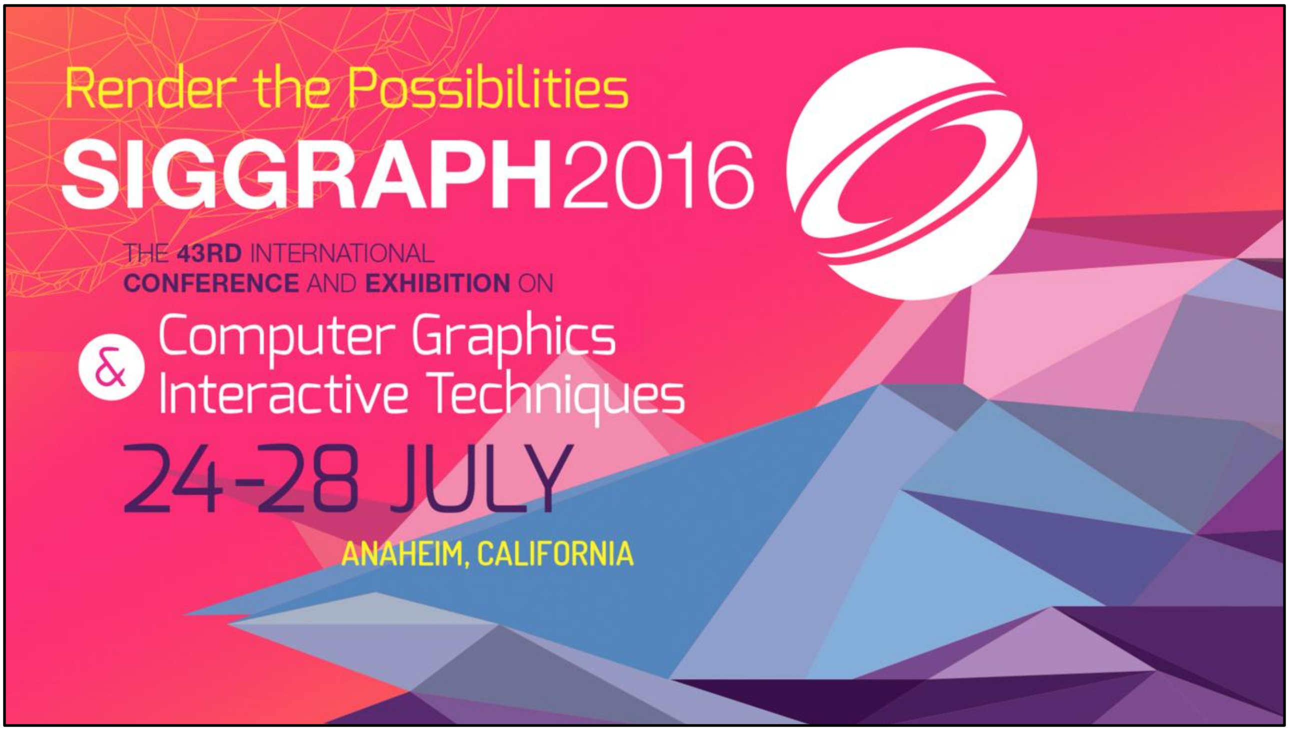

Render the Possibilities
SIGGRAPH 2016
THE 43RD INTERNATIONAL
CONFERENCE AND EXHIBITION ON
Computer Graphics
& Interactive Techniques
24-28 JULY
ANAHEIM, CALIFORNIA

Hi! I'm Sam Martin. I'm a Graphics Architect at ARM and I'd like to welcome you to the 2016 edition of "Moving Mobile Graphics" with all new content.

I love SIGGRAPH and been coming here for as long as I've been involved in graphics. A good number of years ago now, I also became completely captivated by all things mobile which ultimately led me to my current job. This course lives in the intersection of the things I love, and aims to offer a (modest) spotlight to recent advances in mobile graphics. I hope it will not only inform you, but inspire you to develop new mobile graphics innovations.

