

A Performance-Driven Study of Regularization Methods for GPU-Accelerated Iterative CT

Wei Xu and Klaus Mueller

Abstract—Iterative reconstruction algorithms with regularization can produce high-quality reconstructions from few views and even in the presence of significant noise. In the research presented here we focus on the particularities associated with the GPU acceleration of these. Specifically, we not only focus on reconstruction speed but also on reconstruction quality which reveals a number of important interaction effects and trade-offs. To obtain this insight, we use exhaustive benchmark tests to determine the optimal settings of the various parameters associated with the algorithm, here OS-SIRT. The same mindset we also apply in the selection of the most GPU-amenable regularization mechanism, where we compare the traditionally used TVM filter with the less frequently used bilateral filter, which we find to be a viable and cost-effective means for regularization.

Index Terms—Iterative Reconstruction, Ordered Subsets, Computed Tomography, GPU, Bilateral Filter, Total Variation Minimization

I. INTRODUCTION

Iterative reconstruction methods have gathered significant interest in recent years since they can cope well with limited projection sets and noisy data. These scenarios occur most often in low-dose CT, where one seeks to either limit the dose per projection, or the number of projections overall, or both. Low dose CT has been a response to growing concern about the high radiation dose delivered to a patient in multi-slice X-ray CT, but the noise associated with reduced radiation dose decreases SNR and the few-view scenario can lead to prominent streak artifacts in the reconstruction. Both can obliterate the features of interest and generally make the CT image hard to read. While exact or approximate exact CT reconstruction methods do not work well under these conditions, iterative methods can produce acceptable results. These methods, however, suffer from high computational effort, which has prevented a deployment in routine clinical applications so far as these computational demands cannot be met by reasonable CPU-based platforms.

High-performance graphics chips (GPUs) are poised to provide a breakthrough in this problem. In recent work, we have shown that with just a single such board one can filter and back-project cone-beam projections faster (at 50 projections/s) than they can be produced by a modern flat-panel gantry, enabling a new paradigm we call *streaming CT* [8]. Further, in earlier work [6] we have also shown that reconstruction algorithms, both iterative and analytical, can typically be broken down into blocks, which can be accelerated individually on these platforms using dedicated programs (called *shaders*).

In this current work we specifically address the acceleration of iterative optimization algorithms for the purpose of low-dose CT with reduced sets of noisy projections. Our framework alternates projection-space prediction-correction with object-space regularization. The former ensures adherence of the solution to the data, while the latter seeks to drive the former to a more plausible solution. Our prominent aim is to make this procedure amenable to GPU-acceleration.

II. RELATED WORK

We chose algebraic reconstruction as the predictor-corrector method. In expectation maximization (EM), ordered subsets (OS) have long been known to speed up convergence speed, with larger numbers of subsets converging faster. In recent work [7], we have introduced the idea of using ordered subsets also for algebraic settings, giving rise to OS-SIRT. In this scheme SIRT and SART form two extremes, with SIRT having just one and SART having M subsets (M being the number of projections). We showed that while on the CPU there is little difference in the running time per iteration, on the GPU an iteration with SART is typically the slowest, due to the many projection-backprojection context switches which disturb parallelism and data flow. This has significant implications for the overall reconstruction wall clock time, where SART, in the noise-free case, is no longer the fastest method (which it is on the CPU). This effect has also been observed by other authors [2], but there the focus was solely on reconstruction speed. In contrast, we have found, in the present work, that once reconstruction quality is considered as well, these relationships are altered and SART becomes more competitive again. In addition to this insight, we also address the issue of noise, and revisit GPU OS-SIRT under these new circumstances.

