DCSHARP: 3D Gaussian Splatting with Direction Cosine Spherical Harmonics and Shape-Aware Pruning

Ahmed Hasssan[†], Jian Meng, Yuanbo Xiangli and Jae-sun Seo
School of ECE, Cornell Tech, NY, USA [†] AMD, Inc.
{ah2288, jm2787, yx642, js3528}@cornell.edu ahmed.hasssan@amd.com

DCSHARP: 3D Gaussian Splatting with Direction Cosine Spherical Harmonics and Shape-Aware Pruning

Supplementary Material

1. Implementation and Evaluation

Implementation Details. We adopt the training strategy of conventional 3DGS [6] work to benchmark the performance of our proposed algorithms. We re-implement the CUDA forward and backward passes of the existing Gaussian rasterizer to incorporate the DCSH strategy. Furthermore, we adopt and modify the adaptive density control function of 3DGS to implement the DCSHARP algorithm. We would like to highlight that our DCSHARP pruning strategy performs run-time Gaussian pruning during the training process and involves no parallel computation or pseudo-rendering computation overhead. Following the 3DGS convention, we use 30k training iterations for each scene and evaluate the network performance after 7K and 30K iterations on test views. Finally, we ran all experiments on the NVIDIA A6000 GPU with an Intel CPU equipped with PyTorch 2.1.0 and CUDA version 12.1.

Evaluation Metrics. We evaluate the performance of our proposed DCSH and DCSHARP based on peak single-to-noise ratio (PSNR), structural similarity index measure (SSIM), learned perceptual image patch similarity (LPIPS), vector quantized codebook memory and rendered frames per second (FPS). However, we believe that LPIPS is comparatively a better metric to evaluate the rendered 3D scene visual quality since it measures the image perceptual quality close to human perception [1].

2. Algorithm

Initialize: For Training images $\mathcal{I} = \{I_i \in \mathbb{R}^3\}_{i=1}^{D}$ and their viewing direction $\mathcal{V} = \{\phi_i \in \mathbb{R}^{3\times4}\}_{i=1}^{D}$.
$\mathcal{G} = \{G_i \in \mathbb{R}^{3(\mathrm{SH}+1)^2+1+4+3}\}_{i=1}^{N} \leftarrow \mathrm{DCSH}(\mathcal{I}, \mathcal{V})$ # Initial 3D-GS: $\mathcal{G}$
while Iterations **do**
  $COV \leftarrow \mathrm{COMP}(\Sigma)$ # Calculate Covariance using Scale and Rotation
  $\mathcal{G}_i \leftarrow \exp\left(-\frac{1}{2}(r-\mu_g)^\top \Sigma^{-1}(r-\mu_g)\right)$ # Update initialized PCD with Covariance and Mean for each point
  $\mathcal{G} \leftarrow \mathrm{PRUNE}\left(\mathcal{G}_i, \begin{cases} 1, & \text{if } |\nabla P| \in \text{Top-}K\left(\{|\nabla P_i|\}_{i=1}^{N}\right) \\ 0, & \text{otherwise} \end{cases}\right)$ # Prune Least Significant Gaussians based on Plasticity (Scale-gradient) Score
end while
$\mathcal{G} \leftarrow$ Color Computation$(\mathcal{C})$ # Compute color using DCSH degree 3
while Iterations **do**
  $\mathcal{F}(\theta,\phi) \leftarrow \sum_{l=0}^{n-1}\sum_{m=-l}^{l} c_l Y_l^m(\theta,\phi)$
  $\mu_x = \sin\theta\cos\phi, \quad \mu_y = \sin\theta\sin\phi, \quad \mu_z = \cos\theta$
  $Y_l^m(u_x,u_y,u_z) \leftarrow \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\, P_l^m(u_z) \times e^{im\,\arctan\left(\frac{u_y}{u_x}\right)}$ # Direction cosine computation
  $Color \leftarrow \mathrm{ComputeColor}\left(C = \sum_j c_j \alpha_j \prod_{k=1}^{j-1}(1-\alpha_k) f_j\right)$
  $I_r \leftarrow \mathrm{Rasterize}(I(x,y), C(r,g,b))$ # Differentiable Tile Based Rasterizer
  $\nabla L \leftarrow \mathrm{LOSS}(I_r, I_{\mathrm{gt}})$
  $\mathcal{G}' \leftarrow \mathrm{ADAM}(\nabla L)$ # Backprop with gradient and Step Update
end while
Save Model: Save pruned and optimized model $\mathcal{G}'$ to disk.
Inference: Perform inference on pruned and optimized model $\mathcal{G}'$.

