Fast edge-preserving image denoising via group coordinate descent on the GPU

Madison G. McGaffin[a] Jeffrey A. Fessler[a]

[a]EECS Department, University of Michigan, Ann Arbor, MI, USA

ABSTRACT

We present group coordinate descent algorithms for edge-preserving image denoising that are particularly well-suited to the graphics processing unit (GPU). The algorithms decouple the denoising optimization problem into a set of iterated, independent one-dimensional problems. We provide methods to handle both differentiable regularizers and the absolute value function using the majorize-minimize technique. Specifically, we use quadratic majorizers with Huber curvatures for differentiable potentials and a duality approach for the absolute value function. Preliminary experimental results indicate that the algorithms converge remarkably quickly in time.

Keywords: denoising, group coordinate descent, GPU, duality, absolute value, anisotropic total variation, majorize-minimize

1. INTRODUCTION

Consider the image denoising problem

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \geq \mathbf{0}}{\operatorname{argmin}} \left\{ J(\mathbf{x}) = \frac{1}{2} ||\mathbf{x} - \mathbf{y}||_{\mathbf{W}}^2 + \mathsf{R}(\mathbf{x}) \right\}, \tag{1}$$

with noisy data $\mathbf{y} \in \mathbb{R}^N$, diagonal matrix of nonnegative weights $\mathbf{W}$, and edge-preserving regularizer R:

$$\mathsf{R}(\mathbf{x}) = \sum_{d=1}^{D} \beta_d \sum_{k=1}^{N} \kappa_{dk} \phi([\mathbf{C}_d \mathbf{x}]_k). \tag{2}$$

We focus on the nonnegativity-constrained minimization problem 1 because of its relevance to x-ray CT image reconstruction problems. However, the methods presented here are easily simplified for the unconstrained case.

Each $\mathbf{C}_d$ is a first-order finite difference matrix in the "dth direction." These matrices are circulant or Toeplitz and have a band of $+1$ values along the diagonal and a band of -1 values along an off-diagonal. A two-dimensional denoising problem may use $D = 2$ to penalize horizontal and vertical differences or $D = 4$ to penalize the diagonal differences as well. In three dimensions, this becomes $D = 3$ for the cardinal directions and $D = 13$ to include all neighbors. The algorithm here can be trivially extended to higher-order differences, *i.e.*, difference matrices $\mathbf{C}_d$ with more or different bands, but we restrict discussion to the first-order difference case here.

The potential function ϕ is positive, even and convex. Potential functions other than the obvious quadratic choice $\phi(t) = \frac{1}{2}t^2$ are used to avoid over-smoothing edges in the reconstructed image. In this paper, we will consider two classes of potential functions:

1. potential functions that are even, convex, positive and satisfy Huber's majorizer conditions; and
2. the absolute value function, $\phi(t) = |t|$, which is used in anisotropic total-variation regularization.[4]

Supported in part by NIH grant R01 HL 098686, and by equipment donations from Intel Corporation.

M.G.M.: E-mail: `mcgaffin@umich.edu`, Telephone: 734-763-5022
J.A.F.: E-mail: `fessler@umich.edu`, Telephone: 734-763-1434

Computational Imaging XII, edited by Charles A. Bouman, Ken D. Sauer, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9020, 90200P · © 2014 SPIE-IS&T · CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2042593

Because the absolute value function is not differentiable at the origin, it requires special treatment. Many convex potential functions fall into the first category.

To illustrate an important property of this cost function, we rewrite the regularizer:

$$\mathsf{R}(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j} \kappa_{kj} \beta_{kj} \phi(x_j - x_k), \tag{3}$$

where $\mathcal{N}_j$ is the set of indices in the neighborhood of the jth index. As is conventional, we let the neighborhood relationship be symmetric: if $k \in \mathcal{N}_j$, then $j \in \mathcal{N}_k$. This version "double-counts" each difference, hence the $\frac{1}{2}$ factor. Consider solving for the jth pixel of $\mathbf{x}$ while holding all the rest constant:

$$\operatorname*{argmin}_{\alpha \geq -x_j} J(\mathbf{x} + \alpha \mathbf{e}_j) = \operatorname*{argmin}_{\alpha \geq -x_j} \frac{w_j}{2}(x_j + \alpha - y_j)^2 + \sum_{k \in \mathcal{N}_j} \frac{\kappa_{kj}\beta_{kj}}{2} \phi(x_j + \alpha - x_k).$$

The only terms required are x_j, w_j, y_j, some regularizer weights, and the pixels in $\mathcal{N}_j$. Put another way, the cost function J is in some sense Markovian: pixels that are not neighbors are marginally independent. We will exploit this structure below.

