FLASHBACK – Immersive Virtual Reality on Mobile Devices via Rendering Memoization

Kevin Boos
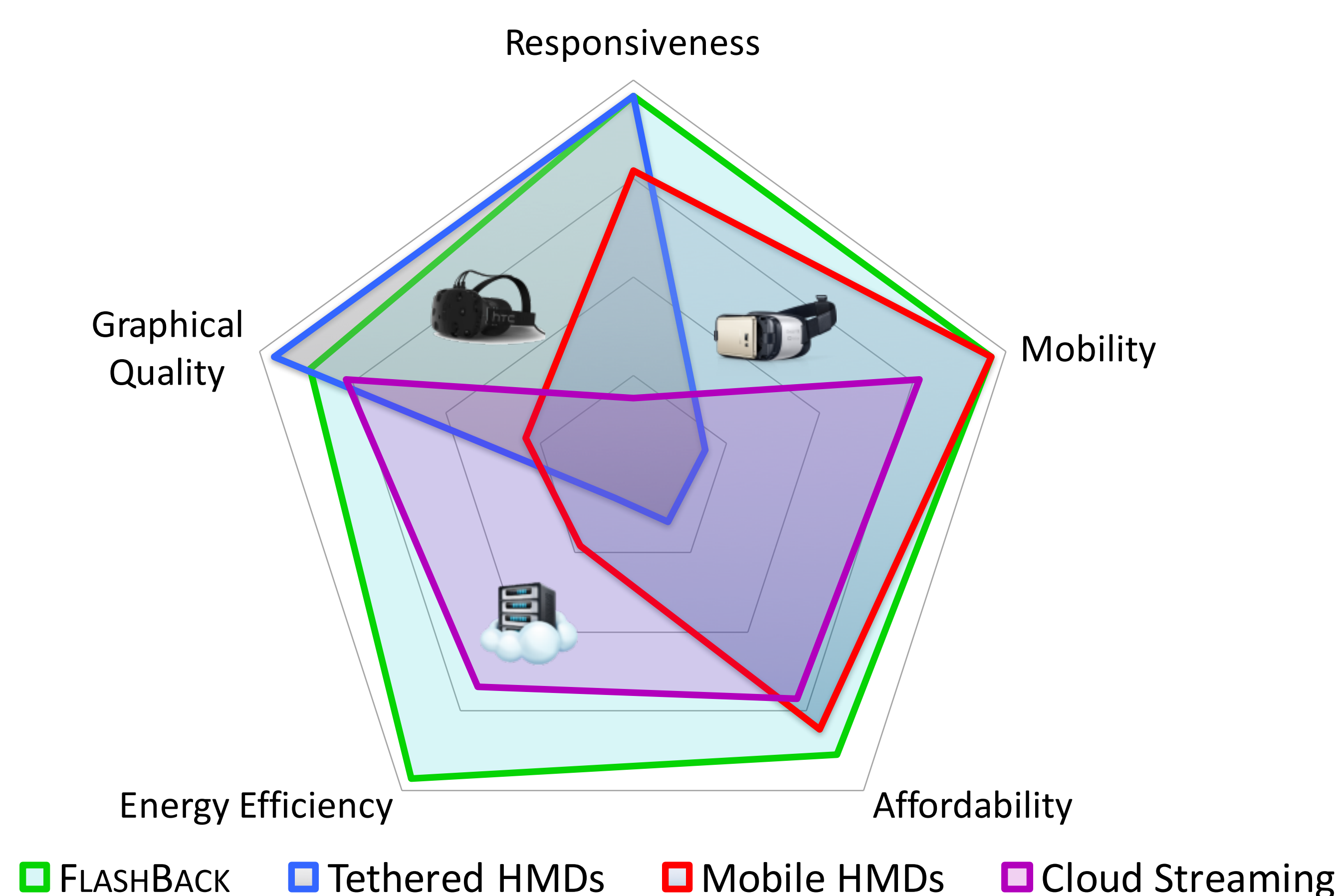David Chu
Eduardo Cuervo

VR is highly demanding

- Complex graphics: photo-real, wide FOV
- Low latency: under 25ms
- High framerate: 60+ FPS

Tradeoffs in VR systems

- Tethered HMDs: powerful, immobile, $
- Mobile HMDs: portable, weak, 🔋
- Cloud streaming: bad latency, needs 📶

