International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2025)
J. A. Jorge and N. Sakata (Editors)

Balancing Speed and Visual Fidelity of Dynamic Point Cloud Rendering in VR

Andre Muehlenbrock[1], Rene Weller[1], and Gabriel Zachmann[1]

[1]Computer Graphics and Virtual Reality Research Lab, University of Bremen, Germany

Abstract

Efficient rendering of dynamic point clouds from multiple RGB-D cameras is essential for a wide range of VR/AR applications. In this work, we introduce and leverage two key parameters in a mesh-based rendering approach and conduct a systematic study of their impact on the trade-off between rendering speed and perceptual quality. We show that both parameters enable substantial performance improvements while causing only negligible visual degradation. Across four GPU generations and multiple deployment scenarios, continuous dynamic point clouds from seven Microsoft Azure Kinects can achieve binocular rendering at triple-digit frame rates, even on mid-range GPUs. Our results provide practical guidelines for balancing visual fidelity and efficiency in real-time VR point cloud rendering, demonstrating that mesh-based approaches are a scalable and versatile solution for applications ranging from consumer headsets to large-scale projection systems.

CCS Concepts

• ***Computing methodologies*** → ***Rendering; Virtual reality;*** *Point-based models; Mesh geometry models;*

1. Introduction

Efficient rendering of dynamic multi-view point clouds is crucial for a wide range of applications, including VR telepresence [GCC*20]—particularly in medical contexts [YGE*21, RYP*21]—as well as performance capture [DKD*16, DDF*17] and the visualization or collision detection of reconstructed environments in VR [NIH*11]. Numerous approaches for real-time point cloud rendering have been explored, ranging from point-based techniques such as 3D Gaussian Splatting [KKLD23, HGSW24], to neural methods [CCR*23], mesh-based pipelines [MWZ24], and implicit surface representations such as Moving Least Squares or TSDFs [DKD*16, DDF*17, EWP*23], which can either be raycast directly or converted into explicit geometry via algorithms such as Marching Cubes. These methods vary significantly in design, complicating systematic comparisons of their real-time performance and practical suitability. Reported performance metrics are typically sparse, tied to specific implementations, and rarely address the requirements of multi-view XR rendering.

Although most techniques can be rendered in real time, many require an offline reconstruction or optimization stage that is not real-time capable. This limitation applies even to recent streamable 4D Gaussian Splatting approaches such as 3DGStream [SJL*24], HiCoM [GMW*24], and IGS [YPW*25], which require at least several seconds per frame for reconstruction before enabling real-time rendering. Reproducing highly optimized but closed-source systems such as Microsoft Holoportation [OERF*16], Fusion4D [DKD*16], or Motion2Fusion [DDF*17] is similarly impractical. An alternative particularly well-suited for low-latency capture with RGB-D cameras is BlendPCR [MWZ24], as it avoids any training or reconstruction step prior to rendering.

Virtual reality imposes particularly high demands on frame rate: maintaining 120 fps is essential to improve user performance and to prevent simulator sickness [WSZ*23]. Consequently, it is crucial to investigate strategies for optimizing the efficiency of point cloud rendering while preserving rendering quality. On high-end hardware such as the NVIDIA GeForce RTX 4090, techniques like BlendPCR [MWZ24] can achieve frame rates exceeding 120 fps with seven RGB-D cameras in typical XR applications. However, as VR headsets continue to increase in resolution, or when running on mid-range hardware, or when substantial additional geometry must be rendered, frame rates may fall significantly below 120 fps.

In this work, we extend BlendPCR by focusing on systematic performance optimization and evaluation. Specifically, we demonstrate that introducing and tuning two key parameters—the grid resolution of the individual meshes and the resolution of the screen-space rendering passes—can lead to significant performance improvements while preserving perceptual quality. We benchmark these optimizations across diverse deployment scenarios, including VR headsets, CAVE systems, and projection mapping environments. With our optimizations, BlendPCR enables real-time multi-view fusion of seven RGB-D streams at triple-digit frame rates for binocular rendering, even on mid-range GPUs, making it highly suitable for many VR/AR applications. Moreover, high-

DOI: 10.2312/egve.20251353

end consumer GPUs are sufficient to support demanding CAVE and projection-mapping scenarios in real time.

