Edge-Avoiding Wavelets and their Applications

Raanan Fattal*
Hebrew University of Jerusalem, Israel

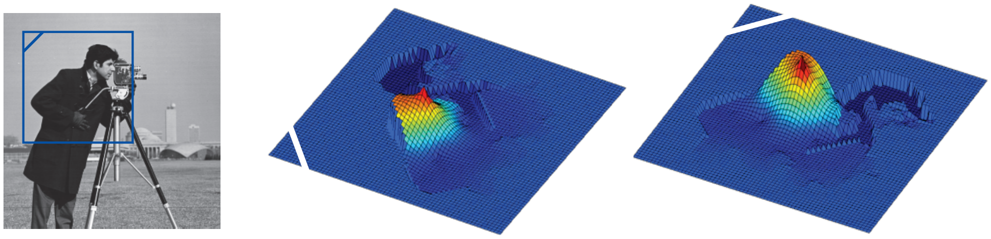

Figure 1: *Two views of the graph of the same edge-avoiding wavelet centered at the shoulder of the Cameraman. The support of the wavelet is confined within the limits set by the strong edges around the upper body.*

Abstract

We propose a new family of second-generation wavelets constructed using a robust data-prediction lifting scheme. The support of these new wavelets is constructed based on the edge content of the image and avoids having pixels from both sides of an edge. Multi-resolution analysis, based on these new *edge-avoiding wavelets*, shows a better decorrelation of the data compared to common linear translation-invariant multi-resolution analyses. The reduced inter-scale correlation allows us to avoid halo artifacts in band-independent multi-scale processing without taking any special precautions. We thus achieve nonlinear data-dependent multi-scale edge-preserving image filtering and processing at computation times which are *linear* in the number of image pixels. The new wavelets encode, in their shape, the smoothness information of the image at every scale. We use this to derive a new edge-aware interpolation scheme that achieves results, previously computed by solving an inhomogeneous Laplace equation, through an *explicit* computation. We thus avoid the difficulties in solving large and poorly-conditioned systems of equations.

We demonstrate the effectiveness of the new wavelet basis for various computational photography applications such as multi-scale dynamic-range compression, edge-preserving smoothing and detail enhancement, and image colorization.

Keywords: wavelets, lifting scheme, data-dependent interpolation, constraint propagation, edge-preserving filtering

1 Introduction

Filtering lies behind almost every operation on images. Explicit linear translation-invariant (LTI) filtering, i.e., convolution, is used extensively in a wide range of applications including noise removal, resolution enhancement and reduction, blurring and sharpening, edge detection, and image compression [Gonzalez and Woods 2001]. Data-dependent filtering with, e.g., the bilateral filter [Tomasi and Manduchi 1998], adjusts the filter stencils at each pixel based on its surrounding. This robust filtering does not average together pixels across edges, thereby avoiding edge-related halo artifacts that plague many image operations that rely on LTI filtering. In addition, switching to such data-dependent filtering requires little or no further algorithmic modifications, making it a very poplar tool in computational photography.

Both the LTI and data-dependent filtering can be used in implicit formulations, where the unknown image appears convolved, allowing to define images through their filtering. The gradient domain [Weiss 2001; Fattal et al. 2002; Pérez et al. 2003] processing, where images are computed from their derivatives, is one popular example for implicit translation-invariant filtering. This approach provides a transparent way of manipulating edges in the image without worrying about the global adjustments involved, yet comes at the cost of solving a Poisson equation. As we discuss in the next section, the inhomogeneous Laplace and Poisson equations can be interpreted as implicit formulations of data-dependent filtering where the requirements over the image derivatives are *weighted* differently in space, based on the input datum. These formulations proved to be useful for edge-aware interpolation of sparse user input and as high-quality edge-preserving smoothing operators, but are costly to compute; they require solving large systems of poorly-conditioned equations.

Multi-resolution analysis (MRA) via wavelet transform [Burrus et al. 1998; Mallat 1999] is widely known as an extremely effective and efficient tool for computing LTI multi-scale decompositions offering a good space/frequency localization trade-off. More specifically, efficient filtering with kernel size proportional to the image dimensions [Burt 1981], detecting edges both in space and scale [Burt and Adelson 1983], bypassing the need for implicit LTI formulations and avoiding the associated costs of solving large linear systems [Li et al. 2005], and preconditioning these systems [Cohen and Masson 1999], can *all* be achieved in *linear-time* computations. In contrast to these results, data-dependent filtering requires performing $\mathcal{O}(N \log N)$ operations in the number of image pixels N since subsampling is avoided [Fattal et al. 2007], solving multiple linear systems [Farbman et al. 2008], coping with the resulting poorly-conditioned systems [Szeliski 2006], or introducing additional dimensions and additional memory cost due to their discretization [Paris and Durand 2006].

In this paper we construct new data-dependent *second-generation wavelets* [Sweldens 1998] based on the edge content of the image. This construction uses ideas from data-dependent filtering, men-

*e-mail: raananf@cs.huji.ac.il

tioned above, to design wavelets with a support that does not contain pixels from both sides of an edge, as shown in Figure 1. Multi-scale decompositions using these new wavelets show a diminished response to strong edges at the transformed coordinates which results in a reduced inter-scale correlation. This allows to avoid halo artifacts in band-independent multi-scale processing, freeing us from taking any special precautions, as in [Li et al. 2005]. The new wavelets encode, in their shape, the edge structure of the image at every scale. We show how this can be used to implement an edge-aware interpolation, normally solved via implicit data-dependent formulations, at the cost of performing two transforms and their inverse. By that we avoid the need to solve numerically-challenging linear systems arising from inhomogeneous Laplace and Poisson equations. Altogether, with this new construction of wavelets we achieve *nonlinear* data-dependent multi-scale decomposition and processing through *explicit* computations running in *linear time* $\mathcal{O}(N)$, and match the performance typical to LTI filtering. We demonstrate the effectiveness of this new MRA on various computational photography applications such as multi-scale dynamic-range compression, edge-preserving smoothing and detail enhancement, and image colorization.

2 Background

Explicit LTI filtering is used in numerous image processing applications, see [Gonzalez and Woods 2001] for a good survey. Implicit formulations define an image through its filtering, e.g., the derivatives, and require solving systems of linear equations, e.g., Poisson equation. This is used for shadow removal [Weiss 2001; Finlayson et al. 2002], dynamic-range compression [Fattal et al. 2002], seamless image editing [Pérez et al. 2003], image completion [Shen et al. 2007], and alpha matting [Sun et al. 2004].

Data-dependent filtering such as the bilateral filter [Tomasi and Manduchi 1998] adjusts the averaging weight of each pixel based on its distance, both in space and intensity, from the center pixel. This operation is nonlinear and does not correspond to filtering in the strict sense of the word, however it serves the same purpose as its LTI counterpart; both operations target the data through a prescribed locality in space and frequency, i.e., they can blur the image or extract its fine-scale detail. Other prototypical methodologies for computing data-dependent filtering include the anisotropic diffusion [Perona and Malik 1990], robust smoothing [Black et al. 1998], and digital total variation [Chan et al. 2001]. In the past two decades or so, these filters became very popular for their ability to smooth an image while keeping its salient edges intact, and became known as *edge-preserving smoothing* filters. One of the main advantages of this property is avoiding the well-known halo artifacts typical to image operations that rely on LTI filtering. Edge-preserving smoothing is used in numerous recent computational photography applications such as smoothing color images [Tomasi and Manduchi 1998], edge-preserving noise removal [Chan et al. 2001; Choudhury and Tumblin 2005], dynamic-range compression [Tumblin and Turk 1999; Durand and Dorsey 2002], flash and no-flash photography [Petschnigg et al. 2004], image editing [Khan et al. 2006], and mesh denoising [Fleishman et al. 2003].

