FlashBack

Immersive Virtual Reality on Mobile Devices via Rendering Memoization

Kevin Boos David Chu Eduardo Cuervo

RICE Unconventional Wisdom

Microsoft Research

MobiSys 2016


VR ONE
STORZ
15
20
30.4
LEFT

high graphic complexity
photo-realistic, wide FOV

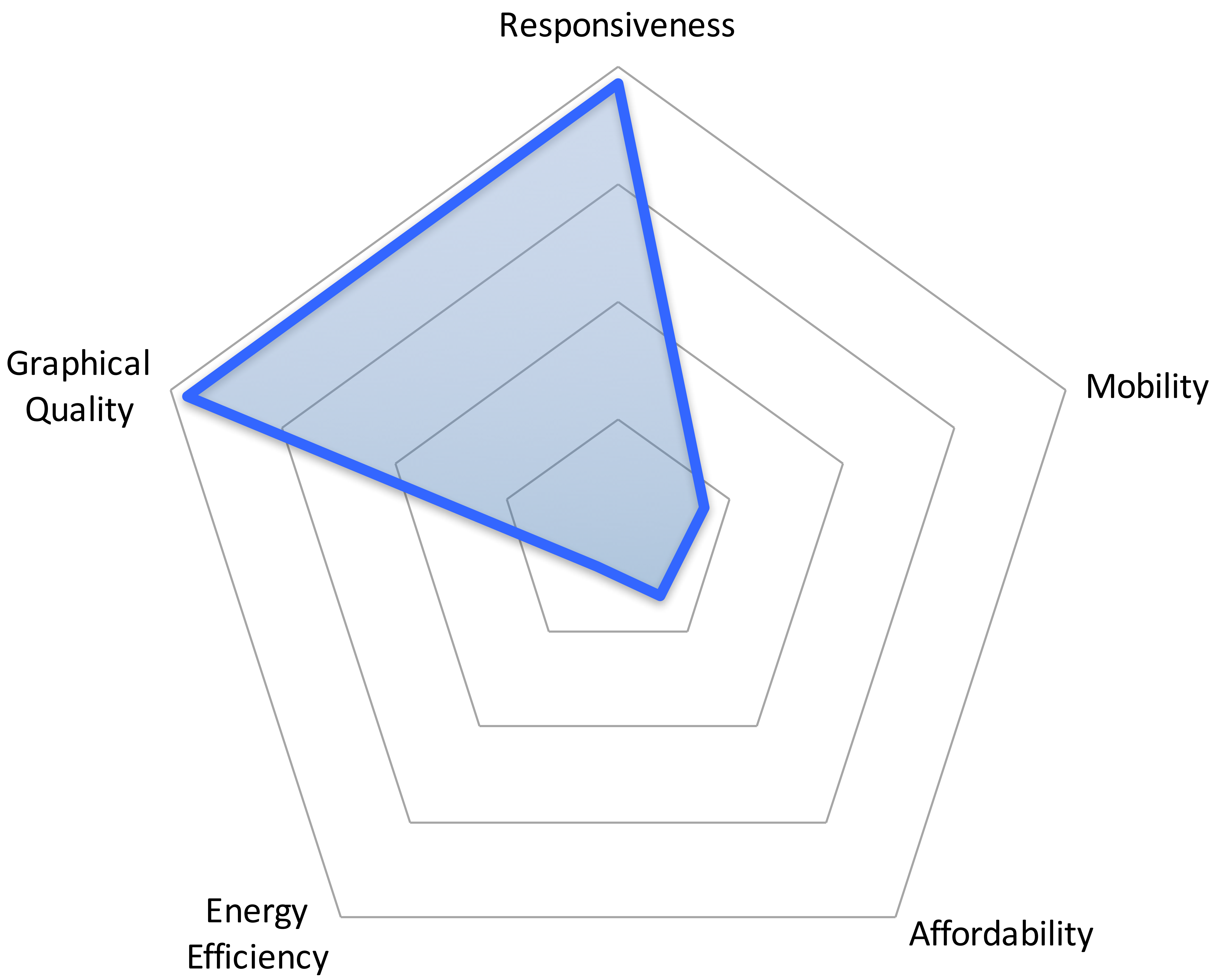
low latency
< 25ms

high framerate
60+ FPS

mobile devices cannot
meet these demands

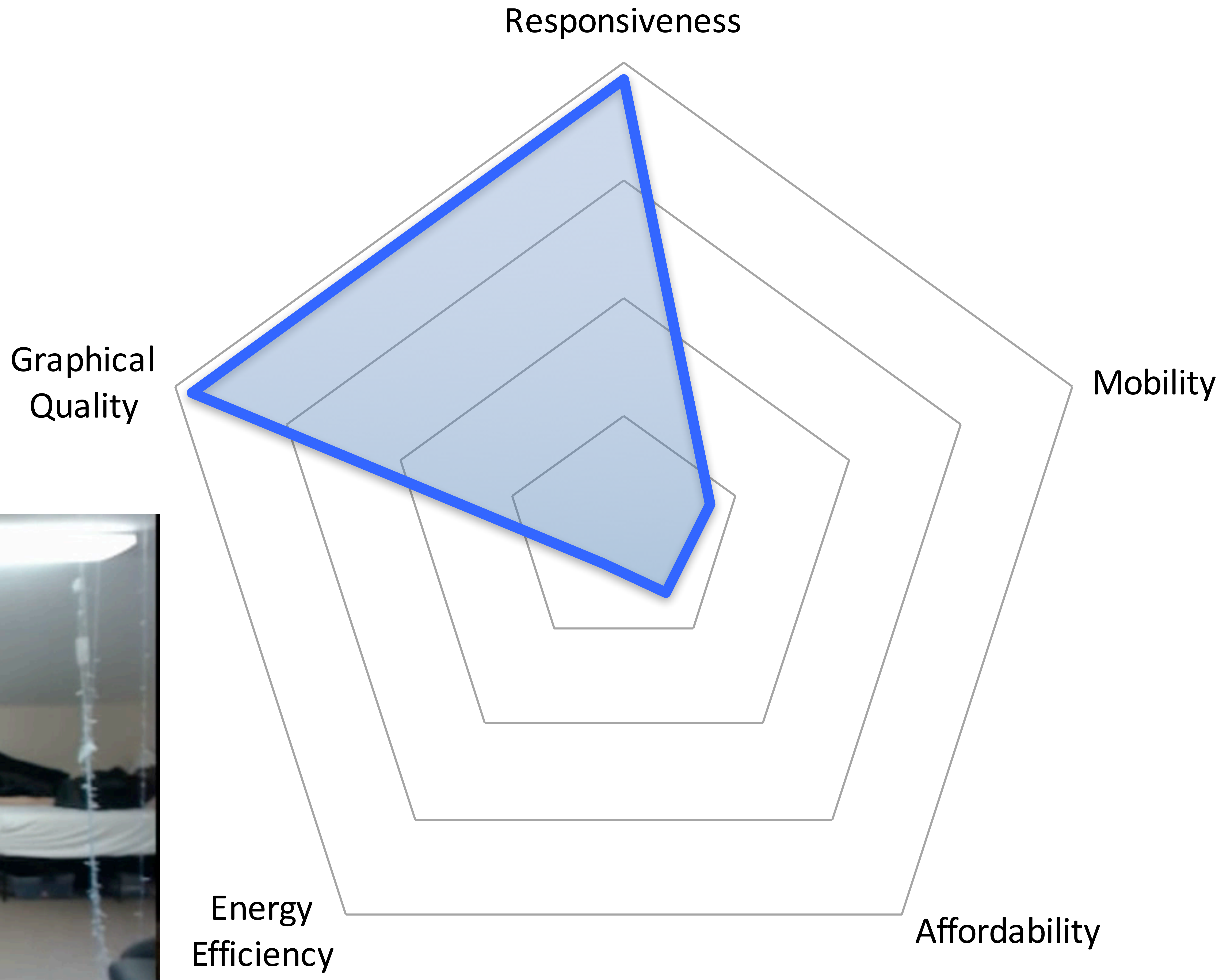
Current VR Landscape

- Tethered HMDs



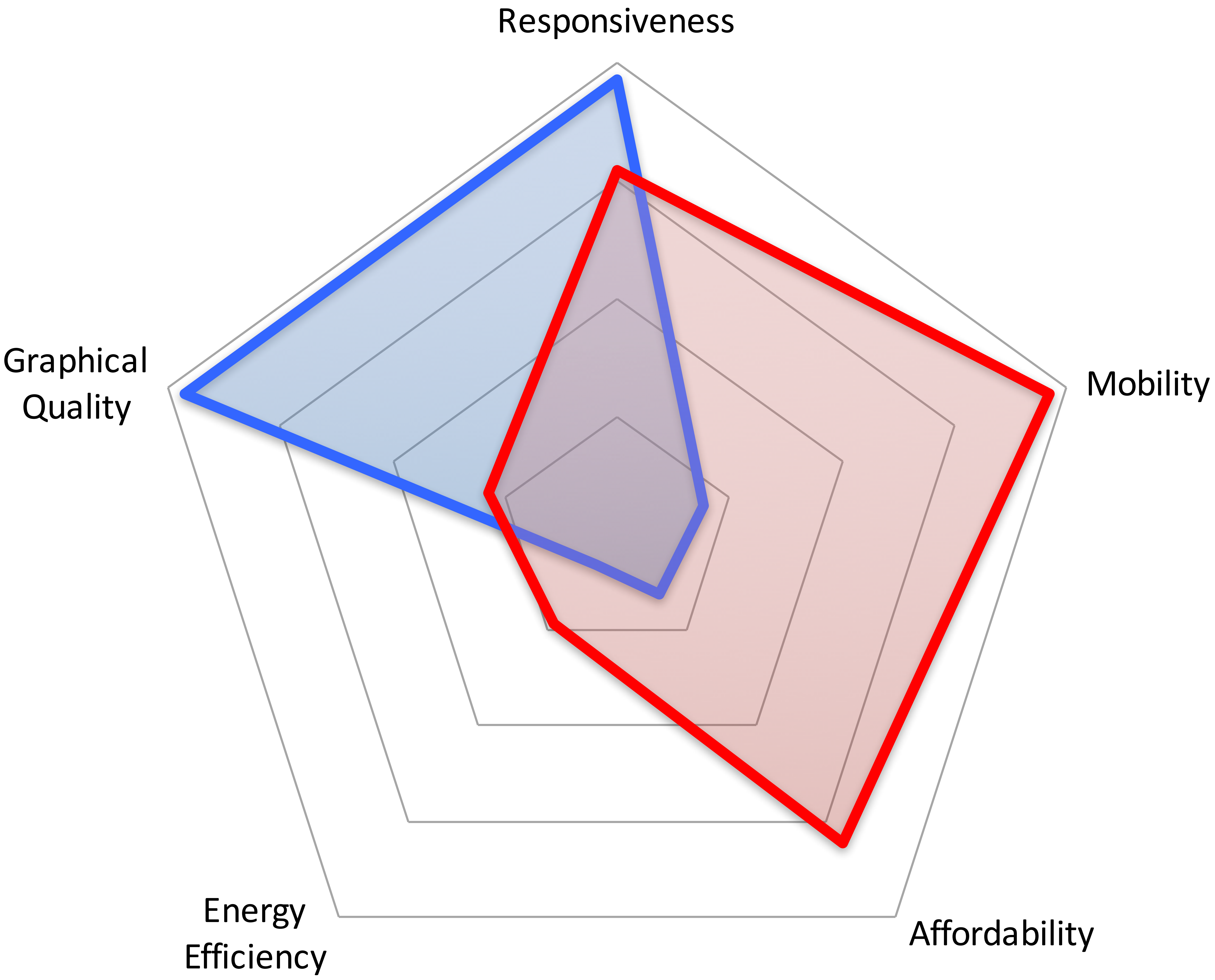