2. Method

Our implementation builds upon the publicly available BlendPCR source code [MWZ24]. The original approach generates one mesh per camera using the inherent topology of depth images, applies edge smoothing, and blends all rendered meshes in screen space to produce seamless transitions. For each fragment, the camera with the smallest distance and a viewing direction most antiparallel to the surface normal is primarily selected, while the blending region is kept as small as possible to minimize blurring. For high performance and compatibility, all processing steps are implemented within GPU shaders.

In this paper, we contribute the following optimizations in order to remove some significant bottlenecks:

1. Eliminating the geometry shader by moving per-triangle culling to the vertex shader via instanced rendering,
2. Uploading 16-bit depth images instead of precomputed point clouds to reduce bandwidth, and
3. Adjusting kernel parameters with no observable impact on visual quality.

Furthermore, we identified two key parameters that significantly influence rendering performance while having only a minor effect on perceived quality: (1) the grid resolution of the individually rendered meshes (Section 2.1) and (2) the rendering resolution (Section 2.2).

2.1. Parameter 1: Mesh Grid Resolution

The point cloud rendering technique BlendPCR builds on the idea of Turk et al. [TL94] to reconstruct a mesh directly from the grid structure of a depth image. Instead of being explicitly fused, the meshes from individual cameras are blended and fused in screen space, yielding significantly fewer artifacts in transition regions while avoiding the computational overhead of mesh fusion.

When reconstructing the individual meshes, BlendPCR utilizes the full resolution of the depth sensors, which is crucial for applications such as geometric correction in projection mapping, where millimeter precision is required [MPS*25]. However, for visualization tasks, this resolution is often excessive, since the level of detail can be enhanced through color textures. Prior volumetric methods typically operate with voxel sizes between 4 mm and 20 mm [DKD*16, EWP*23]. In contrast, depth sensors often provide depth images with a higher lateral sampling density; however, the effective spatial resolution is considerably lower, particularly along the axial direction due to sensor noise, offsets and smoothing by the camera itself.

To account for this, we subsample the vertices, e.g., using only every second or third pixel (Fig. 3) during the individual screen passes, while retaining the original texture resolution, edge distances, normals, and quality passes for the blending stage. This substantially reduces the vertex count of the individual meshes without noticeably degrading visual fidelity (cf. Figures 1 and 2). We explored the possibility of applying dynamic tessellation along edges

(a) *Full (2.58 M)* **(b)** *Half (0.65 M)* **(c)** *Third (0.29 M)*

Figure 1: *Comparison of grid resolutions on the CWIPC-SXR S13 Card Trick scene. Despite a drastic reduction in the number of vertices, hardly any visual difference is noticeable. The top and middle rows show cropped scenes focusing on the people, while the bottom row shows the complete scene.*

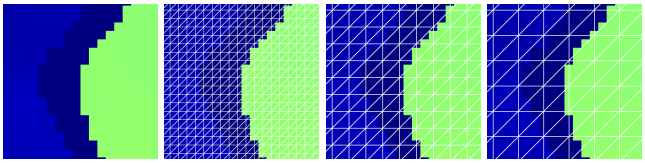
(a) *Full (2.58 M)* **(b)** *Half (0.65 M)* **(c)** *Third (0.29 M)*

Figure 2: *Comparison of grid resolutions on the CWIPC-SXR S21 YMCA R14 scene. Visual differences are only slightly perceptible at the boundaries and in noisy regions.*

Table 1: *Scene clipped to a single full-body human. Average frame times (in ms) when rendering multiple views per frame at the full native display resolution of different device specifications. Frame times were measured as frame-to-frame deltas within the overall application with vertical synchronization (v-sync) disabled. The results include both screen-rendering passes and camera passes from seven simultaneously operated Microsoft Azure Kinect cameras at 640 × 576 and 30 Hz. The RTX 2070 Super (M) refers to a mobile GPU tested in a laptop.*

