6. Supplementary Material

6.1. DSS Rendering

For one of our experimental comparisons, we used Differentiable Surface Splatting (DSS) [46] to minimize the image loss. DSS renders each point as a circle, which projects to an ellipse, where the circle's normal is the surface normal. Surface normals for a particle-represented liquid are computed using the *color field* [28] which is

$$c(\mathbf{p}^i) = \sum_{j=1}^{N} \frac{1}{\rho_i(\mathbf{p})} W(||\mathbf{p}^i - \mathbf{p}^j||, r) \tag{21}$$

at $\mathbf{p}^i \in \mathbb{R}^3$. The surface normals should point outwards from the reconstructed liquid which results in a negative change in color field. Therefore the normal is set to:

$$\mathbf{n}^i = -\left(\frac{\partial c(\mathbf{p}^i)}{\partial \mathbf{p}^i}\right) \Big/ \left|\left|\frac{\partial c(\mathbf{p}^i)}{\partial \mathbf{p}^i}\right|\right| \tag{22}$$

for particle $\mathbf{p}^i$. To compute the liquid volume color's gradient, the following expression is used:

$$\frac{\partial c(\mathbf{p}^i)}{\partial \mathbf{p}^i} = \sum_{j=1}^{N} \frac{1}{\rho_i} \frac{\partial W(||\mathbf{p}^i - \mathbf{p}^j||, r)}{\partial ||\mathbf{p}^i - \mathbf{p}^j||} \left(\frac{\mathbf{p}^i - \mathbf{p}^j}{||\mathbf{p}^k - \mathbf{p}^j||}\right) \tag{23}$$

after applying the chain rule to (21).

Laplacian smoothing is applied for more consistent normals, similar to [47], by averaging the particle positions the color field is being evaluated about in (21). The particle averages are computed as

$$\overline{\mathbf{p}}^j = (1 - \lambda_l)\mathbf{p}^j + \lambda_l \frac{\sum_{k=1}^{N} \mathbf{p}^k W(||\mathbf{p}^j - \mathbf{p}^k||, r)}{\sum_{k=1}^{N} W(||\mathbf{p}^j - \mathbf{p}^k||, r)} \tag{24}$$

where λ_l is the Laplacian average weight and $\overline{\mathbf{p}}^j$ replaces $\mathbf{p}^j$ in (21). In low particle count situations, the normal computation in (22) can produce undesirable effects such as artifacts on the edges of the liquids. To account for this, the evaluation of points $\mathbf{p}^i$ in (22) are given a small offset towards the virtual camera which will eventually render the surface. The offset is computed as:

$$\Delta\overline{\mathbf{p}}^i = \lambda_c \frac{\mathbf{c} - \overline{\mathbf{p}}^i}{||\mathbf{c} - \overline{\mathbf{p}}^i||} \tag{25}$$

where $\mathbf{c} \in \mathbb{R}^3$ is the position of the virtual camera, λ_c is the amount of the offset, and $\Delta\overline{\mathbf{p}}^i$ is added to $\mathbf{p}^i$ in (22).

The particle position, normal pairs, $\{\mathbf{p}^i, \mathbf{n}^i\}_{i=1}^N$, are directly fed into the DSS which renders each point, $\mathbf{p}^i$, as a circle whose plane is tangent to its normal, $\mathbf{n}^i$. The circles are projected to ellipses, denoted as $\mathcal{E}(\mathbf{p}^i, \mathbf{n}^i)$, and averaged with their neighboring projected circles, hence being called Elliptical Weighted Averaging (EWA). In the problem formulation for this work, we assume only knowledge of an observed visibility mask $\mathbb{I}$. Therefore, we simplify the rendering by not conducting the EWA and only render a surface mask from the projected ellipses. Written mathematically, the masked image at pixel $[u, v]^\top$ from a single particle and normal pair is: DSS computes each

$$h_{u,v}(\mathbf{p}^i, \mathbf{n}^i) = \begin{cases} 1 & \text{if } [u,v]^\top \in \mathcal{E}(\mathbf{p}^i, \mathbf{n}^i) \\ 0 & \text{if } \mathbf{p}^i \text{ is occluded} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