Data-dependent filtering also has an implicit counterpart, the inhomogeneous Laplace and Poisson equations. As shown in [Farbman et al. 2008], the inhomogeneous Poisson equation expresses the steady-state condition of linear anisotropic diffusion process and therefore acts as an edge-preserving smoothing operator. Much like the analogy between LTI filtering and Poisson-based image generation, the inhomogeneous Laplace and Poisson equations compute a least squares solution over *weighted* image derivatives [Farbman et al. 2008] and can therefore be regarded as a weighted filtering of the output image. This is used for regulating deblurring operation of noisy images [Lagendijk et al. 1988], manipulating the detail and contrast of images [Farbman et al. 2008], and regulating estimated transmission in hazy scenes [Fattal 2008]. A similar formalism is used in the image colorization [Levin et al. 2004] and tonal adjustment [Lischinski et al. 2006] methods, where sparse user strokes of color or adjustment parameters are propagated across the image in an edge-aware fashion. This results in a spatially-dependent Laplace equation and used for other applications such as material [Pellacini and Lawrence 2007] and appearance [An and Pellacini 2008] editing, and in [Li et al. 2008] this edge-aware interpolation is boosted via a classification step.

Traditional MRA [Burrus et al. 1998; Mallat 1999] which we briefly review[1] in [Fattal 2009] is, in its essence, a linear translation-invariant filtering. This results from a uniform notion of smoothness throughout space, defined by a *single pair* of scaling and wavelet functions, and reveals itself as the convolution operation in the wavelet transform, see [Fattal 2009]. While this analysis excels in separating weak variations based on their scale, it fails to isolate large-magnitude jumps in the data, such as the ones encountered across edges. As indicated in previous reports [Schlick 1994; Tumblin and Turk 1999; Li et al. 2005; Farbman et al. 2008], strong edges respond to filters at *several* scales thus producing multiple 'reads' in multi-scale decomposition. Processing the different scales independently can easily violate the delicate relationship within this multiplicity and result in haloing and other artifacts around the strong edges in the reconstructed image [Tumblin and Turk 1999]. Avoiding these artifacts, within the framework of LTI multi-scale decompositions, requires taking special precautions when processing the different bands [Li et al. 2005].

Very recently, several multi-scale constructions were proposed in the context of data-dependent image filtering. Paris et al. [2006] exploit the facts that the bilateral filter is an LTI filter in the extended neighborhood, consisting of both space and pixel-intensity range [Barash and Comaniciu 2004], and that linear filtering can be computed efficiently through a multi-level strategy [Burt 1981], to achieve a linear-time implementation of bilateral filtering with arbitrarily large kernels. This comes at a memory cost where additional dimensions (intensity ranges) must be discretized. A multi-scale decomposition, based on the dyadic wavelet transform [Mallat 1999] and bilateral filter, is proposed in [Fattal et al. 2007] and operates in $\mathcal{O}(N \log N)$ time. This decomposition runs the bilateral filter repeatedly and results in oversharpened edges that persist in the coarsest scales, independently of the feature size and may lead to gradient reversals when used for image processing [Farbman et al. 2008]. Farbman et al. [2008] show that the weighted least squares, i.e., inhomogeneous Poisson equation, can be used for computing edge-preserving smoothing at multiple scales. This approach requires solving large numerically-challenging linear systems at each scale. Szeliski [2006] proposes a locally-adapted hierarchical basis for preconditioning this type of linear systems. More recently, Fattal et al. [2009] propose an adaptive edge-based image coarsening for tone-mapping operations. In this approach the image is represented by fewer degrees of freedom than the original number of pixels. While it avoids certain bleeding artifacts, this reduced representation supports a limited number of image operations. For example, it does not provide a scale separation and cannot be used to manipulate image details.

In this paper we construct a new data-dependent MRA by combining robust smoothing with the *lifting scheme* [Sweldens 1995], an efficient second-generation wavelets construction framework which we review next.

[1] In this auxiliary material which accompanies this paper, and available at www.cs.huji.ac.il/˜raananf/projects/eaw/sup_text.pdf, we briefly review a few basic terms in wavelets theory which we mention throughout the paper.

2.1 Second-Generation Wavelets

Wavelets bases that do not consist of translates and dilates of a single pair of scaling and wavelet functions, allowing them to change according to local spatial particularities, exist [Dahmen 1994; Donoho 1994; Schröder and Sweldens 1995; Lounsbery et al. 1997; Sweldens 1998; Claypoole et al. 1998; Secker and Taubman 2003; Peyré and Mallat 2005], and are known as *second-generation wavelets* [Sweldens 1998]. These constructions are used for image compression, dealing with irregular sampling, constructing MRA over complicated geometries, adapting wavelets on finite intervals to the boundary, and refining unstructured meshes. Here, as done for image compression [Claypoole et al. 1998; Secker and Taubman 2003], we construct wavelets that depend on the *content* of the input datum and combine this with ideas borrowed from robust smoothing. The new construction scheme builds on data prediction schemes by Harten [1996] and the wavelet lifting scheme by Sweldens [1995] which we briefly review here.

Lifting Scheme. The lifting scheme is an efficient implementation of the fast wavelet transform and more importantly, it provides a methodology for constructing biorthogonal wavelets through space, without the aid of Fourier transform. This makes it a well-suited framework for constructing second-generation wavelets that adapt to the spatial particularities of the data. In this construction one starts off with some given simple and often translation-invariant biorthogonal basis and performs a sequence of modifications that adapt and improve the wavelets. Sweldens [1995] divides this scheme into three steps: split, predict, and update which we briefly describe here. **Split.** Given the input signal data at the finest level $a^0[n]$, where the superscripts denote the level, the split step consists of formally dividing the data variables $a^0[n]$ into two disjoint sets C and F, which define coarse and fine data points respectively. We denote the signal values restricted to these sets by $a^0_C[n]$ and $a^0_F[n]$ (but keep the same index numbering n). This operation is also known as the *Lazy wavelet* transform and one simple and popular choice is, in 1D, splitting the data into sets of even and odd grid points. **Predict.** Next, we use the coarse data points a^0_C to predict the fine a^0_F. Denote this prediction operator by $\mathcal{P} : C \mapsto F$, and define the prediction error by

$$d^1[n] = a^0_F[n] - \mathcal{P}\big(a^0_C\big)[n], \tag{1}$$

at every $n \in F$. This abstract form obscures the fact that the coarse and fine variables are intermixed in space and every fine variable $a^0_F[n]$ has a few neighboring variables within a^0_C that are relevant for its prediction, assuming of course local correlation in the image. The prediction errors $d^1[n], n \in F$ are the wavelet or detail coefficients of the wavelet transform of the next level, see [Fattal 2009]. **Update.** The coarse variables $a^j_C[n]$ are usually not taken as the next-level approximation coefficients, since this would correspond to a naive subsampling of the original data and may suffer severe aliasing. Typically, the lifting scheme makes sure that the overall sum of the approximation coefficients $\sum_n a^j[n]$ is preserved at all levels. This is achieved by an additional update operator $\mathcal{U} : F \mapsto C$ that introduces averaging with the fine variables a^0_F through d^1, i.e,

$$a^1[n] = a^0_C[n] + \mathcal{U}(d^1)[n], \tag{2}$$

at every $n \in C$. These new variables $a^1[n]$ are the approximation coefficients of the next level of the wavelet transform, which is now complete. As in the traditional fast wavelet transform, the following levels are computed recursively by repeating these three steps over the approximation coefficients $a^j[n], j \geq 1$. It is easy to see that by applying these steps in the reverse order and replacing additions with subtractions and vice verse, the prefect-reconstructing inverse transformation is obtained.

