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

Andre Mühlenbrock*, Rene Weller and Gabriel Zachmann

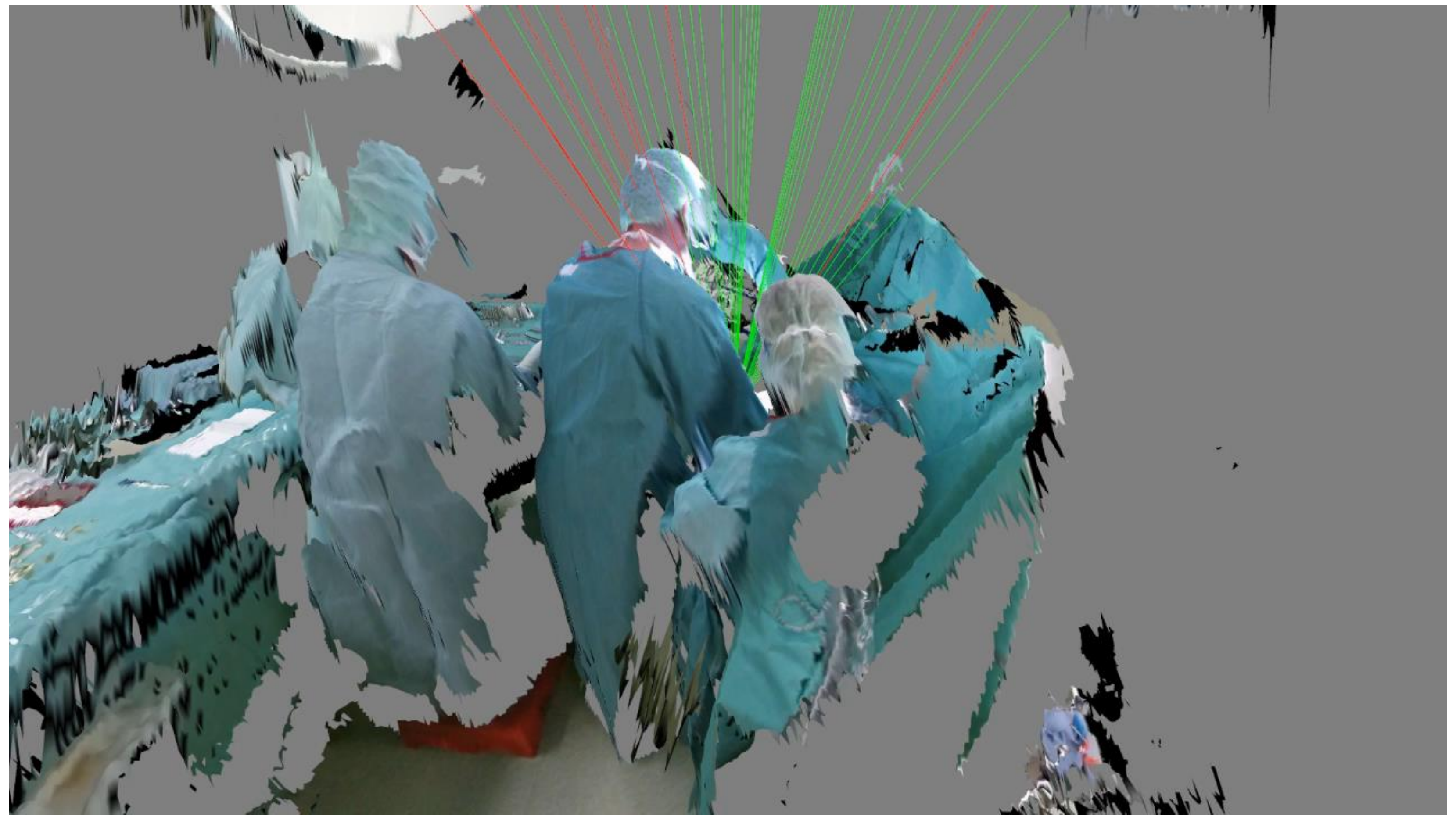
Computer Graphics and Virtual Reality Research Lab (CGVR), University of Bremen

ICAT-EGVE 2025, 3-5 December, Karlskrona, Sweden

Motivation

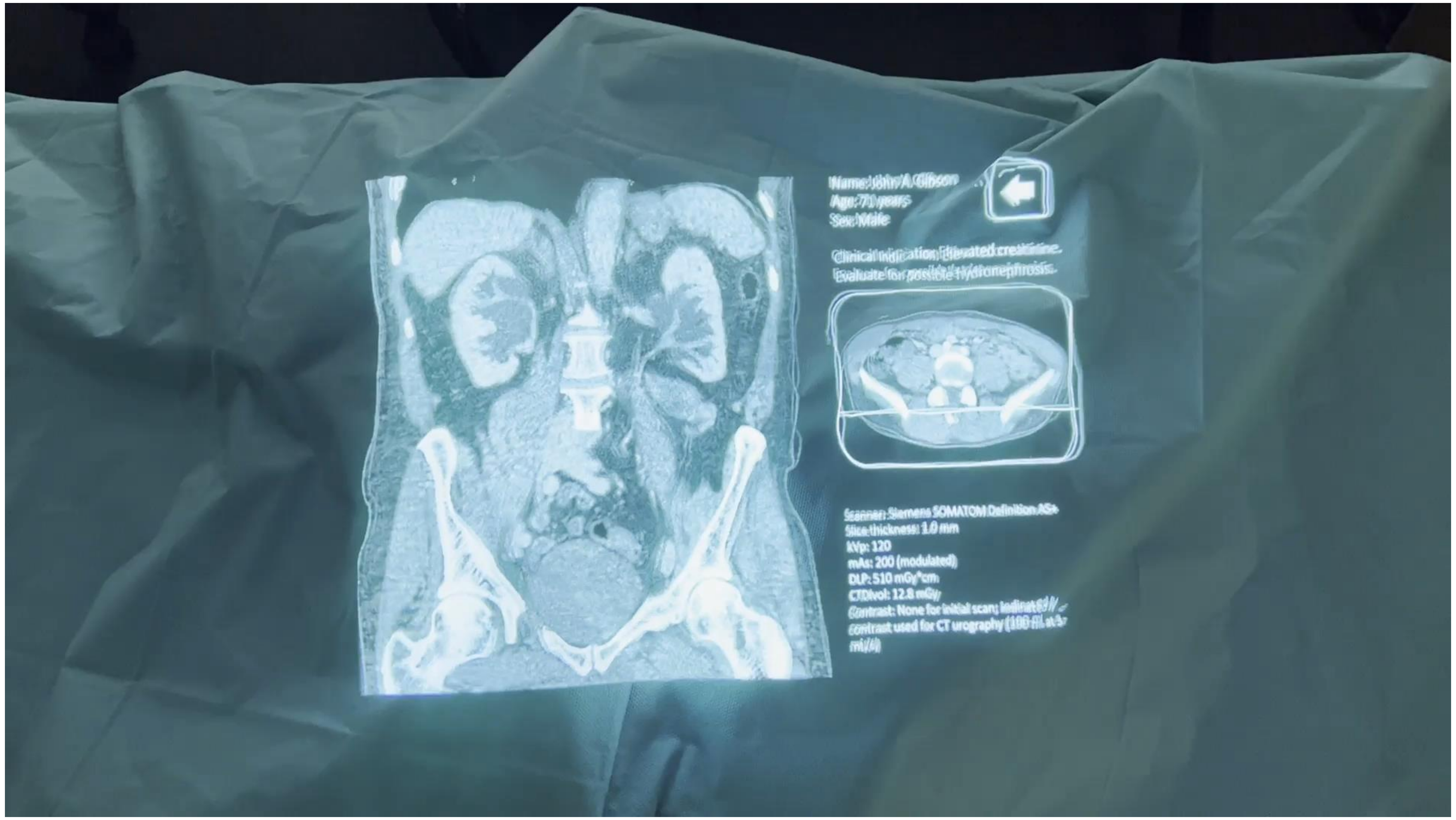
- Live Human Avatars in VR
- Live 3D Scene Reconstruction
  - Live concerts, presentations, ...
  - VR-Telemedicine
- Live Geometric Correction with Multi-Projector-Setups