The rest of this paper is organized as follows. Section 2 describes our proposed coordinate descent algorithm. Sections 2.1.1 and 2.1.2 describe the one-dimensional problems for differentiable potential functions and the absolute value function, respectively. Some experimental results are given in Section 3, and Section 4 contains some concluding remarks.

1.1 Notation

We consider images with $N = N_x N_y$ pixels. Variables in bold are vectors. The jth element of a vector is $[\mathbf{x}]_j$ or x_j. For simplicity, this paper will deal with two-dimensional images, though the extension to three dimensions is conceptually trivial. Sometimes it will be convenient to consider the jth element of a vector in nonnegative integral coordinates, $(p, q) \in \{0, \dots, N_x - 1\} \times \{0, \dots, N_y - 1\}$. In those cases, it is more convenient to number coordinates from zero instead of from one:

$$x_{p,q} = x_{p + qN_x + 1}, \tag{4}$$

and conversely

$$x_j = x_{(j-1) \bmod N_x,\ \lfloor (j-1)/N_x \rfloor N_x}, \tag{5}$$

where $\lfloor \cdot \rfloor$ is truncation.

Iteration number will be indexed by n, and written as a superscript in parentheses, *e.g.*, $\mathbf{x}^{(n)}$. Subiterations will be similarly indicated using the letter m.

2. GROUP COORDINATE DESCENT

We divide $\mathbf{x}$ into K disjoint *groups*; $\mathbf{x}_1, \dots, \mathbf{x}_K$:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{S}_k \mathbf{x}_k, \tag{6}$$

where $\mathbf{x}_k \in \mathbb{R}^{N_k}$, $\sum_{k=1}^{K} N_k = N$, and $\mathbf{S}_k \in \mathbb{R}^{N \times N_k}$ is formed from a subset of the columns from the $N \times N$ identity matrix. We can write the cost function $J(\mathbf{x})$ as a function of these grouped variables:

$$J(\mathbf{x}) = J(\mathbf{x}_1, \dots, \mathbf{x}_K). \tag{7}$$

Group coordinate descent algorithms solve the image denoising problem (1) by iteratively optimizing over each of the K groups while holding the others constant.[2,5] That is, in the nth (outer) iteration, loop over group index k and perform the following optimization:

$$\mathbf{x}_k^{(n+1)} = \underset{\mathbf{x}_k \geq \mathbf{0}}{\operatorname{argmin}} \; J\Big(\mathbf{x}_1^{(n+1)}, \ldots, \mathbf{x}_{k-1}^{(n+1)}, \mathbf{x}_k, \mathbf{x}_{k+1}^{(n)}, \ldots, \mathbf{x}_K^{(n)}\Big). \tag{8}$$

We will assign pixels to groups in a checkerboard-like pattern. We separate the pixels into groups that are independent with respect to the cost function; *i.e.*, no group contains pixels that are neighbors to one another. Let $\mathcal{S}_k$ be the set of pixel indices in the kth group, so the columns of $\mathbf{S}_k$ are the unit vectors $\mathbf{e}_j$ for $j \in \mathcal{S}_k$. For a two-dimensional problem, with first-order differences, $K = 4$. Let $k_p = k \bmod 2$ and $k_q = \lfloor k/2 \rfloor$. Then $\mathcal{S}_k$ is

$$\mathcal{S}_k = \{(k_p + 2r, k_q + 2s) : r, s \text{ are nonnegative integers}\}. \tag{9}$$

2.1 Group update subproblem

Consider now the task of performing an update to the kth group (8):

$$\mathbf{x}_k^{(n+1)} = \underset{\mathbf{x}_k \geq \mathbf{0}}{\operatorname{argmin}} \sum_{m \in \mathcal{S}_k} \left(\frac{w_m}{2} ([\mathbf{x}_k]_m - y_m)^2 + \sum_{j \in \mathcal{N}_m} \frac{\beta_{jm} \kappa_{jm}}{2} \phi([\mathbf{x}_k]_m - x_j) \right).$$

