# Super-Resolution and Sparse View CT Reconstruction (Supplemental Material)

Guangming Zang, Mohamed Aly*, Ramzi Idoughi,
Peter Wonka, and Wolfgang Heidrich

King Abdullah University of Science and Technology, Saudi Arabia

## 1 Additional experiments

### 1.1 Sparse-View Results

Additional comparison between PCG and PSART results in sparse view tomographic reconstruction scenario is shown in Figure 1. Interestingly, we found that the proximal operator for the data term leads to a strongly convex optimization problem and therefore should have a unique optimal solution. However, CG in practice struggles to find it. This is quite well known in the tomography community (which is why SART and other methods remain popular for this application). For example the figure 1 shows results for comparing different solvers for the proximal operator. We can see that in all settings all methods seem to converge against a similar result (albeit at different speeds), except for CG (grey), which stalls at a much lower value. Note that this behavior is observed across different implementations of CG and CGLS. For example in our submission we use the CG implementation from RTK [1] (as well as the projection/backprojection implementation from the same source), in the figure we show results from the ASTRA toolkit[2], and we also have results from our own CG implementation. So the issue for CG is quite reasonable. The detailed explanation is presented in the main text.

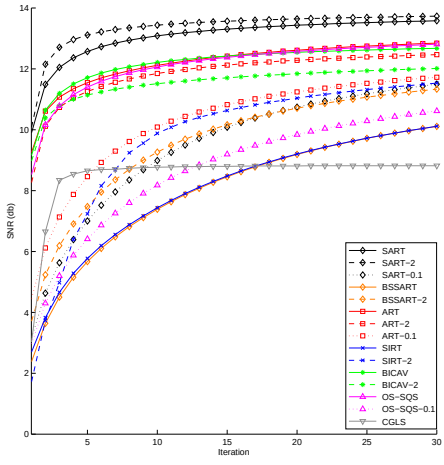
### 1.2 Super Resolution results

We compared PSART-TV with PCG-TV on three different datasets and verified that SART outperforms CG to solve the data term in the scenario of super resolution. The parameters for the experiments are shown in Table 3 and Table 4. As shown in Figure 2, PSART-TV achieves visually better results on an artificial rose, a zone plate pattern, and a real rose.

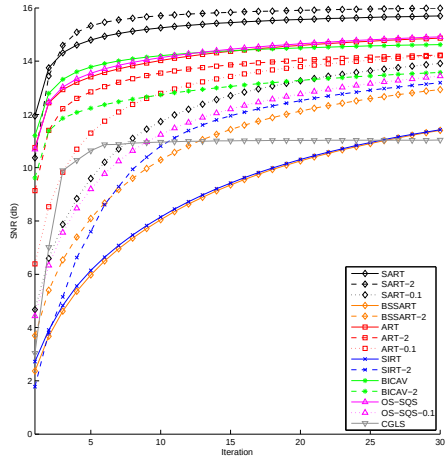
Figure 3 shows the edge detection results from applying Sobel filter in the sagittal plane for **artificial rose**. The size of the volume is $415 \times 314 \times 393$. 120 original-size projection images are used as input for PSART-STP and the best reconstructed result is used as the reference volume for the comparison.

In Table 1, we show the detailed metrics measured on the reconstructed results using different methods, namely PCG-TV, PSART-TV, PSART-SAD,

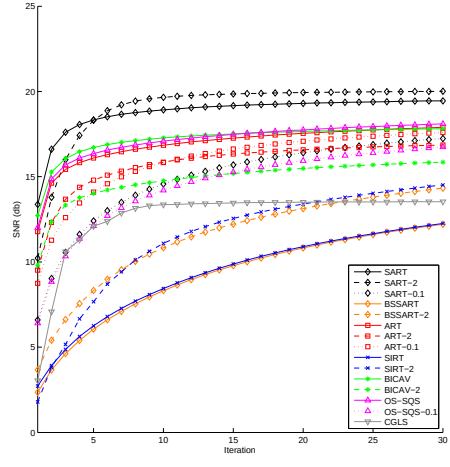
* now at iRobot Corp



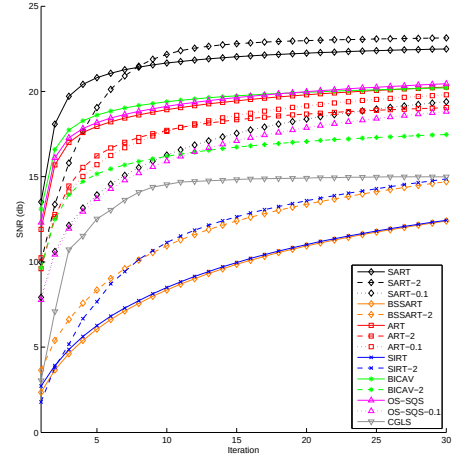

(a) 15 Projections




(b) 30 Projections




(c) 60 Projections




(d) 90 Projections

**Fig. 1.** Comparison of iterative solvers. Plots show SNR per iteration. Relaxation parameter $\alpha = 1$, 2.0, and 0.1 are represented with solid, dashed, and dotted lines.