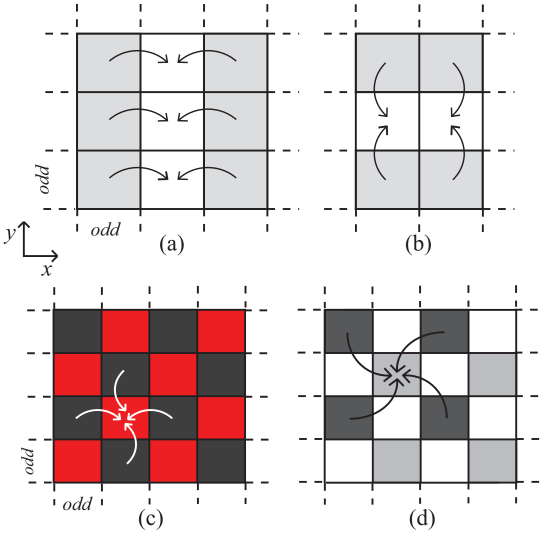

Figure 2: *Wavelets prediction schemes. (a) Illustrates the prediction along the x-axis of the separable CDF wavelets, and (b) along the y-axis. (c) Illustrates how red pixels are predicted from the black pixels at the first step of the weighted red-black wavelet construction, and (d) the second step where the light-gray pixels are predicted from the dark-gray ones which correspond to the locations of the next-level approximation variables. The update steps (not shown) correspond to reverse arrows where coarse variables are updated from the detail variables located at the fine pixels.*

To make this construction more concrete, we will describe a particular example, taken from [Sweldens 1995], which we use as a starting point for deriving our new scheme in the next section. Consider a 1D image signal $a^0[n]$ and the splitting step that takes the odd-indexed pixels as the coarse variables and the even-indexed as the fine. Every even-indexed pixel is predicted by its two odd-indexed neighbors using a simple linear interpolation formula

$$\mathcal{P}\big(a^0_C\big)[n] = \big(a^0_C[n-1] + a^0_C[n+1]\big)/2, \tag{3}$$

for every $n \in F$ (even), thus $n-1, n+1 \in C$ (odd). Next, by choosing the following update operator

$$\mathcal{U}\big(d^1\big)[n] = \big(d^1[n-1] + d^1[n+1]\big)/4, \tag{4}$$

defined over $n \in C$, the approximation average is preserved throughout the different levels (see [Sweldens 1995]). This construction corresponds to the well-known (2,2)-Cohen-Daubechies-Feauveau (CDF) biorthogonal wavelets [Cohen et al. 1992] where both the primal and dual wavelets have two vanishing moments.

3 New Construction

Unlike many existing constructions of second generation wavelets that depend on the irregularities and inhomogeneities of the domain and similarly to recent image compression schemes [Claypoole et al. 1998; Secker and Taubman 2003; Chan and Zhou 2003], we construct the scaling and wavelet functions based on the content of the input data. The main idea behind our new approach is motivated by robust smoothing [Perona and Malik 1990; Tomasi and Manduchi 1998; Black et al. 1998] and consists of mixing pixels depending continuously on their similarity. This is implemented in the context of lifting by defining robust prediction operators that weigh pixels based on their similarity to the predicted pixel. Thus, instead of using data-independent regression formulae, we use posteriori influence functions based on the similarity between the predicted pixel and its neighboring coarse variables. More specifically, we use the edge-stopping function from [Lischinski et al. 2006] to define the following prediction weights

$$w^j_n[m] = \left(\left|a^j[n] - a^j[m]\right|^\alpha + \varepsilon\right)^{-1}, \tag{5}$$

where α is between 0.8 and 1.2 and $\varepsilon = 10^{-5}$, for images with pixels values ranging from zero to one. We use these weights to derive two different two-dimensional wavelet constructions.

Weighted CDF Wavelets (WCDF). Here we derive a two-dimensional weighted prediction based on the (2,2)-CDF wavelet transform, applied along each axis separately. Instead of using an even average of the two coarse variables, we define the following robust average

$$\mathcal{P}\left(a^j_C\right)[x,y] = \frac{w^j_{x,y}[x–1,y]a^j_C[x–1,y] + w^j_{x,y}[x+1,y]a^j_C[x+1,y]}{w^j_{x,y}[x–1,y] + w^j_{x,y}[x+1,y]}, \tag{6}$$

where, like in the 1D case and as shown in Figure 2, $F = \{(x,y)|x\,\text{even}\}$ (white cells) and $C = \{(x,y)|x\,\text{odd}\}$ (gray cells). The next-level detail coefficients are computed according to (1), i.e., $d^{j+1} = a^j_F - \mathcal{P}(a^j_C)$, at the fine points in F. The update operator $\mathcal{U}$ is designed to smooth the next level approximation variables a^{j+1}_C when possible and, once again, we use a robust smoothing to define

$$\mathcal{U}\left(d^{j+1}\right)[x,y] = \frac{w^j_{x,y}[x–1,y]d^{j+1}[x–1,y] + w^j_{x,y}[x+1,y]d^{j+1}[x+1,y]}{2\left(w^j_{x,y}[x–1,y] + w^j_{x,y}[x+1,y]\right)}, \tag{7}$$

where $(x,y) \in C$. The next-level approximation coefficients are computed according to (2), i.e., $a^{j+1} = a^j_C + \mathcal{U}(d^{j+1})$, at the coarse points in C. The analog steps are repeated along the y-image axis. Note that uniform weights, obtained by $\alpha = 0$ in (5), produce a separable two-dimensional (2,2)-CDF wavelet transform.

Weighted Red-Black Wavelets (WRB). Here we construct non-separable wavelets of a lower anisotropy based on the data-independent LTI construction by Uytterhoeven et al. [1999]. This is also a two-step construction that uses the red-black quincunx lattice. At the first step we predict each red pixel by the following weighted averages of its four nearest black pixels

$$\mathcal{P}_{red}\left(a^j_C\right)[x,y] = \frac{\sum_{x',y' \in N_{x,y}} w^j_{x,y}[x',y']a^j_C[x',y']}{\sum_{x',y' \in N_{x,y}} w^j_{x,y}[x',y']}, \tag{8}$$

where $C = \{(x,y)|x{+}y\,\text{even}\}$ (red pixels), $F = \{(x,y)|x{+}y\,\text{odd}\}$ (black pixels) and $N_{x,y} = \{(x{+}1,y), (x{–}1,y), (x,y{–}1), (x,y{+}1)\}$. As before, this operator is used in (1) to compute d^{j+1} in F. The update operator is also defined by averaging the four nearest fine points of every coarse point

$$\mathcal{U}_{red}\left(d^{j+1}\right)[x,y] = \frac{\sum_{x',y' \in N_{x,y}} w^j_{x,y}[x',y']d^{j+1}[x',y']}{2\sum_{x',y' \in N_{x,y}} w^j_{x,y}[x',y']}, \tag{9}$$

at every $(x,y) \in C$ and as before, the approximation coefficients are computed using this operator in (2) over pixels in C.

In the second step, illustrated in Figure 2, we predict the light-gray variables of even coordinates based on their four diagonally-nearest dark-gray neighbors of odd coordinates. The prediction formula for $\mathcal{P}_{black}(a^j_{C'})[x,y]$ is the same as (8) with $C' = \{(x,y)|x + y\text{ even, and } x,y\text{ odd}\}$ (dark-gray pixels), $F' = \{(x,y)|x + y\text{ even, and } x,y\text{ even}\}$ (light-gray pixels), and $N_{x,y} = \{(x+1,y+1), (x-1,y+1), (x+1,y-1), (x-1,y-1)\}$. Based on this prediction we compute $d^{j+1} = a^j_{F'} - \mathcal{P}(a^j_{C'})$ at every pixel in F'. The update operator $\mathcal{U}_{black}$ averages at every pixel C' using its four diagonally nearest d^{j+1} in F' according to (9) and is used to compute the final next-level approximation coefficients a^{j+1} according to (2). Note that uniform weights, obtained by $\alpha = 0$ in (5), recover the red-black wavelet transformation of [Uytterhoeven et al. 1999].

