A Comparative Study of Neighborhood Filters for Artifact Reduction in Iterative Low-Dose CT

Wei Xu, Sungsoo Ha, Ziyi Zheng and Klaus Mueller

Abstract— **Iterative CT algorithms have become increasingly popular in recent years. They have been found useful when the projections are limited in number, irregularly spaced, or noisy, which are often encountered in low-dose CT imaging. One way to cope with the associated streak and noise artifacts is to interleave a regularization objective into the iterative reconstruction framework. In this paper we investigate a number of non-linear neighborhood filters within an iterative CT framework, OS-SIRT, and compare them with total variation minimization (TVM). We find that the Non-Local Means (NLM) filter provides the best performance, in particular its patch-based variant. Further, we also compare a scheme that exploits an artifact-free reference image for even better regularization performance. Finally, we also compare the studied filters in terms of their computational efficiency with acceleration on modern GPUs.**

I. INTRODUCTION

Low dose CT imaging has been gaining considerable momentum in recent years. However, low-dose CT leads to noisy and sparse X-ray projections, which subsequently lead to significant noise and streak artifacts in the reconstructions. In these adverse conditions iterative reconstruction algorithms are more favorably applied, especially when combined with regularization. Here, the method of Total Variation Minimization (TVM), has become rather popular and has been used in many frameworks, such as ASD-POCS [5]. However, TVM is an iterative global optimization algorithm and can be costly in compute, lessen practicality in clinical practice.

We study if non-iterative filters that only operate in a local neighborhood can lead to improved results. An advantage here is that they also lend themselves very well to GPU acceleration. We specifically study and compare the bilateral filter (BLF) [6] and the non-local means filter (NLM) [1]. While the use of local neighborhood filters within an iterative CT reconstruction framework is not conceptually new, this paper's contribution is (1) a comparison of these both in terms of quality and speed, and (2) their extension into an adaptive form [3] and one that uses a prior image of the patient [11] .

II. OVERVIEW

Our reconstruction framework is fully iterative using our OS-SIRT pipeline [7][9] for reconstruction, interleaving regularization within each iteration. The regularization enforces constraints in the object domain, such as local smoothness and coherent edges, while the reconstruction ensures fidelity with the acquired data. This type of pipeline has also been used by others, but with different reconstruction algorithms and regularization schemes. All operations are accelerated on the GPU.

A. Regularization as a denoising task

In the context of mitigating artifacts in CT reconstruction, regularization is similar to the process of *denoising* in image processing. In fact, the notion of noise is quite general and can include for example streak artifacts. We may distinguish between two families of denoising strategies: (i) global optimization and (ii) local filtering. Both can be iterative, where the former seeks to improve some global objective function and the latter repeats the filtering, possibly guided by some error criterion and varying parameters along the way. In fact, the two families can be unified into a mathematical framework which gives them a common theoretical underpinning. Elad [2] shows that both derive from a solid theory of statistical estimators and regularization. More specifically, the bilateral filter emerges from the Bayesian approach as a single iteration of the Jacoby normalized diagonal steepest descent algorithm.

For the remainder, we shall adhere to the terminology of image processing where the goal is to reduce artifacts in images.

B. Regularization by local neighborhood filtering

Local neighborhood filters have become popular in image processing since they can achieve better computational performance and also afford local control. They are non-iterative (although they can be applied repeatedly) and are based on pixel-wise operations over a small neighborhood.

In contrast to global optimization such as TVM, for nonlinear neighborhood filters (NNF) the updated value at a pixel x is determined by a weighted sum of a functional mapping α of its local neighborhood W_x. This typically non-linear function α takes into account both spatial and value discrepancies with respect to x, as expressed in the following equation:

$$NNF(x, f, W_x, \alpha) = \frac{\sum_{t \in W_x} \alpha(x, t, f(x)) f(x+t)}{\sum_{t \in W_x} \alpha(x, t, f(x))} \qquad (1)$$

The normalization forces the sum of pixel weights to 1. The window area W_x defining the local neighborhood can vary in size for pixels at different positions. To compute the weights of the neighborhood, a distance metric measures the similarity between the pixel at $x+t$ and the central pixel at x. Next, we use this general notation to express all filters we have studied.

