Fast Multi-Scale Detail Decomposition via Accelerated Iterative Shrinkage

Hicham Badri, Hussein Yahia, Driss Aboutajdine

Fast Multi-Scale Detail Decomposition via Accelerated Iterative Shrinkage

Hicham Badri[*]
INRIA, Geostat team
200 rue de la Vieille Tour,
33405 Talence Cedex, France

Hussein Yahia[†]
INRIA, Geostat team
200 rue de la Vieille Tour,
33405 Talence Cedex, France

Driss Aboutajdine[‡]
Mohammed V-Agdal University
LRIT, Associated Unit to CNRST (URAC 29)
Rabat, Morocco

Abstract

We present a fast solution for performing multi-scale detail decomposition. The proposed method is based on an accelerated iterative shrinkage algorithm, able to process high definition color images in real-time on modern GPUs. Our strategy to accelerate the smoothing process is based on the use of first order proximal operators. We use the approximation to both designing suitable shrinkage operators as well as deriving a proper warm-start solution. The method supports full color filtering and can be implemented efficiently and easily on both the CPU and the GPU. We demonstrate the performance of the proposed approach on fast multi-scale detail manipulation of low and high dynamic range images and show that we get good quality results with reduced processing time.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering.

Keywords: fast edge-aware smoothing, multi-scale image decomposition, tone mapping.

1 Introduction

During the past few years, there has been a significant amount of work on edge-aware filtering. Unlike regular Gaussian smoothing, edge-aware filters blur the image while preserving sharp edges. Probably the most popular edge-aware filter is the bilateral filter [Tomasi and Manduchi 1998] which performs a weight averaging of the colors in a window based on both space and range distances. The bilateral filter can be seen as a high-dimensional filter working in a 5D space when performed on 2D RGB images [Barash 2002]. A naive implementation of this filter is too slow as it operates on a high-dimension space. Many researchers tried to boost this filter or at least simulate bilateral-like results by either using linear interpolation [Durand and Dorsey 2002], reformulation and downsampling [Paris and Durand 2009; Banterle et al. 2012] or dimensionality reduction [Gastal and Oliveira 2011], among others.

Other methods such as [Farbman et al. 2008] and [Xu et al. 2011] use gradient-based optimization formulation to perform edge-preserving filtering. Farbman *et al.* formulate edge-aware smoothing as a Weighted Least Squares optimization problem (WLS). Xu *et al.* make use of the L_0-minimization framework to progressively suppress details. In this brief, we perform edge-aware smoothing

[*]e-mail:hicham.badri@inria.fr
[†]e-mail:hussein.yahia@inria.fr
[‡]e-mail:aboutaj@ieee.org

by running very few gradient shrinkage-reconstruction iterations. Similar methods such as [Xu et al. 2011] unfortunately need a relatively high number of iterations to produce a suitable result. Our strategy to improve half-quadratic solvers is based on two points : (1) designing new shrinkage operators that can produce a suitable photographic look for multi-scale detail manipulation at low iterations, (2) deriving an efficient warm-start solution. We show that we get good quality results with reduced processing time as can be seen in Figure 1.
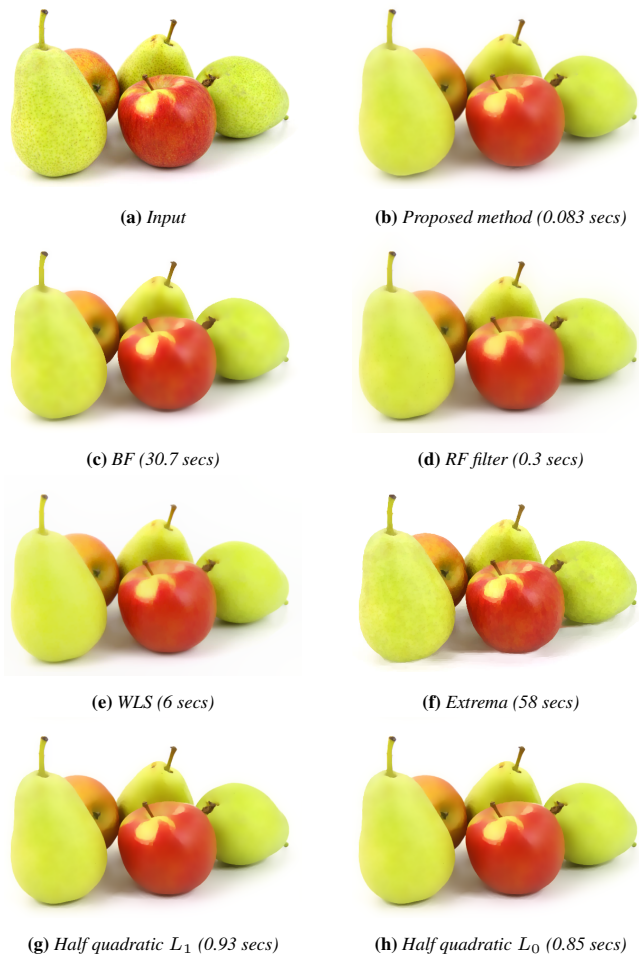

