steps, the computational cost for obtaining $\mathbf{Q}_d$ and $\mathbf{P}_d$ is negligible as the weight values can be updated immediately with their definitions. The main computational burden is with computing $\mathbf{T}$ and $\mathbf{R}$ which requires solving the two linear systems Eq. (10) and Eq. (12) - both in the form of *weighted-least-square* (WLS) [21]. There are a number of iterative solvers that can apply for solving this, like standard *preconditioned conjugate gradient* (PCG). Accelerated solvers especially for this problem are also available using strategies like *separable approximation* (SA) [44]. Overall, the performance of these solvers is linear in the number of pixels. We here test the muGIF runtime on inputs with different resolutions on a PC with Intel i7 8700K@3.7GHz CPU and 32GB RAM. Table 1 summarizes the performance in the dynamic/dynamic mode using two different solvers (with no parallel computing): (1) Matlab build-in PCG solver, and (2) modified version of the fast solver† (Matlab + C++). It is worth to notice that in each iteration, the dynamic only and static/dynamic modes (one image to update) require half time of the dynamic/dynamic mode (two images to update). It can be observed the runtime is roughly linear in the number of pixels and with SA solver the computation is much faster. Note that SA using 3

†. https://sites.google.com/site/globalsmoothing/

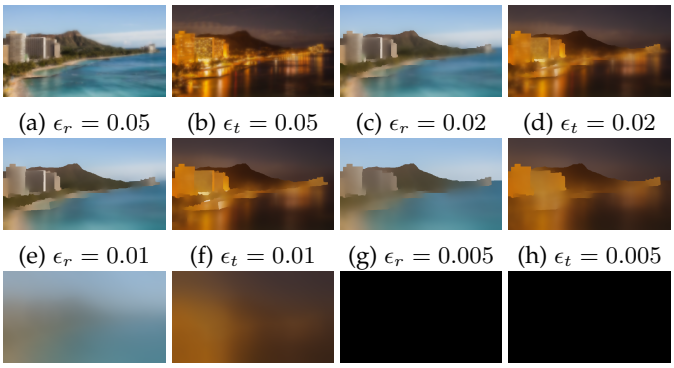


Fig. 7: Impact of ϵ_r and ϵ_t. The results are generated by fixing $\alpha_t = \alpha_r = 0.06$ and varying ϵ_r and ϵ_t.

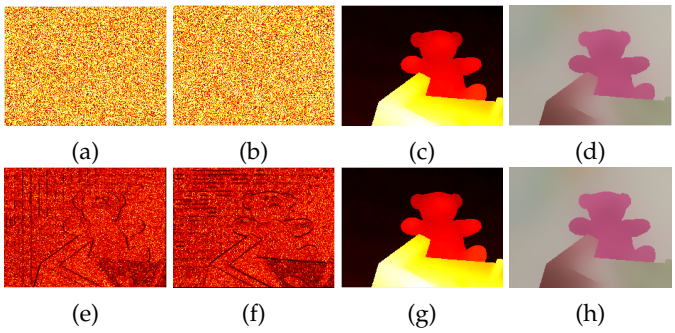


Fig. 8: Insensitive to initialization. (a) and (b) are randomly initialized horizontal and vertical weights. (c) and (d) are the results using (a) and (b) as initialization. (e) and (f) depict the weights computed from the input images. (g) and (h) are the corresponding results using (e) and (f) as initialization.

iterations can obtain similar visual quality with PCG using 10 iterations, hence we report SA using 3 iterations. From the view of processing mechanism, hierarchical and cascaded strategies like [45] can be applied to our problem for acceleration. We can achieve even more significant speed-up with GPU implementation of the solver to make real-time applications possible.

The previous state-of-the-art filtering works, such as WLS [21], L0GM [24], SD [29] and RTV [26], also need to solve linear systems like Eq. (10), thus the complexity of these works is the same as our muGIF on the self-guided and reference-guided filtering. As for mutually guided filtering (mutual structure extraction), JFMS [28] is in nature a locally linear transform model (local method), demanding less computation cost. We also provide the running time of JFMS in Tab. 1. It can be seen that JFMS is faster than muGIF, especially when the inputs are of low resolution. As the input size increases, the gap shrinks. Please note that we give two sets of running time for JFMS, because 10 iterations are not sufficient to generate the desired results, while the authors of JFMS suggested that 20 iterations should be used. In visual quality, muGIF shows its significant superiority over JFMS, please see Sec. 4.3 Mutual Structure Extraction.