The bilateral filter (BLF) [6]: The filter only considers a fixed sized neighborhood around the target pixel x, and the weighting function α_{BLF} is the product of spatial distance weight c_d and range distance weight s_r:

$$\begin{aligned} \alpha_{BLF}(x, t, f) &= c_d(t) s_r(f(x), f(x+t)) \\ c_d(t) &= G_{\sigma_d}(\|t\|) \\ s_r(f(x), f(x+t)) &= G_{\sigma_r}(|f(x) - f(x+t)|) \end{aligned} \qquad (2)$$

where $G_\sigma(x)$ is the Gaussian kernel

$$G_{\sigma_r}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad (3)$$

and σ_d and σ_r control the amount of smoothing. The function c_d acts as a domain filter to ensure spatial closeness to x such that far away pixels have no effects. On the other hand, the function s_r acts as a range filter to ensure value closeness to $f(x)$ such that the values of pixels from different nearby materials cannot

Wei Xu and Klaus Mueller are with the Visual Analytics and Imaging Lab, Computer Science Department, Stony Brook University, Stony Brook, NY 11777 USA (phone: 631-632-1524; e-mail: {wxu, mueller}@cs.sunysb.edu).

Partial support came from NSF grants 1050477, 0959979 and 1117132.

diffuse into the material represented by *x*. Similar to anisotropic diffusion it ensures that sharp edges are well preserved.

The non-local means filter (NLM) [1]: Based on the assumption that there is a high degree of redundancy in a given image, the NLM filer consults similar pixel neighborhoods (called *patches*) in disjoint image regions and average their contributions for a more stable outcome:

$$f_{NLM}(x) = NNF(x, f, W, \alpha_{NLM}) \tag{4}$$

As such, the variable *t* parameterizes the offset within the search window as before. In order to gauge the similarity of a neighborhood patch at *x*+*t* with the neighborhood at *x*, the corresponding pixel-differences are weighted by a Gaussian kernel G_a with standard deviation σ_a, inside the patch area *P*:

$$\alpha_{NLM}(x,t,f) = \exp\left(-\frac{\sum_{p\in P} G_{\sigma_a}(t) \left| f(x+p) - f(x+t+p) \right|^2}{h^2} \right) \tag{5}$$

h acts as a filtering parameter which, when increased, allows for more dissimilar patches to contribute to the smoothing.

The adaptive NLM filter (ANLM) [3]: In the NLM filter, the search window has typically a constant, pre-set size throughout the image. However, picking a good size of the NLM search window can be challenging, especially when noise levels and patterns are not spatially invariant, which is most often the case. Hence it is more appropriate to locally adapt the window size. Kervrann and Boulanger [3] describe an iterative approach (with usually less than 4 iterations) that adaptively grows the local search window to incorporate neighborhood statistics at an increasing level of scale. The expansion is terminated once the deviation bias of the weighted smoothing grows too large (i.e. the local estimates diverge at increasing scale). We call this approach *adaptive NLM (ANLM)* since it also determines the weight of a neighborhood pixel via its patch similarity (the patch size itself is fixed) – however, the similarity measure changes as the iterations proceed. At each iterative step *i*, the smoothed image $f_{ANML(i)}(x)$ and the variance $\sigma_i^2(x)$ at position *x* of a neighborhood are calculated using the adaptive weights w_i as:

$$\begin{aligned} f_{ANLM(i)}(x) &= \sum\nolimits_{t\in W_{x,i}} w_i(x,t,f_{ANLM(i-1)})\, f(x+t) \\ \sigma_i^2(x) &= \sigma_0^2 \sum\nolimits_{t\in W_{x,i}} w_i(x,t,\mu_{i-1})^2 \end{aligned} \tag{6}$$

Here, $W_{x,i}$ is the current neighborhood size and σ_0 is the initial standard deviation, estimated from the input image (more detail is provided in [3]). The current $f_{ANML(i)}(x)$ and $\sigma_i^2(x)$ then serve as input to compute the weights for the next iteration: (7)

$$w_i(x,t,\mu_i) = \frac{\exp(-dist(f_{ANLM(i-1)}(x), f_{ANLM(i-1)}(x+t))/h^2)}{\sum_{t\in W_{x,i}} \exp(-dist(f_{ANLM(i-1)}(x), f_{ANLM(i-1)}(x+t))/h^2)}$$

$$dist = \sum\nolimits_{p\in P} (f_{ANLM(i-1)}(x+p) - f_{ANLM(i-1)}(x+t+p))^2 \left(\tfrac{1}{2\sigma_{i-1}^2(x+p)} + \tfrac{1}{2\sigma_{i-1}^2(x+t+p)}\right)$$

There are five parameters: the initial noise variance $\sigma_0^2(x)$, the patch size *P*, the parameter *h* (and a factor ρ), and the maximal number of iterations *N*. The $\sigma_0^2(x)$ can be automatically generated through robust estimation in the image. For *P*, we found a size of 7×7 practical in most cases. The other parameters were relatively insensitive to change within a normal range.