(a) *Input*

(b) *Proposed method (0.083 secs)*

(c) *BF (30.7 secs)*

(d) *RF filter (0.3 secs)*

(e) *WLS (6 secs)*

(f) *Extrema (58 secs)*

(g) *Half quadratic L_1 (0.93 secs)*

(h) *Half quadratic L_0 (0.85 secs)*

Figure 1: *Image smoothing comparison. (a) Input image. (b) Proposed method with $\lambda = 0.03$, $\gamma = 12$ and $\alpha = 25$ for 2 iterations using f_2. (c) Bilateral filter with $(\sigma_s = 20, \sigma_r = 0.15)$ for 2 iterations. (d) RF filter [Gastal and Oliveira 2011] with $(\sigma_s = 30, \sigma_r = 0.4)$ for 3 iterations. (e) [Farbman et al. 2008] with $\lambda = 0.25$ and $\alpha = 1$ for 2 iterations. (f) [Subr et al. 2009] for 2 iterations. (g) FFT half-quadratic solver as described in section 2.1 for the L_1-norm. (h) FFT half-quadratic solver for the L_0 quasi-norm [Xu et al. 2011].*

2 Iterative Shrinkage Smoothing

2.1 Problem Formulation

Given an input image g, we seek a smooth image u. The problem can be posed as follows :

$$\underset{u}{\operatorname{argmin}} \ \frac{\lambda}{2}||u - g||_2^2 + \psi(\nabla u), \qquad (1)$$

where $\psi(\nabla u)$ is a gradient function and λ is a positive regularization term. Producing a smooth image requires forcing the output u to have "sparse" gradients. Thus, $\psi(.)$ should be a sparsity-inducing function[1]. Problem (1) is not easy to solve as the function $\psi(.)$ can be a non-smooth function or not even convex. One popular method to tackle such problem is by introducing an additional variable v to obtain a half-quadratic form :

$$\underset{u,v}{\operatorname{argmin}} \ \frac{\lambda}{2}||u - g||_2^2 + \psi(v) + \frac{\beta}{2}||\nabla u - v||_2^2, \qquad (2)$$

where β is a new regularization term. The problem above can be solved by alternative minimization :

$$\begin{aligned} v^{(k+1)} &\leftarrow \underset{v}{\operatorname{argmin}} \ \psi(v) + \frac{\beta}{2}||\nabla u^{(k)} - v||_2^2 \\ u^{(k+1)} &\leftarrow \underset{u}{\operatorname{argmin}} \ \lambda||u - g||_2^2 + \beta||\nabla u - v^{(k+1)}||_2^2, \end{aligned} \qquad (3)$$

