Supplemental Material
Space-time Tomography for Continuously Deforming Objects

GUANGMING ZANG, RAMZI IDOUGHI, RAN TAO, GILLES LUBINEAU,
PETER WONKA, and WOLFGANG HEIDRICH,
King Abdullah University of Science And Technology, Saudi Arabia

In this document, we first provide a detailed derivation of the proximal operators used in the paper. Then, we present additional results of our experiments.

ACM Reference Format:
Guangming Zang, Ramzi Idoughi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. 2018. Supplemental Material Space-time Tomography for Continuously Deforming Objects. 1, 1 (April 2018), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 DETAILED DERIVATIONS FOR PROXIMAL OPERATORS

In Algorithms 2 and 3 of the main text, four proximal operators were introduced: $\mathbf{prox}_{\lambda_1 G_{\mathbf{f}}^*}$, $\mathbf{prox}_{\mu_1 F_{\mathbf{f}}}$, $\mathbf{prox}_{\lambda_2 G_{\mathbf{u}}^*}$ and $\mathbf{prox}_{\mu_2 F_{\mathbf{u}}}$. In this section we provide a derivation of these proximal operators. First, we simplify the notations, by denoting:

$$\begin{aligned}\tilde{\mathbf{w}}^j &= \mathbf{w}^j + \lambda_1 \mathbf{K_f}\bar{\mathbf{u}}^j \\ \tilde{\mathbf{g}}^j &= \mathbf{g}^j + \lambda_2 \mathbf{K_u}\bar{\mathbf{f}}^j \\ \tilde{\mathbf{b}_t} &= \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) - \mathbf{f}_t - \nabla_S \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \cdot \uparrow\mathbf{u}_t^{s+1}\end{aligned}$$

For Algorithm 2:
Case of $\mathbf{prox}_{\lambda_1 G_{\mathbf{f}}^*}(\mathbf{w}^j + \lambda_1 \mathbf{K_f}\bar{\mathbf{u}}^j)$
we insert whole function into G, thus we have: $F_{\mathbf{f}}(\mathbf{u}) = \mathbf{0}$ and

$$\begin{aligned}G_{\mathbf{f}}(\mathbf{K_f}\mathbf{u}^s) =& \kappa_1 \sum_{t=1}^{N_t-1} \Big\| (\mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) - \mathbf{f}_t) \\ &+ \nabla_S \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \cdot (\mathbf{u}_t^s - \uparrow\mathbf{u}_t^{s+1}) \Big\|_1 \\ &+ \kappa_4 \sum_{t=1}^{N_t-1} \sum_{i=x,y,z} \left\| \nabla_S \mathbf{u}_{t,i}^s \right\|_{\mathrm{H}_\tau}\end{aligned} \quad (1)$$

where the operator $\mathbf{K_f}$ is defined as:

$$\mathbf{K_f} = \begin{pmatrix} \nabla_S^T & 0 & 0 & \nabla_x \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \\ 0 & \nabla_S^T & 0 & \nabla_y \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \\ 0 & 0 & \nabla_S^T & \nabla_z \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \end{pmatrix}^T = \begin{pmatrix} \mathbf{K_{11}} \\ \mathbf{K_{12}} \\ \mathbf{K_{13}} \\ \mathbf{K_{14}} \end{pmatrix} \quad (2)$$

Hence:

$$\tilde{\mathbf{w}}^j = \mathbf{w}^j + \lambda_1 \begin{pmatrix} \mathbf{K_{11}} \\ \mathbf{K_{12}} \\ \mathbf{K_{13}} \\ \mathbf{K_{14}} \end{pmatrix} \bar{\mathbf{u}}^j = \begin{pmatrix} \tilde{\mathbf{w}}_1^{j+1} \\ \tilde{\mathbf{w}}_2^{j+1} \\ \tilde{\mathbf{w}}_3^{j+1} \\ \tilde{\mathbf{w}}_4^{j+1} \end{pmatrix} \quad (3)$$

