DecoupledGaussian: Object-Scene Decoupling for Physics-Based Interaction

Supplementary Material

A. Organization

In this paper, we introduce **DecoupledGaussian**, a fast and robust method for decoupling static objects from contact surfaces while restoring geometry and texture for improved object-scene interaction. Using the MLS-MPM simulator, our approach extends beyond rigid 3D reconstructions, enabling more dynamic and flexible applications of Gaussian Scene (GS) representations. We encourage readers to view the accompanying video for demos of the dynamic effects. This supplementary material provides detailed material parameters, methodology, and additional experiments to offer a comprehensive understanding of our approach.

Note: Figures, sections, and tables in the supplementary material are prefixed with a letter for distinction, while those without a prefix refer to content in the main paper.

B. Material Properties

We use two constitutive models from Zong et al. [15]: Fixed Corotated and Drucker-Prager Plasticity. The material parameters, including Young's modulus (E) and shear modulus (μ) (Sec. 5.1), for each case are summarized in Table A1.

Table A1. **Material Properties Configuration.**

Case	Figure	Constitutive Model	μ	E
`Bear_collisions`	Fig. 1	Fixed Corotated	0.3	3×10^6
`Bear_melting`	Fig. 1	Drucker-Prager Plasticity	0.3	3×10^6
`Kitchen`	Fig. 3	Fixed Corotated	0.3	3×10^6
`Garden_collisions`	Fig. 6	Fixed Corotated	0.3	3×10^6
`Bonsai_collisions`	Fig. 6	Fixed Corotated	0.4	2×10^6
`Figurines_collisions`	Fig. 6	Fixed Corotated	0.3	3×10^6
`Room`	Fig. 7	Fixed Corotated	0.3	3×10^6
`Truck_Bicycle`	Fig. 7	Fixed Corotated	0.3	3×10^6
`Banana`	Fig. 8	Fixed Corotated	0.3	3×10^6
`Pillow`	Fig. 8	Fixed Corotated	0.3	3×10^6
`Mustard`	Fig. 8	Fixed Corotated	0.3	3×10^6
`Bonsai`	Fig. A4	Drucker-Prager Plasticity	0.4	2×10^6
`Kitchen`	Fig. A4	Drucker-Prager Plasticity	0.3	3×10^6

C. Technique Details

C.1. Wigner D-matrix

The Wigner D-matrix [8, 12] (Sec. 4.3) $D^{(j)}_{m',m}(\alpha, \beta, \gamma)$ describes the rotation of a function on the sphere in terms of Euler angles (α, β, γ):

$$D^{(j)}_{m',m}(\alpha, \beta, \gamma) = e^{-im'\alpha} d^{(j)}_{m',m}(\beta) e^{-im\gamma},$$

where:

- j is the degree of the spherical harmonic,
- m and m' are the magnetic quantum numbers, which range from $-j$ to j,
- $d^{(j)}_{m',m}(\beta)$ is the small Wigner d-matrix, defined as:

$$d^{(j)}_{m',m}(\beta) = \sum_{k=0}^{j+m} \binom{j+m}{k} \binom{j-m}{j+m-k} \cos^{2j-k}\left(\frac{\beta}{2}\right) \sin^{k}\left(\frac{\beta}{2}\right). \quad (1)$$

SH Coefficient Transformation To rotate view-dependent spherical harmonic (SH) coefficients $\mathcal{C}'$ before simulation, we compute the Wigner D-matrix for a given rotation matrix, derived from Euler angles (α, β, γ). Let the view-dependent SH coefficients be denoted as:

$$\mathcal{C}' = \{c^{(j)}_m \mid m = -j, -j+1, \ldots, j\},$$

where $c^{(j)}_m$ corresponds to the coefficient of degree j and magnetic quantum number m.