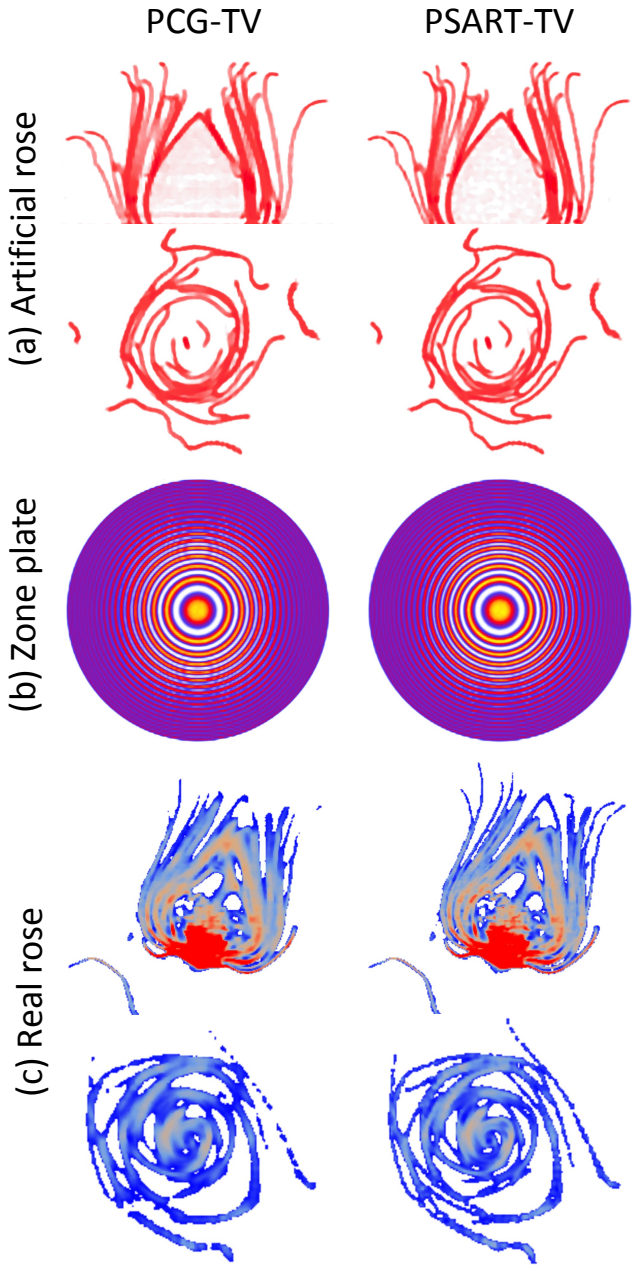



**Fig. 2.** Reconstructed results comparison between PSART-TV and PCG-TV.

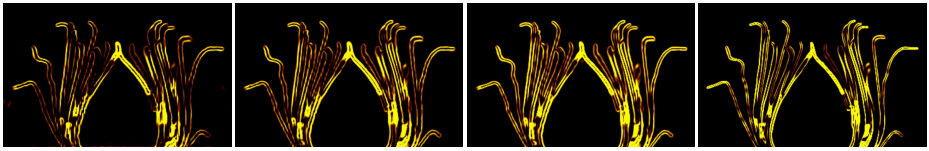



**Fig. 3.** a-d: Representative slice visualization in the sagittal plane for edge detection on a volume reconstructed by PCG-TV, PSART-SAD, PSART-STP, and the reference volume, respectively.

and PSART-STP. The visualization of the reconstructed results are included in our paper. Specifically, in our measurements, the volumes reconstructed from high resolution projections and a full number of projection images for each data set are treated as ground truth. The metrics PSNR and SSIM are calculated as quantitative performance measure. We can observe that our proposed PSART-STP achieves the best performance compared to the other three methods.

**Table 1.** PSNR and SSIM results from different reconstruction methods.

| Method | Metric | Zone Plate | Artificial rose | Plumeria | Toothbrush |
|---|---|---|---|---|---|
| PCG-TV | PSNR | 22.07 | 26.83 | 28.73 | 27.04 |
| | SSIM | 0.976 | 0.958 | 0.963 | 0.934 |
| PSART-TV | PSNR | 22.18 | 27.00 | 28.87 | 27.09 |
| | SSIM | 0.976 | 0.958 | 0.964 | 0.934 |
| PSART-SAD | PSNR | 22.63 | 27.53 | 28.98 | 27.62 |
| | SSIM | 0.978 | 0.961 | 0.969 | 0.938 |
| PSART-STP | PSNR | 24.83 | 29.93 | 30.54 | 31.30 |
| | SSIM | 0.986 | 0.973 | 0.983 | 0.973 |

### 1.3 Segmentations Results

**Table 2.** Segmentations Results, the threshold parameter is adjusted so that the results from PCG-TV and PSART-SAD are close to the Nyquist limit.

| Threshold | Nyquist Limit | PCG -TV | PSART -SAD | PSART -STP | Ground Truth |
|---|---|---|---|---|---|
| 0.55 | 6 | 7 | 7 | 11 | 17 |
| 0.6 | 6 | 5 | 5 | 10 | 17 |
| 0.65 | 6 | 4 | 4 | 8 | 17 |