Recall that we have designed $\mathcal{S}_k$ so $\mathcal{N}_m \cap \mathcal{S}_k = \emptyset$ for all $m \in \mathcal{S}_k$. Consequently, the sum over $\mathcal{S}_k$ in (10) is separable:

$$\mathbf{x}_k^{(n+1)} = \underset{\mathbf{x}_k \geq \mathbf{0}}{\operatorname{argmin}} \sum_{m \in \mathcal{S}_k} \Psi_m^{(n)}([\mathbf{x}_k]_m); \tag{10}$$

$$\Psi_m^{(n)}(x) = \frac{w_m}{2} (x - y_m)^2 + \sum_{j \in \mathcal{N}_m} \frac{\beta_{jm} \kappa_{jm}}{2} \phi(x - x_j). \tag{11}$$

Each of the $\Psi_m^{(n)}$ is a one-dimensional function with the following form:

$$\Psi(x) = \frac{w}{2} (x - y)^2 + \sum_{j \in \mathcal{N}} \beta_j \phi(x - x_j), \tag{12}$$

where we have dropped subscripts and consolidated terms for simplicity. Due to separability, each coordinate of $\mathbf{x}_k$ can be updated independently (and thus simultaneously) by solving the corresponding one-dimensional problem.

2.1.1 Differentiable potential functions

For differentiable ϕ, we use quadratic majorizers with Huber's well-known curvatures [7, p. 185]:

$$\Phi(x) = \frac{w}{2} (x - y)^2 + \sum_{j \in \mathcal{N}} \beta_j \left(g_j^{(m)} x + \frac{c_j^{(m)}}{2} \Big(x - x^{(m)}\Big)^2 \right), \tag{13}$$

$$g_j^{(m)} = \phi'\Big(x^{(m)} - x_j\Big), \tag{14}$$

$$c_j^{(m)} = \omega\Big(x^{(m)} - x_j\Big) = \frac{g_j^{(m)}}{\left|x^{(m)} - x_j\right|}. \tag{15}$$

This one-dimensional quadratic surrogate is solved in closed form, yielding the update

$$x^{(m+1)} = \max \left\{ 0, \frac{w \cdot y + \sum_{j \in \mathcal{N}} \beta_j \Big(c_j^{(m)} x^{(m)} - g_j^{(m)} \Big)}{w + \sum_{j \in \mathcal{N}} \beta_j c_j^{(m)}} \right\}. \tag{16}$$

2.1.2 The absolute value function

If $\phi(t) = |t|$, the Huber curvature is undefined at $t = 0$. Many authors get around this problem by corner rounding, substituting an approximation, *e.g.*, $\phi(t) = \sqrt{t^2 + \epsilon} \approx |t|$, for the absolute value function. This feels disingenuous: why substitute a differentiable function for the absolute value at the last moment instead of using it in the problem formulation?

Another approach is to apply the Huber curvatures optimistically, assuming that for all $k \in \mathcal{N}_j$, $x_k \neq x_j$ and thus the Huber curvatures are bounded. In[12] a probabilistic argument is presented that this is the case for noisy natural images. However, the motivation of using a strong regularizer like total variation is often exactly to force adjacent pixel values to be equal. In medical imaging, pixels outside the patient often end up being exactly uniformly zero (due to the nonnegativity constraint and regularizer) so many neighbors have identical values. In problems with particularly high regularizer strength (*i.e.*, β large), uniformity will certainly be reached. Regardless, we are unsettled by any algorithm that could have numerical issues when neighboring pixel values are identical or nearly identical. Therefore we present a different approach that can work with absolute value potential functions exactly without any approximations.

The absolute value function is majorized by the following version of the Huber *function:*

$$h(t;\delta) = \begin{cases} \frac{1}{2\delta}\left(t^2 + \delta^2\right), & |t| < \delta, \\ |t|, & |t| \geq \delta, \end{cases} \tag{17}$$

for any $\delta \geq 0$. When $\delta = 0$, this reduces to the absolute value function. We use $h(t;\delta)$ to majorize the one-dimensional cost function at $x^{(n)}$:

$$\Phi_h(x) \triangleq \frac{w}{2}(x-y)^2 + \sum_{j\in\mathcal{N}} \beta_j h\Big(x - x_j; \Big|x^{(n)} - x_j\Big|\Big). \tag{18}$$

The Huber function can also be written implicitly, which motivates a second majorization by expanding the dual variable domain:

$$h(t;\delta) = \max_{\gamma\in[-1,1]} \left\{ s(t,\gamma;\delta) = \gamma t - \frac{\delta}{2}\left(\gamma^2 - 1\right) \right\} \tag{19}$$

$$\leq \max_{\gamma\in\Omega} s(t,\gamma;\delta), \tag{20}$$

where Ω is a convex, compact subset of $\mathbb{R}$ with $[-1,1] \subset \Omega$. We describe Ω in more detail below.

