Patch Shift for Structure-preserving Image Decomposition*

Hojin Cho[1] Hyunjoon Lee[1] Henry Kang[2] Seungyong Lee[1]

[1]POSTECH, 77 Cheongam-Ro, Nam-Gu, Pohang, KOREA

[2]University of Missouri at St. Louis, 1 University Blvd. St. Louis, USA

Abstract

This paper presents a novel structure-preserving image decomposition operator called *bilateral texture filter*. As a simple modification of the original bilateral filter [2], it performs local patch-based analysis of texture features and incorporates its results into the range filter kernel. The central idea to ensure proper texture/structure separation is based on *patch shift* that captures the texture information from the most representative texture patch clear of prominent structure edges. Our method outperforms the original bilateral filter in removing texture while preserving main image structures, at the cost of some added computation. It inherits well-known advantages of the bilateral filter, such as simplicity, local nature, ease of implementation, scalability, and adaptability to other application scenarios.

Keywords: bilateral filter, patch shift, texture smoothing, image decomposition

*This is a short version of our published paper [1]. The full version can be found in [1].

1 Introduction

Structure-preserving filtering is an essential operation with a variety of applications in computational photography and image analysis. Such an operation decomposes an image into prominent structure and fine-scale detail, making it easier for subsequent image manipulation such as tone mapping, detail enhancement, visual abstraction, scene understanding, and other tasks. Separating structure from detail often depends on measuring the size of local contrast, where structure is identified as pixels having relatively large contrast. However, when the fine-scale detail represents *texture*, the conventional way of image decomposition may fail because texture often contains strong enough contrast to get confused with structure.

Many of the structure-preserving smoothing operators are based on local filtering [3, 2, 4, 5]. While these nonlinear filters are simple and intuitive to use, they are often ill-equipped to extract structure from texture due to having no explicit measure with which to distinguish the two. On the other hand, there are optimization-based [6, 7, 8, 9, 10, 11, 12] and patch-based [13] solutions as well, some of which have been specifically designed to handle texture and thus outperform local filtering in terms of texture removal. However, they usually come with additional level of complexity and sophistication, which makes them harder to implement, accelerate, scale, or adapt.

In this paper, we present a novel method for nonlinear image decomposition based on a simple modification to bilateral filter [2]. It is in essence a joint bilateral filter [14, 15] that incorporates texture information (instead of color information) into the range filter kernel. We demonstrate that our method effectively removes texture while preserving structure, which the standard bilateral filter often fails to do. Being a simple extension to the popular bilateral filter, our method enjoys the benefits that come with it, such as simplicity, speed, ease of implementation, scalability, and adaptability. To distinguish the proposed method from its original formulation, we call it *bilateral texture filter*.

Our key idea to extract local texture without obscuring structure is *patch shift*, which for each pixel captures the texture information from the patch in the neighborhood that excludes prominent structure edges nearby and best represents the texture region containing the pixel. Patch shift in effect performs structure-preserving soft image segmentation of texture regions. The result of this operation is then used as the guidance image in our joint bilateral filtering. Therefore, the only additional step required over the standard bilateral filter is the computation of guidance image via patch shift, which can be achieved at a small computational cost.

2 Related Work

Bilateral filter [2] is one of the most widely used nonlinear operators for discontinuity-preserving image smoothing and decomposition. Its simplicity, effectiveness, and extendability led to its broader usage in other applications as well, such as tone mapping [16], detail enhancement [17, 4], image editing [18, 19], image upsampling [20], mesh denoising [21, 22], and artistic rendering [23, 24].

Subsequent development of more sophisticated edge-preserving filters, including weighted least squares (WLS) [8], edge-avoiding wavelets [25], local histogram filtering [26], local Laplacian filtering [5], domain transform [27], and L_0 gradient minimization [11], all basically share the same goal of smoothing fine-scale details without degrading image structures, although they are not explicitly designed to deal with texture. Subr et al. [9], on the other hand, defined detail as oscillations between local extrema in order to distinguish small-scale yet high-contrast features, i.e., texture, from real edges.

Regular or near-regular textures may be identified and filtered by exploiting spatial relationship, frequency, and symmetry of texture features [28, 29]. Total variation (TV) [30] on the other hand has proven to work well on filtering arbitrary texture of irregular shapes by enforcing TV regularization constraints to preserve large-scale edges. The original formulation of TV regularization was further extended to achieve better quality, robustness, and efficiency [6, 7, 10, 12]. In particular, Xu et al. [12] introduced the notion of relative total variation (RTV), a spatially-varying total variation measure that helps improve the quality of texture-structure separation.