Current VR Landscape

- Tethered HMDs



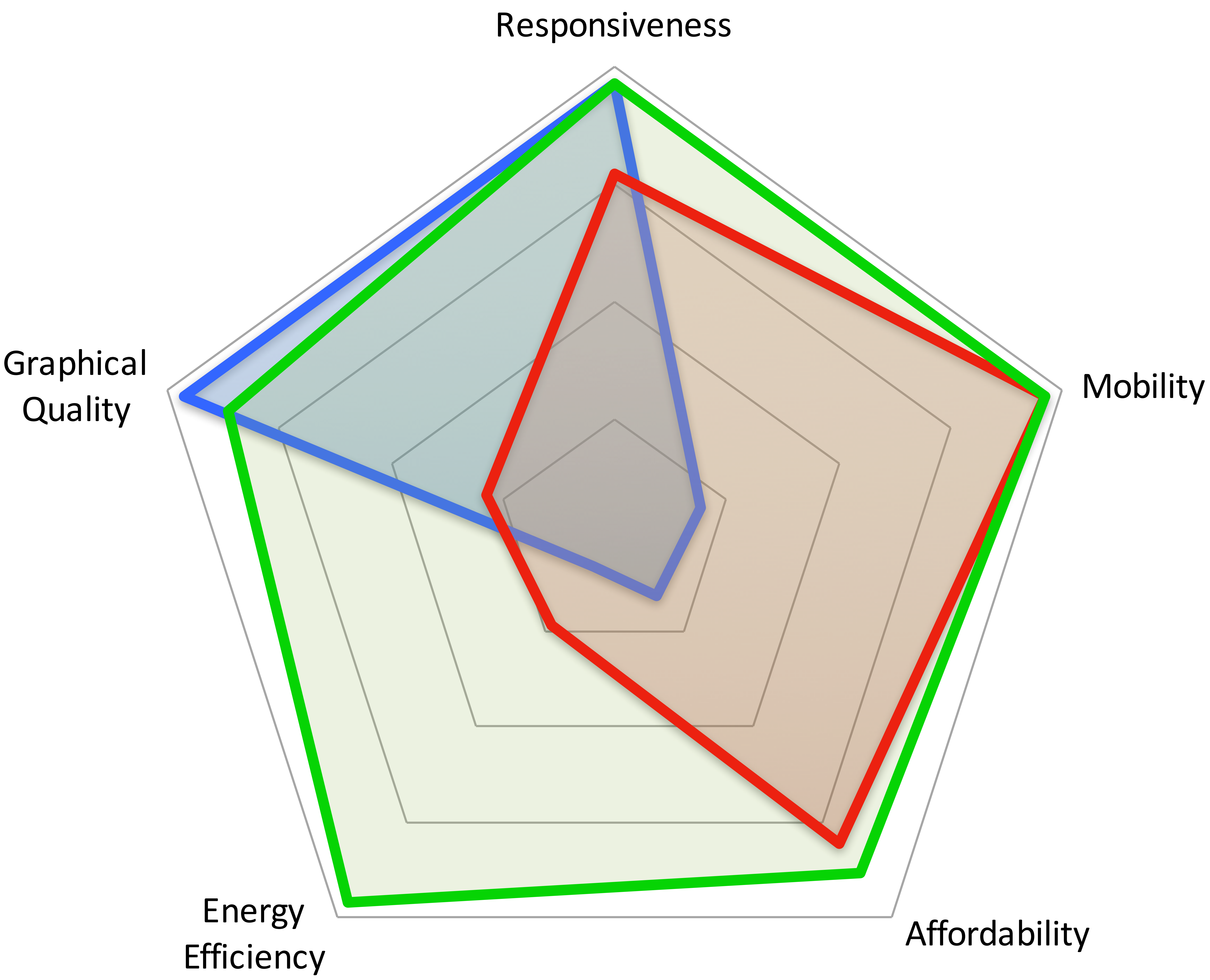
Current VR Landscape

- Tethered HMDs
- Mobile HMDs



Current VR Landscape

- Tethered HMDs
- Mobile HMDs
- FLASHBACK

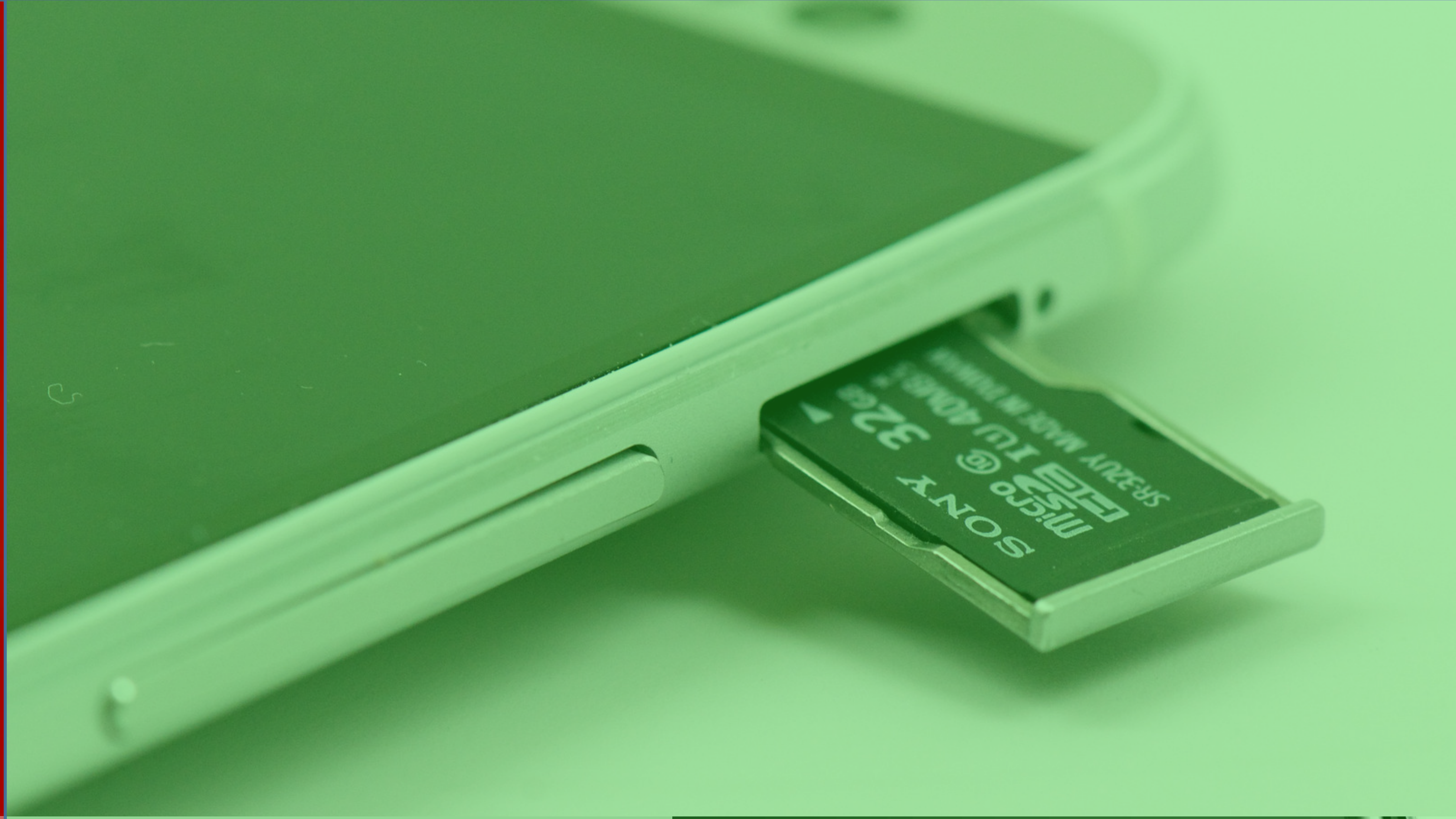


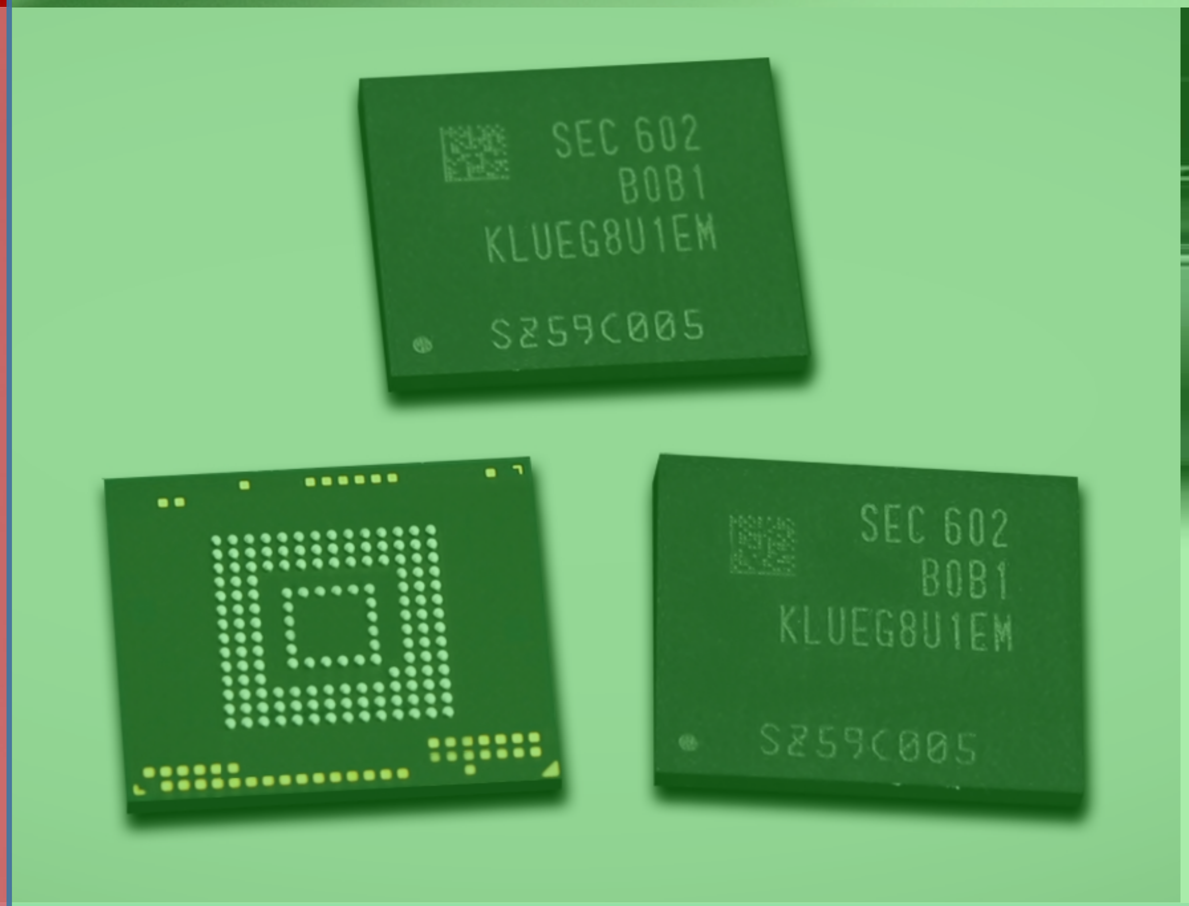
FlashBack objectives

mobility	self-contained on mobile device
graphic quality	desktop-level image quality
responsiveness	low end-to-end latency
energy efficiency	long battery life, low thermal output
affordability	no specialized hardware

