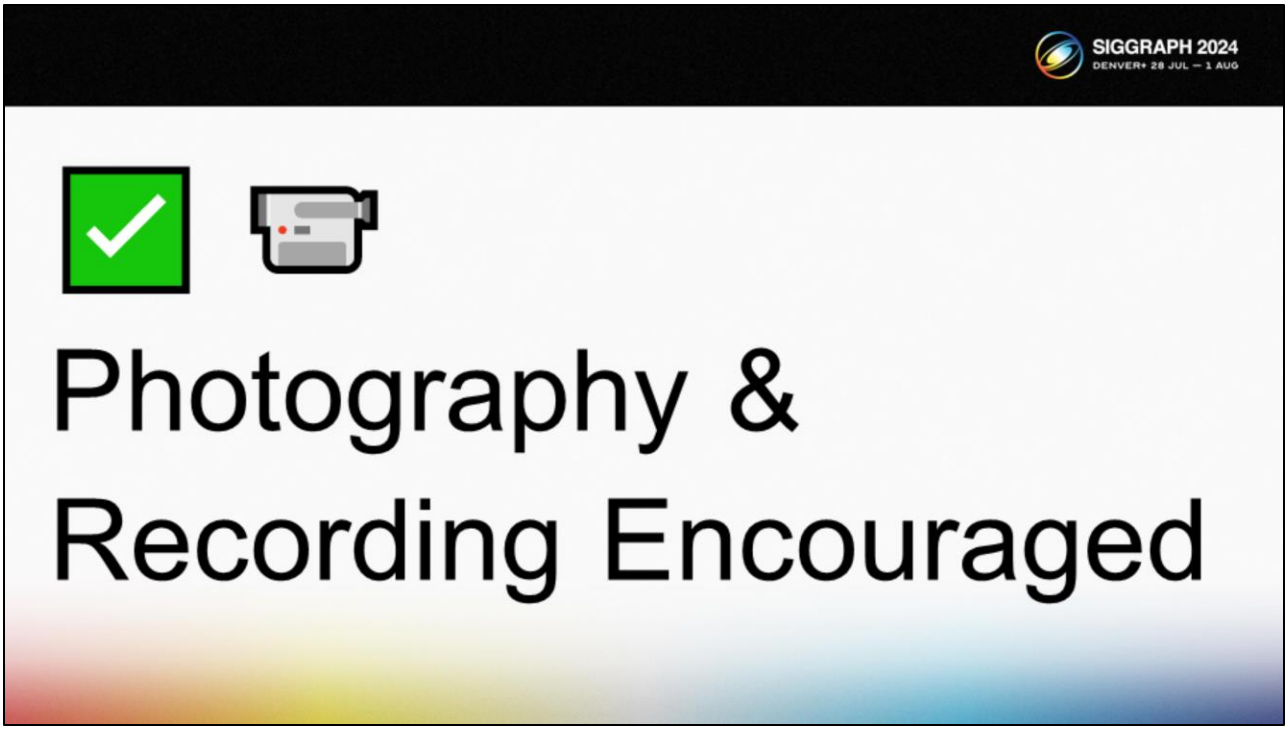
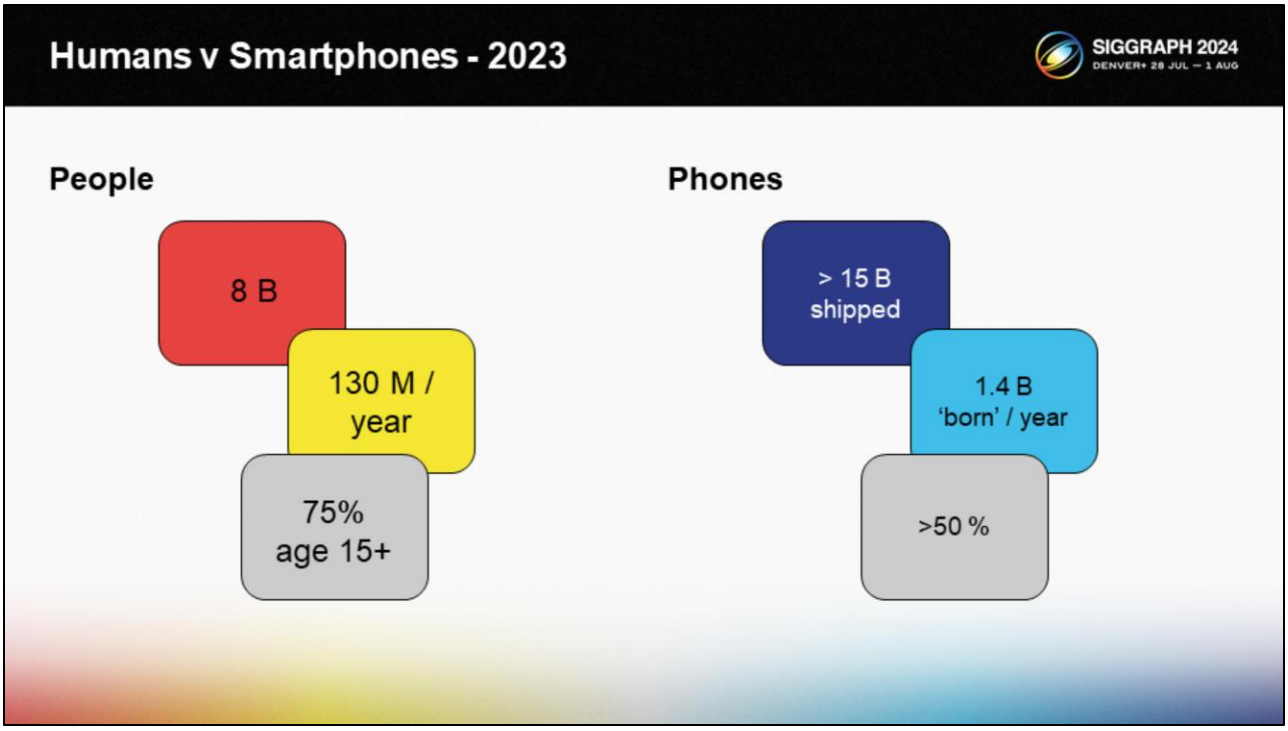
Welcome to the SIGGRAPH 2024 edition of Moving Mobile Graphics. This is the 5th instance of this course over the past 10 years. My name is Jesse Barker, and I work on shader systems at Unity Technologies. I've been honored to be a part of bringing this content to you for the past couple of editions, and I want to call out my partner, Sam Martin of arm, who started us down this path in 2015.