3. Additional Results

3.1. DCSH with Synthetic-NeRF Dataset

We experimented with the synthetic-NeRF dataset using our DCSH and DCSHARP algorithms. From Table 1, the proposed algorithms improve the overall performance with a 6.5× reduction in the number of Gaussians and a 2.3× reduction in training time.

Table 1. Performance comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA 3DGS baseline and 3DGS pruning works on Synthetic-NeRF dataset.

Scene	Metric	chair	drums	ficus	hotdog	lego	materials	mic	ship	Avg.
3DGS [6]	PSNR	35.83	26.15	34.87	37.72	35.78	30.00	35.36	30.80	33.32
	SSIM	0.9877	0.9548	0.9870	0.9854	0.9825	0.9604	0.9926	0.9062	0.9696
	LPIPS	0.01046	0.03657	0.01775	0.01977	0.0161	0.03671	0.00635	0.1058	0.03119
Our DCSH	PSNR	35.98	26.49	35.40	37.95	36.22	30.77	36.86	31.95	33.95
	SSIM	0.986	0.954	0.986	0.985	0.984	0.960	0.992	0.907	0.970
	LPIPS	0.006	0.031	0.006	0.010	0.009	0.016	0.003	0.063	0.018
	Train (m:ss)	12:00	11:00	8:01	7:01	10:05	11:00	12:00	9:03	10:00
	#Gaussians	419901	331408	192927	162923	302373	135288	159164	215364	239668
Comp Gaussian [7]	PSNR	34.91	26.18	35.44	37.38	35.48	29.97	35.81	31.51	33.33
	SSIM	0.986	0.953	0.987	0.984	0.981	0.958	0.991	0.905	0.968
	LPIPS	0.013	0.041	0.013	0.023	0.018	0.042	0.008	0.113	0.034
	Train (m:ss)	9:19	8:55	6:41	8:20	8:23	6:53	7:12	8:46	8:04
	#Gaussians	153,570	178,615	83,910	64,194	171,826	107,188	56,015	148,442	120,470
	FPS	512.10	427.56	706.71	719.85	402.15	638.01	674.34	282.72	545.43
LP-3DGS [10]	PSNR	35.419	26.102	35.354	37.728	35.769	29.883	36.337	31.375	33.496
	SSIM	0.9874	0.9358	0.9867	0.9846	0.9817	0.9566	0.9919	0.9034	0.966
	LPIPS	0.0111	0.03876	0.01217	0.02211	0.018	0.04323	0.00749	0.1151	0.0335
Our DCSHARP	PSNR	35.30	26.28	35.44	37.32	36.11	30.27	36.09	31.38	33.55
	SSIM	0.984	0.952	0.984	0.984	0.981	0.960	0.990	0.902	0.969
	LPIPS	0.009	0.035	0.008	0.015	0.015	0.026	0.006	0.077	0.023
	Train (m:ss)	4:03	4:53	4:14	4:39	4:10	4:33	4:16	4:55	4:23
	#Gaussians	56700	47863	18949	20053	58489	27310	18464	37286	35639
	FPS	540	447	786	755	502	668	684	312	587

Table 2. Quantitative comparison on head avatars. The proposed DCSH (without pruning) method outperforms prior works with improved quality, especially in the fine-texture regions (Figure 2).