For rotation, the transformed SH coefficients $\hat{\mathcal{C}}$ are computed as:

$$\hat{\mathcal{C}} = D^{(j)} \mathcal{C}',$$

where $D^{(j)}$ is the Wigner D-matrix of degree j. Specifically, each SH coefficient $c^{(j)}_m$ is rotated using its corresponding Wigner D-matrix element:

$$\hat{c}^{(j)}_m = \sum_{m'=-j}^{j} D^{(j)}_{m',m}(\alpha, \beta, \gamma) c^{(j)}_{m'}.$$

Here, $\hat{c}^{(j)}_m$ is the transformed coefficient, and the sum runs over all values of m' from $-j$ to j.

C.2. Joint Poisson Fields

To resolve conflicts (Sec. 4.2.1) between the indicator functions $\mathcal{X}_\mathcal{S}$ and $\mathcal{X}^\mathcal{S}_\mathcal{O}$, we first ensure that $\mathcal{S}$ remains smooth and continuous before addressing the conflicts. The process is performed iteratively as follows:

1. Identify Surface Points of $\mathcal{S}$:
   - Intersection Region:

$$\{x \mid 0.5 < \mathcal{X}_\mathcal{S}(x) < 0.6 \text{ and } \mathcal{X}^\mathcal{S}_\mathcal{O}(x) > 0.5\}$$

   - Non-Intersection Region:

$$\{x \mid 0.5 < \mathcal{X}_\mathcal{S}(x) < 0.6 \text{ and } \mathcal{X}^\mathcal{S}_\mathcal{O}(x) < 0.5\}$$

2. Compute Mean Curvature:

- Compute the mean curvature $H(x)$ at each surface point using neighboring points.

3. Adjust Surface Points in Intersection Regions:
   - For each point x in the intersection region, find the nearest point x_{closest} in the non-intersection region. If $|H(x) - H(x_{\text{closest}})| > \tau$, update:

$$\mathcal{X}_{\mathcal{S}}(x) = 0.49.$$

4. Ensure Surface Smoothness:
   - Repeat steps 1–3 for 10 iterations to achieve smoothness.
5. Resolve Conflicts in $\mathcal{X}_{\mathcal{O}}^{\mathcal{S}}$:
   - For points x where $\mathcal{X}_{\mathcal{O}}^{\mathcal{S}}(x) > 0.5$, check neighbors. If any neighbor satisfies $\mathcal{X}_{\mathcal{S}}(x) > 0.5$, update:

$$\mathcal{X}_{\mathcal{O}}^{\mathcal{S}}(x) = 0.49.$$

C.3. Normals Disambiguities

The normals of $\{\boldsymbol{k_g}\}_{g\in\mathcal{S}}$ (Sec. 4.2.1) correspond to the direction of the minimum scale factor of the flattened Gaussian. Due to ambiguity in determining the normal direction, as both directions along the shortest axis are possible, we resolve this by utilizing the training viewing direction. Specifically, we ensure the angle between the normal and viewing directions exceeds 90 degrees, as observations are made from the exterior of the surface. We then count the occurrences of each direction across training views and select the one with the highest number of votes.

C.4. Mesh2Gaussians

We bind new Gaussians to the mesh triangles (Sec. 4.2.4) as follows: for a given triangle with vertices $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$, the center of the new Gaussian is set at the centroid of the triangle, calculated as:

$$\boldsymbol{k} = \frac{1}{3}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3).$$

The normal vector $\boldsymbol{r}_1$ to the plane of the triangle is computed as:

$$\boldsymbol{r}_1 = \frac{(\boldsymbol{v}_2 - \boldsymbol{k}) \times (\boldsymbol{v}_3 - \boldsymbol{k})}{\|(\boldsymbol{v}_2 - \boldsymbol{k}) \times (\boldsymbol{v}_3 - \boldsymbol{k})\|},$$

where $\times$ denotes the cross product. The second Gaussian axis $\boldsymbol{r}_2$ is defined as:

$$\boldsymbol{r}_2 = \frac{\boldsymbol{v}_2 - \boldsymbol{k}}{\|\boldsymbol{v}_2 - \boldsymbol{k}\|}.$$