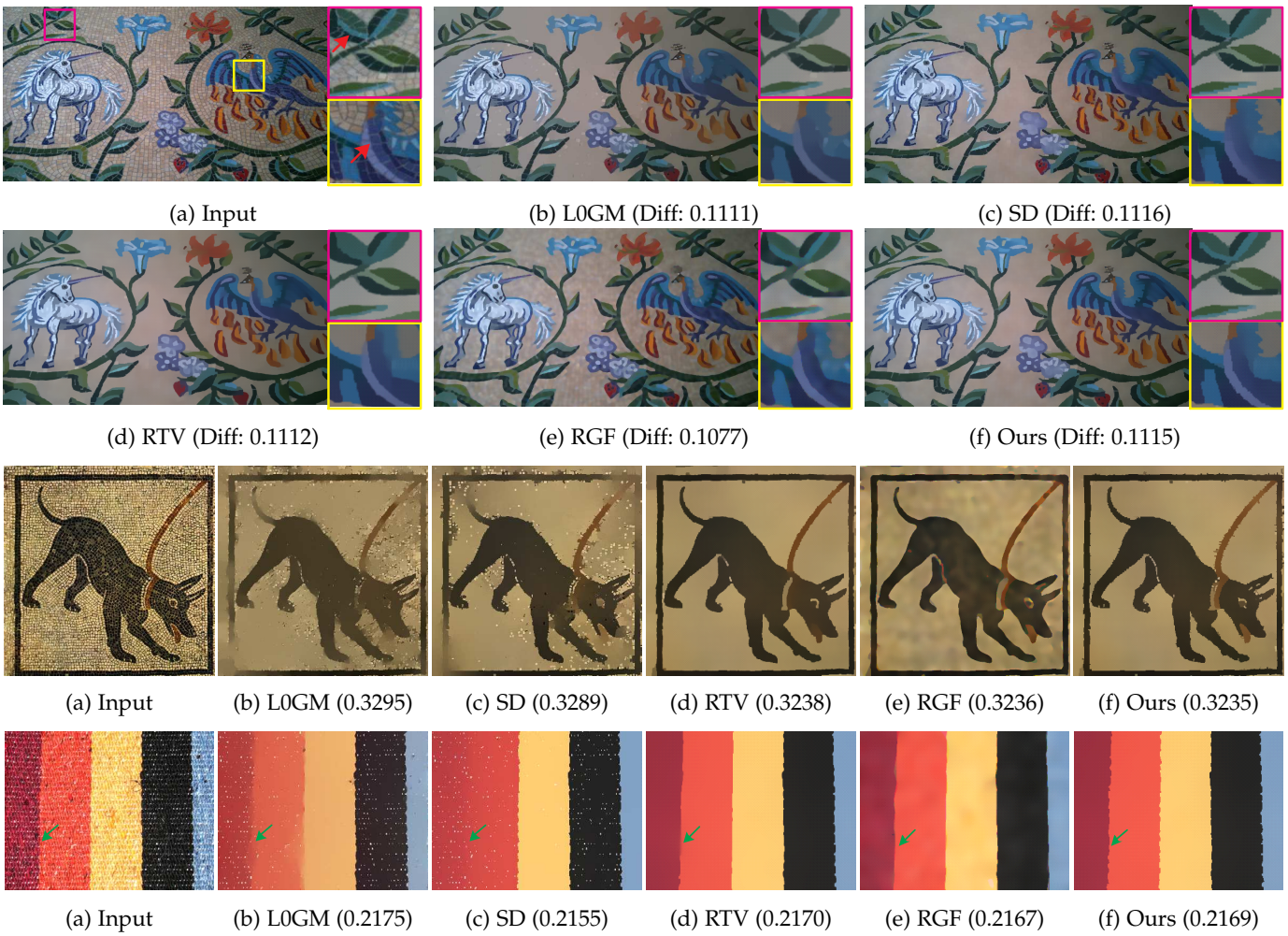
3.4 Parameter Effect & Initialization Insensitivity

In our muGIF (9), there are 6 parameters, including ϵ_r, ϵ_t, α_r, β_r, α_t and β_t. First, we fix the thresholds for gradient stability to $\epsilon_r = 0.01$ and $\epsilon_t = 0.01$ and focus on the rest 4 parameters. In fact, as can be seen in (10) and (12) (the update of $\mathbf{Q}_d$ and $\mathbf{P}_d$ does NOT involve these 4 parameters), the performance of muGIF is determined by α_t/β_t and α_r/β_r, which means that the number of free-parameters reduces from 4 to 2. By simply setting $\beta_t = 1$ and $\beta_r = 1$, only α_t and α_r remain.

From the form of (9), it is easy to tell that a larger α_t (or similarly α_r) leads to a smoother result $\mathbf{T}$ (or $\mathbf{R}$) than a smaller one. For more details about the theoretical explanation about the smoothing scale versus the parameter, please refer to [21]. We provide two examples including a texture removal and a scale-space filtering to experimentally show the parameter effect of α_t in Fig. 5. From the pictures, we can observe that as α_t grows, the smoothing effect increases, which corroborates the intuition and analysis. Thanks to our guidance strategy, the edges/structures at large scale are very well retained while textures/details at small scale are clearly removed. We also give the gradient map on bottom-right corresponding to each sub-picture for understanding the parameter effect of α_t from another perspective.

Here we show the impact of different values of ϵ_r and ϵ_t. Without loss of generality, a dynamic/dynamic (mutually guided) case is presented in Fig. 7. In this experiment, we fix $\alpha_r = \alpha_t = 0.06$ and vary ϵ_r and ϵ_t. As shown in Fig. 7 (a-b), we observe that the results are under-smoothed in most regions. This is because larger ϵ_r and ϵ_t make the gap between the surrogate function and the relative structure larger, and make the corresponding weight $\frac{1}{\max(|\nabla_d \mathbf{R}_i|, \epsilon_r)}$ ($\frac{1}{\max(|\nabla_d \mathbf{T}_i|, \epsilon_t)}$) over small when $|\nabla_d \mathbf{R}_i| < \epsilon_r$ ($|\nabla_d \mathbf{T}_i| < \epsilon_t$). As ϵ_r and ϵ_t decrease, this issue is mitigated, as shown in Fig. 7 (c-h). However, when ϵ_r and ϵ_t go to 0, the results suffer from over-smoothing as given in Fig. 7 (i-l). The reason comes from the instability and unbalance brought by too small ϵ_r and ϵ_t. In addition, when setting ϵ_r and ϵ_t to 0, the proposed algorithm exits abnormally due to zero denominator. Both the static/dynamic (reference-guided) and the dynamic only (self-guided) modes are in similar situations. All the results given in Sec. 4 are produced by setting $\epsilon_r = \epsilon_t = 0.01$.