Taken separately, "mobile" and "graphics" are mature and well advanced things. The combination offers new challenges to work on, and a huge opportunity to impact people's lives, but as a research topic perhaps still needs a bit of TLC.

My starting point is that "mobile" is a technology like no other…

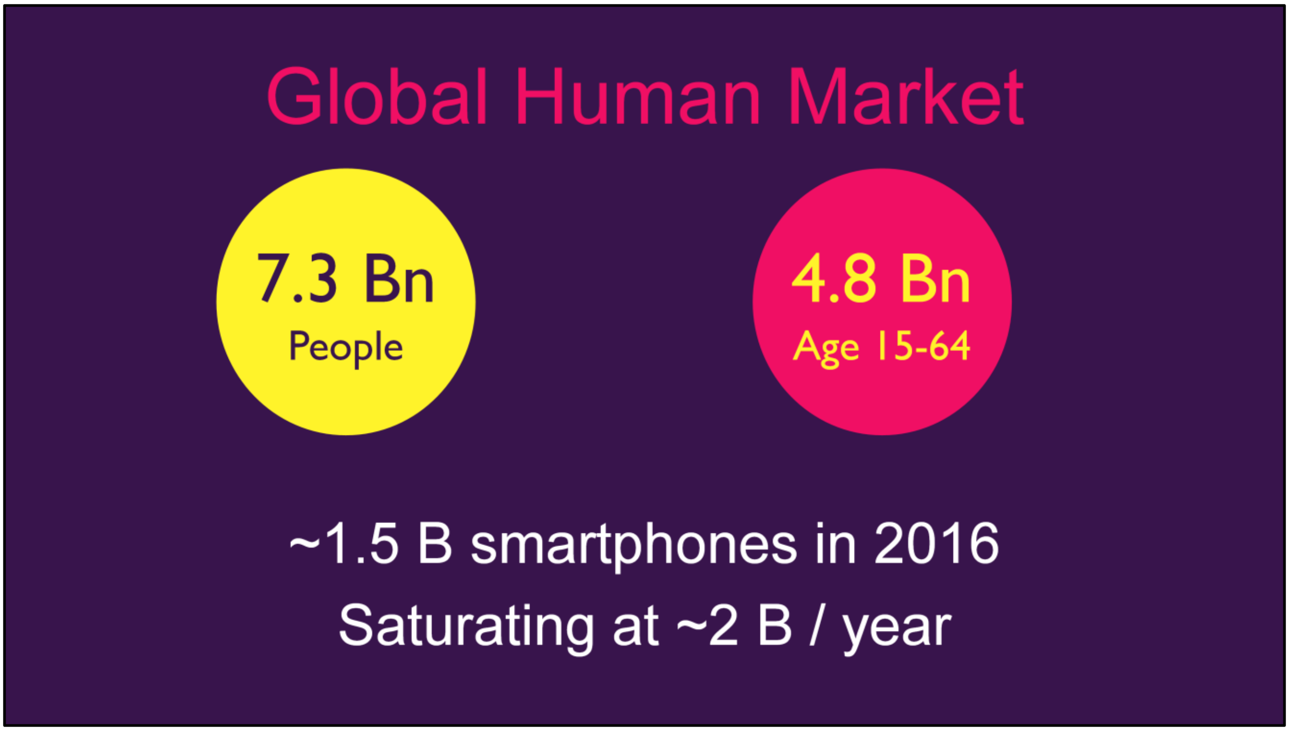
"The smartphone is the defining technology of the age"

The Economist

It's importance is such that the Economist chose to call the smartphone "the defining technology of the age".

To be honest it's hard to even describe it as a technology, as doing so understates the huge amount of innovation and effort that has gone into producing the modern day smart phone.

It is the product of a huge number of people, pushing on as many technical boundaries as possible simultaneously. From innovations in physical engineering and supply line logistics through to new types of stores and a raft of industries that sit on top of the smartphone - with the hw+sw graphics technology bit tucked somewhere in the middle of that stack.