Yu et al. [TVCG, 2021]

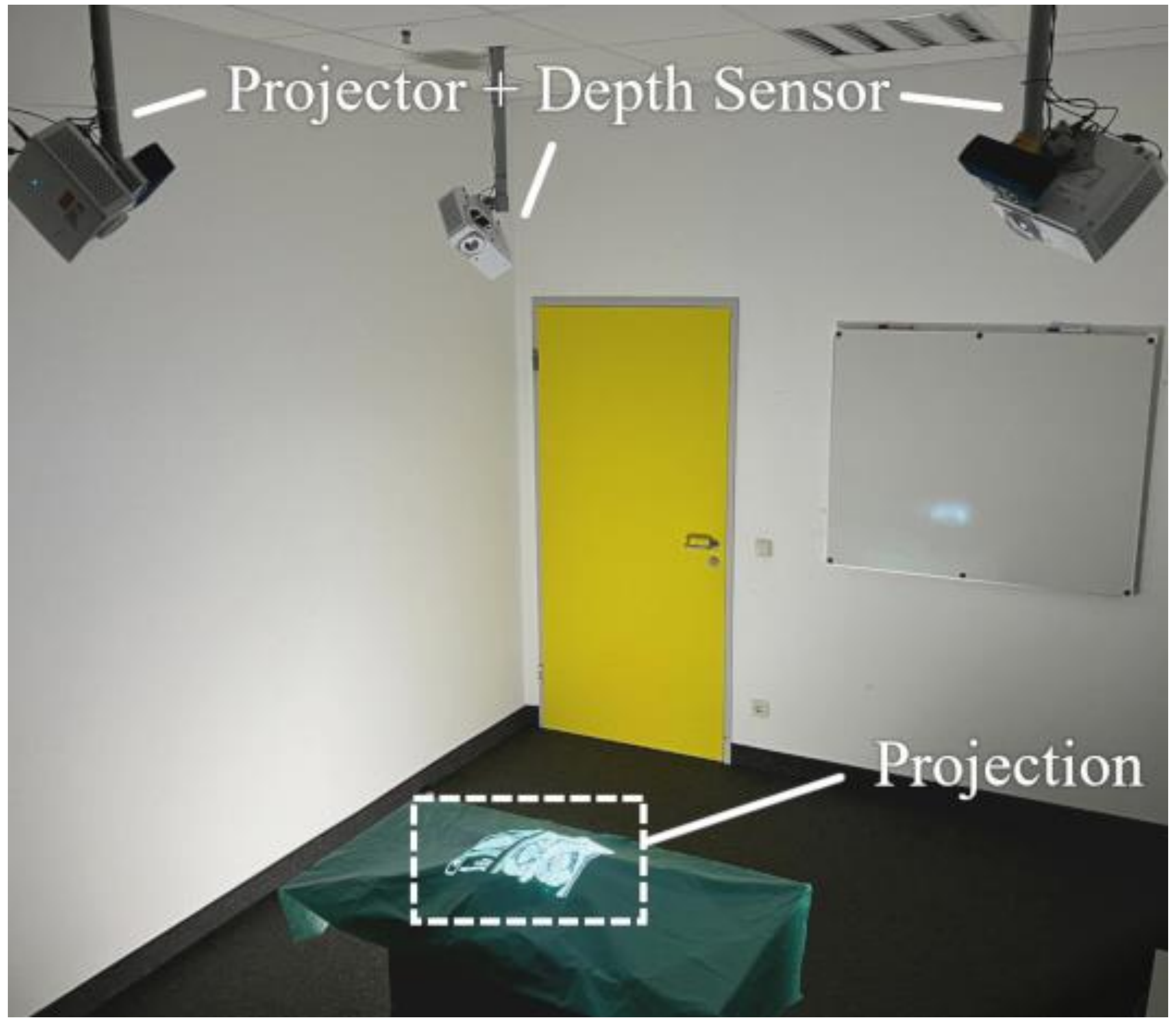


Muehlenbrock et al. [SPIE Medical Imaging, 2022]

Geometric Correction



Muehlenbrock et al. [VRST'25]



Inherently rectified using Point Cloud Rendering!

