Supplementary Material

A. 3DGS Optimizations

A.1. Volume Guided Initialization

When using 3DGS, an initial set of 3D Gaussian kernels is first selected. These Gaussians are then removed, split or re-positioned, and the shape and appearance of the Gaussian kernels is optimized. Kerbl *et al*. [KKLD23] obtain the initial positions of the 3D Gaussians from the given images with structure from motion, or with random initialization where Gaussians are randomly positioned in the scene. For volume rendering, we randomly place Gaussians within the volume bounding box and set their initial color to grey. All other parameters are initialized as proposed by Kerbl *et al*. [KKLD23].

Since in Cinematic Anatomy the 3D object and presets are known, an interesting question is whether the optimization process can be accelerated by initially placing Gaussians at locations were they will end up anyway. Thus, we initially position one Gaussian at every non-empty voxel in a low resolution version of the volume, and set the Gaussians' initial colors and opacities via the transfer function. Regions that are under-sampled by the initial sampling will be nevertheless represented by Gaussians due to adaptive splitting and relocation during optimization.

In Fig. 1 we exemplarily compare the effectiveness of the different initialization schemes based on optimization convergence for one of our test data sets. An initialization with the Gaussians' positions and colors from a previous reconstruction is used as gold standard. As can be seen, while all initialization techniques reach the same level of fidelity, volume-guided initialization does so with less iteration steps. However, it is fair to say that in all of our experiments the performance improvements were overall not significant, so that we decided to use random initialization in all tests.

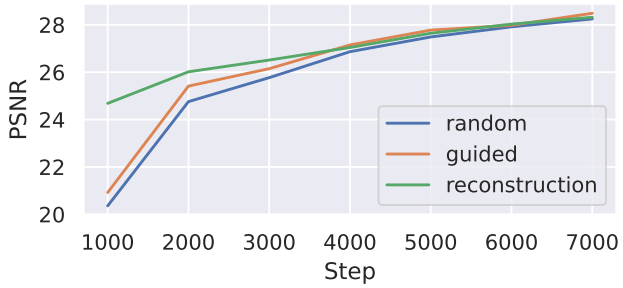

Figure 1: Test image reconstruction using different initialization schemes. Experiments were performed with *Body*.

A.2. Mip Splatting

Scenes rendered with 3DGS can show severe artifacts when novel camera perspectives diverge from those the 3D Gaussian representation was optimized for. Yu *et al*. [YCH*23] name the following two reasons for this behavior: Firstly, the 3D Gaussian representation exhibits frequencies that are too high to be faithfully reconstructed by the used sampling rate. Secondly, during splat-based rendering, a 2D dilation filter is applied that causes artefacts when zooming out and 2D splats become too small.

The problem is mitigated by introducing a 3D smoothing (i.e., low-pass) filter which constrains the size of the 3D Gaussians based on the maximal sampling frequency induced by the input views. A 2D Mip filter is applied in image space to avoid under-sampling. We observe that this extension to 3DGS significantly improves the fidelity of the reconstructed volumes for varying zoom levels.

B. Performance Statistics and Further Results

In Table 1 and Table 2, we provide detailed statistics using HR- and HQ-compression for all used presets. Additional qualitative results are shown in Fig. 4.

C. View Selection

In Fig. 2, we compare the convergence rate of BOS-based view selection to the approach of Kopanas and Drettakis [KD23], using preset 4 of *Body*. The final energy term adapted from Kopanas and Drettakis [KD23] is plotted after selecting a total of 32 cameras using a variable number of tested candidate views. Fig. 3 shows the time it takes the view selection algorithms to select the 32 camera poses for these two strategies. BOS is able to find better maxima than pure random sampling of the candidate camera poses as described by Kopanas and Drettakis, albeit at a slightly higher computation time for the same number of generated candidate poses. The tests were run on a system with an AMD Ryzen 9 3900X 12-core (24-thread) CPU and an NVIDIA GeForce RTX 3090 GPU and averaged over 8 different random seeds. The BOS algorithm is run on the CPU, while the computation of the transmittance as described in the main manuscript is performed on the GPU using ray marching of a downscaled version of the volume.

