1 DETAILED DERIVATIONS FOR PROXIMAL OPERATORS

In Algorithms 2 and 3 of the main text, four proximal operators were introduced: \( \text{prox}_{\lambda G^i_t} \), \( \text{prox}_{\mu F_t} \), \( \text{prox}_{\lambda G^i} \) and \( \text{prox}_{\mu F_{t_j}} \). In this section we provide a derivation of these proximal operators.

First, we simplify the notations, by denoting:

\[ \bar{w}^j = w^j + \lambda_t K_f \tilde{u}^j \]
\[ \tilde{g}^j = g^j + \lambda_t K_u \tilde{f} \]
\[ \tilde{b}_t = \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) - f_t - \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \cdot \uparrow u_t^{s+1} \]

For Algorithm 2:

Case of \( \text{prox}_{\lambda G^i_t}(w^j + \lambda_t K_f \tilde{u}^j) \)

we insert whole function into \( G \), thus we have: \( F_t(u) = 0 \) and

\[ G_t(K_f u^i) = \sum_{i=1}^{N_t} \left\| \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) - f_t \right\| \]
\[ + \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \cdot (u_t^{s} - \uparrow u_t^{s+1}) \]
\[ + \lambda_t \sum_{i=1}^{N_t} \sum_{i=x,y} \left\| \nabla_S u^i_{t+1} \right\|_H \]

where the operator \( K_f \) is defined as:

\[ K_f = \begin{pmatrix} \nabla_T^T & 0 & 0 & \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} \end{pmatrix} \]

For Algorithm 3:

with the proximal operator in the same algorithm, \( \text{prox}_{\lambda G^i} \) is defined. The second proximal operator in the same algorithm, \( \text{prox}_{\mu F_t} \) is simply the identity since \( F_t(u) = 0 \).

For Algorithm 2:

we insert whole function into \( G \), thus we have: \( F_t(u) = 0 \) and

\[ G_t(K_f u^i) = \sum_{i=1}^{N_t} \left\| \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) - f_t \right\| \]
\[ + \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \cdot (u_t^{s} - \uparrow u_t^{s+1}) \]
\[ + \lambda_t \sum_{i=1}^{N_t} \sum_{i=x,y} \left\| \nabla_S u^i_{t+1} \right\|_H \]

where the operator \( K_f \) is defined as:

\[ K_f = \begin{pmatrix} \nabla_T^T & 0 & 0 & \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} \end{pmatrix} \]

Thus the operator \( K_u \) is given by

\[ K_u = (\nabla_S, \nabla_T, W)^T = \begin{pmatrix} K_{21} & K_{22} & K_{23} \end{pmatrix} \]

Hence:

\[ \bar{w}^j = w^j + \lambda_t \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} \end{pmatrix} \tilde{u}^j \]

Now it can be shown that the problem ?? is equal to solve a saddle problem:

\[ \min \max K_f \cdot y + 0 - G_t^f(y), \]

Incorporating it into CP algorithm [Chambolle and Pock 2011], we obtain:

\[ w_t^{j+1} = \text{prox}_{\lambda G^i_t}(\bar{w}_t^{j+1}) \]
\[ w_t^{j+1} = \text{prox}_{\lambda G^i}(\bar{w}_t^{j+1}) \]

\( G_t^f(\cdot) \) is the Huber penalty function. Therefore, the proximal operator of \( G_t^f(\cdot) \) is a point-wise shrinkage operation similar to the case of the TV norm [Chambolle and Pock 2011] with an additional multiplicative term \( H_{f_1} \):

\[ H_{f_1} = - \frac{1}{1 + \lambda_t \cdot c_1} \]

The first term of \( G_t^f(\cdot) \) in Equation 1 is an affine linear L1 norm and the proximal operator can be solved [Boyd et al. 2011] directly as:

\[ w_t^{j+1} = \min(1, \max(w_t^{j+1} + \lambda_t(\bar{b}_t + \nabla_S \text{warp}(f_{t+1}, \uparrow u_t^{s+1}) \cdot \tilde{u}^j - 1))) \]

With this, \( \text{prox}_{\lambda G^i(\cdot)} \) (line 4 in Algorithm 2) is defined. The second proximal operator in the same algorithm, \( \text{prox}_{\mu F_t} \) is simply the identity since \( F_t(u) = 0 \).

