A NEW CLASS OF IMAGE FILTERS WITHOUT NORMALIZATION

Peyman Milanfar and Hossein Talebi

Google Research
Email : {milanfar, htalebi}@google.com

ABSTRACT

When applying a filter to an image, it often makes practical sense to maintain the local brightness level from input to output image. This is achieved by normalizing the filter coefficients so that they sum to one. This concept is generally taken for granted, but is particularly important where non-linear filters such as the bilateral or and non-local means are concerned, where the effect on local brightness and contrast can be complex. Here we present a method for achieving the same level of control over the local filter behavior without the need for this normalization. Namely, we show how to closely approximate any normalized filter without in fact needing this normalization step. This yields a new class of filters. We derive a closed-form expression for the approximating filter and analyze its behavior, showing it to be easily controlled for quality and nearness to the exact filter, with a single parameter. Our experiments demonstrate that the un-normalized affinity weights can be effectively used in applications such as image smoothing, sharpening and detail enhancement.

Index Terms— Edge-aware filters, Image enhancement

1. INTRODUCTION AND BACKGROUND

Edge-aware filters are constructed using kernels that are computed from the given image. The adaptation of the filters to local variations in the image is what endows them with the power and flexibility to treat different parts of the image differently. This adaptability, however, can not be arbitrary. In particular, the local brightness of the image must often be maintained in order to yield a reasonable global appearance. The standard way to achieve this is by normalizing the filter coefficients pointing to each pixel, so that they sum to 1. In this paper we propose a new and different way. Namely, we present a general method to approximate any normalized filter with one that does not require normalization. This produces a rather new class of filters with simpler structure, but with essentially the same functionality.

Approximation ideas centered around nonlinear filters are not new. In particular, the bilateral filter has been subject to various interesting algorithmic enhancements [1] which have resulted in significantly improved computational complexity with almost no loss in quality.

What we propose here is *not* another approximation to the bilateral filter. Our treatment works equally well for *any* normalized filter that has a well-defined kernel; bilateral, non-local means, etc. being just a few popular examples. Our filter avoids local normalization of the filter coefficients, while remaining close in its effect to the base filter. This approximation has several interesting properties. First, the fidelity of the approximation is guaranteed since it is derived from an optimality criterion; furthermore, this fidelity can be controlled easily with a single parameter regardless of the form of the base filter (e.g. bilateral, non-local mean, etc.) Second, the approximate filter is guaranteed to maintain the average gray level just as the base filter would, regardless of tuning. Finally, the approximate filter is easy to analyze and provides intuitively pleasing structure for understanding the behavior of general image-dependent filters. By way of practical motivation, the approximation allows us to start with an arbitrary (normalized) base filter and generate a one-parameter family of simpler nearby filters, which can locally modulate the effect of the base filter. This is different, and more flexible, than the typical approach where the base filters (bilateral, NLM, etc) are controlled with global smoothing parameters. For various applications such as texture-cartoon decomposition, guided filtering, and local (e.g. Laplacian) tone mapping, and even noise suppression, the additional flexibility afforded can be very useful.

Before moving forward, we establish our notation. Consider the vectorized image $\mathbf{y}$ of size n as the input, and the vectorized image $\mathbf{z}$ as the output of the filtering process. The general construction of a filter begins by specifying a symmetric positive semi-definite (PSD) kernel $k_{ij} \geq 0$ that measures the similarity, or affinity, between individual or groups of pixels. This affinity can be measured as a function of both the distance between the spatial variables (denoted by $\mathbf{x}$), but more importantly, also using the gray or color value (denoted by $\mathbf{y}$). While the results of this paper extend to any filter with an PSD kernel, some popular examples commonly used in the image processing, computer vision, and graphics literature are as follows:

Bilateral (BL) [2, 3] and Non-local Mean (NLM) [4, 5]: These filters take into account both the spatial *and* value distances between two pixels, generally in a separable fashion.

