Nonlocal Image Editing

Hossein Talebi, *Student Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*— In this paper, we introduce a new image editing tool based on the spectrum of a global filter computed from image affinities. Recently, it has been shown that the global filter derived from a fully connected graph representing the image can be approximated using the Nyström extension. This filter is computed by approximating the leading eigenvectors of the filter. These orthonormal eigenfunctions are highly expressive of the coarse and fine details in the underlying image, where each eigenvector can be interpreted as one scale of a data-dependent multiscale image decomposition. In this filtering scheme, each eigenvalue can boost or suppress the corresponding signal component in each scale. Our analysis shows that the mapping of the eigenvalues by an appropriate polynomial function endows the filter with a number of important capabilities, such as edge-aware sharpening, denoising, tone manipulation, and abstraction, to name a few. Furthermore, the edits can be easily propagated across the image.**

***Index Terms*— Image editing, non-local filters, Nyström extension.**

I. Introduction

EDGE-AWARE image filtering is a key tool in image processing, computer vision and graphics. In most existing methods the underlying image is first decomposed into piecewise smooth and detail layers. Then, a variety of capabilities, such as tone mapping, edge editing and edit propagation are developed based on this type of decomposition [2]–[6].

The optimal edge-aware filter coarsens details of the image, yet the principal edges are ideally not altered. Given the difficulty in determining what should qualify as an edge, and which edge should be preserved or smoothed, designing such a filter is quite challenging. Many smoothing operators have been proposed in the past few years, and anisotropic diffusion is perhaps one of the most well-known methods [7]. Anisotropic diffusion tends to preserve (and even sharpen) main edges while smoothing the texture regions. However, the iterative nature of this filter can make the computational burden quite heavy. Also, the ideal iteration number for each region of the image could be different; yet, an infinite iteration number can produce a constant image.

Several non-linear (data-dependent) operators such as the bilateral filter [8], [9] have been used for the same task. Chen et al. [10] used the bilateral filter, while progressively incrementing the spatial and range width of the Gaussian for building a pyramid of image layers. In a similar iterative approach [11], the bilateral filter is applied successively on the coarsened image while decreasing the range width. In all of these methods edges are preserved by the gradual change in the tuning parameters of the bilateral kernel. However, the kernel weights have to be recomputed in every iteration.

Almost all existing edge-aware methods use the same general idea: *Using a local operator, they decompose the image into base layer and detail layer and then manipulate each layer and recombine separately to reach the desired edit*. There are two main problems with this approach:

- Since noise is always an unavoidable part of our imaging systems, boosting the detail layer usually worsens the signal-to-noise-ratio (SNR). Even with today's mega-pixel images, the trade off between sharpness and SNR is still a bottleneck. Increasing exposure time will result in higher SNR, but a more blurry image. On the other hand, a short exposure leads to sharper but noisier images.
- While it is often desirable to treat similar edges of an image in the same way, the existing local filters have irregular behaviors when handling edges with slightly different brightness and gradient profiles. In other words, global structure common among similar edges are usually ignored by the low-level feature vectors associated with each pixel. Even with all the edge-aware operators in hand, performing local adjustments to pixels and then *evenly* propagating the edit to the similar regions *all* across the image has proved to be challenging.

To alleviate the first problem, some methods have been proposed that build on classic linear unsharp masking. Adaptive unsharp masking [12] controls contrast enhancement in texture areas and avoids noise magnification by leaving relatively smooth regions unchanged. A hierarchical framework based on Non-Local Means (NLM) kernel [13] is proposed in [14] where noise removal is applied first as a separate step and then the detail layers are extracted. Another technique in [15] applies an offset to the bilateral filter to change its normal coarsening behavior to sharpening. This adaptive bilateral filter sharpens an image by increasing the slope of edges; however, its sharpening strength is limited. Recently a new restoration method for handling mild blur and strong noise has been introduced in [16], where using the steering kernel regression technique [17] both denoising and sharpening are combined in one framework.

To mitigate the second problem, there have been some efforts to interactively propagate the edits to regions with similar appearance. Recently, the sparse optimization formulation is used to provide stroke-based editing workflows

Manuscript received January 27, 2014; revised May 22, 2014; accepted July 28, 2014. Date of publication August 18, 2014; date of current version September 9, 2014. This work was supported in part by the Air Force Office of Scientific Research, Arlington, VA, USA, under Grant FA-9550-07-1-0365, and in part by the National Science Foundation under Grant CCF-1016018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rafael Molina.

The authors are with the Department of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: htalebi@soe.ucsc.edu; milanfar@soe.ucsc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2348870

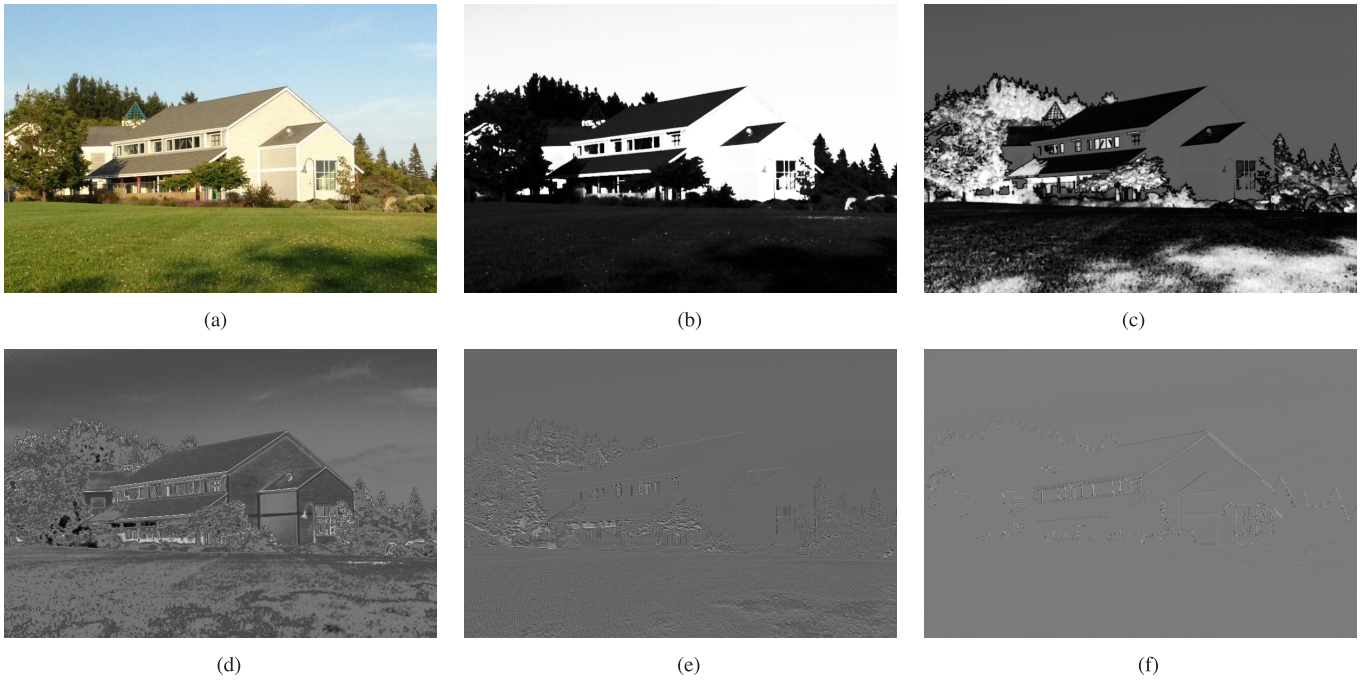


Fig. 1. Some leading eigenvectors computed from the luminance channel of the house image using 0.01% of the pixels. (a) House. (b) $\mathbf{v}_2(\lambda_2 = 0.9917)$. (c) $\mathbf{v}_3(\lambda_3 = 0.9045)$. (d) $\mathbf{v}_{10}(\lambda_{10} = 0.2406)$. (e) $\mathbf{v}_{25}(\lambda_{25} = 0.0508)$. (f) $\mathbf{v}_{50}(\lambda_{50} = 0.0025)$.

with propagative tonal adjustments [6], [18], [19]. Using an edge-aware energy minimization method, the tonal adjustment imposed by the user is interpolated to the pixels with similar luminance. Farbman et al. [5] also proposed an edit propagation method based on the concept of *diffusion distances* which can measure closeness of pixels on a manifold space. By approximating a diffusion map built upon this high-dimensional similarity measure, the input adjustments can propagate to nearby pixels on the manifold.

In our framework, the two above-mentioned shortcomings of the existing methods are tackled at the same time. Our image filter is global in the sense that all the node (pixel) pairs on the graph (image) are directly connected to each other. As shown in [1], the eigen-decomposition of the corresponding symmetric, doubly-stochastic filter matrix can be approximated using the Nyström extension. The obtained eigenvectors are very informative of the similar regions and edge information of the image[1] (Fig. 1). More specifically, the approximated eigenfunctions enable us to employ diffusion distance for propagating the same manipulation over pixels belonging to similar regions globally. Having the spectrum of the filter, simultaneous noise suppression and detail enhancement become much easier by mapping the spectrum of the filter using a polynomial function, with a few parameters (sliders) to tune. Our experimental results show that this strategy reduces the halo artifacts around principal edges, avoids the common noise magnification problem, and can interactively propagate the user's edit across the intended similar regions with ease. In contrast to [5], our approach does not require the solution of a complex optimization problem to achieve this effect.

Our contributions are as follows: *First*, our framework handles noise naturally, because the image is projected into the data-adapted basis obtained from affinity weights. In other words, the noise is naturally separated from the underlying signal components by projecting the image into the approximated leading eigenvectors. *Second*, our global framework is better than the existing propagation approaches where the global affinity approximation is used just as a guide mask. In addition to providing such a mask, the proposed scheme is able to deliver a filtering tool with many capabilities.

In what follows, a description of global filter and its eigen-decomposition are given in Section II. Then, our detail manipulation strategy is explained in Section III. Next, our propagation mask is described in Section IV. After discussing our experimental results for different applications of the work, we conclude the paper in Section VI.

