BM3D as a Superiorization Function within SART

for Denoising of Computed Tomography Images

Jon Henshaw
Department of Mathematics
University of Washington, Bothell
Bothell, WA, USA

jhenshaw87@gmail.com

ABSTRACT

In this paper, we discuss a novel combination of techniques for improving the apparent signal-to-noise ratio in low-dose Computed Tomography (CT) image reconstruction. Our experiments revolve around utilizing the Block-Matching 3D (BM3D) algorithm as a secondary optimality function within a modified 'plug-and-play+' superiorization framework for the simultaneous algebraic reconstruction technique (SART). In contrast to the initial design of the superiorization framework, which relies on a gradient descent strategy to achieve the secondary objective, the plug-and-play version allows for non-differentiable procedures of any kind to be utilized to generate perturbations. With the plug-and-play+ specification, we introduce new design features that bolster BM3D's effectiveness in this context, and find that re-experimentation with previously unsatisfactory optimality functions may be warranted under this new specification.

1 BACKGROUND

1.1 Computed Tomography

Computed tomography is a medical imaging technique that allows for the reconstruction of cross-sectional images of the body via a series of x-ray measurements through the patient along a single plane. The use of x-rays means that CT scans can be dangerous, particularly over a long period of time, or if repeated imaging sessions are required. Magnetic resonance imaging (MRI) is an alternative cross-sectional scanning technology that does not carry the same risk. But MRI is a much less rapid technology, is costlier[2], and may be fully contraindicated in a number of scenarios[4]. Thus, it is worthwhile to explore the possibility of improving the quality and clarity of CT scans, with the end goal of being able to maintain image quality while lowering the total x-ray dose a patient is exposed to during the imaging process.

During a CT scan, x-rays are sent directly through the body, and the drop in intensity of the beam after having passed through the patient is measured on the other side. A decrease in x-ray intensity indicates that the beam has passed through some solid material, as denser material will absorb more energy from the beam, meaning a larger drop in intensity indicates that the beam has passed through a higher overall density. In this way, the reduction in intensity of the beam can be seen as a line integral of the density of the patient's body along that beam's path. A collection of these measurements in a single plane from different angles around the body can be compiled into an image called a sinogram, as in (*fig.*1). Each column of the sinogram represents one set of x-ray measurements taken from a single radial position about the body. The emitter/detector pair updates its radial position for each new measurement, so the sinogram displays some sinusoidal characteristics as specific features of the subject move back and forth along the detector's field of vision.

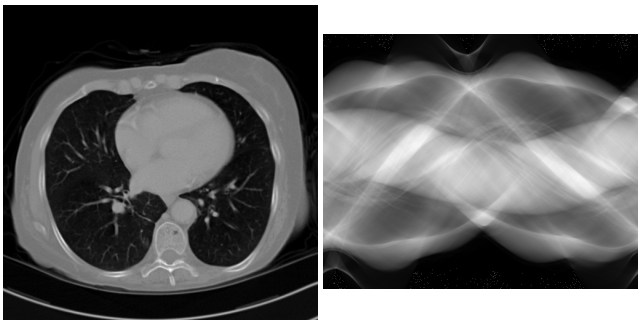


Figure 1: A reconstructed CT image (left), and its sinogram.

X-ray absorption is a somewhat randomized process, and this randomization is largely material-dependent, so it is useful for final image quality to ensure we have a large sample size—both in terms of the number of individual measurements, and the number of x-rays per individual measurement (beam intensity)—in order to maximize the signal-to-noise ratio. However, this is not a trivial matter, as higher x-ray dosing can become dangerous.

1.2 The Simultaneous Algebraic Reconstruction Technique

The measurements which are used to construct the sinogram can also be compiled into a system of linear equations for use in an iterative algorithm called the simultaneous algebraic reconstruction technique (SART)[1]. This technique can be used to reconstruct a full cross-sectional image of the plane of the body through which the measurements are taken. This reconstruction is what is typically thought of as a CT image—again, as shown in (*fig.*1). Since there is necessarily some random error in each measurement during a CT scan, it is highly unlikely there will be an exact solution to our system of linear equations, so iterative techniques like SART are utilized to find a numerical approximation of the true image. This is achieved by minimizing a proximity function $\|A\vec{x} - \vec{b}\|$, where $\vec{x}$ is our current iterative solution, $\vec{b}$ is a log-transformed vector of beam intensity measurements according to Beer's Law, and A is a system of linear equations approximating each pixel of the image's contribution to the drop in beam intensity for any

given measurement, based upon physical system properties and measurement locations.

SART realizes larger features first, and then clarifies finer definition. Each iteration of SART updates the image approximation $\vec{x}_{k\to k+1}$ via a weighted least-squares approximation in the following manner[8]:

$$\vec{x}_{k+1} = \vec{x}_k - \omega_k DA^T M(A\vec{x}_k - \vec{b}) \tag{1}$$

Where the relaxation parameter ω can be any real $\in (0, 2)$, and D, M are defined, for entries a in our $m \times n$ matrix A, as:

$$D = \text{diag}\left\{1 \Big/ \sum_{k=1}^{m} |a_{kj}|,\ j = 1 \dots n\right\} \tag{2}$$

$$M = \text{diag}\left\{1 \Big/ \sum_{k=1}^{m} |a_{ik}|,\ j = 1 \dots m\right\} \tag{3}$$

A distinct feature of SART is the strength of its block-iterative modification. Instead of performing $(eq.1)$ on the entire intermediate image $\vec{x}$, we can break $\vec{x}$ up into any number of interleaved subsets n, each taking a fraction of the time to process individually. While the block iteration does need to be performed n times, leading to a similar overall computation time, it nevertheless leads to a much faster convergence rate. Note that $(eq.1)$ can be rewritten into the form $R(\vec{x}) = T\vec{x} + \vec{c}$, with $T = I - \omega_k DA^T MA$, and our $\vec{c} = \omega_k DA^T M\vec{b}$, as shown below:

$$\begin{aligned}
\Rightarrow R(\vec{x}) &= \vec{x} - \omega_k DA^T M(A\vec{x} - \vec{b}) \\
&= I\vec{x} - \omega_k DA^T MA\vec{x} + \omega_k DA^T M\vec{b} \\
&= \left[I - \omega_k DA^T MA\right]\vec{x} + \left[\omega_k DA^T M\vec{b}\right] \\
&= T\vec{x} + \vec{c}
\end{aligned}$$

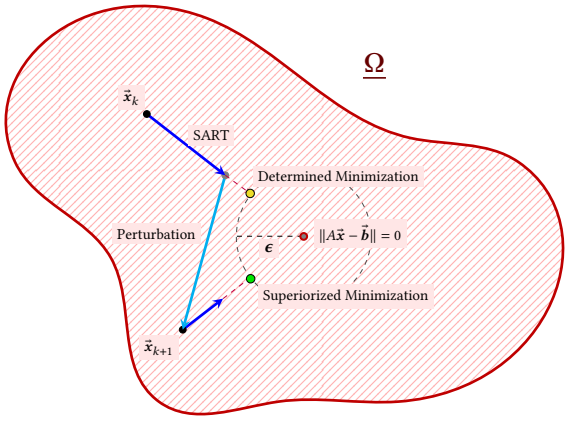


Figure 2: Simplified depiction of the way in which superiorization can guide SART towards a more optimal solution.