Design

Mobile GPUs are constrained





Mobile storage is abundant

key idea:

pre-render all possible views
for fast real-time replay

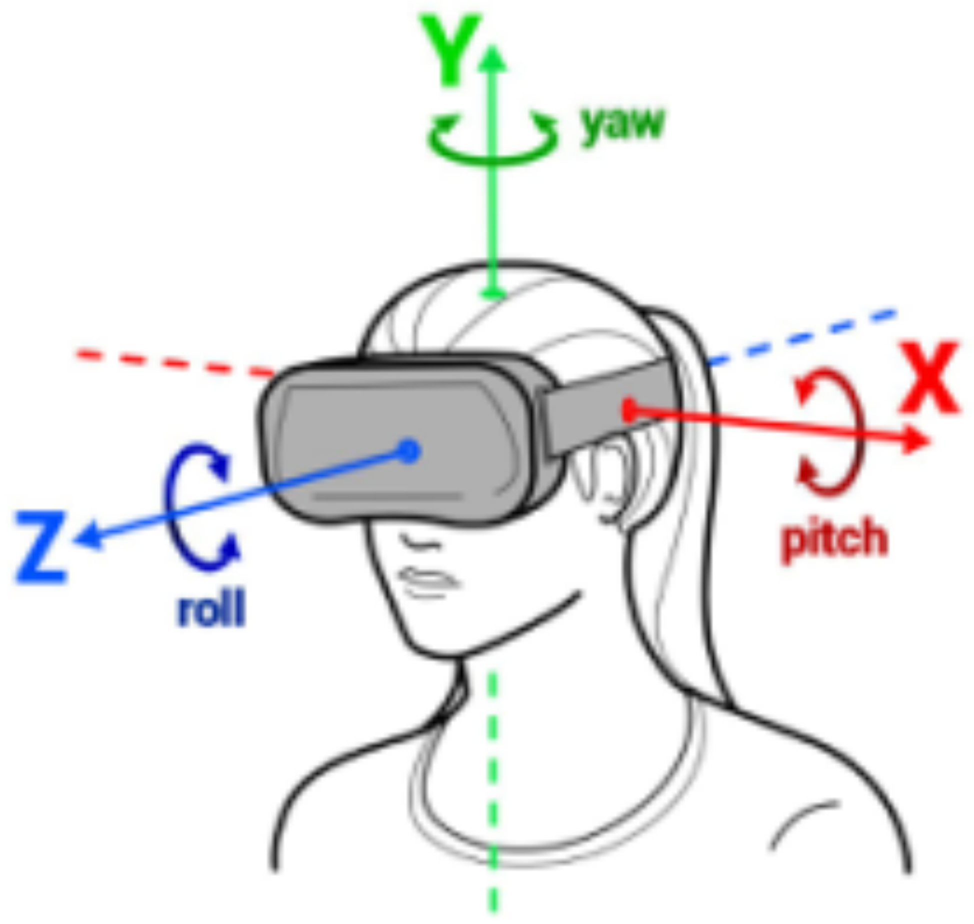
FLASHBACK design & challenges

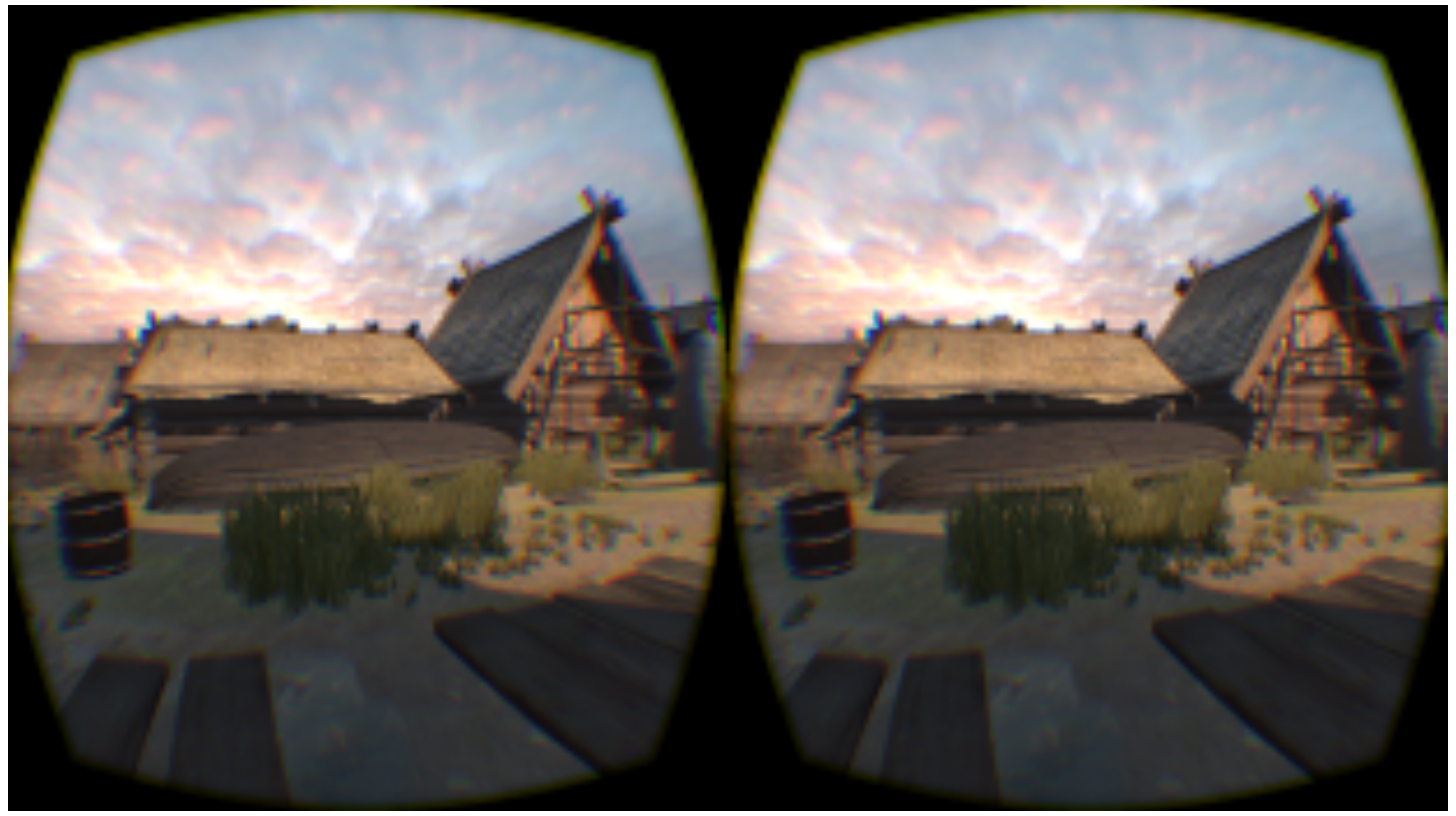
- Pre-rendering [offline]
  - infinite input space
- Live playback [runtime]
  - huge cache, fast retrieval
  - inexact query matching
- Dynamic objects

FlashBack design & challenges

- **Pre-rendering [offline]**
  - infinite input space
- Live playback [runtime]
  - huge cache, fast retrieval
  - inexact query matching
- Dynamic objects

Pre-rendering: map pose to frame





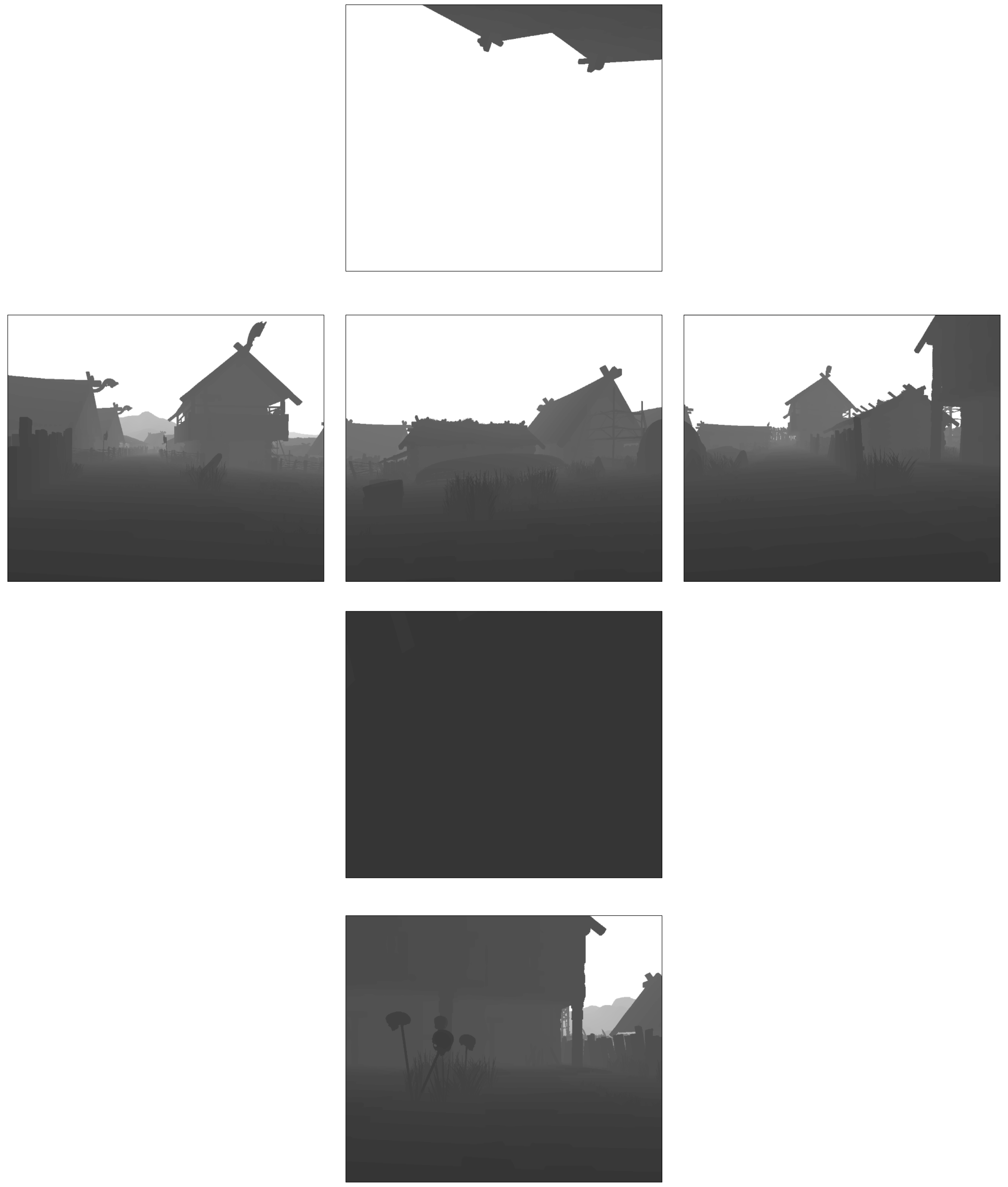
pose
(key)

frame
(value)