From the perspective of non-convex optimization, the initialization would affect the final results. However, for images from different domains with the normalized range, our muGIF algorithm shows its insensitivity even with respect to random initialization. To verify this, Figure 8 gives a comparison on a depth and RGB image pair in dynamic/dynamic mode. The upper row in Fig. 8 contains the randomly initialized horizontal and vertical weights (*i.e.* $\mathbf{Q}_h^{(0)}\mathbf{P}_h^{(0)}$ and $\mathbf{Q}_v^{(0)}\mathbf{P}_v^{(0)}$, please refer to Alg. 1 for details) and the corresponding smoothing results, while the lower row includes the horizontal and vertical weights initialized according to the input images (*i.e.* computing $\mathbf{Q}_d^{(0)}$ and $\mathbf{P}_d^{(0)}$ based on $\mathbf{T}_0$ and $\mathbf{R}_0$) and the corresponding results. We can observe that, although the initializations are very different, the two manners do not show any noticeable difference, which confirms the insensitivity to initialization.



(a) Input (b) L0GM (Diff: 0.1111) (c) SD (Diff: 0.1116)

(d) RTV (Diff: 0.1112) (e) RGF (Diff: 0.1077) (f) Ours (Diff: 0.1115)

(a) Input (b) L0GM (0.3295) (c) SD (0.3289) (d) RTV (0.3238) (e) RGF (0.3236) (f) Ours (0.3235)

(a) Input (b) L0GM (0.2175) (c) SD (0.2155) (d) RTV (0.2170) (e) RGF (0.2167) (f) Ours (0.2169)

Fig. 9: Comparisons on texture removal. Results are by L0GM [24] ($\lambda = 0.021/0.35/0.18$, **top/middle/bottom**), SD [29] ($\lambda = 110/4.5e3/2e4$), RTV [26] ($\lambda = 0.009/0.025/0.07$, $\sigma = 2.0$), RGF [22] ($\sigma_s = 3.5/3.5/9$, $\sigma_r = 0.05$) and ours ($\alpha_t = 0.0075/0.05/0.12$), respectively.

4 Applications and Comparisons

4.1 Texture Removal & Scale-space Filtering

We compare our muGIF in the dynamic only mode with the recently proposed state-of-the-art competitors‡, including L0GM [24], RTV [26], RGF [22] and SD [29], on texture removal. For comparison fairness, we need to set a common smoothing level. To this end, we tune the parameter(s) for each method to reach a similar difference defined as $\|\mathbf{T} - \mathbf{T}_0\|_2 / \|\mathbf{T}_0\|_2$, which is denoted as Diff in figures.

Figure 9 depicts the visual results obtained by the competing techniques. From the first case, we observe that RGF has the problem of edge localization. L0GM, RTV, SD and muGIF outperform RGF in localizing edges. But, these methods are inferior to our muGIF in edge preservation – please refer to the zoomed-in patches. Two more comparisons are provided in Fig. 9. Specifically, the windowed weighting strategy of RTV is suitable for repetitive textures, while at high risk of wrongly filtering out strong but relatively dense edges [26], for instance the 'Z' shape intrinsic boundary on the neck of the bird in the first case, and the collar of the dog in the second case. As for L0GM, its problem comes from the "hard" ℓ_0 regularizer, which is expected to address the scale issue of the ℓ_1 regularizer and thus enhance the edge sparsity. However, its solver very likely sticks into bad minima because of the discreteness [24] – please refer to the second and third cases. Different from the others that update the guidance at each iteration, SD statically utilizes the input. The self-guided filtering is to smooth out undesired textures from itself, however the static component containing undesired information will consistently perform as a part of weight, which would hinder the desired smoothing effect in some regions, and thus lead to defects as shown in the second and third cases.

In addition, we provide a comparison on scale-space filtering in Fig. 10 at three scales altered by controlling the difference. RGF, due to its isotropic Gaussian kernel, poorly retains the boundaries especially at coarse scales. L0GM, though keeping some dominant boundaries, seems not so effective to determine the importance of edges. SD filter improves the result compared to L0GM, but the symptom is not eliminated throughly. RTV performs considerably well, which is, among others, closest to our muGIF. It is worth

‡. All the codes are downloaded from the authors' websites.

(a) Input (b) L0GM (Diff: 0.0997/0.1952/0.2267) (c) SD (Diff: 0.0982/0.1989/0.2290)

(d) RTV (Diff: 0.0992/0.1964/0.2248) (e) RGF (Diff: 0.0994/0.1953/0.2217) (f) Ours (Diff: 0.0995/0.1982/0.2294)

Fig. 10: Comparison on scale-space filtering. Results are obtained by L0GM [24] ($\lambda = 0.033/0.14/0.2$), SD [29] ($\lambda = 1.4e2/1.5e4/4.3e4$), RTV [26] ($\lambda = 0.0092/0.12/0.5$, $\sigma = 2.0$), RGF [22] ($\sigma_s = 3.9/23/700$, $\sigma_r = 0.05$) and ours ($\alpha_t = 0.01/0.3/2.0$), respectively.