Then, as iterative applications of a procedure in this form will effectively apply a power sequence on T, we know that SART will converge if and only if the spectral radius $\rho(T) < 1$[13]. And as SART *is* guaranteed to converge[9], we know, in fact, that $\rho(T) < 1$.

Yet, we might still have a relatively large spectral radius such as $\rho = 0.9998\dots$ If this is the case, SART will require an extremely large number of iterations in order to converge, as an iterative algorithm of this form has a convergence rate inversely proportional to the size of ρ. However, with the block iterative method, $\forall i \in [1, n]$, we know that $\rho(T_i) \leq \rho(T)$, and thus that their combined product will induce a convergence factor on the order of $\rho(T)^n$. This means we can apply multiple factors of the convergence ratio during a single iteration, for the same computational cost as in the original formulation of SART, by instead forming it as[6]:

$$R(\vec{x}) = \left[\prod_{i=1}^{n} T_i\right]\vec{x} + \vec{c} \tag{4}$$

SART is a deterministic function. However, we know there exist infinitely many possible solutions for $\vec{x}$ in our problem space Ω that will minimize the proximity function $\|A\vec{x} - \vec{b}\|$. Knowing where in the problem space the most-ideal reconstruction is located, and how to get there from an initial approximation, is a problem left unsolved by the standard SART algorithm.

1.3 Superiorization

Superiorization[5] is a technique capable of addressing this problem. Every iteration of SART, we can alter, or 'perturb,' our result by utilizing a secondary function ϕ to emphasize any other optimality metric we desire. While SART is deterministic, we can alter its vector of approach so it will converge upon a 'superior' solution according to the quality measured by our secondary function ϕ, as depicted in $(fig.2)$.

Regardless of optimality metric ϕ, a superiorized application of SART is capable of reaching any data fidelity target for $\|A\vec{x} - \vec{b}\|$ attainable by a standard application of SART. In order to guarantee this convergence characteristic, the sequence of perturbations themselves must converge to zero, and be summable. To achieve this, the norm of the initial perturbation is capped at some initial value α, and successive perturbations are reduced by a factor of some $\beta \in (0, 1)$. If we take $\vec{x}_{k+1} = R(\vec{x}_k)$ to be our SART procedure, the superiorized SART algorithm can be described as[8]:

$$\vec{x}_{k+1} = R(\vec{x}_k + \alpha\beta^{k-1}\vec{v}_k) \tag{5}$$

Where $\vec{v}_k = -\dfrac{\nabla\phi(\vec{x}_k)}{\|\nabla\phi(\vec{x}_k)\|}$, the unit vector in the direction of gradient descent for our secondary optimality metric ϕ.

As our aim is to reduce noise, one obvious choice for ϕ would be the total variation (TV) function. The TV function is a measurement of variation between adjacent pixels in an image, and is calculated by

forming the sum of something like the 'euclidean distance' in the saturation of each pixel, with two of its adjacent pixels[7]:

$$TV(\vec{x}) = \sum_{m,n} \sqrt{\left(\vec{x}_{m+1,n} - \vec{x}_{m,n}\right)^2 + \left(\vec{x}_{m,n+1} - \vec{x}_{m,n}\right)^2 + \epsilon^2} \quad (6)$$

This approach reduces noise, but also introduces a blotchy, over-smoothed quality to the image that can obscure small features. As medical imaging is often used to detect small features (fractures, the beginnings of tumors, etc.), this is not optimal for our purposes. As an alternative, we experiment with utilizing the 3D Block-Matching algorithm (BM3D)[3].

1.4 3D Block-Matching

The BM3D algorithm is an image denoising procedure. It works by searching a given image for similar-looking square subsets of the image (blocks), and then utilizing a 3D-transform through a stack of these blocks to further minimize variation across the already similar-looking blocks—assumed, then, to be noise. It can achieve some stunning results in high-definition photo restoration, and through some clever tricks, the complexity of this algorithm has been reduced to $\mathcal{O}(n)$, so it is a strong candidate for use in an iterative application such as superiorization. However, BM3D is not a differentiable function with a gradient—it is a denoising procedure—so a different approach than gradient descent must be used to generate perturbations.

1.5 Plug-and-Play Superiorization

The technique developed as an alternative to the gradient descent strategy is to set the initial perturbation equal to the normalized difference between $\vec{x}_k$ and $\phi(\vec{x}_k)$. We then multiply this by our $\alpha\beta^{k-1}$ as before, and that is the perturbation applied to $\vec{x}_k$ before it is sent back to SART to generate $\vec{x}_{k+1}$. Because a single line of code can be altered in such an algorithm to utilize any optimality algorithm desired (simply alter the function call when calculating $\phi(\vec{x}_k)$), this approach has been coined 'plug-and-play superiorization.' [7]

Our initial tests to get the experiment up-and-running led to some heavy hyperparameter tuning, and a few slight modifications to the plug-and-play approach.

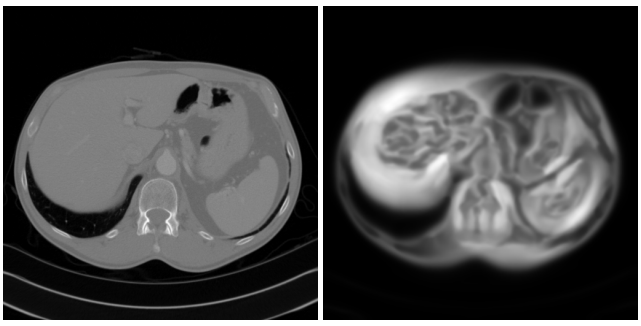


Figure 3: Reference image, and BM3D-Superiorized reconstruction using $\sigma = 10$.

