Pacific Graphics 2011
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin
(Guest Editors)

Volume 30 (2011), Number 7

Edge-Optimized À-Trous Wavelets for Local Contrast Enhancement with Robust Denoising

Johannes Hanika[1,2], Holger Dammertz[1,3] and Hendrik Lensch[1]
[1] Ulm University, [2] Weta Digital, [3] RTT AG

Abstract

In this paper we extend the edge-avoiding à-trous wavelet transform for local contrast enhancement while avoiding common artifacts such as halos and gradient reversals. We show that this algorithm is a highly efficient and robust tool for image manipulation based on multi-scale decompositions. It can achieve comparable results to previous high-quality methods while being orders of magnitude faster and simpler to implement. Our method is much more robust than previously known fast methods by avoiding aliasing and ringing which is achieved by introducing a data-adaptive edge weight. Operating on multi-scale, our algorithm can directly include the BayesShrink method for denoising. For moderate noise levels our edge-optimized technique consistently improves separation of signal and noise.

1. Introduction

The use of edge-preserving operators for image manipulation has become a common tool in image processing and computer graphics applications. Example applications are tone mapping [Sch94, DD02a], general image editing [KRFB06] and image fusion [ED04]. These operations are an important part of modern photo manipulation tools. Since real images exhibit features at multiple scales it is of great importance to allow image manipulation in scale space which is enabled by image decomposition in various layers.

In this paper, we introduce edge-optimized à-trous wavelets for this purpose and extend edge-avoiding wavelets by automatic, locally adaptive edge-weights such that energy is pushed into the coarser layers (see Figure 1). This has been demonstrated to reduce halos and ringing artifacts [Fat09, KS10], and we also show it better matches the assumptions of denoising via wavelet shrinkage. Our method is a lot quicker than previous high-quality approaches while giving comparable results.

Fattal et al. [FAR07] present a multi-scale method to combine multi-light images using an approximation of bilateral decomposition using à-trous wavelets. They use one global hand-chosen edge weight that increases quickly for different scales to avoid aliasing, inherently limiting the flexibility of the multi-scale analysis. In [FFLS08], the authors use quadratic optimization least squares with preconditioned conjugate gradient to achieve high quality detail/base layer decomposition results. While robust, their method is quite time-consuming, requiring about 3 seconds per megapixel. A general framework for edge-preserving decomposition based on decimated wavelets was presented by

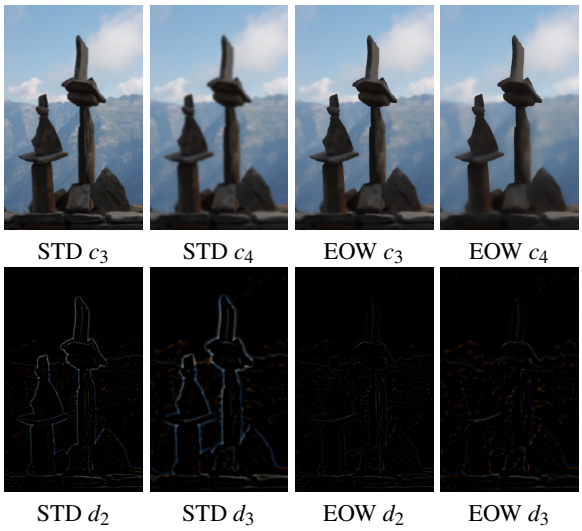

Figure 1: *Coarse images c_i and corresponding detail images d_i for standard à-trous wavelets (STD) and the edge-optimized transform (EOW). Details are multiplied by 3 for better display. Much of the energy from the edges is transferred into the coarse images for the edge-optimized transform, keeping more of the core signal.*

Fattal [Fat09]. In contrast to the à-trous algorithm, decimated wavelets reduce the image resolution at each level. The proposed method is fast ($\approx$ 200 ms per megapixel in our quad-core implementation) but as we show in Section 4 the use of decimated wavelets can lead to severe artifacts due to sparse basis functions in higher levels. Another high-quality method is based on smoothed local histogram filters [KS10]. It allows for quick computation of derivatives and integrals of locally-weighted histograms even with large neighborhoods. Edge-preserving smoothing is supported but it can result in over-sharpening of edges that are visible as gradient reversals when the contrast is enhanced. The authors solve this problem by adding a selective diffusion per decomposition layer. They report a performance of about 1.0 second per megapixel for their GPU implementation. We achieve comparable results in about 0.16 seconds.