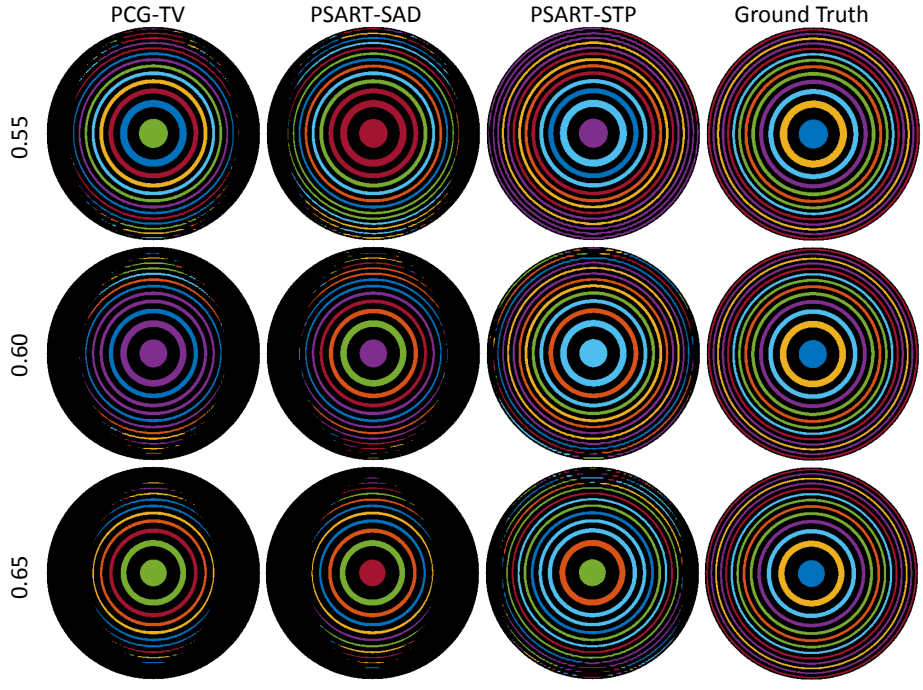



**Fig. 4.** Segmentation results from different methods. From top to bottom: the threshold parameters are 0.55, 0.60, and 0.65, respectively, making the reconstructed results of PCG-TV and PSART-SAD closed to the Nyquist limit. From left to right: different reconstructed methods: PCG-TV, PSART-SAD, and PSART-STP, respectively.

In this part, we show an additional quantitative evaluation on the zone plate data set. We compare the proposed PSART-STP with PSART-SAD and PCG-TV. We use standard image processing algorithms to segment the data. We simply threshold the result and run a connected component algorithm. We check how many rings are correctly reconstructed. A ring is correctly reconstructed if it has no gaps, is not broken into multiple pieces, and if it does not merge with adjacent rings. We report the number of correctly extracted rings in Table 2 and visualize the results in Fig. 4 (The five outmost rings are cut away for better visualization). We can make the following observations:

1. Our method is the best for all threshold values.
2. For higher threshold values, the performance of all methods degrades.
3. We can extract rings over the Nyquist limit.

## 2 Parameters

Table 1 and Table 2 show the parameters used in the experiments of the paper.

**Table 3.** CT Parameters for the datasets used in the paper.

| | zone plate | Artificial rose | Plumeria | Toothbrush |
|---|---|---|---|---|
| SID ($mm$) | 1800 | 536.9627 | 697.0378 | 243.1662 |
| SDD ($mm$) | 5000 | 983 | 983 | 983 |
| Detector pixel | 1024×1024 | 1916×1536 | 1916×1536 | 1916×1536 |
| Detector pixel size ($mm$) | 1 | 0.127 | 0.127 | 0.127 |
| Input pixel | 160 | 168×135 | 120×96 | 1916×1536 |
| Input pixel size ($mm$) | 6.4 | 1.4484 | 2.027 | 0.127 |
| Image downsampled factor | 6.4× | 11.4× | 16× | 1× |
| X-ray penetration ($kV$) | NA | 31 | 25 | 32 |
| X-ray intensity ($\mu$\$A\$) | NA | 725 | 860 | 421 |
| Voxel size ($mm$) | 0.5 | 0.2775 | 0.3605 | 0.0314 |
| Number of projections | 180 | 120 | 180 | 180 |

## 3 Regularization Terms

Different regularization terms have been used for solving the tomography reconstruction problem. Below is a brief overview of the ones considered in this work:

– Anisotropic TV (ATV): This prior is defined as

$$h(x) = \sum_{ijk} |x_{i+1,j,k} - x_{i,j,k}| + |x_{i,j+1,k} - x_{i,j,k}| + |x_{i,j,k+1} - x_{i,j,k}|, \tag{1}$$

**Table 4.** Parameters used with each method, Time elapsed with each method denotes the computing time for each main loop iteration.

| | | Zone plate | Artificial rose | Plumeria | Toothbrush |
|---|---|---|---|---|---|
| FDK | Time elapsed (s) | 22.2 | 8.4 | - | - |
| SART | Number of iterations | 15 | 20 | - | - |
| | Relaxation parameter $\alpha$ | 0.3 | 0.1 | - | - |
| | Time elapsed (s) | 150 | 44 | - | - |
| PCG-TV | Main loop iterations | 40 | 30 | 25 | 20 |
| | Nested CG iterations | 6 | 4 | 4 | 6 |
| | Nested TV iterations | 1 | 1 | 1 | 1 |
| | Prior parameters $\alpha$ | 0.8 | 55 | 35 | 0.1 |
| | ADMM parameters $\beta$ | 500 | 8 | 8 | 1 |
| | Time elapsed (s) | 242 | 36 | 33 | 1158 |
| PSART-TV | Main loop iterations | 20 | 25 | 15 | 15 |
| | Nested SART iterations | 1 | 1 | 1 | 1 |
| | Nested TV iterations | 1 | 1 | 1 | 1 |
| | Prior parameters $\lambda$ | 0.003 | 0.08 | 0.03 | 0.1 |
| | CP parameters $\mu$, ($\mu$= $\tau$) | 0.15 | 0.1 | 0.1 | 0.1 |
| | Time elapsed (s) | 98 | 35 | 33 | 1182 |
| PSART-SAD | Main loop iterations | 20 | 25 | 15 | 15 |
| | Nested SART iterations | 1 | 1 | 1 | 1 |
| | Nested SAD iterations | 1 | 1 | 1 | 1 |
| | Prior parameters $\lambda$ | 0.003 | 0.08 | 0.03 | 0.1 |
| | CP parameters $\mu$, ($\mu$= $\tau$) | 0.15 | 0.1 | 0.1 | 0.1 |
| | Time elapsed (s) | 114 | 42 | 38 | 1228 |
| PSART-STP | Main loop iterations | 25 | 16 | 28 | 25 |
| | Nested SART iterations | 1 | 1 | 1 | 1 |
| | Nested STP iterations | 1 | 1 | 1 | 1 |
| | Prior parameters $\lambda$ | 0.03 | 0.5 | 0.03 | 5.5 |
| | CP parameters $\mu$ ($\mu$= $\tau$) | 0.3 | 0.3 | 0.3 | 0.3 |
| | Time elapsed (s) | 326 | 142 | 95 | 2347 |