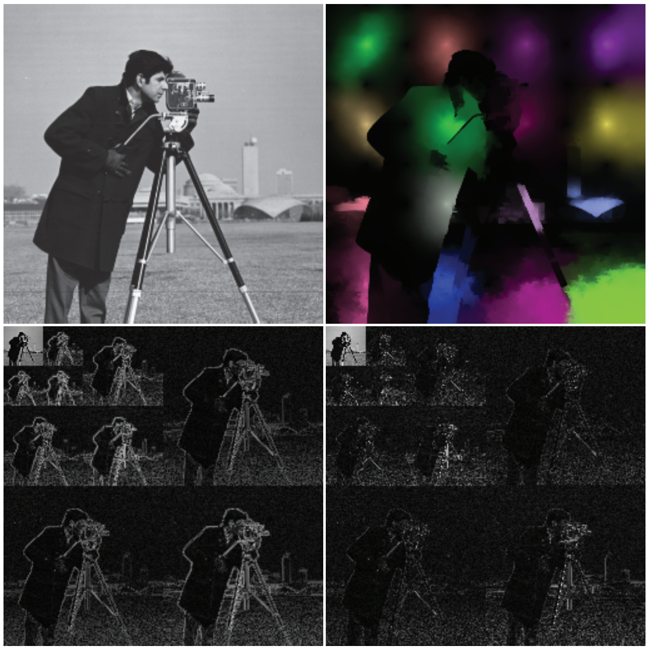

Figure 3: *(top-left) Input image, (top-right) a few dual WRB scaling functions, from the third level ($j = 3$), are shown with unique colors, (bottom-left) translation-invariant red-black wavelet transform, and (bottom-right) weighted red-black wavelet transform (computed with $\alpha = 0.8$).*

In both of these constructions the boundary is treated, typically to the lifting scheme, quite easily. Both the prediction and update operators are simply restricted to variables within the domain and formulae (6), (7), (8) and (9) remain properly normalized. We should note that although we started with schemes that preserve the approximation average at all scales, this property is lost once weighted averaging is introduced. However, constant images produce uniform weights which reproduce the original (2,2)-CDF and red-black schemes and thus preserve the approximation averages at all scales. When transforming general one-megapixel images we observe that the fluctuations in the approximation average are below 4%. The weighted schemes, we defined, can be executed as fast as their original translation-invariant version using precomputed tables for evaluating the power function in (5).

Discussion. In Figure 3 we show a few coarse-level (dual) scaling functions obtained by the WRB construction. These functions are smooth where the input image is flat, resembling the original profiles of the LTI red-black scaling function of [Uytterhoeven et al. 1999]. Near strong edges, however, they terminate abruptly, depending on α in (5), and their support avoids both the edge and the pixels past it. Since the wavelets are made of a linear combination of scaling functions, see [Fattal 2009], they inherit these structural properties, and as shown in Figure 1 they are also confined to only one side of an edge. For this reason, we call these new wavelets

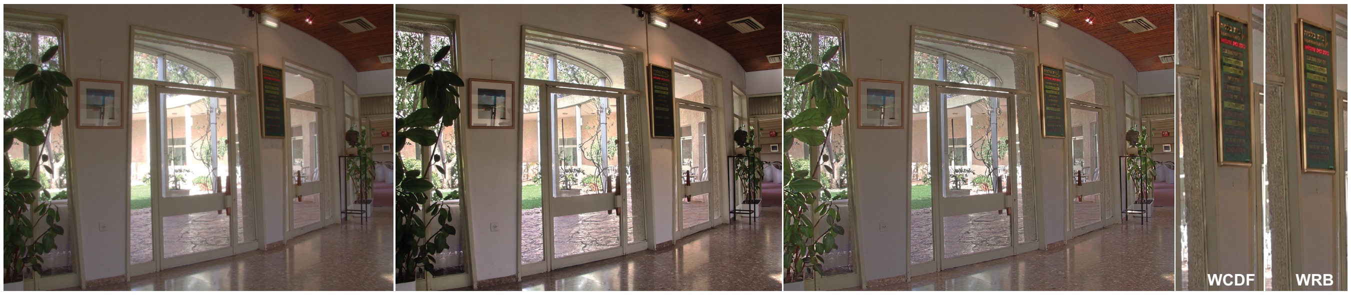

Figure 4: *Left to right; (first) the result produced by Durand and Dorsey [2002], (second) by Farbman et al. [2008], (third) is our result using WRB wavelets (with $\alpha = 0.8$, $\beta = 0.15$, and $\gamma = 0.7$), and (fourth) a comparison between the WCDF and WRB.*

edge-avoiding wavelets (EAW). The decoupling between scaling functions across edges allows representing such discontinuities in the data using the scaling functions or, in other words, such discontinuous functions are well represented by the approximation spaces of the MRA, see [Fattal 2009]. As a consequence, the wavelets response to edges diminishes and, with it, the inter-scale correlation of the detail coefficients. The latter, as we discussed earlier, is a major source of haloing effects in LTI-based operations.

While we will show how this decorrelation or compaction of edge response is useful for image manipulation, one must realize that our scheme is useless for image compression. The 'separation data,' i.e., the different shapes of the edge-avoiding scaling and wavelet functions, are encoded in the averaging weights used in the prediction and update formulae (or alternatively, the spatially-changing conjugate mirror filters, see [Fattal 2009]). This data must be stored, along with the detail and approximation coefficients, during the forward transform in order to compute its inverse. Since these weights are between zero and one, we tested storing them using only 8 bits and observed no visual degradation (the transformed coefficients are stored with floating-point precision). Image compression methods that use the lifting scheme [Claypoole et al. 1998; Secker and Taubman 2003; Chan and Zhou 2003] use fewer bits for encoding this information (on top of quantizing the transformed coefficients) and achieve data compression at the cost of the accuracy at which edges are approximated by the MRA approximation spaces—a critical requirement for obtaining high-quality processing and avoiding halos.

The work by Donoho [1994] aims to achieve a similar goal as our construction—finding a multi-scale representation with low inter-scale correlation. In his work, images are segmented into regions which are analyzed through an independent multi-scale average-interpolation scheme. Chan et al. [2003] use the lifting scheme together with a binary edge detection, based on analytical measures, for both image denoising and compression. In our construction regions, separated by an edge, are not entirely isolated; this is done *continuously* depending on the edge magnitude and spread. This is preferable for image processing since segmentation-based manipulations are known to produce gradient reversals artifacts [Farbman et al. 2008]. As we shall show in the next section, this makes our construction also useful for edge-aware interpolation where, up to a certain extent, data *needs* to propagate through edges.

4 Applications

We implemented both the WCDF and WRB wavelet transformations and the applications we are about to report in C++ and give their running time on a 3.0GHz Intel Core 2 Duo machine. Our main goal here is to demonstrate how various computational-photography applications can be implemented using the new EAW decomposition in a natural and conceptually-simple way, verify the quality of the results against state-of-the-art methods on previously-tested images, and measure the gain in computational performance.

4.1 Dynamic-Range Compression

High-dynamic range imaging became popular in recent years and digital cameras are producing more significant bits per pixel than before. Displaying and printing these images on conventional media requires reducing the range of their luma. This is the subject of many recent works in the field, including [Tumblin and Turk 1999; Durand and Dorsey 2002; Fattal et al. 2002; Li et al. 2005; Lischinski et al. 2006; Farbman et al. 2008].

Using our new EAW decomposition we can achieve detail-preserving dynamic-range compression by 'flattening' the approximation coefficients a^J at the coarsest level J as well as decrease progressively the detail coefficients d^j; more at the coarse scales and less at the fine scales. More specifically, we switch to the YUV color space, and operate on the logarithm of the luma channel, $Y(x, y)$. Given its EAW decomposition, $\log Y \mapsto a^J, \{d^j\}_{j=1}^J$, we compute a dynamically-compressed luma component $Y'(x, y)$, simply by scaling the different components before reconstruction as follows,

$$\beta a^J, \left\{\gamma^j d^j\right\}_{j=1}^J \mapsto \log Y'(x, y), \qquad (10)$$

where $\beta \leq 1$ and $\gamma \leq 1$ are the parameters controlling the amount of compression and $\mapsto$ refers to the forward and backward EAW transform.