II. The Global Filter

Our filtering framework is based on non-parametric regression in which a kernel function K_{ij} measures the similarity between the samples y_i and y_j, located at $\mathbf{x}_i$ and $\mathbf{x}_j$ coordinates, respectively. The NLM kernel [13] is a very popular data-dependent filter in which the photometric similarity is captured in a patch-wise manner:

$$K_{ij} = \exp\left\{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{h_x^2} + \frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h_y^2}\right\} \tag{1}$$

where $\mathbf{y}_i$ and $\mathbf{y}_j$ are *patches* centered at $\mathbf{x}_i$ and $\mathbf{x}_j$ coordinates, and h_x and h_y denote the spatial and photometric smoothing

[1]Supplementary materials, including MATLAB code, additional experimental results and manuscript figures are available at the project website: http://www.soe.ucsc.edu/~htalebi/NLEditing.html

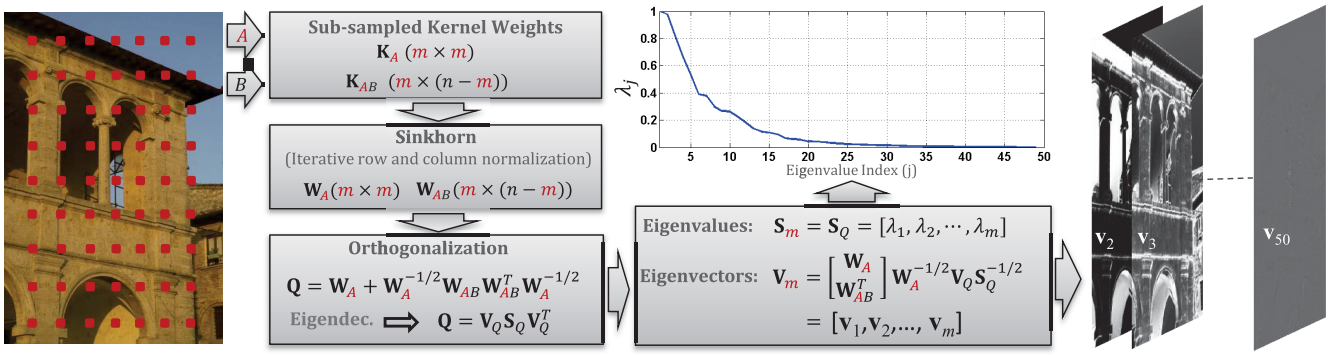


Fig. 2. Filter approximation using Nyström extension. Set A represents m samples from input image and set B contains the rest of pixels $(n-m)$. Matrix $\mathbf{K}_A$ represents the kernel weights of the sample set A and $\mathbf{K}_{AB}$ shows the kernel weights between set A and set B. Sinkhorn algorithm approximates the filter sub-matrices $\mathbf{W}_A$ and $\mathbf{W}_{AB}$ through an iterative normalization procedure. These sub-matrices can be used to approximate m leading orthonormal eigenvectors and eigenvalues of the filter matrix. In this example m is set as 50.

parameters, respectively.[2] These weights form the positive, symmetric kernel (affinity) matrix $\mathbf{K}$. The non-parametric framework yields a global filter description as follows:

$$\widehat{\mathbf{y}} = \mathbf{D}^{-1}\mathbf{K}\mathbf{y} = \mathbf{W}\mathbf{y}, \tag{2}$$

where $\mathbf{y}$ denotes the vectorized image of length n, and the matrix $\mathbf{D} = \text{diag}[d_1, d_2, \ldots, d_n]$ has the row summation of the kernel weights as $d_i = \sum_{j=1}^{n} K_{ij}$.

A. The Filter Matrix

The positive, row-stochastic filter matrix $\mathbf{W}$ is not generally symmetric, though it has real, positive eigenvalues [20]. More specifically, the eigenvalues of $\mathbf{W}$ lie in the interval [0, 1] with the largest one uniquely equal to one with corresponding eigenvector $\mathbf{v}_1 = \frac{1}{\sqrt{n}}[1, 1, \ldots, 1]^T$ ([21], [22]). This implies that the filter $\mathbf{W}$ preserves the average brightness of the input image $\mathbf{y}$.

Although $\mathbf{W}$ is not a symmetric matrix, it can be closely approximated with a symmetric, positive definite, doubly (i.e., row- *and* column-) stochastic matrix [23]. The symmetric $\mathbf{W}$ enables us to compute its eigen-decomposition as follows:

$$\mathbf{W} = \mathbf{V}\mathbf{S}\mathbf{V}^T, \tag{3}$$

in which the eigenvectors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ specify a complete orthonormal basis for $\mathbb{R}^n$ and $\mathbf{S} = \text{diag}[\lambda_1, \ldots, \lambda_n]$ contains the eigenvalues in decreasing order $0 \leq \lambda_n \leq \ldots < \lambda_1 = 1$.

B. Truncated Filter

In our global filtering framework, the filter matrix is not a full-rank local filter and thus can be closely approximated with a low-rank matrix. Namely, the filter matrix can be approximated with its m leading eigenvectors:

$$\mathbf{W}_m = \mathbf{V}_m\mathbf{S}_m\mathbf{V}_m^T, \tag{4}$$

in which $\mathbf{V}_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$ and $\mathbf{S}_m = \text{diag}[\lambda_1, \ldots, \lambda_m]$. Global features of the underlying image are represented in these leading eigenvectors. This has been shown in Fig. 1 where some eigenvectors with different indices are illustrated. Various features of the image are encoded by these basis functions. Eigenvectors with lower indices contain principal edges and corresponding eigenvectors of larger indices represent texture regions. As can be seen, these features are globally separated in each eigenmode.

[2]Typically, in the NLM, h_x is set to fairly large number; hence the non-locality.

C. Nyström Extension

A solution for reducing the computational burden of our global scheme is proposed in [1] where instead of computing each element of the filter $\mathbf{W}$, some sample rows (or columns) of the filter are exactly computed and used to approximate the remaining rows. As shown by Williams and Seeger [24], the Nyström method [25] gives a practical solution when working with huge affinity (similarity) matrices by operating on only a small portion of the complete matrix to produce a low-rank approximation. This approximation enables us to obtain the eigen-decomposition of the filter $\mathbf{W}_m$ given in (4).

Our filter approximation is illustrated in Fig. 2. First, the input image is uniformly sampled to form the sample set A with m pixels (typically $m \ll n$). The rest of $(n-m)$ pixels are assigned to set B. Then, the affinity matrices $\mathbf{K}_A$ (for pixels in set A) and $\mathbf{K}_{AB}$ (between pixels in sets A and B) are computed. These sub-matrices are m rows (or columns) of the kernel matrix as:

$$\mathbf{K} = \begin{bmatrix}\mathbf{K}_A & \mathbf{K}_{AB} \\ \mathbf{K}_{AB}^T & \mathbf{K}_B\end{bmatrix} \tag{5}$$

Then, using Sinkhorn algorithm ([26], [27]), the corresponding filter sub-matrices $\mathbf{W}_A$ and $\mathbf{W}_{AB}$ are approximated where:

$$\mathbf{W} = \begin{bmatrix}\mathbf{W}_A & \mathbf{W}_{AB} \\ \mathbf{W}_{AB}^T & \mathbf{W}_B\end{bmatrix} \tag{6}$$

The sub-matrices $\mathbf{K}_B$ and $\mathbf{W}_B$ are not explicitly computed or stored at any step of our approximation [1]. Eigen-decomposition of the symmetrized matrix $\mathbf{Q}$ (shown in Fig. 2) gives the leading orthonormal eigenvectors $\mathbf{V}_m$ and eigenvalues $\mathbf{S}_m$ of the approximated filter.

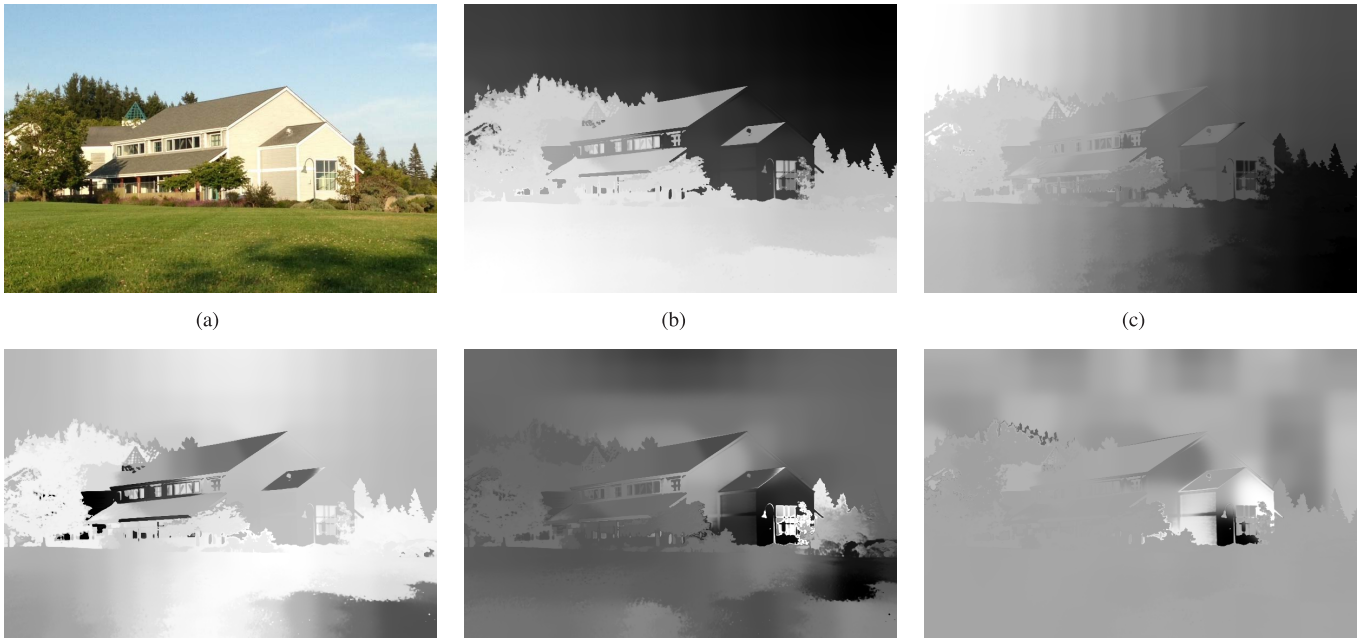


Fig. 3. Some leading eigenvectors computed from the luminance channel of the house image using less than 0.04% of the pixels. ($h_x = 20$, $h_y = 5$). (a) House. (b) $\mathbf{v}_2(\lambda_2 = 0.9966)$. (c) $\mathbf{v}_3(\lambda_3 = 0.9948)$. (d) $\mathbf{v}_{10}(\lambda_{10} = 0.9667)$. (e) $\mathbf{v}_{25}(\lambda_{25} = 0.8810)$. (f) $\mathbf{v}_{50}(\lambda_{50} = 0.7387)$.