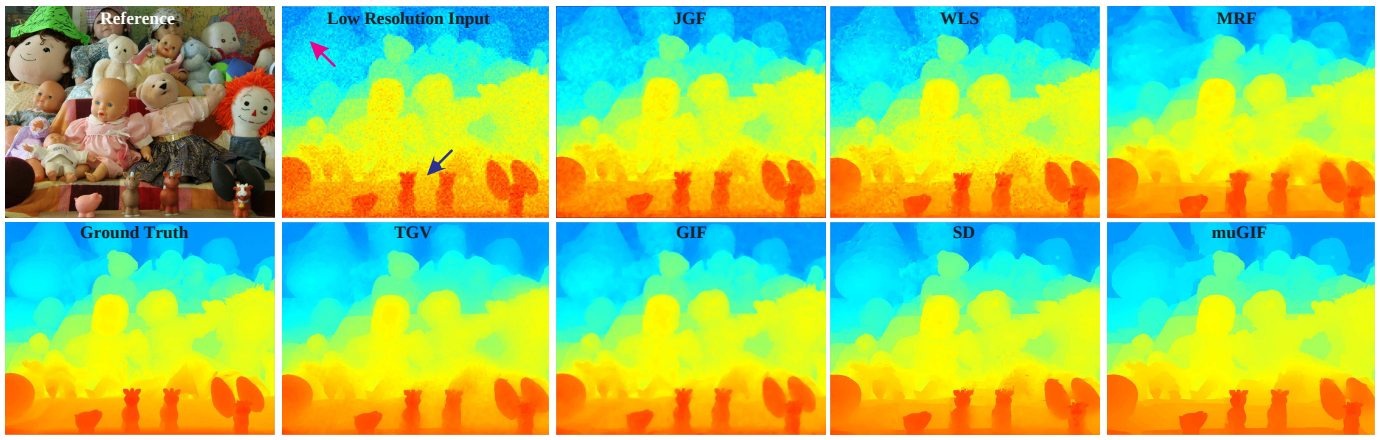


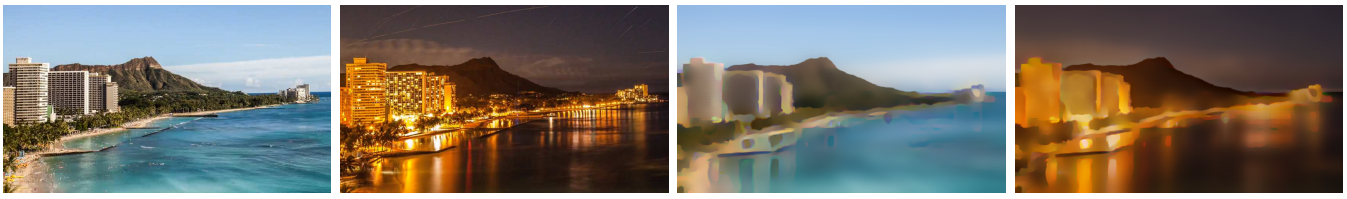
Fig. 11: Visual comparison on the 8x Dolls case. The inputs are the reference RGB and low-resolution noisy depth. Results are by JGF [46], WLS [21], MRF [47], TGV [48], GIF [27], SD [29] and ours, respectively.

mentioning that the windowed weighting strategy of RTV turns to be pixel-wise (same with our muGIF in dynamic only) when facing images/regions without repetitive patterns [26], like the case given in Fig. 10.

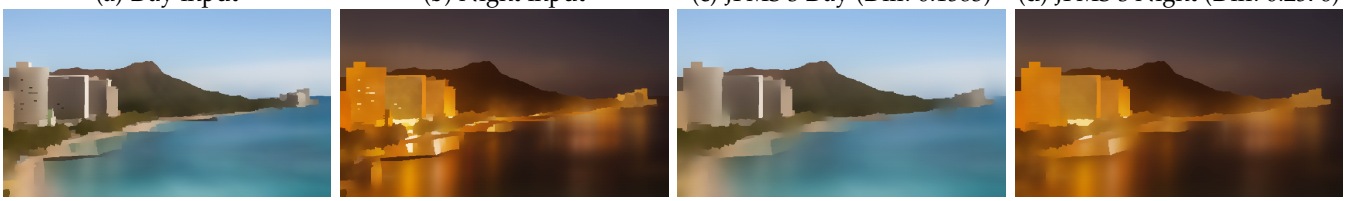
4.2 Depth Upsampling & No-flash/Flash Restoration