While at Advances in Real-Time on Tuesday, I was reminded of something that I have come to believe over my years of involvement at SIGGRAPH: that the story, or the narrative is the key. Anything that detracts from the storyteller's vision is just in the way. Hopefully, we are bringing you content that facilitates compelling narratives, rather than detracting from it.

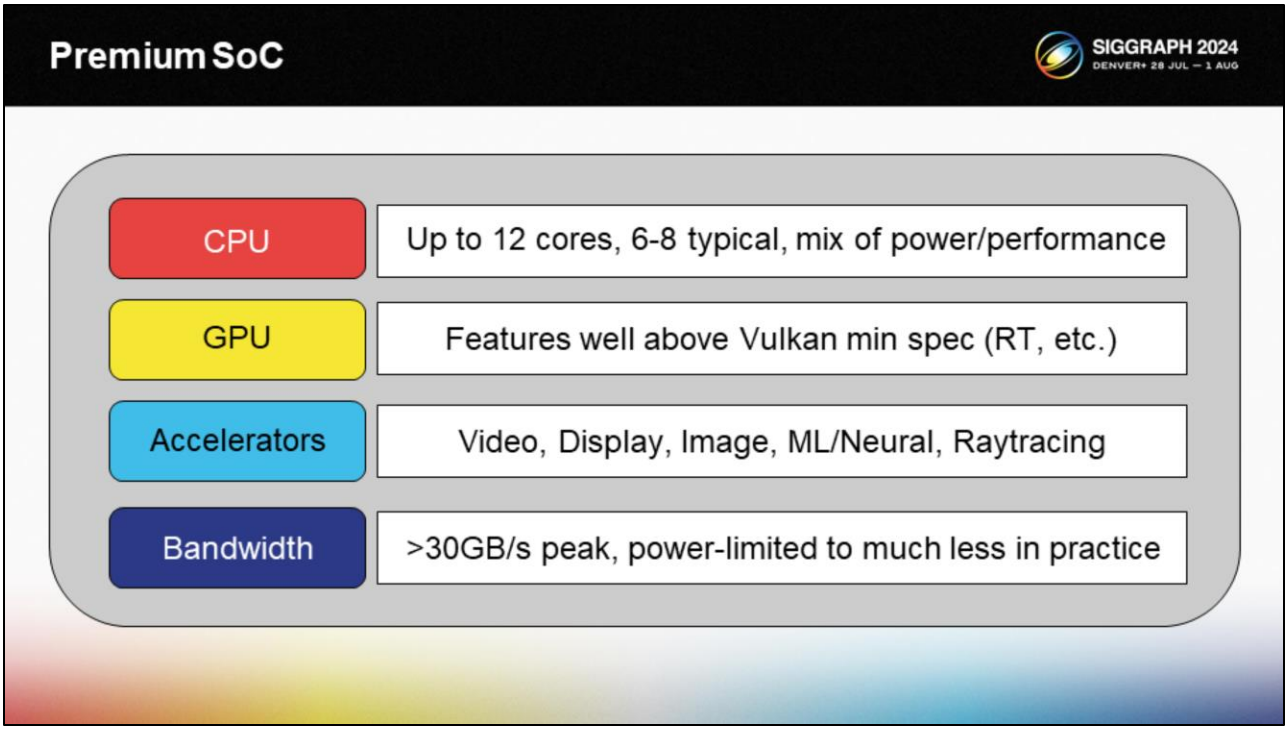
As I was reviewing those previous course instances to prepare for today, I was struck both by how much has changed, and by how little has changed. I have always believed that constraints provide a crucible for creativity. Embedded and mobile device platforms fall firmly into this category.



This session is being recorded and livestreamed, but if you feel you need to take photos, feel free.



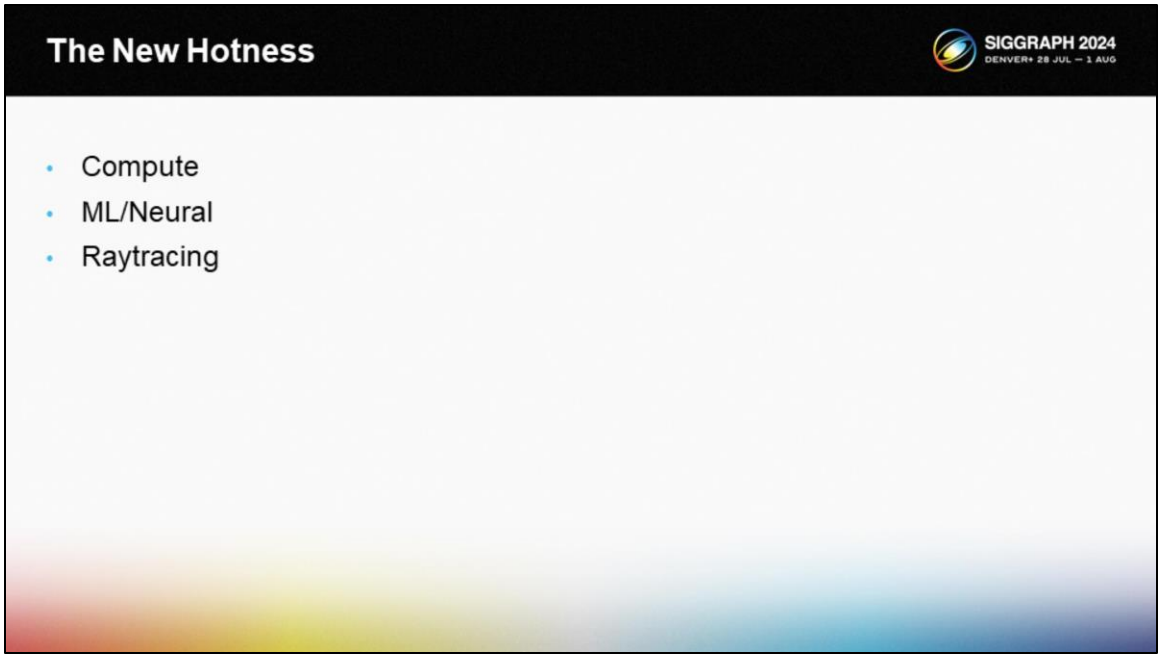
This update is more about saturation rather than raw numbers. Possibly a greater reliance on smartphones, possibly introduction into both younger and older age groups has pushed the saturation numbers up (maybe a combination of both). We list the 15+ age group here because we have been using that in previous instances of the course. However, at least anecdotally, much younger school-age kids either have one because "all the cool kids do", because the school system requires it, or maybe because parents know they can keep track of where their kids are as long as they have them (and they always do, don't they?). statista.com shows shipments at 1.4B last year (back up from the previous few years), and total shipments well north of 15B. The interesting thing here is that a much higher percentage of the population have smartphones.