where k is the current iteration number. Calculating $v^{(k+1)}$ corresponds to a shrinkage operation while calculating $u^{(k+1)}$ corresponds to the screened Poisson equation [Bhat et al. 2008]. This equation can be solved either using the FFT (Fast Fourier Transform) or the preconditioned conjugate gradient method [Saad 2003] with an appropriate preconditioner. In this brief, we use the FFT solver as sparse linear systems are not yet supported for GPU computing in Matlab :

$$u^{(k+1)} \leftarrow \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(\lambda g - \beta \operatorname{div}(v^{(k+1)}) \right)}{\lambda - \beta \operatorname{lap}} \right), \qquad (4)$$

where $\mathcal{F}$ is the Fourier transform, div is the discrete divergence operator [2] and lap is the OTF (optical transfer function) of the discrete Laplacian filter that is calculated only once when the input is loaded. Adopting this splitting scheme, the method needs two FFTs per iteration (per channel) to recover $u^{(k+1)}$ which is costly when the number of iterations needed is high. Our goal in this brief is to perform faster multi-scale detail decomposition by making use of a first order estimation of proximal operators.

2.2 Accelerated Iterative Shrinkage Smoothing

In this subsection, we describe the proposed strategy to make the shrinkage scheme described above faster and suitable for multi-scale detail decomposition.

2.2.1 First Order Proximal Operators

Let h be a differentiable function. We define the proximal operator $\mathbf{prox}_{th}$ of h as follows :

$$\mathbf{prox}_{th}(x) = \underset{v}{\operatorname{argmin}}\left\{ h(v) + \frac{1}{2t}||v - x||_2^2 \right\}, \qquad (5)$$

---

[1] A function which can model heavy-tailed distributions.

[2] The discrete Laplacian operator Δ corresponds to $\Delta := -(\nabla_x^T \nabla_x + \nabla_y^T \nabla_y) := -L$, where L is the discrete Laplacian matrix.

where t is a positive regularization term. This operator is important in splitting algorithms and has many useful interpretations (please refer to [Parikh and Boyd 2013] for an excellent study of proximal algorithms). Solution can be written as :

$$\mathbf{prox}_{th}(x) = (I + t\nabla h)^{-1}(x), \qquad (6)$$

where I is the identity matrix. By using a first order Taylor expansion $h(x) \approx h(v) + \nabla h(v)^T(x - v)$, the proximal operator can be approximated with :

$$\mathbf{prox}_{th}(x) \approx x - t\nabla h(x). \qquad (7)$$

The solution is refined iteratively as in a gradient descent method. In this brief, we use this approximation in two important steps of the smoothing algorithm ; defining new shrinkage operators and deriving an efficient warm-start solution.

2.2.2 Photographic Derivative Prior

One important parameter in multi-scale detail decomposition is defining the smoothing behavior. This is directly related to the derivative distribution prior adopted in the method. For instance, considering ψ as the l_1-norm is supposing that the derivative distribution follows a Laplacian law ($p(x) \propto e^{-\tau|x|}$). However, studies have shown that real-world images' gradients distribution has a heavier tail than a Laplacian. In this brief, we consider modified Cauchy and Welsch derivative priors that we found empirically fitting better natural images than the Laplacian distribution. It is important to define appropriate derivative priors for multi-scale detail decomposition to correctly remove details at each step while preserving salient structures. More important, an appropriate prior will give a better quality result with reduced iterations, which is our main motivation in this brief. Unfortunately, using sophisticated sparse distributions makes the problem non-convex and deriving the shrinkage operator in this case remains a challenge. Using the first order proximal operator defined in the preview subsection, we get accurate estimation of the shrinkage operation. The shrinkage subproblem of problem (3) can be formulated as :

$$v^{(k+1)} \leftarrow \mathbf{prox}_{\frac{1}{\beta}\psi}(\nabla u^{(k)}) \approx \nabla u^{(k)} - \frac{1}{\beta}\nabla\psi\left(\nabla u^{(k)} \right), \quad (8)$$