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

$$\min_x \max_y \mathbf{K_f} \cdot \mathbf{y} + \mathbf{0} - G_{\mathbf{f}}^*(\mathbf{y}), \quad (4)$$

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

$$\mathbf{w}_{1,2,3}^{j+1} = \mathbf{prox}_{\lambda_1 G_{\mathbf{f}}^*} \begin{pmatrix} \tilde{\mathbf{w}}_1^{j+1} \\ \tilde{\mathbf{w}}_2^{j+1} \\ \tilde{\mathbf{w}}_3^{j+1} \end{pmatrix} \quad (5)$$

$$\mathbf{w}_4^{j+1} = \mathbf{prox}_{\lambda_1 G_{\mathbf{f}}^*}(\tilde{\mathbf{w}}_4^{j+1}) \quad (6)$$

$G_{\mathbf{f}}(\cdot)$ is the Huber penalty function. Therefore, the proximal operator of $G_{\mathbf{f}}^*(\cdot)$ is a point-wise shrinkage operation similar to the case of the TV norm [Chambolle and Pock 2011] with an additional multiplicative term H_{f1} :

$$H_{f1} = \frac{1}{1 + \lambda_1 \cdot \epsilon_1 \frac{\kappa_1}{\kappa_4}}$$

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

$$\mathbf{w}_4^{j+1} = \min(1, \max(\tilde{\mathbf{w}}_4^{j+1} + \lambda_1(\tilde{\mathbf{b}}_t + \nabla_S \mathrm{warp}(\mathbf{f}_{t+1}, \uparrow\mathbf{u}_t^{s+1}) \cdot \bar{\mathbf{u}}^j), -1))$$

With this, $\mathbf{prox}_{\lambda_1 G^*}(\cdot)$ (line 4 in Algorithm 2) is defined. The second proximal operator in the same algorithm, $\mathbf{prox}_{\mu_1 F_{\mathbf{f}}}(\cdot)$ is simply the identity since $F_{\mathbf{f}}(\mathbf{u}) = \mathbf{0}$.

For Algorithm 3: We require the proximal operator for data term $G_{\mathbf{u}}(\cdot)$, with

$$\begin{aligned}G_{\mathbf{u}}(\mathbf{K_u}\mathbf{f}) &= \sum_{t=1}^{N_t} \|\nabla_S \mathbf{f}_t\|_{\mathrm{H}_{\epsilon_1}} + \frac{\kappa_3}{\kappa_2} \sum_{t=1}^{N_t-1} \|\nabla_T \mathbf{f}_t\|_2^2 \\ &+ \frac{\kappa_1}{\kappa_2} \sum_{t=1}^{N_t-1} \|\mathrm{warp}(\mathbf{f}_{t+1}, \mathbf{u}_t) - \mathbf{f}_t\|_1\end{aligned} \quad (7)$$

Authors' address: Guangming Zang; Ramzi Idoughi; Ran Tao; Gilles Lubineau; Peter Wonka; Wolfgang Heidrich,
King Abdullah University of Science And Technology, Thuwal, 23955-6900, Saudi Arabia, Email:<firstname>.<lastname>@kaust.edu.sa.

XXXX-XXXX/2018/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Thus the operator $\mathbf{K_u}$ is given by

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

and

$$\tilde{\mathbf{g}}^j = \mathbf{g}^j + \lambda_2 \begin{pmatrix}\mathbf{K_{21}}\\ \mathbf{K_{22}}\\ \mathbf{K_{23}}\end{pmatrix} \bar{\mathbf{f}}^j = \begin{pmatrix}\tilde{\mathbf{g}}_1^j\\ \tilde{\mathbf{g}}_2^j\\ \tilde{\mathbf{g}}_3^j\end{pmatrix} \tag{8}$$

Incorporating these definitions into the CP algorithm [Chambolle and Pock 2011], we obtain:

$$\begin{aligned}\mathbf{g}_1^{j+1} &= \mathbf{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_1^j)\\ \mathbf{g}_2^{j+1} &= \mathbf{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_2^j)\\ \mathbf{g}_3^{j+1} &= \mathbf{prox}_{\lambda_2 G_\mathbf{u}^*}(\tilde{\mathbf{g}}_3^j)\end{aligned} \tag{9}$$

