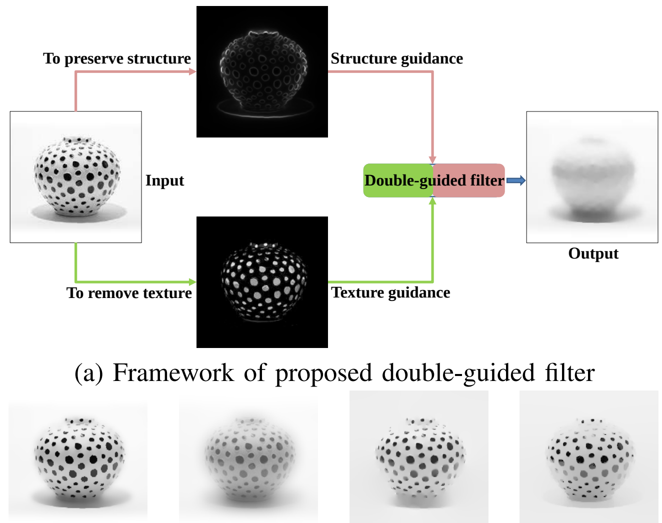
# Double-Guided Filtering: Image Smoothing with Structure and Texture Guidance

Kaiyue Lu, Shaodi You, Nick Barnes
Data61, CSIRO
Research School of Engineering, Australian National University
Email: {Kaiyue.Lu, Shaodi.You, Nick.Barnes}@data61.csiro.au

***Abstract*—Image smoothing is a fundamental technology which aims to preserve image structure and remove insignificant texture. Balancing the trade-off between preserving structure and suppressing texture, however, is not a trivial task. This is because existing methods rely on only one guidance to infer structure or texture and assume the other is dependent. However, in many cases, textures are composed of repetitive structures and difficult to be distinguished by only one guidance. In this paper, we aim to better solve the trade-off by applying two independent guidances for structure and texture. Specifically, we adopt semantic edge detection as structure guidance, and texture decomposition as texture guidance. Based on this, we propose a kernel-based image smoothing method called the double-guided filter (DGF). In the paper, for the first time, we introduce the concept of texture guidance, and DGF, the first kernel-based method that leverages structure and texture guidance at the same time to be both 'structure-aware' and 'texture-aware'. We present a number of experiments to show the effectiveness of the proposed filter.**




(a) Framework of proposed double-guided filter

(b) BLF [4] (c) GF [5] (d) L0 [7] (e) WLS [8]

Fig. 1. Image smoothing with kernel-based and optimization-based methods. The dotted texture has strong edges that will be mislabeled as structure in existing methods. Using the proposed double-guided filter can solve such a problem.

## I. Introduction

Image smoothing, which aims to preserve the important structure or edges and remove insignificant details or texture within the structure, plays an important role in many computer vision applications, such as image abstraction [1], detail enhancement [2], image denoising [3], etc. Existing image smoothing methods can be roughly classified into two types: kernel-based local smoothing, and optimization-based structure and texture separation. Kernel-based methods emphasize 'structure-awareness'. For example, the bilateral filter (BLF) [4] and guided filter (GF) [5] calculate a local average of intensities by convolving with a positive kernel. This operation can retain large gradients by adjusting weights of neighboring pixels according to their color intensities. The averaging operation is able to suppress weak texture or noise (small oscillations with low contrast) effectively. However, as Zhang et al. [6] have pointed out, the essential deficiency of this type of method is the lack of discrimination of strong texture (insignificant details with high contrast) and main structure. For example, as shown in Fig. 1, the input image contains a vase covered with black dots. The removal of these black dots will not affect our cognition of the vase, thus we regard them as insignificant details that can be removed. Unfortunately, the dots are textures with strong edges, which will be mislabeled as '*significant structure*' in existing methods. Specifically, BLF fails in removing these textures because the contrast between dots and background is too large, in which case BLF preserves them as edges, as shown in Fig. 1(b). GF fails due to the same reason as well (shown in Fig. 1(c)).

In contrast, image separation methods emphasize 'texture-awareness', which aims to extract the texture from the image by optimizing a globally-defined objective function based on the assumption that an image can be decomposed into structure and texture layers. For example, L0-smoothing [7] manipulates the total number of non-zero gradients, and WLS [8] leverages the total variation between two layers in terms of gradients. As Fig. 1(d-e) show, the edges are over-smoothed and 'halo' artifacts occur if the method attempts to remove all the texture (the over-smoothing is especially serious at the base of the vase, which has low contrast but important semantic meaning). That is, optimization-based methods always have to trade off between removing texture and preserving structure.