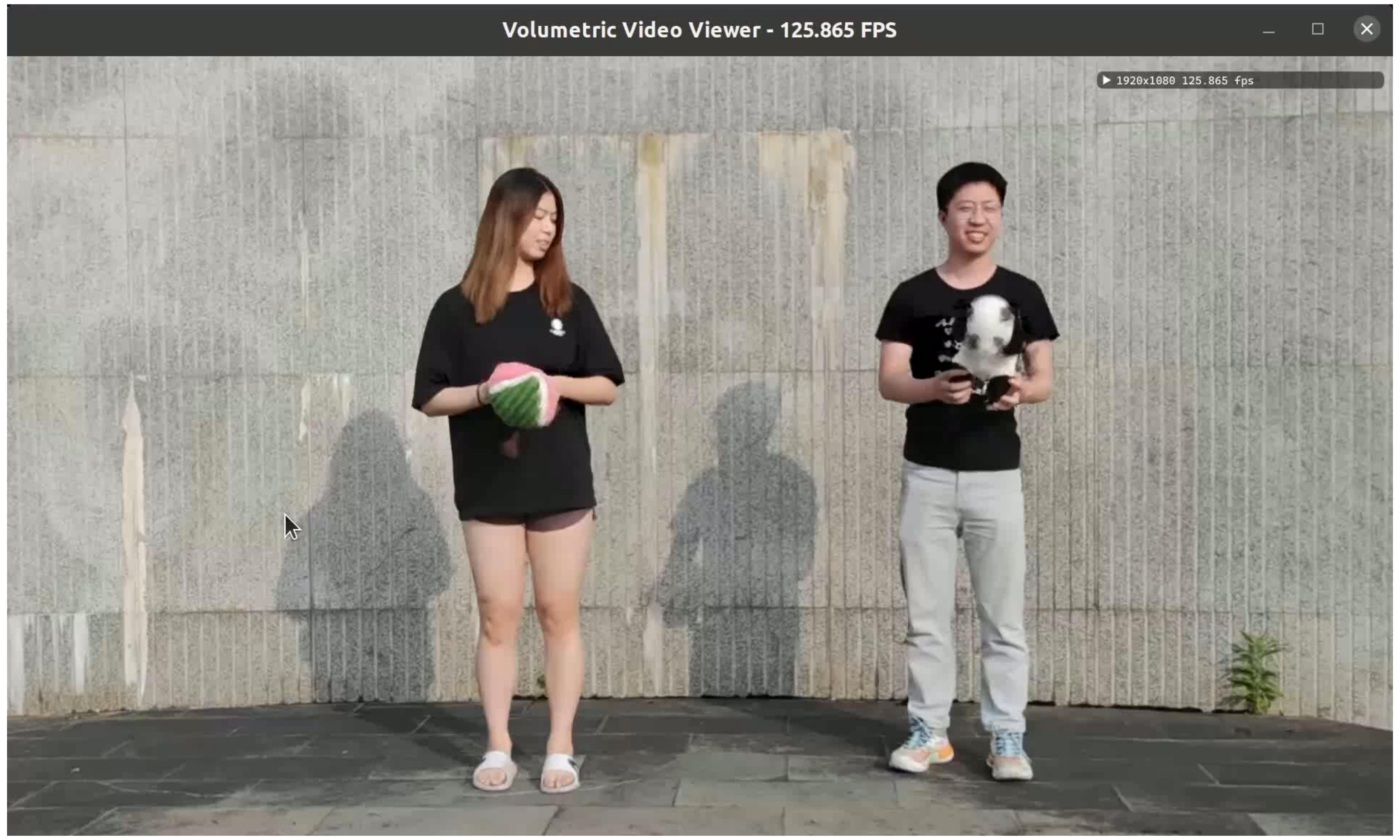
Related Work

- Volumetric Scene Reconstruction
  - Was driven by Microsoft Research
    - E.g., Fusion4D [2016], Motion2Fusion [2017]
    - Highly optimized pipelines utilizing custom RGB-D sensors
  - No source code or comparable re-implementation available
  - Low 'base-level' performance

Motion2Fusion [ToG, 2017]

Related Work

- 4D Gaussian Splatting
  - Very high Quality for static & dynamic scene reconstruction
  - However: **Dynamic ≠ Live!**
  - Requires **Dense** Camera Setup
  - Best ‘Live’ Optimization Techniques:
    - Approx. Seconds for per-frame reconstruction



4K4D [CVPR, 2024]

‘Live’ Technique	GPU	per-Frame
3DGStream [CVPR 24]	3090 RTX	~10s
IGS [CVPR 25]	4x A100	~2.67s
HiCoM [NeurIPS‘24]	4090 RTX	~1.0s

Related Work

- Learned Gaussian Splatting
  - Skips expensive optimization by predicting Gaussian Splats
- Examples:
  - P2ENet [CVPR, 2024]
  - GPS-Gaussian [CVPR, 2024]
  - GPS-Gaussian+ [CVPR, 2025]

P2ENet [CVPR, 2024]

Technique	Technique	GPU	Res	per-Frame
P2ENet [CVPR 24]	RGBD	4090 RTX	512^2	~14 - 37 fps
GPS-G [CVPR 24]	RGBD + RGB	3090 RTX	2048^2 *	~ 25 fps
GPS-G+ [CVPR 25]	RGB only	3090 RTX	1024^2 *	~ 25 fps

Related Work

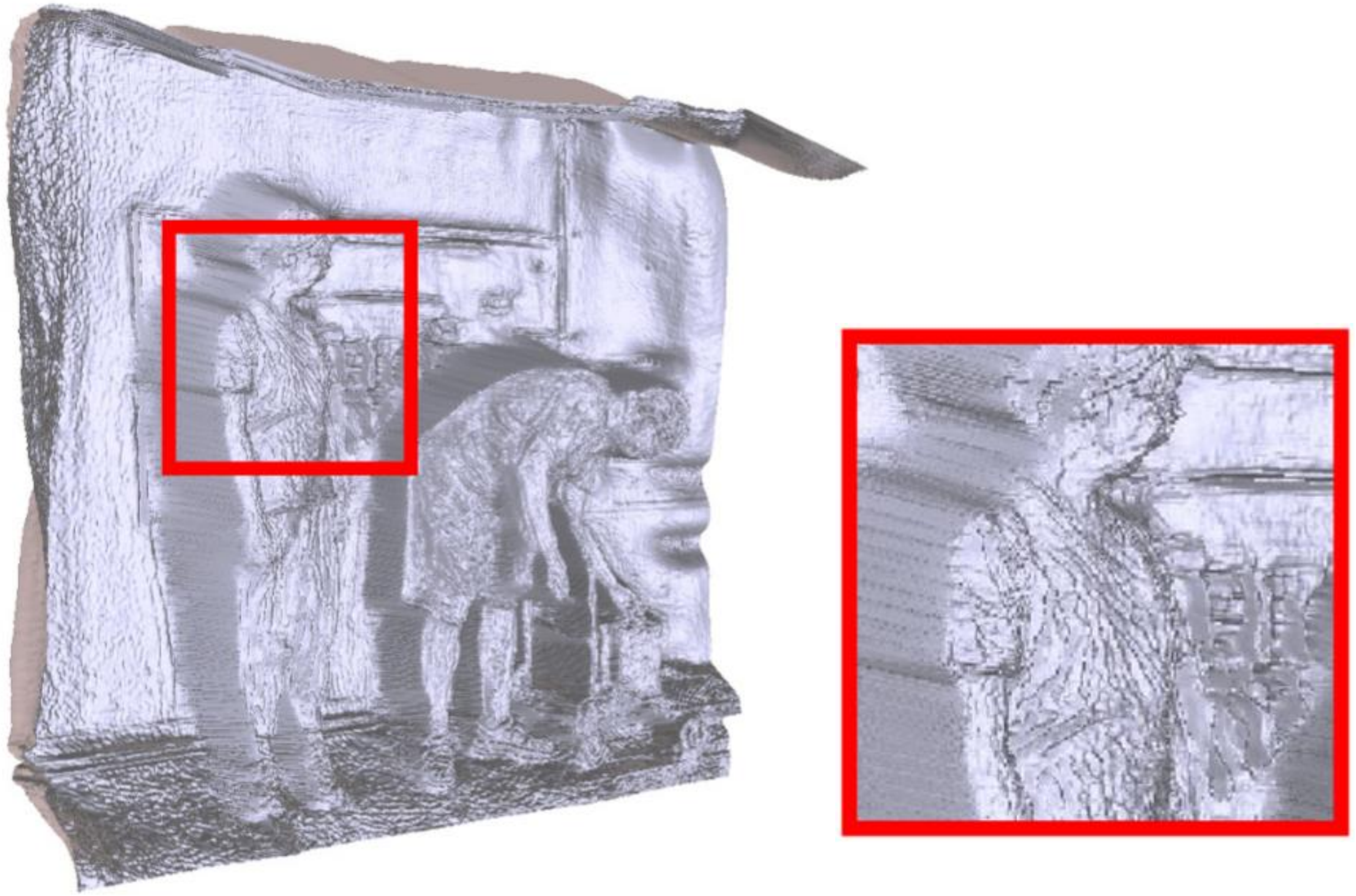
- Learned Gaussian Splatting
  - Skips expensive optimization by predicting Gaussian Splats
- Examples:
  - P2ENet [CVPR, 2024]
  - GPS-Gaussian [CVPR, 2024]
  - GPS-Gaussian+ [CVPR, 2025]