For BL we have:

$$k_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{h_x^2}\right)\exp\left(\frac{-(y_i - y_j)^2}{h_y^2}\right) \quad (1)$$

As seen in the overall exponent, the similarity metric here is a weighted Euclidean distance between the concatenated vectors $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$, where $\mathbf{x}_i$ and y_i represent location and value of i-th pixel.

The NLM kernel is a generalization of the bilateral kernel in which the value distance term (1) is measured patch-wise instead of point-wise:

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

where $\mathbf{y}_i$ and $\mathbf{y}_j$ refer now to *subsets* of samples (i.e. patches) in $\mathbf{y}$.

These affinities are not used directly to filter the images, but instead in order to maintain the local average brightness, they are normalized so that the resulting weights pointing to each pixel sum to one. More specifically,

$$w_{ij} = \frac{k_{ij}}{\sum_{j=1}^{n} k_{ij}}, \quad (3)$$

where each element of the filtered signal $\mathbf{z}$ is then given by

$$z_i = \sum_{j=1}^{n} w_{ij}\, y_j.$$

It is worth noting that the denominator in (3) can be computed by simply applying the filter (without normalization) to an image of all 1's.

In matrix notation, the collection of the weights used to produce the i-th output pixel is the vector $[w_{i1}, \cdots, w_{in}]$; and this can in turn be placed as the i-th row of a filter matrix $\mathbf{W}$ so that

$$\mathbf{z} = \mathbf{W}\mathbf{y}.$$

We note again that due to the normalization of the weights, the rows of the matrix $\mathbf{W}$ sum to one; That is, for each $1 \leq i \leq n$,

$$\sum_{j=1}^{n} w_{ij} = 1.$$

Viewed another way, the filter matrix $\mathbf{W}$ is a *normalized* version of the symmetric positive definite affinity matrix $\mathbf{K}$ constructed from the *unnormalized* affinities k_{ij}, $1 \leq i,\ j, \leq n$. As a result, $\mathbf{W}$ can be written as a product of two matrices

$$\mathbf{W} = \mathbf{D}^{-1}\mathbf{K}, \quad (4)$$

where $\mathbf{D}$ is a diagonal matrix with diagonal elements $[\mathbf{D}]_{ii} = \sum_{j=1}^{n} k_{ij} = d_i$. To avoid the normalization, we will replace the filter $\mathbf{W}$ with an approximation $\widehat{\mathbf{W}}$ that only involves $\mathbf{D}$ rather than its inverse. More specifically,

$$\widehat{\mathbf{W}} = \mathbf{I} + \alpha(\mathbf{K} - \mathbf{D}). \quad (5)$$

In what follows, we will motivate and derive this approximation from first principles, while also providing an analytically sound and numerically tractable choice for the scalar $\alpha > 0$ that gives the best approximation to $\mathbf{W}$ in the least-squares sense. Before doing so, it is worth noting some of the key propeties and advantages of this approximate filter which are evident from the above expression (5).

- Regardless of the value of α, the rows of $\widehat{\mathbf{W}}$ always sum to one.
- While the filter $\mathbf{W}$ is not neccessarily symmetric, the approximate $\widehat{\mathbf{W}}$ is always symmetric. The advantages of having a symmetric filter matrix are many, as documented in the recent work [6].
- The normalized filter weights in $\mathbf{W}$ are typically non-negative valued. The elements in $\widehat{\mathbf{W}}$ however, can be negative valued, meaning that the behavior of the approximate filter may differ from its reference value.

2. THE NORMALIZATION-FREE FILTER $\widehat{\mathbf{W}}$

To derive the approximation promised in the previous section, we first note that the standard filter can be written as:

$$\mathbf{W} = \mathbf{I} + \mathbf{D}^{-1}(\mathbf{K} - \mathbf{D}) \quad (6)$$

Comparing this form to the one presented earlier in (5), we note that the approximation is replacing the matrix inverse (on the right hand side) with a scalar multiple of the identity:

$$\mathbf{D}^{-1} \approx \alpha\mathbf{I}$$

As an illustration, an image containing the normalization terms d_j (which comprise the diagonal elements of $\mathbf{D}$) is shown in Fig. 1. The proposal, as we elaborate below, is to replace these normalization factors in (6) with a constant.

