

# GBC-Splat: Generalizable Gaussian-Based Clothed Human Digitalization under Sparse RGB Cameras

## Supplementary Material

In this supplementary material, we present a visualization of our uniformly predefined and further upsampled Gaussians in Sec. 6, run-time analysis of each part in Sec. 7, and qualitative and quantitative results on real data in Sec. 8.

### 6. Visualization of Mesh-anchored Gaussians

We evaluate the effectiveness of our mesh-anchored Gaussian in this section.

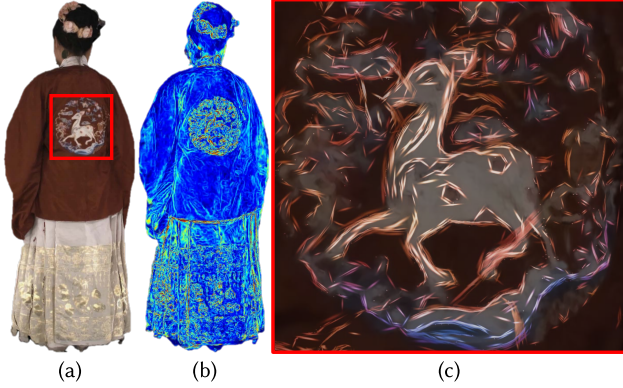


Figure 8. Visualization of uniform Gaussian Primitives. From left to right, (a) rendering result, (b) the ratio between the primary axis scale and the secondary axis scale, hot red stands for a large ratio while cold blue means a small ratio, (c) highlight long stringy Gaussian surfel along texture margin.

**Predefined Uniform Gaussian Primitives.** As mentioned in Sec. 3.2 of the main paper, some space related Gaussian components such as position and rotation are predefined once the geometry is reconstructed. Fig. 8 illustrates that the primary axis of uniformly distributed Gaussian surfel is orthogonal to texture gradient direction, and the ratio between the primary axis scale and the secondary axis scale depends on the difference of texture variance.

**Upsampled Gaussians via subdivision.** We leverage high-frequency input textures to subdivide Gaussians on each surfel, as mentioned in Sec. 3.3 of main paper. According to Eq. 13 of main paper, we first select visible points in different input views by maximizing the inner product of orientation of Gaussian surfel and surface normal, for example, green point means that this point is more visible in the front view as shown in Fig. 9(b), while blue and red parts mean the visible points in right and left view. Then we calculate the color variance on the input image to determine which patch should be subdivided, for example, the

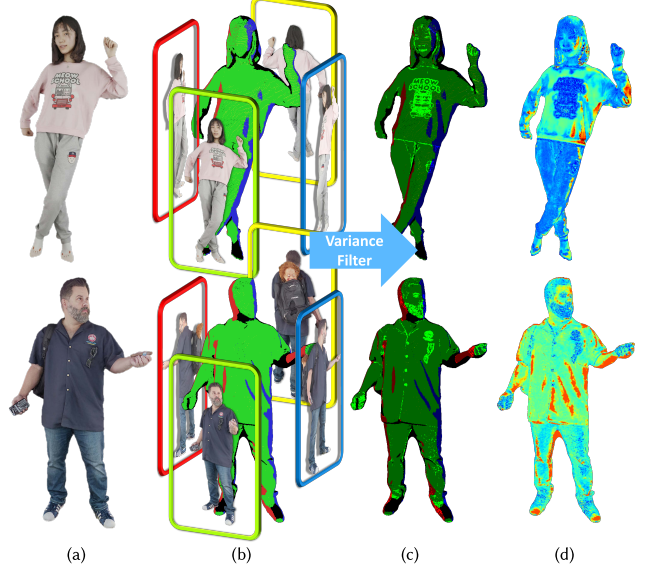


Figure 9. Subdivision process. From left to right, (a) rendering result, (b) view selection according to geometry normal, (c) highlight of the rich texture, and (d) scale distribution, cold color denotes small scale while hot one represents large scale.