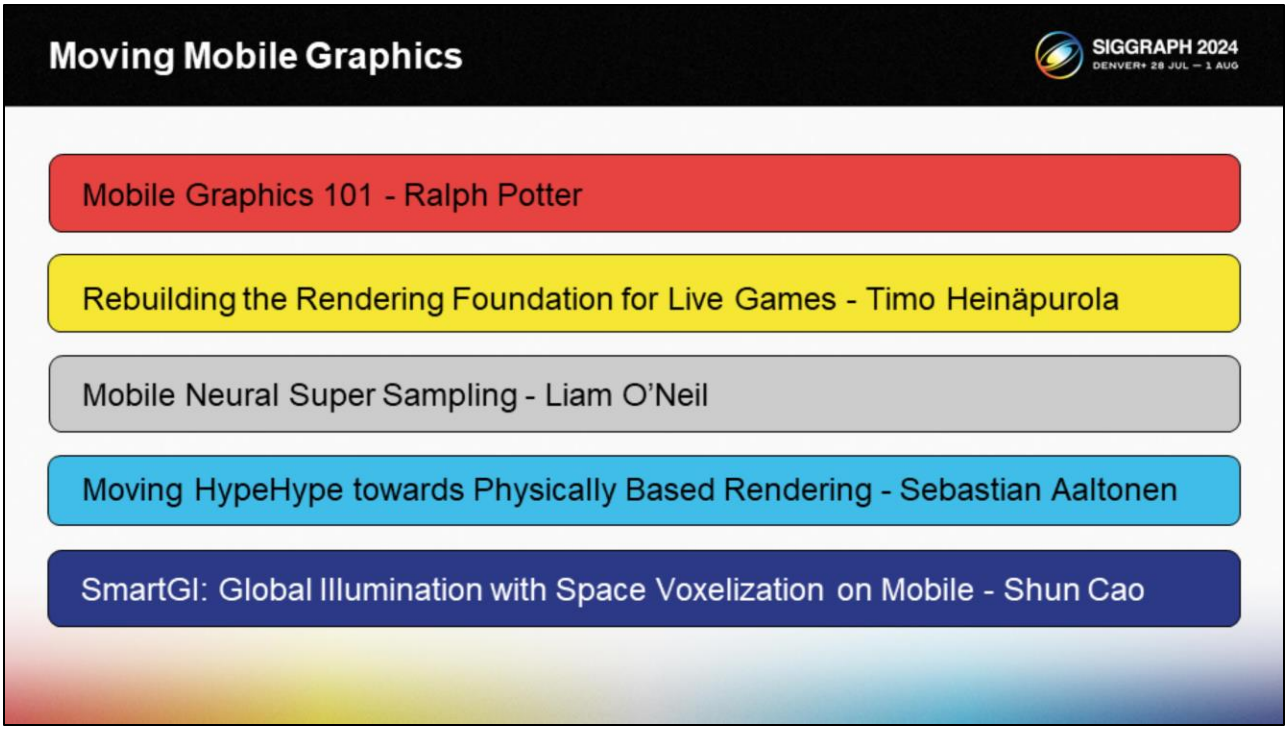
On the surface, not much has changed here, And as always, the key limiting factor is thermal dissipation. The key defining factors for a mobile device (at least smartphones and tablets) are battery power and passive cooling. The form factor (typically 5-6") limits the thermal dissipation, which in turn limits the power draw, typically ~5W total. Bandwidth is about 80mW per GB/s *, so peak numbers are very far from typical sustained numbers. Advances in chip process buy us some wiggle room, but we are still pretty constrained.