2 MODIFICATIONS

2.1 Initial Hyperparameter Tuning

Our goal was to utilize BM3D as our secondary objective function ϕ. BM3D requires one hyperparameter, σ, an estimation of the standard deviation of the noise in the image. Previous research[11] indicated that $\sigma = 10$ or $\sigma = 20$ may be ideal for medical imaging applications of BM3D, as a higher σ led to smearing of finer definition. However, when we utilized this σ value in our superiorization structure, it led to the construction of some bizarre images that, while striking, are not usable in the context of medical imaging technology. An example of this phenomenon is displayed in (*fig.*3). After experimenting with σ values on the integer range $[1, 20]$, we discovered that even a value of $\sigma = 1$ was too high. Splitting the range $(0, 1)$ by steps of 0.001, we subjectively chose $\sigma = 0.02$ as a good value to proceed with for our general experiments.

At the end of each iteration of the algorithm, our proximity function $\|A\vec{x} - \vec{b}\|$ is checked against some value ϵ. If $\|A\vec{x} - \vec{b}\| \leq \epsilon$, then we consider our method to have converged. As we are comparing this method to the general SART algorithm, we decided to first reconstruct our images using a standard SART application, running it for a fixed number of iterations. Afterwards, we read the true $\|A\vec{x} - \vec{b}\|$ for each of these reconstructions, and set those as individualized ϵ-targets for the given image at the given exposure level. This approach was chosen because the quality of any given reconstruction is highly subject-dependent, so generalized ϵ-targets may not have served us very well globally. Further, as these targets are taken from a successful application of SART, we know that we do not run the risk of setting an unattainable target fidelity for our superiorized algorithms.

2.2 Algorithmic Alterations

An issue we ran into early on was that it seemed BM3D was not doing much of anything. After a period of some discovery, we learned that our initial cap on perturbations α was entirely too small. The norm of our initial perturbations was typically over 60, while we were utilizing an $\alpha = 0.97$. This led to the first algorithmic modification to the plug-and-play structure: Instead of hyperparameterizing α, we set our initial $\alpha = \|\vec{x} - \phi(\vec{x}_1)\|$. This means the very first perturbation per reconstruction is not normalized, as before. However, to maintain summability, α is multiplied in successive iterations by a hyperparameter $\gamma \in (0, 1)$. The behavior of γ and its reaction to fine-tuning is different enough from the original behavior of β to be distinguished with a different symbol.

The next big hurdle we came across was that the algorithm was taking an exceedingly large number of iterations to converge, with no subjective improvement in overall quality by the final iteration. We investigated intermediate iteration renders, and found that the superiorization step was inducing huge perturbations at early iterations, effectively erasing all progress made by SART in the previous step. These perturbations were eventually dampened by γ, so the algorithm still converged. However, by the end, our ϕ was so strongly suppressed that it did nothing to prevent SART from re-introducing fine-grained noise during the final iterations of the algorithm.

As SART realizes broad features first, the fine-grained systemic noise from our measurements doesn't begin to appear in our intermediate reconstructions until after the first handful of iterations, so inducing perturbations early on achieves little more than blurring the image, preventing those broad features from coalescing until ϕ is sufficiently suppressed by γ. And with the full magnitude of initial perturbation being utilized, SART requires a few iterations to pull itself back on track once we've nudged it in the direction we want.

So our solution to the problem was to modify when in the convergence cycle ϕ would be invoked. We landed upon a two-pronged approach. First, wait a certain number of SART iterations before the first superiorization step occurs (set by hyperparameter *k*min). And second, only perform the superiorization step every few iterations (set by hyperparameter *k*step). There is a *k*max hyperparameter as well, but we set this arbitrarily large for our experiments to ensure each algorithm would have the chance to fully express itself. Then, for any given image reconstruction, our plug-and-play+ superiorization algorithm reads as in (*alg*.1).

for $k \in [1, kmax]$ **do**
  if $(k \geq kmin) \wedge ((k - kmin) \bmod (kstep) = 0)$ **then**
    $x_\phi \leftarrow \phi(x)$; // BM3D
    $perturb \leftarrow (x - x_\phi)$;
    $pnorm \leftarrow \|perturb\|_{fro}$;
    if $k = kmin$ **then**
      $\alpha \leftarrow pnorm$;
    else
      $\alpha \leftarrow \alpha \times \gamma$;
    end
    if $pnorm > \alpha$ **then**
      $perturb \leftarrow \frac{\alpha \times perturb}{pnorm}$;
      $x \leftarrow x + p$;
    else
      $x \leftarrow x_\phi$;
    end
  end
  $x \leftarrow R(x)$; // SART
  if $\|Ax - b\|_{fro} \leq \epsilon$ **then**
    export image;
    break;
  end
end

Algorithm 1: Plug-and-Play+ Superiorization Algorithm.

3 EXPERIMENTAL METHODOLOGY

Our experiments began by selecting a set of 20 real-world CT image reconstructions at 512×512 resolution. We then took these images and reverse-transformed them back into sinograms by remeasuring them with fanbeam projections constructed using the Astra toolbox[10] for Python. We created reconstructions for each sinogram at three different exposure levels, introducing Poisson noise proportional to the simulated count rate. The three different exposure levels used were:

- 1×10^4
- 2.5×10^4
- 5×10^4

We then reconstructed approximations of the original reconstructions $\vec{x}_{\text{ref}}$ by passing these noisy sinograms through four different variations on the algorithms described above:

- SART
- BM3D-superiorized plug-and-play+ SART
- TV-superiorized SART
- SART with BM3D post-processing (single application)

Each resultant reconstruction was then compared to its reference image. For each combination of reconstruction technique and exposure level, the mean across the entire image set of four different metrics are taken. The four metrics included in our analysis are:

- PSNR (peak signal-to-noise ratio)
- SSIM[12] (structural similarity index)
- ΔTV $(TV(\vec{x}) - TV(\vec{x}_{\text{ref}}))$
- Time to Converge (iterations to converge, times *n*subsets)

Since the number of subsets utilized is inversely proportional to the number of iterations required to converge, if we wish to effectively measure and compare convergence rates of these algorithms, n needs to be taken into account. We have done so by multiplying n into the number of iterations to converge, including for the fixed-number of iterations used to generate the ϵ-targets as discussed in the previous section. We tuned the rest of our hyperparameters to $\gamma = 0.75$, $\sigma = 0.02$, and the variable hyperparameters displayed in (*fig*.4). The entire suite of command line literals used to generate the data for our experiment is displayed in (*fig*.10).