	Resolution	Views	RTX 4090			RTX 3090			RTX 2060 Super			RTX 2070 Super (M)		
	(per view)		Third	Half	Full	Third	Half	Full	Third	Half	Full	Third	Half	Full
HTC Vive Pro	1440×1600	2	2.1	2.7	5.2	2.4	3.0	6.6	3.5	4.0	7.1	6.2	6.7	10.5
Meta Quest 3	2064×2208	2	2.8	3.1	5.7	3.5	4.2	7.6	5.8	6.3	9.3	8.8	9.4	13.7
HTC Vive Pro 2	2448×2448	2	3.5	3.6	6.5	4.3	4.8	8.3	7.2	7.8	10.5	10.9	11.4	16.4
Apple Vision Pro	3660×3200	2	8.2	8.6	10.6	7.8	8.3	11.5	13.7	14.5	17.2	22.2	23.3	27.0
HTC Vive Pro ($\frac{1}{2}$)	720×800	2	1.6	2.1	4.6	1.7	2.4	5.8	1.9	2.4	5.6	4.6	5.2	8.5
Meta Quest 3 ($\frac{1}{2}$)	1032×1104	2	1.8	2.3	4.7	1.9	2.6	6.1	2.5	2.9	6.1	5.1	5.9	9.8
HTC Vive Pro 2 ($\frac{1}{2}$)	1224×1224	2	1.9	2.3	5.0	2.2	2.8	6.4	2.7	3.3	6.4	5.0	5.8	9.9
Apple Vision Pro ($\frac{1}{2}$)	1830×1600	2	2.3	2.8	5.4	2.7	3.4	6.9	4.1	4.7	7.8	6.8	7.4	11.6
UHD Screen	3840×2160	1	2.8	3.1	4.3	3.3	3.5	5.1	5.4	5.7	7.2	8.1	8.8	10.9
UHD Screen	3840×2160	3	6.7	7.4	11.0	8.5	9.4	14.3	15.4	16.1	20.4	20.0	21.3	27.0
UHD Screen	3840×2160	6	12.3	13.5	20.8	16.4	17.9	27.8	29.4	31.2	38.5	35.7	38.5	47.6
UHD Screen	3840×2160	10	20.0	21.7	33.3	26.3	29.4	43.5	47.6	50.0	62.5	52.6	55.6	71.4
FHD Screen	1920×1080	1	1.5	1.7	3.0	1.5	1.9	3.6	2.0	2.3	3.8	5.0	4.7	6.6
FHD Screen	1920×1080	3	2.7	3.4	7.1	3.2	4.3	9.5	5.0	5.9	10.4	7.6	8.5	15.6
FHD Screen	1920×1080	6	4.5	5.8	13.3	5.9	7.9	18.2	9.5	11.1	20.0	13.0	15.4	27.8
FHD Screen	1920×1080	10	6.8	8.9	21.7	9.5	12.7	29.4	15.4	18.2	33.3	19.6	23.3	41.7

Color legend: $\leq$ 8.3 ms (green), $\leq$ 11.1 ms (green-yellow), $\leq$ 33.3 ms (yellow), $>$ 33.3 ms (orange).



(a) *Depth Image* **(b)** *Full (2.58 M)* **(c)** *Half (0.65 M)* **(d)** *Third (0.29 M)*

Figure 3: *Illustration of the grid resolution for the individual meshes. The seven Microsoft Azure Kinects produce 2.58 M vertices at full resolution. Halving the resolution results in 0.65 M vertices, while reducing to one-third yields 0.29 M vertices.*

to further reduce visual differences. However, even inserting a trivial tessellation or geometry shader into the pipeline—simply passing the data through—would already negate performance benefits.