In figures 4 and 6 we compare our results to the ones obtained by current state-of-the-art methods [Durand and Dorsey 2002] and [Farbman et al. 2008]. This comparison shows that the image quality produced by the EAW-based compression does not fall much below the two alternatives. At this strong level of compression (higher than what is applied by the other two) we observe in Figure 4 a weak darkening at the top-left door bar and some mild ripples at the ceiling, and in Figure 6 there are some visible irregularities at the dome. Using our implementation, it takes 12 milliseconds to compress the dynamic range of one-megapixel image. This is more than 200 times faster than [Farbman et al. 2008] which requires 3.5 seconds on a 2.2GHz machine and about five times faster than [Durand and Dorsey 2002] which is a single-scale compression and known to be limited to only fine scales [Farbman et al. 2008]. In this figure we also compare the WCDF and WRB, and see that the WCDF shows some weak ripple along the vertical edges of the door which are much weaker in the result produced by WRB. Figure 5 shows that even without taking any special precautions during compression in (10), such as smoothing the gain control Li et al. [2005] do, the EAW-based compression produce halo-free images.

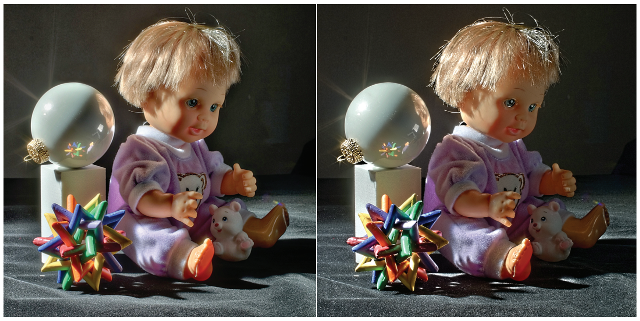

Figure 5: *(left) Result produced by Li et al. [2005], and (right) is our result using WRB wavelets (with $\alpha = 0.8$, $\beta = 0.11$, and $\gamma = 0.68$).*

4.2 Multi-Scale Detail Enhancement and Edge-Preserving Smoothing

Detail enhancement and smoothing are basic and common image processing operations, available in most image editing software. Recently, several new approaches were proposed for performing these operations while keeping the strong edges in tact and thus avoiding halos and gradient-reversal artifacts. Here we show that these operations can be computed in a simple and efficient way using our new MRA which produce results that meet the quality standards set by latest state-of-the-art methods.

Both the smoothing and enhancement are achieved by an operation similar to the one we used to compress the dynamic range. We transform the the logarithm of the luma component of the input image and reconstruct the image according to (10) with $\beta = 1$ and $\{\gamma^j\}_{j=1}^{J}$ determined by a cubic polynomial $p(j)$. This polynomial interpolates the amount of enhancement we want at the finest-scale $p(1)$, the mid-scale $p(J/2)$, and the coarsest-scale $p(J)$ which are values set by the user. Smoothing is achieved by setting a small $p(1)$ whereas enhancing fine-scale detail is achieved with a large $p(1)$ and a mid-scale enhancement is obtained with a large $p(J/2)$.

In Figure 7, we compare our method with [Fattal et al. 2007] and [Farbman et al. 2008]. This test shows that our output is halo-free with an increased fine-scale detail and does not show the gradient-reversals seen in the result of [Fattal et al. 2007]. Figure 8 shows edge-preserving smoothing at two different scales as well as another comparison with Farbman et al. [2008] where we boost details at two different scales. While the resulting image quality is comparable our method is much faster to compute, even compared to Fattal et al. [2007] which computes every scale, of a one-megapixel image, in 15 milliseconds. One-megapixel image has about 7-8 relevant scales, meaning there is about one order of magnitude time factor between the multi-scale bilateral filter, which does not subsample the coarse levels, and our $\mathcal{O}(N)$ EAW transform that computes *all* the scales in 12 milliseconds.

4.3 Edge-Aware Interpolation

Optimization-based edge-aware interpolation became a very popular tool in computational photography [Levin et al. 2004; Lischinski et al. 2006; Pellacini and Lawrence 2007; An and Pellacini 2008; Li et al. 2008]. Typically it requires solving inhomogeneous Laplace equations with the user input defining non-trivial boundary conditions. These matrices are large (N-by-N) and poorly-conditioned due to the presence of weak and strong inter-pixel connections and require preconditioning [Szeliski 2006]. Here we propose to use EAW for edge-aware interpolation *without* the need to solve these numerically-challenging large linear systems altogether.

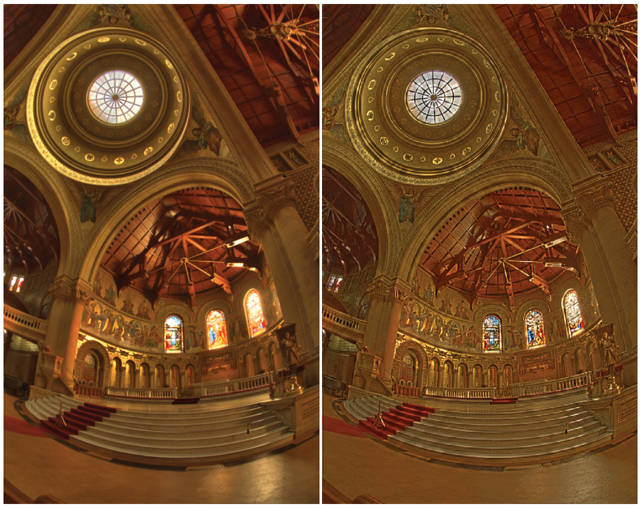

Figure 6: *(left) Result produced by Durand and Dorsey [2002], and (right) is our result using WRB wavelets (with $\alpha = 0.8$, $\beta = 0.125$, and $\gamma = 0.6$).*

We use ideas from [Gortler et al. 1993] and [Gortler et al. 1996] to derive a pull-push mechanism using the EAW. This interpolation scheme is designed to spread a sparse set of data points, e.g., sparse user scribbles, over the entire domain through a smoothing procedure. It consists of two steps: a first step where data is 'pulled' into fewer points by computing the approximation of the input image $I(x,y)$ in a coarser space, and a second step in which the data is 'pushed' back to original image grid by evaluating this approximation at this fine resolution. In the context of wavelet transforms these steps have natural analog counterparts; the pull step corresponds to computing the approximation coefficients $\tilde{a}^j[x,y]$ which is the low-passed downsampled image computed by the forward transform, and the push step corresponds to the inverse transform without adding detail, see [Fattal 2009]. Note that the detail coefficients are *not* needed in this process. We use the *dual* scaling function rather than the primal one when computing the forward transform (hence the tilde over the approximation coefficients $\tilde{a}^j[x,y]$) for reasons we explain below. Analogously to what Gortler et al. [1993] do, we perform this process at every scale $1 \le j \le J$, and blend the results weighted by 2^{-j}. This factor is meant to prefer information coming from shorter distances over longer ones; allowing closer pixels to have a stronger effect than farther ones. Mathematically, this operation is given by

$$\bar{I}(x,y) = \sum_{j=1}^{J} 2^{-j} \underbrace{\sum_{\tilde{\phi}\in\tilde{V}^j} \tilde{\phi}(x,y) \overbrace{\left\langle I(x,y), \tilde{\phi}(x,y)\right\rangle}^{\text{Fwd. Trans. } \tilde{a}^j[x,y]}}_{\text{Backward Transform}}, \qquad (11)$$

where $\langle\cdot,\cdot\rangle$ denotes the dot-product, the summation runs over all dual scaling functions $\tilde{\phi}$ belonging to the dual approximation spaces $\tilde{V}^j$ at every scale j. Readers unfamiliar with these terms are referred to [Burrus et al. 1998; Mallat 1999] and [Fattal 2009]. We use the dual scaling function $\tilde{\phi}$ rather than the primal one, ϕ, in the forward transform, since both in the WCDF and WRB constructions the dual scaling functions are non-negative (while ϕ do have negative values). This ensures a positive interpolation which is a property required by most applications needing edge-aware interpolation. In the Appendix Section we derive the weights for computing the approximation coefficients $\tilde{a}^j[x,y]$. Finally, we provide