Our idea is to combine the benefits of both methods: we take the advantage of kernel-based methods for 'structure-awareness', and optimization-based methods for 'texture-awareness'. In this paper, we regard image smoothing as denoising and cartoon/texture separation that are commonly associated with structure preservation, the same task as [1-4, 7, 11-12, 15-16, 21-23, 27-32]. We design a double-guided kernel-based filter (**DGF**), which is able to preserve meaningful structure with the guidance of newly-proposed semantic edge detection [9] (**structure guidance**), and distinguish and remove texture with the guidance of image separation [10] (**texture guidance**) without over-smoothing any structure. Fig. 1(a) illustrates our concept and shows an example of our result compared with existing methods. More importantly, the kernel only takes two parameters corresponding to 'structure-awareness' and 'texture-awareness', and is easy to use.

The contributions of the proposed method are threefold:
1. We give theoretical insights on balancing 'structure-awareness' and 'texture-awareness' for image smoothing.
2. It is the first time that structure and texture guidance have been applied simultaneously to image smoothing. More specially, the two guidance images are generated independently.
3. The proposed easy-to-use double-guided filter outperforms existing methods on simultaneous 'structure-awareness' and 'texture-awareness'. That is, it can remove even stronger textures without blurring other important structures.

The rest of the paper is organized as follows: Section II introduces related work on image smoothing. Section III illustrates the motivation of structure and texture guidance. Section IV is the formulation of our proposed DGF. A number of experiments and analysis are demonstrated in Section V. Section VI gives a conclusion.

## II. Related work

### A. Kernel-based image smoothing

In essence, the most important aspect in this type of method is the calculation of weights. In the bilateral filter [4], the weights are determined by the color intensity difference (range kernel) and spatial distance (spatial kernel) at the same time. The joint bilateral filter [11] extends on the bilateral filter, obtaining the range kernel from another image called the guidance image. If the guidance image is fixed throughout the smoothing process, it is known as static guidance [5], [12]. In contrast, dynamic guidance means the guidance image will be updated after each iteration [13]. A new direction is to combine static and dynamic guidance [14], [15]. Additionally, [16] utilizes region covariances to measure the difference between two kernels. In [6], the double weights are dependent on the tree distance [17] between pixels and the overlapping areas between superpixels. Other related methods use propagation distance [18], a co-occurrence matrix [19], patch shift [12], multipoint estimation [20], [21], or fully connected regions [22]. In our work, we obtain the weights from static structure guidance and texture guidance.

### B. Optimization-based structure and texture separation

Unlike kernel-based image smoothing, optimization-based structure and texture separation methods focus on defining an objective function based on image gradients and then finding the optimal solution. As [8] pointed out, the objective of global optimization is to make the output image as smooth as possible while as close as possible to the input. In early work, Total Variation (TV) [23] was proposed to separate structure from texture, based on the assumption that an image can be decomposed into a structure and a texture layer, and they are totally unrelated. In [8], partial derivatives of the input image are added to the objective function, which is optimized with weighted least squares. In [24], the regularizer was replaced with Relative Total Variation (RTV). Both are based on the L2 norm. It has been suggested that in image filtering, the L1 and L0 norm may be more effective than the L2 norm [25]. [26] utilizes the L1 norm to define the objective function, which contains local flattening, global sparsity, and image approximation. One popular L0 norm based method was proposed in [7], which counts and manipulates the number of non-zero gradients. Further explorations have been made to improve this, including minimization by region fusion [27], [28]. [10] found that the correlation between the structure gradients and the texture layer is extremely low in the ideal decomposition. The Structure Gradient Texture Decorrelating (SGTD) regularizer thus models both the structure and texture layers. One method leverages both global and local optimization [29]. It regards edges as high variance in range values of neighbouring local extrema and details as oscillations between local extrema. Despite the trade-off nature of this type of method, good performance in separating texture helps us to construct the texture guidance map, which will be discussed in Section III.

## III. Structure Guidance and Texture Guidance

### A. Motivation

To the best of our knowledge, existing guided image smoothing methods depend largely on structure guidance. However, this is not sufficient because in most circumstances, texture may also have distinct edges, which will confuse structure guidance. Thus, we need texture guidance to tell the filter where to smooth more when encountering strong textures. We note that optimization-based methods decompose the image into structure and texture layers, which achieves a tradeoff between preserving structure and removing texture (it outperforms kernel-based methods in texture removal). Here we use the texture layer to guide local filtering. To be more specific, texture guidance reduces the possibility of preserving insignificant details caused by texture. The two guidance will be introduced in the following.