The rendered surface is evaluated as a summation of all the masked images from (26):

$$\hat{\mathbb{I}}_{u,v}(\mathbf{p}) = \eta_i \sum_{i=1}^{N_s} h_{u,v}(\mathbf{p}^i, \mathbf{n}^i) \tag{27}$$

where η_i normalizes the pixel value. Finally, gradients of the rendered liquid surface with respect to particle positions are computed using the approximation presented by Yifan et al. to minimize the image loss [46]. The normal smoothing values are set to $\lambda_l = 0.2$ and $\lambda_c = 0.2r$, and the original proposed kernels are used for (21) [28] and (24) [47]. The gradient step size and its threshold for detecting a local-minima are set to $\alpha_{\mathbb{I}} = 10^{-4}$ and $\lambda_s = 0.2$ respectively.

6.2. Schenck and Fox Constraints

Schenck and Fox previously proposed liquid position constraints that represent: pressure, cohesion, and surface tension [41]. These constraints replaced the proposed density constraint from (5) for comparison in our experiments. This is done by replacing $\Delta\mathbf{p}_\rho$ to solve (5) in lines 7 and 8 in Algorithm 1 with:

$$\Delta\mathbf{p}_p + \alpha_c \Delta\mathbf{p}_c + \alpha_s \Delta\mathbf{p}_s \tag{28}$$

where $\Delta\mathbf{p}_p, \Delta\mathbf{p}_c, \Delta\mathbf{p}_s$ solve the pressure, cohesion, and surface tension constraints respectively and α_c, α_s are the cohesion and surface tension weights respectively. Refer to the original paper for exact expressions to the constraint solutions [41]. The cohesion and surface tension weights are optimized for in the original work to conduct real-to-sim registration. However, this cannot be done with our problem setup because it requires prior information on the amount of liquid volume there is (i.e. how many particles there are). Therefore, instead the weights are preset to $\alpha_c = 0.05$ and $\alpha_s = 0$ (the surface tension constraint only yielded unstable behavior so it was turned off).

6.3. Source Estimation

A simple single, static source estimation technique is developed to highlight how the proposed method can be extended. Let $\hat{\mathbf{s}}_t \in \mathbb{R}^3$ be the estimated liquid source location

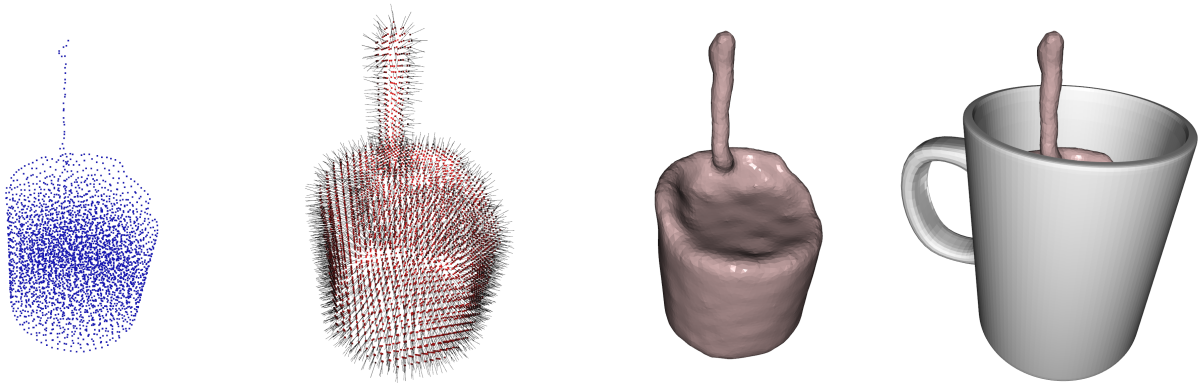

Figure 9. From left to right, the sequence shows the steps for mesh generation from the reconstructed liquid. The left-most figure is the reconstructed liquid in particle representation. The next figure shows the densely generated surface points and normals from the reconstructed liquid. The last two figures show the generated surface mesh from the surface points and normals without and with the collision mesh.

in the camera frame at time t and particles are inserted according to an estimated flow rate of $\hat{f}_t$ particles per timestep at the source location. Note that no velocity prediction is conducted for the inserted particles as there is no initial velocity. This reduces the number of parameters to estimate to $\hat{\mathbf{s}}_t$ and $\hat{f}_t$.