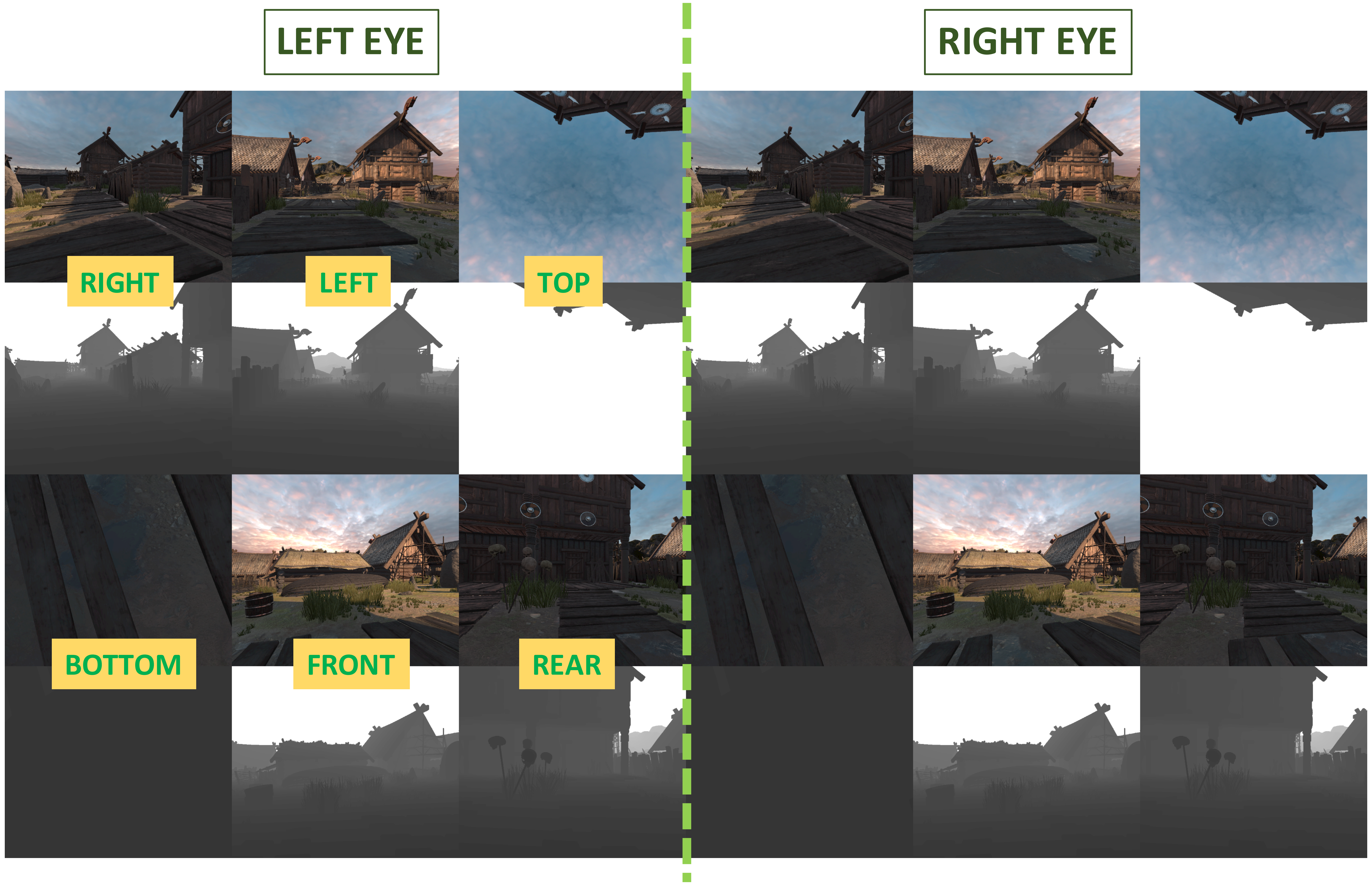
infinite input space

Pre-rendering

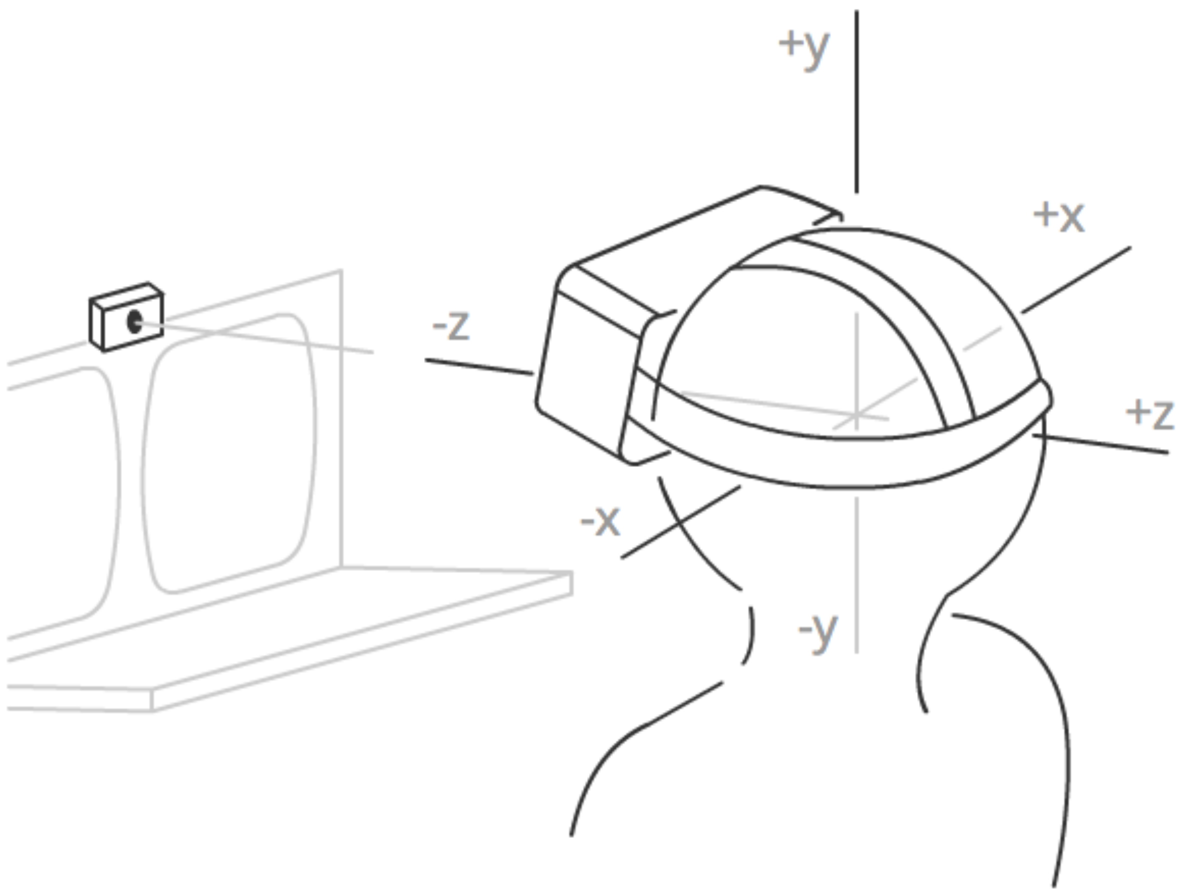
input = { 3D position, ~~3D orientation~~ }



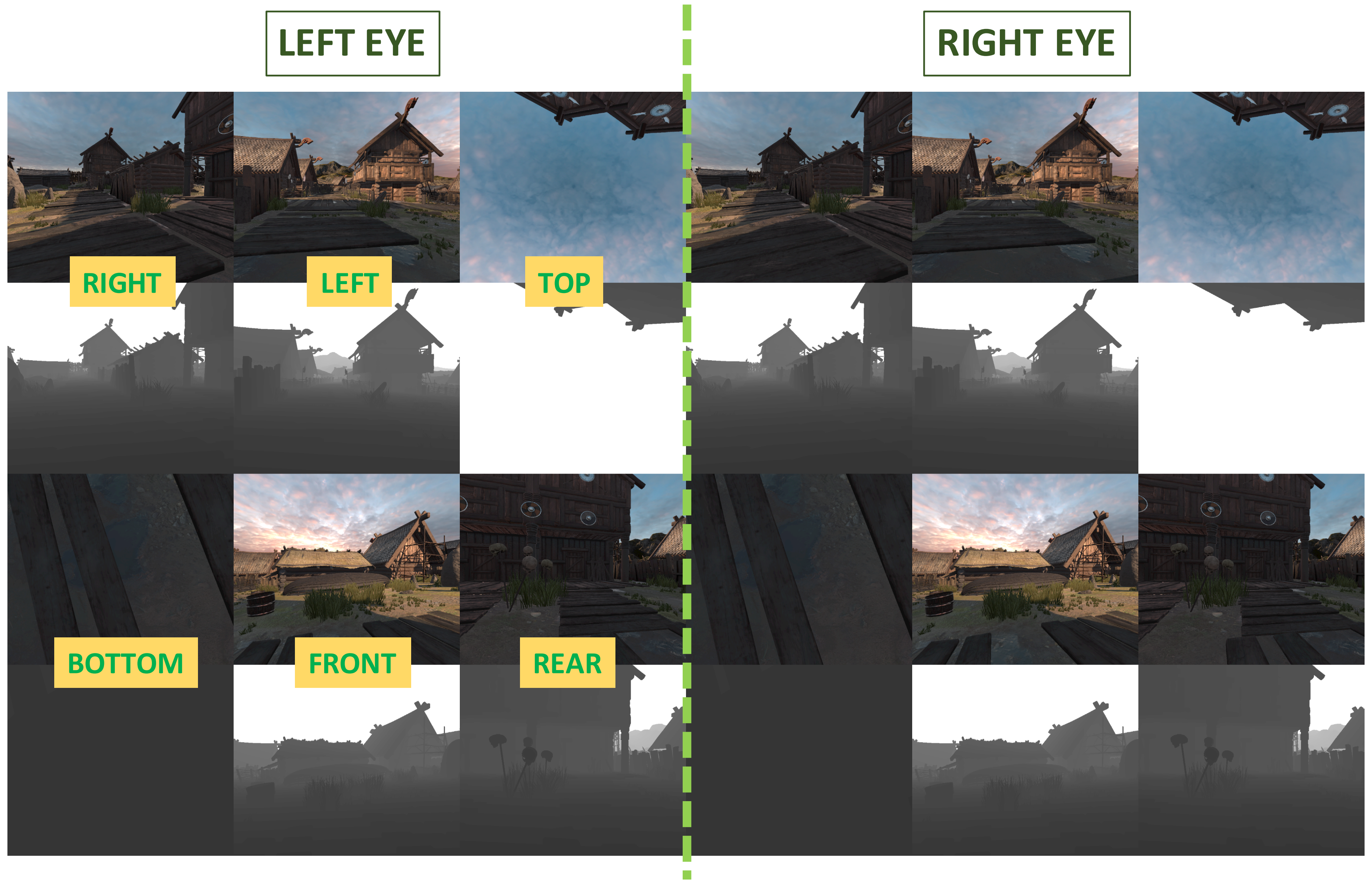
Pre-rendering: megaframes



Pre-rendering: megaframes



position
(key)



megaframe
(value)

Further reducing the input space

- Automate iteration over possible inputs
  - Prune unreachable player positions
- Configurable quantization granularity