GPS-Gaussian+ [CVPR, 2025]

Technique	Technique	GPU	Res	per-Frame
P2ENet [CVPR 24]	RGBD	4090 RTX	512^2	~14 - 37 fps
GPS-G [CVPR 24]	RGBD + RGB	3090 RTX	2048^2 *	~ 25 fps
GPS-G+ [CVPR 25]	RGB only	3090 RTX	1024^2 *	~ 25 fps

Related Work

- Learned Gaussian Splatting
  - Skips expensive optimization by predicting Gaussian Splats
- Examples:
  - P2ENet [CVPR, 2024]
  - GPS-Gaussian [CVPR, 2024]
  - GPS-Gaussian+ [CVPR, 2025]



GPS-Gaussian+ [CVPR, 2025]

Technique	Technique	GPU	Res	per-Frame
P2ENet [CVPR 24]	RGBD	4090 RTX	512^2	~14 - 37 fps
GPS-G [CVPR 24]	RGBD + RGB	3090 RTX	2048^2 *	~ 25 fps
GPS-G+ [CVPR 25]	RGB only	3090 RTX	1024^2 *	~ 25 fps

Related Work

- So many different and specialized techniques
  - Template-based Fitting, e.g. Runte et al. [VRST '25]
  - Face-Avatars, e.g. Tang et al. [CVPR '25]
  - Pre-Reconstructed Volumetric Videos (e.g. Volograms)
  - …

Related Work

- Still, live VR Applications require:
  - very high frame rates
    - ≥ 90 Hz
  - very high resolutions
    - e.g. Vive Focus Vision: 4896 x 2448

So we want **really fast** point cloud rendering!

➢ Requires faster techniques

➢ For real applications, every millisecond is important!