The main contributions of our paper are a multi-scale decomposition based on edge-optimized à-trous wavelets that features robust separation in detail coefficients and coarse images, locally adaptive (per-pixel) edge-weights, improved denoising using BayesShrink, and a fast and simple to implement algorithm. The robust detail separation allows us to avoid halos and gradient reversals which are a common problem in contrast enhancement. The optimization of the edge-weights improves the decomposition and keeps more of the important edge information in the coarse layers (see Figure 1). Due to the use of a wavelet transform we can easily incorporate BayesShrink to robustly separate noise from signal (in contrast to a decomposition based on the bilateral filter). In addition, due to the edge-optimized decomposition our method improves the denoising quality for moderate noise levels.The run time performance is about an order of magnitude faster than previous approaches producing similar image quality (see Section 4).

2. Background

Layer decomposition, where the input image is separated into a locally smoothed base layer and a detail layer that contains the deviation to the original, allows for modifying both layers individually enabling fine-grained control in image manipulation, e.g. scaling the detail layer for contrast enhancement and then recombining the modified layers. Simply using linear filters for the decomposition can produce artifacts near edges when altering the detail layer. These artifacts can be reduced by edge preserving non-linear filters like anisotropic diffusion [PM90] or the more commonly used bilateral filter [TM98]. Recombination with the base layer introduces either ringing (when the base layer is too smooth) or gradient reversal (when the base layer is too sharp) [Sch94]. To correctly deal with image features at different spatial extends a multi-scale representation can be achieved by recursive decomposition of the resulting base layer.

Local Contrast Enhancement A high-quality algorithm for local contrast enhancement is presented in [FFLS08]. The authors propose an edge-preserving multi-scale image decomposition where the information of edges is optimized using weighted least squares. This method effectively removes the problem of halos and gradient reversal. Our approach is similar to their approach in that we use a similar error measure. But instead of solving a minimization problem iteratively we embed the problem into the multi-scale edge-avoiding wavelet framework, which gives us a similar level of control while allowing orders of magnitude faster processing.

Bilateral Filter and Edge-Avoiding Wavelets As the bilateral filter (BLF) is is quite expensive to compute for large filter sizes many acceleration methods have been proposed (e.g. [DD02b, PD09, AGDL09, ABD10]). While the bilateral filter is very good at removing noise it is shown in [FFLS08] that applying only one bilateral filter cannot extract features at all scales equally well, and repeated application with extended kernel size to achieve multi-scale is quite computationally expensive.

In [Fat09], decimating edge-avoiding wavelets are introduced. They extend second-generation wavelets [Swe97] by an edge-crossing function corresponding to the range filter in BLF to allow for edge-preserving multi-scale decompositions and modifications.

Undecimated Wavelet Transformation The discrete undecimated wavelet-transform is a useful tool for multi-scale image manipulations but a naive implementation requires the convolution with increasing filter sizes per level. In [Mal89], this problem is removed by the introduction of decimation (downsampling). Fast filter transforms [Bur81] are adapted for the use with wavelets in [HKMMT89, Dut89, Mal98] and significantly speed up the computation of undecimated transforms. The idea of these filters is to contain always a constant number of non-zero coefficients per level. These coefficients are spread by a factor of 2^{level}. This concept is known as the *algorithme à-trous* ("with holes") [She92]. In [FAR07], the à-trous algorithm is used to compute a fast multiscale bilateral decomposition, approximating the bilateral filter but risking aliasing. In [DSHL10], the edge-avoiding à-trous algorithm is used for an efficient tri-lateral image smoothing that uses additional information to influence the edge weights. As our work is based on the edge-avoiding à-trous algorithm we will explain the details in the next section.

2.1. Edge-Avoiding À-Trous

Computing the basic à-trous wavelet transform (without edge weights) is an iterative process where at each scale the next base layer c_{i+1} is obtained by discrete convolution with a Gaussian h_i while the detail layer d_i captures the difference to the original. For the multi-scale analysis the filter size is doubled in each iteration. The core idea of the algorithme à-trous for saving compute time is to keep the samples for evaluating the convolution $*$ constant in each iteration and to increase only the distance between the samples by 2^i in

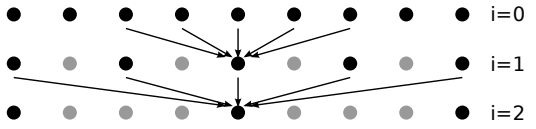

Figure 2: *Visualization of the à-trous wavelet transform process. The first row is the input image. The second row evaluates a convolution with a filter mask h_0 with support of 5 pixels. The third row shows the filling-in of zeros (holes) into the filter mask for the next iteration. Note that this process is undecimated, i.e. it computes a full resolution image at each iteration.*

order to achieve the increasing filter size. The algorithm performs the following steps:

1. Level $i = 0$ starts with the input signal $c_0(p)$
2. $c_{i+1}(p) = c_i(p) * h_i$ (next base layer)
3. $d_i(p) = c_i(p) - c_{i+1}(p)$ (next detail layer)
4. if $i < N : i := i+1$; goto 2
5. $\{d_0, d_1, ..., d_{N-1}, c_N\}$ is the wavelet transform of c.