Scene	Preset	Resolution	Duration	Size	Points	Train Images	Test Images	SSIM	PSNR	PSNR (Alpha)
Brain	1	2k	108 Min	170 MB	5.1 M	87	12	0.64	20.73	32.76
	2	2k	88 Min	156 MB	5.0 M	87	12	0.81	25.78	35.86
Kidney	1	2k	50 Min	73 MB	2.7 M	91	13	0.81	23.93	27.70
	2	2k	49 Min	60 MB	2.2 M	87	12	0.87	27.89	32.85
Body	1	2k	58 Min	30 MB	1.0 M	87	12	0.89	28.93	31.54
	2	2k	43 Min	28 MB	1.0 M	87	12	0.87	25.38	29.94
	3	2k	43 Min	30 MB	1.0 M	87	12	0.88	27.65	30.40
	4	1k	20 Min	55 MB	1.9 M	224	32	0.86	28.68	25.41

Table 1: High Quality reconstruction details for all scenes and presets.

Scene	Preset	Resolution	Size	Gaussians	SSIM	PSNR	PSNR (Alpha)
Brain	1	2k	69	4.8 M	0.63	20.67	32.48
	2	2k	69	4.8 M	0.81	25.79	35.70
Kidney	1	2k	37	2.6 M	0.81	23.91	27.81
	2	2k	30	2.1 M	0.87	27.70	32.58
Body	1	2k	7	1.0 M	0.88	28.51	30.60
	2	2k	7	0.9 M	0.86	25.03	29.08
	3	2k	7	1.0 M	0.87	27.17	29.03
	4	1k	22	1.5 M	0.81	26.13	24.46

Table 2: High Range compression details for all scenes and presets

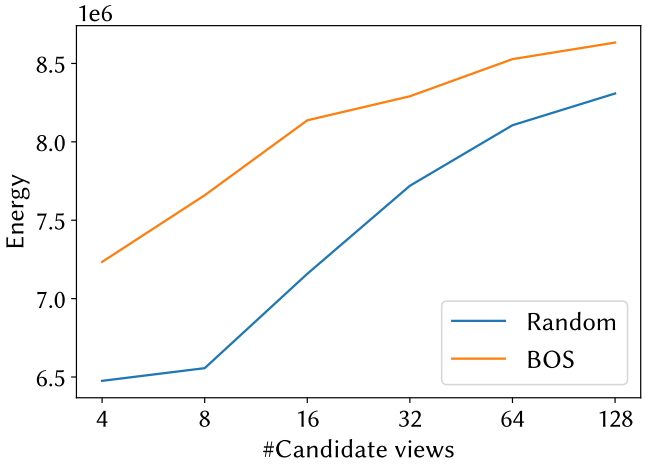

Figure 2: Achieved energy (the higher the better) when 32 camera poses are selected to generate test images of *Body*. On the *x*-axis, the batch size of candidate poses in each iteration is shown. On the *y*-axis, the energy, as adapted from Kopanas and Drettakis [KD23], is shown.

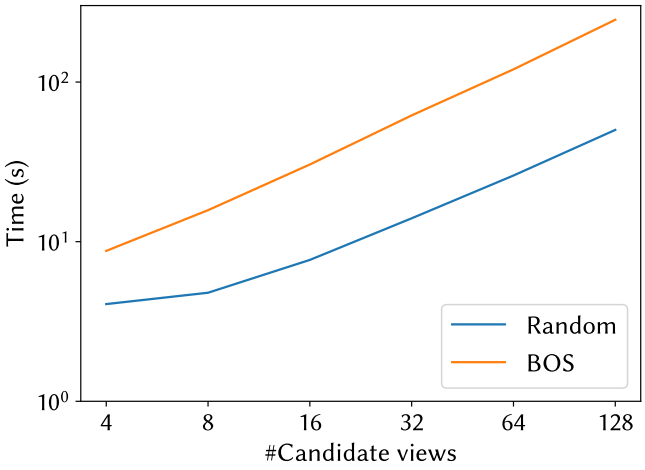

Figure 3: Run time for selecting 32 camera poses for *Body*. On the *x*-axis, the batch size of candidate poses in each iteration is shown. On the *y*-axis, the run time is shown.

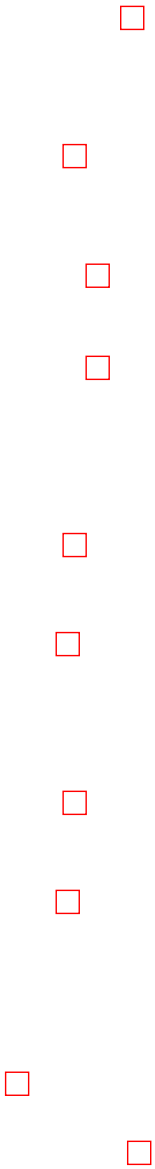

Figure 4: Quality comparison for HQ-compressed and HR-compressed Gaussian representations. All images are from the test set.