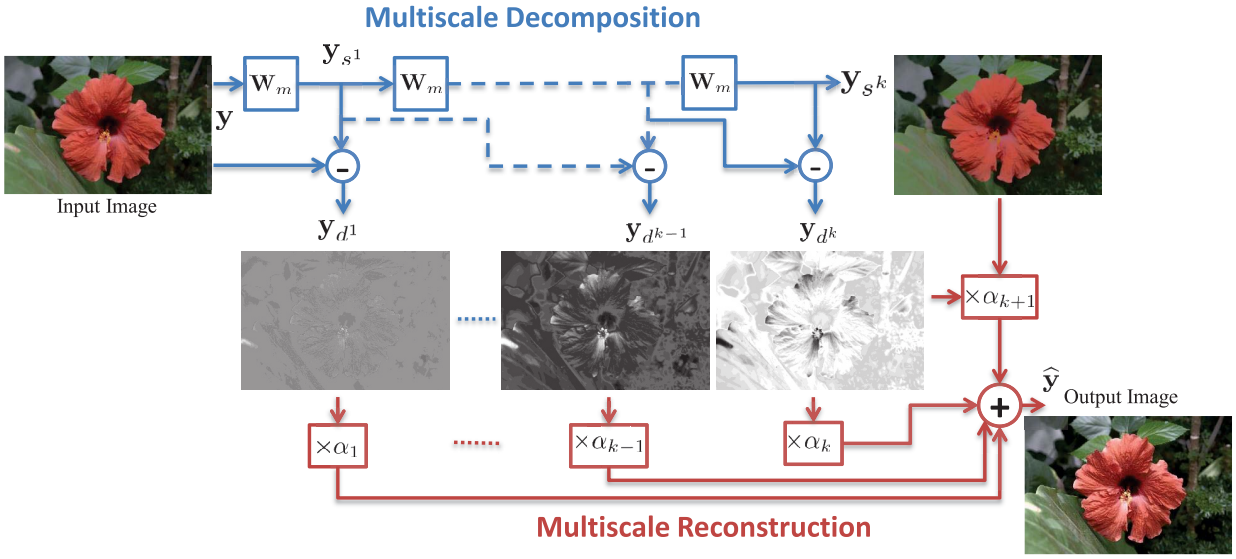


Fig. 4. (a) Mutiscale decomposition: The low-pass filter $\mathbf{W}_m$ is used to extract detail layers $\mathbf{y}_{d^i}$. Multiscale reconstruction: Weighting each layer with α_i and adding them together.

D. Non-Local Affinities

The spatial and photometric smoothing parameters of the kernel (h_x and h_y in (1)) can affect the rank of the kernel matrix $\mathbf{K}$. Specially small h_x makes the kernel matrix $\mathbf{K}$ to be high-rank (more diagonally dominant), and as a result, the Nyström extension will need more samples for an accurate reconstruction of the eigenvectors of the filter matrix. However, we observed that this approximation of the filter matrix $\mathbf{W}_m$ also has some nice applications. To illustrate, the leading eigenvectors of Fig. 1(a) are computed for the kernel with small spatial smoothing parameter in Fig. 3. Comparing these eigenvectors to the ones presented in Fig. 1, it can be seen that the spatial term forces the eigenvectors to become piecewise smooth. As we will discuss in Section V, these eigenfunctions force the filter to become a local smoother. In this paper, h_x is assumed to be very large (practically NLM kernel without the spatial term) for the purpose of detail manipulation, and fixed as 20 for other applications. Based on image content, we also optimize h_y to minimize the filter approximation error (See Appendix A).

III. Eigenvalue Mapping Function

The filtered image $\widehat{\mathbf{y}}$ can be expressed as:

$$\widehat{\mathbf{y}} = f(\mathbf{W})\mathbf{y} \tag{7}$$

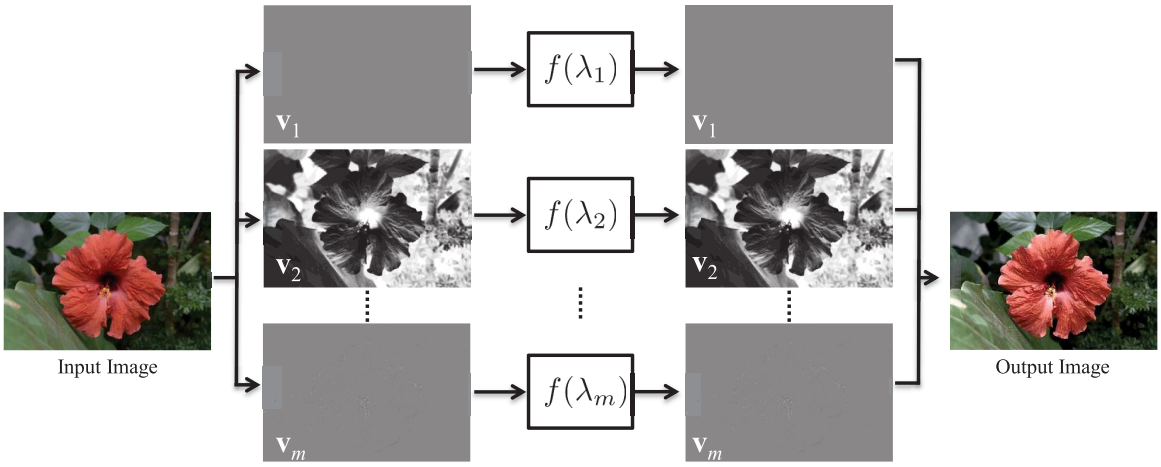


Fig. 5. The process given in Fig. 4 can be interpreted as the band-pass $\mathbf{V}_m f(\mathbf{S}_m)\mathbf{V}_m^T$ in which the eigenvalues are a polynomial function of the low-pass filter's eigenvalues given by (14).

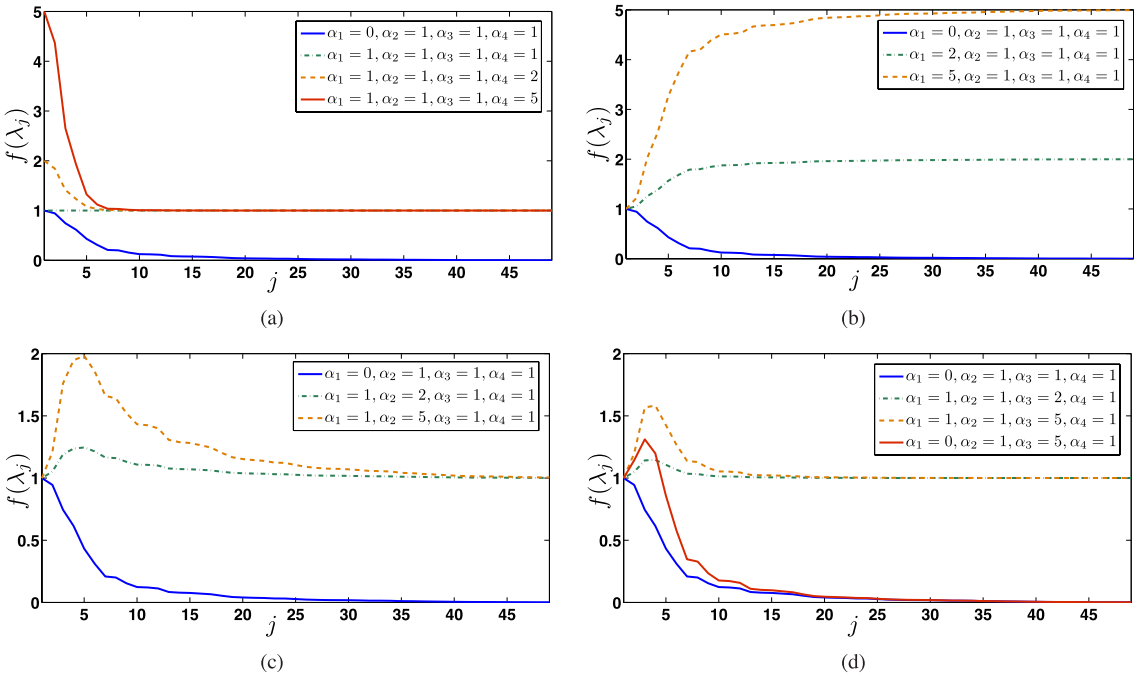


Fig. 6. (a)-(d) The 3th order function $f(\lambda_j)$ is evaluated for different α_i weights.