Figure 4: Variable Hyperparameters for Test Set Generation

Exposure	*k*min	*k*step	Mean ϵ-Target	Conv. Time to Gen. ϵ
1×10^4	5	4	41.869	144
2.5×10^4	10	5	26.040	216
5×10^4	15	5	16.170	324

4 RESULTS

A more in-depth table with thumbnails of each of our 20 reference images, and the individual ϵ-targets set for each of them at each exposure level, can be found in the table in (*fig*.9). Also included is the mean ϵ-target across the three exposure levels for each individual image. As the ϵ-targets were generated from a fixed convergence time, this illustrates the principle that imaging denser material with CT results in less accurate results—as we see that the images of bonier areas have higher, and thus noisier, mean ϵ-targets.

In (*fig*.5), we show the results of our metrics analysis in both table and bar graph form. The first graph shows that the single, post-processing application of BM3D does little to improve the PSNR score over the standard SART algorithm alone. Both of our superiorized algorithms perform somewhat better than the standard SART procedure, and it does seem that the BM3D version pulls

ahead of the TV one at the higher exposure level, but it is unclear from this data whether BM3D outperforms TV overall in terms of PSNR.

The SSIM scores are extremely close, and are even closer than they may at first appear, as the y-axis is on the range $[0.9, 1]$. Perhaps not surprisingly (as BM3D was developed explicitly with SSIM in mind), the single-application of BM3D as a SART post-processing step seems to do the most in improving this score. Again, it is somewhat unclear from this data which of our superiorized algorithms outperforms the other. There is only $\sim 0.8\%$ difference in their performance at the 5×10^4 exposure level, which does not show significance. Yet, it does seem they are each reliably somewhat better than the standard SART algorithm in terms of SSIM score.

While the PSNR and SSIM metrics do not seem to deliver definitive results, there is one category with a clear outlier: TV-superiorization underperforms in terms of convergence time compared to the other algorithms by a factor of over 400% at the 5×10^4 level. It should be noted that we have not taken into account the cost of evaluating BM3D itself versus evaluation of the gradient descent of the TV function in our calculations for convergence time. Nor do we track the difference in number of evaluations of each of these functions due to the introduction of the $k$min and $k$step hyperparameters. If considered, there may be found an even more extreme difference than what is calculated here, as we know BM3D is not performed every iteration, and that it is of order $\mathcal{O}(n)$. However, this data does illuminate a possible cause for some of the aggressive over-smoothing often induced by TV-superiorization, as the perturbations in this method are applied much more numerously than with the plug-and-play+ approach.

We also see that TV-superiorization may not even outperform the others in terms of ΔTV. Notice that this graph has both positive and negative values, with negative values representing a reduction in TV compared to the original image. It may be more useful to look at magnitudes for this metric, as we *should* want to roughly *match* the TV of the true image. But in terms of TV *reduction*, we see the BM3D post-processing procedure goes far beyond what the other algorithms are doing, especially at the 5×10^4 exposure level. This does seem to correspond to the way BM3D post-processed images look blurrier (albeit less noisy) than the others. Surprisingly, the standard SART algorithm seems to most closely match the TV of the reference image at the two lowest exposure levels, which may indicate that TV-superiorization is doing too much in early iterations—further suggesting that its aggressive oversmoothing might be tamed by utilizing tactics similar to that of the $k$min and $k$step hyperparameters introduced with the plug-and-play+ specification. The BM3D-superiorized algorithm also seems to perform better than the TV-superiorized algorithm during the first two exposure levels, but after flipping sign compared to the reference image, TV-superiorization is shown capable of reducing TV more handily than either of the others.

Finally, in (*fig.*6), (*fig.*7), and (*fig.*8), we have chosen three individual images to juxtapose for subjective comparison: cross-sections of a pelvic region, abdomen/liver, and chest/lungs, respectively. Each row of these figures contains the reference image—and then the SART, BM3D-superiorized, TV-superiorized, and BM3D post-processed reconstructions at a single exposure level. Immediately below the full images are subsections of each image that have been zoomed-in on, where some interesting detail can be observed for finer comparison. The first pair of rows of each figure are the 1×10^4 reconstructions, with the second pair being the 2.5×10^4 reconstructions. The 5×10^4 reconstructions are along the bottom.

Subjectively, the BM3D-superiorized pelvic reconstructions look the best to us, a fact that is especially clear at the lower exposure levels. As this area is extremely dense with bone, and thus noisier with inverse proportionality to the exposure level, the fact that it does even better at lower-exposure levels is impressive. Features look as they should, and while there is some contrast loss, much of the detail remains. Investigating the 1×10^4 row of this figure, we can also see the clearest demonstration of the qualitative differences between each algorithm: SART looks noisier, TV-superiorized looks blotchier, and the BM3D post-processed looks blurrier. The abdominal reconstructions also display these qualities, but to a lesser degree. There are many organs in this area, and the liver in particular is a fairly dense organ—so while there isn't as much bony material in these scans, this area of the body is still fairly dense.

With the sparse density of the chest cavity, and the finer detail of the lung, it is not as clear to us, subjectively, which is the best reconstruction method. We can definitively say that the BM3D post-processed reconstructions do not seem particularly usable in a medical context, but the other three algorithms produce images that are fairly comparable.

It is worth noticing, from the zoomed-in portions of any of these images at the 5×10^4 exposure level, that the cartoonish oversmoothing typically displayed by the TV-superiorized reconstruction technique is not prevalent. In tandem with TV's convergence time being so high, this may be even further suggestion that TV-superiorization might benefit from introduction of hyperparameters similar to $k$min and $k$step, as this behavior seems to align with that which we observed when first experimenting with the use of BM3D as our secondary function, before introducing those hyperparameters to the plug-and-play approach.

5 FURTHER RESEARCH

Further research might find it worthwhile to experiment more heavily with the effects of fine-tuning hyperparameters $k$min, $k$step, and γ, particularly in relation to expected density of material. Since blurriness continues being an issue, experimenting further with σ is not out of the question either. It may also be possible that these hyperparameters will react differently to tuning for different superiorization functions ϕ, so previously dissatisfactory optimality metrics may be worth another look under the plug-and-play+ approach. As mentioned numerous times already, our results strongly suggest that introduction of hyperparameters similar to $k$min and $k$step to the superiorization procedure for differentiable optimality functions (such as TV) may improve the performance of these algorithms, and thus warrants further research. We also notice that each algorithm seems to have its own characteristic flaws, so it may be worth experimenting with the utilization of multiple distinct optimality metrics ϕ within a single superiorization framework.