For few-view, limited-angle, and noisy projection scenarios, the application of regularization operators between reconstruction iterations seeks to tune the final or intermediate results to some *a-priori* model. A simple regularization scheme is to enforce positivity. In [4], the method of total variation (TV) was proposed for additional regularization (in conjunction with POCS reconstruction). TV minimization (TVM) has the effect of flattening the density profile in local neighborhoods and thus is well suited for noise and streak artifact reduction. Based on the assumption of a relatively sparse gradient object, the method has been shown to work quite well under a variety imperfect imaging situations, yet this assumption may not be realistic in general. In computer vision, two prominent TV models are frequently used, that is, the ROF model and the TV-L¹ model [3]. A number of variational algorithms have been designed as a minimizer of the energy functional of the models. They are

Wei Xu and Klaus Mueller are with the Computer Science Department, Stony Brook University, Stony Brook, NY 11790 USA (phone: 631-632-1524; e-mail: {wxu, mueller}@cs.sunysb.edu). Funding was provided by NSF CCF-0702699, CCF-0621463 and NIH EB004099-01.

mainly based on solving the associated Euler Lagrange differential equation with optimization techniques. These methods are well suited for the removal of noise and other unwanted fine scale details while preserving edges. However, in the context of high performance computing, due to its iterative procedure TVM is quite time-consuming, even when accelerated on GPUs.

III. METHODOLOGY

We aim to devise a method that is not iterative but has the same goals as TVM, that is, the reduction of local variations (noise, streaks) while preserving coherent local features. The bilateral filter [5] is such a method. It combines a range filter with a domain filter, giving rise to a non-linear filter designed for edge-preserving smoothing. When based on the Gaussian function, two parameters are required, σ_r and σ_d , to control the weight of each filter. We then compare this filter with a TVM method [1] to explore its performance under different scenarios.

An important aspect of Ordered Subsets-EM (OS-EM) is that it balances noise suppression with convergence speed – typically in the presence of noise using a smaller number of subsets leads to faster convergence and better results, due to the inherent smoothing provided by the larger projection sets. These issues are also relevant for our GPU-accelerated OS-SIRT, but with the added constraints imposed by the GPU hardware architecture. Finally, in contrast to EM, algebraic methods also offer a relaxation factor λ which has a great effect on convergence speed. In [7], a simple linear selection scheme for λ (as a function of subset size) was used, which we found sub-optimal in the current work. We therefore propose a scheme that determines the optimal setting of λ (and subset number) based on an exhaustive set of benchmark tests under different noise conditions. This framework is more detailed described in a companion publication [9].

A. OS-SIRT

The correction update for OS-SIRT is given as follows:

$$v_j^{(k+1)} = v_j^{(k)} + \lambda \sum_{p_i \in OS_s} \frac{p_i - r_i}{\sum_{l=1}^N w_{il}} \quad r_i = \sum_{l=1}^N w_{il} \cdot v_l^{(k)} \quad (1)$$

where the weight factor w_{ij} determines the contribution of a voxel v_j to a ray r_i (starting from a projection pixel p_i) and is given by the interpolation kernel. This equation is a generalization of the original SART and SIRT equations to support any number of subsets [7]. The p_i are the pixels in the M/S acquired images that form a specific (ordered) subset OS_s , where $1 \leq s \leq S$ and S is the number of subsets.

B. Bilateral Filter

The bilateral filter non-linearly averages similar and nearby pixels values. To achieve effective and efficient computation, the averaging only occurs inside a fixed window area. It consists of two filter components, the domain filter and the range filter:

$$h(x) = \frac{\sum_{\varepsilon \in W} f(\varepsilon) c(\varepsilon, x) s(f(\varepsilon), f(x))}{\sum_{\varepsilon \in W} c(\varepsilon, x) s(f(\varepsilon), f(x))} \quad (2)$$

Here, W is the window centered at x , ε and x represent the spatial variables, f is the input image, and c and s are the measured closeness and pixel value similarity, respectively. The geometric closeness function acts as the domain filter controlling the contribution according to spatial distance, while the pixel value similarity function acts as a range filter generating very low weights for dissimilar pixel values. Normalization forces the sum of pixel weights to 1. In our work, we model the closeness and similarity functions as Gaussians:

$$c(\varepsilon, x) = e^{-\frac{\|\varepsilon - x\|^2}{2\sigma_d^2}} \quad s(\varepsilon, x) = e^{-\frac{(f(\varepsilon) - f(x))^2}{2\sigma_r^2}} \quad (3)$$

where σ_r and σ_d control the amount of smoothing.