where $x_{i,j,k}$ is the voxel value at position $(i, j, k)$. This can be represented as $\tilde{g}(Mx)$, where $\tilde{g}(\cdot) = \|\cdot\|_1$ is the $\ell_1$ norm and $M = D \in \mathbb{R}^{3n\times n}$ is the forward difference matrix. The proximal operator of $\tilde{g}^*(\cdot)$ can be shown to be [3]

$$\mathbf{prox}_{\mu\tilde{g}^*}(u) = \Pi_{B_\infty}(u) = \begin{cases} 1 & u > 1 \\ u & |u| \leq 1 \\ -1 & u < -1 \end{cases}, \tag{2}$$

where the operations are component-wise, and is equivalent to the projection on the unit ball $B_\infty$ of the $\ell_\infty$ norm. Note that we do not need to store the matrix $D$, and multiplication by $D$ (computing the gradient) or by $D^T$ (computing the divergence) can be efficiently computed on-the-fly [4].

– Isotropic TV (ITV): This prior is defined as

$$h(x) = \sum_{ijk} \sqrt{\left|x_{i+1,j,k} - x_{i,j,k}\right|^2 + \left|x_{i,j+1,k} - x_{i,j,k}\right|^2 + \left|x_{i,j,k+1} - x_{i,j,k}\right|^2},$$

where it sums the magnitude of the gradient at each voxel. Using the forward matrix $D$ above and defining a new matrix $E \in \mathbb{R}^{3n \times n}$ that denotes the positions of the forward differences [4], we can define the function $h(x)$ as a norm $\|u\|_E$ for $u = Dx \in \mathbb{R}^{3n}$ defined as

$$\|w\|_E = \|\sqrt{E^T w^2}\|_1 = \sum_v \|w^v\|_2,$$

where the square root and square functions are component-wise, and $w^v$ is the gradient at voxel $v = (i, j, k)$. Now we can express the ITV prior $h(x)$ in terms of the $\|u\|_E$ norm as

$$h(x) = \|Dx\|_E = \tilde{g}(Dx) \text{ where } \tilde{g}(u) = \|u\|_E.$$

The proximal operator for $\tilde{g}^*(\cdot)$ can be shown to be [4]

$$\mathbf{prox}_{\mu\tilde{g}^*}(u) = \Pi_{B^*}(u) = \frac{u}{E \max\left(\sqrt{E^T u^2}, 1\right)} \tag{3}$$

which is the projection on the unit ball $B^*$ of the dual norm $\|u\|_{E^*}$, and where the division and max operations are performed component-wise.

– Sum of Absolute Differences (SAD): This prior is defined as

$$h(x) = \sum_{ijk} \sum_{x_n \in N(x_{i,j,k})} |x_n - x_{i,j,k}|, \tag{4}$$

where $N(x_{i,j,k})$ is the $3 \times 3$ neighborhood around voxel $x_{i,j,k}$ (excluding voxel $x_{i,j,k}$ itself). It can be seen as an extension to the ATV prior, just with a different matrix $D$ where more edges are considered for every voxel instead of just three. Hence its proximal operator is similar to Eq. (2). It has been shown [5] to produce excellent results in stochastic tomography reconstruction.

## 4 Structure Tensor Prior (STP)

### 4.1 Structure Tensor

The structure tensor [6] $S_K(x_i) \in \mathbb{S}^3_+$ for a 3D volume at voxel $i$ is a $3 \times 3$ positive semi-definite matrix that captures the local structure around a voxel, and is defined as

$$\begin{aligned} S_K(x_i) &= \sum_{j \in \mathcal{N}(q_i)} K(q_j - q_i) \begin{bmatrix} (\delta_j^1)^2 & \delta_j^1\delta_j^2 & \delta_j^1\delta_j^3 \\ \delta_j^1\delta_j^2 & (\delta_j^2)^2 & \delta_j^2\delta_j^3 \\ \delta_j^1\delta_j^3 & \delta_j^2\delta_j^3 & (\delta_j^3)^2 \end{bmatrix} \\ &= \sum_{j \in \mathcal{N}(q_i)} K(q_j - q_i) \left(\nabla x_j \nabla x_j^T\right), \end{aligned} \tag{5}$$

where $q_i = [i_1, i_2, i_3]^T \in \mathbb{R}^3$ is the coordinates of voxel $i$ and $x_i = x_{i_1,i_2,i_3}$ is the voxel value, $K(q_j - q_i) : \mathbb{R}^3 \to \mathbb{R}$ is a 3D rotationally-symmetric smoothing kernel that down-weights the contributions of voxel $j$ in the set $\mathcal{N}(q_i)$ of the $l$ neighboors of the voxel $i$ and $\nabla x_j \in \mathbb{R}^3$ is the local gradient at voxel $j$. $\delta_j^k = \nabla^k x_j$ is its $k^{th}$ component

$$\nabla x_j = \begin{bmatrix} \nabla^1 x_j \\ \nabla^2 x_j \\ \nabla^3 x_j \end{bmatrix} = \begin{bmatrix} x_{j_1+1,j_2,j_3} - x_{j_1,j_2,j_3} \\ x_{j_1,j_2+1,j_3} - x_{j_1,j_2,j_3} \\ x_{j_1,j_2,j_3+1} - x_{j_1,j_2,j_3} \end{bmatrix}. \tag{6}$$

So we can regard the structure tensor as a weighted average of the outer product of the local gradients at the neighborhood of the voxel.