There is also concern about how to choose initial ϵ-targets when the true image is unknown, as is the case in real-world applications. Too high a target, and our image won't be as clear as a standard SART application, but too low, and it will refuse to converge until it becomes as noisy as a standard SART application. Our own choices for convergence time to generate the target ϵ for these experiments was somewhat arbitrary, and influenced by subjective measures. It may be possible to calculate an estimate of the expected tolerances for $\|A\vec{x} - \vec{b}\|$ based upon expected overall density of the area to be scanned, and the properties of the scanner itself. Another idea we had was to perform an initial full iteration of SART up to a conservatively-large fixed number, and have the target ϵ read from an image in this set that has been manually-selected by a technician or similar as "clear but not noisy." An extension of this idea might have a neural network trained to achieve the same goal, thus maintaining the automated nature of the system.

ACKNOWLEDGMENTS

Special thanks to Dr. Thomas Humphries for his guidance, encouragement, and understanding over the course of this project.

REFERENCES

[1] Anders H Andersen and Avinash C Kak. 1984. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrasonic imaging* 6, 1 (1984), 81–94.

[2] John P. Cunha Charles Patrick Davis. 2022. CT Scan vs. MRI: Differences between Machines, Costs, Uses. *MedicineNet [Internet]* (2022). https://www.medicinenet.com/ct_scan_vs_mri/article.htm

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16, 8 (2007), 2080–2095.

[4] Maryam Ghadimi and Amit Sapra. 2021. Magnetic resonance imaging contraindications. *StatPearls [Internet]* (2021). https://www.ncbi.nlm.nih.gov/books/NBK551669/

[5] Gabor T Herman, Edgar Garduño, Ran Davidi, and Yair Censor. 2012. Superiorization: An optimization heuristic for medical physics. *Medical physics* 39, 9 (2012), 5532–5546.

[6] H Malcolm Hudson and Richard S Larkin. 1994. Accelerated image reconstruction using ordered subsets of projection data. *IEEE transactions on medical imaging* 13, 4 (1994), 601–609.

[7] Thomas Humphries. 2021. Plug-and-Play Superiorization for CT Image Reconstruction. In *SIAM Meeting on Optimization*.

[8] T Humphries, M Loreto, B Halter, W O'Keeffe, and L Ramirez. 2020. Comparison of regularized and superiorized methods for tomographic image reconstruction. *J. Appl. Numer. Optim* 2, 1 (2020), 77–99.

[9] Ming Jiang and Ge Wang. 2003. Convergence of the simultaneous algebraic reconstruction technique (SART). *IEEE Transactions on image processing* 12, 8 (2003), 957–961.

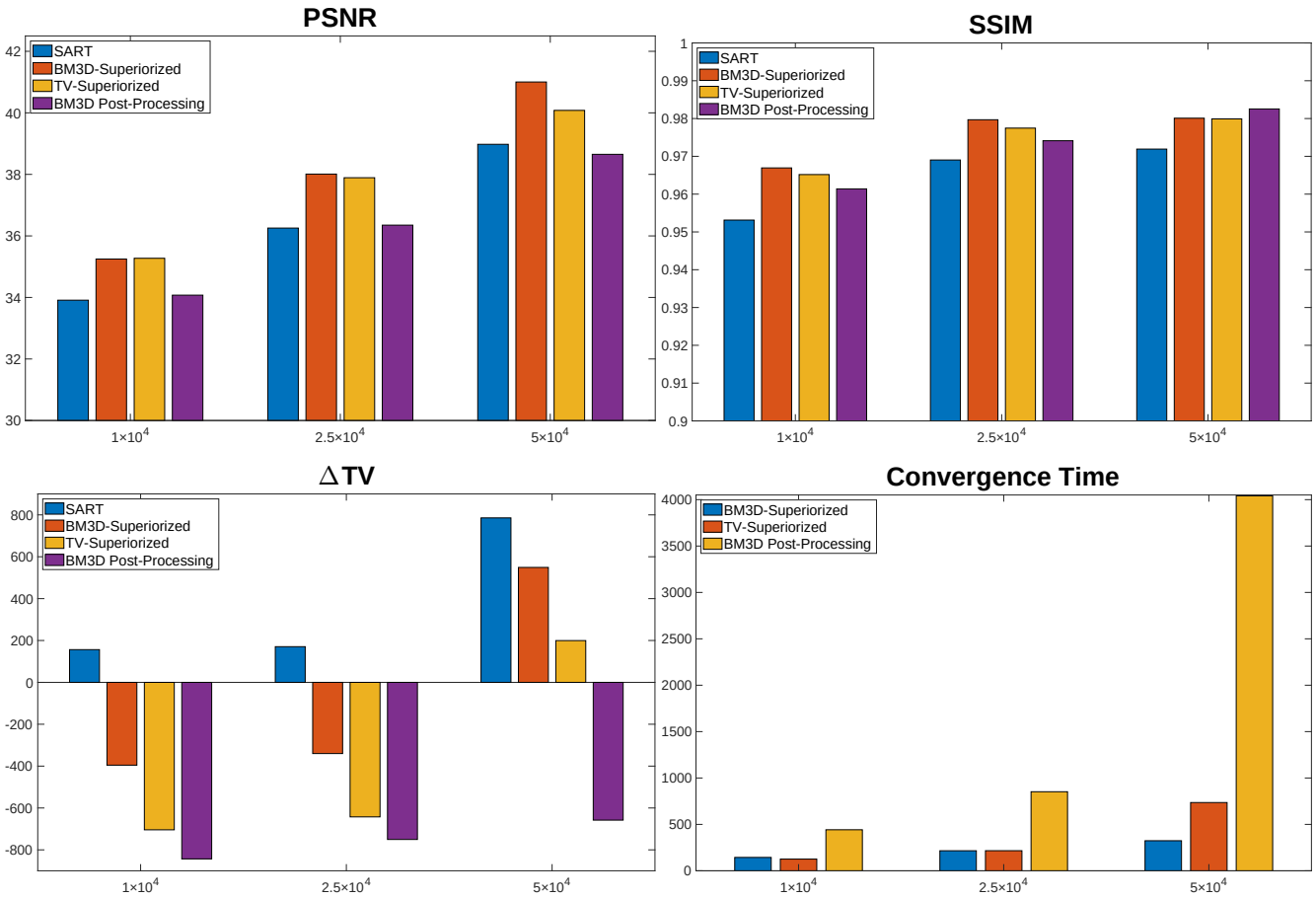
[10] Willem Jan Palenstijn, K Joost Batenburg, and Jan Sijbers. 2011. Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs). *Journal of structural biology* 176, 2 (2011), 250–253.