With this expanded-domain Huber-like function , we form the new majorizer Φ_m:

$$\Phi_m(x) \triangleq \max_{\boldsymbol{\gamma}\in\Omega^D} S(x;\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_D)) \tag{21}$$

$$S(x;\boldsymbol{\gamma}) = \frac{w}{2}(x-y)^2 + \sum_{j\in\mathcal{N}} \beta_j s\Big(x - x_j, \gamma_j; \Big|x^{(n)} - x_j\Big|\Big). \tag{22}$$

An example of Ψ, Φ_h and Φ_m is given in Figure 1. Observe that at the solution $x^{(n)} = 1$, both majorizers have a nondifferentiable kink, as does the cost function Ψ.

Eliminating $\boldsymbol{\gamma}$ from (22) produces a quadratic with curvature that goes to infinity as $x^{(n)} \to x_j$. It might seem that we have gotten nowhere. Fortunately, minimizing (22) has a convenient dual problem. Instead of solving

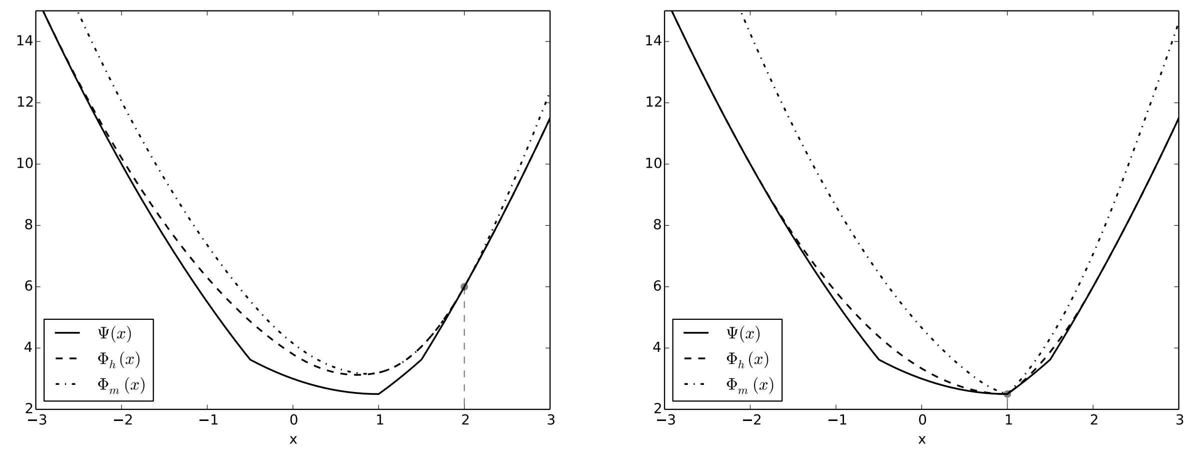
$$x^{(n+1)} = \operatorname*{argmin}_{x\geq 0} \max_{\boldsymbol{\gamma}\in\Omega^D} S(x;\boldsymbol{\gamma}), \tag{23}$$

we reverse the order of the minimization and maximization operations:

$$x^{(n+1)} = \max\{x(\boldsymbol{\gamma}_*), 0\}, \tag{24}$$

$$x(\boldsymbol{\gamma}) = \operatorname*{argmin}_{x} S(x;\boldsymbol{\gamma}) = y - \frac{1}{w}\boldsymbol{\beta}'\boldsymbol{\gamma}, \tag{25}$$

$$\boldsymbol{\gamma}_* \in \operatorname*{argmax}_{\boldsymbol{\gamma}\in\Omega^D} S(x(\boldsymbol{\gamma});\boldsymbol{\gamma}), \tag{26}$$



(a) Evaluated at $x^{(m)} = 2$ (b) Evaluated at $x^{(m)} = 1$

Figure 1: Illustrations of Ψ involving the absolute value function and its majorizers Φ_h and Φ_m. The example has three x_j at -1.5, 1 and 1.5; $y = 0$. On the left the majorizers are generated at the non-optimal point 2, and on the right at the optimum 1.