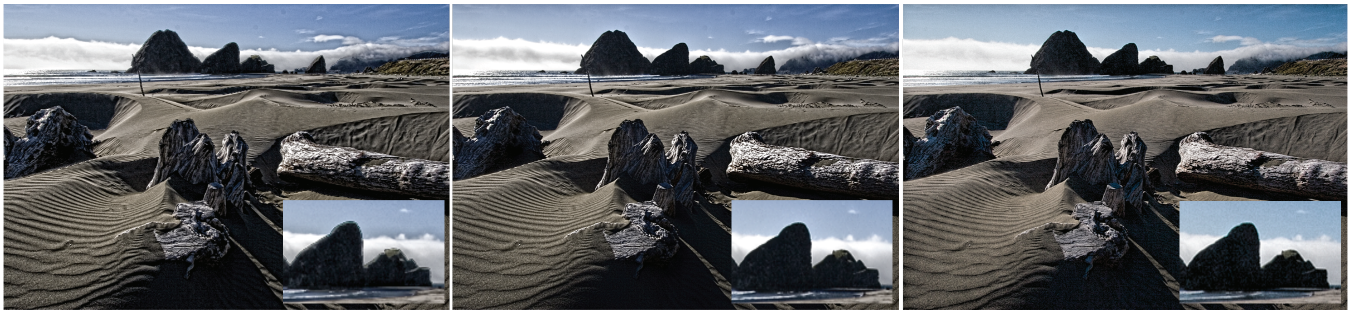

Figure 7: *(left) Results by the multi-scale bilateral filter Fattal et al. [2007], (middle) Farbman et al. [2008], and (right) using the WRB edge-avoiding wavelets (with $\alpha = 1$). The mountain blow-ups show the artifacts due to gradient-reversals in the multi-scale bilateral filter.*

a fast-cascade algorithm for computing $\bar{I}$ given I in the following lines of pseudo-code:

```
set $\tilde{a}^0 = I$
for $j = 1$ to $J$
    compute the forward transform, $\tilde{a}^j$ from $\tilde{a}^{j-1}$, and *keep* $\tilde{a}^{j-1}$ stored
    (according to equation (13) or (16) at the Appendix Section)
end
for $j = J$ to 1
    compute a one-level inverse transform of $\tilde{a}^j$ into $\tilde{t}$ (which belongs to level $j$–1)
    set $\tilde{a}^{j-1} = \tilde{a}^{j-1} + \tilde{t}/2$
end
set $\bar{I} = \tilde{a}^0$
```

The fact that we use the edge-avoiding scaling functions to perform this operation yields an edge-aware interpolation; edge-aware interpolation means that pixels, not separated by an edge, should exchange information and vice verse. In the EAW construction pixels that are not separated by an edge belong, at some scale, to the support of the same scaling function. Thus, the dot-product with this scaling function will average the values of these pixels. Later, at the inverse transform, when this scaling function is linearly combined at the image-resolution, this averaged value reaches all these pixels (its support). Pixels separated by an edge, on the other hand, do not belong to the support of the same scaling function at any scale and hence do not communicate in this process.

Ultimately, $\bar{I}$ corresponds to a blurred version of I where non-zero scribbles are spread acroses the domain, in an edge-aware fashion, to arbitrary distances proportional to 2^J. Next, we explain how this operation is used for computing edge-aware interpolations which are similar to ones computed via inhomogeneous Laplace equations and demonstrate its effectiveness with image colorization [Levin et al. 2004].

Image Colorization. Given an input grayscale image $Y(x, y)$ and color user-stroke images $U(x, y)$ and $V(x, y)$, all given in the YUV color coordinates, we wish to interpolate the color information based on Y's edge content. We do this by introducing a normalization function $N(x, y)$ which is defined to be one at pixels where user-scribbles are provided and zero otherwise. This function allows us to keep track of how much color propagates to each pixel in the image and normalize it accordingly. The image colorization procedure using EAW is described by the following three steps:

1. Given Y, we compute the robust averaging weights of WCDF or WRB wavelets which basically define the edge-avoiding scaling functions. This is done by running the forward transform described at the Appendix and storing all the weights. The resulting approximation coefficients of Y are discarded (the detail coefficients are not computed).

2. Given these weights, computed from Y, we compute $\bar{U}, \bar{V}$, and $\bar{N}$ from U, V, and N, as described in the pseudo-code lines above.

3. Compute the normalized output color components $\hat{U}(x, y) = \bar{U}(x, y)/\bar{N}(x, y)$ and $\hat{V}(x, y) = \bar{V}(x, y)/\bar{N}(x, y)$, which define the output image.

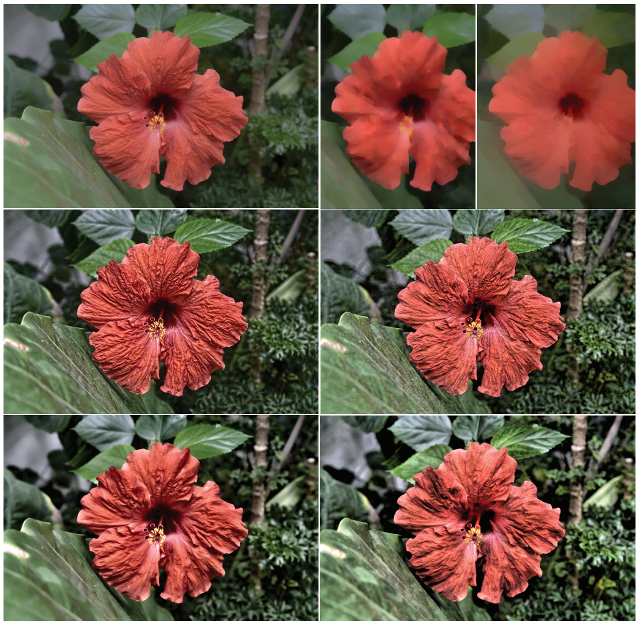

Figure 8: *(top-left) Input image, (top-right) edge-preserving smoothing using WRB wavelets at two different scales, (middle- and bottom-left) edge-preserving detail enhancement at two scales by Farbman et al. [2008], and (middle- and bottom-right) the results by our WRB edge-avoiding wavelets (with $\alpha = 1$).*

The pointwise division, appearing at the third step, ensures an overall unit color contribution at every pixel in the image. This is a standard step used in many radial basis based scattered-data interpolation schemes, e.g., [Shepard 1968]. This construction is positive for positive input data and preserves its minima and maxima.

In compliance with [Levin et al. 2004], we do not use the weights defined by (5) for this application, but rather the following exponential weights,

$$w_n^j[m] = e^{-\left(a^j[n] - a^j[m]\right)^2/\sigma^2}, \qquad (12)$$

and set $\sigma^2 = 15$.

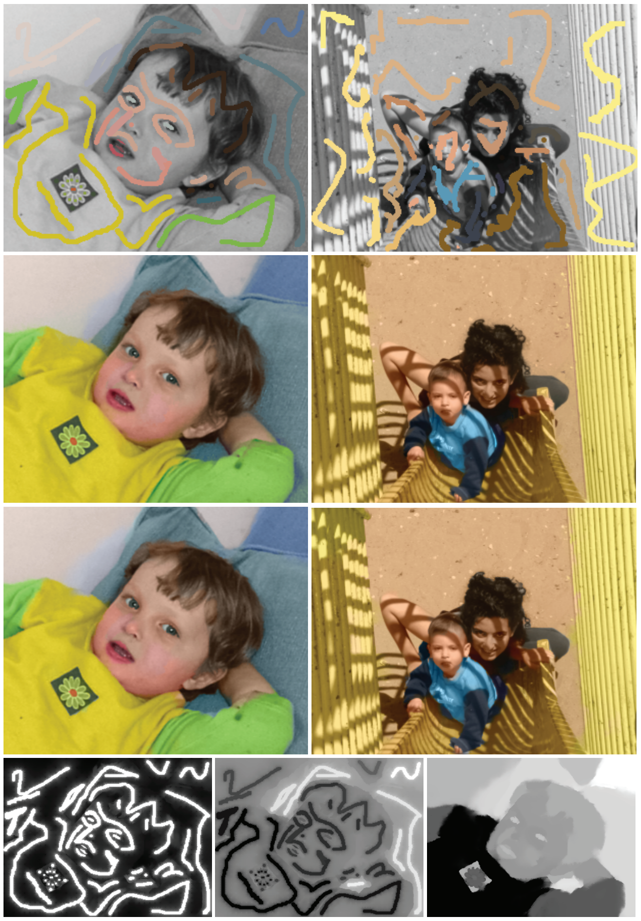

Figure 9: *(top-row) Input images and user color strokes, (second row) results produced by Levin et al. [2004], and (third row) is our results using WRB wavelets (with $\sigma^2 = 15$). At the bottom row we show (left-to-right) $\bar{N}, \bar{U}$, and the resulting $\hat{U}$.*

