Supplementary Material: Physics-Based Interaction with 3D Objects via Video Generation

Tianyuan Zhang[1], Hong-Xing Yu[2], Rundi Wu[3], Brandon Y. Feng[1], Changxi Zheng[3], Noah Snavely[4], Jiajun Wu[2], and William T. Freeman[1]

[1] Massachusetts Institute of Technology
[2] Stanford University
[3] Columbia University
[4] Cornell University

1 Metrics

We compare the visual quality of our method with two baseline methods, PhysGaussian [9] and DreamGaussian4D [6], by computing the Frechet Video Distance (FVD) [8] against real captured videos. We compute the FVD with a 16-frame window, 2-frame stride, based on the I3D [1] model trained on the Human Kinetics Dataset [3]. All videos are resized (short edge to 144 pixels) and center-cropped to 128×128 pixels prior to FVD computation. We compare each method against real captured videos, creating 272 clips per scene for evaluation. The results are shown in Table 1.

We further compare methods using the Frechet Inception Distance (FID) [2, 5], as shown in Table 2. FID calculation incorporates all frames across all objects, totaling 4200 frames per method.

Table 1: Frechet Video Distance (FVD) between real captured video and PhysDreamer (Ours) and baseline methods (PhysGaussian [9] and DreamGaussian4D [6])

FVD (↓)	Alocasia	Carnation	Hat	Rose O.	Rose W.	Cord	Tulip	**Avg.**
Ours	272	282	54	231	640	185	228	**270.3**
PhysGaussian	560	629	50	408	961	184	586	482.6
DreamGaussian	308	359	75	200	1379	210	497	432.6

2 Effect of Poisson's ratio on synthesized motion

As mentioned in section three, we observed that Poisson's ratio has minimal effect on synthesized motions We run simulations of the same object with identical initial conditions but varying Poisson's ratios (0.05, 0.1, 0.2, and 0.3). We used

Table 2: Frechet Inception Distance (FID) between real captured video and PhysDreamer (Ours) and baseline methods (PhysGaussian [9] and DreamGaussian4D [6])

Method	**FID (↓)**
Ours	**47.7**
PhysGaussian	63.2
DreamGaussian	52.8

the rendered video with a Poisson's ratio of 0.05 as the reference and compared the PSNR of other rendered videos against it, shown in Table 3. A PSNR higher than 40 indicates indistinguishable differences.

Table 3: PSNR of rendered videos with varying Poisson's ratio. Results indicating varying Poisson's ratio has minimal effect on synthesized motions.

Scene / Poisson's ratio	0.1	0.2	0.3
Cord	57.05	54.94	53.62
Carnation	56.89	50.14	42.93

3 Effect of simulation hyperparameters

Within the same MPM solver, changing the grid size significantly affects numerical biases and dissipation, leading to drastically different synthesized motion. For instance, for the carnation scene, the PSNR between two videos simulated with a grid size of 48 and of 96 is only 26.5.

4 User study

We use Prolific[5] to recruit participants for the human preference evaluation. Participants span over multiple continents (primarily Europe, Africa, and North America). We only recruited users who are fluent in English. We use Google forms to present the survey. The survey is fully anonymized for both the participants and the host. We attach an example anonymous survey link in the footnote[6] for reference. Reviewer can enter any text such as "test" for Prolific ID.

An interesting result in the user-study at Table-1 of the main paper is that, under "Motion Realism", 86% of the users indicate the Alocasia outputs are more realistic than the actual captures. However, one would expect this to be around

[5] https://www.prolific.com/

[6] An example user study survey (comparing to PhysGaussian): https://forms.gle/CZfwxGHX2LaA7KxGA. Google forms require signing in to participate, but it does not record any participant's identity.

50% (when two videos are indistinguishable). One possible reason is that "Motion Realism" might be too abstract and ambiguous for a user study; Thus, we conducted an additional user study with a more specific prompt: "Compare the two videos below. One video shows real motion. Please select the real one." The results, shown in Table 4, exhibit a similar phenomenon. Thus, we tend to believe another potential explanation. For thin geometries, such as Alocasia leaves, the Material Point Method tends to produce lower-frequency and slower motions (can be observed in the video). Humans are poor at judging the naturalness of motion and may be biased towards these smoother and slower motions when rating "Motion Realism," as shown in prior studies [4,7].

Table 4: Percentage of humans who preferred our video over real captured videos in a 2AFC human study. We repeated the study with more specific prompts, involving 100 subjects. A synthetic video that is indistinguishable from a real one will achieve a percentage of 50%. Thus, a mean selection rate of 53.5% suggests our results are basically as realistic as real videos.

Alocasia	Carnation	Hat	Rose O	Rose W	Cord	Tulip	**Mean**
77%	56%	60%	69%	41%	47%	25%	**53.5%**

5 Website

We encourage the readers to explore videos in the attached website. Open the `index.html` to see the website.

6 Algorithm details

We present python-style pseudo-code for accelerating material point methods with K-Means downsampling in Algorithm 1.

Algorithm 1 Acclerate material point method with downsampling

```
# x, alpha, R, Sigma, c: the position, opacity, rotation, covariance and
    color of each Gaussian particle. x of shape [N, 3]
# num_drive_pts: int, top_k: int default as 8

clusters = KMeans(x, num_drive_pts)
drive_x = clusters.x # [M, 3]

# pre-compute the index of neighboor points
cdist = -1.0 * torch.cdist(x, drive_x) # [N, M]
_, top_k_index = torch.topk(cdist, top_k, -1)

# query initial velocity and material params, and simulate
drive_v = VeloField(drive_x)
drive_material = MaterialField(drive_x)
drive_x_simulated = Simulate(drive_x, drive_v, drive_material)

neighboor_drive_x = drive_x[top_k_index] # [N, top_k, 3]
neighboor_drive_x_simulated = drive_x_simulated[top_k_index]
# R: [N, 3, 3], t: [N, 3]
R_sim, t_sim = fitRigidTransform(drive_x, drive_x_simulated)

# apply transform to interpolate points
x = x + t_sim
R = R_sim @ R
# render
frame = Render(x, alpha, R @ Sigma @ R.T, c)
```

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017) 1
2. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 1
3. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) 1
4. Kobayashi, M., Motoyoshi, I.: Perceiving natural speed in natural movies. i-Perception **10**(4), 2041669519860544 (2019) 3
5. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11410–11420 (2022) 1
6. Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., Liu, Z.: Dreamgaussian4d: Generative 4d gaussian splatting. arXiv preprint arXiv:2312.17142 (2023) 1, 2
7. Stocker, A.A., Simoncelli, E.P.: Noise characteristics and prior expectations in human visual speed perception. Nature neuroscience **9**(4), 578–585 (2006) 3
8. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018) 1
9. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. arXiv preprint arXiv:2311.12198 (2023) 1, 2