The reconstruction is then achieved by simply adding the detail coefficients back to the results image c_N from the last iteration

$$c = c_N + \sum_{i=N-1}^{0} d_i. \tag{1}$$

Note that this transformation is undecimated, i.e. each level produces detail coefficients in the original image resolution. The Filter h is based on a B_3 spline interpolation $b = (\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$ [Mur97], approximating a Gaussian, and at each level $i > 0$ the filter doubles its extent by filling in 2^{i-1} zeros between the initial entries

$$\begin{aligned} c_{i+1}(p) &= c_i(p) * h_i \\ &= \sum_{y=-2}^{2} \sum_{x=-2}^{2} b(x)b(y) \cdot c_i\left(p + \begin{pmatrix} 2^i x \\ 2^i y \end{pmatrix}\right). \end{aligned} \tag{2}$$

Thus, the number of non-zero entries remains constant. This process is illustrated in Figure 2.

Extending this algorithm to edge-avoiding à-trous just requires to add a data-dependent weighting function w_{σ_r}, which, as in the bilateral filter, is based on value difference of color or luminance

$$w_{\sigma_r}(p,q) = e^{\frac{\|c_i(p) - c_i(q)\|^2}{\sigma_r}}. \tag{3}$$

The discrete convolution in step 2 becomes

$$c_{i+1}(p) = \frac{1}{k} \sum_{q \in \Omega} h_i(q) \cdot w_{\sigma_r}(p,q) \cdot c_i(p), \tag{4}$$

with Ω denoting the positions of non-zero coefficients in h_i. A simple edge-avoiding à-trous wavelet decomposition step can be calculated very quickly (in 5 ms per megapixel on a GTX480).

3. Fast and Robust Local Contrast

In this section, we describe our edge-optimization algorithm for better local contrast enhancement avoiding halos and gradient reversals. As a wavelet scheme, it consists of decomposition and synthesis. In the first step, the edge weights σ_r are optimized per pixel. Denoising and local contrast enhancement are done during synthesis.

3.1. Decomposition

To better represent the edges in the coarse coefficients c, we will choose locally adaptive filter ranges σ_r. At each layer this is done by computing multiple decompositions with different parameters and then choosing optimal parameters for each pixel. More precisely, for each wavelet scale i, iterate through a list of tentative edge parameters $\sigma_r^j, j = 0...j_{max}$. For each j we separate the current image into coarse $c_{i,j}$ and detail $d_{i,j}$ using the current σ_r^j as global edge weight. The resulting decomposition is used to evaluate an error image

$$e_j = d_{i,j}^2 + \lambda \cdot \|\nabla c_{i,j}\|, \tag{5}$$

which penalizes pushing information to the detail layer (first term) and prefers smooth base images (second term). The step tries to keep as much of the information of the input image in the coarse layer while removing all detail and noise that cannot be represented properly at the next scale. This error measure is similar to the one used by Farbmann et al [FFLS08] for quadratic optimization. In our framework it is sufficient to find the optimal parameter locally at each scale rather than performing a costly global optimization. The mixture parameter λ was fixed at $\lambda = 0.003$ for all images in this paper. Since this error measure will be noisy, we evaluate it over an area of the same size as the à-trous support on this scale, (i.e. $(2^i \cdot 5)^2$) using 25 samples on a regular grid with Cranley Patterson rotation [CP76] to decorrelate adjacent pixels.

After all j have been processed at scale i, we search for the minimum error e_j per pixel and determine the optimal edge weight $\sigma_r^k(p)$ with $k = argmin_j\{e_j\}$. In order to avoid too drastic changes in the edge weights, the per-pixel weights are smoothed once into a buffer z

$$z = \sigma_r^k * h_0. \tag{6}$$

Finally, the actual wavelet decomposition step takes place as in Section 2.1, but taking the locally optimal parameters $\sigma_r(p)$ from the blended buffer z at each pixel's position.

We found $j_{max} = 4$ sufficient for all images in this paper, and chose a simple linear sampling of the interval $[0, 4*(i+1))$. More sophisticated methods such as Metropolis sampling [MRR*53] or simulated annealing [KGV83] might lead to even better results.

3.2. Synthesis

The synthesis step without denoising is straight forward and consists simply of adding up the coarse base layer and the

potentially scaled and soft thresholded details at a given scale to obtain the next finer base layer.