The implementation of GPU-accelerated bilateral filtering is as follows. The rendering target is a texture of the size of the reconstructed image, with image texture and other parameters (size of image, σ_r , σ_d , etc.) passed into the GPU. We avoid the expensive evaluation of the exponential function by pre-computing both closeness and similarity functions and storing them into two 1-D lookup textures. We implemented bilateral filtering both in 2D and 3D.

C. Total Variation Minimization (TVM)

We also implemented a TVM algorithm [1] to compare it with our bilateral filter framework. The TVM solution is obtained by minimizing the following energy functional:

$$\min_u \left\{ \frac{\|u(x) - f(x)\|^2}{2\lambda} + \sum_{x \in \Omega} |\nabla u(x)| \right\} \quad (4)$$

where Ω is the image domain, x is the spatial variable, f is the input image, u is the sought-after solution and λ is a parameter controlling the level of smoothing. The TV of u is:

$$\sum_{\Omega} |\nabla u| = \sum_{\Omega} \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2} \quad (5)$$

In this equation, x and y are the horizontal and vertical coordinates, respectively. The minimization is transformed to its dual formulation, and a semi-implicit gradient descent algorithm is used to compute the nonlinear projection of f . The solution u is then obtained after convergence, with τ set to some value constraint:

$$u = f - \lambda \operatorname{div} p^n \quad p^{n+1} = \frac{p^n + \tau \nabla(\operatorname{div} p^n - f/\lambda)}{1 + \tau |\nabla(\operatorname{div} p^n - f/\lambda)|} \quad (6)$$

Here, div is the divergence. In practice, when $\tau \leq 1/4$ the algorithm converges.

D. Regularized OS-SIRT

In our new regularized OS-SIRT, bilateral filtering is applied after each iteration (after backprojecting all subsets). This removes artifacts at the very beginning when the errors are just generated and thus steers the reconstruction towards more plausible and favorable solution regions. Since the target texture (to be filtered) is already in GPU memory, this operation does not require any expensive texture upload/download operations between the CPU and GPU.

IV. RESULTS

Our experiments were conducted on an NVIDIA GTX 280 GPU, programmed with GLSL and an Intel Core 2 Quad CPU @ 2.66GHz and 2.67GHz. We group our results into two sections: (1) the OS-SIRT results showing the relationship between noise levels and parameters settings, and (2) the performance of our GPU-accelerated bilateral filter using both Cg and CUDA and the reconstruction results using bilateral filter and total variation minimization.

A. OS-SIRT with noisy data

We used the 2D Baby Head test image (size 256^2) to evaluate the performance of the different reconstruction schemes. We obtained 180 projections at uniform angular spacing of $[-90^\circ, +90^\circ]$ in a parallel projection viewing geometry. We then added different levels of Gaussian noise to the projection data to obtain SNRs of 15, 10, 5, and 1. Fig. 1 presents the best reconstruction results (using the correlation coefficient CC between original and reconstructed image), for each SNR, at the smallest wall-clock time.

The optimal settings greatly depend on the particular imaging situation at hand, such as SNR, total number of projections and their angular range, the imaged object, scanner, etc. Fig. 2 presents results on the influence of SNR. The plot gives quantifying hints on how to pick the best-performing number of subsets and the associated λ (to obtain the best possible quality within the smallest time), for each expected SNR level. For example, we observe that low SNR requires a low number of subsets. As for the relaxation factor λ , it is related to both subset number and noise level. For each noise level, the curve of λ is approximately piece-wise linear with a turning point at some subset number. For example, the λ values for SNR 10 are 1 from subset number of 1 to 60, then decreasing until hitting the lowest value of 0.4 at subset number of 180. This is a strong departure from the linear model used on [7] – a higher λ will lead to faster convergence and confirmed by our exhaustive benchmark tests we know it also leads to more accurate results.