2.2. Parameter 2: Rendering Resolution

High display resolutions in VR and AR setups can heavily impact performance due to the increased fragment shading load. In contrast to games with high-fidelity textures, where downsampling visibly reduces image quality, point clouds from low-resolution sensors are often oversampled. As an example, consider a depth sensor capturing 640 horizontal pixels over a 75° horizontal FoV at a distance of 2 m. When viewed in VR from the same distance using an HTC Vive Pro 2 (2448 horizontal pixels, 120° FoV), this corresponds to approximately 8.5 rendered pixels per degree, or about 20 rendered pixels per input pixel along the horizontal flow. As the virtual observer moves closer to scene objects, this oversampling increases further.

A straightforward way to exploit this when rendering the point cloud into the virtual environment is to render the point cloud in a separate, lower-resolution FBO, using the same model-to-world and view matrices, and then blit it into the high-resolution buffer of the virtual world. We investigate this as a second option to improve performance with only minor perceptual differences.

3. Evaluation

In this section, we assess the impact of the two parameters introduced in the previous section using the Structural Similarity Index (SSIM) relative to the original image (Section 3.1) and present a comprehensive performance evaluation across a wide range of VR and AR hardware configurations (Section 3.2).

3.1. Quality Comparison

To evaluate the influence of mesh grid resolution (Parameter 1) on rendering quality, we computed the SSIM score across 660 renderings of the CWIPC-SXR dataset [RAJ*21], comparing different grid resolutions against the original full-resolution rendering (see Table 2). Overall, the SSIM values remain consistently high across all renderings. Even using only every third vertex, perceptual quality remains very close to the full mesh (SSIM > 0.97, PSNR $\approx$ 31 dB). The most pronounced deviations occur in distant background regions (e.g., walls), where lower grid resolutions lead to larger holes, whereas differences around human subjects remain negligible. This effect is caused by the increasing grid-cell size of individual meshes with greater RGB-D camera distance. Since VR/AR applications commonly target human subject segmentation, and miss-

(a) *UHD* **(b)** *FHD* **(c)** *UHD* **(d)** *FHD*

Figure 4: *Visual comparison between native UHD resolution (a,c) and half resolution, linearly upsampled to UHD (b,d), for an object with an oversampled texture. Visible differences are minimal and primarily confined to object boundaries. Pairs (a) and (b) exhibit an SSIM of 0.975, while (c) and (d) exhibit an SSIM of 0.978.*

Table 2: *Quality scores comparing the half and third grid resolutions to the full grid resolution. Reducing the grid resolution has only a minor effect on perceptual fidelity. A total of $n = 660$ comparison images from two scenes were used, clipped and cropped to include only the human subject. Values in parentheses indicate standard deviations.*

	SSIM	RMSE	PSNR (dB)
Full	1.000	0.000	∞
Half	0.982 (0.027)	5.8 (4.0)	33.8 (3.7)
Third	0.976 (0.016)	7.0 (2.2)	31.5 (2.0)

ing regions can be handled with hole-filling or inpainting methods, this issue has only minor impact.

In addition, Figure 4 illustrates the effect of rendering resolution (Parameter 2). Due to texture oversampling, rendering at a lower resolution followed by linear upscaling results in only minor visible differences. This effect is distance-dependent, but oversampling typically occurs at common viewing distances.

3.2. Performance Measurements

For performance evaluation, we benchmarked specifications of a range of commonly used VR headsets, namely (a) HTC Vive Pro, (b) Meta Quest 3, (c) HTC Vive Pro 2, and (d) Apple Vision Pro.

In addition to immersive head-mounted displays, point cloud rendering is also relevant for dynamic projection-based setups, e.g. [MPS*25]. In these setups, the point cloud rendering implicitly performs geometric correction, undistorting projections for specific viewpoints. To evaluate this, we further benchmarked simultaneous multi-view rendering with 1, 3, 6, and 10 screens, each tested at both Full HD and UHD resolution. All benchmarks were conducted using seven synchronized Microsoft Azure Kinects (640×576 at 30 Hz) as input and tested on four NVIDIA GPUs (RTX 4090, RTX 3090, RTX 2060 Super, and RTX 2070 Super (Mobile)). For each configuration, we rendered a 5000-frame trajectory circling the human subject in the CWIPC-SXR dataset [RAJ*21] and recorded the average frame time. Results are reported for both a cropped scenario containing only a single full-body human (see Table 1 and Figure 5), and the full captured scene (see Appendix, Table 3).