where in general, $f(\mathbf{W})$ denotes a matrix function. Analogously to scalar analytic functions, matrix functions of an $n\times n$ square matrix $\mathbf{W}$ can be defined using a power series:

$$f(\mathbf{W}) = \sum_{i=0}^{\infty} c_i \mathbf{W}^i \tag{8}$$

The above series exists and is finite for a given argument, if the coefficients c_i satisfy $\sum_{i=0}^{\infty} c_i x^i < \infty$ [31]. Therefore for any analytic function $f(x)$ there exists a corresponding matrix function $f(\mathbf{W})$ constructed by the power series. The analytic (differentiability) constraint can be relaxed over a *closed* interval using the Weierstrass approximation theorem [32] which only needs the function $f(x)$ to be continuous on an interval $[a, b]$. Based on this theorem, for every $\epsilon > 0$, there exists a polynomial $p_k(x)$ of sufficiently high degree k, such that for all x in $[a, b]$, we have $\|f(x) - p_k(x)\| < \epsilon$. Consequently, the matrix function $f(\mathbf{W})$ given in (8) can be approximated by a matrix polynomial $p_k(\mathbf{W})$ of order k:

$$f(\mathbf{W}) \approx p(\mathbf{W}) = c_0\mathbf{I} + c_1\mathbf{W} + c_2\mathbf{W}^2 + \cdots + c_k\mathbf{W}^k \tag{9}$$

with k a function of ϵ, as long as the eigenvalues of $\mathbf{W}$ reside in a closed compact interval. Having the matrix filter eigendecomposition as $\mathbf{W} = \mathbf{V}\mathbf{S}\mathbf{V}^T$, *any* continuous function on the eigenvalue interval [0, 1] can be approximated as:

$$\begin{aligned} f(\mathbf{W}) &\approx c_0\mathbf{V}\mathbf{I}\mathbf{V}^T + c_1\mathbf{V}\mathbf{S}\mathbf{V}^T + c_2\mathbf{V}\mathbf{S}^2\mathbf{V}^T + \cdots + c_k\mathbf{V}\mathbf{S}^k\mathbf{V}^T \\ &= \mathbf{V}(c_0\mathbf{I} + c_1\mathbf{S} + c_2\mathbf{S}^2 + \cdots + c_k\mathbf{S}^k)\mathbf{V}^T \\ &= \mathbf{V}p_k(\mathbf{S})\mathbf{V}^T \end{aligned} \tag{10}$$

where $p_k(\mathbf{S}) = \text{diag}[p_k(\lambda_1), p_k(\lambda_2), \cdots, p_k(\lambda_n)]$ denotes the mapped eigenvalues. Instead of approximating, we chose to directly use a polynomial function for $f(x)$, because it conveniently gives the interesting interpretation of the multiscale decomposition/reconstruction explained in the following.

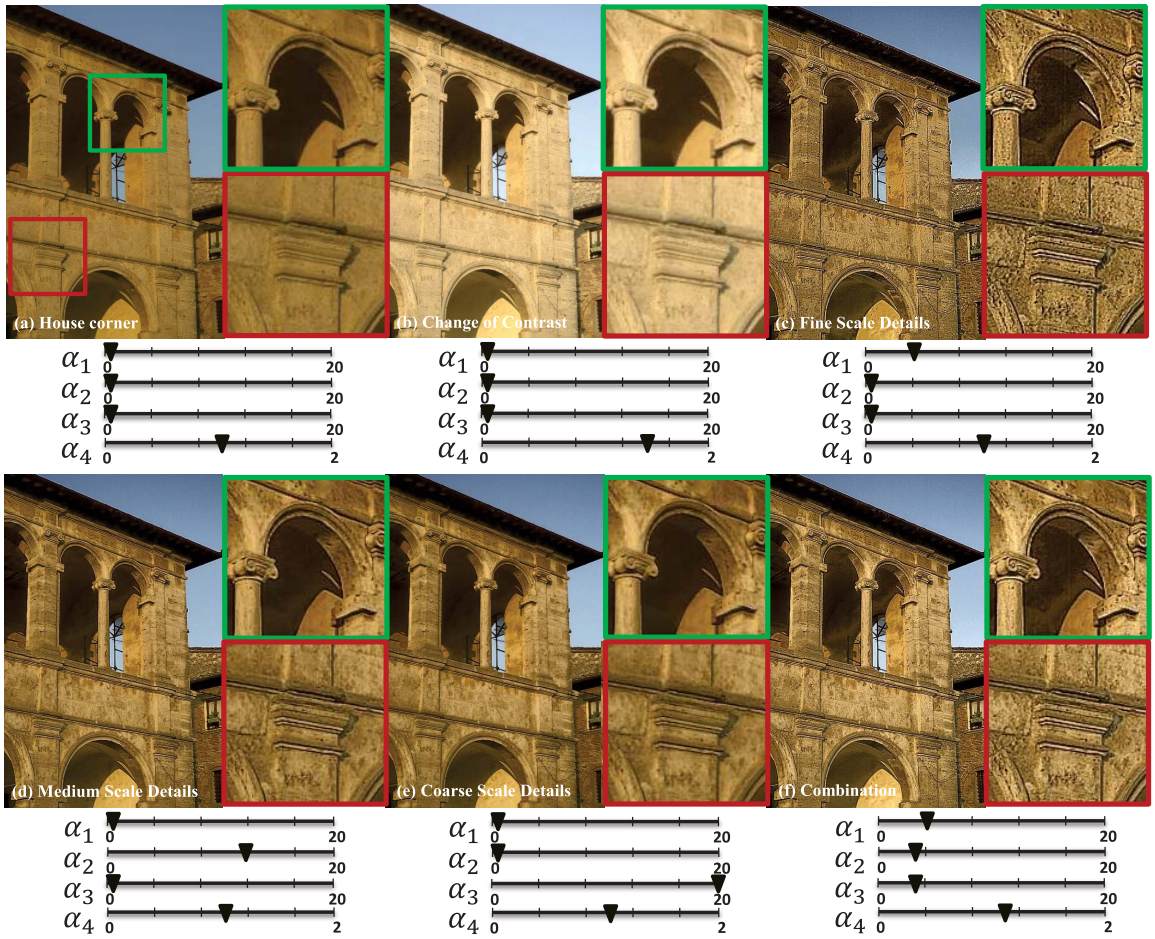


Fig. 7. Contrast and detail manipulation of the house corner image. (a) Input image, (b) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1.4$, (c) $\alpha_1 = 4, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1$, (d) $\alpha_1 = 1, \alpha_2 = 12, \alpha_3 = 1, \alpha_4 = 1$, (e) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 20, \alpha_4 = 1$, (f) $\alpha_1 = 4, \alpha_2 = 3, \alpha_3 = 3, \alpha_4 = 1.05$.

A. Multiscale Detail Manipulation

Fig. 4 depicts the multiscale decomposition and reconstruction where the input image $\mathbf{y}$ is layered to k detail layers $\mathbf{y}_{d^i}$ and one basic smooth layer $\mathbf{y}_{s^k}$ such that the exact decomposition is:

$$\mathbf{y} = \mathbf{y}_{d^1} + \ldots + \mathbf{y}_{d^k} + \mathbf{y}_{s^k} \tag{11}$$

The edited image $\widehat{\mathbf{y}}$ can be computed by weighting each layer and adding them back together:

$$\widehat{\mathbf{y}} = \alpha_1 \mathbf{y}_{d^1} + \ldots + \alpha_k \mathbf{y}_{d^k} + \alpha_{k+1} \mathbf{y}_{s^k} \tag{12}$$

Replacing the orthonormal eigen-decomposition of the filter $\mathbf{W}_m$ in the above, the equivalent filter is (Fig. 5):

$$\begin{aligned}
\widehat{\mathbf{y}} &= \alpha_1(\mathbf{I} - \mathbf{W}_m)\mathbf{y} + \alpha_2 \mathbf{W}_m(\mathbf{I} - \mathbf{W}_m)\mathbf{y} \\
&\quad + \ldots + \alpha_k \mathbf{W}_m^{k-1}(\mathbf{I} - \mathbf{W}_m)\mathbf{y} + \alpha_{k+1}\mathbf{W}_m^k \mathbf{y} \\
&\approx \mathbf{V}_m(\alpha_1(\mathbf{I} - \mathbf{S}_m) + \alpha_2 \mathbf{S}_m(\mathbf{I} - \mathbf{S}_m) \\
&\quad + \ldots + \alpha_k \mathbf{S}_m^{k-1}(\mathbf{I} - \mathbf{S}_m) + \alpha_{k+1}\mathbf{S}_m^k)\mathbf{V}_m^T \mathbf{y} \\
&= \mathbf{V}_m(\alpha_1 \mathbf{I} + \mathbf{S}_m(\alpha_2 - \alpha_1) + \mathbf{S}_m^2(\alpha_3 - \alpha_2) \\
&\quad + \ldots + \mathbf{S}_m^k(\alpha_{k+1} - \alpha_k))\mathbf{V}_m^T \mathbf{y} \\
&= \mathbf{V}_m f(\mathbf{S}_m)\mathbf{V}_m^T \mathbf{y}
\end{aligned} \tag{13}$$

where the approximation concerns replacing $\mathbf{y}$ by $\mathbf{V}_m \mathbf{I} \mathbf{V}_m^T \mathbf{y}$ (see Appendix B for detailed explanations). Function f has the following effect on each eigenvalue λ_j:

$$\begin{aligned}
f(\lambda_j) &= \alpha_1 + (\alpha_2 - \alpha_1)\lambda_j + (\alpha_3 - \alpha_2)\lambda_j^2 \\
&\quad + \ldots + (\alpha_k - \alpha_{k-1})\lambda_j^{k-1} + (\alpha_{k+1} - \alpha_k)\lambda_j^k
\end{aligned} \tag{14}$$

This is a special polynomial with $f(0) = \alpha_1$ and $f(1) = \alpha_{k+1}$. The two coefficients α_1 and α_{k+1} correspond to the first detail layer and the basic smooth image, respectively. For example, a 3rd order polynomial is evaluated as a function of the input eigenvalues in Fig. 6. While the input eigenvalues act as a low-pass filter (this is equivalent to $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1$ in the new filter $f(\mathbf{W}_m)$, the function f can change the filter's behavior with α_i as the tuning parameter. The mapped eigenvalues in Fig. 6(a) keep the details of the image untouched, but the leading eigenvalue manipulates the contrast of the image. Fig. 6(b) is a high-pass filter with the opposite effect as compared to part (a). Fig. 6(c) and (d) are two band-pass filters boosting the eigenvectors containing the main edges and possibly suppressing the existing noise.

Figs. 7-10 give a visual comparison of the filters described above. As can be seen, applying the filter $f(\mathbf{W}_m)$ can boost the

Fig. 8. Contrast and detail manipulation of the flower image. (a) Input image, (b) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1.5$, (c) $\alpha_1 = 5, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1$, (d) $\alpha_1 = 1, \alpha_2 = 10, \alpha_3 = 1, \alpha_4 = 1$, (e) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 15, \alpha_4 = 1$, (f) $\alpha_1 = 3, \alpha_2 = 5, \alpha_3 = 10, \alpha_4 = 1.1$.

contrast and details of the image in fine, medium and coarse scales.