### B. Structure Guidance

Traditional structure guidance is derived from a pre-generated image (a smoothed copy of the input). One important goal of image smoothing is to keep the output as close to the input as possible. The high similarity between the input and guidance images limits the ability of specific smoothing. Recent work has attempted to construct structure maps based on pixel gradients and then use them as guidance [12], [30]. However, large gradients do not always correspond to the structure because some strong textures may also have large gradients along their edges. This phenomenon can be found in both Fig. 1 (the black dots on the vase) and Fig. 2

(the black dots in the background, which have large gradients as shown in Fig. 2(b)).

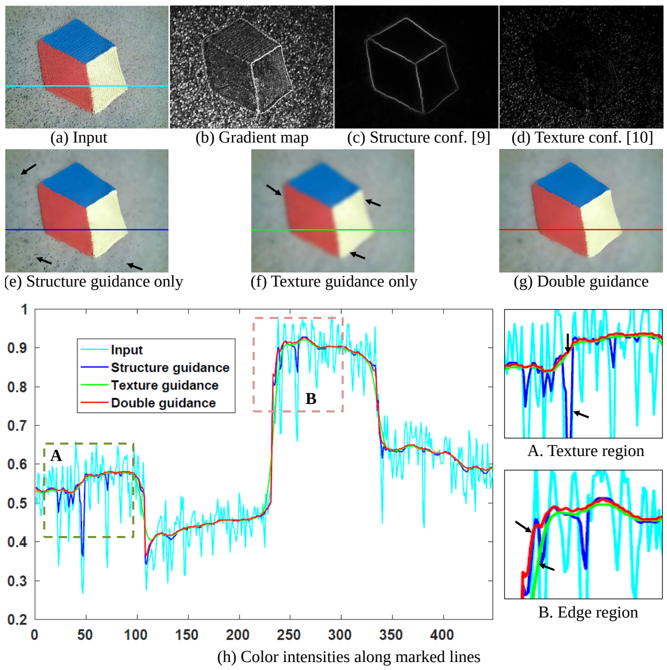



Fig. 2. Illustration of double guidance process.

Semantic edge detection [31] has provided a new direction for generating structure guidance [32]. Detection is based on training from hundreds of human-labeled samples. Using this, the filter can preserve meaningful structure, to produce images that correspond better to human perception. Inspired by this, we use a state-of-the-art semantic edge detection method [9] to construct the structure map. Fig. 2 shows the gradient (b) and semantic structure map (c) (larger pixel values are more likely to be structure). By comparison, the gradient map widely used by traditional methods is largely affected by textures while the semantic structure map eliminates the interference of them. Thus, the semantic map is more suitable for structure guidance. We denote the structure confidence map as $E$.

### C. Texture Guidance

The essential reason for the tradeoff in optimization-based methods is that they assume the original input can be exactly decomposed into two layers (structure and texture layers are negative correlated because the total sum is fixed). This is the natural deficiency of total variation based methods. We note that although the method in [10] (SGTD) is also based on this assumption, it explores a new way to minimize the correlation between structure gradient and texture components (magnitude). The two measurements are different so that we can regard the two layers are independently generated. This method has been shown to outperform most existing optimization-based methods in removing textures. Thus, we utilize this approach for texture guidance. We normalize the magnitude of the texture layer to $[0, 1]$ and take it as the texture confidence map, as shown in Fig. 2(d) (larger magnitude corresponds to higher contrast, implying the texture component is harder to remove). We denote the texture confidence map as $T$.

## IV. Double-guided filter

Based on the analysis above, we define an easy-to-use double-guided filter which only relies on one parameter for each guidance. In detail, given input $I$, the filtering is given as:

$$S_p = \frac{1}{\kappa_p} \sum_{q \in \Omega} w_s(p, q) \cdot w_t(q) \cdot I_q$$

where $S_p$ is the output pixel value, and $\Omega$ is a $k \times k$ square kernel centered at $p$. $w_s(p, q)$ and $w_t(q)$ denotes the weights from structure and texture guidance respectively. $\kappa_p = \sum_{q \in \Omega} w_s(p, q) \cdot w_t(q)$ is used for normalization.