As we are working with wavelets, results from wavelet theory such as BayesShrink [CYV00] directly transfer to our algorithm. BayesShrink is a soft thresholding method designed for a wide range of input signal priors, such as Laplacian and Gaussian distributions assuming additive *iid* Gaussian noise.

As a consistent and robust estimator for the noise standard deviation, we use the median absolute deviation (MAD) over all pixels at the finest level, and obtain

$$\sigma_n = \frac{median(|d_0|)}{0.6745}. \tag{7}$$

Now we iterate for each wavelet scale, this time starting at the coarsest level $i = i_{max}...0$. Given the signal variance $\sigma^2_{y,i}$, the soft shrinkage threshold T is calculated as in [CYV00] to minimize the risk of destroying the signal, by

$$T = \frac{\sigma^2_{n,i}}{\sqrt{\max\{0, \sigma^2_{y,i} - \sigma^2_{n,i}\}}} \tag{8}$$

$$\text{with } \sigma^2_{y,i} = \frac{1}{N}\sum_p d_i(p)^2 \text{ and } \sigma_{n,i} = \sigma_n \cdot 2^{-i}.$$

With this threshold, shrinkage and contrast boost is applied to the detail coefficients at the same time, by setting

$$d'_i = \max\{0, |d_i| - T\} \cdot sign(d_i) \tag{9}$$

$$\text{and } c_{i-1} = c_i + \beta \cdot d'_i, \tag{10}$$

where β is the contrast boost parameter. In case denoising is not wanted, only the last step is relevant with $d' = d$.

Our edge-optimized wavelet scheme tends to move as much edge information to the coarser levels as possible, so the detail coefficients even correspond more to the Gaussian distribution than they would with usual wavelet transformation. Also, the edge data will not be affected by thresholding this way, resulting in a better PSNR of the denoised results.

4. Results

We have implemented the above algorithm in CUDA and perform all our measurements on an NIVIDA GTX480 GPU. We show comparison images with the bilateral filter, the decimated wavelet transform, the high-quality weighted least squares method presented in [FFLS08], and the local contrast enhancement method based on smoothed local histograms and selective edge diffusion [KS10]. In addition, we show results of denoising with BayesShrink.

À-trous wavelets can be used to approximate the behavior of the bilateral filter. This behavior has already been recognized by [FAR07] and is an important difference to the similarly fast (see Table 1) decimated edge-avoiding wavelet transform [Fat09], as this has only very sparse coarse basis functions, which can result in quite pronounced artifacts, see Figure 3. When video animations are processed, this problem is much more apparent due to the sensitivity to translation of the decimated wavelets.

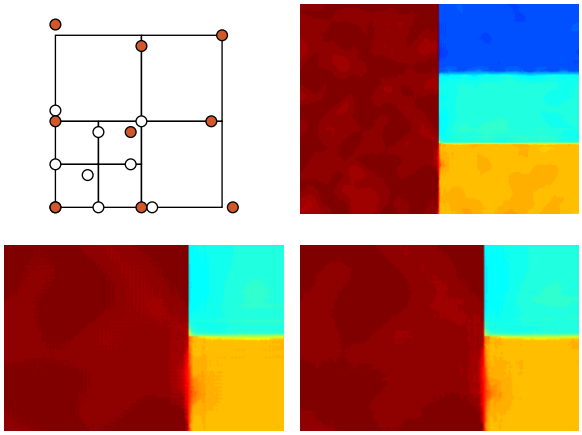

Figure 6: *Subtle aliasing artifacts in the bottom left image (closeup of our result from Figure 5) can be ameliorated by choosing a point set which avoids collisions between different scales, see drawing on top left: white circles indicate the top right quarter of h_{i-1}, and h_i is in red (the whole point set extends to all four quarters symmetrically). The offsets from the regular à-trous sample positions are always just one pixel, so no collision can take place. Top right is the full image created with this point set.*

Figure 4 shows common artifacts when the base layer does not match the edge behavior of the input image well. If an edge in the base layer is smoother than in the input, halos around the enhanced edge can be seen (such as when using standard unsharp masking). On the other hand, if the edge is oversharpened, gradient reversals are the result [FFLS08]. Our approach can avoid both these artifacts in the entire image independent of the strength and scale of an edge by optimizing edge weights σ_r locally on a per-pixel basis.

A comparison of edge-aware smoothing techniques with respect to edges of varying contrast is presented in Figure 5. The synthetic test patch has been taken from [FFLS08] for better comparison with their method. The input image only contains shades of gray, the comparison images have been colorized using a gradient. The weak edge requires pronouncing edge weights while the noisy area inside a patch can only be smoothed over a larger region when the influence of edge weight is reduced. The standard à-trous result (second from left) uses global edge weights and has thus problems with both sharp edges as well as smoothing for example the red region. While the bilateral filter could be applied recursively and with adaptive edge weights, too, it's execution time is so much longer than à-trous wavelets that optimization is hardly feasible in reasonable time. Our result (rightmost) is comparable in quality to the weighted least squares solution but can be evaluated a lot faster (this image has three scales and five candidate σ_r and is computed in 90ms/megapixel, compared to 3500ms/megapixel and scale).