IV. Globalizing Mask

As discussed earlier, the proposed global filter connects all the pixels in the image with weights commensurate to their similarity. As a result, any set of edits can be appropriately globalized to the whole image. In particular, one can automatically endow similar pixels with likewise similar editing parameters. Using the concept of diffusion maps [33], each pixel located at position $\mathbf{x}_i$ is mapped into a manifold defined by the weighted eigenvectors as:

$$\boldsymbol{\Psi}_t(\mathbf{x}_i) = (\lambda_2^t v_{i2}, \lambda_3^t v_{i3}, \ldots, \lambda_m^t v_{im}) \tag{15}$$

where t denotes the diffusion parameter and v_{ik} denotes the i-th entry of the k-th eigenvector. The squared diffusion distance between two pixels located at positions $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as:

$$D_t^2(\mathbf{x}_i, \mathbf{x}_j) = \|\boldsymbol{\Psi}_t(\mathbf{x}_i) - \boldsymbol{\Psi}_t(\mathbf{x}_j)\|_2^2 = \sum_{l=2}^{m} \lambda_l^{2t}(v_{il} - v_{jl})^2 \tag{16}$$

As t grows, the diffusion distance between any two pixels on the manifold shrinks. We employ a simple Gaussian function to embed this squared distance into $[0, 1]$ interval and define the propagation mask as:

$$\mathbf{M}_t(\mathbf{x}_s) \equiv \exp(-D_t^2(\mathbf{x}_i, \mathbf{x}_s)) \tag{17}$$

where $\mathbf{x}_s$ refers to the average diffusion map (given in (15)) over a region S selected by the user. $\mathbf{M}_t$ values close to one represent the pixel candidates to be propagated with the edit and $\mathbf{M}_t$ values close to zero stand for pixels that receive a small fraction of the edit. Fig. 11 illustrates the globalizing mask evaluated for different diffusion parameters obtained from the user's input. As can be seen, larger diffusion parameters lead to more aggressively propagated masks. Using these masks, it can be seen that the edited region is propagated to the similar pixels in Fig. 12 and Fig. 13. Comparing results of the global editing (Fig. 12(d) and Fig. 13(d)) and propagated edit (Fig. 12(e) and Fig. 13(e)), we can conclude that: (1) The main edges of the image are preserved and there is almost no halo effect on them, (2) Existing noise in image regions with low SNR is no longer boosted. Our results are also compared to the adaptive [12] and constraint [28] unsharp masking

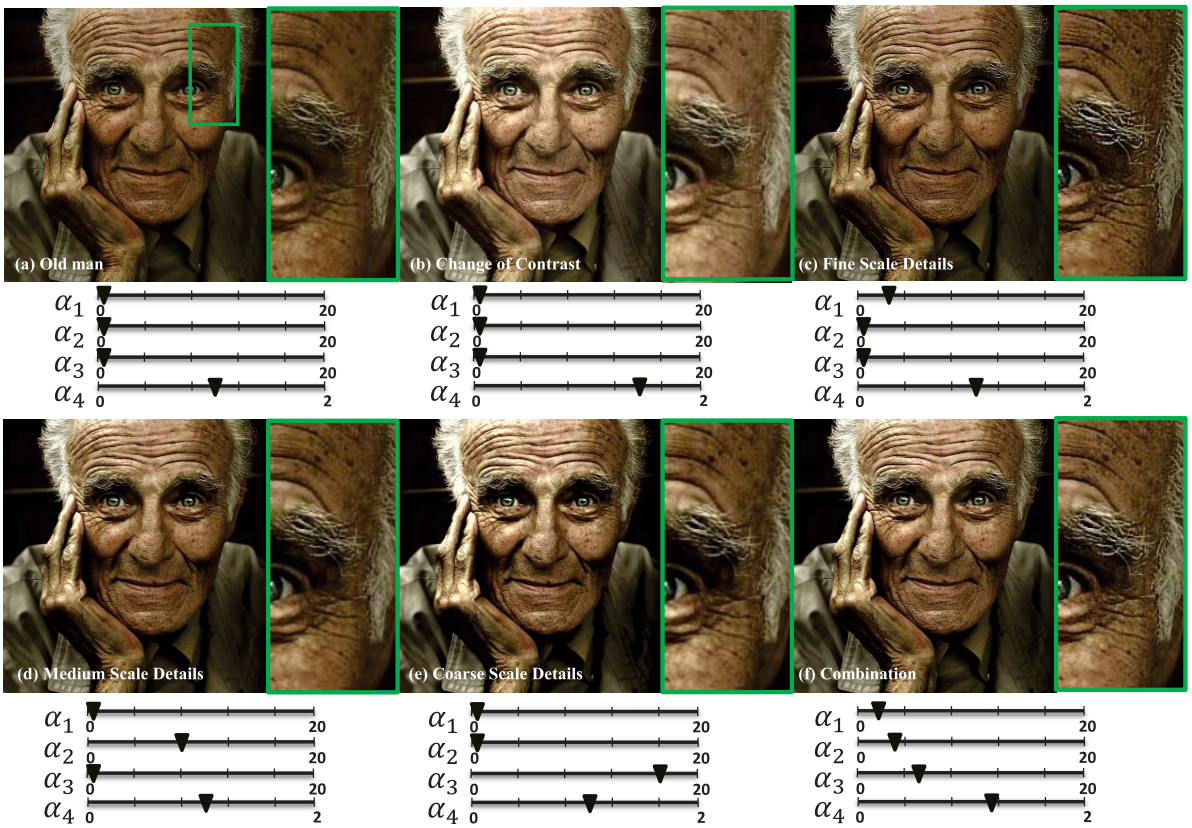


Fig. 9. Contrast and detail manipulation of the old man image. (a) Input image, (b) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1.4$, (c) $\alpha_1 = 3, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1$, (d) $\alpha_1 = 1, \alpha_2 = 8, \alpha_3 = 1, \alpha_4 = 1$, (e) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 16, \alpha_4 = 1$, (f) $\alpha_1 = 2, \alpha_2 = 3, \alpha_3 = 5, \alpha_4 = 1.1$.

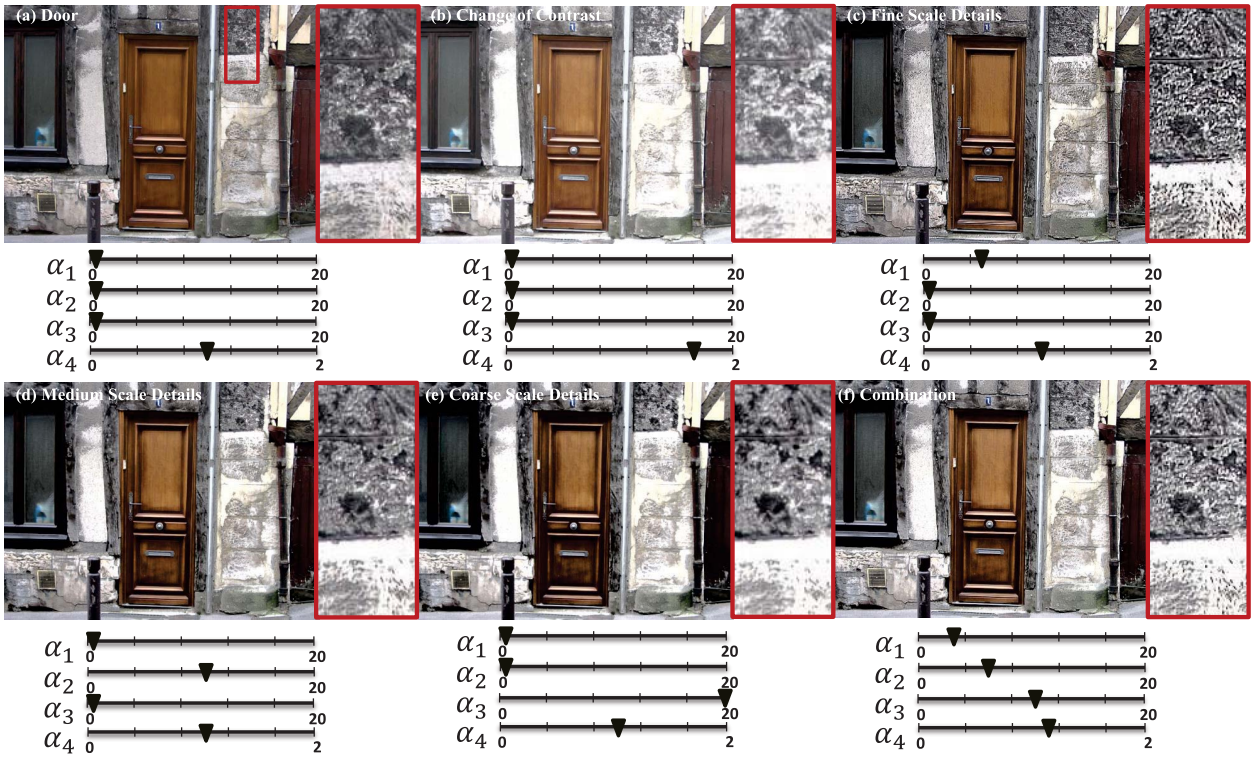


Fig. 10. Contrast and detail manipulation of the door image. (a) Input image, (b) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1.5$, (c) $\alpha_1 = 5, \alpha_2 = 1, \alpha_3 = 1, \alpha_4 = 1$, (d) $\alpha_1 = 1, \alpha_2 = 10, \alpha_3 = 1, \alpha_4 = 1$, (e) $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 20, \alpha_4 = 1$, (f) $\alpha_1 = 3, \alpha_2 = 6, \alpha_3 = 10, \alpha_4 = 1.1$.

in Fig. 12 and Fig. 13. As can be seen, while the sharpness is well enhanced, both noise and artifacts are suppressed in our results.