The eigenvalue decomposition of the structure tensor $S_K(x_i)$ gives an idea about the neighborhood. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the structure matrix [7]. We have three cases:

1. $\lambda_1 \gg \lambda_2 \approx \lambda_3$: the area around the voxel is sheet-like (a surface in 3D), in which case we have one large eigenvalue and two small ones.
2. $\lambda_1 \approx \lambda_2 \gg \lambda_3$: the area around the voxel is line-like (or resembles a tube or filament), in which case we have two large eigenvalue and a small one.
3. $\lambda_1 \approx \lambda_2 \approx \lambda_3$: the area around the voxel is isotropic, in which case we have three almost equal eigenvalues. It might be that it is a constant area in which case the eigenvalues are very small, or that the changes are equal in all directions (an isotropic region) in which case they might have larger values.

### 4.2 Definition

The STP regularizer was introduced by [8, 7]. It includes the standard TV as a special case, when the smoothing kernel is a Dirac delta i.e. it is a *local* structure tensor at each voxel [7]. Intuitively, the STP tries to estimate the volume such that its structure tensor is low rank, by minimizing the deviation of voxel values in the region around it. We will introduce the STP and develop its solver by extending it from the case of images in [8, 7] to 3D volumes and by employing more efficient proximal algorithms for its computation.

The STP at a voxel $i$ is defined as the $\ell_p$ norm of the square roots of the eigenvalues of the structure tensor $S_K(x_i)$ defined in Eq. (5). Let $\Lambda\left(S_K(x_i)\right) \in \mathbb{R}^3$ be the vector of eigenvalues of $S_K(x_i)$:

$$STP_p(x_i) = \|\sqrt{\Lambda\left(S_K(x_i)\right)}\|_p = \left(\sum_{j=1}^{3} \left(\sqrt{\lambda_j}\right)^p\right)^{\frac{1}{p}}. \tag{7}$$

In the case when the kernel is the Dirac delta $K(q) = \delta(q)$, the STP becomes the standard isotropic TV regularizer i.e. the $\ell_2$ norm of the gradient vector [7] because the structure tensor simplifies to the outer product of the gradient

$$S_\delta(x_i) = \nabla x_i \nabla x_i^T$$

which has a rank of 1 and only one non-zero eigenvalue $\lambda_1$ whose value equals to the gradient magnitude square (its trace):

$$STP_p(x_i) = \sqrt{\lambda_1} = \|\nabla x_i\|_2.$$

In general, however, the structure tensor will aggregate information in the neighborhood of the voxel that will help in having a better regularization of the volume.

Next, we will define how to represent the STP in a form that fits Eq. 1 in the main paper. Define the "patch-based Jacobian" [7] as a linear map $J_K : \mathbb{R}^n \to \mathbb{R}^{nl\times 3}$ between the space of volumes and a set of weighted gradients that are computed from the $l$-neighborhood of each of the $n$ voxels. We can write the patch-based Jacobian at voxel $i$ as $J_K(x_i) \in \mathbb{R}^{l\times 3}$ by stacking the weighted local gradients side-by-side:

$$J_K(x_i) = \left[\, \kappa_{j_1} \nabla x_{j_1} \cdots \kappa_{j_l} \nabla x_{j_l} \,\right]^T \in \mathbb{R}^{l\times 3}, \tag{8}$$

where $\{j_1, \ldots, j_l\} = \mathcal{N}(q_i)$ denotes the indices of the neighbors of voxel $i$ (including $i$ itself), and $\kappa_{j_k} = \sqrt{K(q_i - q_{j_k})}$. The patch-based Jacobian for the whole volume $J_K x \in \mathbb{R}^{nl\times 3}$ is now formed by stacking "local" components $J_K(x_i)$ on top of each other

$$J_K x = \begin{bmatrix} J_K(x_1) \\ \vdots \\ J_K(x_n) \end{bmatrix} \in \mathbb{R}^{nl\times 3}. \tag{9}$$

Using this linear operator $J_K$, Equation 5 can be rewritten as follows:

$$S_K(x_i) = J_K(x_i)^T J_K(x_i), \tag{10}$$

which means that the singular values of $J_K(x_i)$ are actually equal to the square root of the eigenvalues of $S_K(x_i)$

$$\sigma\left(J_K(x_i)\right) = \sqrt{\Lambda\left(S_K(x_i)\right)}, \tag{11}$$

where $\sigma\left(J_K(x_i)\right) \in \mathbb{R}^3_+$ is the vector of singular values of patch-based Jacobian $J_K(x_i)$.

From Eq. (7) and (11) we get the definition of $STP_p$ as

$$STP_p(x) = \sum_{i=1}^{n} \|J_K(x_i)\|_{\mathcal{S}_p}, \tag{12}$$

where $\|B\|_{\mathcal{S}_p}$ is the $p-$Schatten norm of $B$ i.e. the $\ell_p$ norm of its singular values

$$\|B\|_{\mathcal{S}_p} = \|\sigma\left(B\right)\|_p. \tag{13}$$

There are usually three options for $\mathcal{S}_p$ [3]:

1. $p = 1$ is equivalent to the *nuclear norm* i.e. the sum of singular values of the matrix:
$$\mathcal{S}_1(B) = \sum_i |\sigma_i(B)| = \|B\|_*.$$
2. $p = 2$ is equivalent to the *Frobenius norm*:
$$\mathcal{S}_2(B) = \sqrt{\sum_i \sigma_i^2(B)} = \|B\|_F.$$
3. $p = \infty$ is equivalent to the *spectral norm* i.e.
$$\mathcal{S}_\infty(B) = \max_i |\sigma_i(B)| = \|B\|_2.$$

Now we can write this regularizer in a more compact *compound norm*:

$$STP_p(x) = \|J_K x\|_{1,p}$$

where the $(1, p)$ norm for a matrix $J = J_K x \in \mathbb{R}^{nl\times 3}$ is defined as

$$\|J\|_{1,p} = \|J_K x\|_{1,p} = \sum_{i=1}^{n} \|J_i\|_{\mathcal{S}_p}, \tag{14}$$

where $J_i \in \mathbb{R}^{l\times 3}$ represents the patch-based Jacobian at some voxel $i$.