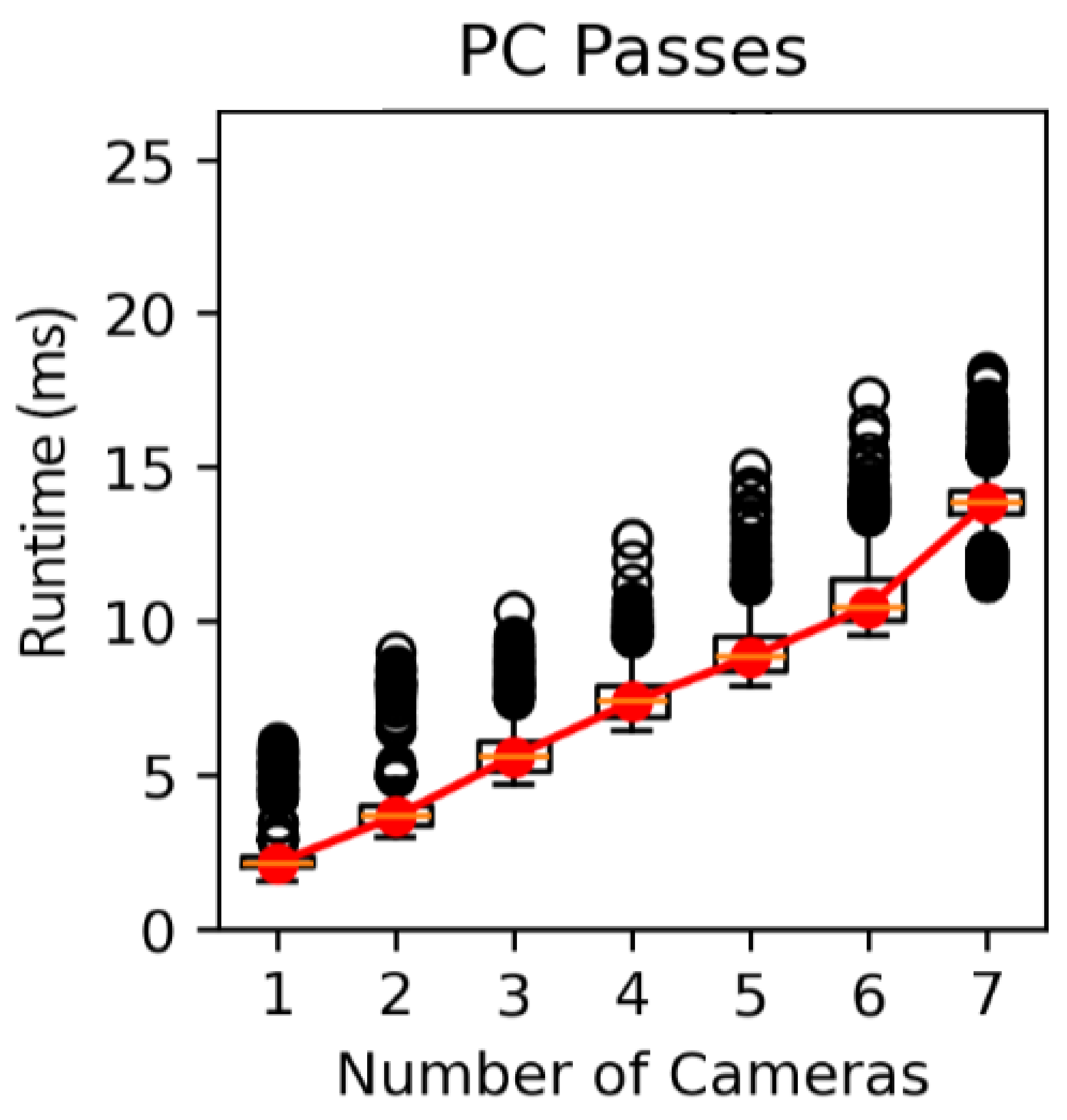
Related Work

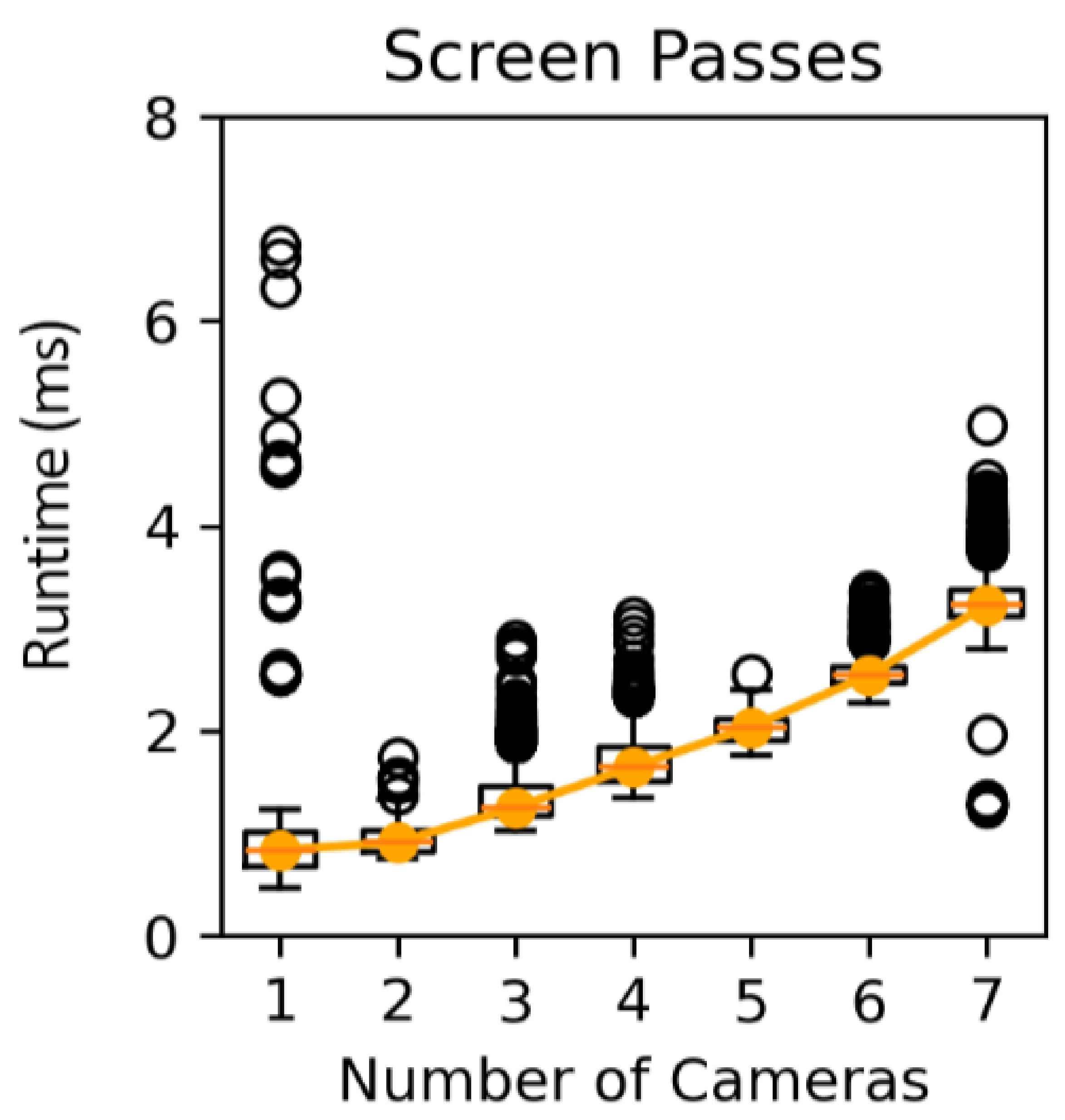
BlendPCR [ICAT-EGVE, 2024]

Related Work

BlendPCR [ICAT-EGVE, 2024]