The third vector $\boldsymbol{r}_3$ is computed through a one-step Gram-Schmidt projection [1]:

$$\boldsymbol{r}_3 = \text{proj}(\boldsymbol{v}_3 - \boldsymbol{k}; \boldsymbol{r}_1, \boldsymbol{r}_2).$$

The Gaussian rotation matrix is then defined as:

$$\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3].$$

The scaling values are calculated as follows: $s_2 = \|\boldsymbol{v}_2 - \boldsymbol{k}\|$ for the direction $\boldsymbol{r}_2$, $s_3 = \|\boldsymbol{r}_3^T(\boldsymbol{v}_3 - \boldsymbol{k})\|$ for the direction $\boldsymbol{v}_3$, and $s_1 = \epsilon$, where $\epsilon = 1 \times 10^{-8}$, for the shortest axis $\boldsymbol{r}_1$ to account for the flattened 3D Gaussian.

D. Experiments

Figure A1. **Object Restoration.** Restored objects from the Decoupling benchmark are rendered from different viewpoints. From top to bottom: Banana, Pillow, and Mustard.

D.1. User Study Statistics

The statistics for each participant (Sec. 5.2) in our user study on in-the-wild video evaluation are summarized in Tab. A2. Notably, all participants rated our method the highest across three tasks: scene restoration, object restoration, and object-scene interactive simulation.

D.2. Additional Evaluations

Object Restoration As shown in Fig. A1, we present restored objects rendered from various viewpoints derived from the interactive simulation in Fig. 8. Our joint Poisson field $\mathcal{W}$ effectively repairs incomplete and broken surfaces of $\mathcal{O}$, outperforming PhysGaussian [13] and GIC [2]. By bounding the dense points $P_{\mathcal{O}}$ within the object's interior, our method achieves superior restoration of both texture and geometry compared to these approaches.

Interactive Simulation As shown in Fig. A4, we provide additional qualitative evaluations of our method applied to object-scene interactive simulation, which are not included in the main paper or supplementary video.

Table A2. **User Study Statistics.** U1, U2, ..., U10 represent the IDs of individual participants.

Metrics	Methods	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
Scene Restoration (SRQ ↑)	GScream	1.80	2.00	2.00	2.40	2.20	1.80	2.20	1.60	2.20	1.20
	VR-GS	2.00	2.20	2.20	2.40	2.20	2.00	2.60	1.80	2.20	1.60
	Ours	**3.20**	**3.60**	**3.60**	**4.00**	**3.60**	**3.20**	**3.40**	**3.80**	**3.60**	**2.80**
Object Restoration (ORQ ↑)	PhysGaussian	1.50	1.50	1.00	1.50	1.00	1.25	1.50	1.50	1.50	1.75
	GIC	2.00	1.75	1.50	1.75	1.25	2.00	1.25	2.00	1.50	1.00
	Ours	**4.00**	**4.50**	**4.25**	**4.00**	**3.25**	**4.00**	**3.75**	**4.25**	**4.00**	**4.25**
Interactive Simulation (ISF ↑)	VR-GS($\mathcal{S}$)+PhysGaussian($\mathcal{O}$)	1.50	1.50	1.25	1.75	1.50	1.50	1.25	1.75	1.50	1.50
	Ours($\mathcal{S}$)+PhysGaussian($\mathcal{O}$)	2.50	2.75	2.25	2.50	2.00	3.00	2.50	3.00	2.75	2.75
	Ours($\mathcal{S}$)+GIC($\mathcal{O}$)	3.00	3.00	2.50	3.00	2.75	2.25	2.75	2.50	2.50	3.00
	Ours($\mathcal{S}$)+Ours($\mathcal{O}$)	**4.25**	**4.50**	**4.25**	**4.00**	**4.50**	**4.25**	**4.50**	**4.25**	**4.50**	**4.50**

