BlendED NVIDIA Project: Team2 Track 4

Gaussian SPH Fluid: Physics-integrated 3D Gaussians for SPH Fluid Dynamics

Jinhyung Ahn, LiLeonard, Chih-Jung Hsu

Introduction

Why fluid + 3DGS a challenge?

Recent work integrates physics with 3D Gaussian Splatting (3DGS). PhysGaussian treats Gaussians as a continuum and advances them with an MPM solver: centers and covariances evolve from the deformation gradient F, and SH lobes rotate consistently, enabling a WS² ("what you see is what you simulate") pipeline. However, **liquids are governed by incompressibility**. MPM-based approaches need extra handling to enforce density and velocity constraints, and the surface-biased distribution of 3DGS becomes problematic beyond "hollow interiors." PhysGaussian itself suggests internal filling.

Our idea

We **simulate Gaussians directly** in an SPH solver. Using **DFSPH** (divergence-free + constant-density constraints), we obtain fluid-specialized behavior. We also update each Gaussian covariance Σ from the SPH velocity gradient $\nabla\mathbf{v}$ via an implicit integration rule and render the updated Gaussians directly.

Background

From Navier–Stokes to DFSPH

Governing equations

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p + \mu\nabla^2\mathbf{v} + \rho\mathbf{g}, \qquad \nabla\cdot\mathbf{v} = 0$$

These encode momentum conservation such as pressure, viscous, and gravity forces and mass conservation which represents incompressibility.

SPH discretization

- The continuum is represented by particles carrying mass, density, velocity.
- Local quantities are estimated by smoothing kernels, $W(\|\mathbf{r}_i - \mathbf{r}_j\|, h)$.
- Discrete forces are pairwise-symmetric, preserving mass and momentum, and are stable for free surfaces.

Gaussian representation

Instead of an isotropic kernel, each particle is a 3D Gaussian field:

$$g_i(\mathbf{x}) = \alpha_i \exp\left[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right],$$

with center μ_i, covariance Σ_i(anisotropic shape), and density weight α_i.

This keeps the same physical operators while improving smoothness and anisotropy for rendering.

Why DFSPH for fluid?

DFSPH enforces incompressibility at two levels in each time step:

- constant-density correction $\rho \approx \rho_0$,
- divergence-free velocity $\frac{D\rho}{Dt} \approx 0$.

It converges quickly and remains stable with practical time steps—ideal for fluid-like behavior.

Related Work

- **PhysGaussian (MPM)**: evolve Gaussians by F; rotate SH; optional internal filling to avoid hollow interiors. Strong for solids/granulars, less direct for liquids.
- **Gaussian Splashing (PBD/PBF)**: adds SPH density constraints but decouples simulation particles from render Gaussians and later interpolates; the initial Gaussians themselves are not simulated.

Gap: a unified pipeline that simulates the original Gaussians while strictly enforcing incompressibility.

Method

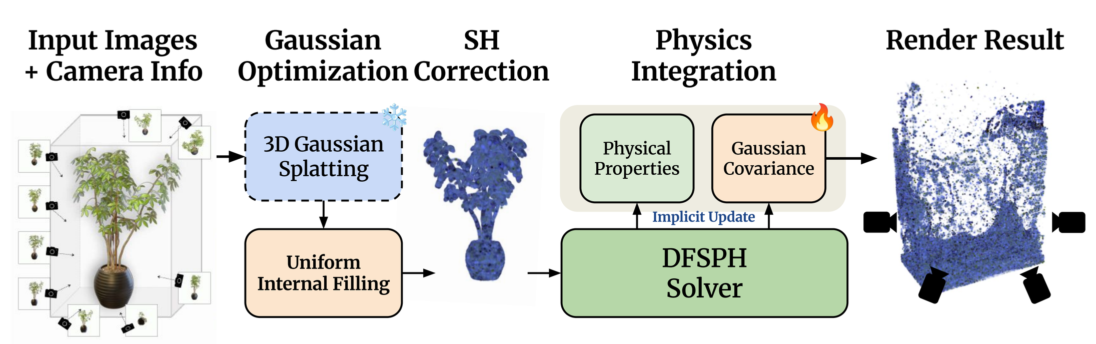

Figure 1. Method Overview 3DGS optimization → uniform internal filling (+isotropic Σ, surface subsampling) → SH color correction → DFSPH (density & divergence constraints) with implicit Σ update from SPH ∇v → Gaussian rendering.