The reference-based NLM filter (RNLM) [11]: Often prior scans of the patient are available which could be used as an external site for patch-based neighborhood matching. This gives rise to the following equations:

$$\begin{aligned} & f_{RNLM}(x) = NNF_{RNLM}(x, f, f_r, f_{rn}, W, \alpha_{RNLM}) \\ & NNF_{RNLM}(x, f, f_r, f_{rn}, W_x, \alpha) = \frac{\sum_{t\in W_x} \alpha(x,t,f,f_{rn}(x))\, f_r(x+t)}{\sum_{t\in W_x} \alpha(x,t,f,f_{rn}(x))} \\ & \alpha_{RNLM}(x,t,f,f_{rn}) = \exp\left(-\frac{\sum_{p\in P} G_{\sigma_a}(t)\, | f(x+p) - f_{rn}(x+t+p) |^2}{h^2} \right) \end{aligned} \tag{8}$$

Here, f_r is the reference image containing similar features as the image *f* to be denoised, f_{rn} is the same reference image now augmented with similar artifact statistics, and *t* is some offset to *x* that locates areas with similar features. The value of *t* could be determined by a rough registration or an approximate feature matching using block-based histograms, etc. Our work in [11] used this filter to restore an image reconstructed with filtered backprojection. In the research presented here we use it as a regularization operator in an iterative reconstruction scheme.

III. Results

We interfaced an NVIDIA GTX 480 GPU with an Intel 2 Quad CPU @ 2.66GHz host processor. For testing, we employed the NIH Visible Human's torso (size 256^3) which has a prominent spine structure with different bone sizes and small structures and a brain dataset (NIH Visible Human brain, size 256^3) which also has some finer structures. We used a high-quality X-ray simulator to obtain various projection sets for torso and brain.

In our experiments, we explore the various regularization schemes within the interleaved reconstruction pipeline, for both the few-view and the noisy projection scenarios. We found the best parameters for each filter via experimentation.

A. Qualitative and quantitative comparison: torso dataset

We simulated 180 uniformly distributed projections over a half-circle trajectory. For the few-view case we selected every 9th projection from the set, yielding a total of 20 projections. Then, for each of the 4 regularization schemes (BLF, TVM, NLM, and ANLM), we interleaved regularization with OS-SIRT (10 subsets) and ran this pipeline for a total of 200 iterations. For the second series of experiments, we added significant Gaussian noise (SNR=10) to all 180 projections and ran the same pipeline again, but this time for only 20 iterations since this yields about the same number of updates as the few-view case (this much noise typically also causes the reconstruction procedure to diverge when the noisy projections are not pre-filtered).

The results of these two experiments are shown in Figure 1 along with the corresponding parameter settings and the best E-CC metric scores they could achieve. The E-CC is a perceptual quality metric and was introduced in [10] – it measures the cross-correlation (CC) of an edge-filtered image. We provide ROI zoomed results for two critical regions, spine and lung. The left-most full-body reconstructions were obtained without regularization. The first observation we make is that streaks seem to be easier to remove than heavy noise – the E-CC obtained with regularization is roughly 12-15% higher for the former for all regularization schemes. We also readily observe that all filters can reduce streaks and noise, recovering some structural parts which can be hardly seen in the non-filtered result. In the following we focus our detailed discussion on the spine – similar observations can also be made for the lung.