B. Bilateral Filter Regularized OS-SIRT

We tested the speed of both 2D and 3D bilateral filters with different sizes of images and windows on both CPU and GPU (using Cg). Table I shows that speedups of more than two orders

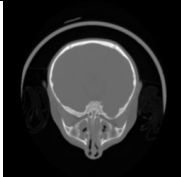
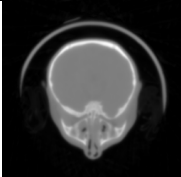

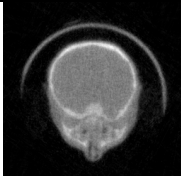
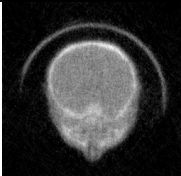
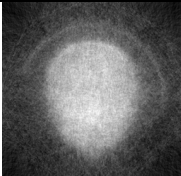
| | | |
|---|---|---|
|  |  |  |
| Original | Noise-free | SNR15 |
| CC | 0.99 | 0.98 |
| Time (s) | 0.045 | 0.035 |
|  |  |  |
| SNR10 | SNR5 | SNR1 |
| 0.98 | 0.96 | 0.82 |
| 0.035 | 0.027 | 0.012 |

Fig. 1. Reconstructions obtained with different SNR levels for the Baby head test image

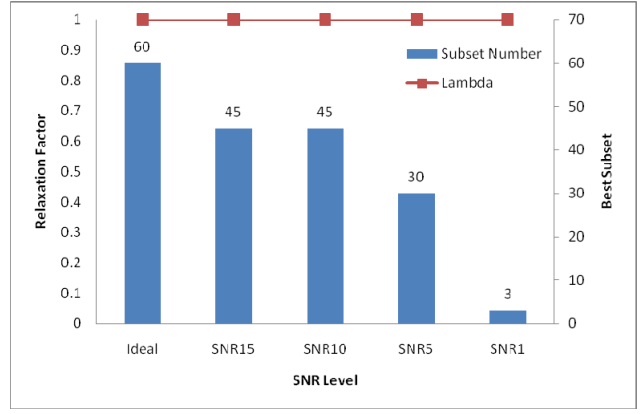


Fig. 2. Best performing (both in terms of time and image quality) subset number and relaxation factor as a function of imaging condition, here SNR

TABLE I
WALL CLOCK TIME (IN S) OF GPU VS. CPU BILATERAL FILTER

| Test size | Window size | CPU time (s) | GPU (Cg) time (s) | GPU (CUDA) time (s) |
|-------------|-------------|--------------|-------------------|---------------------|
| 256x256 | 11x11 | 0.622 | 0.007 | 0.005 |
| | 31x31 | 4.891 | 0.013 | 0.011 |
| | 61x61 | 18.626 | 0.037 | 0.033 |
| | 91x91 | 39.031 | 0.069 | 0.066 |
| | 11x11 | 2.652 | 0.011 | 0.007 |
| 512x512 | 31x31 | 19.998 | 0.038 | 0.032 |
| | 61x61 | 74.319 | 0.119 | 0.112 |
| | 91x91 | 164.065 | 0.253 | 0.241 |
| | 11x11 | 10.811 | 0.033 | 0.017 |
| | 31x31 | 84.618 | 0.133 | 0.098 |
| 1024x1024 | 61x61 | > 300 | 0.452 | 0.368 |
| | 91x91 | > 300 | 0.983 | 0.823 |
| | 3x3x3 | 46.831 | 0.492 | N/A |
| 256x256x256 | 7x7x7 | 592.969 | 1.535 | N/A |
| | 11x11x11 | > 600 | 4.823 | N/A |

of magnitude can be obtained by using the GPU. For 2D images, we also implemented a CUDA version of our scheme.

To gauge the performance of the regularized reconstruction for both the few-view and the noise (SNR=10) scenario, we used the NIH Visible Human dataset at 512^3 resolution. We ran SART with 8 iterations for the noise-free few-view case. The filter window size was fixed to 11. Fig. 4 shows one slice of the