We get some new accelerators, but they come with some responsibility from developers.

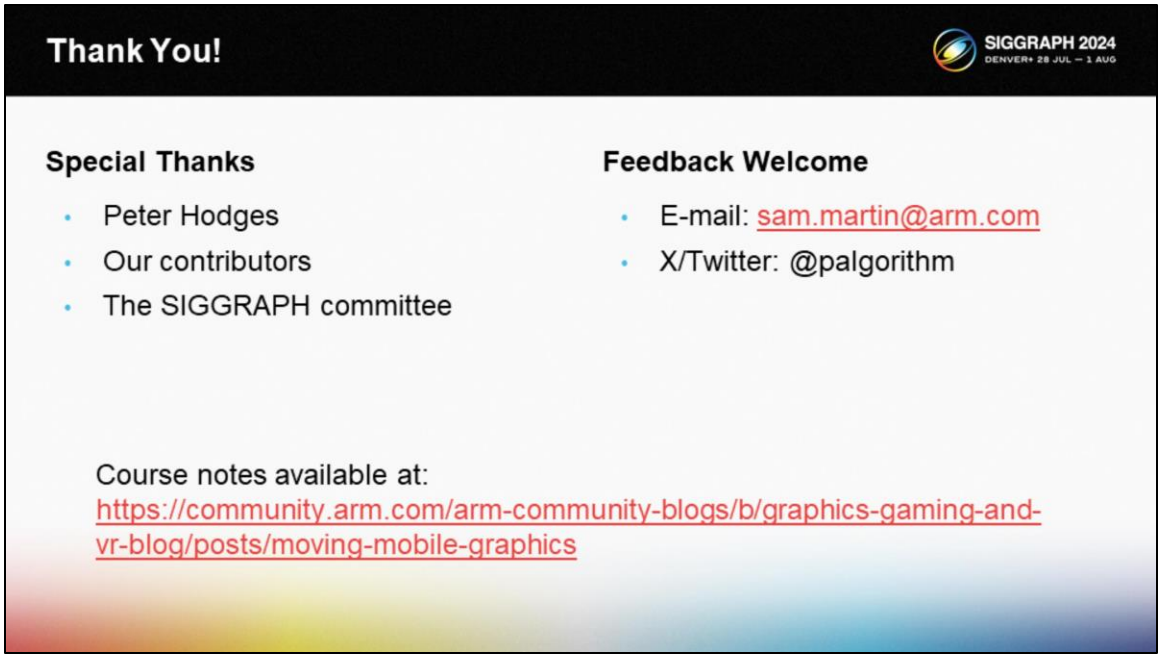
*Source: Peter Harris' Vulkanised talk



ML and raytracing get some new specialized accelerators, which is great. We know from other areas (the Vulkan Roadmap Profile update from Khronos, the intro to real-time raytracing course earlier this week) that these feature sets are now available (more or less) across the board. Compute has been with us a long time now (GLES 3.1/Vulkan 1.0 minspec), so why is it listed here? In practice, compute has not been widely utilized in production applications, however, advances in the feature set within compute shaders are shifting that significantly (see the latest Vulkan roadmap profiles). While these features are fully supported, they are not always able to take advantage of some of the bandwidth saving measures provided by the GPU, so they are going to require some careful thought, and a bit of hard work to use well without blowing out our power and thermal budgets.



We will have a short Q&A following Liam's talk, and a 15 minute break following that. There will be another short Q&A after the final talk



As always, we would love to hear from you on how we've done, and whether there are topics you might like to see covered in future versions of this course.