Just as in Equation 5, the solution for Equation 9 is a point-wise shrinkage operation multiplied by a Huber factor H_{f2}. For Equation 9, the solution is:

$$\mathbf{g}_p^{j+1} = \frac{\kappa_3}{\kappa_3 + 2\kappa_2\lambda_2}\tilde{\mathbf{g}}_p^j \quad p = 1, 2, 3 \tag{10}$$

The proximal operator for data term $F_\mathbf{u}(\cdot)$ is

$$\mathbf{prox}_{\lambda_2 F_\mathbf{u}}(u) = \underset{\mathbf{f}}{\operatorname{argmin}} \|\mathbf{Af} - \mathbf{p}\|_2^2 + \frac{1}{2\lambda_2}\|\mathbf{f} - u\|_2^2. \tag{11}$$

This is a least squares problem, and we use SART solving this function iteratively in a matrix free manner[Aly et al. 2016]. The update equation for each voxel $\mathbf{f}'_j$ in the volume $\mathbf{f}'$ is:

$$\mathbf{f}_j'^{(t+1)} = \mathbf{f}_j'^{(t)} + \varphi \frac{\sum_{i\in S} \frac{\sqrt{2\lambda_2}p_i - \sqrt{2\lambda_2}\sum_k a_{ik}x_k^{(t)} - y_i^{(t)}}{\sqrt{2\lambda_2}\sum_k a_{ik}+1} a_{ij}}{\sum_{i\in S} a_{ij}}. \tag{12}$$

where t is the iteration, φ is a relaxation parameter , $\mathcal{S}$ 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. and p_i is measured projection value. In practice, φ 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

We rotated the volume between successive projections with a fixed angle ϕ. Figure 1 shows frames 150 and 300 of the sequence with $\phi = 0.1°$. Different values of ϕ were given in the Table 1, and demonstrate that the method starts breaking down around values of $\phi > 0.3°$. 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.

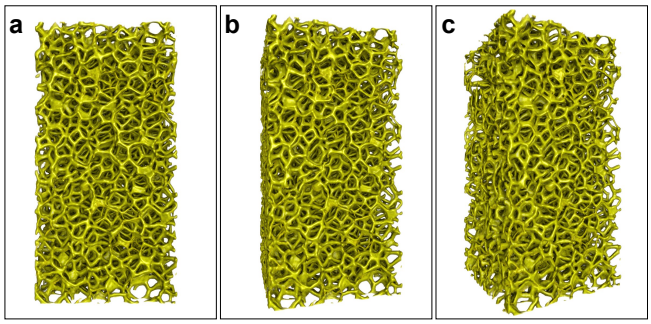

Fig. 1. Synthetic deformation for a copper foam volume. (**a**) Initial volume obtained with real CT acquisition. The deformation is a uniform rotation, where the volume rotates from left to right. Frames (**b**) and (**c**) frames 150 and 300 of this sequence.

Metric	0.1	0.2	0.3	0.4	0.5
PSNR	34.81	30.56	26.15	19.76	16.68
SSIM	0.95	0.88	0.79	0.67	0.54

Table 1. Calculated PSNR [dB], and SSIM for different rotation velocities $[\circ/\Delta t]$.

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

Mohamed Aly, Guangming Zang, Wolfgang Heidrich, and Peter Wonka. 2016. TRex: A Tomography Reconstruction Proximal Framework for Robust Sparse View X-Ray Applications. *arXiv preprint arXiv:1606.03601* (2016).