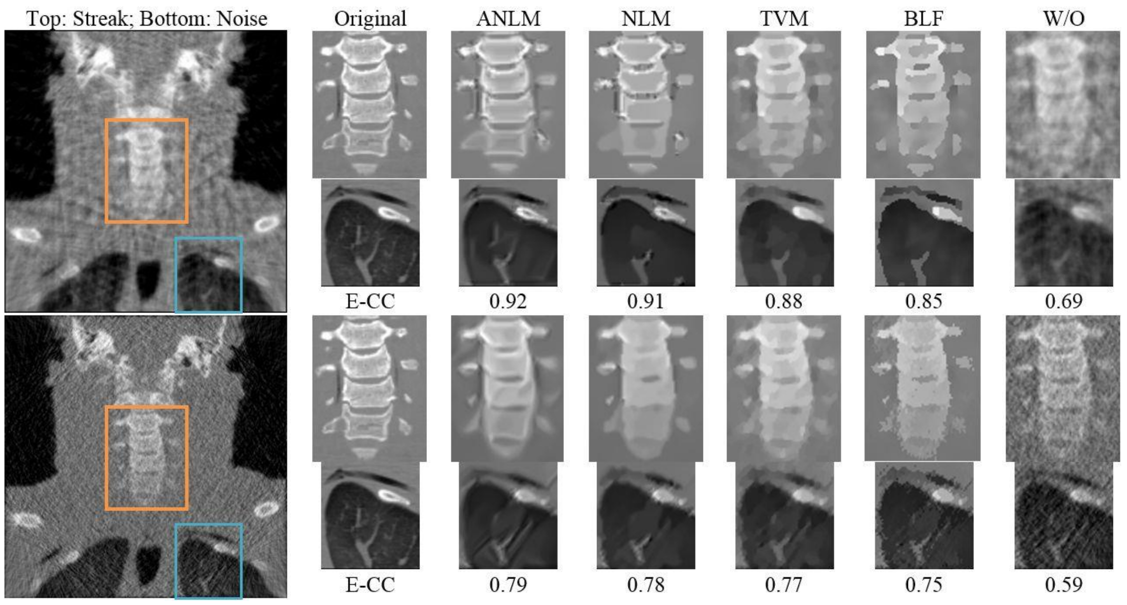


Figure 1. Torso dataset, reconstructed with the interleaved regularization pipeline both for the few-view (top, 20 projections) and noisy (bottom, 180 projections, SNR 10) scenarios. Zoomed results for two critical regions are shown, indicated by the orange (spine) and blue (lung) boxes in the left-most reconstructions obtained without regularization. The reconstructions appear ordered according to their E-CC scores.

The BLF and TVM perform quite similarly, but while the BLF keeps sharper edges and provides better streak removal than TVM, it also gives the image a more binary look signified by abrupt changes along adjacent varying-intensity areas. TVM, on the other hand, has smoother transitions here, and it also seems to perform better with noise. However, neither of the filters is able to recover more subtle features.

The two NLM-based methods successfully master the problems encountered with BLF and TVM – both recover the gaps separating the individual vertebrae. The shape and structure of the vertebrae is also better described, delineating the bony shell around the vertebrae body well. However, for the noisy projections case, the ANLM filter is the only one to do so.

B. *Qualitative comparison: brain dataset*

We used the same conditions as for the torso dataset (20 projections for the few-view case, 180 projections with SNR 10 Gaussian noise added for the noisy case) and the same regularized construction strategy (200 iterations with interleaved OS-SIRT 10 for the few-view case, 20 iterations of OS-SIRT 10 for the noisy projection case).

Figure 2a shows the results we obtained for the few-view case. We observe that in terms of sharpness and detail preservation ANLM and NLM have similar outcomes, but that the ANLM better preserves the small structures pointed by the arrow in the Original image. We further observe that the BLF produces slightly sharper and detailed images than TVM, but not quite as good as the NLM filter. The figure also examines the result obtained with the RNLM filter. Here we explored two different strategies (i) apply the reference image-based regularization only once (after the final iteration step), and (ii) apply it in an interleaved fashion. It can be clearly observed that the interleaved RNLM scheme preserves detail much better and restores some fine detail that the ANLM filter cannot, especially some of the interior detail of the bone structures. This fine detail is just not expressed at a strength that is sufficient enough for the (A) NLM filter to restore it from the patches found in the local image, making it necessary to use a clean source for these.

Figure 2b shows the results for the noise case. Here the differences of NLM and ANLM are not as profound as for the streak case. However, similar qualitative differences can be observed for the BLF and TVM, as well as for the two RNLM strategies (see above). Interesting for the latter is the dark feature pointed to by the arrow in the Original image. This feature does not exist in either reference image and is instead restored using local NLM since the reference-image based matching did not return a sufficiently high sum of weights (while similar is also true for the streak case above but there the greater number of iterations also enabled a better OS-SIRT data-driven reconstruction). Nevertheless, this is a clear indicator that the RNLM scheme is very sensitive to the richness of the underlying prior and ongoing research seeks to improve on this.