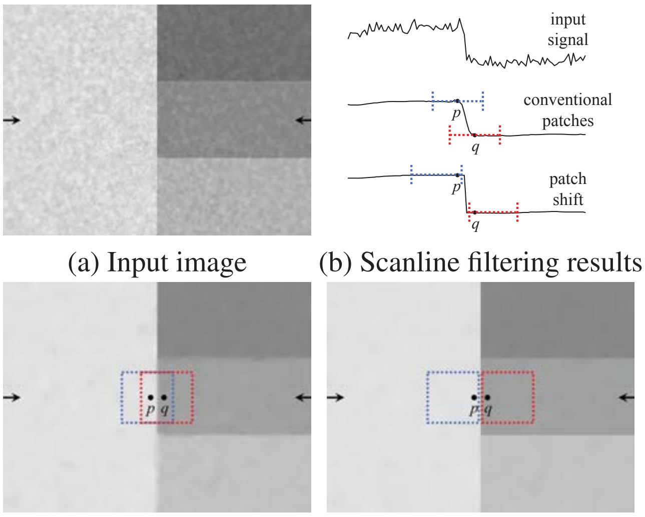
Recently, Karacan et al. [13] proposed a patch-based texture removal algorithm that uses the similarity measures based on a region covariance descriptor. Compared to the conventional pixel-based image decomposition methods, the use of covariance matrix associated with each patch in the neighborhood enables a more accurate description and identification of texture feature, leading to better performance in separating texture from structure. On the other hand, a patch-based approach is also prone to overblur the structure edges since the overlapping patches near an edge inevitably share similar statistics.

3 Patch Shift

Given a scalar-valued input image I, the bilateral filter [2] computes an output image J by

$$J_p \quad = \quad \frac{1}{k_p} \sum_{q \in \Omega_p} f(||q-p||) g(||I_q - I_p||) I_q, \tag{1}$$

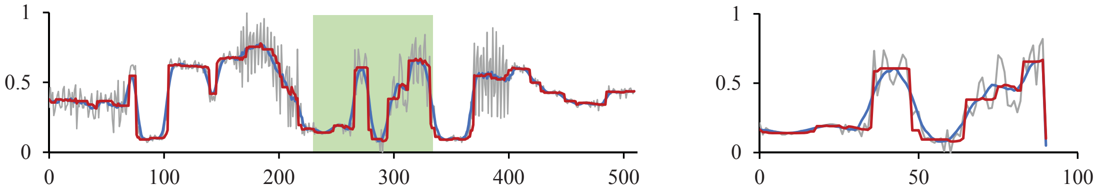
where k_p is a normalizing term. The output J_p at pixel p is a weighted average of I_q in the spatial neighborhood Ω_p. The spatial kernel f and the range kernel g are typically Gaussian functions. The data-dependent weight g is inversely proportional to the size of contrast between two pixels p and q. This



(a) Input image (b) Scanline filtering results

(c) Conventional patches (d) Patch shift

Figure 1: Patch shift. Conventionally, texture feature is computed in a patch centered at each pixel, in which case the patches for two adjacent pixels should have a large overlap, reducing the feature discriminability. In contrast, patch shift finds a nearby patch that stays clear of a prominent structure edge. (b) Filtering of the scanline marked by arrows. (c) Filtered by [13]. (d) Filtered with patch shift. The results in (b) show that our approach preserves structure edges, unlike the conventional approach.



Input I (gray), local average B (blue), and texture value G (red) Close-up of the green region

Figure 2: 1D example of guidance image computation using patch shift. The input signal is a scanline of the Barbara image.

nonlinear weighting enables bilateral filter to blur small-scale intensity variations while preserving salient edges.

We extend the bilateral filter by substituting a texture description image G in the range kernel g:

$$J_p = \frac{1}{k_p} \sum_{q \in \Omega_p} f(||q-p||)g(||G_q - G_p||)I_q, \tag{2}$$

This is a texture-filtering variant of Eq. (1), and its success depends heavily on the design of G, which is also called *guidance image* in the context of joint bilateral filtering [14, 15].