[11] Ke Sheng, Shuiping Gou, Jiaolong Wu, and Sharon X Qi. 2014. Denoised and texture enhanced MVCT to improve soft tissue conspicuity. *Medical physics* 41, 10 (2014), 101916.

[12] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[13] NJ Young. 1981. The rate of convergence of a matrix power series. *Linear Algebra Appl.* 35 (1981), 261–278.

Figure 5: Graphical and Numerical Representation of Result Metrics.
Please note: Scales of y-axes are dissimilar.



Comparison of Mean Similarity Metrics Across Entire Set for Each Modified SART Algorithm					
Metric	Exposure	Standard SART	BM3D-Superiorized	TV-Superiorized	BM3D Post-Processing
PSNR	1×10^4	33.9114	35.2484	35.2730	34.0748
	2.5×10^4	36.2556	38.0098	37.8916	36.3510
	5×10^4	38.9779	41.0029	40.0803	38.6524
SSIM	1×10^4	0.95314	0.96694	0.96518	0.96139
	2.5×10^4	0.96904	0.97968	0.97747	0.97413
	5×10^4	0.97192	0.98009	0.97991	0.98254
ΔTV	1×10^4	+156.403	-395.548	-704.306	-843.237
	2.5×10^4	+170.532	-340.048	-642.536	-750.439
	5×10^4	+785.614	+549.064	+199.582	-658.002
Time to Converge	1×10^4	144	126.0	442.2	—
	2.5×10^4	216	216.6	852.0	—
	5×10^4	324	735.6	4041.0	—

Figure 6: Comparison of image 3817 (pelvis) with each resultant reconstruction. Lowest exposure level first.

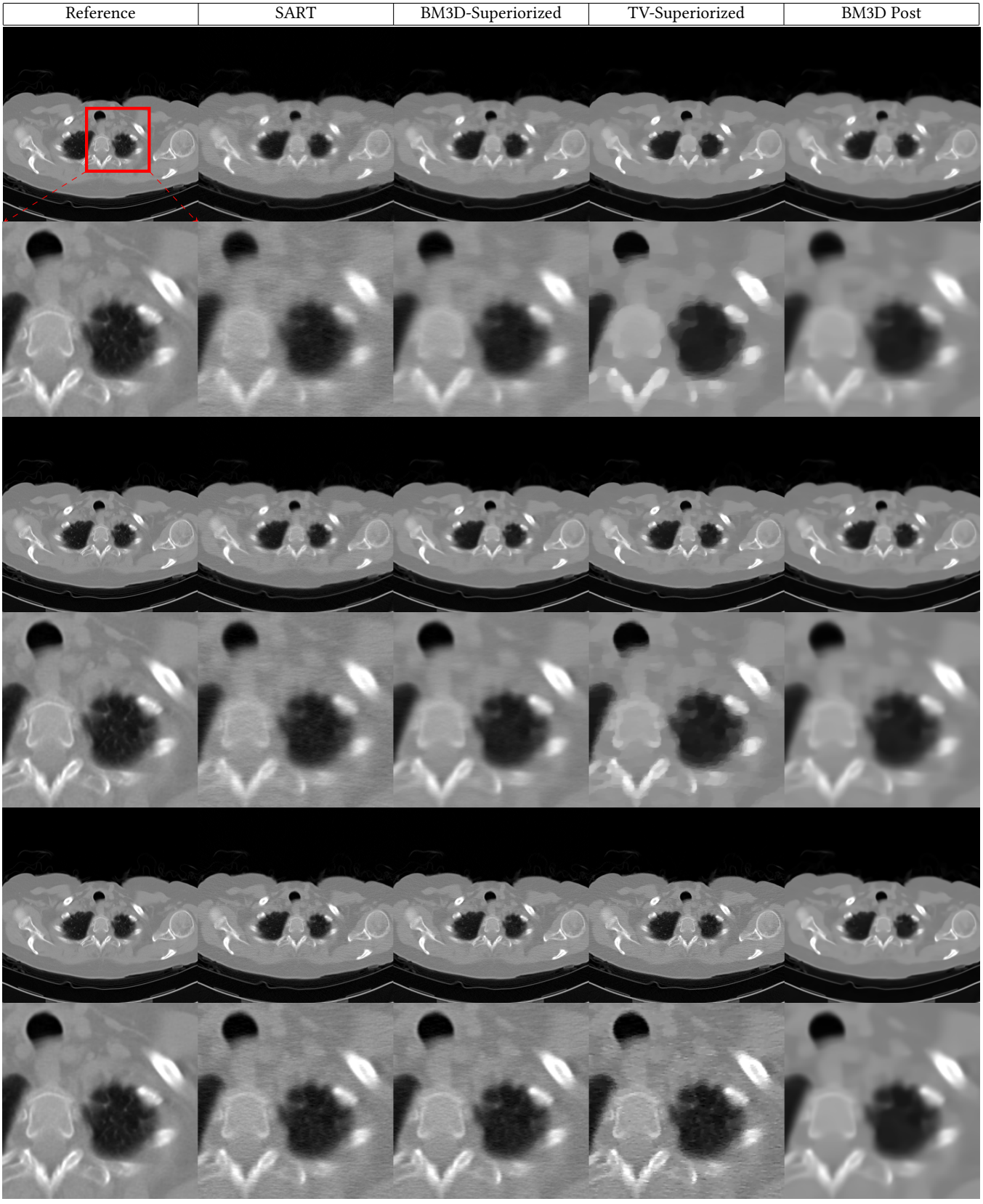


Figure 7: Comparison of image 3883 (liver) with each resultant reconstruction. Lowest exposure level first.