where $\boldsymbol{\beta} = \text{vec}_j\{\beta_j\}$. Let $\mathbf{D} = \text{diag}_j\{\beta_j|x^{(n)} - x_j|\}$ and $\boldsymbol{\Lambda} = \text{diag}_j\{y - x_j\}$. Plugging (25) into (26) yields the following quadratic concave optimization problem:

$$\gamma_* \in \underset{\gamma \in \Omega^D}{\text{argmax}} \; -\frac{1}{2}||\gamma||^2_{\mathbf{D}+\frac{1}{w}\boldsymbol{\beta}\boldsymbol{\beta}'} + \gamma'\boldsymbol{\Lambda}\boldsymbol{\beta}. \tag{27}$$

We choose $\Omega = [-R, R]$ with

$$R = \max\left\{1, \left|\left|\underbrace{\left(\mathbf{D} + \frac{1}{w}\boldsymbol{\beta}\boldsymbol{\beta}'\right)^+ \boldsymbol{\Lambda}\boldsymbol{\beta}}_{\gamma_{\text{pinv}}}\right|\right|_\infty\right\}. \tag{28}$$

This selection for Ω has two important properties:

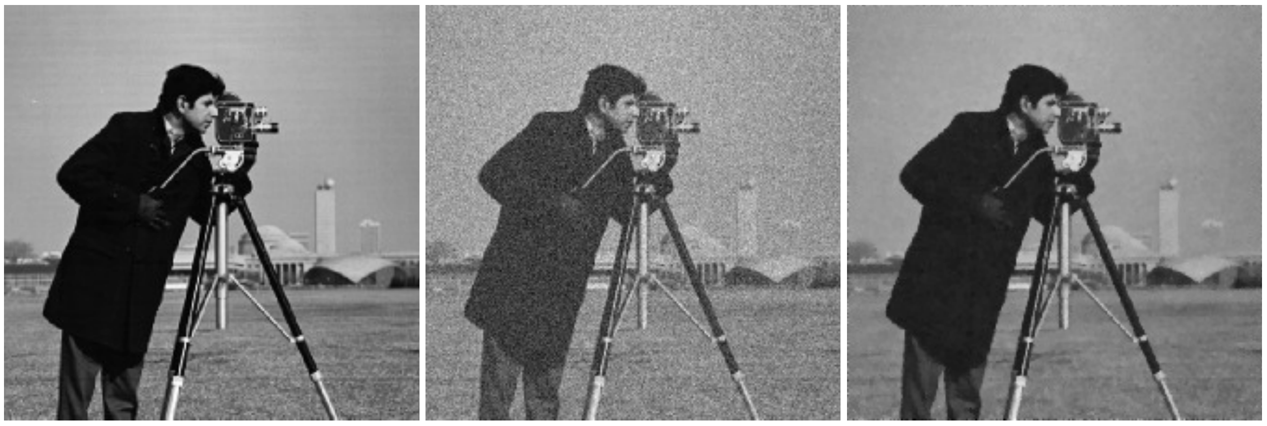
1. clearly $[-1, 1] \subset \Omega$, which makes the inequality in (2.1.2) valid; and
2. the pseudoinverse $\gamma_{\text{pinv}} \in \Omega^D$ is a solution to (27), so we can relax the dual problem to an unconstrained optimization:

$$\gamma_* \in \underset{\gamma}{\text{argmax}} \; -\frac{1}{2}||\gamma||^2_{\mathbf{D}+\frac{1}{w}\boldsymbol{\beta}\boldsymbol{\beta}'} + \gamma'\boldsymbol{\Lambda}\boldsymbol{\beta}. \tag{29}$$

When $\mathbf{D}$ is full rank, (29) can be solved trivially using the matrix inversion lemma. When $\mathbf{D}$ is rank-deficient, *i.e.*, when $x^{(n)} = x_j$ for at least one j, finding the solution requires either an iterative algorithm or a relatively computationally expensive direct method using, *e.g.*, an eigenvalue decomposition or the "matrix pseudoinverse lemma".[8] In the experiments below, we ran two iterations of the following minorize-majorize algorithm:

$$\gamma^{(m+1)} = \gamma^{(m)} + \left(\mathbf{D}_\epsilon + \frac{1}{w}\boldsymbol{\beta}\boldsymbol{\beta}'\right)^{-1}\left(-\left(\mathbf{D} + \frac{1}{w}\boldsymbol{\beta}\boldsymbol{\beta}'\right)\gamma^{(m)} + \boldsymbol{\Lambda}\boldsymbol{\beta}\right), \tag{30}$$

where $\mathbf{D}_\epsilon$ is a diagonal matrix with $[\mathbf{D}_\epsilon]_{dd} = \max\{\epsilon, [\mathbf{D}]_{dd}\}$ for a small $\epsilon > 0$. This recursion can be efficiently implemented using the matrix inversion lemma.