Depending on the application, the user can include the spatial term in the NLM kernel (1). This leads to a class of eigenvectors shown in Fig. 3 where similarity is captured

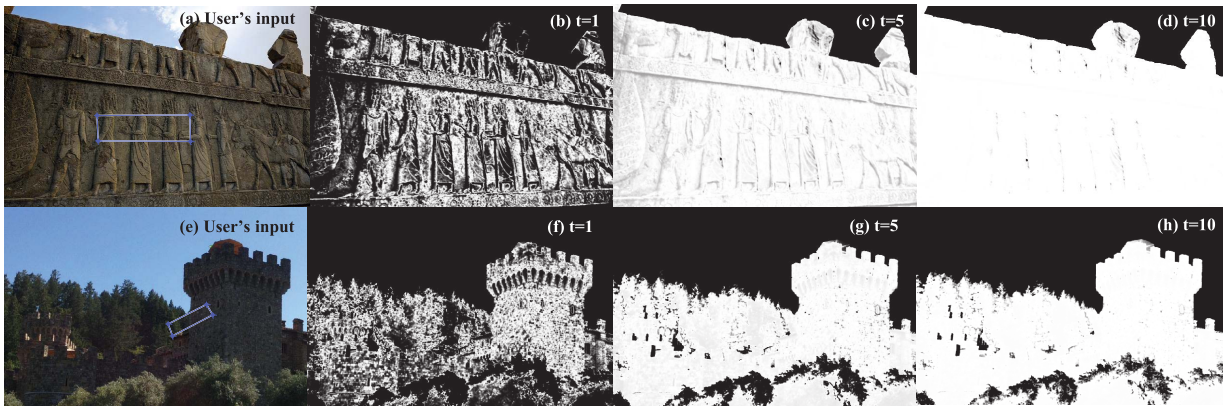


Fig. 11. Propagation masks with different diffusion parameters for Persepolis and Castle images.

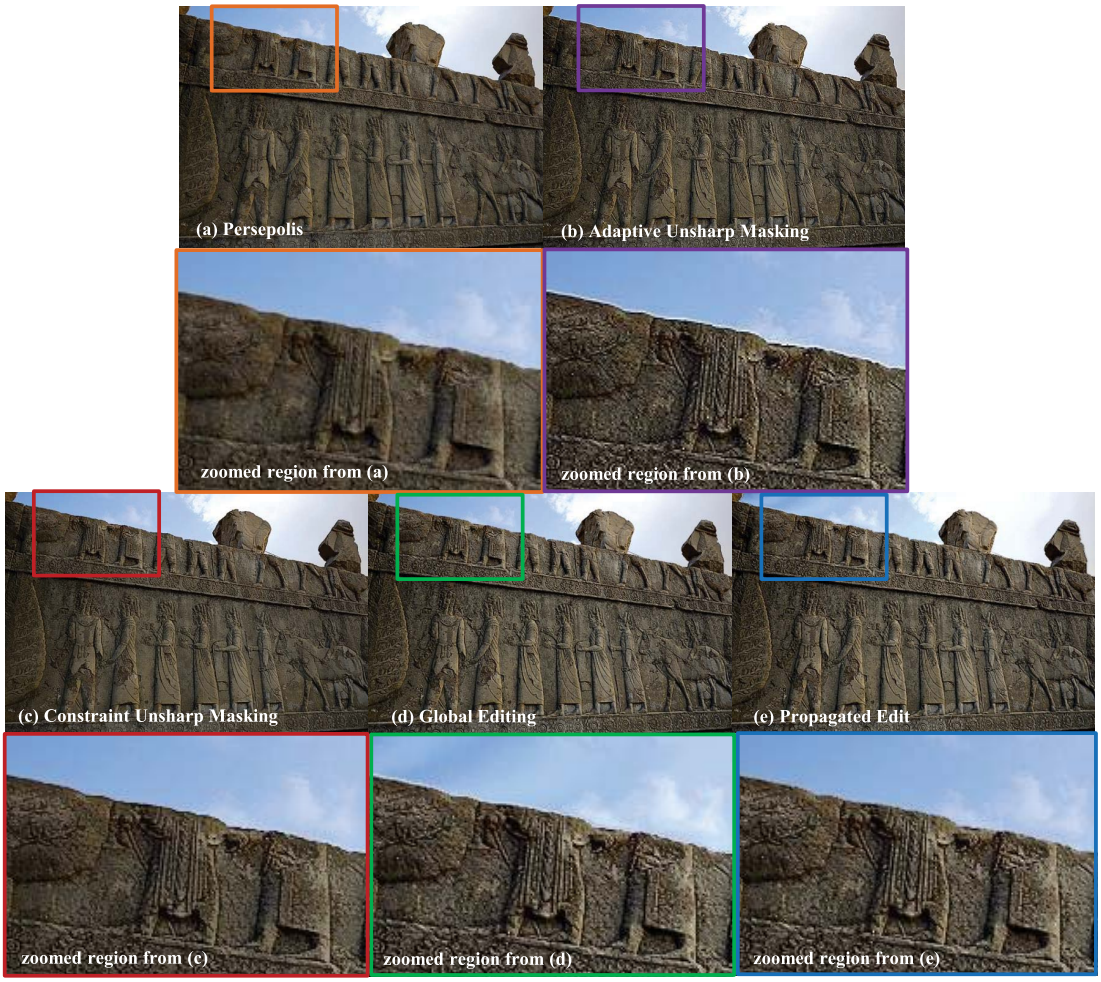


Fig. 12. Detail propagation of the Persepolis image compared to the results from adaptive unsharp masking [12] and constraint unsharp masking [28]. Edit propagation (shown in (e)) reduces the halo artifacts compared to the global edit (shown in (d)) and adaptive unsharp masking [12].

more locally. Propagation masks built on these eigenvectors are used to explore other applications of our framework in the following.

V. Practical Applications

The global filter studied here has many applications such as sharpening, denoising, image recoloring, colorization, image abstraction and fake depth of field. For the purpose of recoloring and abstraction the kernel can be computed from the luminance channel. Color channels can be used to make the kernel for colorization and fake depth of field. In general, after computing the leading eigenvectors, an appropriate propagation mask is built to apply the user's input.

A. Recoloring

Figs. 14-15 illustrate recoloring examples where the color strokes are propagated. These color strokes are applied on the chrominance channels. As discussed previously, image pixels can be represented by their corresponding diffusion map

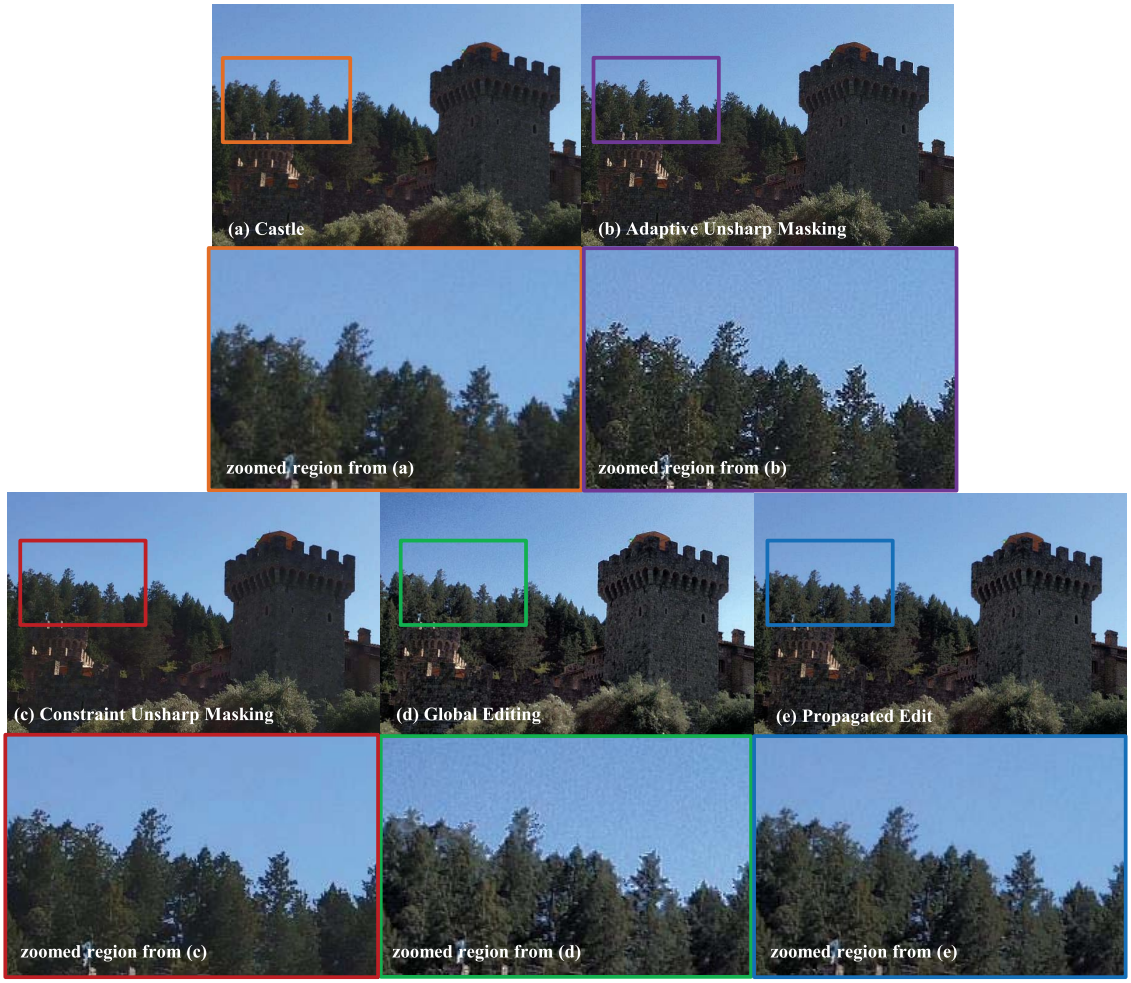


Fig. 13. Detail propagation of the castle image compared to the results from adaptive unsharp masking [12] and constraint unsharp masking [28]. Edit propagation (shown in (e)) reduces noise and halo artifacts compared to the global edit (shown in (d)) and adaptive unsharp masking [12].