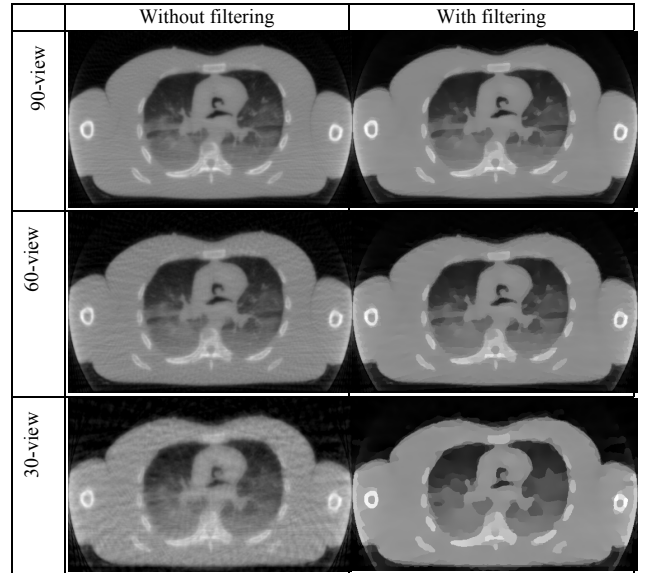


Fig. 4. Comparison of Bilateral Filtering for the noise-free, few-view case

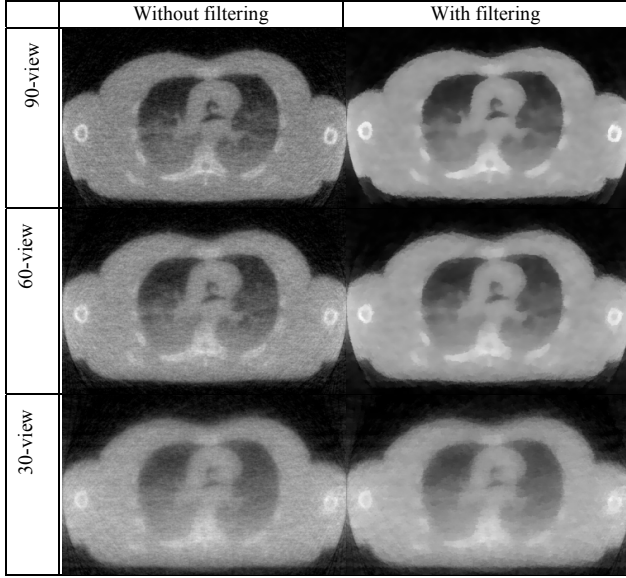


Fig. 5. Comparing bilateral filtering for the noisy (SNR=10), few-view case.

reconstructed volume with and without filtering, respectively, for reconstructions from 90, 60, and 30 views. We notice that SART is already well suited for the few-view reconstruction. For the regularized reconstructions we tested a number of combinations of representative σ_r and σ_d and selected the best results. In particular the 30-view reconstruction shows prominent streaking artifacts, which can be avoided by intermediate bilateral filter regularization.

Fig. 5 shows the results for the noisy few-view case, after 5 iterations. Like in the noise-less case we observe that the salient features are well preserved in both size and shape.

Finally, Table II lists the GPU-accelerated reconstruction time required for one SART iteration, for the Visible Human dataset at 512^3 resolution for both 180 and 30 projections. The 1-ch time uses only the R-channel of the GPU hardware, while the 4-ch time uses all 4 (RGBA) channels in parallel. Using 4 channels yields a 2.5-fold speedup, while regularization with bilateral filtering (BF) adds only a moderate time overhead.

C. Bilateral Filter vs. Total Variation Minimization

We tested the same slice with identical settings for both the few-view (30 projections) and the noisy few-view (30 projections and SNR=10) case and show the results in Fig. 6.

We observe that for the noise-free case, bilateral filtering achieves similar results than TVM (maybe even slightly better). However, TVM works better for the noisy case. This is not surprising since for TVM the energy functional imposes a constraint over the image, while bilateral filtering just averages the neighboring values which cannot eliminate all noise for higher noise levels. Nevertheless, both successfully preserve salient features and remove noise and streaking artifacts.