As already stated in [FAR07], edge-avoiding à-trous wavelets can produce ringing artifacts. Compared to the

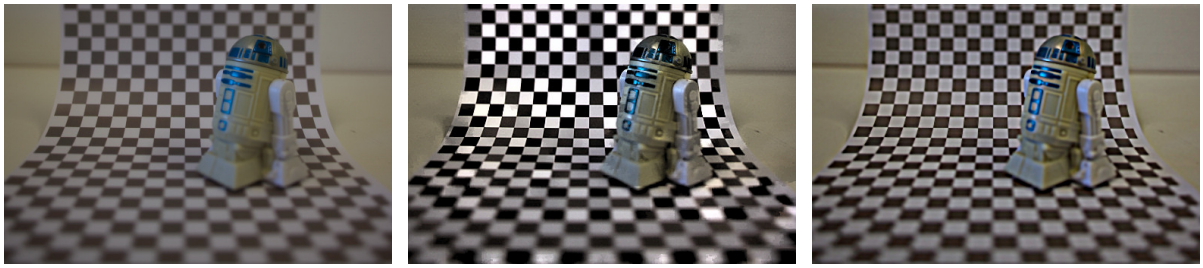

Figure 3: *Failure case for coarse decimated wavelets. Left: input photograph. Middle: decimated edge-avoiding wavelets [Fat09] used for local contrast enhancement. Right: our result. The coarse basis functions of decimated wavelets are just too far apart to capture the checker board. This becomes especially apparent when the input image is translated with respect to the basis function, which results in flickering.*

Figure 4: *Avoiding halos and gradient reversals in contrast enhancement. Selecting a too large σ_r results in halos, a too small σ_r in gradient reversals. Optimized per-pixel edge weights avoid both artifacts.*

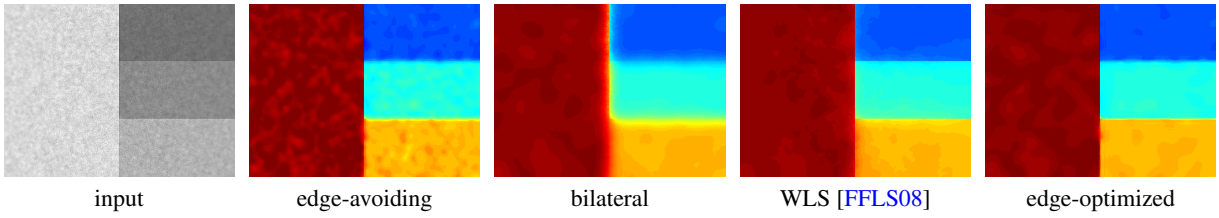

Figure 5: *Importance of adaptive edge weights for edge-preserving smoothing of edges with different contrast. Leftmost is the input image, the result images are actually black and white as well but are visualized using a colored gradient ramp to emphasize the differences. The first image applies edge-avoiding wavelets with one global, very sharp edge weight. Due to its global preset σ_r it fails to remove the high frequency noise and blurs the low contrast edge. The bilateral filter uses a spatial Gaussian with $\sigma_s = 12$ and a range Gaussian with $\sigma_r = 0.45$. The weighted least squares result (WLS) [FFLS08] with $\alpha = 1.8$ and $\lambda = 0.35$ performs significantly better by adapting the edge weights. Optimizing the per-pixel edge weights in our approach yields comparable results with even slightly sharper region separation at only one tenth of the run time.*

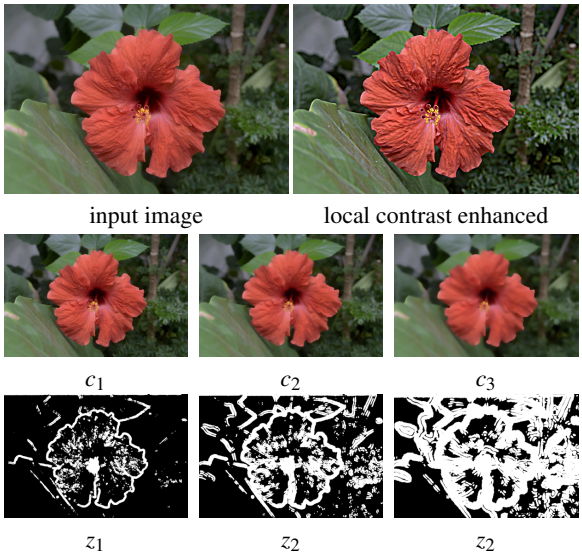

Figure 7: *A flower with local contrast enhancement. The bottom row shows the blended edge weight buffers (black and white, white means smaller σ_r for better display) and the resulting coarse base layers c_i.*