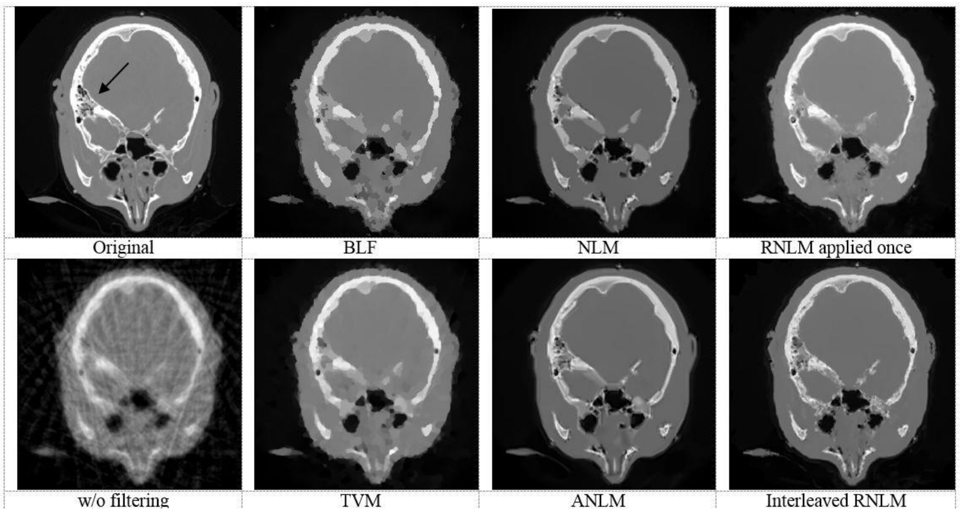
C. *Time performance*

Table 1 lists the run times to filter images of 3 different sizes (256^2, 512^2, and 1024^2) on the GPU. We found that performing filtering in 3D did not yield any improvements so restricting our experiments to 2D is well justified. In the table we list both the timings for the non-optimized (NOPT) and the optimized (OPT) GPU implementation reported in [12]. This optimization achieve a speedup of about 1.2 for the BLF, about 4 for the NLM filter, and about 3.2 for the ANLM filter. Please note that these speedups are in addition to the two orders of magnitude speedup over a corresponding CPU implementation, as reported in [8].

To estimate the TVM performance on the GPU we used TVM GPU implementation of Pock et al. [4] as a reference. They used a NVIDIA 8800 GTX for their experiments and we report their timings in Table 2 as well. In order to make these timings comparable to ours we extrapolated them to the GTX 480 using commonly reported speedup numbers. We may add, however, that once the parameter λ grows larger, which is needed for the rather noisy data we have used here, the computation time tends to increase significantly over those listed here.

Overall there is about an order of magnitude difference in the run times for each of the filters: BLF, NLM, and ANLM, with BLF being the fastest. The TVM requires about the same time as NLM. The timing of the RNLM filter is comparable to that of the NLM filter since the matching process is similar.

(a)



(b)