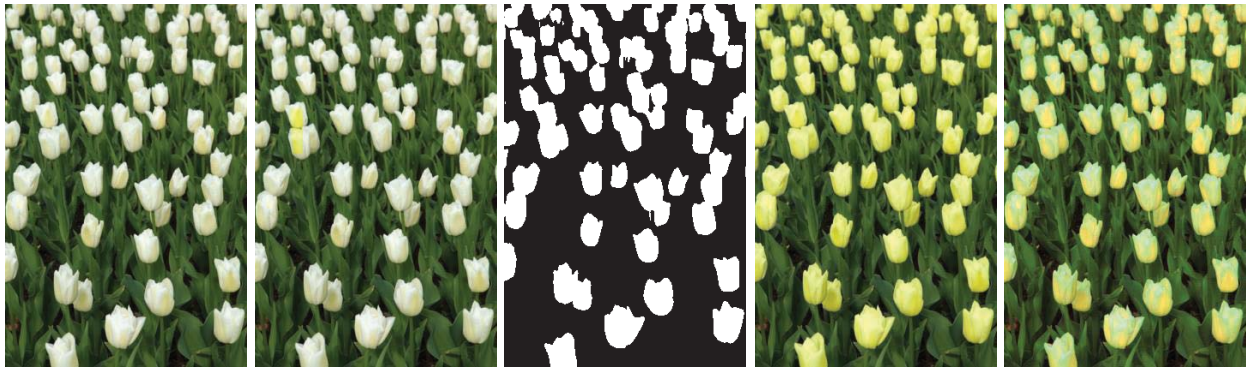


Fig. 14. Recoloring example based on the propagation mask. Results from the proposed method (d) and color replacement tool of Photoshop CC (e) are compared. (a) Tulips. (b) User's input. (c) Mask. (d) Output. (e) Photoshop CC.

coordinates given in (15). Thus, each image region selected by user strokes contains a group of pixels with the diffusion map vectors assigned to them. A binary mask can be obtained (see Figs. 14(c)-15(c)) using k-means clustering with the center of each diffusion map group as its initialization points. As shown in Figs. 14(d)-15(d), using this guide mask, the input color brushes are propagated in the chroma channels. Again, the mask can be tuned using the diffusion parameter t. Our globalizing mask for colorization application is also obtained through the same procedure described here.

Non-local nature of our recoloring method is better illustrated in Fig. 14 where a small color brush on one tulip is effectively propagated to other tulips. Color replacement tool in Photoshop CC was used for the same purpose in Fig. 14(e).

Fig. 15. Recoloring example based on the propagation mask. The mask (c) is built based on the two input color brushes (b).

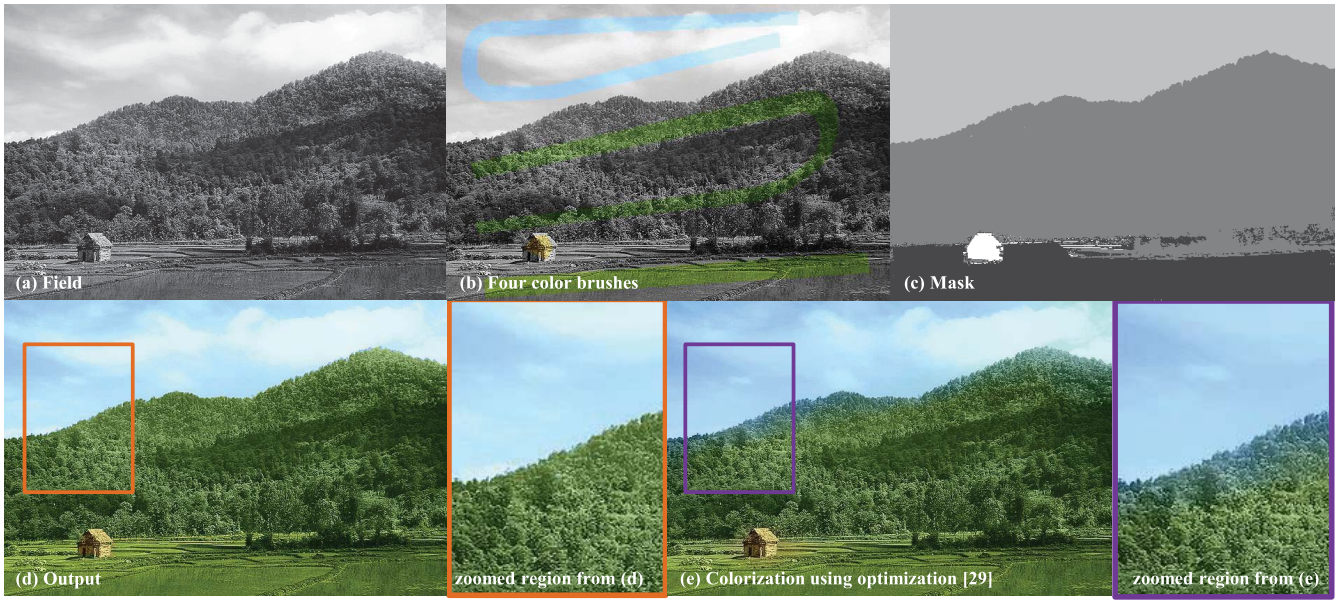


Fig. 16. Colorization example based on the propagation mask. The gray scale image is given in (a) and our input brush colors are shown in (b). Using the mask in (c) the color brushes are propagated throughput the gray image (shown in (d)). For a better comparison, our output is shown next to the results from [29].

As can be seen, the result obtained from Photoshop fails to evenly propagate the color stroke. Given that the color replacement tool employs a soft propagation mask, it tends to unequally spread the input color even across the same objects. Fig. 15 shows recoloring example by two different color brushes. In this case the mask was tuned by the diffusion parameter (t) in a way that the right and left side flowers in the image fall into different clusters.

B. Colorization

Our colorization example is shown in Fig. 16 where three input color brushes are propagated on the gray input image. The mask is based on affinities computed from the luminance channel. Same as recoloring, the mask is binerized using k-means to obtain a segmented guide map Fig. 16(c). The edit is propagated to the chroma components and the luminance channel is not changed. We also compared our result with the colorization method of Levin et al. [29] where an optimization problem is solved to propagate the input color scribbles. For the purpose of fair comparison, we fed the same color brushes shown in Fig. 16(b) to both methods. This example indicates superiority of our method while working with a few color brushes. As can be seen in the optimization colorization result, blue color is leaked on the trees. While adding more color brushes specially around the object boundaries may result in a better performance from [29], our method takes advantage

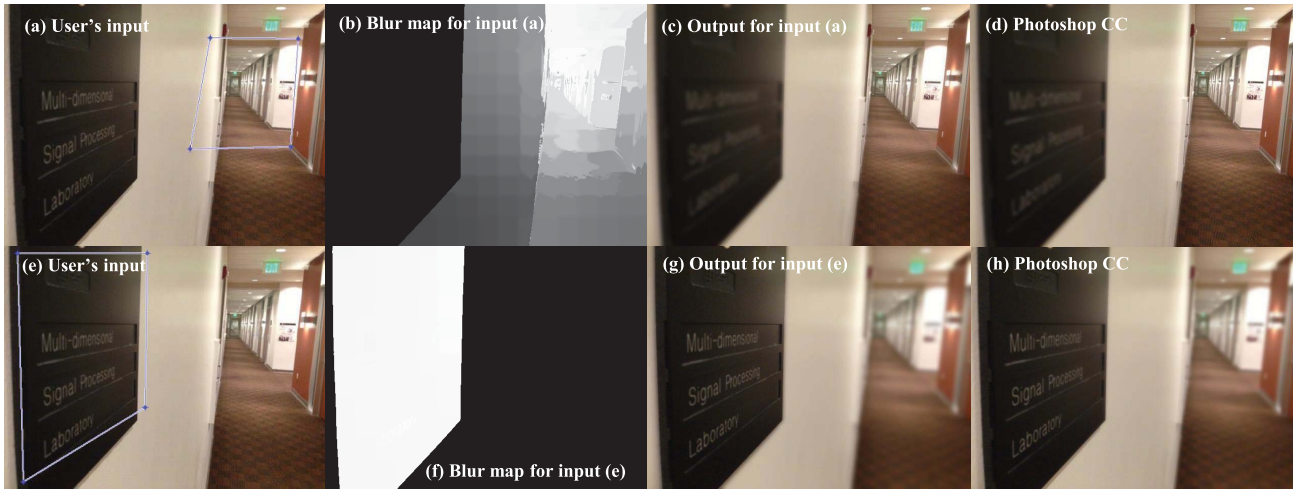


Fig. 17. Fake depth of field example. Based on the depth map, pixels with lower map values are more blurred. Our results shown in (c) and (g) are competitive to the manually edited outputs from Photoshop software in (d) and (h).

Fig. 18. Image abstraction application. (a) Input image and zoomed regions, (b) Our abstraction result as $\widehat{\mathbf{y}} = \mathbf{W}_m^k \mathbf{y}$ with iteration number $k = 0.1$, (c) The abstracted image given in (b) is stylized using the edge-exaggeration and luminance quantization of [30], (d) Result from [30].

of the guide mask to easily distinguish the sky and forrest regions. Yet, working with segmented guide masks might sometimes lead to false color propagation too. The project webpage contains more colorization examples comparing both methods and also shows some failed cases.

C. Fake Depth of Field

We can also employ the proposed propagation mask for applying fake depth of field. Although the produced mask does not represent the true depth information of the image, it still can be used to apply selective blur to each image region. Two examples are illustrated in Fig. 17, where the user selects the region which is supposed to stay in focus. Then, our blur map is built using the globalizing mask explained in Sec. IV and based on this, each pixel is blurred using a Gaussian function. The blur intensity of each pixel is proportional to the mask such that mask values close to 0 receive more blur. Our results are compared with the manually edited results obtained from Photoshop CC software where the same Gaussian blur was selected. The blur propagation of the results from Photoshop are manually tuned to get the best possible output. As can be seen, results obtained from our blur map is competitive with the Photoshop outputs.

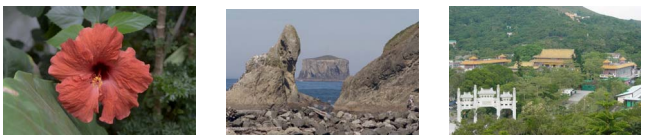
Based on our observation, the kernel's spatial and photometric terms play important roles in building the blur map. We used the RGB channels in the photometric term of the NLM kernel and included the spatial term as:

$$K_{ij} = \exp\left\{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{h_x^2} + \frac{-\|\mathbf{y}_i^{RGB} - \mathbf{y}_j^{RGB}\|^2}{h_y^2}\right\} \quad (18)$$

The spatial term enforces the locality of the propagated blur and the RGB photometric term distinguishes object boundaries with different colors. Relatively large h_y values suit blur maps of images with distinctive color boundaries in background and foreground. On the other hand, blur maps of images with comparably similar color in background and foreground could be better localized by tuning h_x. Of course any fake depth of field method has its own limitations since the real depth information is not available.