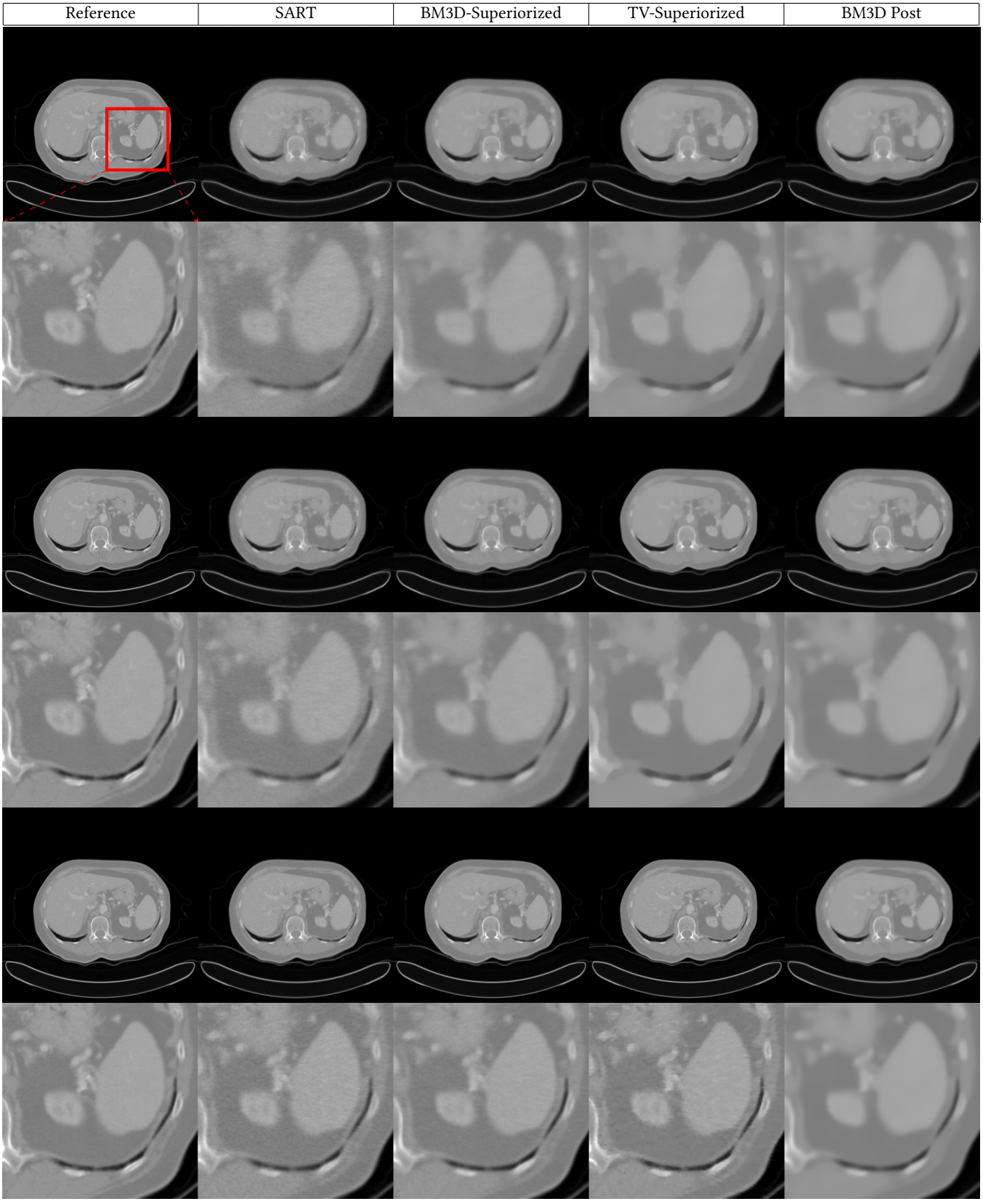


Figure 8: Comparison of image 3803 (lung) with each resultant reconstruction. Lowest exposure level first.