☐ FLASHBACK ☐ Tethered HMDs ☐ Mobile HMDs ☐ Cloud Streaming

VR system hardware

- Head-Mounted Display consists of smartphone-class hardware + sensors
- Sensor readings generate a player *pose*
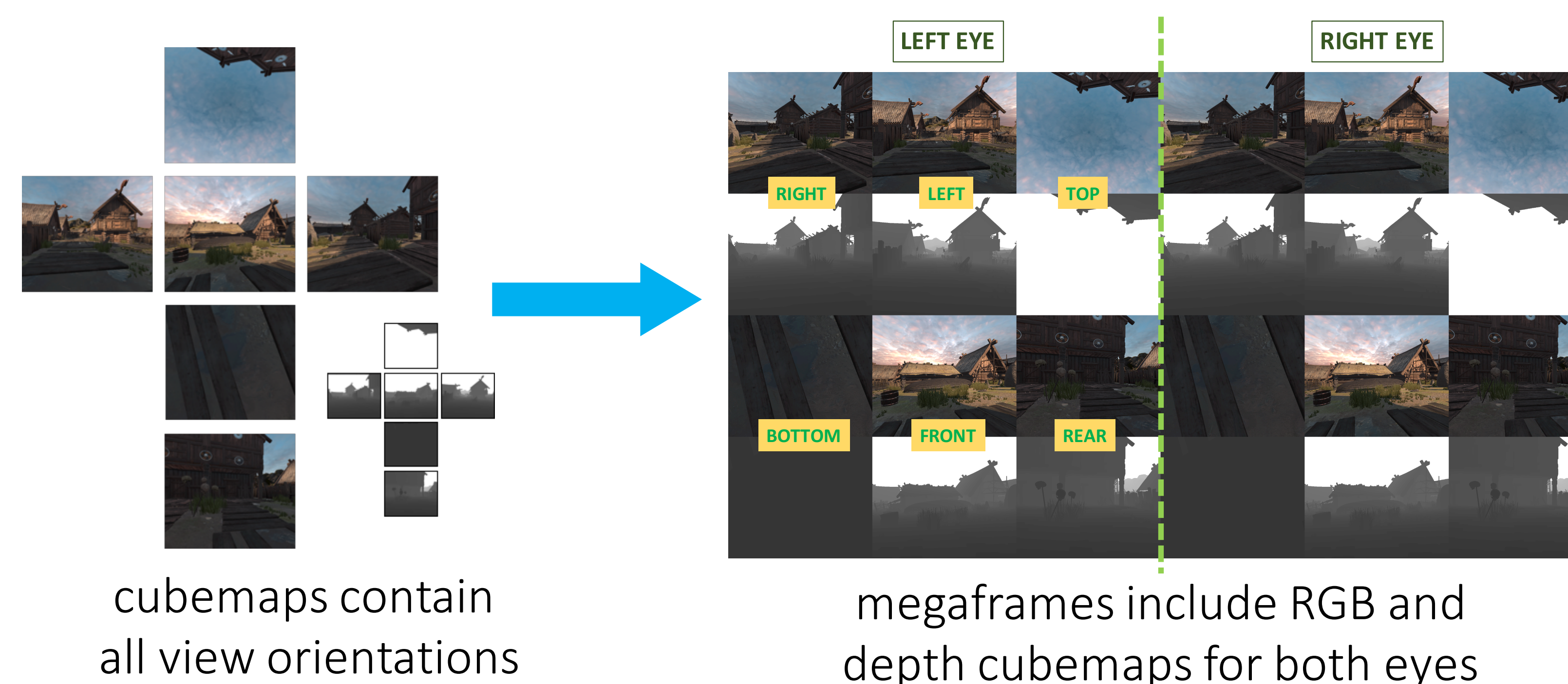 - 3D position
  - 3D orientation