Now, we can define the regularizer function $g(\cdot)$ as

$$g(J_K x) = \lambda \|J\|_{1,p}. \tag{15}$$

### 4.3 Proximal Operator for STP

To solve the reconstruction problem we need to solve

$$\min_x \|Ax - b\|_2^2 + \lambda STP_p(x) \equiv \min_x \|Ax - b\|_2^2 + \lambda \|J_K x\|_{1,p}, \tag{16}$$

where

$$f(x) = \|Ax - b\|_2^2 \tag{17}$$

represents the $\ell_2$ data fidelity term assuming Gaussian measurement noise, and

$$g(Mx) = \lambda \|J_K x\|_{1,p} \tag{18}$$

is the regularization term where $\lambda$ is the tradeoff parameter with linear mapping $M = J_K$. We will first start with the $(1, p)$ norm, whose dual norm is the $(\infty, q)$ [9] defined as

$$\|J\|_{\infty,q} = \max_{i=1\ldots n} \|J_i\|_q \quad \forall J \in \mathbb{R}^{nl\times 3} \tag{19}$$

with $q$ such that $\frac{1}{p} + \frac{1}{q} = 1$. Now we can write (18) as in [10]

$$\|J\|_{1,p} = \max_{H\in\mathcal{B}_{\infty,q}} \langle H, J\rangle_{\mathbb{R}^{nl\times 3}} \tag{20}$$

where

$$\langle H, J\rangle_{\mathbb{R}^{nl\times 3}} = \sum_i \mathrm{tr}(H_i^T J_i) \tag{21}$$

is the inner product in $\mathbb{R}^{nl\times 3}$ that induces the norm

$$\|H\|^2_{\mathbb{R}^{nl\times 3}} = \sqrt{\langle H, H\rangle_{\mathbb{R}^{nl\times 3}}}, \tag{22}$$

and $\mathcal{B}_{\infty,q}$ is the unit ball for the dual norm defined as

$$\mathcal{B}_{\infty,q} \triangleq \{H \in \mathbb{R}^{nl\times 3} : \|H_i\|_{\mathcal{S}_q} \leq 1 \quad \forall i = 1, \dots n\}. \tag{23}$$

Using (20) we can write (18) as

$$\begin{aligned}
\lambda\|J\|_{1,p} &= \lambda \max_{H\in\mathcal{B}_{\infty,q}} \langle H, J\rangle_{\mathbb{R}^{nl\times 3}} \\
&= \max_{H\in\mathcal{B}_{\infty,q}} \langle \lambda H, J\rangle_{\mathbb{R}^{nl\times 3}} \\
&= \max_{V/\lambda\in\mathcal{B}_{\infty,q}} \langle V, J\rangle_{\mathbb{R}^{nl\times 3}} \\
&= \max_{V\in\lambda\mathcal{B}_{\infty,q}} \langle V, J\rangle_{\mathbb{R}^{nl\times 3}}
\end{aligned} \tag{24}$$

where we defined $V = \lambda H \implies H = V/\lambda$ and $\lambda\mathcal{B}_{\infty,q}$ is the norm ball of radius $\lambda$ i.e.

$$\lambda\mathcal{B}_{\infty,q} \triangleq \{H \in \mathbb{R}^{nl\times 3} : \|H_i\|_{\mathcal{S}_q} \leq \lambda \quad \forall i = 1, \dots N\}. \tag{25}$$

Using (24) we can write (16) as the following saddle point problem [10]

$$\min_x \max_{H\in\lambda\mathcal{B}_{\infty,q}} \langle H, J_K x\rangle_{\mathbb{R}^{nl\times 3}} + \|Ax - b\|_2^2, \tag{26}$$

which is equivalent to

$$\min_x \max_H \underbrace{\|Ax - b\|_2^2}_{f(x)} + \langle H, J_K x\rangle_{\mathbb{R}^{nl\times 3}} - \underbrace{\imath_{\lambda\mathcal{B}_{\infty,q}}(H)}_{g*(H)} \tag{27}$$

where $\imath_{\lambda\mathcal{B}_{\infty,q}}$ is the indicator function of the norm ball of radius $\lambda$

$$\imath_{\lambda\mathcal{B}_{\infty,q}}(H) = \begin{cases} 0 & H \in \lambda\mathcal{B}_{\infty,q} \\ \infty & \text{otherwise} \end{cases} \tag{28}$$

and $g^*(\cdot)$ is the convex conjugate of $g(\cdot)$ [3]

$$g^*(H) = \max_{J\in\mathbb{R}^{nl\times 3}} \langle H, J\rangle - g(J).$$

Note that solving Eq. (27) is equivalent to solving Eq. (18), and we will use the efficient primal-dual CP algorithm [11] to solve it. We will need to define the proximal operators [3, 12] of $f$ and $g^*$:

– The proximal operator for $f(x)$ is

$$\mathbf{prox}_{\mu f}(u) = \underset{x}{\operatorname{argmin}} \|Ax - b\|_2^2 + \frac{1}{2\mu}\|x - u\|_2^2. \tag{29}$$

which can be solved directly using SART proximal operator .