This part first assesses the performance of the proposed muGIF on a static/dynamic filtering task, say ToF depth upsampling. The datasets employed are from [49], containing six groups from Middlebury benchmarks[§]. The data are formed by introducing noise into depth images and downsampling them at four scales $\{2, 4, 8, 16\}$ to simulate ToF-like depth degradation. The upsampling and denoising can be jointly addressed by adopting a registered high-resolution RGB as a reference. Table 2 reports the *mean absolute difference* (MAD) between ground truth depth maps and the results by different methods including Bicubic, JGF [46], WLS [21], MRF [47], TGV [48], GIF [27], SD [29] and our muGIF. The proposed method consistently outperforms the other methods, like Bicubic, JGF, WLS, MRF, GIF and SD, for all the cases. Our muGIF, although falling behind TGV on the 2x cases with a slight 0.15 gap on average, achieves the second best performance. From the averages, we can clearly see that, on 4x, 8x and 16x cases, muGIF shows its superiority over the others. Moreover, the advance of muGIF gets more and more conspicuous as the upsampling rate increases, with muGIF *vs* the second best: 4x [0.97 *vs* 1.16], 8x [1.49 *vs* 1.89], 16x [2.56 *vs* 3.60], respectively. Figure 11 provides the visual results by the competitors on the 8x Dolls case. The inferior performance of SD is from its regularizer defined as $\sum_d \sum_i \exp(-\mu \nabla_d \mathbf{R}_i^2)(1-\exp(-\nu \nabla_d \mathbf{T}_i^2))/\nu$

§. http://vision.middlebury.edu/stereo/

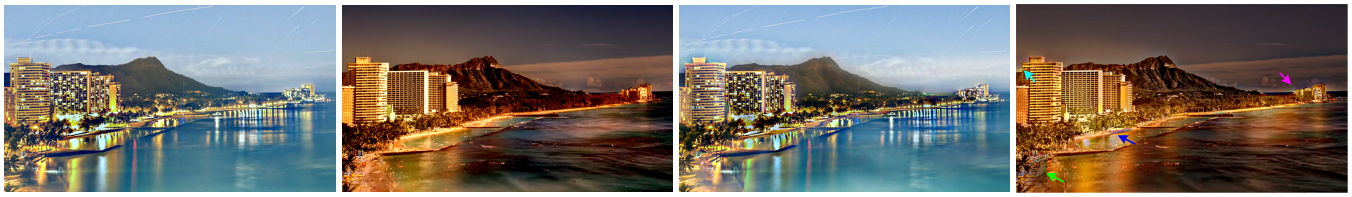


(a) Day input (b) Night input (c) JFMS's Day (Diff: 0.1583) (d) JFMS's Night (Diff: 0.2576)



(e) Our Day (Diff: 0.1491) (f) Our Night (Diff: 0.2665) (g) Our Day (Diff: 0.1676) (h) Our Night (Diff: 0.3006)

Fig. 13: Visual comparison on mutual structure extraction. (a) and (b) are the day and night images, respectively. (c) and (d) are the results by JFMS [28] ($\varepsilon_I = \varepsilon_G = 2e-4$ and $\lambda_I = \lambda_G = 1$). (e) and (f) are the results by muGIF ($\alpha_t = \alpha_r = 0.03$). (g) and (h) are the results by muGIF ($\alpha_t = \alpha_r = 0.06$).



(a) Day to night by muGIF (b) Night to day by muGIF (c) Day to night by JFMS (d) Night to day by JFMS

Fig. 14: Day-night transfer. (a) and (b) are produced by our muGIF. (c) and (d) are by JFMS [28].

Method	*Art*				*Book*				*Moebius*				*Reindeer*				*Laundry*				*Dolls*				Average			
	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x
Bicubic	3.52	3.84	4.47	5.72	3.30	3.37	3.51	3.82	3.28	3.36	3.50	3.80	3.39	3.52	3.82	4.45	3.35	3.49	3.77	4.35	3.28	3.34	3.47	3.72	3.35	3.49	3.76	4.31
MRF[47]	1.69	2.40	3.60	5.75	1.12	1.44	1.81	2.59	1.13	1.45	1.95	2.91	1.20	1.60	2.40	3.97	1.28	1.63	2.20	*3.34*	1.14	1.54	2.07	*3.02*	1.26	1.68	2.34	*3.60*
TGV[48]	**0.82**	**1.26**	*2.76*	6.87	**0.50**	**0.74**	*1.49*	2.74	**0.56**	**0.89**	1.72	3.99	**0.59**	**0.84**	*1.75*	4.40	**0.61**	1.59	*1.89*	4.16	**0.66**	1.63	1.75	3.71	**0.62**	*1.16*	*1.89*	4.31
WLS[21]	1.34	1.90	2.95	*4.63*	1.25	1.70	2.39	3.29	1.34	1.92	2.66	3.56	1.47	2.05	2.82	4.09	1.11	1.55	2.24	3.49	1.34	1.85	2.55	3.50	1.31	1.83	2.60	3.76
JGF [46]	2.36	2.74	3.64	5.46	2.12	2.25	2.49	3.25	2.09	2.24	2.56	3.28	2.18	2.40	2.89	3.94	2.16	2.37	2.85	3.90	2.09	2.22	2.49	3.25	2.17	2.37	2.82	3.85
GIF [27]	1.49	1.97	3.00	4.91	0.80	1.22	1.95	3.04	1.18	1.90	2.77	3.55	1.29	1.99	2.99	4.14	1.28	2.05	3.04	4.10	1.19	1.94	2.80	3.50	1.21	1.85	2.76	3.87
SD [29]	1.05	*1.66*	3.16	5.78	0.77	0.98	1.53	2.74	0.81	1.08	*1.66*	*2.85*	0.89	1.21	1.89	*3.82*	0.86	*1.20*	1.94	3.82	*0.85*	*1.14*	*1.74*	3.05	0.87	1.21	1.99	3.67
muGIF	*1.00*	**1.26**	**2.00**	**3.46**	*0.73*	*0.89*	**1.35**	**2.15**	*0.67*	*0.85*	**1.35**	**2.25**	*0.78*	*0.94*	**1.39**	**2.52**	*0.64*	**0.87**	**1.36**	**2.57**	*0.85*	**1.04**	**1.50**	**2.45**	*0.77*	**0.97**	**1.49**	**2.56**