(a) Original image (b) Noisy (c) Denoised reference

Figure 2: Original, noisy and reference images for the experiments in Section 3.1.

3. EXPERIMENTS

We present two experiments below. One experiment uses a standard test image and an everywhere-differentiable potential function, and the second experiment uses a larger three-dimensional test image with the absolute value function. All experiments were run on a server with 48 GB of RAM and an NVIDIA Tesla C2050 GPU. All calculations were performed on the GPU.

3.1 Small image denoising problem

We corrupted the 512×512-pixel `cameraman` standard test image (pixel values $0 - 255$ gray levels) with additive white Gaussian noise with $\sigma = 20$ gray levels. We posed the denoising problem (1) with $\mathbf{W} = \mathbf{I}$, and the eight-neighbor ($D = 4$) edge preserving regularizer with $\beta_d = 10$, $\kappa_{dk} = 1$ for all d, k, and the Fair potential function,

$$\phi_{\text{Fair}}(x) = \delta^2(|x/\delta| - \log(1 + |x/\delta|)), \tag{31}$$

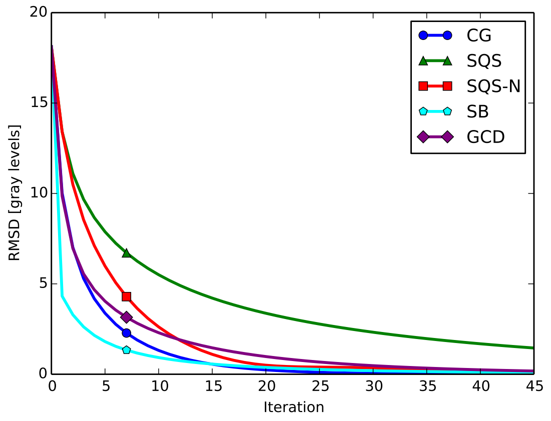
with $\delta = 10$ gray levels. The Fair potential is differentiable everywhere and admits a closed-form ISTA-like shrinkage update. We generated a reference image by running 1,024 iterations of a split Bregman algorithm.[6] Figure 2 shows the original image, noise-corrupted data and reference image.

We ran conjugate gradients (CG), a split Bregman algorithm[1] (SB), a majorize-minimize algorithm with separable quadratic surrogates[3] (SQS), the SQS algorithm with Nesterov's acceleration[11] (SQS-N), and the proposed group coordinate descent (GCD) algorithm and recorded the root mean square difference (RMSD) to the reference image at each iteration. Plots of RMSD as a function of iteration and time are given in Figure 3.

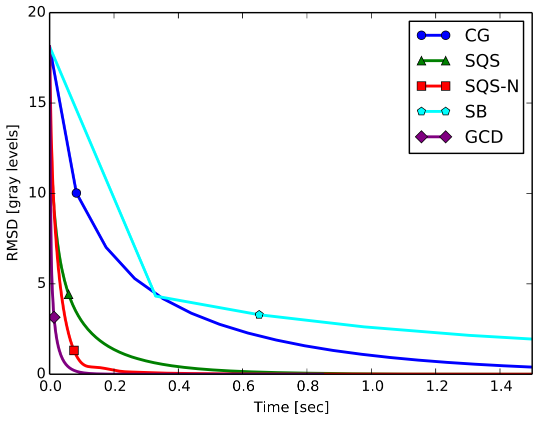
The results illustrate the trade-off between per-iteration accuracy and computational complexity. The algorithms with comparatively low computational complexity per iteration (SQS, SQS-N and GCD) converge more quickly in time than the more complex CG and split Bregman algorithms. We expect that in problems involving larger images, SB and CG would perform better. The GCD algorithm also converges more quickly than SQS-N in early iterations, though SQS-N overtakes GCD in later iterations. This early-iteration performance, combined with the fact that GCD requires only one copy of the image $\mathbf{x}^{(n)}$ (as opposed to the multiple copies that Nesterov's momentum requires), allow GCD to converge more quickly than SQS-N in time.