Full-coverage frame cache is available at runtime

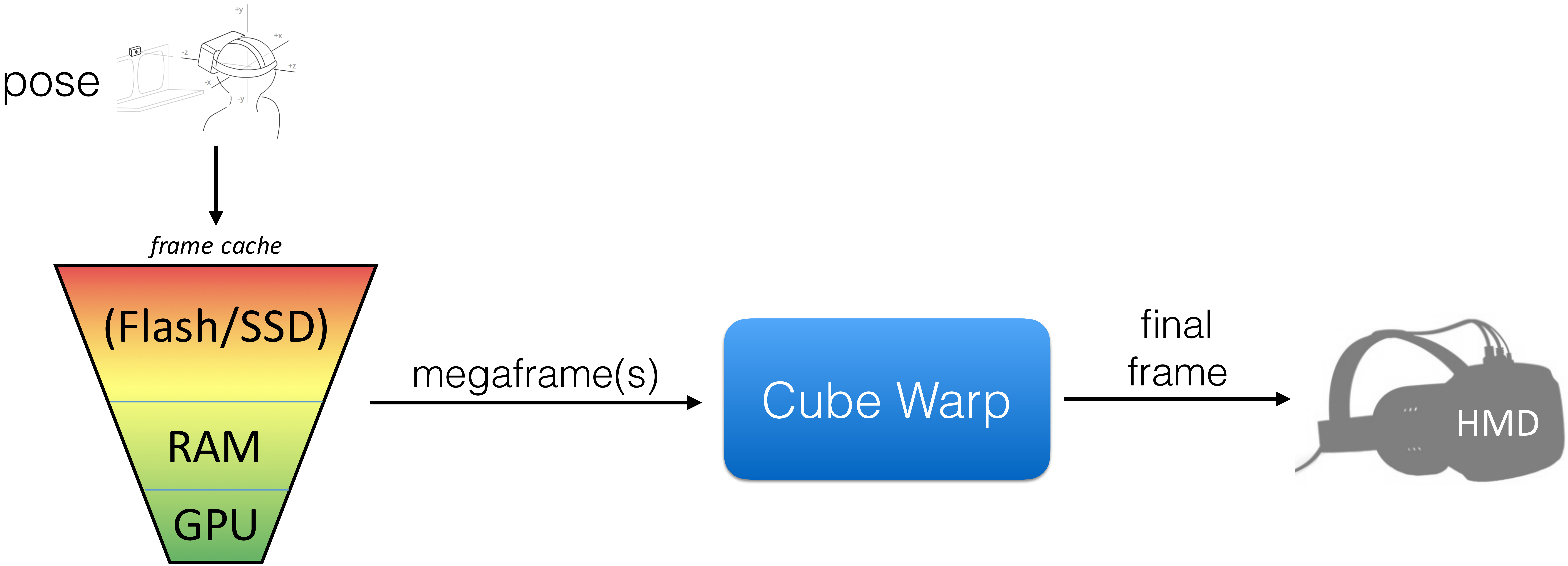
FLASHBACK design & challenges

- Pre-rendering [offline]
  - infinite input space

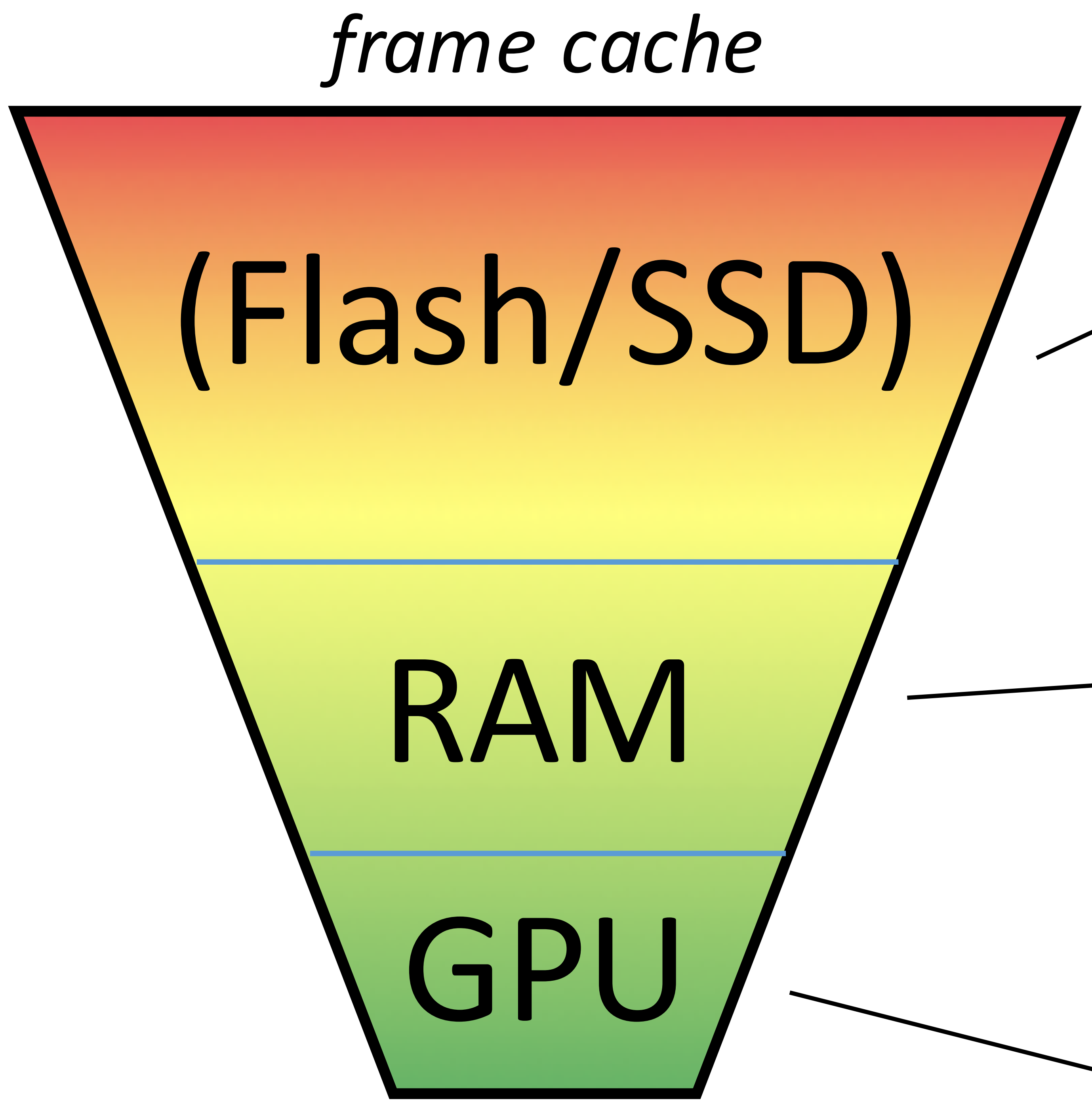
- **Live playback [runtime]**
  - huge cache, fast retrieval
  - inexact query matching

- Dynamic objects

Live playback overview



Building the cache



L3: secondary storage – 9.2 ms

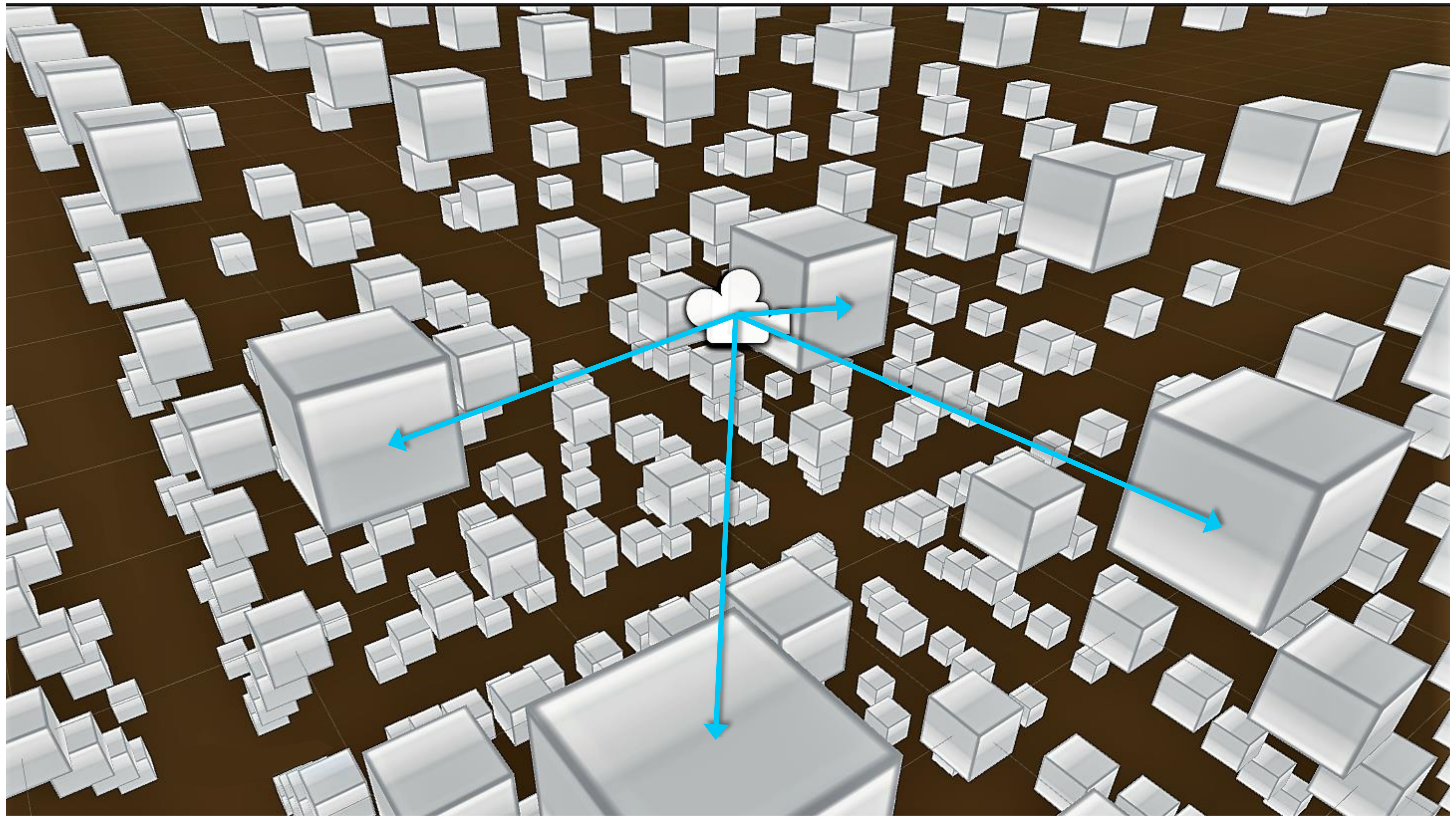
L2: system RAM – 8.7ms

L1: GPU VRAM – **0.35 ms**

- raw, decoded megaframes

✓ fast retrieval from huge cache

Spatially indexing the cache



inexact
query matching

- R-trees: fast, n-D nearest-neighbor search algorithm
- Two cache indices:
  - Universal
  - GPU-only

FlashBack design & challenges

- Pre-rendering [offline]
  - infinite input space
- Live playback [runtime]
  - huge cache, fast retrieval
  - inexact query matching
- **Dynamic objects**

Pre-rendering dynamic objects

- Extension of static scene
- 7D input space:
  - 3D relative position
  - 3D object rotation
  - 1D animation sequence