*1) Structure weight:* $w_s(p, q)$ takes the form of:

$$w_s(p, q) = (1 - E(q)) \cdot \exp(\frac{-\|I(p) - I(q)\|^2}{2\sigma_s^2})$$

where $E(q)$ denotes the edge confidence at pixel $q$, and $\sigma_s$ is a user-specified parameter. The right part of the structure weight is the range kernel found in the bilateral filter, which modulates smoothing by color intensity difference. This kernel essentially prevents attenuation of strong textures in the bilateral filter because both main structures and strong textures have large color difference. The left-hand part multiplies by $(1 - E(q))$, so that *color difference will be retained unless the structure confidence is relatively low*. Even though some part of the structure is weak (has low contrast), the guidance lowers the weights to preserve color difference. Although $(1 - E(q)) \in [0, 1]$, it can indeed make a difference after normalization.

*2) Texture weight:* $w_t(q)$ takes the form of:

$$w_t(q) = \exp(\frac{-T(q)^2}{2\sigma_t^2})$$

where $T(q)$ denotes the texture confidence at pixel $q$, and $\sigma_t$ is a user-specified parameter. Texture weight replaces the spatial (Gaussian) kernel in the bilateral filter. Intuitively, pixels with high texture confidence should be smoothed whereas those with low confidence should be preserved. Thus, we assign small weights to pixels with higher texture confidence.

*3) Effect of single and double guidance:* The highlight of the proposed DGF is that structure and texture guidance support each other to preserve structure and remove texture. To illustrate this, Fig. 2 shows (c) structure confidence and (d) texture confidence, which are used for (e) structure guidance only, (f) texture guidance only, and both are used for (g) the proposed DGF based on (a) the input image. Intuitively, the result using only structure guidance has sharpened edges but some strong texture is still retained. In contrast, the result using only texture guidance does not contain any insignificant textures but the overall structure appears blurred. To visualize the effect, we plot the color intensity distribution along one line (marked in the images) in Fig. 2(h). We first find that in texture regions (e.g., dashed box A), the results with texture guidance and double guidance overlap in most circumstances

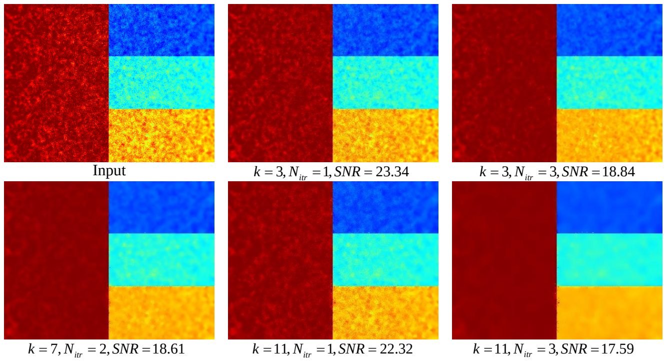



Fig. 3. Double-guided filtering with different kernel sizes and iterations.

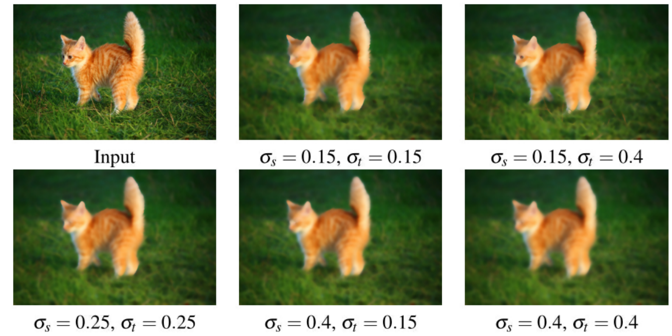



Fig. 4. Double-guided filtering with different $\sigma_s$ and $\sigma_t$.

while the structure-guided result shows apparent deviation or oscillations. This is because the semantic structure map is not perfect and still cannot eliminate the negative effect of some strong textures. In contrast, in regions with structural edges (e.g., dashed box B), the results with structure guidance and double guidance are almost the same (except the texture regions near the edge), indicating that our DGF has preserved the structure well. However, the result of texture guidance in this case is inadequate because the edges are over-smoothed (the green line is more rounded and less sharp) as they are not explicitly preserved. It is clear that our proposed DGF has combined the benefits of two guidance effectively.