The motivation for this approximation is a Taylor series in terms of $\mathbf{D}$ for the filter matrix. In particular, let's consider the first few terms in the series around a nominal $\mathbf{D}_0$:

$$\mathbf{D}^{-1}\mathbf{K} \approx \mathbf{I} + \mathbf{D}_0^{-1}(\mathbf{K} - \mathbf{D}) - \mathbf{D}_0^{-2}(\mathbf{D} - \mathbf{D}_0)(\mathbf{K} - \mathbf{D}) \quad (7)$$

The series expressses the filter as a perturbation of the identity, where the second and third terms are linear and quadratic in $\mathbf{D}$. For simplicity, we can elect to retain only the linear term, arriving at the approximation

$$\mathbf{D}^{-1}\mathbf{K} \approx \mathbf{I} + \mathbf{D}_0^{-1}(\mathbf{K} - \mathbf{D}). \quad (8)$$

Letting $\mathbf{D}_0 = \alpha^{-1}\mathbf{I}$, we arrive at the suggested approximation in (5).

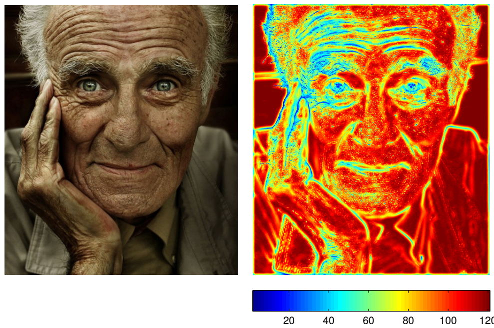

Fig. 1. (Left) Input, (Right) Values of d_j for the old man photo. Large values shown in red indicate pixels that have many "nearest neighbors" in the metric implied by the bilateral kernel.

2.1. Choosing the best α

A direct approach to optimizing the value of the parameter α is to minimize the following cost function using the matrix *Frobenius* norm:

$$\min_{\alpha} \|\mathbf{W} - \widehat{\mathbf{W}}(\alpha)\|^2 \tag{9}$$

We can write the above difference as

$$J(\alpha) = \|\mathbf{W} - \widehat{\mathbf{W}}(\alpha)\|^2 = \|(\mathbf{D}^{-1}\mathbf{K} - \mathbf{I} - \alpha(\mathbf{K} - \mathbf{D})\|^2$$

This is a quadratic function in α. Upon differentiating and setting to zero, we are led to the global minimum solution:

$$\widehat{\alpha} = \frac{\text{tr}(\mathbf{K}\mathbf{D}^{-1}\mathbf{K}) - 2\text{tr}(\mathbf{K}) + \text{tr}(\mathbf{D})}{\text{tr}(\mathbf{K}^2) - 2\text{tr}(\mathbf{K}\mathbf{D}) + \text{tr}(\mathbf{D}^2)} \tag{10}$$

For sufficiently large n, the terms tr($\mathbf{D}$) and tr($\mathbf{D}^2$) dominate the numerator and the denominator, respectively. Hence,

$$\widehat{\alpha} \approx \frac{\text{tr}(\mathbf{D})}{\text{tr}(\mathbf{D}^2)} = \frac{s_1}{s_2}, \tag{11}$$

where

$$s_1 = \sum_{i=1}^{n} d_i, \quad \text{and} \quad s_2 = \sum_{i=1}^{n} d_i^2 \tag{12}$$

This ratio is in fact bounded as $\frac{1}{n} \leq \frac{s_1}{s_2} \leq \frac{1}{\overline{d}}$, which for large n justifies a further approximation:

$$\widehat{\alpha} \approx \frac{1}{\overline{d}} \tag{13}$$

where $\overline{d} = \text{mean}(d_j)$. An image smoothing example using our approximated filter is shown in Fig. 2. As can be seen, output of the approximated filter is visually close to the exact filter result.