The reason for the astonishing scale of the mobile industry is that it is a market like no other. It is a market the size of mankind.

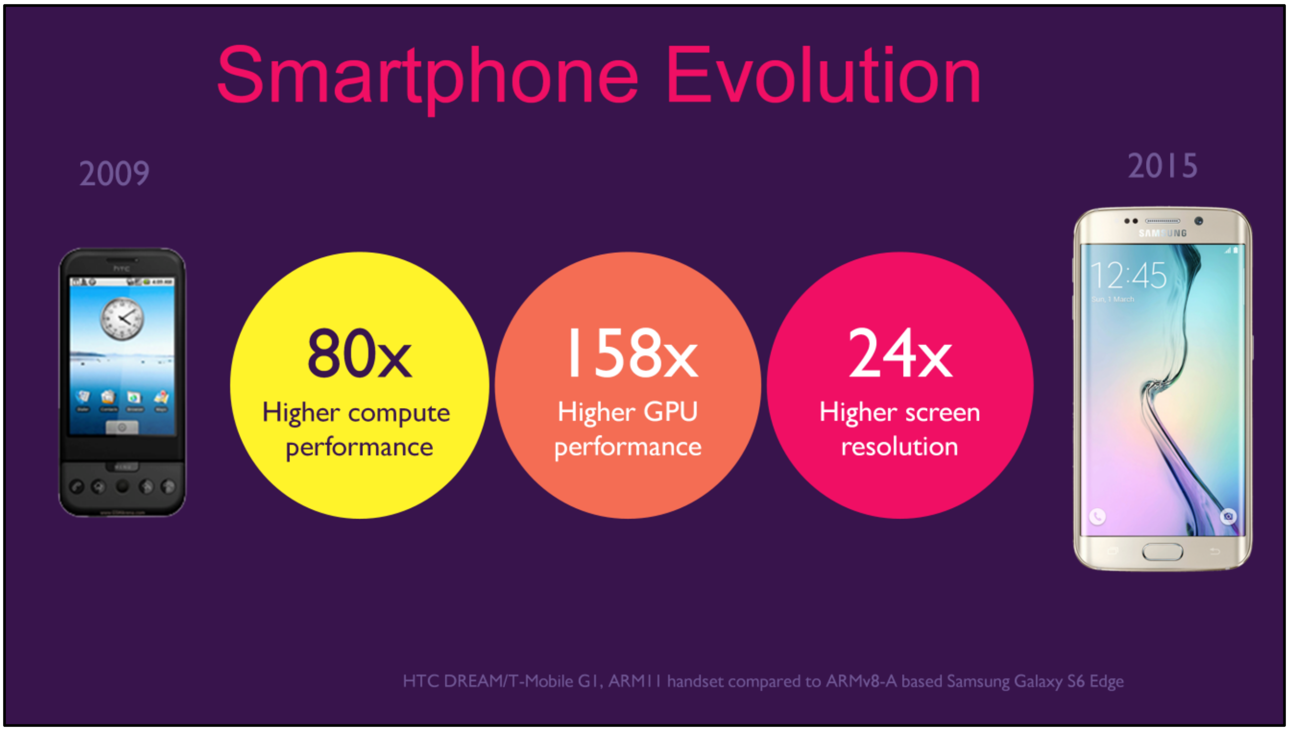
There are approximately 7.4B people in the world about now, of which 4.8B are estimated to be between 15-64*.

Now, 2016 will see around 1.5B smartphones ship. That's a phone for 1 in every 3 in that group. It seems hard to imagine how there could possibly be more demand for smartphones, and so the market is expected to saturate at around 2B phones/year at some point in the next 5 or so years.

This is a market limited only by **world population** and **distribution of wealth**.

Successful advances in graphics technologies that are applicable to mobile will be felt by hundreds of millions of people. Successful advances that are cost-effective will be felt across the globe.