– The proximal operator for $g^*(H)$

$$\mathbf{prox}_{\eta g^*}(H) = \underset{J}{\operatorname{argmin}}\, \imath_{\lambda \mathcal{B}_{\infty,q}}(J) + \frac{1}{2\eta}\|J - H\|_F^2 \tag{30}$$

which is the projection on the convex set $\lambda\mathcal{B}_{\infty,q}$, and decomposes over the $n$ components $H_i$ of $H$. The projection of $H_i$ is defined as

$$\Pi_{\lambda \mathcal{B}_{\infty,q}}(H_i) = U\hat{\Sigma}V^T \tag{31}$$

where $H_i = U\Sigma V^T$ is its SVD and $\hat{\Sigma} = \operatorname{diag}(\hat{\sigma}_i)$ with

$$\hat{\sigma} = \Pi_{\lambda \ell_q}(\sigma) \tag{32}$$

which is the projection of the vector of singular values of $H_i$ on the $q$ norm ball with radius $\lambda$. For example, when $p = 1$, we have $q = \infty$ and the projection function simplifies to simple truncation of the singular values of $H_i$

$$\hat{\sigma} = \Pi_{\lambda \ell_\infty}(\sigma) = \min(\sigma, \lambda). \tag{33}$$

The steps to solve the reconstruction problem in Eq. (16) are outlined in Algorithm 2.

### 4.4 Implementation Details

**Chambolle-Pock Primal-Dual algorithm (Algorithm 1)** We calculate the norm of matrix [13] $M$ to set the optimal values for $\mu$ and $\eta$ in our PSART framework for faster convergence. The parameter $\theta$ is set to 1 in all experiments.

**STP regularizer (Algorithm 2)** To compute the STP, we used a truncated 3D Gaussian kernel with support of $3 \times 3 \times 3$ voxels (i.e. $l = 27$) and standard deviation $\sigma = 0.5$ voxels . The linear mapping $J_K$ that computes the patch-based Jacobian is not stored explicitly, and is computed on the fly using discrete forward differences and scaling. In particular, we can decompose $J_K \in \mathbb{R}^{3nl \times n}$ into two operators

$$J_K = CD$$

where $D \in \mathbb{R}^{3n \times n}$ is a discrete forward-difference matrix that computes local gradients for the voxels and $C \in \mathbb{R}^{3nl \times 3n}$ extracts the patch-based Jacobian for each voxel over its neighborhood and scales them appropriately using the kernel $K(\cdot)$. The adjoint $J_K^* = D^*C^*$ is also computed on the fly.

However, the output of the application of the linear map $J_K$ to the volume $x$, i.e. the patch-based Jacobian $J_K(x)$, needs to be stored in the memory. In particular, for every voxel $x_i$, we need to store a matrix $J_K(x_i) \in \mathbb{R}^{l\times 3}$ that has the weighted local gradients at its $l-1$ neighbors and itself (e.g. $l = 27$ for a neighborhood of 3 pixels in each dimension). This means that we need a storage of 82 times the size of the volume to be reconstructed (81 for $J_K(x)$ plus 1 for the volume itself). Moreover, the temporary and slack variables in Algorithm 2 have also to be stored. In the experiments, the largest volume reconstructed has $690 \times 668 \times 776$ voxels, and the memory required for storing the volume and the patch-based Jacobian is 109 GB using single-precision (4-byte) floating point numbers.

## 5 SART For Solving The Data Term

For the SART algorithm, the update equation for each voxel $x_j$ in the volume $x$ is:

$$x_j^{(t+1)} = x_j^{(t)} + \alpha \frac{\sum_{i\in\mathcal{S}} c_i^{(t)} a_{ij}}{\sum_{i\in\mathcal{S}} a_{ij}}, \tag{34}$$

where

$$c_i^{(t)} = \frac{b_i - \hat{b}_i^{(t)}}{\sum_k a_{ik}} \tag{35}$$

is the normalized correction factor for ray $i$ that measures the residual between the measured projection value $b_i$ and the current estimate at iteration $t$:

$$\hat{b}_i^{(t)} = \sum_k a_{ik} x_k^{(t)}, \tag{36}$$

$\alpha$ is a relaxation parameter usually $0 < \alpha < 2$, $\mathcal{S}$ is a set of projection rays under consideration, and $a_{ij}$ is the element in row $i$ and column $j$ of the system matrix $A$ and defines the contribution to ray sum $i$ from voxel $j$. Basically the update operations in the equation can be decomposed into three steps [14]:

1. **Forward projection**: computes the estimated projection $\hat{b}_i^{(t)}$ for each ray $i$ from the current volume $x^{(t)}$ (Eq.(36)). This corresponds to a volume rendering operation.
2. **Correction**: computes $c_i^{(t)}$, the normalized deviation of this estimate from the true projection $b_i$, where the correction is normalized by the contribution of this ray to all the voxels it goes through (Eq.(35)).
3. **Backprojection**: where this correction factor is distributed back to all the voxels that contribute to this ray sum (Eq.(34)).