To update the liquid source location, $\hat{\mathbf{s}}_t$, we compare the source particle locations after completing the optimization in (1) against their pre-optimized location. Let the initial and optimized particle locations emitted from the source be denoted as $\mathbf{p}_s^n \in \mathbb{R}^3$ for $n = 1, \ldots, \hat{f}_t$ and $\mathbf{p}_s^{n*} \in \mathbb{R}^3$ for $n = 1, \ldots, \hat{f}_t$ respectively. Then the update rule given to the source location is:

$$\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}_t + \frac{\alpha_{\hat{\mathbf{s}}}}{\hat{f}_t} \sum_{n=1}^{\hat{f}_t} (\mathbf{p}_s^{i*} - \mathbf{p}_s^i) \tag{29}$$

where $\alpha_{\hat{\mathbf{s}}}$ is adjusted according to:

$$\alpha_{\hat{\mathbf{s}}} = 1 / \sum_{i=1}^{t-1} \hat{f}_i \tag{30}$$

so the source becomes less adjusted as more particles have been inserted since the source is assumed stationary.

The liquid source rate, $\hat{f}_t$, has an integer effect on the reconstruction, and we adjust it at every time step based on how many particles are duplicated or removed during the optimization of (1) after inputting the source particles for that timestep. The expression is:

$$\hat{f}_t = \alpha_{\hat{f}} \Delta N_t + \hat{f}_{t-1} - \lambda_f \tag{31}$$

where ΔN_t is the cumulative increase of particles (e.g. could be negative if particles are removed) at timestep t, $\alpha_{\hat{f}}$ adjusts the reaction rate to the insertion/removal of particles, and λ_f is a constant decay rate. Note that $\hat{f}_t$ is estimated as a non-integer value, however is applied as an integer by rounding (i.e. only an integer number of particles can be inserted per timestep). The decay rate, λ_f is used ensures stability by driving the flow rate to 0 when no new information from ΔN_t can be leveraged. The reaction rate and decay rates are set to $\alpha_f = 0.1$ and $\lambda_f = \alpha_{\hat{f}}/2$ respectively.

6.4. Mesh Generation

For visualization purposes, the reconstructed liquid can be converted to a surface mesh. A dense, uniformly spaced, grid of 3D points is generated. Surface points, $\mathbf{g}^k$, from the grid points are then selected by thresholding the gradient of the color field [28]:

$$\partial c(\mathbf{g}) / \partial \mathbf{g}^k \geq \lambda_g \tag{32}$$

where the color field, $c(\cdot)$, is defined in (21) and λ_g is the threshold. The surface normals for each surface point is computed the same as (22). The collection of surface points and normals are then converted to a mesh using Open3D's implementation of [49] Poisson surface reconstruction [20]. Fig. 9 shows an example of this process. The grid points, which the surface points are selected from, are spaced at 3mm, the gradient threshold, λ_g, is set to 0.5, and the depth for Poission surface reconstruction is set to 12. Note that figures of particles and mesh renderings in this paper are done with Open3D [49].