which simplifies to :

$$v_p^{(k+1)} \leftarrow \nabla u_p^{(k)}\left(1 - \frac{1}{\beta}w_\psi\left(\nabla u_p^{(k)} \right) \right), \qquad (9)$$

where $w_\psi(x) = \frac{\psi(x)'}{x}$ is a weight function of ψ and p is a pixel location. For the sake of simplicity, we consider fixed $\beta = 1$ throughout the brief. Shrinkage can thus be approximated with simple pointwise multiplication $v^{(k+1)} \leftarrow \nabla u^{(k)} \circ f(\nabla u^{(k)})$. For color images, we define the following gradient measure instead of the norm of each channel of ∇u :

$$T(\nabla u) = \sqrt{\left(\sum_{k=1}^{ch}|\frac{\partial u_k}{\partial x}| \right)^2 + \left(\sum_{k=1}^{ch}|\frac{\partial u_k}{\partial y}| \right)^2}, \qquad (10)$$

where ch is the number of channels. As discussed before, we consider modified Cauchy and Welsch priors that give the following weights :

$$f_1(T(\nabla u)) = 1 - \frac{1}{1 + (T(\nabla u)/\gamma)^\alpha} \qquad (11)$$

$$f_2(T(\nabla u)) = 1 - e^{-(T(\nabla u)/\gamma)^\alpha}, \qquad (12)$$

where α and γ are positive parameters. In the original distributions $\alpha = 2$. We introduce the parameter α for more sparsity [3].

---

[3] In fact, it is easy to see that when $\alpha \to +\infty$, f_1 and f_2 correspond to hard-thresholding operators.

2.2.3 Efficient Warm-Start

Solving problem (3) corresponds to an iterative process. Thus, the initial solution $u^{(0)}$ plays an important role in terms of speed of the algorithm. To accelerate the method, we derive a warm-start solution $u^{(0)}$. This initial solution should be very fast to compute and has to be calculated only once, when the input image is loaded by the user. Clearly, the difficulty is due to solving the screened Poisson equation in (3). We rewrite the problem in the matrix form

$$\underbrace{(\beta L + \lambda I)}_{A}\underbrace{u^{(0)}}_{x} = \underbrace{\left(-\beta \text{div}(v^{(0)}) + \lambda g\right)}_{b}, \qquad (13)$$

where L is the discrete Laplacian matrix and I is the identity matrix. Applying the first order proximal operator gives :

$$u_{k+1}^{(0)} \leftarrow u_k^{(0)} - t\left\{(\beta L + \lambda I)\, u_k^{(0)} + \beta \text{div}(v^{(0)}) - \lambda g\right\}. \quad (14)$$

For one iteration, the warm-start solution $u^{(0)}$ can be approximated with the following formula :

$$u^{(0)} \approx g + \xi \text{div}\left(\nabla g - \nabla g \circ f_i(T(\nabla g))\right) \;\;, \;\; \xi > 0. \qquad (15)$$

We fix $\beta = 1$ and run very few iterations (1 to 2) of problem (3) to recover the base layer in the multi-scale detail decomposition process. For the warm-start parameters, ξ is set between 0.01 and 0.25, α is fixed to 2 and γ is set to around 400. We summarize the proposed method for edge-aware smoothing in algorithm 1.

Data: Input image g, parameters λ, γ, α, warm-start parameters and the number of iterations $iter$.
Result: Smoothed image u.
Step 1 : Calculate the filter lap and store it;
Step 2 : Calculate the warm-start solution $u^{(0)}$ and store it;
for $l=1$ to $iter$ **do**
    Step 3-1 : Calculate the shrinkage weights $f_i(T(\nabla g))$;
    Step 3-2 : Calculate $v^{(l)} = u^{(l-1)} \circ f_i(T(\nabla g))$;
    Step 3-3 : Calculate $u^{(l)}$ with equation (4);
end
Return the smoothed image $u \leftarrow u^{(iter)}$;
      Algorithm 1: Our solution to fast edge-aware filtering.