standard à-trous algorithm, edge-avoiding wavelets inherently keep some higher frequencies in the base layer which results in aliasing by the highly undersampled à-trous kernel. Another source for aliasing is the inter-scale dependencies of the sampling pattern: some samples of the enlarged filter h_{i+1} coincide with the ones from h_i. This way, very subtle errors propagate all the way to the coarsest scale and produce objectionable ringing artifacts. While we don't want to spend more samples to avoid this, we can easily reduce the inter-scale dependency by choosing a point set which doesn't share pixels between scales (see Figure 6), by offsetting each sample location randomly by one pixel. Experiments with specialized point sets created using dart throwing to decorrelate the different scales did not improve the output further. Full brute force search for point sets to optimize image quality over a set of images using quasi-Monte Carlo methods such as [GHSK08] might yield even better results.

In order to illustrate the adaptive edge weights, Figure 7 shows three scales of local contrast enhancement with blended edge weights and the corresponding base layer. While a lot of detail can be blurred away in flat areas, the edges are detected by small local σ_r and correctly preserved.

A comparison to the high-quality histogram-based edge diffusion method [KS10] is given in Figure 8. On the GPU their approach consumes 0.83 seconds for the closest mode filter plus 0.25 seconds for the selective diffusion per megapixel. Using edge-optimization, we can create a comparable image with three wavelet scales in 0.087 seconds per megapixel. If the edge weights can be set globally by the user, the optimization step can be omitted and this image would compute in 15 ms/megapixel.

More detailed timings can be found in Table 2 (just edge-optimized wavelet transform). Estimating the parameters for BayesShrink denoising introduces approximately an additional 2ms per layer.

algorithm		wallclock	cpu time
[Fat09]	(core i7, $\alpha = 1$)	0.088s	0.430s
[Fat09]	(core i7, $\alpha = 0.8$)	0.296s	1.520s
this paper	(core i7)	0.249s	1.740s

Table 1: *Timings are median of three runs. To make the comparison as fair as possible, a megapixel has been processed with three scales on a core i7 (8 threads) using the CPU version of both algorithms, with only one tested σ_r. [Fat09] with $\alpha = 1$ removes an expensive exponentiation.*

	number of σ_r tested				
	1	2	3	4	5
1 scale	19	23	26	32	39
2 scales	27	35	43	51	63
3 scales	35	48	61	75	87
4 scales	42	61	81	102	120
5 scales	55	80	109	134	163

Table 2: *Performance measurements for the edge-optimized wavelet transform on a GTX480 for a one megapixel image. Numbers are in milliseconds. Testing more σ_r currently introduces a linear cost as each test calls a new kernel. Rearranging the computation – the same data is used over and over – could significantly accelerate this step. The slowest configuration (five scales of wavelet bases tested for five parameters) takes 163 ms, which is in the range of our parallel CPU implementation of the decimated edge-avoiding wavelets [Fat09] on a quad-core PC (200ms per megapixel).*

Figure 9 summarizes the results of including BayesShrink in the à-trous framework. We compare standard à-trous + BayesShrink with global edge-avoiding à-trous + BayesShrink and our edge-optimized à-trous + BayesShrink on images with 5%, 10%, and 40% added RGB noise. The performance of edge-avoiding à-trous wavelets depends a lot on the chosen global threshold. We chose a very small σ_r, which results in sharp images but also pushes smooth edges into the details, where they are attenuated by the wavelet shrinkage. This explains why the PSNR in the 10% image is even worse than the non-edge-aware version. 40% noise results in too high variance due to noise as compared to the signal, and thus the optimized edge weights detect noise as edges and fail to produce good images.

5. Conclusion

Edge-optimized à-trous wavelets for multi-scale image decomposition are a high-quality and fast alternative to currently used methods for image manipulation. They feature robust denoising and contrast enhancement. With our locally adaptive edge weights more signal information is robustly kept in the coarse layers, avoiding the artifacts of halos and gradient reversals in the context of local contrast enhancement. Compared to decimated wavelet transforms which suf-

Figure 8: *Base layer and contrast enhanced version of a bird. From left to right: original, result from [KS10] with explicit selective diffusion to match the edges in the base layer, our result with 3 scales and 2.5× and with 4× local contrast enhancement, indicating robustness.*

fer from a limited number of basis functions at the coarsest level our à-trous -based approach keeps the full image resolution, avoiding aliasing artifacts.