## V. Experiments and analysis

### A. Parameter adjustment

*1) Kernel size and iterations:* In our method, the kernel size, $k$, and the number of iterations, $N_{itr}$, determine the scale of textures to be smoothed and the extent of texture suppression respectively. Fig. 3 shows the smoothing results with various kernel sizes and iterations to an image with artificial random noise. We examine the signal-to-noise-ratio (SNR) to measure the effect of removing noise quantitatively. Compared with the noisy input, a smaller SNR indicates that noise is better suppressed. With increasing kernel size, larger scale texture is more easily removed. This can be also achieved by increasing the number of iterations. Empirically, 3-5 iterations with kernel size of $\{5, 7, 9, 11\}$ can yield desirable results.

*2) Smoothing effect factors $\sigma_s$ and $\sigma_t$:* The two parameters control the effect of smoothing in terms of preserving structure and removing texture respectively. Normally, a smaller $\sigma_s$ can retain more edges and a smaller $\sigma_t$ can smooth out more textures. Empirically, to ensure good performance, $\sigma_s$ falls into $[0.1, 0.3]$ and $\sigma_t$ into $[0.2, 0.4]$. Fig. 4 shows results with various $\sigma_s$ and $\sigma_t$.

### B. Comparison with other methods

*1) Visual comparison:* Visual comparison is widely used in almost all the image smoothing papers. In Fig. 5, we compare our filter with 2 classical algorithms (total variation (TV) [23], bilateral filter (BLF) [4]), and 6 state-of-the-art algorithms (relative total variation (RTV) [24], guided filter (GF) [5], rolling guidance filter (RGF) [13], fast L0 smoothing [27], segment graph filter (SGF) [6], static and dynamic guidance filter (SDF) [14]). Among them, BLF, GF, RGF, SGF are kernel based, while TV, RTV, fast L0 smoothing, SDF are optimization based. We use the default parameters defined in their open-source code. In our method, we set $k = 9$, $\sigma_s = 0.15$, $\sigma_t = 0.2$, and $N_{itr} = 3$. With a clearer visualization with close-ups, our method outperforms other kernel-based and optimization-based methods in that it is able to suppress texture more effectively without over-smoothing the main structure.

One special and difficult example is the third vase example, in which the vase body is covered with very strong texture while the object (vase and base) itself has relatively low contrast to its background. Ideally, the texture should be removed while object-background contrast should be retained. As can be observed, only our DGF removes all the black textures on the vase and preserves contrast between the object and the background simultaneously. Other methods cannot achieve both goals effectively. Even though in some cases, e.g., the results produced by TV, RTV and SGF, texture is eliminated somewhat, however, the base-background contrast is completely lost as a penalty. This example further shows how our method outperforms other methods in preserving the main structure and removing texture.

TABLE I
SNR values of images in Fig. 6

| Method | Fig. 6(1) | Fig. 6(2) | Fig. 6(3) | Average |
|---|---|---|---|---|
| Noisy input | 30.99 | 33.12 | 36.06 | 33.39 |
| BLF [4] | 45.73 | 44.54 | 48.28 | 46.18 |
| GF [5] | 44.77 | 42.95 | 47.59 | 45.10 |
| RGF [13] | 54.45 | 49.59 | 56.20 | 53.41 |
| L0 [7] | 35.82 | 45.37 | 48.14 | 43.11 |
| Fast L0 [27] | 47.37 | 47.10 | 51.73 | 48.73 |
| SGF [6] | 50.63 | 42.25 | 50.45 | 47.78 |
| SDF [14] | 41.39 | 42.01 | 46.82 | 43.41 |
| Proposed | **58.36** | **63.25** | **62.69** | **61.43** |

*2) Quantitative evaluation:* Since denoising is a basic function of image smoothing, we can further evaluate the denoising performance with SNR quantitatively, similar to [6, 10, 13,