TABLE 2: Quantitative comparison of the depth upsampling task in terms of MAD on the data from Middlebury benchmarks [49]. The best results are highlighted in bold, while the second best ones are underlined and in italic.

with μ and ν being two coefficients. Since the reference is fixed, by eliminating the effect of constant terms and the constant coefficient ν, we obtain $\sum_d \sum_i \frac{-1}{\exp(\mu \nabla_d \mathbf{R}_i^2 + \nu \nabla_d \mathbf{T}_i^2)}$, which shows an additive relation between $\nabla_d \mathbf{R}_i^2$ and $\nabla_d \mathbf{T}_i^2$. Considering the characteristic of $\frac{-1}{\exp(\mu \nabla_d \mathbf{R}_i^2 + \nu \nabla_d \mathbf{T}_i^2)}$, for regions with relatively small $\nabla_d \mathbf{R}_i^2$, SD may over-smooth those regions in the target image because a slight $\nabla_d \mathbf{T}_i^2$ will cause a great cost (please see the boundaries of the toy horses in Fig. 11). While for regions with very large $\nabla_d \mathbf{R}_i^2$, SD may under-smooth those regions in the target image or even generate new edges since an intense $\nabla_d \mathbf{T}_i^2$ will only result in a small penalty (please see the top-left region corresponding to the green hat region in the reference with strong textures in Fig. 11).

In addition, our method is also applicable to flash/no-flash image restoration. We use the flash image to guide image restoration on the corresponding no-flash noisy image. A comparison between JBF [1], GIF [27], SD [29] and muGIF is presented in Fig. 12. JBF and GIF can suppress noise, but also blur both structures and textures. SD is much better at preserving details than JBF and GIF, but in trouble with staircase-like artifacts. Overall, our recovered results are sharp and clean.

4.3 Mutual Structure Extraction

This part tests the dynamic/dynamic filtering ability of muGIF on extracting shared structures between two images captured from distinct domains and/or under different conditions. There are few works specifically developed for this task. Recently, JFMS [28] has been proposed to fulfill such demand, which is employed as the compared method. An example on a pair of noisy depth and RGB images is shown in Fig. 1 (g)-(i). From the results, we can see that, given the inputs, both JFMS and muGIF can effectively distill the common structures. By taking a closer look at the pictures, we find that our muGIF ($\alpha_t = 0.005$ for depth and $\alpha_r = 0.02$ for RGB) exceeds JFMS ($\varepsilon_I = \varepsilon_G = 5e-5$, $\lambda_I = 0$ for depth and $\lambda_G = 100$ for RGB) in both noise suppression on depth and structure extraction on RGB. Another comparison executed

(a) Flash image (c) Jointly denoised by MJIR [50] (e) muGIF Dyn./Dyn. + detail transf.

(b) No-flash image (d) Jointly denoised by muGIF Sta./Dyn. (f) Light enhancement on (e) by LIME [51]

Fig. 15: Flash/no-flash transfer based on mutual structure extraction.

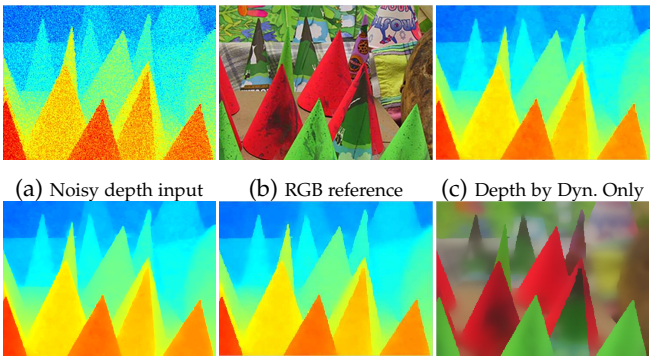
Fig. 12: Qualitative comparison on flash/no-flash image restoration. The parameters are tuned for the methods to obtain their best possible results.

on a couple of day and night images from a complex scene is given in Fig. 13. In this experiment, JFMS can not offer satisfactory results with heavy visual artifacts introduced, as can be seen from Fig. 13 (c) and (d). The reason of such a failure is probably that both the two inputs are of complex details, so that the patch level measurement of structure similarity degrades or loses its ability to accomplish the task. We notice that JFMS's effectiveness on the previous noisy depth/RGB case is because one of the two images (the depth) is, although noisy, simply structured. Our results, as shown in Fig. 13 (e) and (f), significantly improve those by JFMS, for example the building facades and water surface. Further, we provide one more pair of results by turning up the smoothing parameters as shown in Fig. 13 (g) and (h), from which, we see that more details disappear while the dominant structures still exist steadily.

To intuitively validate the advantage of our muGIF over JFMS, we conduct a comparison on an interesting task, which tries to transfer features/details of one of day and night images to the other based on the extracted mutual structures. As shown in Fig. 14 (a) and (b), the transfered results based on our muGIF very well maintain the original day's and night's basic appearances, meanwhile their details/features are exchanged¶. For example, colorful lights on the water surface of the day, and ambient light on the building facade of the night. Though the results by JFMS are reasonable, the halo and feature-residue artifacts frequently appear in Fig. 14 (c) and (d).