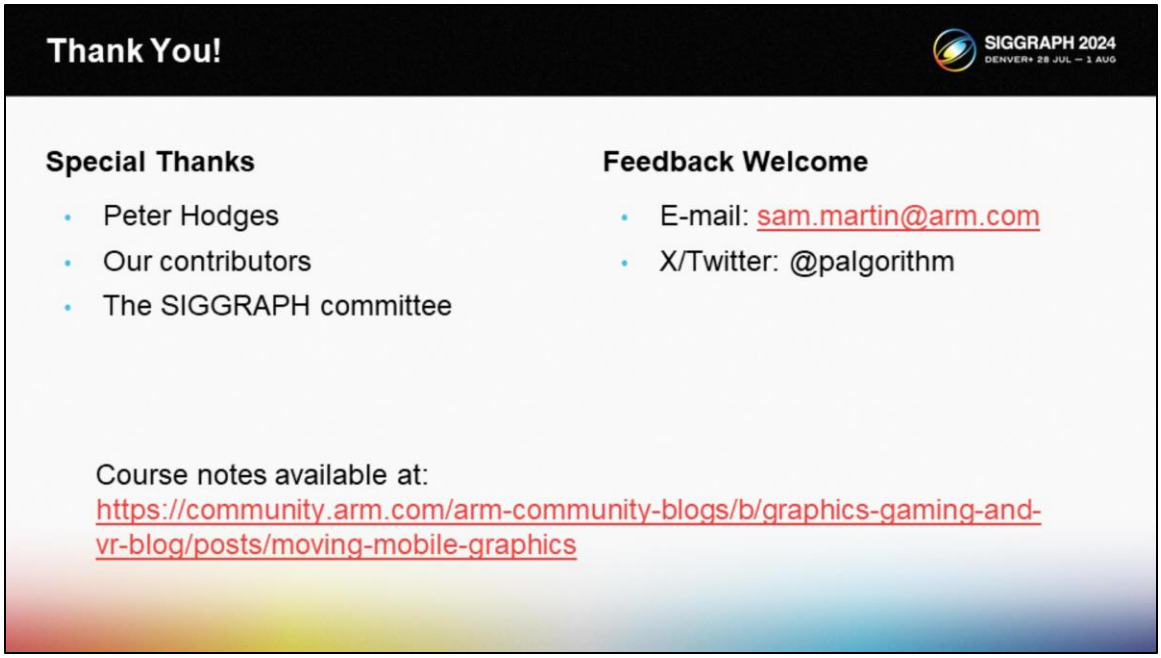
Ralph Potter (Samsung R&D Institute (UK)) is the lead Khronos engineer at Samsung R&D Institute (UK) his interests include computer graphics, GPU computing, compilers and programming models. He is also Samsung’s principal representative within the Khronos Group and sits on its board of directors.

Timo currently leads the development of Supercell's in-house game engine that is used to develop games such as Clash of Clans, Clash Royale and Brawl Stars. Before joining Supercell he worked at Bugbear Entertainment and Next Games on games including Wreckfest and The Walking Dead: No Man's Land.

Liam O'Neil (Arm) is an R&D engineer at Arm working with expertise in optimising machine learning workloads for computer graphics and image processing.

Sebastian Aaltonen (HypeHype Inc.) has over 20 years of experience in graphics rendering technology. His main focus areas are engine architecture, low level rendering APIs, performance optimization and GPU compute. Sebastian was pioneering GPU-driven rendering development at Ubisoft and distance field ray-tracing at Second Order (Claybook). He also led Unity's DOTS rendering team until he joined HypeHype to rebuild their mobile rendering technology.

Shun Cao (R&D center, Tencent Games) is Head of Engine Technology. He received a bachelor's degree in software engineering from Beijing University of Posts and Telecommunications. He is an expert engineer and tech lead at the R&D center of Tencent Games. Shun has a broad range of research interests with a particular focus on global illumination and animation simulation systems. In the past few years, he has been leading the development of various GI solutions, such as the distributed offline light baking tool Dawn and real-time mobile GI solution SmartGI, which are widely used by commercial projects at Tencent Games.



As always, we would love to hear from you on how we've done, and whether there are topics you might like to see covered in future versions of this course.