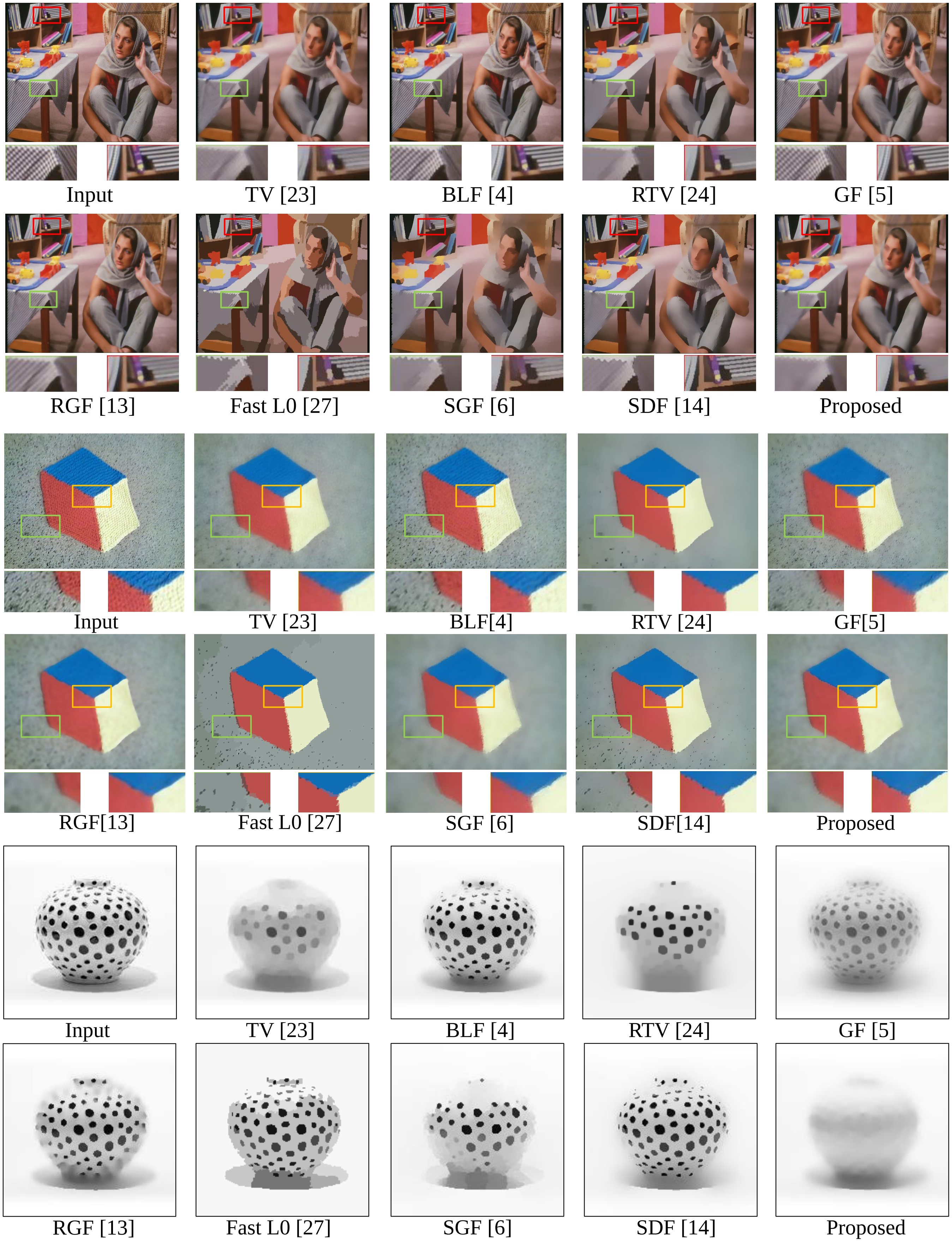



Fig. 5. Comparison of image smoothing results with different methods.