Insight: leverage local storage

- Mobile **storage** is increasingly cheap, abundant, low-power, and underutilized
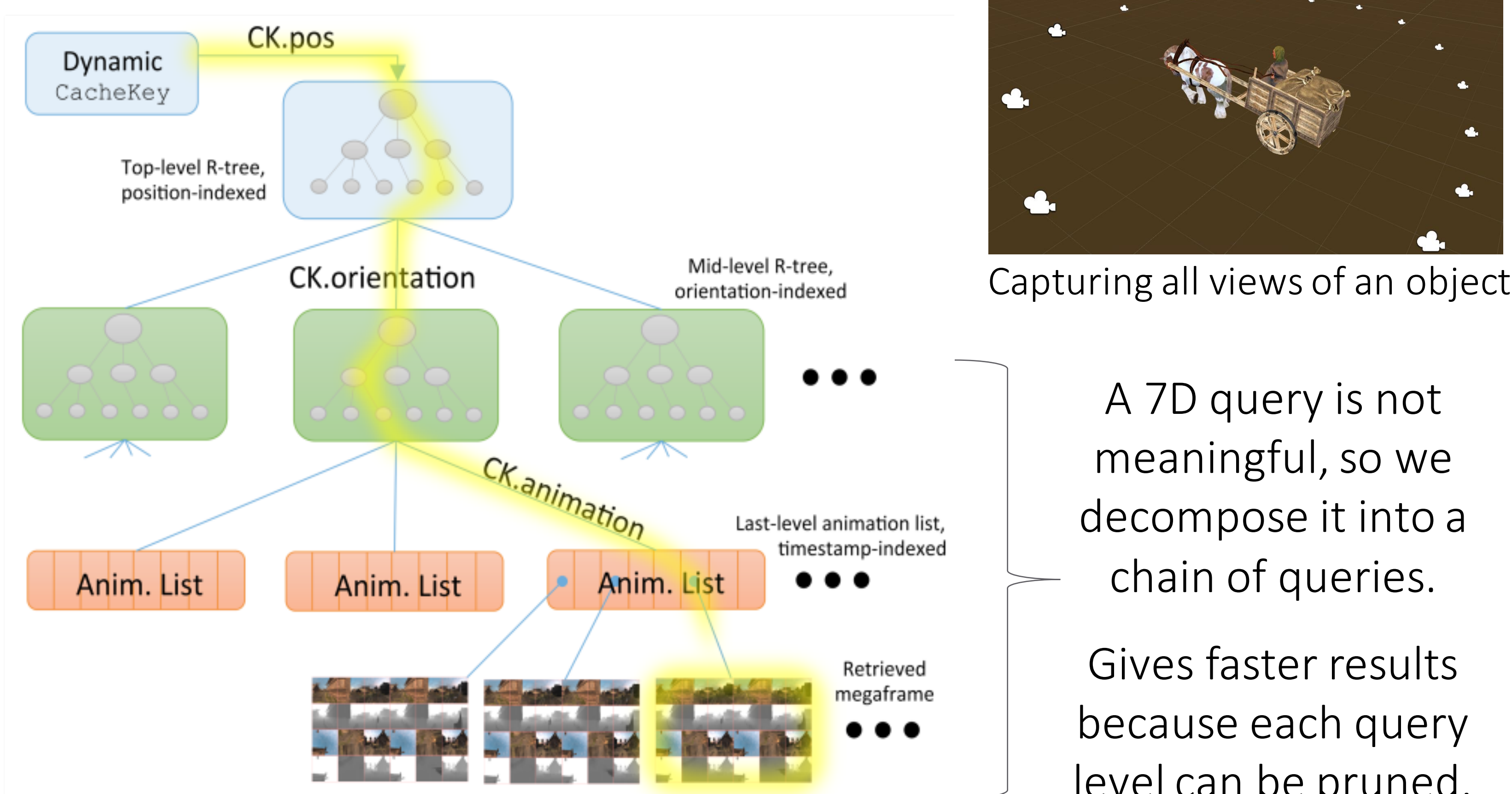- Mobile **graphics** processing is restricted due to thermal and energy constraints

FlashBack Pre-Rendering [offline]

1. Quantize virtual environment into chunks
   - Pruning infinite input space geometrically
2. Render *cubemap* at each reachable position (chunk)
3. Save cubemaps into a position-tagged *megaframe*