3.2 Three-dimensional denoising problem

To illustrate the proposed algorithm's behavior on a larger problem, we generated a $512 \times 512 \times 256$-pixel instance of the XCAT phantom,[13] a popular test image in medical imaging. The image is piecewise continuous with pixel values from $0 - 1700$ gray levels, and interesting features in the $800 - 1200$ gray-level window. We corrupted the phantom with additive white Gaussian noise with $\sigma = 20$ gray levels, and posed the denoising problem as above, using all 26 (three-dimensional) pixel neighbors, $\beta_d = 8$, and the absolute value potential function. Figure 5 shows the central slices of the original, noisy and reference images.

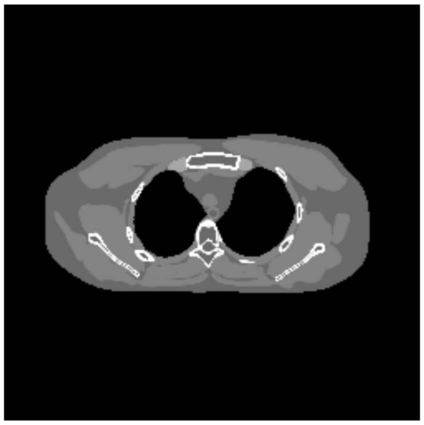


(a) RMSD by iteration

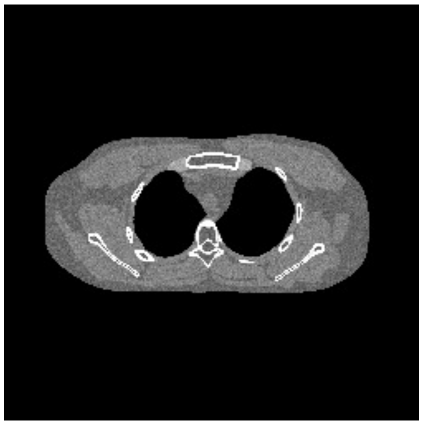


(b) RMSD by time

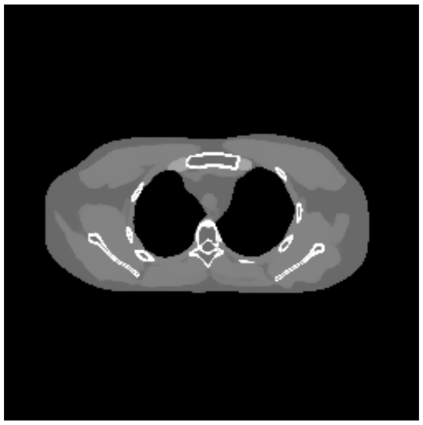
Figure 3: Convergence plots for the experiments in Section 3.1. The proposed GCD algorithm behaves competitively with SQS-N in per-iteration convergence, and converges very rapidly in time.



(a) Original image



(b) Noisy



(c) Denoised reference

Figure 4: Central slices of original, noisy and reference images for the experiments in Section 3.2, displayed on the 800 - 1200 gray level window.

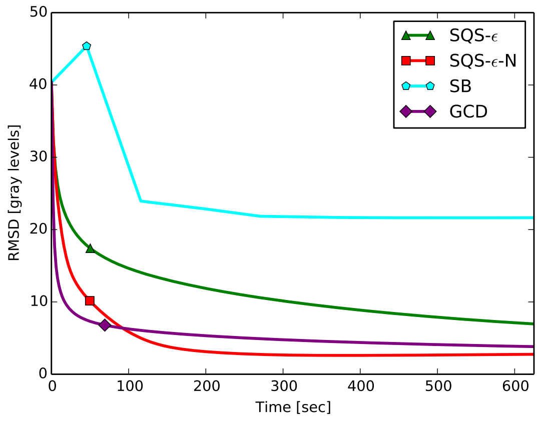
Although the noise in CT reconstruction is not additive white Gaussian, image-domain Gaussian-noise denoising problems (1) do appear in variable-splitting-based CT reconstruction algorithms.[9,10] Solving these problems efficiently is an important "inner step" in splitting-based CT reconstruction algorithms.