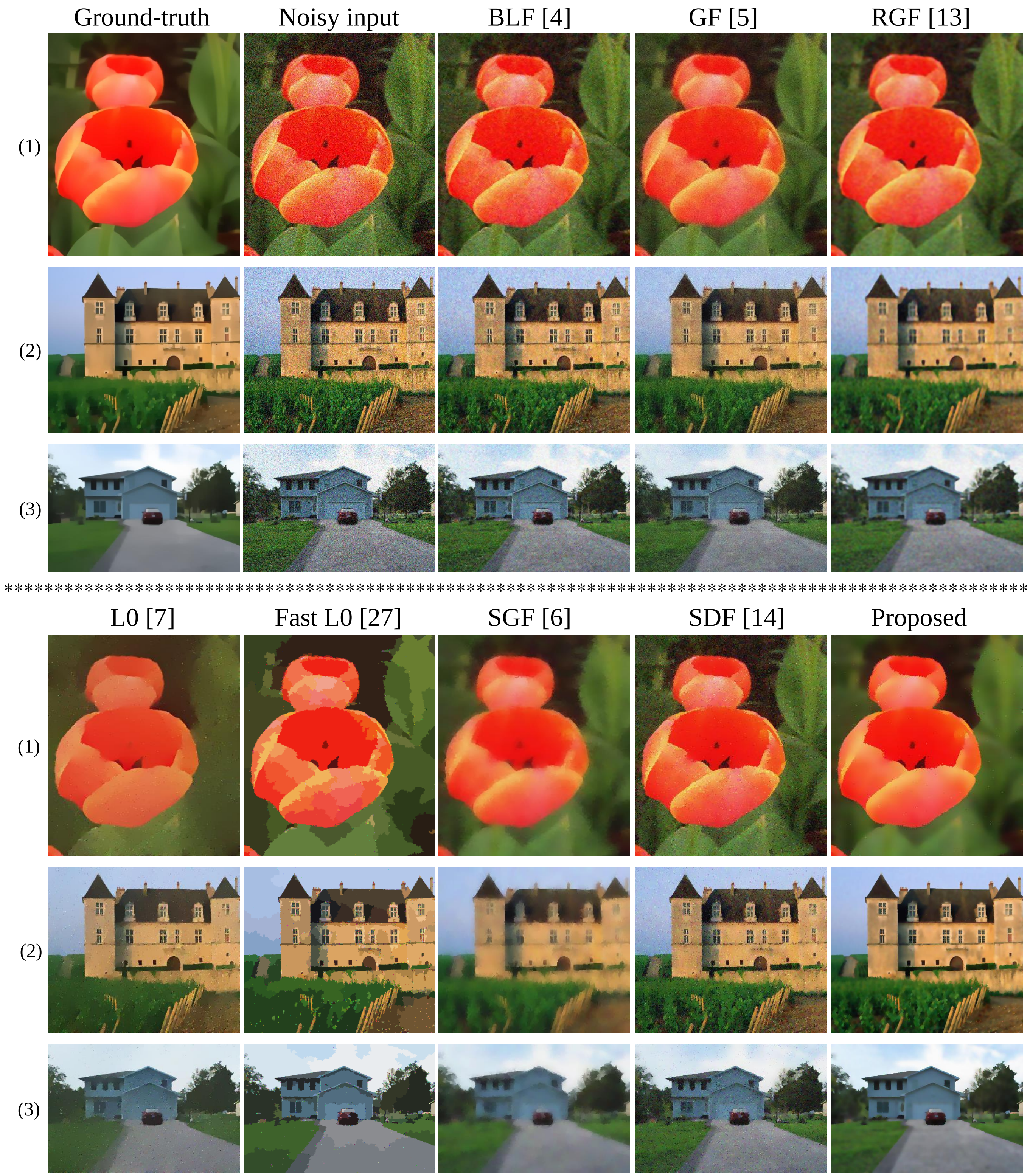
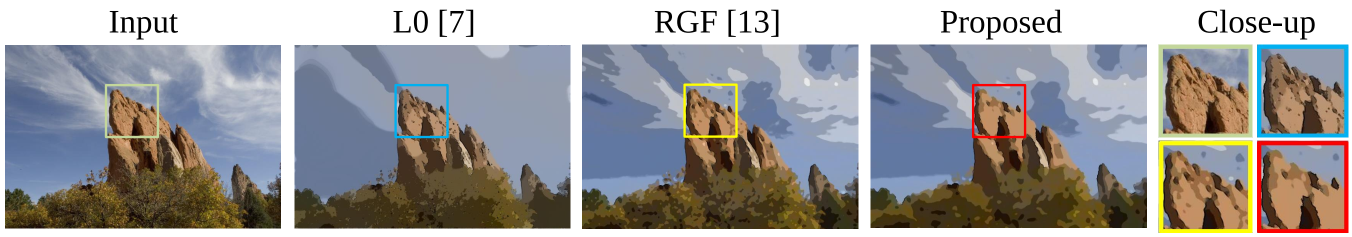



Fig. 6. Image denoising results with different methods.

27, 32]. More specifically, we first take a smoothed image as ground-truth (original signal), and then add Gaussian noise and texture manually. The SNR here measures the effect of removing noise (compared with ground-truth, a larger SNR indicates that the noise are removed more effectively). We show three groups of results in Fig. 6, and list corresponding SNR values in Table I. It is clear that the SNR values of our filter are largest in all the three examples, showing that




(a) Image abstraction

(b) JPEG artifacts removal

(c) Detail enhancement

Fig. 7. Image smoothing applications with different methods.

our method can suppress noises more effectively. Moreover, the results with our method are very close to the ground-truth from visual comparison, indicating that it can suppress noise and preserve the main structure and color distribution at the same time. It should be noted that although RGF and fast L0 can both yield relatively large SNR, RGF makes the smoothed image look blurry, especially at the significant edges or corners, while fast L0 has unexpectedly introduced more noticeable quantization into the results, which makes the output look less smoothed and less similar to the ground-truth.