The value of G_p can be defined by analyzing local image statistics [31, 32] in a rectangular *patch* Ω_p centered at p. When a patch contains both texture and structure, however, such local statistics may obscure the existence of salient edges or region boundaries. For example, two neighboring patches (each with size $k \times k$) centered at two adjacent pixels $p = (i, j)$ and $q = (i, j+1)$, respectively, must have a large overlap of size $k \times (k-1)$. Consequently, these local statistics should be similar even when p and q happen to be on the opposite sides of an edge. While Karacan *et al.* [13] alleviated this problem by including pixel position in computing the region covariance, it is often not enough to guard against edge blurring (Fig. 1c).

We overcome this limitation by introducing a novel method called *patch shift*. Assuming a $k \times k$ box representing a patch, each pixel p has a total of k^2 patches in I that contains p. Among these k^2 patches, we find the patch Ω_q that is least likely to contain a prominent structure edge. Once we have found Ω_q that has this property, we use the average intensity within this patch, denoted B_q, as the representative texture value G_p at p in Eq. (2). In a nutshell, patch shift finds the texture patch in the neighborhood that most likely stays clear of the structure edge (if present) and best represents the texture region that the pixel belongs to (Fig. 1d).

There are several possible choices for defining a texture measure in such a way that it ensures separation from structure edges. In this paper, we define texture as *fine-scale spatial oscillations* of signals, as in [9, 12]. Let us assume for the time being that texture signal has smaller amplitude than the neighboring structure edge (this requirement will be lifted in Section 4), then we can simply measure the likelihood of containing structure edge for a patch Ω_q via its *tonal range* $\Delta(\Omega_q)$:

$$\Delta(\Omega_q) \quad = \quad I_{max}(\Omega_q) - I_{min}(\Omega_q). \tag{3}$$

where $I_{max}(\Omega_q)$ and $I_{min}(\Omega_q)$ denote the maximum and the minimum image intensities in Ω_q, respectively. We then let patch shift select the patch with the minimum tonal range, which is to minimize the probability of involving a salient edge when computing texture feature.

Fig. 2 illustrates how patch shift works on a 1D signal, when tonal range is used as the texture measure. A patch is in this case defined as an interval of width k. For every p in the input signal I, we precompute the average intensity B_p within its own center patch Ω_p. Then the texture signal G_p at p is obtained by copying B_q at q that has the smallest $\Delta(\Omega_q)$ in the neighborhood of p. Note that the patch shift process successfully flattens the (small) oscillations in texture regions without degrading the structure edges. In case p is part of a thin texture region, there may be more than one structure edges in the neighborhood, in which case patch shift still tries its best to stay away from the biggest edge.

4 Algorithm

Our 2D filtering process is an extension of the 1D process described above. Given an input image I, we first apply $k \times k$ box kernel to compute the average image B. For each pixel p, we also compute the tonal range $\Delta(\Omega_p)$ in Eq. (3). We then obtain the guidance image G via patch shift on each pixel. That is, we

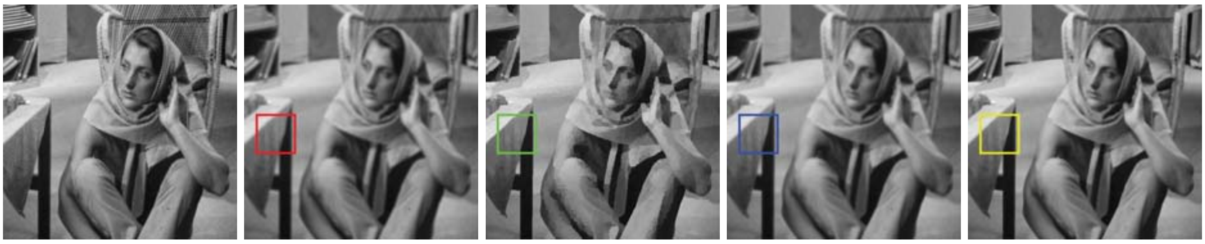

(a) Input I (b) Blurred B (c) Guidance G (d) Guidance G' (e) Result J

Figure 3: Overall process and intermediate images of our bilateral texture filtering.

find the patch Ω_q whose $\Delta(\Omega_q)$ is the minimum among k^2 candidates, then copy B_q to G_p. Finally we obtain the output image J by applying joint bilateral filter on I, using G as the guidance image. While this process generally performs well in terms of texture-structure decomposition, we make two modifications to improve the robustness of our scheme.