Method	PSNR	SSIM	LPIPS
NHA [4]	20.29	0.883	0.145
INSTA [13]	26.42	0.924	0.080
PointAvatar [11]	27.84	0.913	0.067
SplattingAvatar [9] + SH + FLAME	28.19	0.931	0.063
SplattingAvatar [9] + DCSH + FLAME	28.32	0.933	0.060

3.2. DCSH with Zip-NeRF Dataset

Furthermore, we evaluate the performance of DCSH and DCSHARP on the Zip-NeRF dataset. For training purposes, we adopt the same strategy reported in section 1. We compare DCSH and DCSHARP performance with SoTA MERF [2] and SMERF [2] work on Zip-NeRF. From Tables 3, 4, 5, 6, and 7, the proposed DCSHARP improves

Table 3. PSNR comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA works on the Zip-NeRF dataset.

Method	BERLIN	NYC	ALAMEDA	LONDON
3DGS	26.60	26.41	23.52	25.45
MERF [2]	25.27	24.82	20.34	23.53
SMERF [2]	26.79	25.40	23.71	25.86
DCSH	28.11	26.95	23.57	26.52
DCSHARP	27.30	26.85	22.95	25.99

Table 4. SSIM comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA works on Zip-NeRF dataset.

Method	BERLIN	NYC	ALAMEDA	LONDON
3DGS	0.879	0.829	0.733	0.797
MERF [2]	0.840	0.765	0.646	0.737
SMERF [2]	0.852	0.771	0.701	0.789
DCSH	0.901	0.849	0.780	0.845
DCSHARP	0.880	0.847	0.740	0.801

Table 5. LPIPS comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA works on the Zip-NeRF dataset.

Method	BERLIN	NYC	ALAMEDA	LONDON
3DGS	0.335	0.343	0.407	0.392
MERF [2]	0.395	0.426	0.501	0.456
SMERF [2]	0.380	0.417	0.445	0.407
DCSH	0.131	0.117	0.20	0.19
DCSHARP	0.169	0.155	0.240	0.235

the overall performance across all the metrics with a 2.25× reduction in memory and 1.21× improvement in FPS.

Table 6. Memory (MB) comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA works on Zip-NeRF dataset. We adopt vector quantization from Light-Gaussian [3] to compute the 3DGS, DCSH and DCSHARP memory utilization for a fair comparison.

Method	BERLIN	NYC	ALAMEDA	LONDON
3DGS	215	270	250	168
MERF [2]	504	466	546	571
SMERF [2]	472	449	511	590
DCSH	215	270	250	168
DCSHARP	86.69	112.5	114.15	74.67

3.3. DCSH with Splatting Avatar

We extend the evaluation of the proposed method to photo-realistic human avatar [9, 12]. We adopt the model of SplattingAvatar [9] and replace the default spherical harmonics with the proposed direction cosine spherical harmonics. We evaluate the quality of DCSH against the head avatars with tracked FLAME meshes [8]. As shown in Table 2, the proposed DCSH outperforms the vanilla Splatting Avatar [9] with better quality. Qualitatively, DCSH accurately characterizes the fine-texture regions (e.g., hair) with improved quality compared to the default SH, as shown in Figure 2.

Table 7. FPS comparison of DCSH (without pruning) and DCSHARP (with pruning) with SoTA works on Zip-NeRF dataset. For a fair comparison, 3DGS*, DCSH* and DCSHARP* experiments are conducted on NVIDIA RTX A6000 GPU.

Method	BERLIN	NYC	ALAMEDA	LONDON
3DGS*	236	237	183	269
MERF [2]	184.8	469.5	269.5	348.4
SMERF [2]	220.3	480.8	358.4	363.6
DCSH*	236	238	184	269
DCSHARP*	310	301	309	328

3.4. DCSH with Complex Outdoor Scenes

Considering the complexity of new 3D rendering challenges, we evaluate of the proposed method on ULTRA challenge scenes [5]. The proposed method reports competitive quantitative and qualitative performance in comparison to the existing works. Qualitatively, DCSH accurately characterizes both varying altitude and dense image scenes in comparison to the 3DGS, as show in Figure 3.