From the perspective of high performance computing, the

TABLE II
TIME FOR ONE GPU-ACCELERATED SART ITERATION (512^3 VOLUME)

| #proj | 1-ch | 1-ch w/ BF | 4-ch | 4-ch w/ BF |
|-------|---------|------------|--------|------------|
| 180 | 91.8137 | 94.9598 | 34.789 | 34.944 |
| 30 | 21.942 | 25.6891 | 9.21 | 10.124 |

The 1-ch and 4-ch accelerate the reconstruction with 1 (R) or 4 (RGBA) color channels, respectively. A NVIDIA GTX 280 GPU was used.

bilateral filter is a better choice. Table I shows that the computation time is less than 1s. Although a GPU-accelerated version of TVM exists [3], once the parameter λ grows larger, which is needed for very noisy data, the computation time (usually $\gg 1$ s) is still far greater than with the bilateral filter.

V. CONCLUSIONS

We have demonstrated that careful parameter-tuning taking into account reconstruction quality results in better speed performance. This is particularly true for ordered subsets approaches in the presence of adverse data scenarios, such as noise and sparse views. We also demonstrated that bilateral filtering represents a viable option for regularization compared with Total Variation Minimization (TVM), with the added advantage that it accelerates very well on GPUs.

REFERENCES

- [1] A. Chambolle, "An algorithm for total variation minimizations and applications," *J. Math. Imaging Vis.*, 20(1-2):89-97, 2004.
- [2] B. Keck, H. Hofmann, et al., "GPU-accelerated SART reconstruction using the CUDA programming environment," *Proc. SPIE*, 7258, 2009.
- [3] T. Pock, M. Unger, D. Cremers, H. Bischof, "Fast and Exact Solution of Total Variation Models on the GPU," *Comp. Vis. Patt. Reco*, 1-8, 2008.
- [4] E. Y. Sidky, C.-M. Kao, X. Pan, "Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT," *J. X-ray Sci.Tech.* 14: 119-139, 2006.
- [5] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images," *Intern. Conf. Comp. Vis.*, pp. 839-846, 1998.
- [6] F. Xu, K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. on Nucl. Sci.*, 52: 654-663, 2005.
- [7] F. Xu, K. Mueller, et al. "On the efficiency of iterative OS Reconstruction algorithms for acceleration on GPUs," *MICCAI Workshop on High-Performance Medical Image Computing*, New York, 2008.
- [8] F. Xu, K. Mueller, "Real-time 3D CT reconstruction using commodity graphics hardware," *Phys. Med. Biol.*, 52: 3405-3419, 2007.
- [9] W. Xu, K. Mueller, "Learning effective parameter settings for iterative CT reconstruction algorithms," *Fully 3D Reconstruction 2009*.

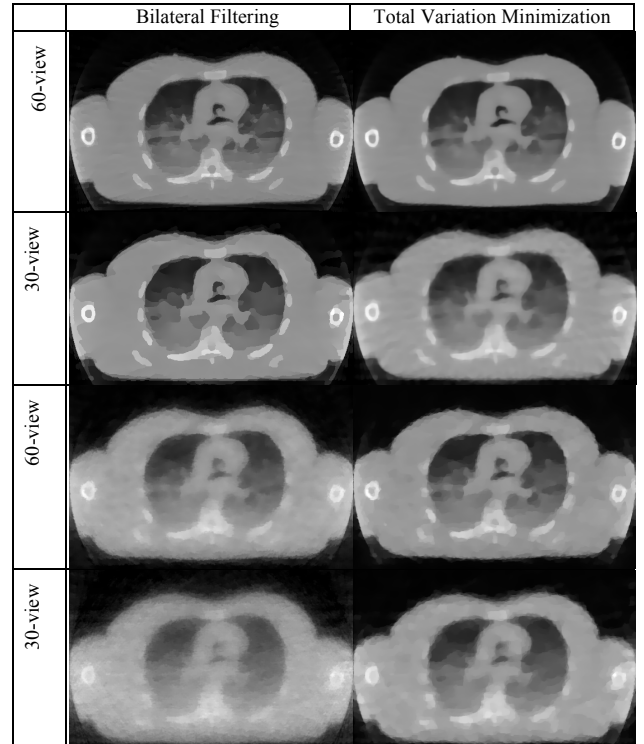


Fig. 6. Bilateral Filtering vs. TVM: (first two rows): the noise-free few-view case; (last two rows): the noisy (SNR=10) few-view case.