Uniform Internal Filling

- Build a **smoothed opacity field** from surface Gaussians to counter hollow interiors.
- Evaluate a **uniform 3D grid** in the scene AABB (spacing 4r, r=SPH particle radius).
- At grid point $\mathbf{x}$, select the **k-nearest surface Gaussians**:

$$\hat{d}(\mathbf{x}) = \frac{\sum_{i\in N_k(\mathbf{x})}\exp\big(-\|\mathbf{x}-\mathbf{p}_i\|^2/(2\sigma^2)\big)\,\sigma_i}{\sum_{i\in N_k(\mathbf{x})}\exp\big(-\|\mathbf{x}-\mathbf{p}_i\|^2/(2\sigma^2)\big)}$$

- Compute a smoothed opacity from them, and **seed a particle if it exceeds a threshold** $\hat{d}(\mathbf{x}) > \sigma_{\text{th}}$.
- Initialize Gaussian attributes: copy **SH color/opacity** from the nearest surface Gaussian → set covariance $\boldsymbol{\Sigma}_0 = \operatorname{diag}(r_{\text{iso}}^2)$

DFSPH Coupling

- Apply DFSPH's two constraints each step:
  - Constant density: correct predicted density to ρ_0.
  - Divergence-free: drive $D\rho/Dt \to 0$ to remove velocity divergence.
- Warm-start iterations; feed updated positions/velocities straight to rendering.

Covariance update from SPH $\nabla\mathbf{v}$

- SPH discretization of $\nabla\mathbf{v}$:

$$\nabla\mathbf{v}_i = \sum_j \frac{m_j}{\rho_j}\left(\mathbf{v}_j - \mathbf{v}_i\right)\otimes\nabla_i W(\|\mathbf{x}_i - \mathbf{x}_j\|, h)$$

Where j is the neighbor index, m is particle mass, ρ is particle density, W is SPH kernel, and h is support radius.

- Update the covariance matrix Σ at the next time step by implicit Euler integration:

$$\boldsymbol{\Sigma}_i^{t+\Delta t} \approx (\mathbf{I} - \Delta t\,\nabla\mathbf{v}_i)^{-1}\,\boldsymbol{\Sigma}_i^t\,(\mathbf{I} - \Delta t\,\nabla\mathbf{v}_i^\top)^{-1}$$

Color handling

SH colors for static scenes are not ideal for water. We apply hue/gamma adjustment to obtain legible fluid appearance while keeping **WS²** rendering.

Results

Internal Filling Ablation Study

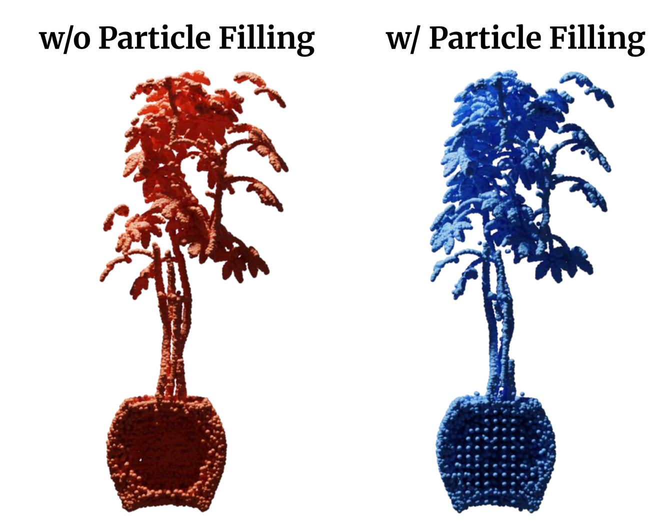

Figure 2. Internal Filling (ablation) Left: initialization without filling shows a hollow, surface-biased interior. Right: with filling, uniformly seeded interior particles stabilize the initial density for DFSPH.

Figure 2 contrasts our initialization with and without internal filling.

The right panel uses uniform interior seeding obtained by thresholding a Gaussian-smoothed opacity field.

Parameters: opacity threshold 0.40, $k = 8$ neighbors, neighbor-radius scale 3.0, and σ scale 1.0.

