Towards a Benchmark Suite for Augmented and Virtual Reality

Muhammad Huzaifa, PhD Candidate
Sarita Adve, PI
University of Illinois at Urbana-Champaign

adacenter.org  @ADA_Center

This work is supported by the Semiconductor Research Corporation (SRC) and DARPA
AR/VR PPA Recap

Samsung Odyssey
- 2.3 Mpx/eye
- 90 Hz
- 110° FoV
- Latency?
- 250 W
- 500 mm²

200x perf!

250x power!

5x area!

Ideal headset
- 100 Mpx/eye
- 144 Hz
- 175° FoV
- 20 ms or less
- 1 W
- 100 mm²

Sarita Adve
Why AR/VR?

• Up and coming killer application

• Challenges span entire system

• Great driver for hardware specialization
  • Several AR/VR kernels are shared across domains
Challenges

• State-of-the-art closely guarded by industry

• No open-source benchmark suite

• No open-source cycle-accurate graphics simulator

Where do we start?

Sarita Adve
Approach

• Final Goal: End-to-end “application” emulating VR pipeline
  • Cannot use Unity and Oculus SDK – blackboxes
  • Developed aspirational VR pipeline to capture key components and key system interactions

- Challenges: Many domains, difficult to integrate multiple code bases, modeling graphics and system interactions
• Step 1: Collect state-of-the-art codes for each component
• Step 1: Collect state-of-the-art codes for each component
• Step 2: Analyze each component in isolation
  • This talk: SLAM
• Step 3: Create intermediate mini VR pipelines to understand interactions
  • This talk: SLAM ➔ Renderer ➔ Adaptive Display
• Step 4: Use analysis to drive scalable hardware specialization
SLAM

• Simultaneous Localization and Mapping
  • What does the world look like?
  • Where am I in the world?

• “Frontend” of AR
  • 15k LoC, several complex kernels
  • Solving SLAM == consumer AR glasses (VentureBeat)

• Visual SLAM uses RGB-D camera
  • Dense methods
  • Sparse methods
ElasticFusion

• State-of-the-art dense visual SLAM

• Directly uses pixel intensities and depths for tracking

• Dense reconstruction

• Computationally expensive
ElasticFusion Findings

• ~25 ms for 640x480 @ 30fps video on **Titan Xp**

• Processing time split equally between tracking (CUDA) and mapping (OpenGL)

• Insights
  • ~50% of tracking time spent on data-structure reductions
  • ~30% of tracking time spent on small memcpys
  • Cannot use OpenGL textures in CUDA

• Optimizations: Unified shared memory, flexible coherence, data layout transformations

*Sarita Adve*
ORB-SLAM2

• State-of-the-art sparse visual SLAM

• Performs feature extraction and tracks keypoints

• Superb tracking accuracy

• No vanilla dense reconstruction
ORB-SLAM2 Findings

• Sparse method *much more* scene dependent than dense method
  • Critical to select representative scenes
  • Critical to understand scene-compute relationship

• Real time on desktop-grade CPU in isolation

• Computer vision frontend most expensive computation

• Need dense reconstruction for apples to apples comparison
Mini VR Pipeline

Input
- Head tracking
- Eye tracking
- SLAM
- Networking
- Controller input

Rendering
- Stereoscopic rendering
- Foveated rendering
- 3D audio
- 3D video

Post-processing
- Contrast preservation
- Chromatic aberration correction
- Lens distortion correction
- Temporal anti-aliasing

Optimizations
- Time warp
- Space warp

Output
- Adaptive display

Sarita Adve
Mini VR Pipeline Findings

• No longer real time on Titan Xp at Vive’s resolution (2160x1200)
  • ElasticFusion (20 ms) + Renderer (5 ms) + Hologram (18 ms) = 43 ms
  • Deadline: 33 ms

• Common compute pattern: reductions
  • ~50% of CUDA execution time in ElasticFusion; ~63% in Hologram
  • Reductions are on custom data-structures!

• But different data layouts for each kernel

• Need communication specialization techniques such as Spandex
Vanilla ElasticFusion

Sarita Adve
ElasticFusion + Hologram

Sarita Adve
What’s Next?

• Finish application
  • Aberration shaders between ElasticFusion and hologram *(easy medium)*
  • Head & eye tracking + time & space warping combo *(medium)*
  • Replace simple renderer with stereoscopic & foveated renderers *(hard)*

• Analysis
  • Profile at each intermediate step
  • Repeat experiments on embedded platform; e.g., NVIDIA Jetson TX2

• Use analysis to guide accelerator and memory system development
  • Spandex is a promising platform: unified shared memory, flexible coherence and communication

• Release application!

Sarita Adve