### *C. Applications*

*1) Image abstraction:* Image abstraction aims to create a cartoon-like style from an input image. We use the method in [1] for image abstraction, which involves smoothing the input and retaining main structures, detecting difference-of-Gaussian edges, and abstracting the image with soft color quantization. The results of image abstraction for a mountain are shown in Fig. 7(a). It is clear that our method can suppress more details while still preserving the main structure, especially on the surface of the mountain. Note that L0-smoothing outperforms two other kernel based methods in abstracting the **sky** because it performs a global smoothing to the whole image by manipulating the number of non-zero gradients and the gradients within the sky region are similar.

*2) JPEG artifacts removal:* JPEG compression images have artifacts, that degrade quality. Artefact removal results are shown in Fig. 7(b). We can observe that our method removes artifacts more effectively, while retaining better similarity to the input compared with other two methods.

*3) Detail enhancement:* Suppose $I$ is the input image, and $S$ is the smoothed output. We define detail enhancement $DE$ as: $DE = S + \alpha \cdot (I - S)$, where $\alpha \geq 1$ controls the extent ($\alpha = 2$ in this case). The results with different methods are shown in Fig. 7(c). As shown, with close inspection of some texture regions, our method performs better in boosting the details without affecting the overall color tone.

## VI. CONCLUSION

The proposed double-guided filter outperforms existing image smoothing methods in preserving the main structure and removing insignificant texture. In this paper, for the first time, we introduce the concept of texture guidance which fundamentally improves traditional kernel-based methods in terms of distinguishing texture from structure. The combination of structure guidance and texture guidance makes the filter both 'structure-aware' and 'texture-aware'. Our method performs consistently well in both image smoothing and denoising tasks, and a number of experiments have demonstrated the effectiveness of the proposed filter. Our future work will focus on implementing new methods for constructing structure and texture guidance, and accelerating the filtering process with GPU parallel computing.

## REFERENCES


[1] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," in *ACM Transactions On Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 1221–1226.

[2] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Trans. Graph.*, vol. 26, no. 3, p. 51, 2007.

[3] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.

[4] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 839–846.

[5] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[6] F. Zhang, L. Dai, S. Xiang, and X. Zhang, "Segment graph based image filtering: Fast structure-preserving smoothing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 361–369.

[7] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l 0 gradient minimization," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 174.

[8] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 67.

[9] S. Hallman and C. C. Fowlkes, "Oriented edge forests for boundary detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1732–1740.

[10] Q. Liu, J. Liu, P. Dong, and D. Liang, "Sgtd: Structure gradient and texture decorrelating regularization for image decomposition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1081–1088.

[11] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 664–672, 2004.

[12] H. Cho, H. Lee, H. Kang, and S. Lee, "Bilateral texture filtering," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 128, 2014.

[13] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European Conference on Computer Vision*. Springer, 2014, pp. 815–830.

[14] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4823–4831.

[15] ——, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[16] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 176, 2013.

[17] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang, "Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 555–569, 2014.

[18] J.-H. Rick Chang and Y.-C. Frank Wang, "Propagated image filtering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 10–18.

[19] R. J. Jevnisek and S. Avidan, "Co-occurrence filter," *arXiv preprint arXiv:1703.04111*, 2017.

[20] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do, "Cross-based local multipoint filtering," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 430–437.

[21] X. Tan, C. Sun, and T. D. Pham, "Multipoint filtering with local polynomial approximation and range guidance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2941–2948.

[22] L. Dai, M. Yuan, F. Zhang, and X. Zhang, "Fully connected guided image filtering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 352–360.

[23] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[24] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 139, 2012.

[25] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese, "Fast cartoon+ texture image filters," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 1978–1986, 2010.

[26] S. Bi, X. Han, and Y. Yu, "An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 78, 2015.

[27] R. M. Nguyen and M. S. Brown, "Fast and effective l0 gradient minimization by region fusion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 208–216.

[28] C.-T. Shen, F.-J. Chang, Y.-P. Hung, and S.-C. Pei, "Edge-preserving image decomposition using l1 fidelity with l0 gradient," in *SIGGRAPH Asia 2012 Technical Briefs*. ACM, 2012, p. 6.

[29] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 147, 2009.

[30] Y. Zang, H. Huang, and L. Zhang, "Guided adaptive image smoothing via directional anisotropic structure measurement," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 9, pp. 1015–1027, 2015.

[31] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1558–1570, 2015.

[32] Q. Yang, "Semantic filtering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4517–4526.