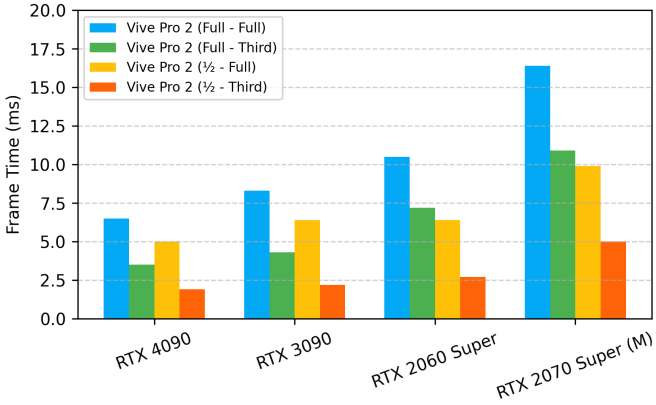


Figure 5: *Performance impact of reduced mesh and framebuffer resolution in an HTC Vive Pro 2 full-body human rendering.*

4. Conclusions

We presented performance optimizations and a systematic evaluation of BlendPCR for low-latency rendering of dynamic multi-view point clouds. In addition to lossless optimizations, the following near-lossless adaptation enable substantial performance gains:

1. Finding the right frame buffer resolution: When using high-resolution VR headsets, the RGB-D camera textures are frequently oversampled beyond necessity, as these textures inherently lack the resolution that modern VR headsets could provide, e.g., for recognizing text and fine details. In such cases, utilizing the full framebuffer resolution for rendering point clouds would not improve the image quality; consequently, reducing frame buffer resolution leads to substantial performance improvements while not degrading image quality.
2. Mesh grid resolution: For human-centered recordings, the mesh resolution can be decreased without noticeable degradation in visual quality by retaining the full-resolution color textures. Since depth maps—limited by sensor noise and preprocessing—do not fully utilize their nominal spatial precision, this trade-off yields significant computational benefits. For full-scene reconstructions, however, slightly enlarged geometric holes may occur and should be assessed on a case-by-case basis.

With these optimizations, performance improvements of up to 3–5× over the original BlendPCR implementation can be achieved, enabling real-time operation even on mid-range GPUs. On high-end GPUs, depending on configuration and hardware, frame rates of up to 600 fps can be achieved for humans captured simultaneously with seven RGB-D cameras, without requiring any additional reconstruction steps. Source code is available†.

Future work will aim to further reduce residual rendering artifacts—such as minor holes at lower grid resolutions—and to integrate BlendPCR natively into established game engines.

Acknowledgements

This work was partially supported by BMBF grant 16SV9239.

† https://www.github.com/muehlenb/blendpcr

References

[CCR*23] CHANG J.-H. R., CHEN W.-Y., RANJAN A., YI K. M., TUZEL O.: Pointersect: Neural rendering with cloud-ray intersection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2023). 1

[DDF*17] DOU M., DAVIDSON P., FANELLO S. R., KHAMIS S., KOWDLE A., RHEMANN C., TANKOVICH V., IZADI S.: Motion2fusion: real-time volumetric performance capture. *ACM Trans. Graph. 36*, 6 (nov 2017). doi:10.1145/3130800.3130801. 1

[DKD*16] DOU M., KHAMIS S., DEGTYAREV Y., DAVIDSON P., FANELLO S. R., KOWDLE A., ESCOLANO S. O., RHEMANN C., KIM D., TAYLOR J., KOHLI P., TANKOVICH V., IZADI S.: Fusion4d: real-time performance capture of challenging scenes. *ACM Trans. Graph. 35*, 4 (jul 2016). doi:10.1145/2897824.2925969. 1, 2

[EWP*23] ECK U., WECHNER M., PANKRATZ F., YU K., LAZAROVICI M., NAVAB N.: Real-time 3d reconstruction pipeline for room-scale, immersive, medical teleconsultation. *Applied Sciences 13* (09 2023), 10199. doi:10.3390/app131810199. 1, 2