In Figure 9 we show the results obtained by this scheme next to the ones computed by Levin et al. [2004] by solving an inhomogeneous Laplace equation. The images are surprisingly similar to one another with mean-squared errors below 4×10^{-4} (for image values ranging between zero and one). Since there are about four transforms involved, it takes our scheme 50 milliseconds to colorize a one-megapixel image. Solving an inhomogeneous Laplace equation, as done in in [Levin et al. 2004], using the preconditioning proposed by Szeliski [2006], takes about 2 seconds for setting up the matrices and another 1.5 seconds for performing each preconditioned conjugate-gradient iteration. Szeliski showed that one iteration is enough to obtain reasonable results, but this still involves almost two orders of magnitude time factor compared to what is required here. Yatiz et al. [2006] propose to solve this problem using efficient geodesic distance computations. The complexity of their approach depends on the number of distinct chrominance values used, on top of the number of image pixels, and their reported running times are comparable to Szeliski's.

In Figure 10 we show a slightly different application, *selective decolorization*, where given an input color image the user indicates regions which she wants to turn grayscale and ones to remain colored. These user strokes define a function $C(x, y)$ which is zero and one respectively and a normalization function $N(x, y)$ which is one wherever the user has clicked and zero otherwise. We compute $\hat{C}$, exactly as we did above, and use it to modulate the chrominance components, i.e., the output image given by $\hat{C}(x, y)U(x, y)$ and $\hat{C}(c, y)V(x, y)$ and the unchanged luma channel $Y(x, y)$.

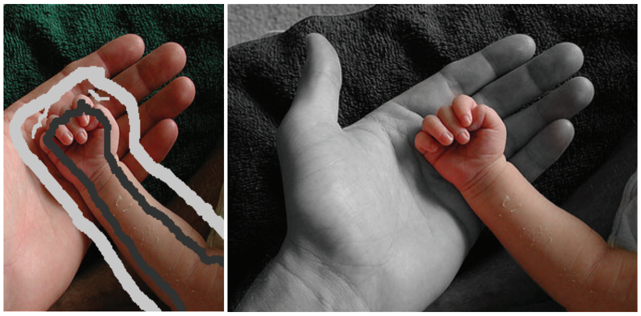

Figure 10: *(left) Input image with color (dark gray) and de-color (bright gray) user strokes and (right) resulting selective decolored image using our WRB wavelets (with $\sigma^2 = 15$). Image courtesy of Eric Jeschke.*

5 Conclusions

In this paper we presented a new family of second-generation wavelets constructed using robust data-prediction lifting schemes. These new wavelets adapt to the edge content of the image and avoid having, in their support, pixels from both sides of an edge. This achieves a multi-scale decomposition which decorrelates data better than the common linear translation-invariant multi-resolution analyses. This nonlinear perfect-reconstructing data-dependent filtering inherits the lifting scheme's fast performance and is computed in linear time, typical to LTI filtering. We showed that this multi-scale representation can be used to speedup various edge-preserving operations as well as manipulate the transformed variables without taking special precautions and result in halo-free images. The fact that the new MRA encodes the image edge structure allowed us to derive an edge-aware interpolation scheme achieving, through fast and explicit computation, results traditionally obtained by implicit formulations requiring sophisticated linear solvers.

The approach we proposed here combines two successful methodologies, robust smoothing and the lifting scheme, to achieve a conceptually simple, new, and natural scheme for performing multi-scale data-dependent analysis and processing. One of the practical benefits of this elementariness is the algorithmic simplicity and straightforward programming required to implement the lifting-based transformation and the processing based on it. The results we presented show that the image quality we obtain, with EAW-based processing, on edge-preserving enhancement and smoothing as well as image colorization is comparable to the ones produced by recent state-of-the-art methods.

Limitations. While EAW can be used to accelerate the dynamic-range compression of images, the quality of the resulting images fall below the current state-of-the-art which contain several artifacts, pointed out in Section 4.1. The EAW are constructed based on a given input image and define an MRA which is *unique* to that image. This has a few limiting implications: (i) the new MRA requires storing additional data, the averaging weights, and is therefore ill-suited for image compression, (ii) the in-place calculations, ordinarily allowed by the lifting scheme, is lost due to the need to compute the averaging weights based on the approximation values, and (iii) the EAWs are tailored to the edges of the input image thus this representation cannot be used for performing any geometric operation, even as simple as copying and pasting pixels.

Aside of dealing with the remaining quality issues, in the future we would like to understand what possibilities a three-dimensional extension of this work opens in video processing. Also, we would like to explore the wide range of EAW that can be constructed and ways to adapt them to specific problems such as alpha matte extraction.

References

An, X., and Pellacini, F. 2008. Appprop: all-pairs appearance-space edit propagation. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 1–9.

Barash, D., and Comaniciu, D. 2004. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing 22*, 73–81.

Black, M. J., Sapiro, G., Marimont, D. H., and Heeger, D. 1998. Robust anisotropic diffusion. *Image Processing, IEEE Transactions on 7*, 3, 421–432.

Burrus, C., Gopinath, R., and Guo, H. 1998. *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice Hall, N.J.

Burt, P. J., and Adelson, E. H. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications COM-31,4*, 532–540.

Burt, P. J. 1981. Fast filter transforms for image processing. Proc. SPIE 2825, vol. 16, 396–408.

Chan, T., and Zhou, H. M. 2003. *ENO-wavelet Transforms and Some Applications in the book Beyond Wavelets*. Academic Press.

Chan, T. F., Osher, S., and Shen, J. 2001. The digital tv filter and nonlinear denoising. *IEEE Trans. Image Process 10*, 231–241.

Choudhury, P., and Tumblin, J. 2005. The trilateral filter for high contrast images and meshes. In *ACM SIGGRAPH Courses*, ACM, New York, NY, USA, 5.

Claypoole, R., Davis, G., Sweldens, W., and Baraniuk, R. 1998. Adaptive wavelet transforms for image coding using lifting. In *Proceedings of Data Compression Conference*, 537–543.

Cohen, A., and Masson, R. 1999. Wavelet methods for second-order elliptic problems, preconditioning, and adaptivity. *SIAM J. Sci. Comput. 21*, 3, 1006–1026.

Cohen, A., Daubechies, I., and Feauveau, J. C. 1992. Biorthogonal bases of compactly supported wavelets. In *Comm. Pure Applied Math.*

Dahmen, W. 1994. Some remarks on multiscale transformations, stability and biorthogonality. In *Wavelets, Images, and Surface Fitting*, Academic Press, 157–188.

Donoho, D. L. 1994. On minimum entropy segmentation. Tech. rep., Preprint:, Departement of Statistics, Standford University, Stanford, CA, 1994.

Durand, F., and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 257–266.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM, New York, NY, USA, vol. 27, 1–10.

Fattal, R., Lischinski, D., and Werman, M. 2002. Gradient domain high dynamic range compression. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 249–256.

Fattal, R., Agrawala, M., and Rusinkiewicz, S. 2007. Multiscale shape and detail enhancement from multi-light image collections. In *ACM TOG*, ACM, New York, NY, USA, 51.

Fattal, R., Carroll, R., and Agrawala, M. 2009. Edge-based image coarsening. *to appear in ACM Trans. Graph.* available at: www.cs.huji.ac.il/˜raananf/projects/bic/m_paper.pdf.

Fattal, R. 2008. Single image dehazing. *In Proc. ACM SIGGRAPH 27*, 3, 1–9.

Fattal, R. 2009. A brief introduction to first- and second-generation wavelets. Tech. rep., Leibniz Center, Hebrew University, URL www.cs.huji.ac.il/˜raananf/projects/eaw/sup_text.pdf.