Figure 1. Qualitative comparison of DCSH (without pruning) and DCSHARP (with pruning) with 3DGS [6] across different indoor and outdoor scenes.

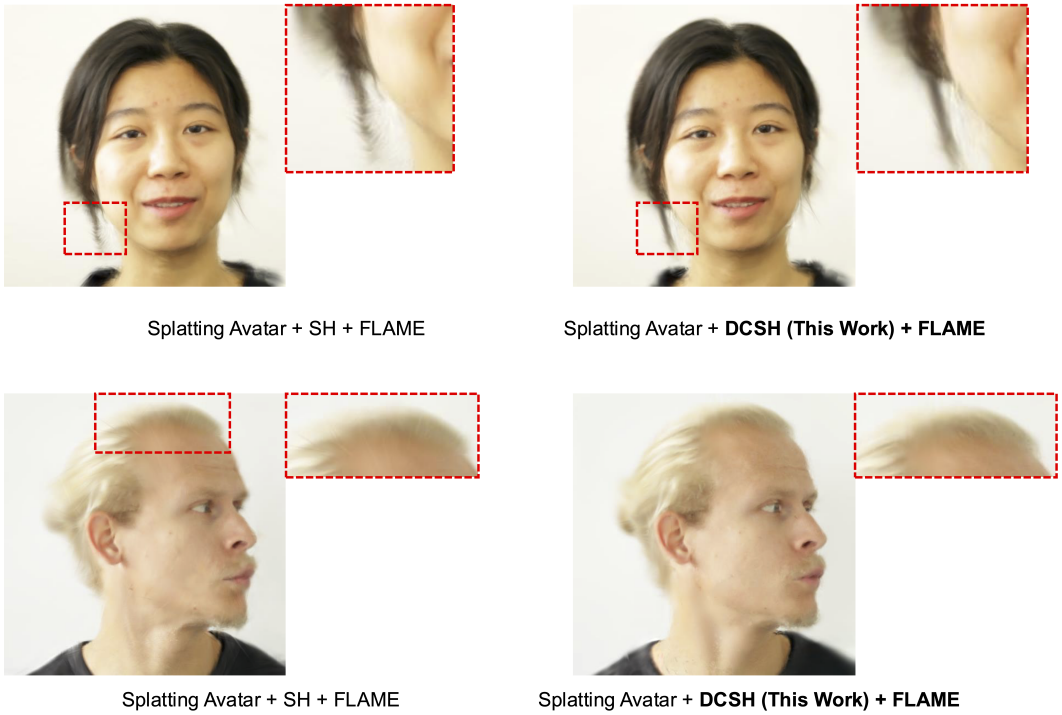

Figure 2. Comparison between conventional spherical harmonics and the proposed direction cosine spherical harmonics (DCSH) based on Splatting Avatar [9].