Our implementation is based on CUDA and is significantly faster (more than one order of magnitude) even than previous GPU methods with comparable quality and robustness. We achieve real-time rates for camera images captured at about one megapixel.

While our performance numbers are already practical for some use cases such as interactive image editing, low-dimensional parameter optimization and with some restrictions to interactive video, there is still room for improvement such as reordering computations to avoid duplicate memory accesses and to make better use of the shared memory on the GPU. The robustness of edge-optimized à-trous wavelets might in the future lend itself to tone mapping or image abstraction.

References

[ABD10] ADAMS A., BAEK J., DAVIS M. A.: Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum (EG 2010 Proceedings* (2010). 2

[AGDL09] ADAMS A., GELFAND N., DOLSON J., LEVOY M.: Gaussian kd-trees for fast high-dimensional filtering. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 21:1–21:12. 2

[Bur81] BURT P. J.: Fast filter transform for image processing. *Computer Graphics and Image Processing 16*, 1 (1981), 20 – 51. 2

[CP76] CRANLEY R., PATTERSON T.: Randomization of Number Theoretic Methods for Multiple Integration. *SIAM Journal on Numerical Analysis 13* (1976), 904–914. 3

[CYV00] CHANG G., YU B., VETTERLI M.: Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on image processing 9*, 9 (2000), 1532–1546. 4

[DD02a] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. 21* (July 2002), 257–266. 1

[DD02b] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH* (2002), pp. 257–266. 2

[DSHL10] DAMMERTZ H., SEWTZ D., HANIKA J., LENSCH H.: Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proc. High Performance Graphics 2010* (2010), pp. 67–75. 2

[Dut89] DUTILLEUX P.: An Implementation of the "algorithme à trous" to Compute the Wavelet Transform. In *Wavelets. Time-Frequency Methods and Phase Space* (1989), J.-M. Combes, A. Grossmann, & P. Tchamitchian, (Ed.), pp. 298–+. 2

[ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. In *ACM Transactions on Graphics (Proceedings of Siggraph Conference)* (2004), vol. 23, ACM Press. 1

[FAR07] FATTAL R., AGRAWALA M., RUSINKIEWICZ S.: Multiscale shape and detail enhancement from multi-light image collections. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (2007), p. 51. 1, 2, 4

[Fat09] FATTAL R.: Edge-avoiding wavelets and their applications. *ACM Trans. Graph. 28*, 3 (2009), 1–10. 1, 2, 4, 5, 6

[FFLS08] FARBMAN Z., FATTAL R., LISCHINSKI D., SZELISKI R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 67:1–67:10. 1, 2, 3, 4, 5

[GHSK08] GRÜNSCHLOSS L., HANIKA J., SCHWEDE R., KELLER A.: (t,m,s)-nets with maximized minimum distance. In *Proc. Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 2008, pp. 397–412. 6

[HKMMT89] HOLSCHNEIDER M., KRONLAND-MARTINET R., MORLET J., TCHAMITCHIAN P.: A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets, Time-Frequency Methods and Phase Space* (1989), Springer-Verlag, pp. 289–297. 2

[KGV83] KIRKPATRICK S., GELATT D., VECCHI M.: Optimization by simulated annealing. *Science 220*, 4598 (1983), 671–680. 3

[KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. Graph. 25* (July 2006), 654–663. 1

[KS10] KASS M., SOLOMON J.: Smoothed local histogram filters. *ACM Trans. Graph. 29* (July 2010), 100:1–100:10. 1, 2, 4, 6, 7