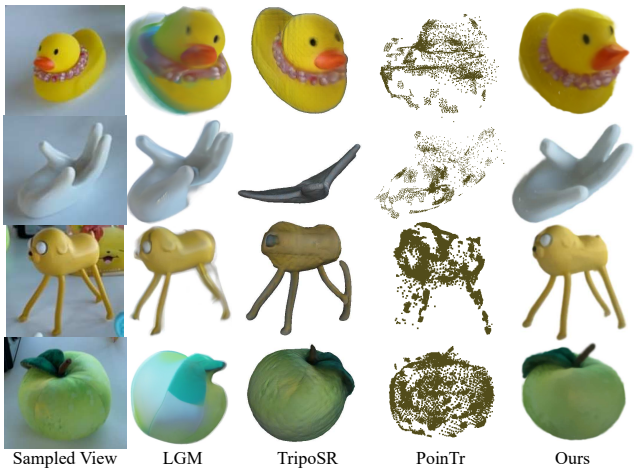

Figure A2. **Generative Models.** Results for state-of-the-art models using a single image as input: Gaussian generative model LGM [9] and mesh generative model TripoSR [10]; and proxy points as input: point cloud completion model PoinTr [14].

Figure A3. **Additional Methods.** Results for GaussianEditor [3] and Infusion [6] reveal significant limitations when using their provided implementations.

Simulator Our MLS-MPM implementation leverages both NVIDIA Warp [7] and Taichi [4]. We performed timing comparisons using the Wolf, Pillow2Sofa, and VaseDeck datasets provided by PhysGaussian [13]. The results are shown in Table A3, where the computation time per update timestep is expressed in 10^{-3} s.

Table A3. Timing comparison of MLS-MPM simulation engines. Computation time per update timestep (in 10^{-3} s).

Method	**Wolf**	**Pillow2Sofa**	**VaseDeck**
Taichi [4]	2.560	**1.390**	0.583
Warp [7]	**2.290**	2.510	**0.538**

Generative Models We evaluate state-of-the-art generative models for mesh generation [10, 11], Gaussian generation [9], and point cloud completion [5, 14]. The first two models take a single frame as input, while the point cloud completion models use our proposed proxy points $\mathcal{P}_{\mathcal{O}}$ as input. Although some models generate reasonable shapes (see Fig. A2), they often fail for untrained inputs, exhibiting inaccuracies in geometry and texture that diverge from the target properties in the raw scene.

Additional Methods We evaluate two recent approaches, GaussianEditor [3] and Infusion [6], for scene $\mathcal{S}$ restoration. As shown in Fig. A3, both methods exhibit significant limitations. Infusion suffers from severe errors due to inaccurate depth estimation and projection issues in its implementation. Similarly, GaussianEditor demonstrates inconsistent segmentation propagation across views, leading to incomplete object removal or residual artifacts. Additionally, our experiments show that GaussianEditor runs approximately five times slower than the runtime reported in the original paper. These limitations have also been noted by other users

Figure A4. **Additional Interactive Simulations.** Evaluations are rendered with moving cameras. From top to bottom: Banana (collision and elasticity), Mustard (collision and elasticity), Bonsai (fracture and granular flow), and Kitchen (melting and granular flow).

on the GitHub issue channels for the respective implementations.

References

[1] Åke Björck. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316, 1994.

[2] Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. Gaussian-informed continuum for physical property identification and simulation. *Advances in Neural Information Processing Systems*, 2024.

[3] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.

[4] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Trans. Graph.*, 38(6), 2019.

[5] Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point cloud completion with pretrained text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[6] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024.

[7] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. `https://github.com/nvidia/warp`, 2022. NVIDIA GPU Technology Conference (GTC).

[8] Jie Shen, Jie Xu, and Pingwen Zhang. Approximations on so (3) by wigner d-matrix and applications. *Journal of Scientific Computing*, 74:1706–1724, 2018.

[9] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025.

[10] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d

object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

[11] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457. Springer, 2025.

[12] Eugen Wigner. Gruppentheorie und ihre anwendung auf die quantenmechanik der atomspektren. 1931.

[13] 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.

[14] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.

[15] Zeshun Zong, Xuan Li, Minchen Li, Maurizio M. Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. In *SIGGRAPH Asia 2023 Conference Papers*, New York, NY, USA, 2023. Association for Computing Machinery.