Finlayson, G. D., Hordley, S. D., Drew, M. S., and Tj, E. N. 2002. Removing shadows from images. In *In ECCV 2002: European Conference on Computer Vision*, 823–836.

Fleishman, S., Drori, I., and Cohen-Or, D. 2003. Bilateral mesh denoising. *ACM Trans. Graph. 22*, 3, 950–953.

Gonzalez, R. C., and Woods, R. E. 2001. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Gortler, S. J., Schröder, P., Cohen, M. F., and Hanrahan, P. 1993. Wavelet radiosity. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 221–230.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. 1996. The lumigraph. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 43–54.

Harten, A. 1996. Multiresolution representation of data: a general framework. *SIAM J. Numer. Anal. 33*, 3, 1205–1256.

Khan, E. A., Reinhard, E., Fleming, R. W., and Bülthoff, H. H. 2006. Image-based material editing. *ACM Trans. Graph. 25*, 3, 654–663.

Lagendijk, R. L., Biemond, J., and Boekee, D. E. 1988. Regularized iterative image restoration with ringing reduction. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on 36*, 12, 1874–1888.

Levin, A., Lischinski, D., and Weiss, Y. 2004. Colorization using optimization. *In Proc. ACM SIGGRAPH 23*, 3, 689–694.

Li, Y., Sharan, L., and Adelson, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *In Proc. ACM SIGGRAPH 24*, 3, 836–844.

Li, Y., Adelson, E. H., and Agarwala, A. 2008. Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. *Comput. Graph. Forum 27*, 4, 1255–1264.

Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. 2006. Interactive local adjustment of tonal values. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 646–653.

Lounsbery, M., DeRose, T. D., and Warren, J. 1997. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph. 16*, 1, 34–73.

Mallat, S. 1999. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, September.

Paris, S., and Durand, F. 2006. A fast approximation of the bilateral filter using a signal processing approach. In *In Proceedings of the European Conference on Computer Vision*, 568–580.

PELLACINI, F., AND LAWRENCE, J. 2007. Appwand: editing measured materials using appearance-driven optimization. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 54.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *In Proc. ACM SIGGRAPH 22*, 3, 313–318.

PERONA, P., AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, 629–639.

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3, 664–672.

PEYRÉ, G., AND MALLAT, S. 2005. Surface compression with geometric bandelets. *ACM Trans. Graph. 24*, 3, 601–608.

SCHLICK, C. 1994. Quantization techniques for visualization of high dynamic range pictures. In *Photorealistic rendering techniques, proc. EGWR*, Springer-Verlag, 7–20.

SCHRÖDER, P., AND SWELDENS, W. 1995. Spherical wavelets: efficiently representing functions on the sphere. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 161–172.

SECKER, A., AND TAUBMAN, D. 2003. Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression. *Image Processing, IEEE Transactions on 12*, 12 (Dec.), 1530–1542.

SHEN, J., JIN, X., ZHOU, C., AND WANG, C. C. L. 2007. Technical section: Gradient based image completion by solving the poisson equation. *Comput. Graph. 31*, 1, 119–126.

SHEPARD, D. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM, New York, NY, USA, 517–524.

SUN, J., JIA, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Poisson matting. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 315–321.

SWELDENS, W. 1995. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In *in Wavelet Applications in Signal and Image Processing III*, 68–79.

SWELDENS, W. 1998. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal. 29*, 2, 511–546.

SZELISKI, R. 2006. Locally adapted hierarchical basis preconditioning. In *In Proc. ACM SIGGRAPH*, ACM, New York, NY, USA, 1135–1143.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, USA, 839.

TUMBLIN, J., AND TURK, G. 1999. Lcis: a boundary hierarchy for detail-preserving contrast reduction. In *In Proc. ACM SIGGRAPH*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 83–90.

UYTTERHOEVEN, G., ROOSE, D., AND BULTHEEL, A. 1999. Integer wavelet transforms using the lifting scheme. World Scientific and Engineering Society Press, 198–200.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. *IEEE International Conference on Computer Vision 2*.

YATZIV, L., AND SAPIRO, G. 2006. Fast image and video colorization using chrominance blending. *Image Processing, IEEE Transactions on 15*, 5, 1120–1129.

Appendix

In Section 4.3 we need to compute the approximation coefficients based on the dual scaling function $\tilde{\phi}$ and not the primal ϕ which is what the forward transform computes. In order to find the interpolation weights of the dual scaling functions we have to compute the contribution of each coarse-level approximation coefficient to each finer-level approximation coefficient. This is equivalent to setting only one $a^j[n] = \delta[n-k]$, for each pixel k at a time, and computing a one-level inverse transform *without* adding the detail component. The inverse transform of the lifting scheme at level j starts with the inverse action of the update step (2), i.e., $a_C^{j-1}[n] = a^j[n] - \mathcal{U}(d^j)[n]$, which becomes $a_C^{j-1}[n] = a^j[n]$ since the detail coefficients are all set to zero. Then, it proceeds with the inverse action of (1), $a_F^0[n] = d^1[n] + \mathcal{P}\left(a_C^0\right)[n]$, which gives $a_F^0[n] = \mathcal{P}\left(a_C^0\right)[n]$ with a null detail component. Altogether, we get that every approximation coefficient in level j contributes one to every matching coarse data point in level $j-1$, i.e., $a_C^{j-1}[n]$ and any fine data point $a_F^{j-1}[n]$ gets a contribution equal to the amount used to predict it.

In Section 4.3 we need to use these weights to define a forward transform, i.e., reduce resolution rather than interpolate it. This means that we need to define new update-like steps that average the approximation coefficients at one level to the next level approximation coefficients. Note that since no detail coefficient is needed in this process, we define these steps directly based on the approximation coefficients rather than the detail coefficients as in (2).

In case of the weighted CDF dual scaling function we get the following update-like step,

$$\tilde{a}^{j+1}[x,y] = \tilde{a}^j[x,y] + \hat{w}^j_{x-1,y}[x,y]\tilde{a}^j[x-1,y] + \hat{w}^j_{x+1,y}[x,y]\tilde{a}^j[x+1,y], \tag{13}$$

at every $(x,y) \in C = \{(x,y) | x \text{ odd}\}$, where

$$\hat{w}^j_{x-1,y}[x,y] = w^j_{x-1,y}[x,y] / (w^j_{x-1,y}[x,y] + w^j_{x-1,y}[x-2,y]), \tag{14}$$

$$\hat{w}^j_{x+1,y}[x,y] = w^j_{x+1,y}[x,y] / (w^j_{x+1,y}[x,y] + w^j_{x+1,y}[x+2,y]). \tag{15}$$

Note that the weights w are computed using the approximation coefficients computed here, i.e., $\tilde{a}^j$ inserted in (5). The analog formula is applied to the y-image axis.

The same consideration for the weighted red-black dual scaling function gives

$$\tilde{a}^{j+1}[x,y] = \tilde{a}^j[x,y] + \sum_{x',y' \in N_{x,y}} \hat{w}^j_{x',y'}[x,y]\tilde{a}^j[x',y'], \tag{16}$$

at every $(x,y) \in C = \{(x,y) | x+y \text{ even}\}$. The neighborhood $N_{x,y}$ consists of the four nearest pixels, as in (8), and

$$\hat{w}^j_{x',y'}[x,y] = w^j_{x',y'}[x,y] \;/ \sum_{x'',y'' \in N_{x',y'}} w^j_{x',y'}[x'',y'']. \tag{17}$$

Also here, the weights w are computed using $\tilde{a}^j$ inserted in (5). And at the second WRB step we get a formula analog to (16) for $(x,y) \in F' = \{(x,y) | x+y \text{ even, and } x,y \text{ odd}\}$ with the diagonal neighborhood $N_{x,y} = \{(x+1,y+1), (x-1,y+1), (x+1,y-1), (x-1,y-1)\}$.

Note that it is easy to see that both (13) and (16) preserve positiveness and thus guarantee a positive interpolation in Section 4.3.