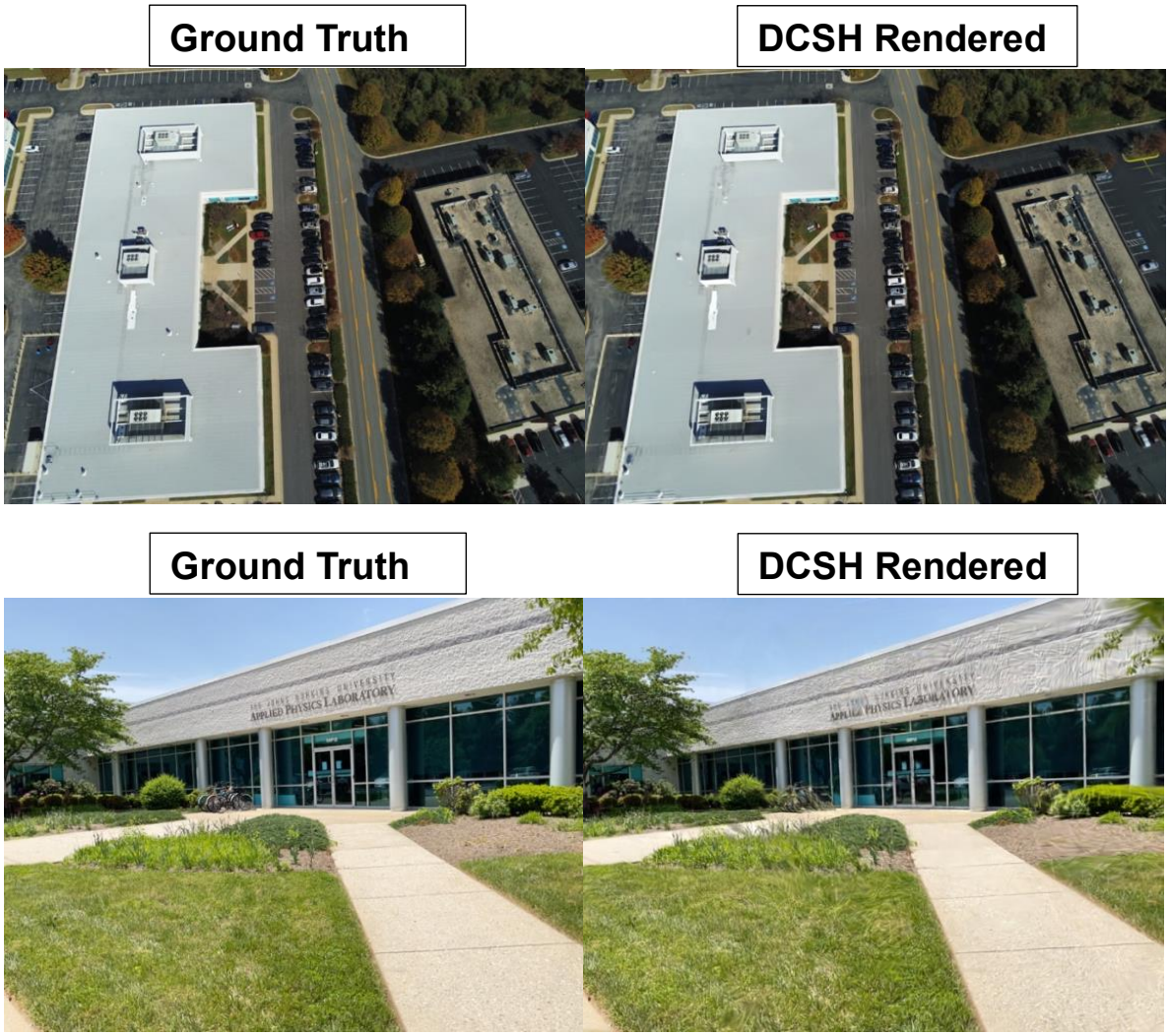

Figure 3. Qualitative comparison of proposed DCSH based rendering for image-density and varying altitude scenes of WACV-2025 ULTRA-challenge-dataset [5].

References

[1] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 1

[2] Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lučić, Richard Szeliski, and Jonathan T Barron. Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024. 1, 2

[3] Zhiwen Fan et al. LightGaussian: Unbounded 3D Gaussian compression with 15× reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*, 2023. 2

[4] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[5] Neil Joshi, Joshua Carney, Nathanael Kuo, Homer Li, Cheng Peng, and Myron Brown. Unconstrained large-scale 3d reconstruction and rendering across altitudes. *arXiv preprint arXiv:2505.00734*, 2025. 2, 5

[6] Bernhard Kerbl et al. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 3

[7] Joo Chan Lee et al. Compact 3D Gaussian representation for radiance field. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[8] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[9] Zhijing Shao et al. SplattingAvatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 4

[10] Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. Lp-3dgs: Learning to prune 3d gaussian splatting. *arXiv preprint arXiv:2405.18784*, 2024. 1

[11] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *IEEE/CVF conference on computer vision and pattern recognition*, 2023. 1

[12] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[13] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1