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

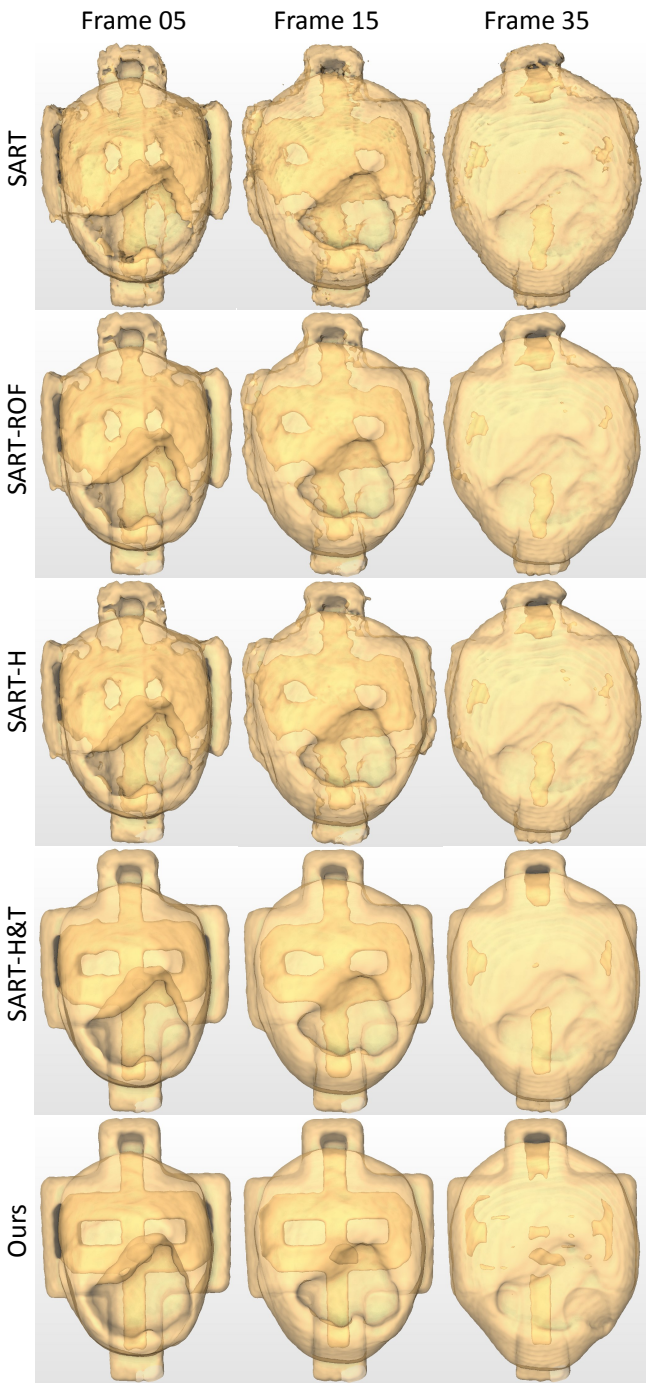

Fig. 2. Isosurface rendering results for liquid data with different methods.