Part	Time (ms)
Geo. image encoder $\mathcal{E}_{geo}$	24
Geo. feature vol. & 3D net $\mathcal{E}_{geo}^{3d}$	22
Geo. cross-attn & SDF regression	11
Coarse mesh extraction	5
RAFT decoding	71
Poisson fusion	50
Fine mesh extraction	7
Gaussian image encoder $\mathcal{E}_{app}$	68
Gaussian feature vol. & 3D net $\mathcal{E}_{app}^{3d}$	35
Gaussian components regression	16
Gaussian upsampling via subdivision	49
<b>Total</b>	<b>358</b>

Table 5. Run-time breakdown of our pipeline.

light part denotes the rich texture region in Fig. 9(c). After subdivision, the cold blue area in Fig. 9(d) denotes the well subdivided patches with small scales with respect to large-scale Gaussians in red area.

### 7. Run-time Breakdown

Our method consumes around 20G video memory during inference, which is able to fit in consumer-grade graphics

Method	DNA-Rendering			THuman			Twindom			2K2K		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
2DGS(4)*	16.01	0.654	0.443	20.01	0.813	0.275	18.35	0.710	0.322	20.74	0.804	0.285
2DGS(8)*	17.63	0.719	0.363	<b>27.50</b>	0.908	0.146	23.60	0.812	0.202	26.77	0.894	0.163
DoubleField(4)	20.36	0.827	0.274	19.82	0.866	0.300	19.51	0.777	0.374	19.65	0.839	0.265
DoubleFiled(8)	20.49	0.831	0.271	19.96	0.870	0.296	19.75	0.782	0.369	19.75	0.843	0.259
GPS-Gaussian(6) <sup>†</sup>	—	—	—	19.92	0.835	0.241	20.28	0.795	0.267	21.16	0.839	0.216
GPS-Gaussian(8)	21.28	0.840	0.209	24.62	0.902	0.207	23.95	<b>0.850</b>	0.222	25.80	0.911	0.162
GHG(4)	18.62	0.787	0.302	18.99	0.836	0.299	16.59	0.716	0.403	18.10	0.801	0.281
GHG(8)	18.55	0.784	0.307	19.05	0.834	0.304	16.61	0.714	0.409	18.11	0.798	0.285
Ours	22.39	0.851	0.202	25.51	0.920	0.108	24.10	0.845	0.138	25.38	0.909	0.136
Ours(Subdiv.)	<b>22.53</b>	<b>0.853</b>	<b>0.198</b>	25.73	<b>0.921</b>	<b>0.107</b>	<b>24.27</b>	0.848	<b>0.131</b>	<b>25.81</b>	<b>0.917</b>	<b>0.118</b>

Table 6. Quantitative comparison of novel-view rendering on real data DNA-Rendering and synthetic data THuman, Twindom and 2K2K with 2DGS, DoubleField, GPS-Gaussian and GHG.

cards. The breakdown of the running time for our pipeline is shown in Tab. 5, which is tested with an NVIDIA RTX 4090. We note that the majority of time is spent running fine explicit geometry reasoning and Gaussian parameter regression, while our upsampling technique takes relatively little time in Tab. 5.

## 8. More Results

As mentioned in Sec. 4.1 of the main paper, we prepare 150 synthetic scan data from THuman [80], Twindom [1], and 2K2K [7] datasets as validation set. To test the robustness of our method on real data, we additionally collect real-captured data of 200 characters from DNA-Rendering [9], which would retain imperfections of camera calibration, white balance, and foreground matting. For rendering result, we further compare with human-template based GHG [30] and NeRF-based DoubleField [58], besides optimization-based 2DGS [23] and depth-based GPS-Gaussian [84] in main paper. We note that GHG [30] requires a human template, SMPL-X [48], fitting and it is not trivial to fit SMPL-X under sparse views. We do our best to reproduce GHG on our collected real data, however, GHG is not able to generate a reasonable rendering of characters wearing loose clothing in Fig. 10(d). We also note that we evaluate DoubleField in a feed-forward way for fair comparison of other generalizable methods. Therefore the rendering results of DoubleField in Fig. 10(b) and 11(b) are blurry due to the lack of specific fine-tuning on each character. In Tab. 6 our method outperforms the other methods, especially on real data DNA-Rendering with the global perception oriented LPIPS metric. In addition, GPS-Gaussian is only evaluated with an 8-input-camera setting due to the camera sparsity of DNA-Rendering, and it can not avoid artifacts of marginal regions in Fig. 10(c) and Fig. 11(c) due to its partial geometry representation. When using sparse views as input, the optimization-based 2DGS could be overfitted to input views, thus renderings are degraded in novel views as shown in Fig. 10(a).