Inspired by the day-night transfer idea, we wonder whether it is possible that a no-flash image containing noise and unbalanced lights can be enhanced by its corresponding flash image. Figure 15 positively responses, in which (a) and (b) are the flash and no-flash inputs. We can see from Fig. 15 (b) that the no-flash image hides noise, shadows, highlight and low-light regions. As previously shown in Fig. 12, the flash image can be employed to guide denoising on the no-flash. Figure 15 (c) and (d) are the denoised versions by MJIR [50] (the state-of-the-art method for multispectral joint image restoration) and muGIF in static/dynamic respectively, which are very close and well-done from the view of denoising. However, the unbalanced light remains. Figure 15 (e) gives the result transfered from

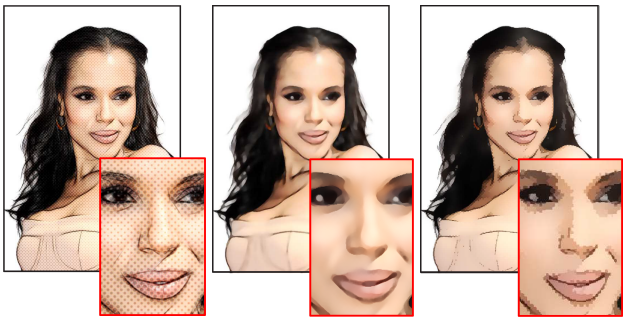
¶. The residuals between the original signal and the extracted mutual structure act as the details/features.



(a) Noisy depth input (b) RGB reference (c) Depth by Dyn. Only

(d) Depth by Sta./Dyn. (e) Depth by Dyn./Dyn. (f) Our mutual structure

Fig. 16: Comparison of muGIF's three modes. (c) is the filtered depth using itself as the guidance. (d) is obtained by employing the RGB as the reference. (e) and (f) are the mutually guided results.



(a) Input (b) Result by RGF (c) Our Result

Fig. 17: Limitation. (a) is a halftone example. (b) and (c) are inverse halftoning results by RGF [22] and our muGIF in dynamic only, respectively.

the flash image, which reveals that the lighting unbalance issue is greatly addressed and the transfered textures are cleaner and sharper than those in (c) and (d). Using low-light enhancement techniques like [51] can further adjust brightness as shown in Fig. 15 (f). Moreover, we conduct an experiment to reveal the difference of muGIF between the three modes including the dynamic only mode (self-guided, Fig. 16 (c)), the static/dynamic one (RGB guided, Fig. 16 (d)) and the dynamic/dynamic one (mutually guided, Fig. 16 (e) and (f)).

5 Conclusion and Limitation

Image filters are fundamental and important tools for visual data processing. This paper has defined a novel measurement, *i.e.* relative structure, to manage the structure similarity between two input signals. Based on the new measure, a robust joint image filter, called mutually guided image filter (muGIF), has been proposed, which is flexible to perform in one of the dynamic only, static/dynamic and dynamic/dynamic modes. A global optimization objective and an effective algorithm have been designed to achieve high-quality filtering performance. To verify the efficacy of muGIF and demonstrate its advantages over other state-of-the-arts, the experimental results on a number of applications, such as texture removal, scale-space filtering, depth upsampling, flash/no-flash image restoration and mutual structure extraction, have been conducted.

Our current muGIF exposes its limitation when the design principle is violated. For instance, when an RGB image is intruded by corruptions, especially on tiny structures, the restored result even with another assistance (*e.g.* NIR) will not be satisfied. This situation is encountered not just by our muGIF but also by (most of) existing image filters. We note that the work [50] is specified to multispectral joint image restoration with promising performance. Another situation is halftone-like cases, as shown in Fig. 17, in which the advantage of precisely localizing edges instead becomes the disadvantage. In contrast, RGF can produce a visually better result.

Acknowledgement. We thank Semir Elezovikj for carefully proofreading the manuscript. We also thank the anonymous reviewers for valuable comments and suggestions.

References

[1] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital photography with flash and non-flash image pairs," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 664–672, 2004.
[2] X. Shen, Q. Yan, L. Ma, and J. Jia, "Multispectral joint image restoration via optimizing a scale map," *IEEE Trans. PAMI*, vol. 37, no. 12, pp. 2518–2530, 2015.
[3] E. Gastal and M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, p. 69, 2011.
[4] C. Cao, S. Chen, W. Zhang, and X. Tang, "Automatic motion-guided video stylization and personalization," in *ACM MM*, pp. 1041–1044, 2011.
[5] J. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *ACM MM*, pp. 161–170, 2005.
[6] K. Yoon and I. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 650–656, 2006.
[7] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. PAMI*, vol. 35, no. 2, pp. 504–511, 2013.
[8] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. PAMI*, vol. 34, no. 9, pp. 1744–1757, 2012.
[9] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *CVPR*, pp. 1164–1172, 2015.
[10] T. Zhou, Y. Lee, S. Yu, and A. Efros, "FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences," in *CVPR*, pp. 1191–1200, 2015.
[11] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *CVPR*, pp. 3475–3484, 2016.
[12] R. Gonzalez and R. Woods, *Digital Image Processing*. Prentice Hall, 2002.
[13] J. van de Weijer and R. van den Boomgaard, "Local mode filtering," in *CVPR*, pp. II–428–II–433, 2001.
[14] B. Weiss, "Fast median and bilateral filtering," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 519–526, 2006.
[15] M. Kass and J. Solomon, "Smoothed local histogram filters," *ACM Trans. Graph.*, vol. 29, no. 4, p. 100, 2010.
[16] Z. Ma, K. He, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *ICCV*, pp. 49–56, 2013.
[17] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *CVPR*, pp. 2830–2837, 2014.
[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, pp. 839–846, 1998.
[19] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, no. 3, p. 103, 2007.