- Supports arbitrary motion paths and animations
  - but most are periodic

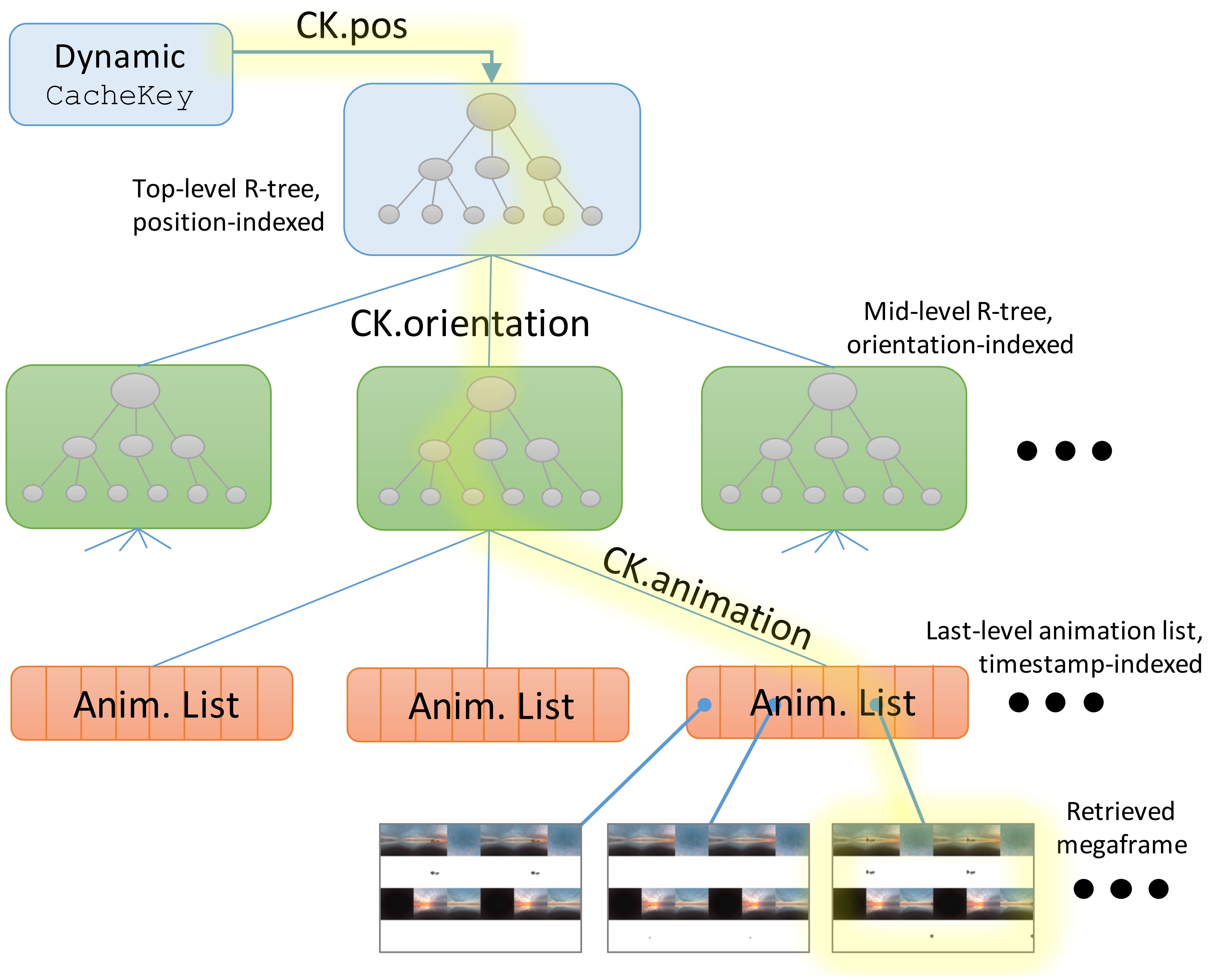
Automated megaframe capture

Dynamic object cache indexing

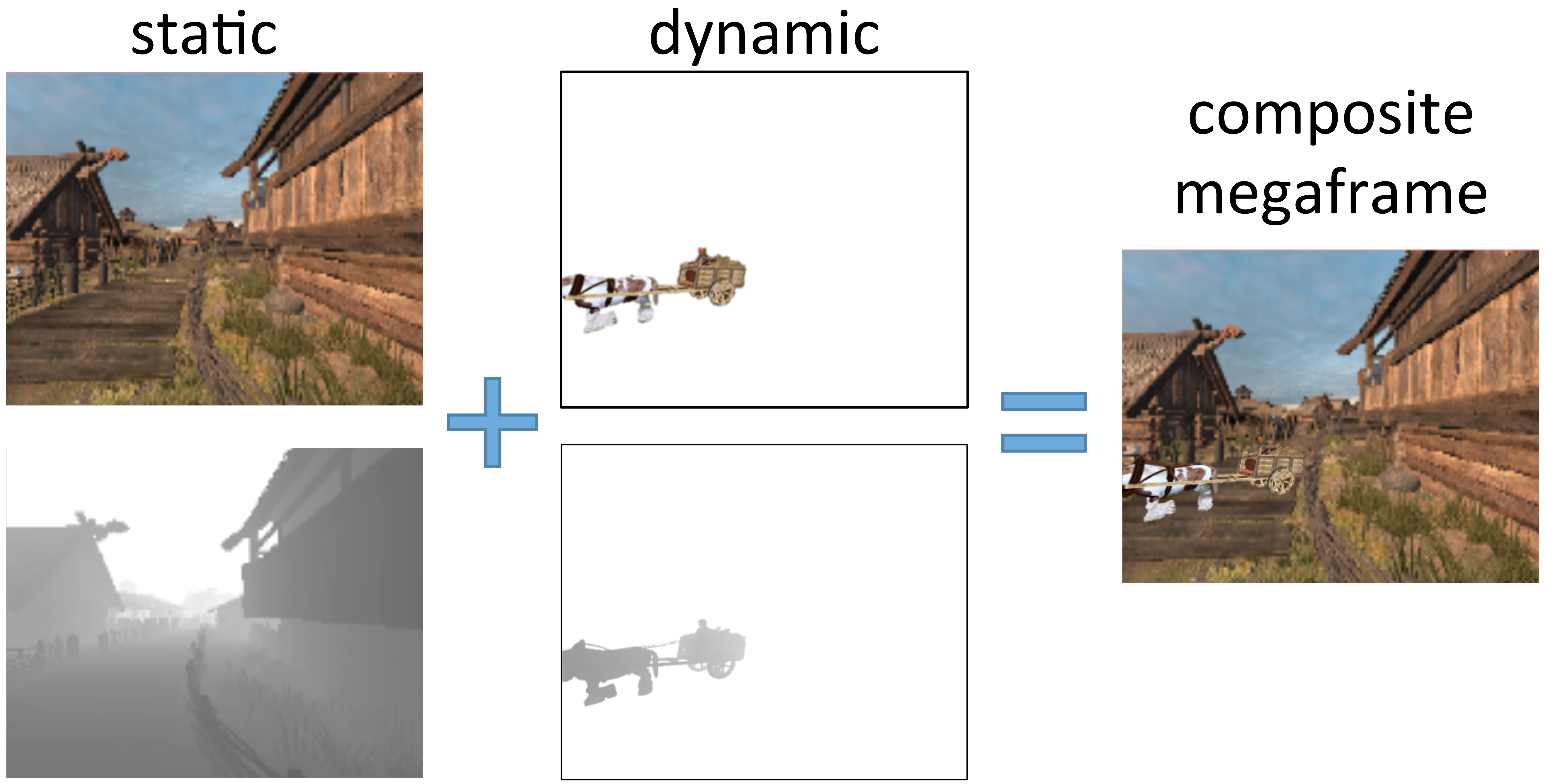
- A 7D query is not meaningful
  - Decompose into chain of queries

= Faster queries

- Can prune search space at each level



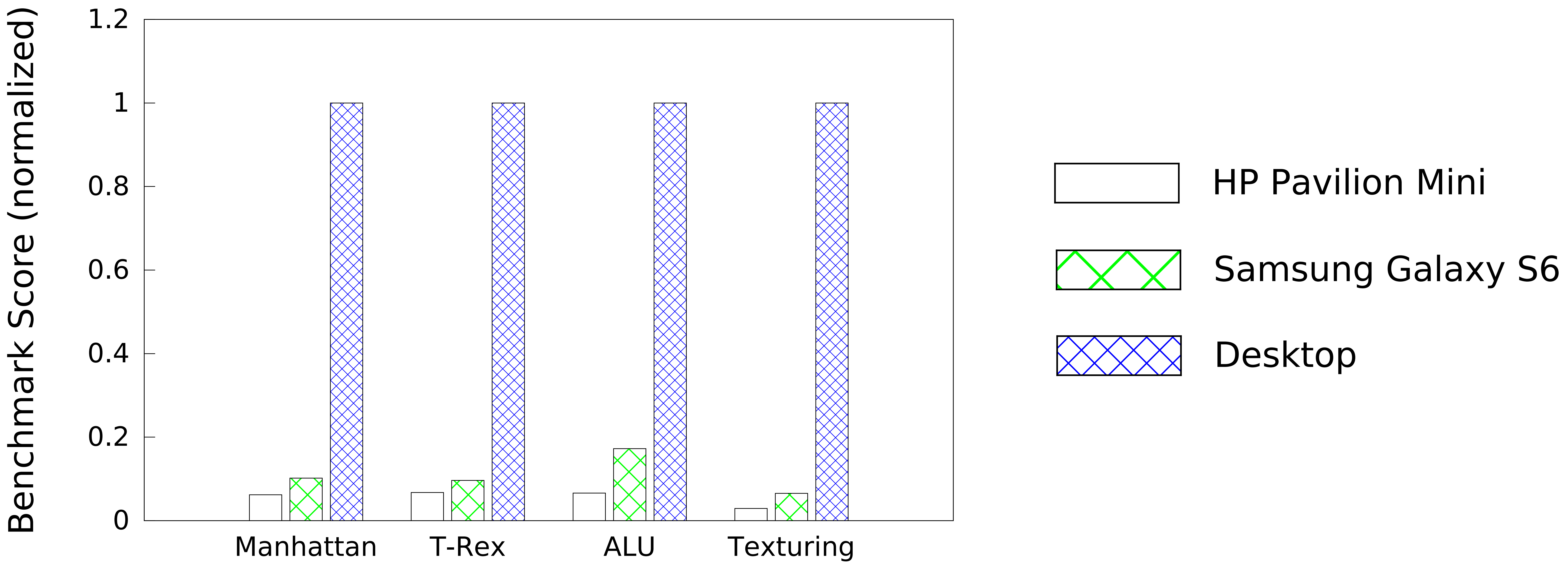
Dynamic + static compositing



Evaluation

Evaluation setup

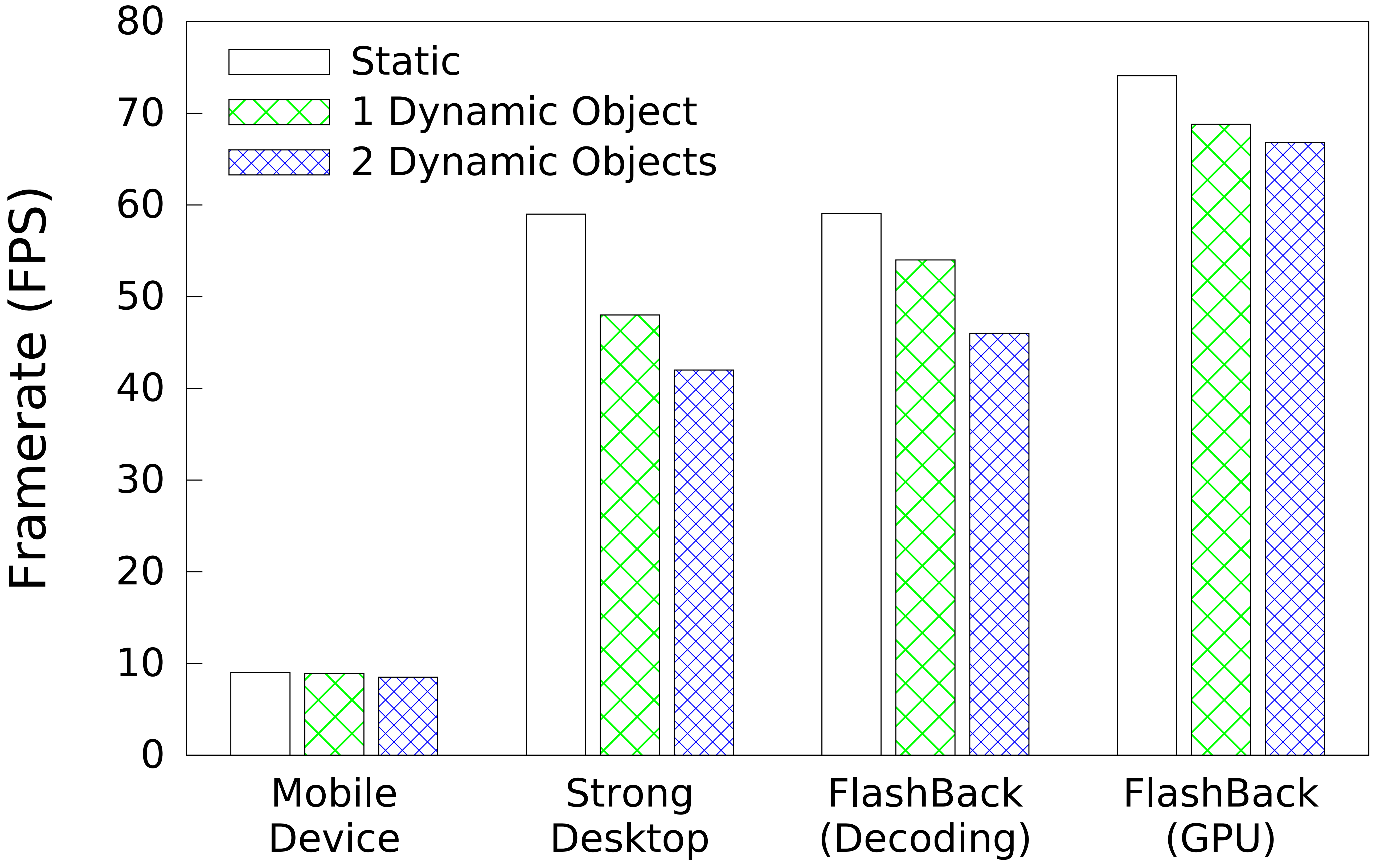
- HP Pavilion Mini + Oculus Rift DK2
  - Small weak computer approximates mobile device
  - Underperforms Samsung Galaxy S6 Gear VR