### 5.1 Definition

We will show how to use SART to solve the data term proximal operator. In particular, we want to solve

$$\text{prox}_{\lambda f}(u) = \underset{x}{\text{argmin}} \, \|Ax - b\|_2^2 + \frac{1}{2\lambda}\|x - u\|_2^2. \tag{37}$$

Recall that SART solves a minimum norm problem. Eq. (37) is equivalent to solving the following problem

$$\min_x 2\lambda\|Ax - b\|_2^2 + \|x - u\|_2^2. \tag{38}$$

We will introduce new variables as follows: let $y = \sqrt{2\lambda}(p - Ax)$ and $z = x - u$. The problem now becomes

$$\begin{aligned} \min_{y,z} \quad & \|y\|_2^2 + \|z\|_2^2 \\ \text{subject to } & y + \sqrt{2\lambda}Az = \sqrt{2\lambda}(p - Au). \end{aligned} \tag{39}$$

Rewriting Eq. (39) we arrive at the system

$$\begin{aligned} \min_{y,z} \quad & \left\| \begin{bmatrix} y \\ z \end{bmatrix} \right\|_2^2 \\ \text{subject to } & \begin{bmatrix} I & \sqrt{2\lambda}A \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \sqrt{2\lambda}\,(p - Au)\,, \end{aligned}$$

which can be written as

$$\begin{aligned} \min_{\tilde{x}} \quad & \|\tilde{x}\|_2^2 \\ \text{subject to } & \tilde{A}\tilde{x} = \tilde{b}, \end{aligned} \tag{40}$$

where $\tilde{x} \in \mathbb{R}^{m+n}$, $\tilde{A} \in \mathbb{R}^{m \times m+n}$, and $\tilde{b} \in \mathbb{R}^m$. This is now an under-determined linear system, and can be solved using the SART algorithm.

### 5.2 Algorithm

Although we introduced new variables $y$ and $z$ and increased the dimensionality of the problem from $n$ to $n + m$, we can solve the modified SART efficiently with very little computational overhead. Instead of solving SART explicitly for the optimal $y^\star$and $z^\star$, we can manipulate the algorithm to solve it directly for the optimal $x^\star$. In particular, the Eq. (34) for the augmented system $\tilde{A}\tilde{x} = \tilde{b}$ becomes (by substituting all variables)

$$\begin{aligned} \tilde{x}_j^{(0)} &= \mathbf{0}, \\ \tilde{x}_j^{(t+1)} &= \tilde{x}_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sum_k \tilde{a}_{ik}\tilde{x}_k^{(t)}}{\sum_k \tilde{a}_{ik}} \tilde{a}_{ij}}{\sum_{i \in S} \tilde{a}_{ij}}, \end{aligned} \tag{41}$$

which can be expanded in terms of $y$, $z$, and $A$ as

$$y_j^{(t+1)} = y_j^{(t)} + \frac{\alpha \sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sqrt{2\lambda} \sum_k a_{ik} z_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \delta_{ij}}{1},$$

$$z_j^{(t+1)} = z_j^{(t)} + \alpha \frac{\sum_{i \in \mathcal{S}} \frac{\tilde{b}_i - \sqrt{2\lambda} \sum_k a_{ik} z_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \sqrt{2\lambda} a_{ij}}{\sqrt{2\lambda} \sum_{i \in S} a_{ij}},$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Using the fact that $z = x - u$ and simplifying we get

$$y_j^{(t+1)} = y_j^{(t)} + \alpha \sum_{i \in S} \frac{\sqrt{2\lambda} b_i - \sqrt{2\lambda} \sum_k a_{ik} x_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \delta_{ij},$$

$$x_j^{(t+1)} = x_j^{(t)} + \alpha \frac{\sum_{i \in S} \frac{\sqrt{2\lambda} b_i - \sqrt{2\lambda} \sum_k a_{ik} x_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda} \sum_k a_{ik} + 1} \sqrt{2\lambda} a_{ij}}{\sqrt{2\lambda} \sum_{i \in S} a_{ij}}.$$

Algorithm 3 summarizes the steps for the modified version of the SART algorithm used to solve the proximal operator. We note the following:

1. The initialization is different since we need to initialize $y$ and $x$.
2. The update for $y$ is very fast because only one index $y_j$ is updated for every projection pixel $i = j$.
3. The update for $x$ is very similar to standard version of the SART with the exception of the term $y_i^{(t)}$ in the formula for $c_i^{(t)}$.

## References


1. S. Rit, M. V. Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. C. Sharp, "The reconstruction toolkit (rtk), an open-source cone-beam ct reconstruction toolkit based on the insight toolkit (itk)," in *Journal of Physics: Conference Series*, vol. 489, no. 1. IOP Publishing, 2014, p. 012079.
2. W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, "Fast and flexible x-ray tomography using the astra toolbox," *Optics express*, vol. 24, no. 22, pp. 25 129–25 147, 2016.
3. N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
4. E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.
5. J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich, "Stochastic tomography and its applications in 3d imaging of mixing fluids," *ACM Trans. Graph. (Proc. SIGGRAPH 2012)*, vol. 31, no. 4, pp. 52:1–52:10, 2012.
6. J. Weickert, *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998, vol. 1.

7. S. Lefkimmiatis, A. Roussos, P. Maragos, and M. Unser, "Structure tensor total variation," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1090–1122, 2015.
8. S. Lefkimmiatis, A. Roussos, M. Unser, and P. Maragos, "Convex generalizations of total variation based on the structure tensor with applications to inverse problems," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, A. Kuijper, K. Bredies, T. Pock, and H. Bischof, Eds. Springer Berlin Heidelberg, 2013, vol. 7893, pp. 48–60. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38267-3_5
9. S. Lefkimmiatis, J. P. Ward, and M. Unser, "Hessian schatten-norm regularization for linear inverse problems," *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 1873–1888, 2013.
10. S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
11. A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.
12. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
13. E. Y. Sidky, J. H. Jørgensen, and X. Pan, "Convex optimization problem prototyping for image reconstruction in computed tomography with the chambolle–pock algorithm," *Physics in medicine and biology*, vol. 57, no. 10, p. 3065, 2012.
14. K. Mueller, R. Yagel, and J. J. Wheller, "Fast implementations of algebraic methods for three-dimensional reconstruction from cone-beam data," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 6, pp. 538–548, 1999.