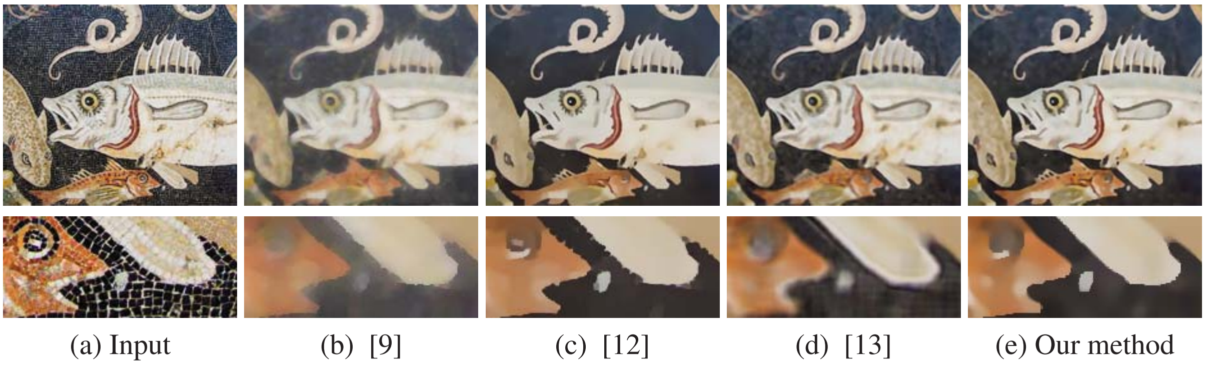
Eq. (3) suggests that patch shift may not work properly if the tonal range within a pure texture region is as large as (or larger than) the nearby structure edge. We resolve this by adapting Relative Total Variation (RTV) [12]. We define *modified Relative Total Variation (mRTV)* as

$$\text{mRTV}(\Omega_q) \quad = \quad \Delta(\Omega_q)\,\frac{\max\limits_{r\in\Omega_q}|(\partial I)_r|}{\sum\limits_{r\in\Omega_q}|(\partial I)_r|+\epsilon}, \tag{4}$$

where $|(\partial I)_r|$ denotes the gradient magnitude at pixel $r \in \Omega_q$ and ϵ is a small value to avoid division by zero. In our implementation, we use $|(\partial I)_r| = \sqrt{(\partial_x I)_r^2 + (\partial_y I)_r^2}$ and $\epsilon = 10^{-9}$. The tonal range $\Delta(\Omega_q)$ serves as a scale factor in Eq. (4) to reflect the absolute magnitude of the signal.

The mRTV value is relatively large in a structure patch containing only a few edges, and relatively small in a texture patch having frequent oscillations. For a $k \times k$ 2D patch, $\frac{\text{mRTV}(\Omega_q)}{\Delta(\Omega_q)}$ would be approximately $\frac{1}{k}$ for a horizontal or vertical step edge, and $\frac{1}{k^2}$ for a texture patch with full oscillations. Note that this is true even when the texture amplitudes are as large as (or larger than) the edges nearby. Therefore, the use of mRTV enables filtering of texture with arbitrarily large magnitudes. Our modified patch shift operation should now locate a pure texture patch among k^2 candidates by finding the patch Ω_q with the minimum mRTV value. Fig. 3c shows the guidance image G obtained via mRTV-driven patch shift, which effectively restores structure edges from the blurred image B.

The mRTV values in a smooth or flat image region tend to be very small and thus may become sensitive to image noise. For example, in a smooth region where intensity changes gradually, small noisy peaks can be misinterpreted as edges, resulting in a wrong B_q value being copied to G_p and thus disrupting the gradual intensity variation. To prevent this, we examine the mRTV values of Ω_p and Ω_q when copying B_q to G_p. If the two mRTV values are similar (meaning similar local statistics), B_p is preferred over B_q as the



(a) Input (b) [9] (c) [12] (d) [13] (e) Our method

Figure 4: Comparison with previous methods on “Pompeii Fish Mosaic”. Previous methods, (b)-(d), went through careful manual parameter tuning. Parameters: [9] ($k = 13$), [12] ($\lambda = 0.015, \sigma = 6$), [13] ($k = 19, \sigma = 0.2$, Model 1), and our method ($k = 7, n_{itr} = 5$). Input image courtesy Chris Beckett.

value of G_p. If and only if $\text{mRTV}(\Omega_q)$ is considerably smaller than $\text{mRTV}(\Omega_p)$ (meaning Ω_q is obviously more flat or homogeneous), B_q is used for G_p.