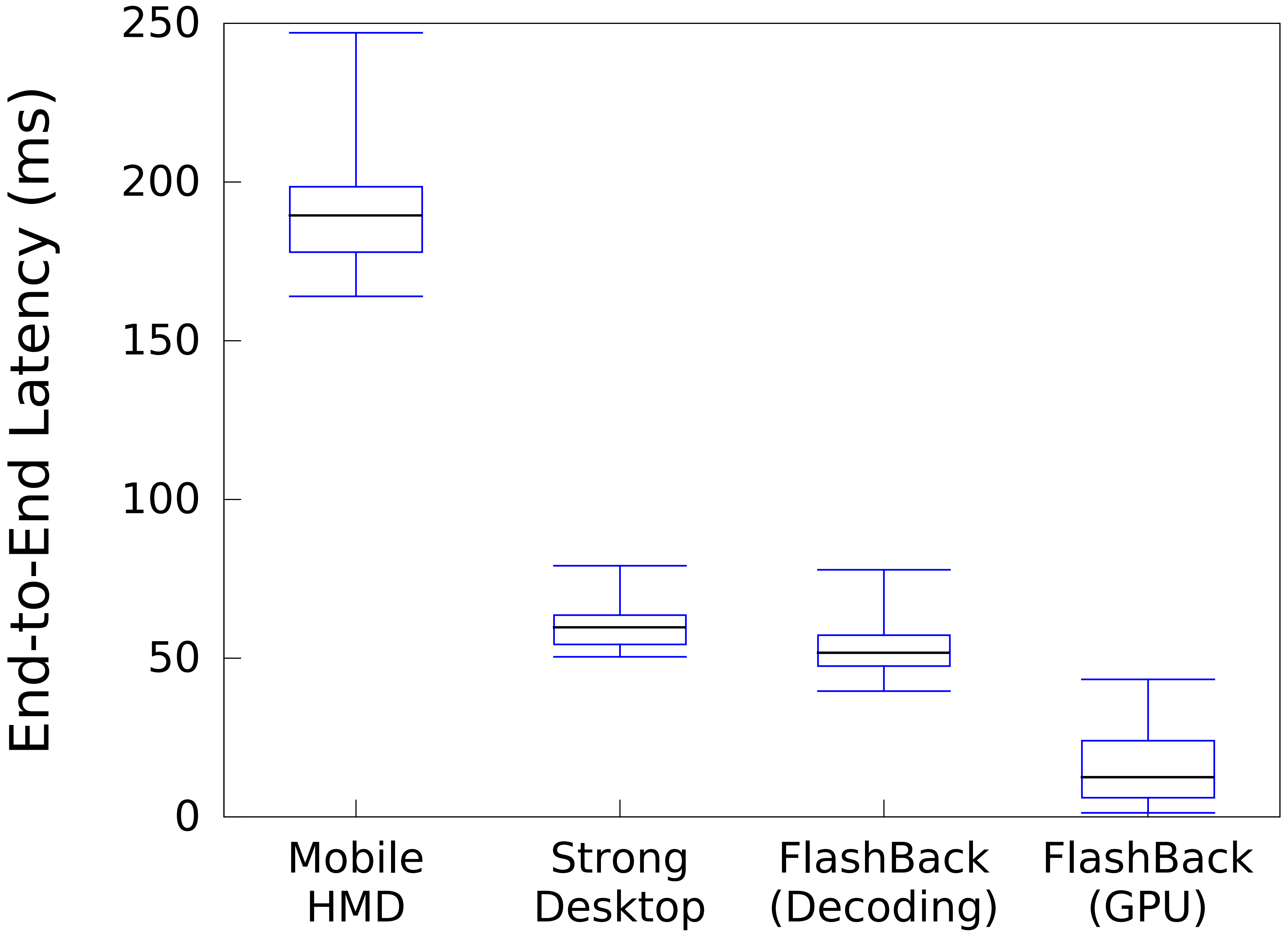
FlashBack

Local Rendering

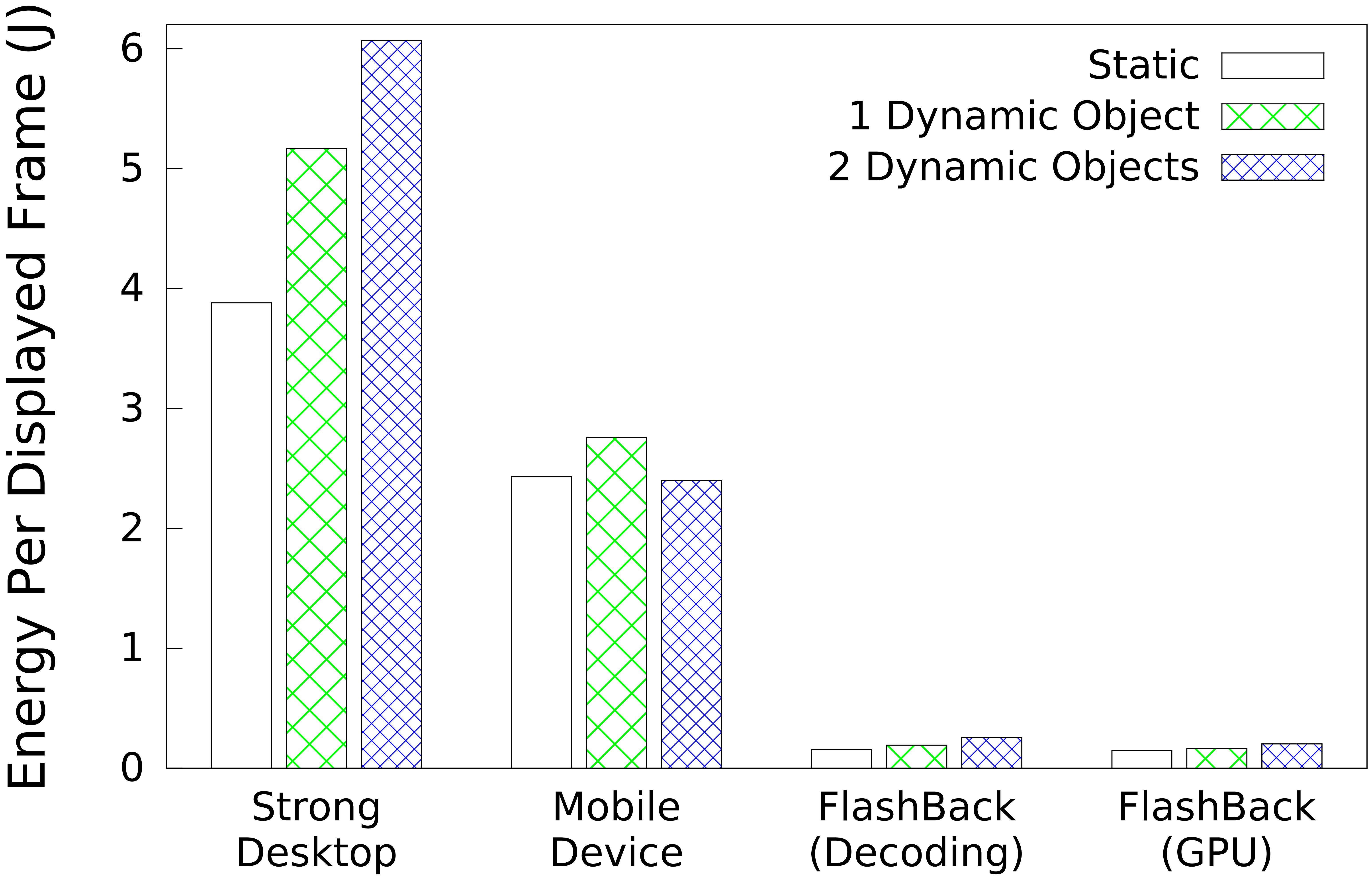
15x higher framerate



8x lower latency



97x more energy efficient

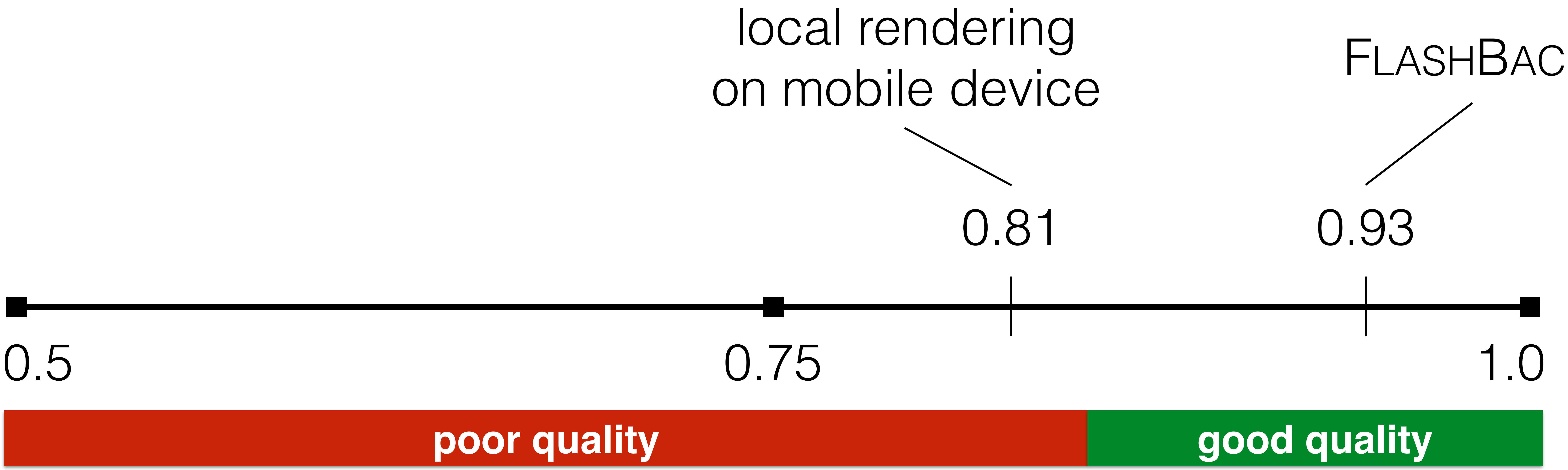


longer battery life

less thermal discomfort

FLASHBACK maintains image quality

- Measure perceived visual quality via SSIM
  - Compares rendered scene against a pristine image



Limitations

- Dynamic object scalability is moderate
  - con: per-pixel megaframe compositing is slow
  - pro: object complexity is irrelevant
- Lighting models are limited
- Restricted by hardware decoder

Related work

precaching	web search results database queries	data types, behavior, and design choices are not applicable to VR domain.
offloading	compute offload wearable AR on Glass rendering prelim. work on HMDs	requires good network connection. latency/quality less demanding. ignores local device storage.
caching objects as rendered images	QuickTime VR reuse past renderings caching with *impostors*	static video playback only. focused on desktop environments. inaccuracies in object representation. very limited dynamic object support.
warping cubemaps	VR address recalculation	requires specialized hardware added to high-end GPUs.

FlashBack in conclusion

- Avoids real-time rendering by pre-generating frames
  - flattens complex VR app behavior into data structures
- Supports static scene and dynamic animated objects

framerate ⬆

latency ⬇

energy ⬇

kevinaboos.web.rice.edu

RICE Unconventional Wisdom

Microsoft Research

Backup Slides

References

[3] D. Lymberopoulos, et al., “Pocketweb: Instant web browsing for mobile devices,” ASPLOS 2012.

[4] D. Barbara, et al., “Sleepers and workaholics: Caching strategies in mobile environments,” SIGMOD 1994.

[5] E. Cuervo, et al., “Maui: Making smartphones last longer with code offload,” MobiSys 2010.

[6] B. Chun, et al., “Clonecloud: Elastic execution between mobile device and cloud,” EuroSys 2011.

[7] M. Gordon, et al., “Comet: Code offload by migrating execution transparently,” OSDI 2012.

[8] K. Ha, et al., “Towards wearable cognitive assistance,” MobiSys 2014.

[9] E. Cuervo, et al., “Kahawai: High-quality mobile gaming using gpu offload,” MobiSys 2015.

[10] Y. Degtyarev, et al., “Demo: Irides, attaining quality, responsiveness, and mobility for VR HMDs,” MobiSys 2015.

[11] S. Chen, “Quicktime VR: An image-based approach to virtual environment navigation,” SIGGRAPH 1995.

[12] G. Schaufler, “Exploiting frame-to-frame coherence in a virtual reality system,” IEEE VR AIS 1996.

[13] G. Schaufler and W. Sturzlinger, “A Three Dimensional Image Cache for Virtual Reality,” CG Forum 1996.

[14] J. Shade, et al., “Hierarchical image caching for accelerated walkthroughs of complex environments,” SIGGRAPH 1996.