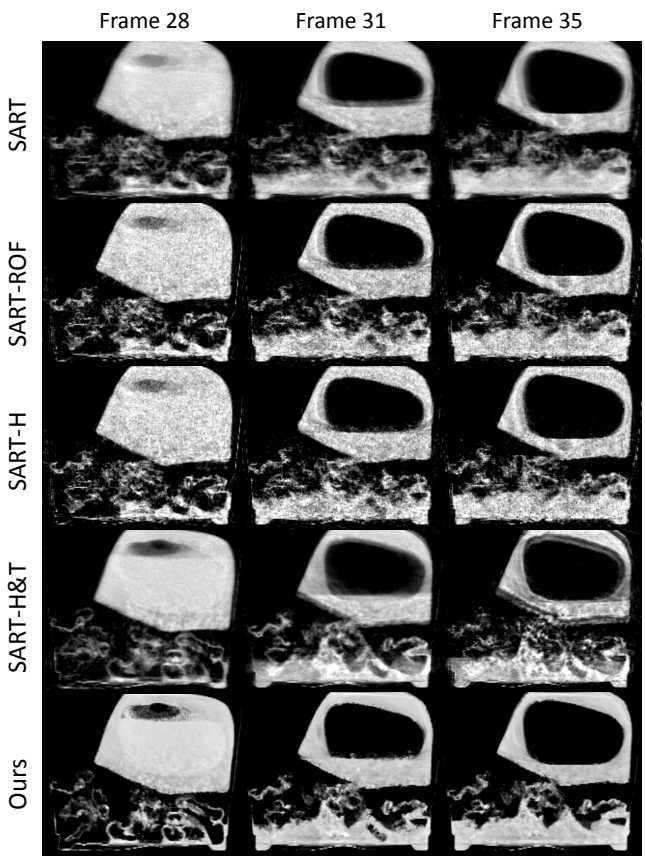

Fig. 3. Slice visualization and comparison for mushrooms data,

Antonin Chambolle and Thomas Pock. 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging and Vision* 40, 1 (2011), 120–145.

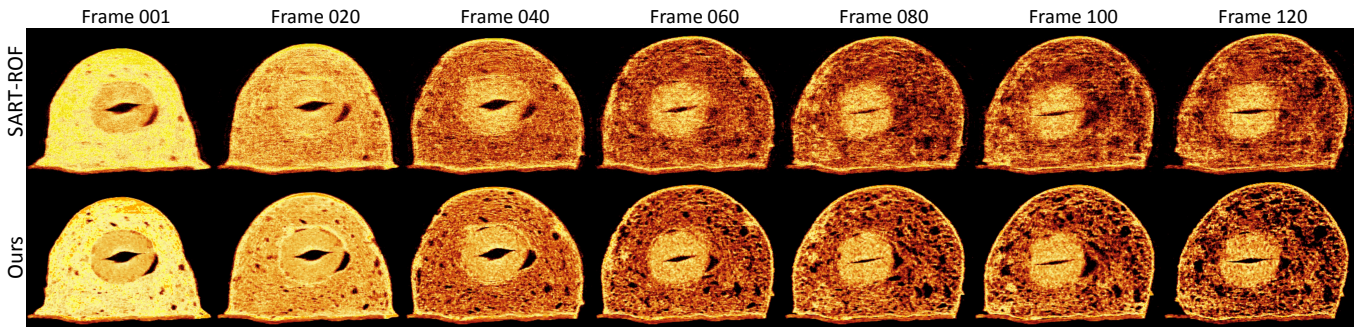

Fig. 4. An additional visualization of the dough dataset, SART-ROF is compared with our method.