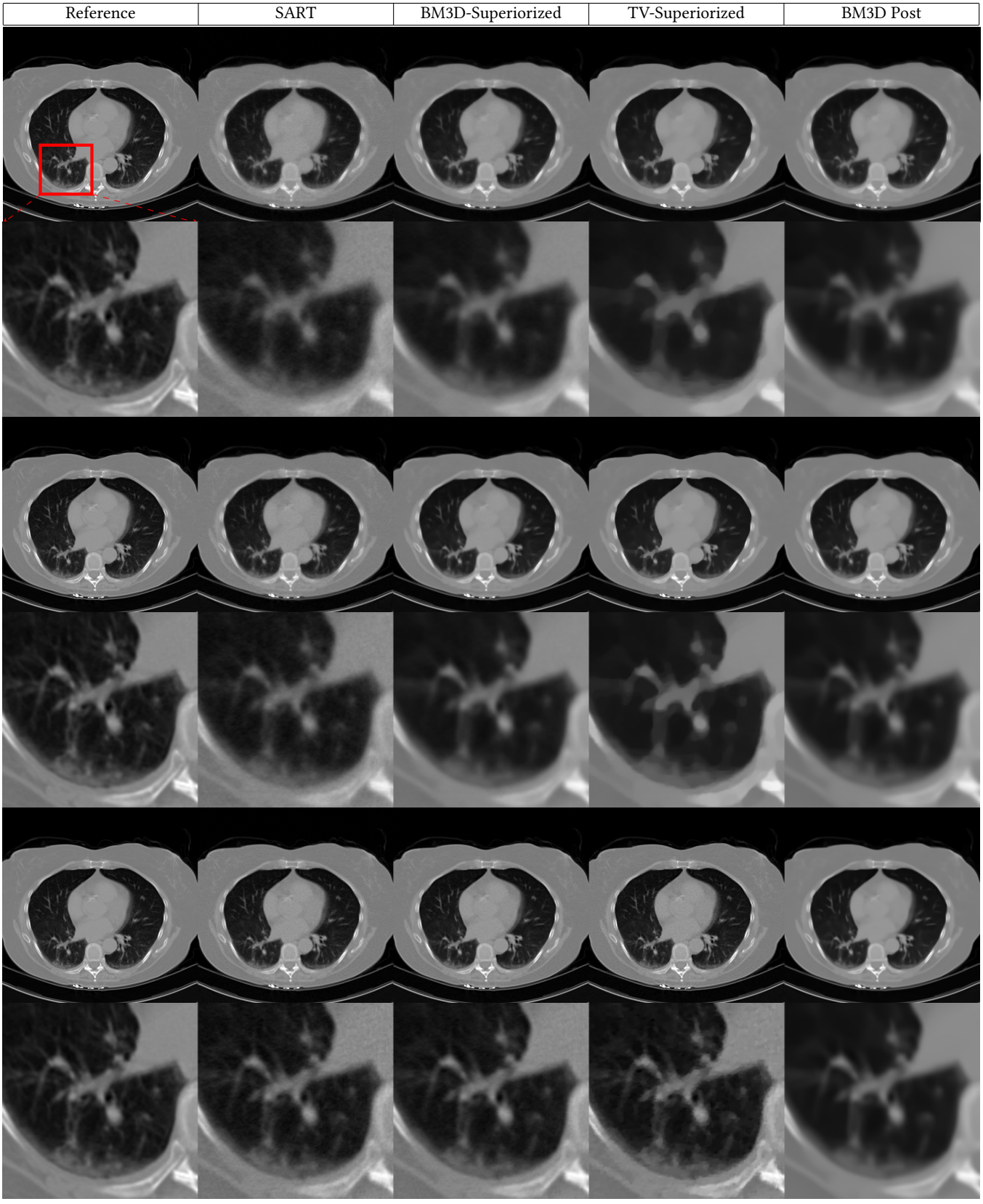


Figure 9: ϵ-Targets for individual images at each exposure level.

Image	1×10^4	2.5×10^4	5×10^4	Mean (img)	Image	1×10^4	2.5×10^4	5×10^4	Mean (img)
3629 	33.798305	21.034120	12.507258	22.446561	3792 	49.304163	30.389597	19.694704	33.129488
3645 	37.816196	23.376304	14.643529	25.278676	3803 	40.873823	25.612410	15.892331	27.459521
3665 	36.010247	22.460129	13.554755	24.008377	3816 	36.329424	22.630560	13.878041	24.279341
3689 	55.240813	34.216626	21.513963	36.990467	3817 	47.173994	29.035184	18.776478	31.661885
3696 	40.633332	25.548050	15.330123	27.170501	3845 	32.889000	20.913214	12.570747	22.124320
3705 	40.798881	25.625331	15.199934	27.208048	3883 	44.649725	27.431934	17.188574	29.756744
3716 	40.734763	25.687267	15.374747	27.265592	3895 	55.245939	34.078170	22.485325	37.269811
3743 	40.442785	25.538445	14.986390	26.989206	3924 	30.201796	18.853552	11.022670	20.026006
3782 	37.717053	23.428681	14.476626	25.207453	3932 	51.042544	31.393727	20.456598	34.297623
3784 	40.918422	25.368276	15.900568	27.395755	3952 	45.559636	28.183549	17.954075	30.565753
					Mean (set)	41.869	26.040	16.170	28.026333

Figure 10: Command line arguments used for batch generation of files.

SINOGRAMS

```
python make_sino.py --in /data/CT_data/bm3d_test_set/ --out sinos/1e4 --psize 0.0568 --range 0 360
                    --counts 1e4

python make_sino.py --in /data/CT_data/bm3d_test_set/ --out sinos/2.5e4 --psize 0.0568 --range 0 360
                    --counts 2.5e4

python make_sino.py --in /data/CT_data/bm3d_test_set/ --out sinos/5e4 --psize 0.0568 --range 0 360
                    --counts 5e4
```

SART & ϵ-TARGETS

```
python SART_pnp_BM3D_new.py --sino sinos/1e4 --out recons/1e4 --psize 0.0568 --range 0 360 --nsubs 18
                            --numits 8 --make_png True

python SART_pnp_BM3D_new.py --sino sinos/2.5e4 --out recons/2.5e4 --psize 0.0568 --range 0 360 --nsubs 18
                            --numits 12 --make_png True

python SART_pnp_BM3D_new.py --sino sinos/5e4 --out  recons/5e4 --psize 0.0568 --range 0 360 --nsubs 18
                            --numits 18 --make_png True
```

BM3D SUPERIORIZED

```
python SART_pnp_BM3D_new.py --sino sinos/1e4 --out recons/1e4 --psize 0.0568 --range 0 360 --nsubs 12
                            --numits 10000000000 --make_png True --sup_params 5 4 0.75 0.02
                            --epsilon_target recons/1e4/t_residuals.txt

python SART_pnp_BM3D_new.py --sino sinos/2.5e4 --out recons/2.5e4 --psize 0.0568 --range 0 360 --nsubs 12
                            --numits 10000000000 --make_png True --sup_params 10 5 0.75 0.02
                            --epsilon_target recons/2.5e4/t_residuals.txt

python SART_pnp_BM3D_new.py --sino sinos/5e4 --out recons/5e4 --psize 0.0568 --range 0 360 --nsubs 12
                            --numits 10000000000 --make_png True --sup_params 15 5 0.75 0.02
                            --epsilon_target recons/5e4/t_residuals.txt
```

TV SUPERIORIZED

```
python SART.py --sino recons/1e4/2_SINO --out recons/1e4/6_TVsup --psize 0.0568 --range 0 360 --nsubs 12
               --numits 10000000000 --sup_params 0.9995 20 0.5 --epsilon_target recons/1e4/t_residuals.txt

python SART.py --sino recons/2.5e4/2_SINO --out recons/2.5e4/6_TVsup --psize 0.0568 --range 0 360 --nsubs 12
               --numits 10000000000 --sup_params 0.9995 20 0.5 --epsilon_target recons/2.5e4/t_residuals.txt

python SART.py --sino recons/5e4/2_SINO --out recons/5e4/6_TVsup --psize 0.0568 --range 0 360 --nsubs 12
               --numits 10000000000 --sup_params 0.9995 20 0.5 --epsilon_target recons/5e4/t_residuals.txt
```

BM3D POST-PROCESSED

```
python SART_pnp_BM3D_new.py --sino sinos/1e4 --out recons --psize 0.0568 --range 0 360 --nsubs 12 --numits 9
                            --make_png True --make_intermediate True --sup_params 9 5 0.75 0.02

python SART_pnp_BM3D_new.py --sino sinos/2.5e4 --out recons --psize 0.0568 --range 0 360 --nsubs 12 --numits 13
                            --make_png True --make_intermediate True --sup_params 13 5 0.75 0.02

python SART_pnp_BM3D_new.py --sino sinos/5e4 --out recons --psize 0.0568 --range 0 360 --nsubs 12 --numits 16
                            --make_png True --make_intermediate True --sup_params 19 5 0.75 0.02
```