This strategy can be implemented by interpolating images B and G using the difference in mRTV values as blending weight. That is,

$$G'_p = \alpha_p G_p + (1 - \alpha_p) B_p, \tag{5}$$

where

$$\alpha_p = 2 \left(\frac{1}{1 + \exp(-\sigma_\alpha(\text{mRTV}(\Omega_p) - \text{mRTV}(\Omega_q)))} - 0.5 \right). \tag{6}$$

The weight $\alpha_p \in [0, 1]$ is small inside smooth/texture regions, and large around edges. In Eq. (6), σ_α controls the sharpness of the weight transition from edges to smooth/texture regions, where a bigger σ_α means sharper transition. We use $\sigma_\alpha = 5k$ in our experiments. The interpolated image G' is the modified guidance image (Fig. 3d) that we finally use in our joint bilateral filtering of Eq. (2).

For image denoising, a single iteration of bilateral filtering is often sufficient. However, texture may have spatial and/or range scales that are much bigger than that of noise. Therefore, depending on the input, more than one (usually $3 \sim 5$) iterations of bilateral texture filtering might be necessary to obtain a desired effect.

5 Results

Comparison with state-of-the-art In Fig. 4, we compare our method with the state-of-the-art nonlinear image smoothing techniques that were specifically designed to perform texture removal [9, 12, 13]. In generating results for these techniques, we used the implementations provided online by the authors and

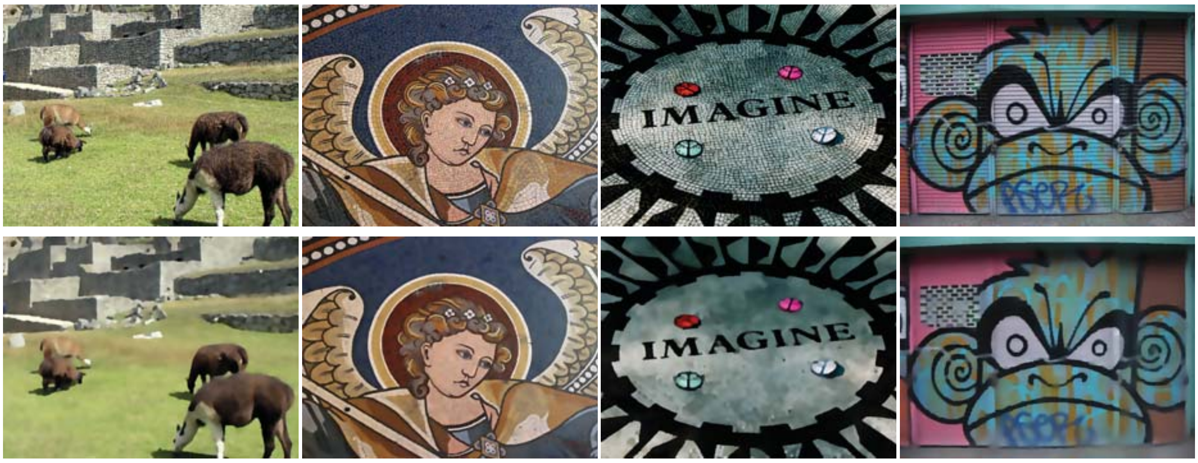


Figure 5: More results of bilateral texture filtering. (top) Input images. (bottom) Our filtering results. (left to right) Input image courtesy flickr users YoTuT, Lawrence Rice, Alexander Kauschanski, and bixentro.

fine tuned the parameters manually. All the methods we tested generally succeeded in extracting prominent image structure while filtering out texture. As pointed out in [12, 13], however, we noticed that the method of Subr et al. [9] often degrades image structures and exhibits blur artifacts, due to the difficulty of locating extrema in regions containing the mixture of texture and structure (Fig. 4b). The method of Xu et al. [12] shows a robust performance in both texture removal and structure preservation/enhancement. As a byproduct of global optimization, however, oversmoothing of details may occur, which obscures the surface shading and makes the resulting image look somewhat flat. It also appears to be difficult with this method to eliminate texture located near a structure edge, possibly due to their strong edge preservation property (Fig. 4c). The covariance-based method proposed by Karacan *et al.* [13] removes texture effectively while preserving edges and surface shading, but may oversmooth structure due to the inherent limitation of covariance descriptor in locating edges (Fig. 4d). On the other hand, our method consistently preserves both structure and shading information without leaving unprocessed texture (Fig. 4e). The supplementary material contains more comparisons using other input images. Fig. 5 shows additional results of our bilateral texture filtering.