The absolute value function is not differentiable, so we used a corner-rounded SQS algorithm for comparison here. Instead of using the Huber curvature for the jth difference, we used

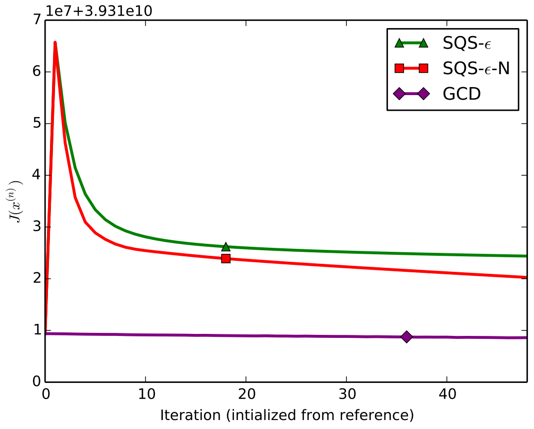
$$\tilde{c}_j^{(n)} = \frac{1}{\epsilon + |x - x_j|}, \tag{32}$$

with $\epsilon = 10^{-2}$. A quadratic "surrogate" with curvature $\tilde{c}_j^{(n)}$ does not lie above the absolute value function near the origin, so this substitution invalidates the majorization property of the SQS algorithm. However, *ad hoc* modifications like this are common enough in practice to merit consideration.

We ran the SQS algorithm with corner rounding (SQS-ϵ), SQS with Nesterov's acceleration (SQS-ϵ-N), the split Bregman algorithm (SB) and our proposed GCD algorithm. Figure 5a plots the RMSD of each algorithm over time. Due to the large disparity in the computational cost per iteration (see Table 1), we provide only a versus-time plot. Though SQS-ϵ-N converges rapidly in early iterations, the corner-rounding approximation punishes it in later iterations, whereas the



(a) RMSD by time



(b) Cost function by iteration, initialized from reference

Figure 5: Convergence plots for the experiments in Section 3.2. While SQS-ϵ-N converges rapidly, it is not as robust as GCD in later iterations.

	SQS-ϵ	**SQS-ϵ-N**	**SB**	**GCD**
Avg. time per iter (sec)	2.81	2.81	101.93	1.86

Table 1: Average computation time per iteration for the four algorithms run in the large image denoising experiment in Section 3.2.

proposed GCD algorithm converges towards the solution. Figure 5b plots the *cost function* per iteration after initializing from the reference image shown in Figure 4c to explore how the algorithms behave when initialized near the minimizer. The corner-rounded family of algorithms become nonmonotonic in cost, but because the proposed GCD algorithm uses a proper majorizer, it continues to very slowly decrease the cost function.

4. CONCLUSIONS AND FUTURE WORK

We presented group coordinate descent algorithms for edge-preserving image denoising that support a wide range of potential functions. The algorithms perform a iterated sequences of independent optimizations, compute no “global” operations like inner products, and require no additional auxiliary variables to be stored in memory. These qualities make the proposed algorithms remarkably well-suited to the GPU, which favors independent computation and conservative memory use. Future work will explore methods to increase parallelism by distributing these algorithms across multiple devices.

REFERENCES

[1] M. V. Afonso, José M Bioucas-Dias, and Mário A T Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Im. Proc.*, 19(9):2345–56, September 2010.

[2] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, 2 edition, 1999.

[3] H. Erdoğan and J. A. Fessler. Ordered subsets algorithms for transmission tomography. *Phys. Med. Biol.*, 44(11):2835–51, November 1999.

[4] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Trans. Im. Proc.*, 13(10):1327–44, October 2004.

[5] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange. Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction. *IEEE Trans. Med. Imag.*, 16(2):166–75, April 1997.

[6] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–43, 2009.

[7] P. J. Huber. *Robust statistics*. Wiley, New York, 1981.

[8] K. Kohno, M. Kawamoto, and Y. Inouye. A matrix pseudoinversion lemma and its application to block-based adaptive blind deconvolution for MIMO systems. *IEEE Trans. Circ. Sys. I, Fundamental theory and applications*, 57(7):1499–1512, July 2010.

[9] M. McGaffin and J. A. Fessler. Sparse shift-varying FIR preconditioners for fast volume denoising. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 284–7, 2013.

[10] M. G. McGaffin, S. Ramani, and J. A. Fessler. Reduced memory augmented Lagrangian algorithm for 3D iterative X-ray CT image reconstruction. In *Proc. SPIE 8313 Medical Imaging 2012: Phys. Med. Im.*, page 831327, 2012.

[11] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–76, 1983.

[12] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: A majorization-minimization approach. *Signal Processing*, 89(9):1683–93, September 2009.

[13] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui. Realistic CT simulation using the 4D XCAT phantom. *Med. Phys.*, 35(8):3800–8, August 2008.