For Algorithm 3:

we require the proximal operator for data term \( G_u(\cdot) \), with

\[ G_u(K_u f) = \sum_{t=1}^{N_t} \| \nabla_S f_t \|_{H_{e_1}} + \frac{k_3}{k_2} \sum_{t=1}^{N_t-1} \| \nabla_T f_t \|_2^2 \]
\[ + \frac{k_1}{k_2} \sum_{t=1}^{N_t-1} \| \text{warp}(f_{t+1}, u_t) - f_t \|_1 \]

Thus the operator \( K_u \) is given by

\[ K_u = (\nabla_S, \nabla_T, W)^T = \begin{pmatrix} K_{21} & K_{22} & K_{23} \end{pmatrix} \]
where \( t \) is the iteration, \( \phi \) is a relaxation parameter, \( S \) 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 \( j \) and \( p_i \) is measured projection value. In practice, \( \phi \) is set as 0.5, and 3 iterations of plain SART algorithm are applied as initialization for the proposed optimization framework.

2 ADDITIONAL RESULTS

In the following we show additional results for both synthetic data and real scan data.

2.1 Quantitative evaluation with synthetic data

![Fig. 1. Synthetic deformation for a copper foam volume. (a) Initial volume obtained with real CT acquisition. The deformation is a uniform rotation, where the volume rotates from left to right. Frames (b) and (c) frames 150 and 300 of this sequence.](image)

We rotated the volume between successive projections with a fixed angle \( \phi \). Figure 1 shows frames 150 and 300 of the sequence with \( \phi = 0.1^\circ \). Different values of \( \phi \) were given in the Table 1, and demonstrate that the method starts breaking down around values of \( \phi > 0.3^\circ \). As stated in the paper, notice that these results can not necessarily be generalized to arbitrary data, since the performance of our method also depends on the amount of local volume structure.

<table>
<thead>
<tr>
<th>Metric</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>34.81</td>
<td>30.56</td>
<td>26.15</td>
<td>19.76</td>
<td>16.68</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.95</td>
<td>0.88</td>
<td>0.79</td>
<td>0.67</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 1. Calculated PSNR [dB], and SSIM for different rotation velocities \( \phi / \Delta t \).

2.2 Qualitative evaluation with real scans

Figure 2 compares our method to different alternative reconstructions for the fluid dataset. The methods are

- a standard implementation of 3D tomography using SART
- the ROF regularized 3D reconstruction, i.e. SART with an additional Total Variation prior
- SART with a Huber norm spatial gradient prior instead of TV
- SART with a Huber norm spatial prior, as well as temporal smoothing. This is a joint 4D reconstruction, but without deformation estimation and warping.
- Our full space-time tomography approach

For the fluid dataset, the mold is a stationary object, while the fluid deforms. The 3D reconstruction methods all fail to resolve the sharp geometric features of the mold as well as the bubbles in the fluid. In the video, these errors are also visible as temporal noise. The temporal smoothing prior significantly improves the reconstruction of the static mold, although there is still some noise left (see video), but it cannot significantly improve the fluid reconstruction. Our method, by comparison, reconstructs sharp, noise-free mold features as well as fluid details.

Additional visualizations for the mushrooms, dough, rose, and sugar dataset are shown in Figure 3, Figure 4, Figure 5, Figure 6, respectively.

REFERENCES


Fig. 2. Isosurface rendering results for liquid data with different methods.

Fig. 3. Slice visualization and comparison for mushrooms data,
Fig. 4. An additional visualization of the dough dataset, SART-ROF is compared with our method.

Fig. 5. Additional comparison for rose dataset between SART-ROF and our method.
Fig. 6. An additional visualization of the sugar dataset, SART-ROF is compared with our method.