Related Work

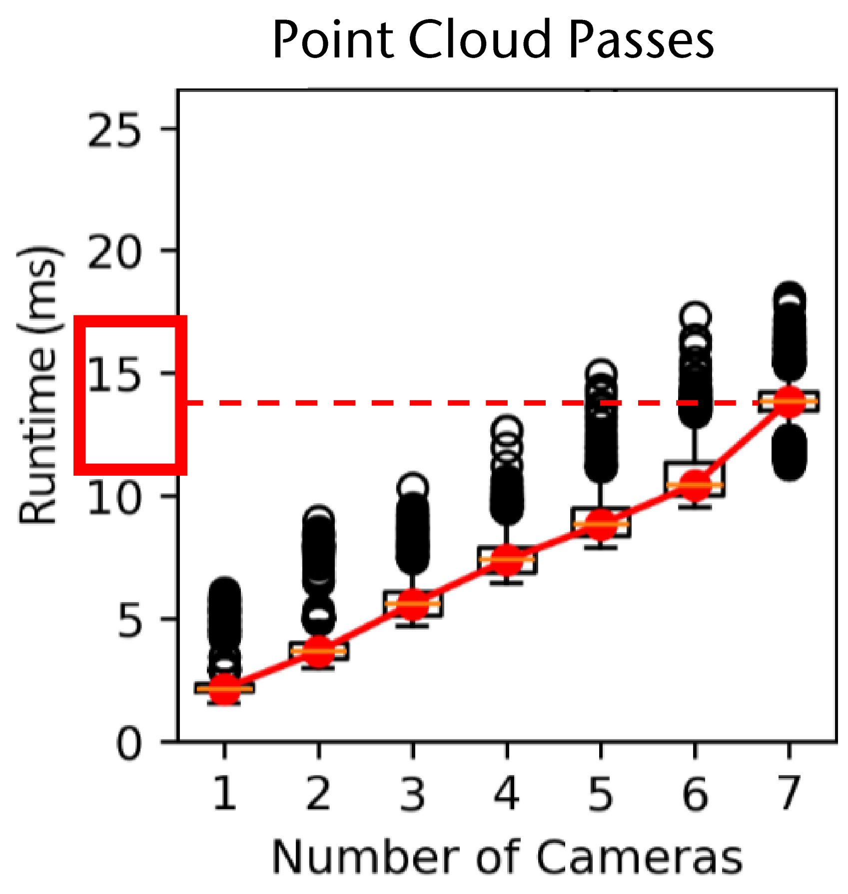


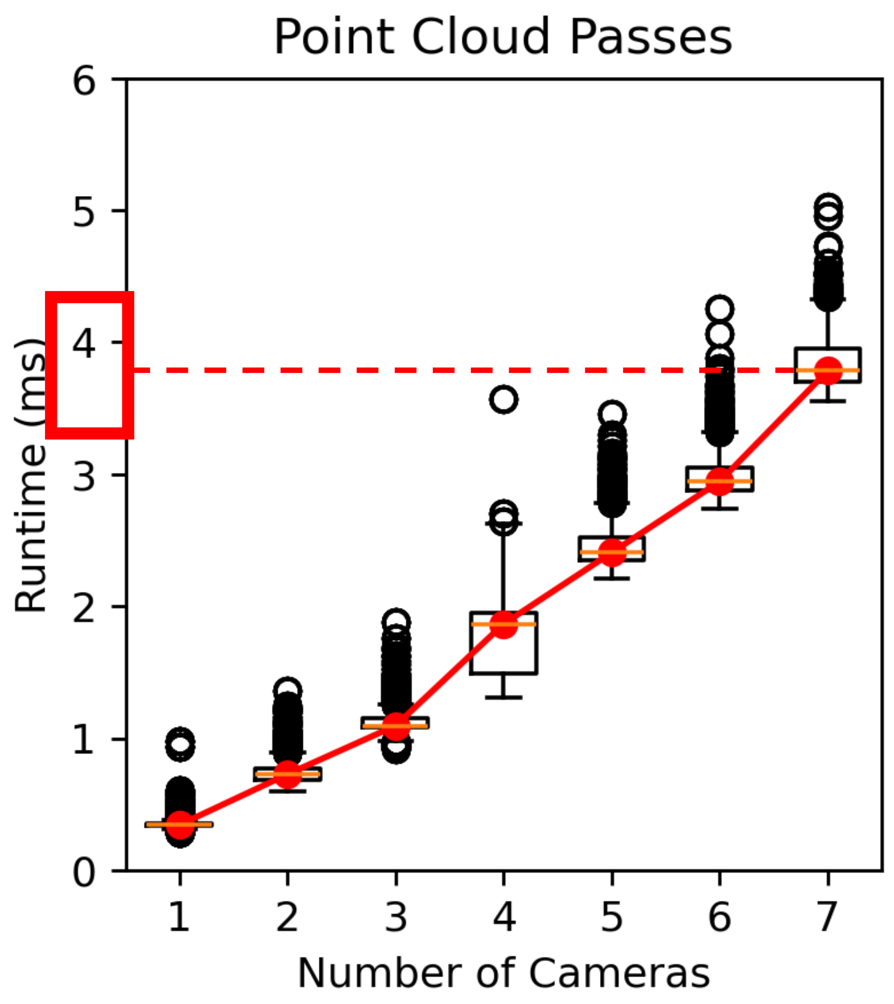


Our Work

Preliminary optimizations:

- Delayed generation of point clouds on GPU
  - saves memory bandwidth (factor: 4!)
- Discarding Geometry Shader
  - implementing triangle rejection in Vertex Shader
- Optimizing kernel parameters (e.g. size) without impacting visual quality



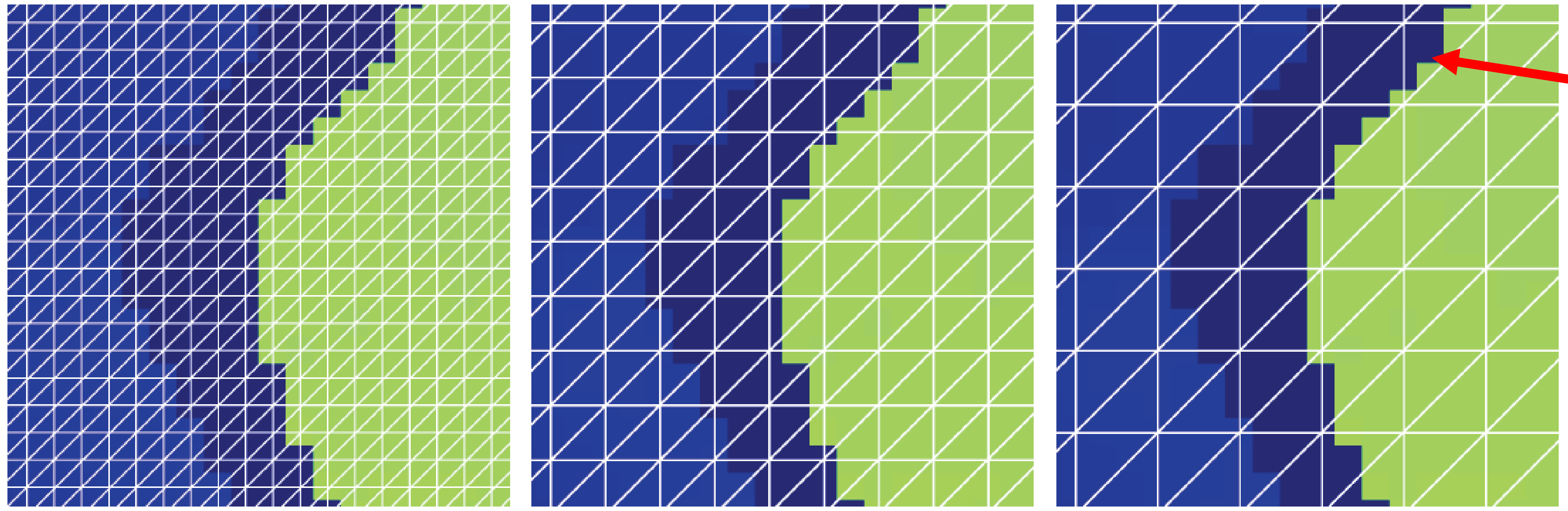


Our Work

Identifying Major Parameters

1. Mesh Grid Resolution

   - Mesh Resolution less important for reality perception compared to textures, e.g. Warsinke et al. [VRST'25]