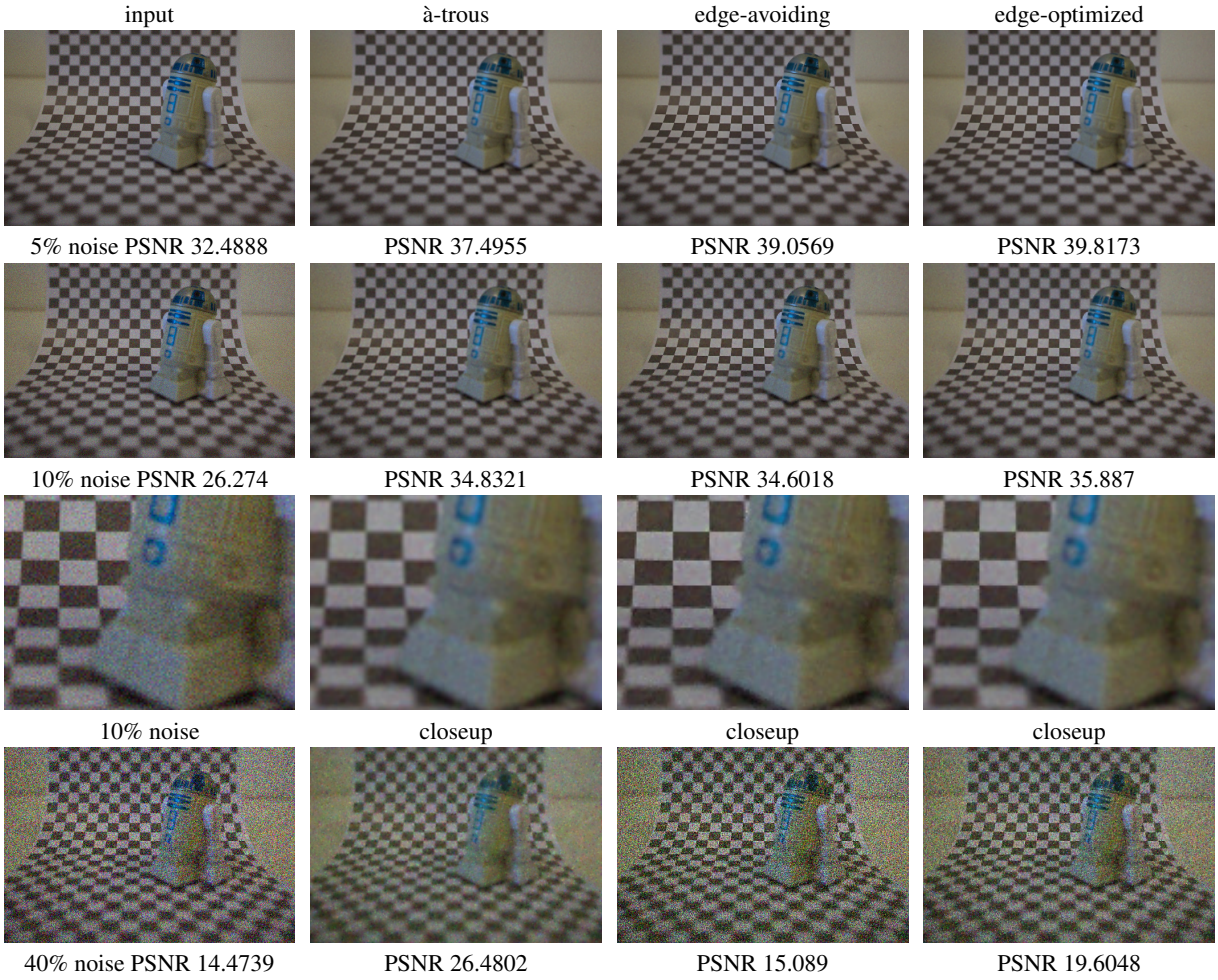

Figure 9: *Denoising can be improved using edge optimized à-trous wavelets: Top row: 10% RGB noise added, bottom row: 40% RGB noise added. From left to right: noisy image, standard à-trous wavelets, edge-avoiding à-trous wavelets with global parameters, and edge-optimized à-trous wavelets. When the global edge weight is not carefully chosen for edge-avoiding wavelets, it tends to oversharpen, so the PSNR is low. The 40% noise image is a failure case for the edge-avoiding and edge-optimized wavelets: since color differences are more pronounced there as in the actual edges, noise might be detected as edges and is propagated to the base layer. All images are created with three wavelet scales.*

[Mal89] MALLAT S.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11* (1989), 674–693. 2

[Mal98] MALLAT S.: *A Wavelet Tour of Signal Processing*. Academic Press, 1998. 2

[MRR*53] METROPOLIS N., ROSENBLUTH A., ROSENBLUTH M., TELLER A., TELLER E.: Equations of state calculations by fast computing machines. *Journal of Chemical Physics 21*, 6 (1953), 1087–1092. 3

[Mur97] MURTAGH F.: Multiscale transform methods in data analysis. 3

[PD09] PARIS S., DURAND F.: A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision 81*, 1 (2009), 24–52. 2

[PM90] PERONA P., MALIK J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 7 (1990), 629–639. 2

[Sch94] SCHLICK C.: Quantization techniques for visualization of high dynamic range pictures. In *Photorealistic Rendering Techniques* (1994), Springer-Verlag, pp. 7–20. 1, 2

[She92] SHENSA M.: The discrete wavelet transform: wedding the a trous and mallat algorithms. *Signal Processing, IEEE Transactions on 40*, 10 (oct 1992), 2464 –2482. 2

[Swe97] SWELDENS W.: The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal. 29*, 2 (1997), 511–546. 2

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision* (Washington, DC, USA, 1998), IEEE Computer Society, p. 839. 2