[20] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 637–645, 2006.
[21] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, p. 67, 2008.
[22] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *ECCV*, pp. 815–830, 2014.
[23] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. PAMI*, vol. 12, no. 7, pp. 629–639, 1990.
[24] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via ℓ_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, p. 174, 2011.
[25] L. Rudin, S. Osher, and E. Ftemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.
[26] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, p. 139, 2012.
[27] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. PAMI*, vol. 35, no. 6, pp. 1397–1409, 2013.
[28] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *ICCV*, pp. 3406–3414, 2015.
[29] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Trans. PAMI*, 2017.
[30] X. Guo, Y. Li, and J. Ma, "Mutually guided image filtering," in *ACM MM*, pp. 1283–1290, 2017.
[31] D. Krishnan and R. Szeliski, "Multigrid and multilevel preconditioners for computational photography," *ACM Trans. Graph.*, vol. 30, no. 6, 2011.
[32] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.
[33] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 646–653, 2006.
[34] R. Szeliski, "Locally adapted hierarchical basis preconditioning," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1135–1143, 2006.
[35] I. Daubechies, R. DeVore, M. Fornasier, and C. Güntürk, "Iteratively reweighted least-squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.
[36] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.
[37] D. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.
[38] A. Hamidi, M. Ménard, M. Lugiez, and C. Ghannam, "Weighted and extended total variation for image restoration and decomposition," *Pattern Recognition*, vol. 43, pp. 1564–1576, 2010.
[39] M. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Weighted ℓ_1-minimization for sparse recovery with prior information," in *IEEE International Symposium on Information Theory*, pp. 483–487, 2009.
[40] D. Needell, R. Saab, and T. Woolf, "Weighted ℓ_1-minimization for sparse recovery under arbitrary prior information," *Information and Inference: A Journal of the IMA*, vol. 00, pp. 1–26, 2017.
[41] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.
[42] X. Guo and Y. Ma, "Generalized tensor total variation minimization for visual data recovery," in *CVPR*, pp. 3603–3611, 2015.
[43] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Trans. Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.
[44] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. Do, "Fast global image smoothing based on weighted least squares," *IEEE Trans. Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.
[45] Y. Li, D. Min, M. Do, and J. Lu, "Fast guided global interpolation for depth and motion," in *ECCV*, pp. 717–733, 2016.
[46] M. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *CVPR*, pp. 169–176, 2013.
[47] J. Park, H. Kim, Y. Tai, M. Brown, and I. Kweon, "High quality depth map upsampling for 3d-tof cameras," in *ICCV*, pp. 1623–1630, 2011.
[48] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *ICCV*, pp. 993–1000, 2013.
[49] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang, "Color-guided depth recovery from rgbd data using an adaptive autoregressive model," *IEEE Trans. Image Processing*, vol. 23, no. 8, pp. 3443–3458, 2014.
[50] X. Shen, Q. Yan, L. Ma, and J. Jia, "Multispectral joint image restoration via optimization a scale map," *IEEE Trans. PAMI*, vol. 37, no. 12, pp. 2518–2530, 2015.
[51] X. Guo, Y. Li, and H. Ling, "LIME: Low-light image enhancement via illumination map estimation," *IEEE Trans. Image Processing*, vol. 26, no. 2, pp. 982–993, 2017.

Xiaojie Guo (M'13) received his Ph.D. degree in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China. He is currently an Associate Professor with tenure (Peiyang Young Scientist) at Tianjin University. Prior to joining TJU, he spent about 4 years at the Institute of Information Engineering, Chinese Academy of Sciences. He was a recipient of the Piero Zamperoni Best Student Paper Award in ICPR 2010, and the Best Student Paper Runner-up in ICME 2018.

Yu Li (M'16) received his Ph.D. degree in National University of Singapore. He is now with Advanced Digital Sciences Center, a research center founded by University of Illinois at Urbana-Champaign (UIUC) and the Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include computer vision, computational photography, and computer graphics.

Jiayi Ma received the B.S. degree from the Department of Mathematics and the Ph.D. degree from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently an Associate Professor with the Electronic Information School, Wuhan University, Wuhan.

Haibin Ling received his B.S. and M.S. degrees from Peking University in 1997 and 2000, respectively, and his Ph.D. degree from the University of Maryland, College Park in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia. From 2006 to 2007, he worked as a postdoctoral scientist at the University of California Los Angeles. After that, he joined Siemens Corporate Research as a research scientist. Since fall 2008, he has been with Temple University where he is now an Associate Professor. He received the Best Student Paper Award at the ACM UIST in 2003, and the NSF CAREER Award in 2014. He serves as Associate Editors for IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Pattern Recognition (PR), and Computer Vision and Image Understanding (CVIU). He has also served as Area Chairs for CVPR 2014, CVPR 2016 and CVPR 2019.