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

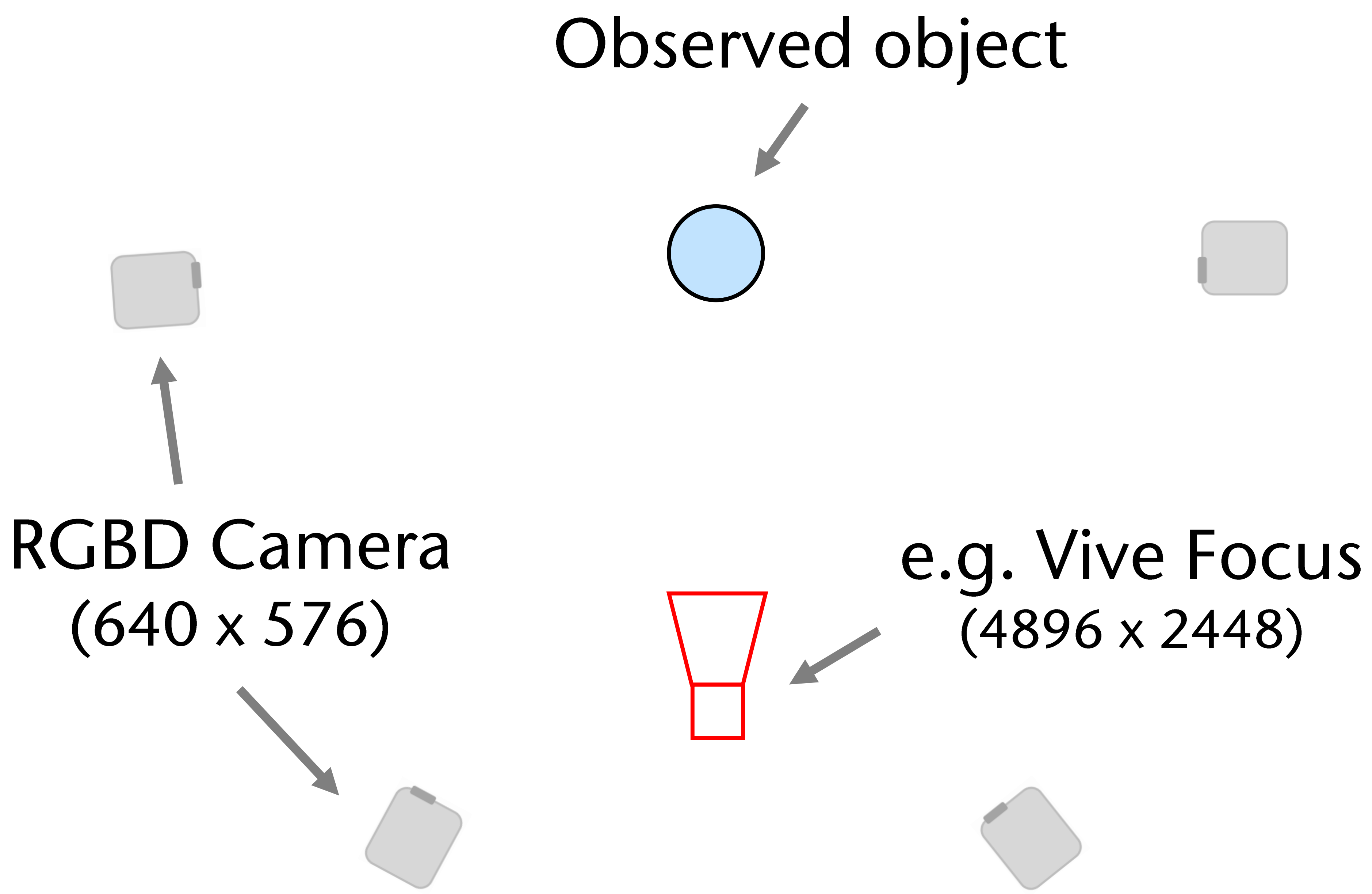
We also tried adaptive resolution at borders, but this directly cancled the performance advantage.