Comparison Experiments

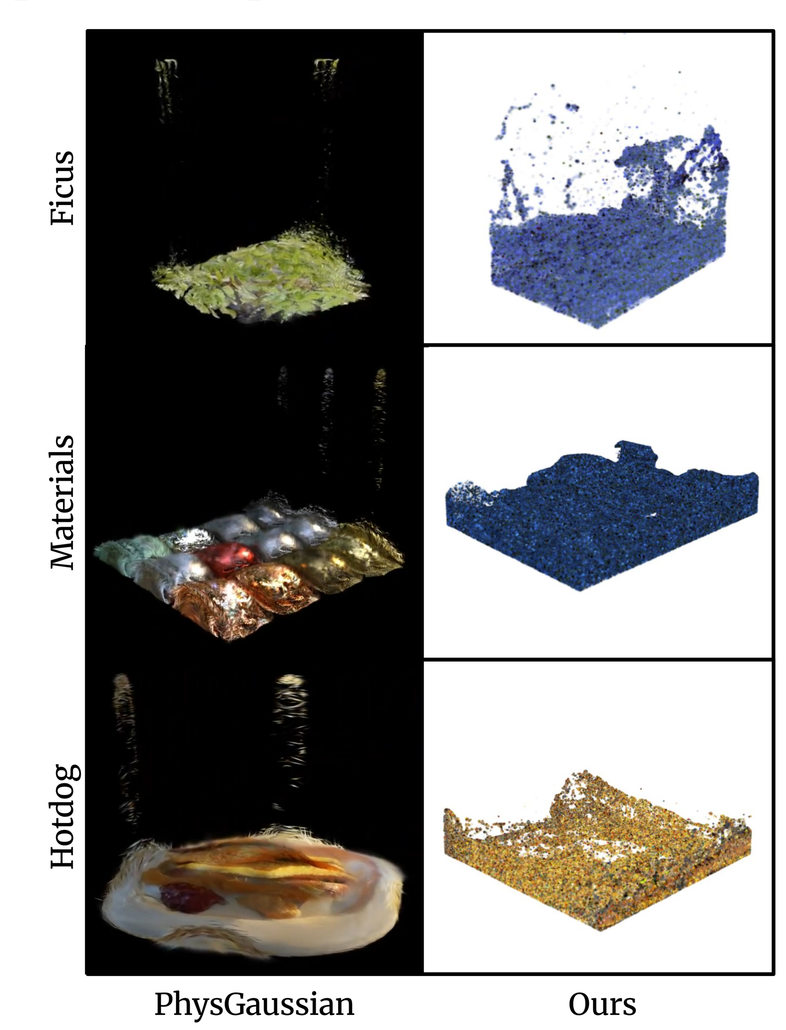

Figure 3. Comparison Three scenes from the dataset Synthetic NeRF: Materials, Hotdog, and Ficus.

Dataset: Synthetic-NeRF (Ficus, Materials, Hotdog).

Baseline (PhysGaussian): no fluid model available; evaluated with the authors' granular "sand" material.

Ours: $\Delta t = 10^{-3}$; particle radius 5×10^{-3}; termination $< 10^{-2}$ for both density error and velocity divergence.

Collisions: axis-aligned bounding box for both methods.

Rendering: SH hue/gamma correction for all scenes except Hotdog (raw SH).

Solver Convergence

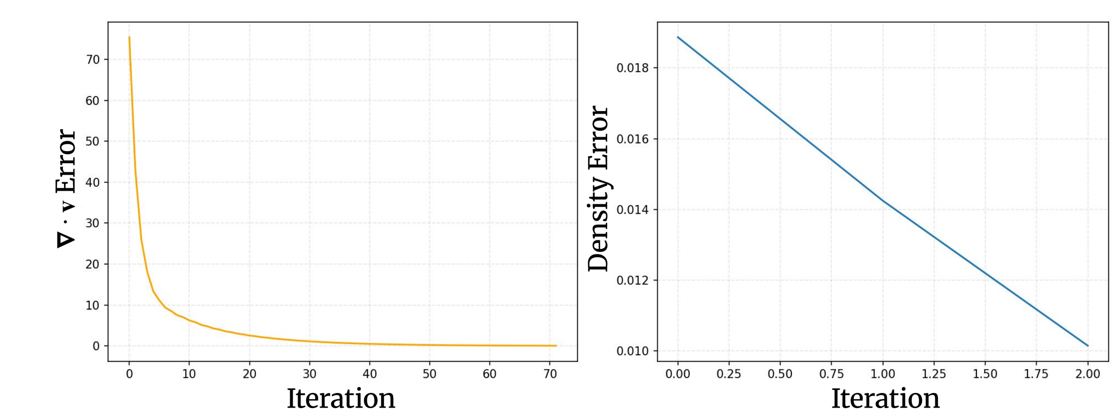

Figure 4. Error rate on frame 120 Velocity divergence on the left, and density error on the right. Both errors smoothly converges, reaching 0.01% of compression rate.

Conclusion

Summary

We present **Gaussian SPH Fluid**, a unified simulation–rendering pipeline that advances 3D Gaussians with an SPH (DFSPH) solver and renders them directly. A **uniform internal filling** converts surface-biased 3DGS into SPH-ready volumes, and an **SPH ∇v–based implicit covariance update** aligns anisotropic shapes with the local flow. The method enforces incompressibility via DFSPH, exhibiting stable evolution and clear convergence; the ablation confirms that internal filling is critical for robust initialization. Compared with MPM-based baselines, our coupling preserves liquid-like coherence and density while maintaining a single, point-based representation for simulation and rendering.

Limitations and Future Work

Sensitivity to surface-biased initialization can produce mild free-fall jitter in early frames, and appearance currently relies on SH hue/gamma adjustments rather than a fluid-specific optical model. Future work will address multi-phase flows, coupling with rigid/soft bodies, and improved optics/BSDF toward a fully end-to-end, VFX-ready pipeline.

References

[1] Jan Bender and Dan Koschier. **Divergence-free smoothed particle hydrodynamics.** In *Proceedings of the 14th ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 147–155, 2015.

[2] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, et al. **Gaussian splashing: Dynamic fluid synthesis with gaussian splatting.** *CoRR*, 2024.

[3] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. **Physgaussian: Physics-integrated 3d gaussians for generative dynamics.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024.