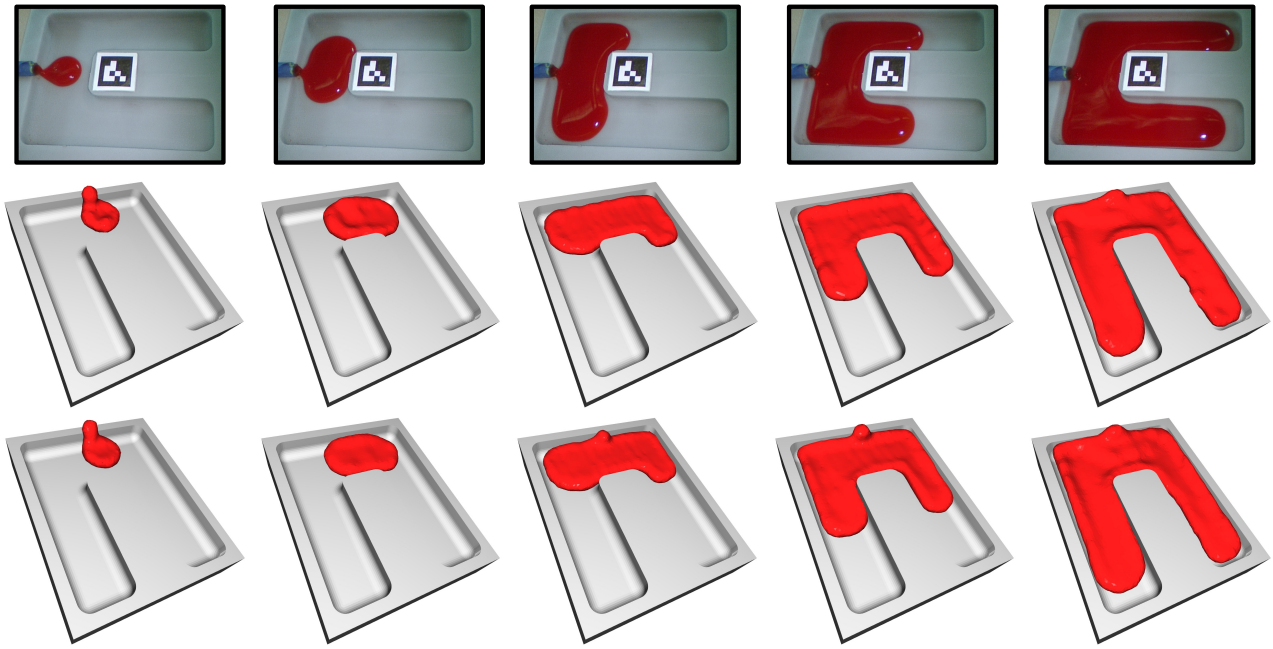

Figure 10. Sequence of reconstruction results from Endoscopic Trail 3 where the rows from top to bottom show: endoscopic image, our complete approach, and our source approach.

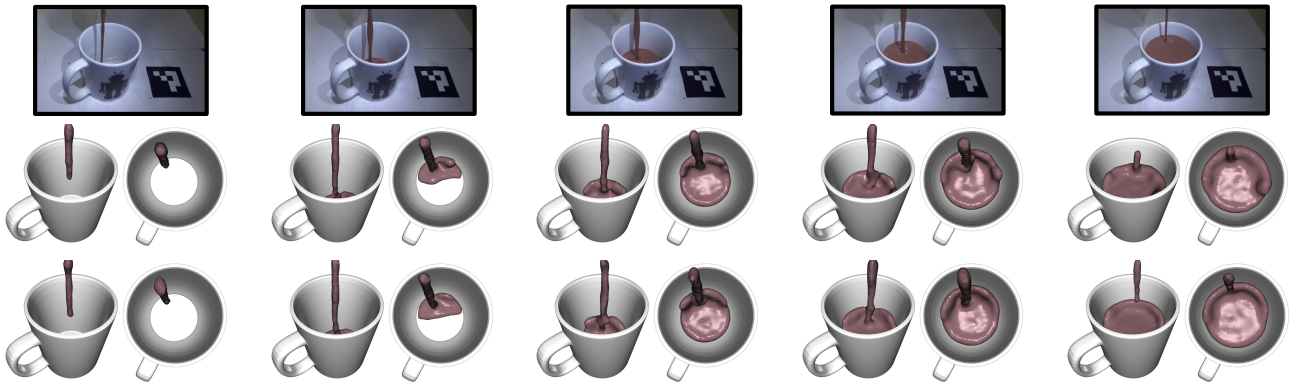

Figure 11. Sequence of reconstruction results from Pouring Milk dataset where the rows from top to bottom show: image, our complete approach, and our source approach.