6 Discussion

Our bilateral texture filter retains the simplicity of the original bilateral filter, yet provides significantly enhanced performance in separating texture details from image structures. We expect this simplicity, efficiency, and effectiveness to open up interesting application possibilities. The proposed patch shift mechanism plays a key role in our method as it finds appropriate texture/smooth patch for each pixel that is needed to generate a guidance image. Patch shift is a general concept and does not depend on any specific definition of texture feature. Therefore, its usefulness and applicability could be further explored in a larger context of research on image processing.

References

[1] H. Cho, H. Lee, H. Kang, and S. Lee, "Bilateral texture filtering," *ACM Transactions on Graphics*, vol. 33, pp. 128:1–128:8, July 2014.

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV 1998*, pp. 839–846, 1998.

[3] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[4] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 51:1–51:9, 2007.

[5] S. Paris, S. W. Hasinoff, and J. Kautz, "Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid," *ACM Trans. Graphics*, vol. 30, no. 4, pp. 68:1–68:12, 2011.

[6] W. Yin, D. Goldfarb, and S. Osher, "Image cartoon-texture decomposition and feature selection using the total variation regularized L1 functional," *Variational, Geometric, and Level Set Methods in Computer Vision*, vol. 3752, pp. 73–84, 2005.

[7] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition–modeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.

[8] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 67:1–67:10, 2008.

[9] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Trans. Graphics*, vol. 28, no. 5, pp. 147:1–147:9, 2009.

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

[11] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L0 gradient minimization," *ACM Trans. Graphics*, vol. 30, no. 5, pp. 174:1–174:12, 2011.

[12] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graphics*, vol. 31, no. 6, pp. 139:1–139:10, 2012.

[13] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graphics*, vol. 32, no. 6, pp. 176:1–176:11, 2013.

[14] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graphics*, vol. 23, pp. 664–672, 2004.

[15] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Trans. Graphics*, vol. 23, pp. 673–678, 2004.

[16] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 257–266, 2002.

[17] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 637–645, 2006.

[18] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, “Image-based modeling and photo editing,” in *Proc. ACM SIGGRAPH 2001*, (New York, NY, USA), pp. 433–442, ACM Press, 2001.

[19] J. Chen, S. Paris, and F. Durand, “Real-time edge-aware image processing with the bilateral grid,” *ACM Trans. Graphics*, vol. 26, no. 3, pp. 103:1–103:9, 2007.

[20] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” *ACM Trans. Graphics*, vol. 26, no. 3, p. 96, 2007.

[21] T. R. Jones, F. Durand, and M. Desbrun, “Non-iterative, feature-preserving mesh smoothing,” *ACM Trans. Graphics*, vol. 22, no. 3, pp. 943–949, 2003.

[22] S. Fleishman, I. Drori, and D. Cohen-Or, “Bilateral mesh denoising,” *ACM Trans. Graphics*, vol. 22, no. 3, pp. 950–953, 2003.

[23] H. Winnemöller, S. C. Olsen, and B. Gooch, “Real-time video abstraction,” *ACM Trans. Graphics*, vol. 25, no. 3, pp. 1221–1226, 2006.

[24] H. Kang, S. Lee, and C. Chui, “Flow-based image abstraction,” *IEEE Trans. Visualization and Computer Graphics*, vol. 15, pp. 62–76, 2009.

[25] R. Fattal, “Edge-avoiding wavelets and their applications,” *ACM Trans. Graphics*, vol. 28, no. 3, pp. 22:1–22:10, 2009.

[26] M. Kass and J. Solomon, “Smoothed local histogram filters,” *ACM Trans. Graphics*, vol. 29, no. 4, pp. 100:1–100:10, 2010.

[27] E. S. L. Gastal and M. M. Oliveira, “Domain transform for edge-aware image and video processing,” *ACM Trans. Graphics*, vol. 30, no. 4, pp. 69:1–69:12, 2011.

[28] Y. Liu, W.-C. Lin, and J. Hays, “Near-regular texture analysis and manipulation,” *ACM Trans. Graphics*, vol. 23, no. 3, pp. 368–376, 2004.

[29] J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, “Discovering texture regularity as a higher-order correspondence problem,” in *Proc. ECCV 2006*, pp. 522–535, 2006.

[30] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D*, vol. 60, pp. 259–268, 1992.

[31] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[32] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” in *Proc. ECCV 2006*, pp. 589–600, 2006.