Figure 1 demonstrates the performance of the proposed approach in comparison with various state-of-the art methods. Smoothing was performed in the CIELab colorspace. Note the quality of the result after only 2 iterations. The method takes less processing time with improved quality compared to the RF filter of the domain transform method [Gastal and Oliveira 2011]. Note that the domain transform is one of the fastest methods up to now. The proposed strategy is able to make half-quadratic solvers about 10 times faster while still producing a good result.

2.3 Performance Evaluation

We have implemented all the applications present in this paper in Matlab2012b. The proposed filter requires only 1 to 2 iterations. All the experiments run on Intel Xeon CPU E5-2606 @2.4Ghz and Nvidia Tesla C2075 GPU. Filtering a 1-megapixel full color image with our filter takes around 0.15 seconds, 0.33 seconds for a 1080p color image and less than 2 seconds for a 12 megapixel color image on the CPU. On the GPU, we filter 720p and 1080p images in real-time. As mentioned before, solving the screened Poisson equation can be also performed using fast sparse linear solvers. We found that the proposed filter preserves time consistency and can be thus

used also for fast video edge-aware manipulation. Please refer to the supplementary material for video examples (© copyright 2008, Blender Foundation / www.bigbuckbunny.org).

3 Fast Multi-Scale Detail Manipulation

Multi-scale Detail Enhancement Edge-aware filtering can be used to decompose one image into several layers according to its degrees of details. One can thereafter manipulate each layer and recombine them to boost details on multiple scales [Farbman et al. 2008]. Let $B_0,...,B_k$ be different smoothed versions of the input image g. As k becomes larger, B_k becomes coarser, with $g = B_0$. These layers are called *base layers*. Detail layers D_l are extracted by subtracting the base layer B_{l+1} from its richer version B_l as $D_l = B_l - B_{l+1}$. Each detail layer is then multiplied by a parameter as well as the coarsest layer B_k and summed to form the output image. Figure 2 (b) presents an example of fine-scale detail
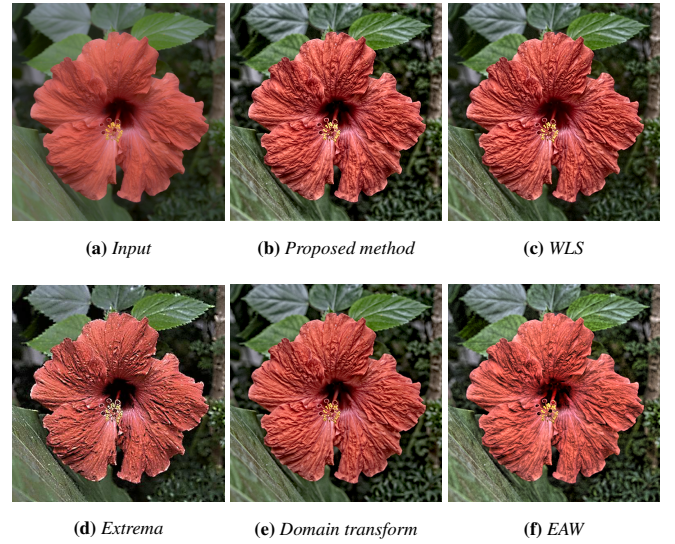

(a) *Input*   **(b)** *Proposed method*   **(c)** *WLS*

(d) *Extrema*   **(e)** *Domain transform*   **(f)** *EAW*

Figure 2: *Fine Detail Enhancement. (a) Input image. (b) Proposed method using f_1. (c) Weighted-least squares approach [Farbman et al. 2008]. (d) Local extrema decompositing [Subr et al. 2009]. (e) Domain transform [Gastal and Oliveira 2011]. (f) Edge-avoiding wavelets from [Fattal 2009].*

enhancement applied to the flower (a). As can be seen, the result generated with the proposed method is visually similar to the one produced with the WLS filter (c). However, generating the two layers took only 0.047 seconds with our approach. WLS method [Farbman et al. 2008] took 2.7 seconds in Matlab R2012b with the direct solver and around 1.5 seconds with the PCG method [Saad 2003] using an incomplete Cholesky factorization preconditioner.