Our Work

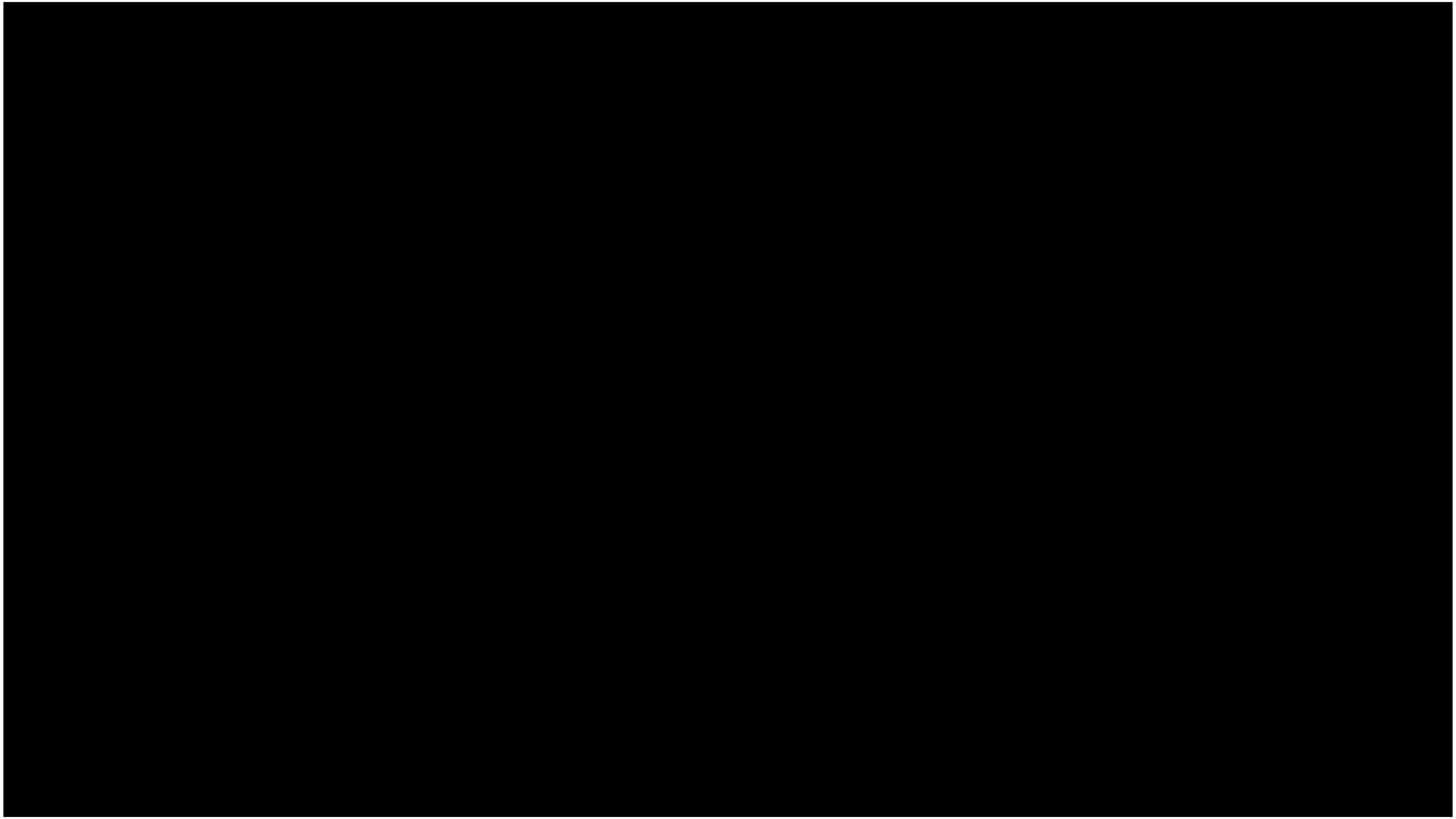
Defining Quality Parameters

2. Framebuffer Object Resolution
   - Textures typically oversampled
   - At most viewing distances, halving FBO resolution has almost no visible impact:

Ultra HD | Full HD | Ultra HD | Full HD



Result: Effect of Mesh Grid Resolution



Result: Effect of Both Combined

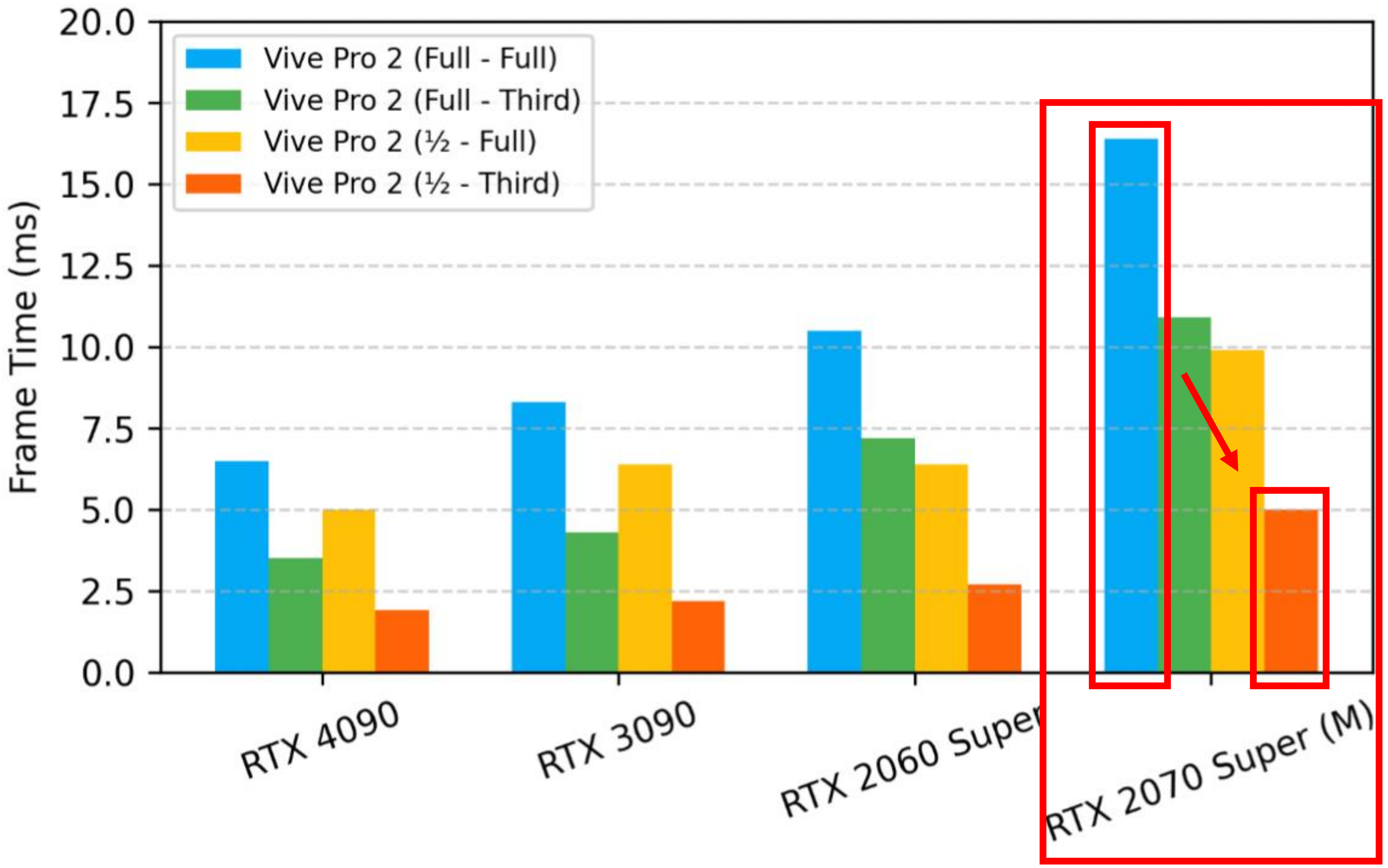
Performance Measurement

Total frame time of a single human in milliseconds (ms)

	Resolution (per view)	Views	RTX 4090 Third	RTX 4090 Half	RTX 4090 Full	RTX 3090 Third	RTX 3090 Half	RTX 3090 Full	RTX 2060 Super Third	RTX 2060 Super Half	RTX 2060 Super Full	RTX 2070 Super (M) Third	RTX 2070 Super (M) Half	RTX 2070 Super (M) 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

≤ 8.3 ms | ≤ 11.1 ms | ≤ 33.3 ms | > 33.3 ms

Performance Measurement



Limitations

- Reducing Mesh Grid Resolution
  - Quality of distant objects (with already low resolution) may degrade visibly.
- Reducing Framebuffer Object Resolution
  - At large viewing distances (with less oversampling) this may lead also to visible quality reduction.
- Possible solution: Make it adaptive.

Conclusion

- Significantly optimized point cloud rendering
  - Preliminary lossless performance optimizations
  - Two optimization parameters with huge performance but marginal visible impact
    - Mesh Grid Resolution
    - Framebuffer Object Resolution
  - Factor: 3 – 5 x faster

Source Code (GL)

Thank you for your attention!

Andre Muehlenbrock, Rene Weller, Gabriel Zachmann
Computer Graphics & Virtual Reality Lab (CGVR), University of Bremen
Contact: muehlenb@uni-bremen.de