Figure 10. Qualitative comparison on real data DNA-Rendering. From left to right, (a) 2DGS, (b) DoubleField, (c) GPS-Gaussian, (d) GHG, (e) Ours, and (f) Ground Truth. All methods are under an 8-camera setting, while our method takes 4 pairs of stereo cameras.



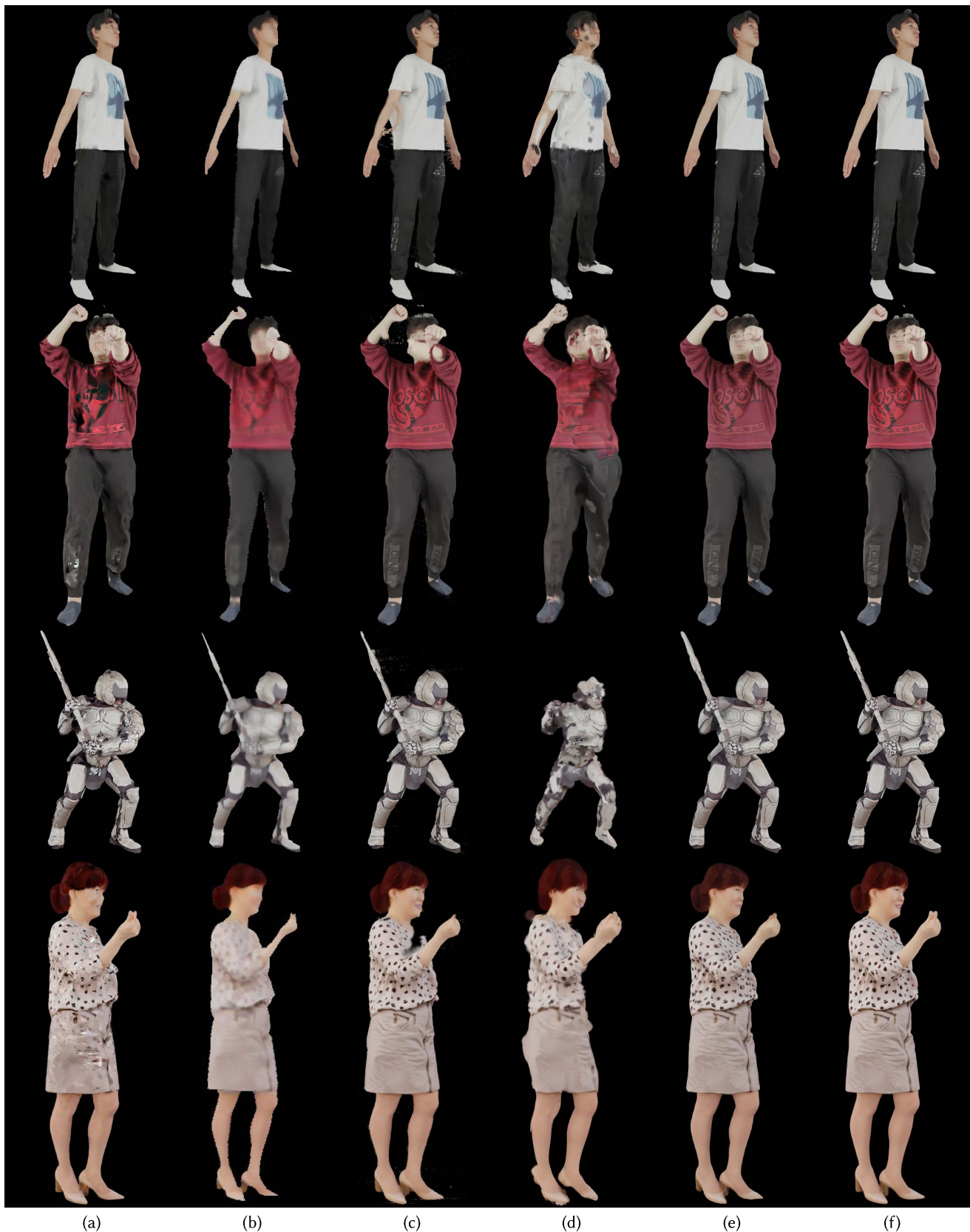


Figure 11. Qualitative comparison on synthetic data. From left to right, (a) 2DGS, (b) DoubleField, (c) GPS-Gaussian, (d) GHG, (e) Ours, and (f) Ground Truth. All methods are under an 8-camera setting, while our method takes 4 pairs of stereo cameras.