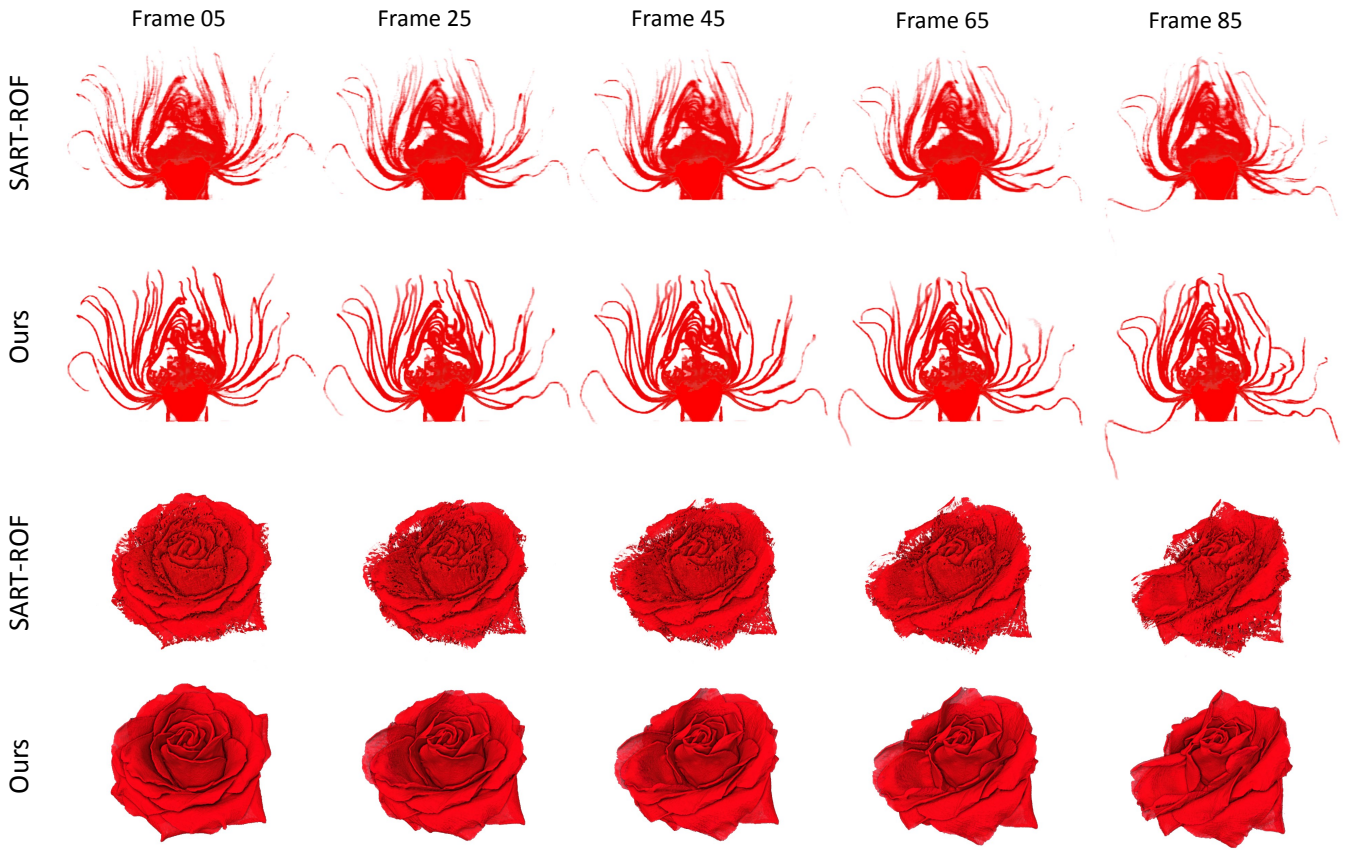

Fig. 5. Additional comparison for rose dataset between SART-ROF and our method.

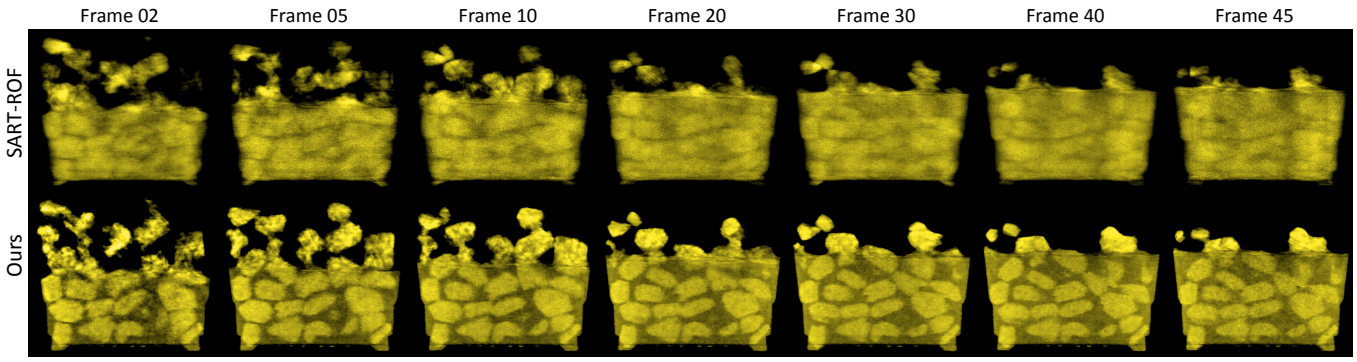

Fig. 6. An additional visualization of the sugar dataset, SART-ROF is compared with our method.