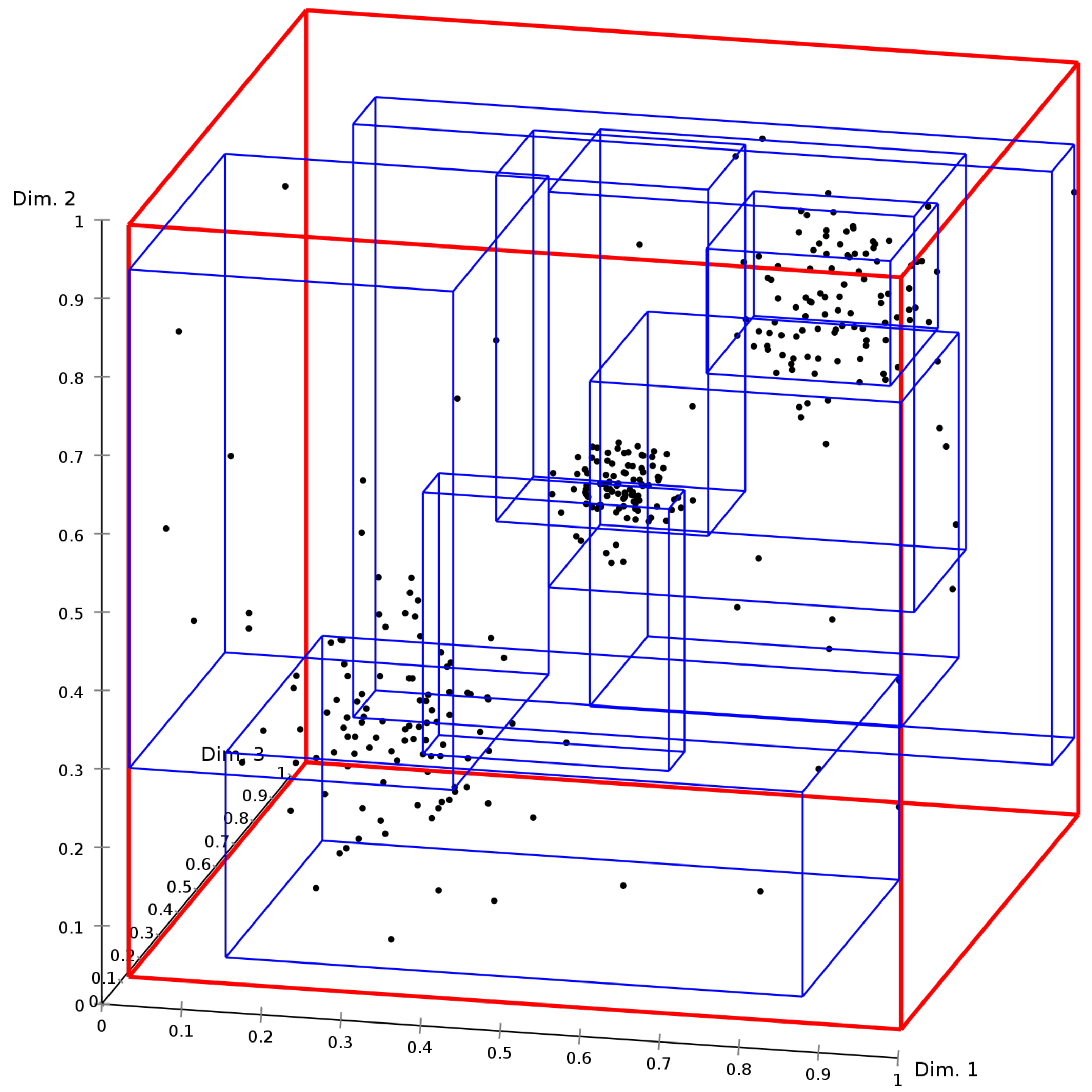
[15] M. Regan and R. Pose, “Priority rendering with a virtual reality address recalculation pipeline,” SIGGRAPH 1994.

[16] M. Regan and R. Pose,” An interactive graphics display architeture,” IEEE VR AIS 1993.

Cache lookup with R-trees

- create minimally-overlapping bounding boxes around 3D points

- characteristics fit our needs
  - good insertion and deletion
  - fast lookup is priority
  - better querying semantics

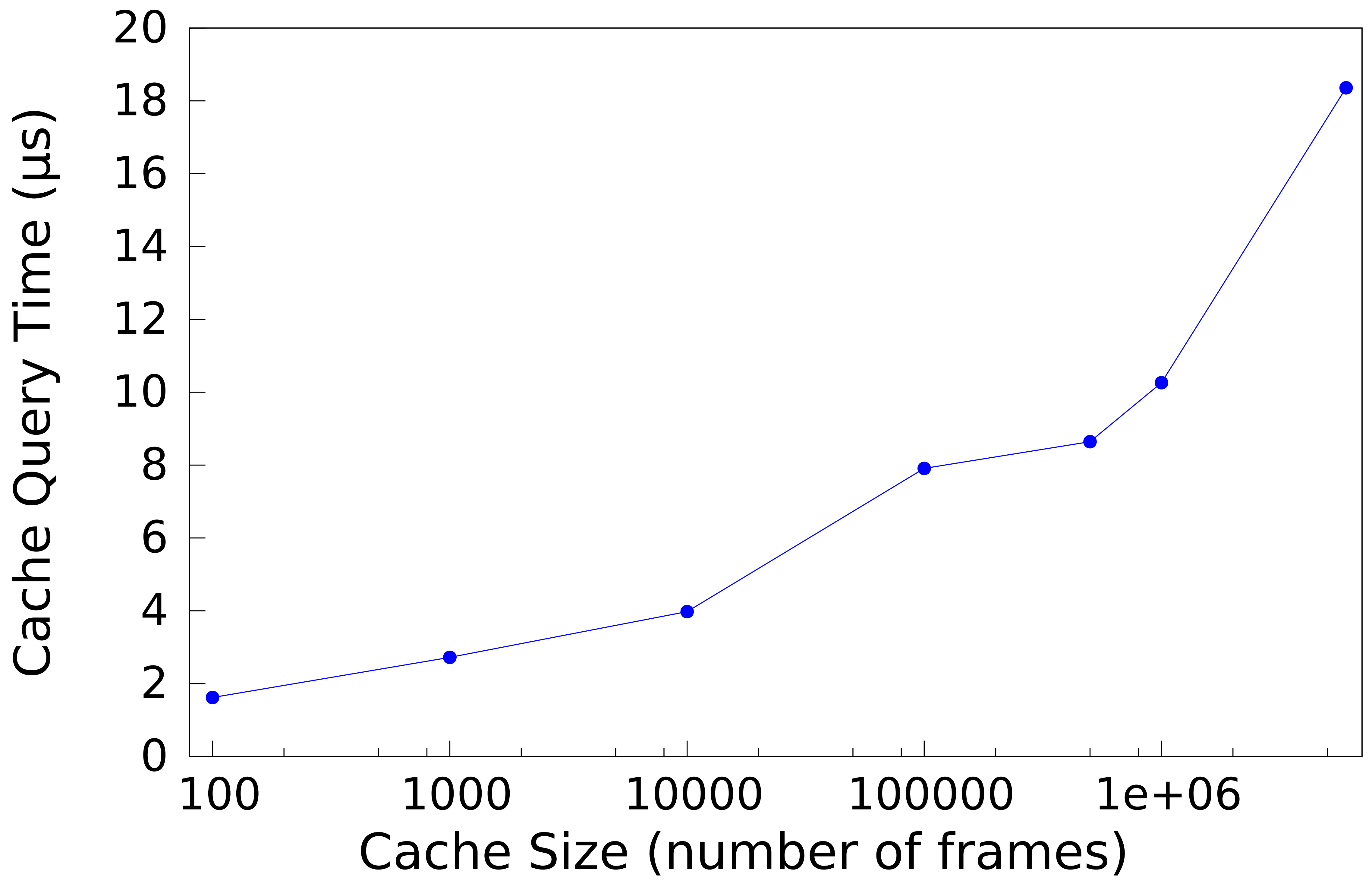


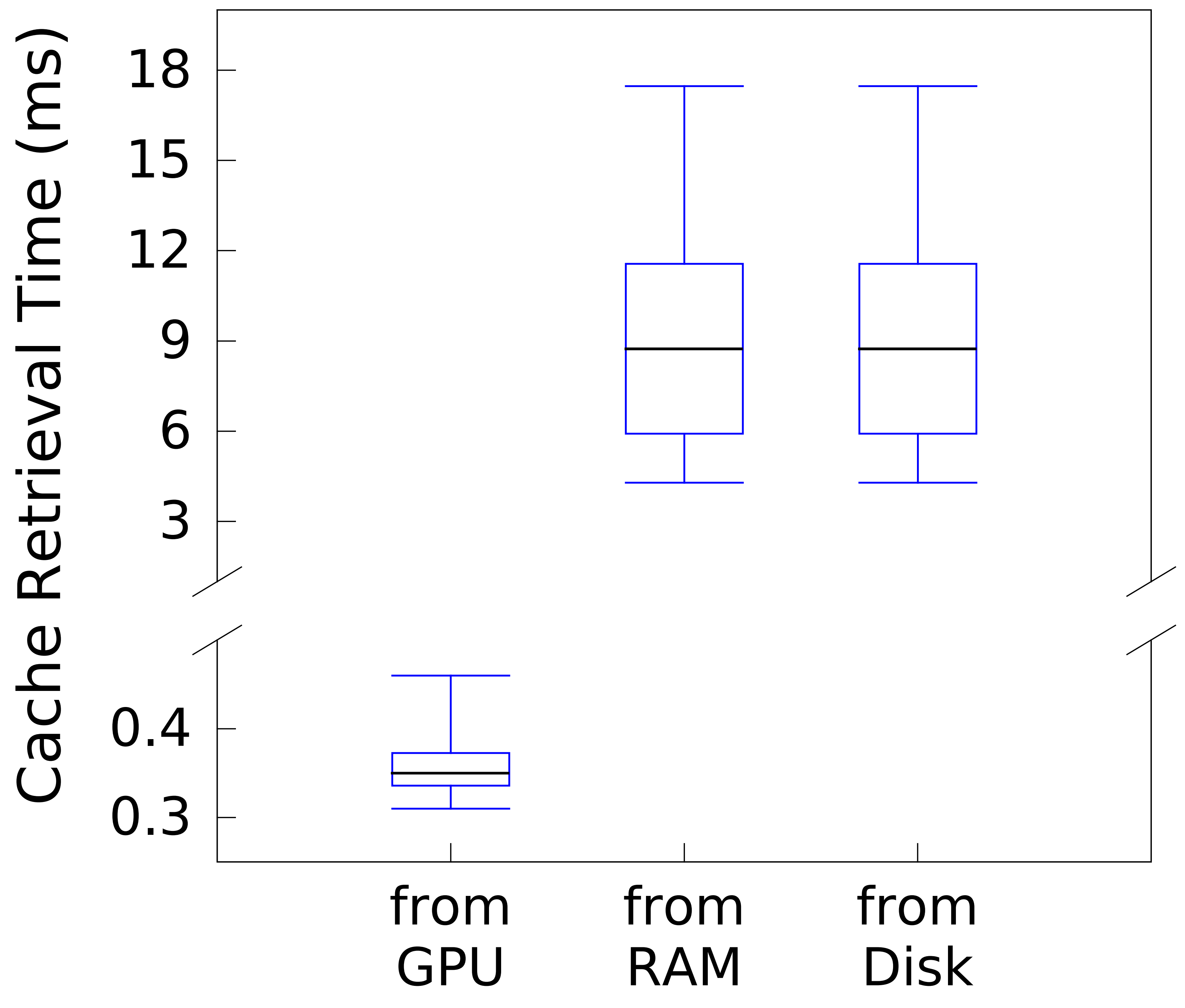
Guttman, A. "R-Trees: A Dynamic Index Structure for Spatial Searching". ACM SIGMOD ‘84.

VR system in a nutshell

- Head-Mounted Display (HMD)
  - Smartphone-class hardware
  - Internal sensors and external trackers
- Combine sensor readings → player pose
  - 3D position
  - 3D orientation

Microbenchmarks





Typical cache sizes (uncompressed)

car interior	115 MB
bedroom	730 MB
living room	2.8 GB
two-story house	8.7 GB
basketball arena	29 GB
Viking Village	54 GB

can be compressed using video codec for efficient deployment