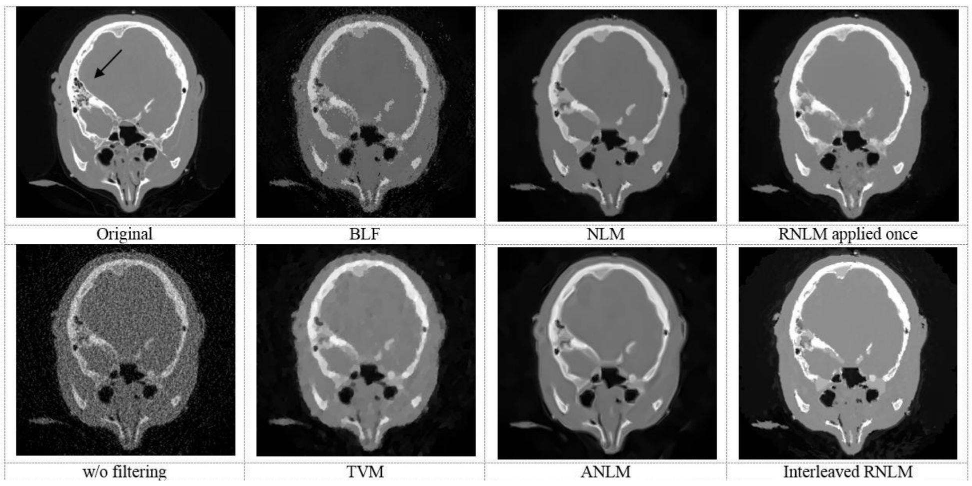


Figure 2. Brain dataset, reconstructed with the interleaved regularization pipeline for the (a) few-view scenario (20 uniformly distributed projections, 200 iterations with OS-SIRT 10). (b) noisy data (SNR 10, 180 equi-angular projections, 10 iterations with OS-SIRT 10).

IV. CONCLUSIONS

We have explored the use of local nonlinear neighborhood filtering as a non-iterative alternative to the popular TVM method for regularized CT reconstruction. Our results indicate that these types of filters can be advantageous to TVM, meeting and exceeding its capabilities.

REFERENCES:

[1] A. Buades, B. Coll and J. Morel, "A Review of Image Denoising Algorithms with A New One," *Multi-scale Modeling Simulation* 4(2) 490-530 (2005).

[2] M. Elad, "On the Bilateral Filter and Ways to Improve It," *IEEE Trans. On Image Processing* 11(10) 1141–1151 (2002).

[3] C. Kervrann and J. Boulanger, "Optimal Spatial Adaptation for Patch-based Image Denoising," *IEEE Trans. on Image Processing* 15(10) 2866-2878 (2006).

[4] T. Pock, M. Unger, D. Cremers and H. Bischof, "Fast and Exact Solution of Total Variation Models on the GPU," *IEEE Computer Vision and Pattern Recognition Workshop*s 1–8 (2008).

[5] E. Sidky, X. Pan, "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization," *Phys. Med. Bio.* 53(17):4777-4807 (2008).

[6] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. Intern. Conf. on Computer Vision (ICCV)* 839-846 (1998).

[7] F. Xu, W. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard and K. Mueller, "On the Efficiency of Iterative Ordered Subset Reconstruction Algorithms for Acceleration on GPUs," *Computer Methods and Programs in Biomedicine* 98(3) 261-270 (2010).

[8] W. Xu and K. Mueller, "A Performance-Driven Study of Regularization Methods for GPU-Accelerated Iterative CT," *HPIR,* 20-23 (2009).

[9] W. Xu, F. Xu, M. Jones, B. Keszthelyi, J. Sedat, D. Agard and K. Mueller, "High-Performance Iterative Electron Tomography Reconstruction with Long-Object Compensation using Graphics Processing Units (GPUs)," *J. Structural Biology*, 171(2) 142-153, (2010).

[10] W. Xu and K. Mueller, "Learning Effective Parameter Settings for Iterative CT Reconstruction Algorithms," *Proc. International Meeting on Fully 3D Image Reconstruction,* 251-254 (2009).

[11] W. Xu, K. Mueller, "Efficient Low-Dose CT Artifact Mitigation Using an Artifact-Matched Prior Scan," *Medical Physics*, 39:4748-4760, 2012.

[12] Z. Zheng, W. Xu, and K. Mueller, "Performance Tuning for CUDA-Accelerated Neighborhood Denoising Filters," *HPIR* (2011).

Test Size	**BLF**			**NLM**			**ANLM**			**TVM (**from [4])	
	NOPT	**OPT**	**Ratio**	**NOPT**	**OPT**	**Ratio**	**NOPT**	**OPT**	**Ratio**	**8800 GTX**	**GTX 480**
256^2	0.65	**0.53**	1.23	51.09	**12.70**	4.02	142.32	**43.57**	3.27	17.50	6.74
512^2	2.15	**1.76**	1.22	182.49	**42.06**	4.34	374.8	**117.24**	3.20	59.60	22.95
1024^2	8.08	**6.54**	1.24	699.23	**161.25**	4.34	2072.67	**597.91**	3.47	504.10	194.15

Table 1. Wall clock time (in *ms*) of the GPU-accelerated BLF, NLM, and ANLM filters both optimized (OPT) and non-optimized (NOPT) for different image sizes. Ratio is the speedup NOPT/OPT. For TVM, the parameter λ grows larger for noisy data which further increases time.