Method	Simulation	Endo Trail 1	Endo Trial 2	Endo Trail 3	Pouring
No Constraints [22]	-0.03 ± 0.12	-0.25 ± 0.20	-0.09 ± 0.23	-0.01 ± 0.29	-0.08 ± 0.14
No Density	0.89 ± 3.7	0.48 ± 0.65	1.0 ± 0.94	0.91 ± 1.0	66 ± 85
No Collision	$(-5.26 \pm 61)10^{-5}$	$(-3.2 \pm 4.9)10^{-3}$	$(-1.4 \pm 3.3)10^{-3}$	$(-3.7 \pm 8.7)10^{-4}$	$(-0.33 \pm 11)10^{-3}$
Schenck & Fox [41]	$(-2.3 \pm 83)10^{-2}$	-0.11 ± 0.04	-0.12 ± 0.05	-0.11 ± 0.04	0.21 ± 0.37
DSS [46]	$(-1.3 \pm 25)10^{-3}$	$(-4.6 \pm 8.4)10^{-3}$	$(-4.9 \pm 6.6)10^{-3}$	$(-6.4 \pm 6.4)10^{-3}$	$(-5.2 \pm 32)10^{-3}$
Uniform	$(-0.05 \pm 14)10^{-3}$	$(-2.1 \pm 4.5)10^{-3}$	$(-2.2 \pm 3.4)10^{-3}$	$(-1.4 \pm 3.8)10^{-3}$	$(1.3 \pm 7.3)10^{-3}$
No Prediction	$(-0.51 \pm 9.9)10^{-3}$	$(-3.1 \pm 4.6)10^{-3}$	$(-1.5 \pm 3.4)10^{-3}$	$(-3.5 \pm 5.0)10^{-3}$	$(-4.6 \pm 14)10^{-3}$
Ours	$(1.2 \pm 13)10^{-2}$	$(-1.7 \pm 3.8)10^{-3}$	$(-2.3 \pm 4.1)10^{-3}$	$(-1.2 \pm 3.0)10^{-3}$	$(-0.08 \pm 4.4)10^{-3}$
Our Source	$(-1.6 \pm 15)10^{-2}$	$(-1.2 \pm 3.5)10^{-3}$	$(-1.9 \pm 4.5)10^{-3}$	$(-1.6 \pm 8.5)10^{-3}$	$(-0.24 \pm 7.3)10^{-3}$

Table 1. Mean and standard deviation of the density constraint, defined in (5), for the real life experiments. The density constraint ensures incompressibility for the reconstructed liquid and should be 0 when the constraint is satisfied. These results show that when applying our constraint solver, the incompressibility property is met. Meanwhile Schenck & Fox's constraints were unable to reach similar performance.

Method	Simulation	Endo Trail 1	Endo Trial 2	Endo Trail 3	Pouring
No Constraints [22]	0.469 ± 0.133	0.907 ± 0.034	0.904 ± 0.058	0.871 ± 0.063	0.798 ± 0.167
No Density	0.822 ± 0.096	0.915 ± 0.055	0.874 ± 0.214	0.919 ± 0.025	0.867 ± 0.054
No Collision	0.410 ± 0.241	0.904 ± 0.026	0.899 ± 0.048	0.761 ± 0.177	0.869 ± 0.078
Schenck & Fox [41]	0.217 ± 0.113	0.437 ± 0.319	0.882 ± 0.052	0.830 ± 0.100	0.039 ± 0.221
DSS [46]	0.759 ± 0.123	0.916 ± 0.042	0.909 ± 0.081	0.917 ± 0.032	0.815 ± 0.079
Uniform	0.826 ± 0.127	0.900 ± 0.056	0.891 ± 0.061	0.891 ± 0.071	0.576 ± 0.205
No Prediction	0.896 ± 0.049	0.904 ± 0.031	0.899 ± 0.046	0.905 ± 0.020	0.890 ± 0.084
Ours	0.889 ± 0.049	0.902 ± 0.054	0.905 ± 0.051	0.910 ± 0.026	0.849 ± 0.071
Our Source	0.891 ± 0.041	0.911 ± 0.034	0.908 ± 0.055	0.913 ± 0.026	0.843 ± 0.061

Table 2. Mean and standard deviation of IoU for the real life experiments. The results show that our reconstruction approach is able to achieve comparable image loss performance as the best from No Constraints, No Density, No Collision and No Prediction comparisons. This implies that our approach is effective at converging in image loss with additional constraints (density and collision) and prediction.