HDR Tone Mapping As regular display screens have a low dynamic range, high dynamic range images (HDR) require compression before displaying them. This operation is known as "tone mapping". Edge-aware filtering can be used for tone mapping by performing a multi-scale decomposition of the log-luminance channel similar to the one discussed in the previous paragraph. We present in Figure 3 an example of HDR tone mapping with our approach. Our result (d) is artifact-free and visually similar to (c). The proposed solution took only 0.025 seconds on Matlab2012b to extract the base layer. Another example of HDR tone mapping is presented in Figure 4 with a 3 layers decomposition. The method produces a suitable photographic look with reduced processing time.

(a) *[Durand and Dorsey 2002]*   **(b)** *[Gastal and Oliveira 2011]*   **(c)** *[Farbman et al. 2008]*   **(d)** *Proposed method*

Figure 3: *HDR Tone Mapping. (a) Result with the bilateral filter. (b) Result using the domain transform approach. (c) Result using WLS [Farbman et al. 2008]. (d) Our result for 1 iteration with $\lambda = 0.006$, $\gamma = 40$ and $\alpha = 2$ using f_2;*

(a) *[Farbman et al. 2008]*   **(b)** *Proposed method*

Figure 4: *Multi-layer HDR Tone Mapping. (a) Result with WLS. (b) Proposed method. (HDR image © Industrial Light & Magic. All rights reserved.)*

4 Conclusion

We present a fast solution for multi-scale detail manipulation. Our approach is based on an accelerated iterative shrinkage method. We introduce new sparsity-inducing weights and an efficient warm-start solution for fast processing. We show that we get good quality results with reduced processing time. We demonstrate the performance of our solution on low and high dynamic range images.

Acknowledgements

References

BANTERLE, F., CORSINI, M., CIGNONI, P., AND SCOPIGNO, R. 2012. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. *Comput. Graph. Forum 31*, 1, 19–32.

BARASH, D. 2002. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 6 (June), 844–847.

BHAT, P., CURLESS, B., COHEN, M., AND ZITNICK, C. L. 2008. Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *Proceedings of ECCV : Part II*, Springer-Verlag, Berlin, Heidelberg, ECCV '08, 114–128.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '02, 257–266.

FARBMAN, Z., FATTAL, R., LISCHINSKI, D., AND SZELISKI, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph. 27*, 3 (Aug.), 67:1–67:10.

FATTAL, R. 2009. Edge-avoiding wavelets and their applications. *ACM Trans. Graph. 28*, 3, 1–10.

GASTAL, E. S. L., AND OLIVEIRA, M. M. 2011. Domain transform for edge-aware image and video processing. *ACM TOG 30*, 4, 69:1–69:12. Proceedings of SIGGRAPH 2011.

PARIKH, N., AND BOYD, S. Proximal algorithms. To appear, *Foundations and Trends in Optimization, 1(3):123-231, 2013.*.

PARIS, S., AND DURAND, F. 2009. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision 81*, 1 (Jan.), 24–52.

SAAD, Y. 2003. *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

SUBR, K., SOLER, C., AND DURAND, F. 2009. Edge-preserving multiscale image decomposition based on local extrema. In *ACM SIGGRAPH Asia 2009 papers*, ACM, New York, NY, USA, SIGGRAPH Asia '09, 147:1–147:9.

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

XU, L., LU, C., XU, Y., AND JIA, J. 2011. Image smoothing via l0 gradient minimization. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, ACM, New York, NY, USA, SA '11, 174:1–174:12.