D. Abstraction

Fig. 18 illustrates the abstraction application where iteration of the filter leads to stylized filtered image as $\widehat{\mathbf{y}} = \mathbf{W}_m^k \mathbf{y}$ (effect of the iteration number k is illustrated in the project website where different results are shown with various iteration numbers). In practice, instead of multiplication of the filter matrix, the iterative filtering is implemented by raising the eigenvalues to a power.



(a) Flower (b) Rock (c) Monastery

Fig. 19. Test images with the following mean gradient magnitudes: (a) 3.66, (b) 7.51, (c) 12.78.

Fig. 18 shows our stylized result compared to the one from Winnemöler et al. algorithm [30]. In this method, the input image is filtered by an iterative bilateral filter to produce a piece-wise smooth output. Then, an edge-exaggeration step followed by a luminance quantization is employed to stylize the abstracted image (see Fig. 18(d)). To further stylize our abstracted result shown in Fig. 18(b) and also for a better comparison to [30], the edge-exaggeration and luminance quantization of [30] was applied on our abstracted result to produce Fig. 18(c). Although comparing abstracted images is very subjective, Fig. 18(c) and Fig. 18(d) show that replacing the iterative bilateral smoother of [30] with our filter can result in visually superior results.

VI. Conclusion

In the current work, some applications of the global, affinity based filters are explored. Having eigenfunctions of these filters, complex nonlinear smoothing or sharpening operators are easily implemented by means of mapping the corresponding eigenvalues. The global nature of the eigenvectors let us propagate these edits easily throughout the image. This type of editing enables us to prevent noise magnification and also effectively reduce common artifacts such as halo. Beside filtering applications, the obtained propagation map can be used to effect many other types of edits.

Appendix A
Parameter Tuning of the Filter

The approximated filter spans the input image onto its eigenvectors. These projected image coefficients are $b_j = \mathbf{v}_j^T \mathbf{y}$ where $j = 1, \ldots, m$. For any $m \leq n$, energy of these coefficients is less than the energy of the input image as $\sum_{j=1}^{m} b_j^2 \leq \sum_{i=1}^{n} y_i^2$ (ideally, the two energy terms are equal). The approximated filter should preserve energy of the image, which means β should be close to one, where:

$$\beta = \frac{\sum_{j=1}^{m} b_j^2}{\sum_{i=1}^{n} y_i^2} \tag{19}$$

Assuming that the sample number m is fixed, β is determined by the kernel smoothing parameter.

The approximation accuracy is directly affected by the smoothing parameter of the NLM kernel (h_y given in (1)). For the images given in Fig. 19, β is computed using different smoothing parameters in Table I. As can be seen, β gets closer to one for larger h_y. However, for a better multiscale decomposition, it is necessary to select a small smoothing parameter. We set $\beta \geq 99.9\%$ to determine the optimal

TABLE I

β Percentage Va lues for Images Given in Figure 19

h_y	2	4	6	8	10
Flower	**99.95**	99.99	100	100	100
Rock	99.83	**99.95**	99.97	99.98	99.99
Monastery	99.63	99.80	99.88	**99.93**	99.96

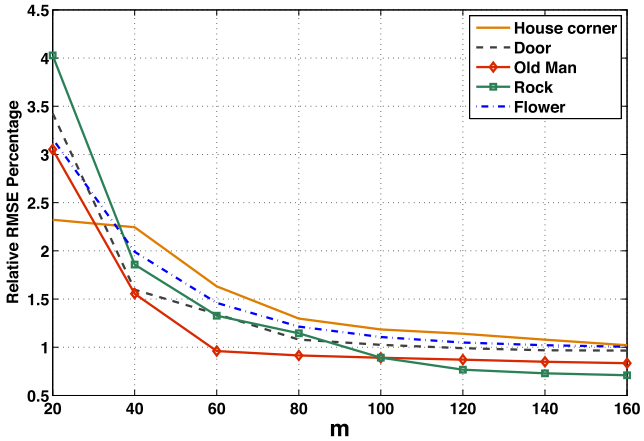


Fig. 20. Approximation error of (13) computed as relative RMSE $= \frac{\|\mathbf{y}-\mathbf{V}_m\mathbf{I}\mathbf{V}_m^T\mathbf{y}\|}{\|\mathbf{y}\|} \times 100\%$ for different numbers of retained eigenvectors m.

kernel parameter. This test could be useful after approximation to check if the kernel parameter is not too small or large. To avoid the computation burden, a simple ad-hoc method is introduced in the following to adapt the kernel smoothing parameter into the latent image.

Image content has a direct impact on the accuracy of the approximation. Images containing more variants tend to settle for larger h_y (for a fixed β). This can be observed by comparing the mean gradient magnitude of the test images (given in Fig. 19) and the β values in Table I. Without any need to compute β, we empirically estimate the smoothing parameter as a function of the mean gradient magnitude:

$$\widehat{h}_y = \frac{0.6}{n}\sqrt{\mathbf{g}_{\mathbf{x}_1}^T \mathbf{g}_{\mathbf{x}_1} + \mathbf{g}_{\mathbf{x}_2}^T \mathbf{g}_{\mathbf{x}_2}} \tag{20}$$

where $\mathbf{g}_{\mathbf{x}_1}$ and $\mathbf{g}_{\mathbf{x}_2}$ denote the gradient vectors in $\mathbf{x}_1$ and $\mathbf{x}_2$ directions.

Appendix B
Approximation of Eq. 13

The approximation made in (13) is $\mathbf{y} \approx \mathbf{V}_m\mathbf{I}\mathbf{V}_m^T\mathbf{y}$, or in effect replacing the identity matrix with $\mathbf{V}_m\mathbf{V}_m^T$. This assumption means that the underlying image $\mathbf{y}$ should be "well represented" (or spanned) by these eigenvector bases. For a better validation of this approximation, Fig. 20 shows relative root-mean-square-error (relative RMSE $= \frac{\|\mathbf{y}-\mathbf{V}_m\mathbf{I}\mathbf{V}_m^T\mathbf{y}\|}{\|\mathbf{y}\|} \times 100\%$) for some of the test images. The relative RMSE values are computed for images in range [0,255] and various values of m. Unsurprisingly, the approximation error shrinks as the number of retained eigenvectors grows. Overall, keeping $m > 75$ in our framework leads to around 1% approximation error.

REFERENCES

[1] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, Feb. 2014.

[2] 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, Aug. 2008, Art. ID 67.

[3] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Trans. Graph.*, vol. 28, no. 5, 2009, Art. ID 147.

[4] S. Paris, S. W. Hasinoff, and J. Kautz, "Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. ID 68.

[5] Z. Farbman, R. Fattal, and D. Lischinski, "Diffusion maps for edge-aware image editing," *ACM Trans. Graph.*, vol. 29, no. 6, Dec. 2010, Art. ID 145.

[6] X. An and F. Pellacini, "AppProp: All-pairs appearance-space edit propagation," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. ID 40.

[7] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 629–639, Jul. 1990.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th ICCV*, Bombay, India, Jan. 1998, pp. 836–846.

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

[10] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, no. 3, 2007, Art. ID 103.

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

[12] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 505–510, Mar. 2000.

[13] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[14] A. Choudhury and G. Medioni, "Perceptually motivated automatic sharpness enhancement using hierarchy of non-local means," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 730–737.

[15] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 664–678, May 2008.

[16] X. Zhu and P. Milanfar, "Restoration for weakly blurred and strongly noisy images," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 103–109.

[17] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[18] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 646–653, 2006.

[19] F. Pellacini and J. Lawrence, "AppWand: Editing measured materials using appearance-driven optimization," *ACM Trans. Graph.*, vol. 27, no. 3, 2007, Art. ID 54.

[20] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[21] E. Seneta, *Non-Negative Matrices and Markov Chains* (Statistics). New York, NY, USA: Springer-Verlag, 1981.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[23] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 263–284, 2013.

[24] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, vol. 13. Cambridge, MA, USA: MIT Press, 2001, pp. 682–688.

[25] E. Nyström, "Über die praktische auflösung von linearn ingtegraglechungen mit anwendungen auf randwertaufgaben der potentialtheorie," *Commentationes Phys.-Math.*, vol. 4, no. 15, pp. 1–52, Apr. 1928.

[26] P. Knight, "The Sinkhorn–Knopp algorithm: Convergence and applications," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 261–275, 2008.

[27] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, no. 2, pp. 876–879, 1964.

[28] R. C. Bilcu and M. Vehvilainen, "Constrained unsharp masking for image enhancement," in *Proc. 3rd Int. Conf. Image Signal Process.*, 2008, pp. 10–19.

[29] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

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

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[32] J. A. Dieudonne, *Foundations of Modern Analysis*. New York, NY, USA: Academic, 1960.

[33] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," in *Proc. Nat. Acad. Sci.*, 2005, pp. 7426–7431.

Hossein Talebi (S'11) received the B.S. and M.S. degrees in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the University of California at Santa Cruz, Santa Cruz, CA, USA, where he is also a Researcher with the Multi-Dimensional Signal Processing Laboratory. His research interests are in image and video processing (i.e., denoising, superresolution, and editing).

Peyman Milanfar (F'10) received the bachelor's degree in electrical engineering and mathematics from the University of California at Berkeley, Berkeley, CA, USA, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. Until 1999, he was with SRI International, Menlo Park, CA USA, and was a Consulting Professor of Computer Science with Stanford University, Stanford, CA, USA. He has been with the Faculty of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA, USA, since 1999, and served as an Associate Dean of the School of Engineering from 2010 to 2012. Since 2012, he has been on leave with Google Research and Google-X, Mountain View, CA, USA, where he was involved in computational photography, and in particular, the imaging pipeline for Google Glass. He was a recipient of the National Science Foundation CAREER Award, in 2000, and the Best Paper Award from the IEEE Signal Processing Society in 2010.