[GCC*20] GAMELIN G., CHELLALI A., CHEIKH S., RICCA A., DUMAS C., OTMANE S.: Point-cloud avatars to improve spatial communication in immersive collaborative virtual environments. *Personal Ubiquitous Comput. 25*, 3 (jul 2020), 467–484. doi:10.1007/s00779-020-01431-1. 1

[GMW*24] GAO Q., MENG J., WEN C., CHEN J., ZHANG J.: Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)* (2024). 1

[HGSW24] HU Y., GONG R., SUN Q., WANG Y.: Low latency point cloud rendering with learned splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2024), pp. 5752–5761. 1

[KKLD23] KERBL B., KOPANAS G., LEIMKUEHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592433. 1

[MPS*25] MUEHLENBROCK A., PURGIN Y., STEINKE N., USLAR V., WEYHE D., WELLER R., ZACHMANN G.: Shadow-free projection with blur mitigation on dynamic, deformable surfaces. In *Proceedings of the 31st ACM Symposium on Virtual Reality Software and Technology (VRST '25)* (New York, NY, USA, 2025), VRST '25, Association for Computing Machinery, p. to appear. doi:10.1145/3756884.3766018. 2, 4

[MWZ24] MÜHLENBROCK A., WELLER R., ZACHMANN G.: BlendPCR: Seamless and Efficient Rendering of Dynamic Point Clouds captured by Multiple RGB-D Cameras. In *ICAT-EGVE 2024 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments* (2024), Hasegawa S., Sakata N., Sundstedt V., (Eds.), The Eurographics Association. doi:10.2312/egve.20241366. 1, 2

[NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality* (2011), pp. 127–136. doi:10.1109/ISMAR.2011.6092378. 1

[OERF*16] ORTS-ESCOLANO S., RHEMANN C., FANELLO S., CHANG W., KOWDLE A., DEGTYAREV Y., KIM D., DAVIDSON P. L., KHAMIS S., DOU M., TANKOVICH V., LOOP C., CAI Q., CHOU P. A., MENNICKEN S., VALENTIN J., PRADEEP V., WANG S., KANG S. B., KOHLI P., LUTCHYN Y., KESKIN C., IZADI S.: Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (New York, NY, USA, 2016), UIST '16, Association for Computing Machinery, p. 741–754. doi:10.1145/2984511.2984517. 1

[RAJ*21] REIMAT I., ALEXIOU E., JANSEN J., VIOLA I., SUBRAMANYAM S., CESAR P.: Cwipc-sxr: Point cloud dynamic human dataset for social xr. In *Proceedings of the 12th ACM Multimedia Systems Conference* (New York, NY, USA, 2021), MMSys '21, Association for Computing Machinery, p. 300–306. doi:10.1145/3458305.3478452. 3, 4

[RYP*21] ROTH D., YU K., PANKRATZ F., GORBACHEV G., KELLER A., LAZAROVICI M., WILHELM D., WEIDERT S., NAVAB N., ECK U.: Real-time mixed reality teleconsultation for intensive care units in pandemic situations. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (Los Alamitos, CA, USA, apr 2021), IEEE Computer Society, pp. 693–694. doi:10.1109/VRW52623.2021.00229. 1

[SJL*24] SUN J., JIAO H., LI G., ZHANG Z., ZHAO L., XING W.: 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2024), pp. 20675–20685. 1

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, Association for Computing Machinery, p. 311–318. doi:10.1145/192161.192241. 2

[WSZ*23] WANG J., SHI R., ZHENG W., XIE W., KAO D., LIANG H.-N.: Effect of frame rate on user experience, performance, and simulator sickness in virtual reality. *IEEE Trans Vis Comput Graph 29*, 5 (Mar. 2023), 2478–2488. 1