LEFT EYE   RIGHT EYE

RIGHT LEFT TOP
BOTTOM FRONT REAR

cubemaps contain all view orientations

megaframes include RGB and depth cubemaps for both eyes
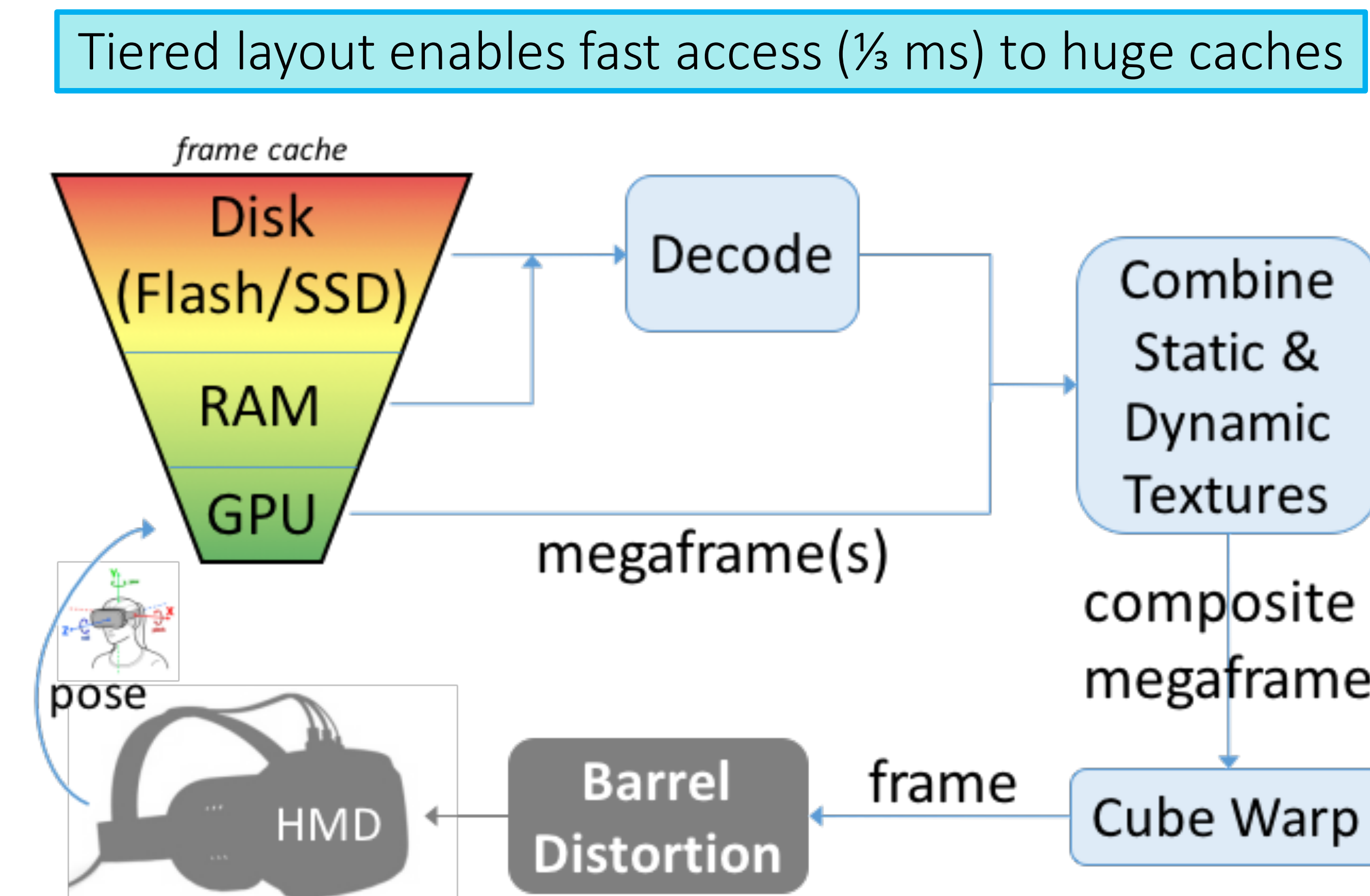
Handling Dynamic Objects

- Extend static scene caching: 1 cache per object
- Generate *pose trace* file
  - Defines object movement & animation sequence
- Index dynamic megaframe cache with 7D key:
  - Relative player 3D position
  - Dynamic object's 3D orientation
  - Dynamic object's animation phase

Dynamic CacheKey

CK.pos

Top-level R-tree, position-indexed

CK.orientation

Mid-level R-tree, orientation-indexed

CK.animation

Anim. List   Anim. List   Anim. List

Last-level animation list, timestamp-indexed

Retrieved megaframe

Capturing all views of an object

A 7D query is not meaningful, so we decompose it into a chain of queries.

Gives faster results because each query level can be pruned.

FLASHBACK Live Playback [online]

1. Query caches using spatial R-tree index for quick nearest-neighbor search
2. Retrieve (and decode) closest megaframe
3. Reconstruct final frame via graphical warp
   - Quick IBR approximation on cache miss

Tiered layout enables fast access (⅓ ms) to huge caches

frame cache
Disk (Flash/SSD)
RAM
GPU

Decode

megaframe(s)

Combine Static & Dynamic Textures

composite megaframe

Cube Warp

pose

HMD

Barrel Distortion ← frame

Evaluation vs. Mobile Rendering

- 8x higher framerate
- 15x lower latency
- 97x more energy efficient per frame