Fig. 2. (Left) Input $\mathbf{y}$; (Center) exact BL filter $\mathbf{z}$, and (right) un-normalized BL filter $\widehat{\mathbf{z}}$

3. PROPERTIES OF $\widehat{\mathbf{W}}$

As a matter of practical importance, we look at how the approximation changes the weights applied for computing any given pixel of the output image. To construct the pixel z_i of the output from the input image, the exact weights used (in the j-th row of $\mathbf{W}$) are:

$$z_i = \mathbf{w}_i^T\ \mathbf{y} = \frac{1}{d_i}\left[k_{i1}, \cdots, k_{ii}, \cdots, k_{in}\right]\mathbf{y}$$

In contrast to this, the approximate filter uses the weights

$$\widehat{z}_i = \widehat{\mathbf{w}}_i^T\ \mathbf{y} = \alpha\left[k_{i1}, \cdots, \alpha^{-1} + k_{ii} - d_i, \cdots, k_{in}\right]\mathbf{y}$$

Note that the center (self) weight corresponding to the position of interest i has been changed most prominently, and the other weights are pushed in the opposite direction as the change in this weight in order to maintain the sum as 1. The center weight in fact can become negative, while the other weights must remain positive.

Another way to make the comparison is more illustrative. Define the shifted Dirac delta vector:

$$\delta_i = [0, 0, \cdots, 0, 1, 0, 0, \cdots, 0]$$

where the subscript i indicates that the value 1 occurs in the i-th position. We have

$$\widehat{\mathbf{w}}_i^T = \delta_i + \alpha\left(\left[k_{i1}, \cdots, k_{ii}, \cdots, k_{in}\right] - d_i\delta_i\right) \tag{14}$$

Rewriting this last expression we have a rather simple relationship between the exact and un-normalized filter coefficients:

$$\widehat{\mathbf{w}}_i^T - \delta_i = \alpha\ d_i\left(\mathbf{w}_i^T - \delta_i\right) \tag{15}$$

So if we subtract 1 from the self-weight, then the resulting two filters are simply scaled by the coefficient αd_i, which controls the difference between the exact and the approximated filter. In particular, if at pixel location i the normalization factor d_i

is close to the mean $\overline{d}$, then at that pixel, the approximation is nearly perfect. More generally, gathering all terms like (15), the respective filter matrices are related as

$$(\widehat{\mathbf{W}} - \mathbf{I}) = \mathbf{R}(\mathbf{W} - \mathbf{I}) \tag{16}$$

where $\mathbf{R} = \alpha\mathbf{D}$.

Canonically, if we consider the two filters $\mathbf{W}$ and $\widehat{\mathbf{W}}$ as edge-aware low-pass (or smoothing) filters, then their counter-parts $\mathbf{W} - \mathbf{I}$ and $\widehat{\mathbf{W}} - \mathbf{I}$ are high pass filters. In fact, these filters are directly related to different (but well-established) definitions of the graph Laplacian operator emerging from the same affinity matrix $\mathbf{K}$. Namely, $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{K} = \mathbf{I} - \mathbf{W}$ is known as the *random walk* Laplacian [7], whereas $\widehat{\mathcal{L}} = \mathbf{D} - \mathbf{K} = (\mathbf{I} - \widehat{\mathbf{W}})/\alpha$ is known as the *un-normalized* graph Laplacian [7].

So what does all this tell us about how approximating the filter distorts the output image? As (16) makes clear, the distortion is concentrated in the higher-frequency components of the output. Furthermore, the degree of distortion is given pixel-wise by the ratio $d_j/\overline{d}$. The overall distortion is small when the coefficients d_j are tightly concentrated around the mean $\overline{d}$ (see Fig. 1).