[YGE*21] YU K., GORBACHEV G., ECK U., PANKRATZ F., NAVAB N., ROTH D.: Avatars for teleconsultation: Effects of avatar embodiment techniques on user perception in 3d asymmetric telepresence. *IEEE Transactions on Visualization and Computer Graphics 27*, 11 (2021), 4129–4139. doi:10.1109/TVCG.2021.3106480. 1

[YPW*25] YAN J., PENG R., WANG Z., TANG L., YANG J., LIANG J., WU J., WANG R.: Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting, 2025. arXiv:2503.16979. 1

Table 3: *Full scene comprising the human subject, surrounding walls, and floor. Average frame times (in ms) when rendering multiple views per frame at the full native display resolution of different device specifications. Frame times were measured as frame-to-frame deltas within the overall application with vertical synchronization (v-sync) disabled. The results include both screen-rendering passes and camera passes from seven simultaneously operated Microsoft Azure Kinect cameras at 640 × 576 and 30 Hz. The RTX 2070 Super (M) refers to a mobile GPU tested in a laptop.*

	Resolution	Views	RTX 4090			RTX 3090			RTX 2060 Super			RTX 2070 Super (M)		
	(per view)		Third	Half	Full	Third	Half	Full	Third	Half	Full	Third	Half	Full
HTC Vive Pro	1440×1600	2	2.5	2.6	5.2	3.2	3.7	7.1	6.0	6.7	10.0	9.9	10.8	15.4
Meta Quest 3	2064×2208	2	3.6	4.1	6.3	5.1	5.6	8.7	10.5	11.2	14.9	16.1	16.9	21.7
HTC Vive Pro 2	2448×2448	2	5.1	5.3	9.6	6.5	6.8	9.7	13.3	13.9	17.9	21.3	22.2	27.0
Apple Vision Pro	3660×3200	2	11.5	11.9	13.7	11.6	12.3	15.2	27.8	29.4	32.3	43.5	43.5	47.6
HTC Vive Pro ($\frac{1}{2}$)	720×800	2	1.7	2.2	4.8	1.9	2.5	6.3	2.7	3.3	6.8	5.8	6.8	11.0
Meta Quest 3 ($\frac{1}{2}$)	1032×1104	2	1.9	2.4	4.9	2.3	2.9	6.6	3.7	4.5	7.9	7.1	7.6	12.5
HTC Vive Pro 2 ($\frac{1}{2}$)	1224×1224	2	2.1	2.6	5.1	2.6	3.2	6.9	4.5	5.2	8.6	7.8	8.8	13.7
Apple Vision Pro ($\frac{1}{2}$)	1830×1600	2	2.9	3.1	5.6	3.8	4.2	7.6	7.4	8.1	11.5	11.1	11.8	16.7
UHD Screen	3840×2160	1	3.9	4.0	4.9	4.8	5.1	6.5	10.1	10.4	12.3	14.9	15.9	17.5
UHD Screen	3840×2160	3	9.8	10.2	13.2	13.0	13.9	18.2	27.8	29.4	33.3	35.7	37.0	41.7
UHD Screen	3840×2160	6	18.5	19.6	24.4	25.0	27.0	34.5	50.0	52.6	58.8	58.8	62.5	71.4
UHD Screen	3840×2160	10	30.3	32.3	40.0	40.0	43.5	55.6	76.9	76.9	90.9	90.9	100.0	111.1
FHD Screen	1920×1080	1	1.7	2.0	3.2	1.9	2.2	3.9	3.2	3.6	5.3	6.1	6.5	8.7
FHD Screen	1920×1080	3	3.2	3.8	7.6	4.4	5.2	10.4	8.6	9.5	14.7	12.5	14.3	20.8
FHD Screen	1920×1080	6	5.5	6.6	14.1	8.2	9.8	20.0	16.4	18.2	28.6	22.7	25.0	37.0
FHD Screen	1920×1080	10	8.5	10.2	22.7	13.2	16.1	32.3	25.6	28.6	43.5	34.5	37.0	52.6

Color legend: $\leq$ 8.3 ms (green), $\leq$ 11.1 ms (green-yellow), $\leq$ 33.3 ms (yellow), $>$ 33.3 ms (orange).