4. APPLICATIONS

A linear combination of the normalization-free filters can represent a multiscale image enhancement framework [8] as:

$$\mathbf{z} = \beta_1(\mathbf{I} - \widehat{\mathbf{W}}_1)\mathbf{y} + \beta_2(\widehat{\mathbf{W}}_1 - \widehat{\mathbf{W}}_2)\mathbf{y} + \widehat{\mathbf{W}}_2\mathbf{y} \tag{17}$$

where $\widehat{\mathbf{W}}_1$ and $\widehat{\mathbf{W}}_2$ are filters with different smoothing parameters h_1 and h_2. Coefficients β_i control the behavior of this filtering framework by varying effect of the band-pass and high-pass terms, $(\widehat{\mathbf{W}}_1 - \widehat{\mathbf{W}}_2)\mathbf{y}$ and $(\mathbf{I} - \widehat{\mathbf{W}}_1)\mathbf{y}$, respectively[1]. Several enhancement applications such as detail/tone enhancement, sharpening and edge-aware smoothing are feasible using this proposed filtering paradigm.

Here we select NLM as our baseline kernel (computed in a neighborhood of size 5×5) to showcase applications of (17). Fig. 3 depicts examples of smoothing ($\beta_1 = 0.2$, $\beta_2 = 1.2$) and detail enhancement ($\beta_1 = 1.2$, $\beta_2 = 3$) using the normalization free filter. Another enhancement scenario can be sharpening of degraded images. Fig. 4 shows detail enhancement of the mildly blurred input image. As can be seen, unlike other methods, our algorithm effectively boosts the image details and supresses the existing artifacts. Finally, the proposed filtering method is applied on images corrupted by real noise (see Fig. 5). We compare our results with the edge-preserving filtering scheme of He et al. [9]. For nearly the same running time budget, our approach produces sharper results ($\beta_1 = 0.05$, $\beta_2 = 1.25$) . The experimental results presented in this work are obtained by our C++ implementation, which works at nearly 21 *Mega pixel per seconds* on an Intel Xeon CPU @ 3.5 GHz.

Fig. 3. (Left) Input; (Center) Edge-aware smoothing, and (Right) Tone and detail enhancement.

(a) Input (b) [10] (c) [11] (d) Ours

Fig. 4. Comparing existing detail enhancement methods with our proposed algorithm.

Fig. 5. Denoising example. (Left) Input; (Center) smoothed by [9], and (Right) Ours.

5. CONCLUSION

We presented a conceptually simple method for approximating a class of normalized non-linear filters with ones that avoid pixel-wise normalization. The approximate filters are easy to construct, and surprisingly accurate. We studied the behavior of the approximated filter and showed how it can be controlled with a single parameter for both quality and nearness to the base filter. The approximated filter can be used for various real-time detail manipulation applications.

[1] Equivalently $\mathbf{z} = \mathbf{y} + (\beta_1 - \beta_2)\alpha_1\widehat{\mathcal{L}}_1\mathbf{y} + (\beta_2 - 1)\alpha_2\widehat{\mathcal{L}}_2\mathbf{y}$, which represents a multi-Laplacian interpretation.

6. REFERENCES

[1] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *ACM Transactions on Graphics (TOG)*. ACM, 2002, vol. 21, pp. 257–266. 1

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *ICCV, Bombay, India*, pp. 836–846, January 1998. 1

[3] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1150, October 2002. 1

[4] A. Buades, B Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, vol. 4, no. 2, pp. 490–530, 2005. 1

[5] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *Image Processing, IEEE Transactions on*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006. 1

[6] P. Milanfar, "Symmetrizing smoothing filters," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 263–284, 2013. 2

[7] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, 2013. 4

[8] H. Talebi and P. Milanfar, "Nonlocal image editing," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4460–4473, October 2014. 4

[9] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, June 2013. 4

[10] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Transactions on Image Processing*, vol. 9, pp. 505–510, 2000. 4

[11] R. C. Bilcu and M. Vehvilainen, "Constrained unsharp masking for image enhancement," *Proceedings of 3rd International Conference on Image and Signal Processing*, pp. 10–19, 2008. 4