(*Old people – sorry I don't know why you get ruled out as a purchasing demographic in these stats. I suspect a lot of young teenagers have phones too, even if they didn't buy them out of their own pocket money ☺.)

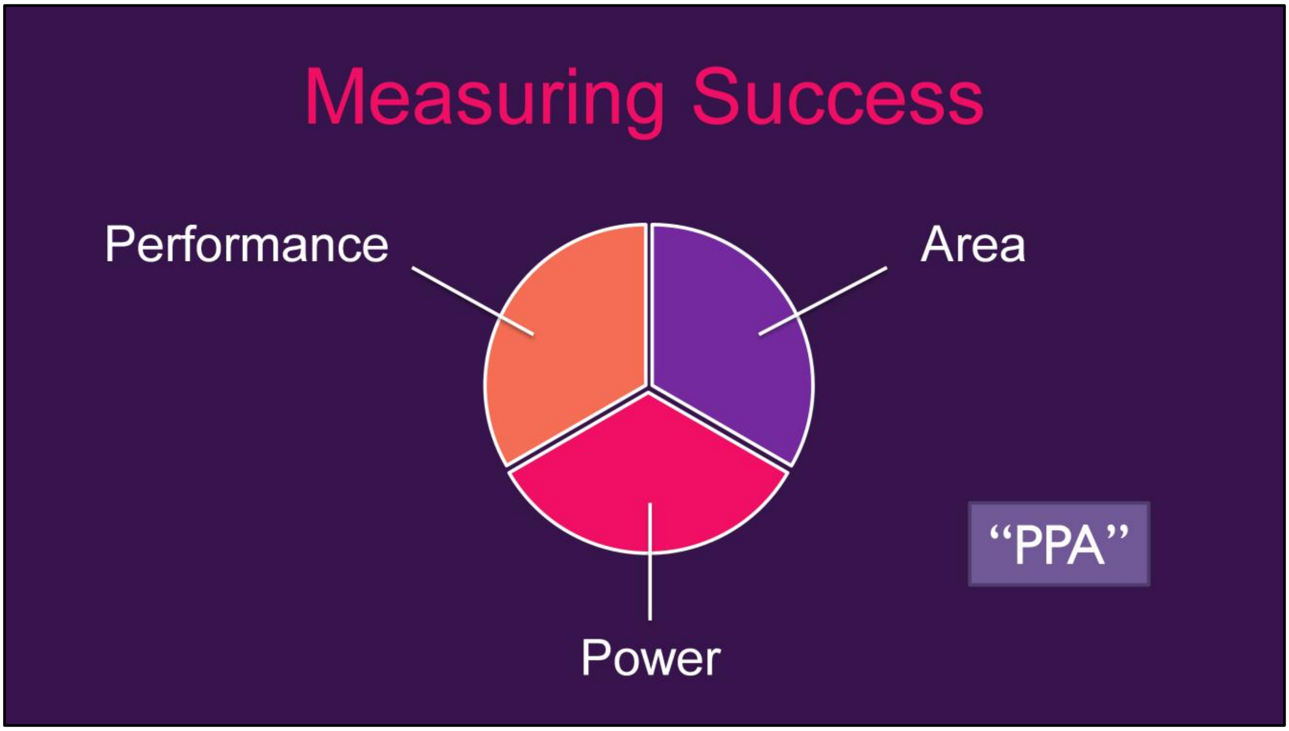


You can also see the demand for smartphones in how quickly they have advanced.

Comparing a phone from 2009 to last years S6 gives you crazy looking numbers. It shows the cumulative effect of many years of innovation, and cannot be explained by process shrinkage alone.

Background for comparison

Device	HTC Dream/T-Mobile G1	Samsung Galaxy S6
CPU	528MHz ARM11	2.1 GHz Cortex-A57MP4 + 1.5 GHz Cortex-A53MP4 big.LITTLE
GPU	Adreno 130 1.2GFLOPS	Mali-T760MP8 190.4 GFLOPS
Screen res	320 x 480 (~180 ppi density)	1440x2560 (~577 ppi density)



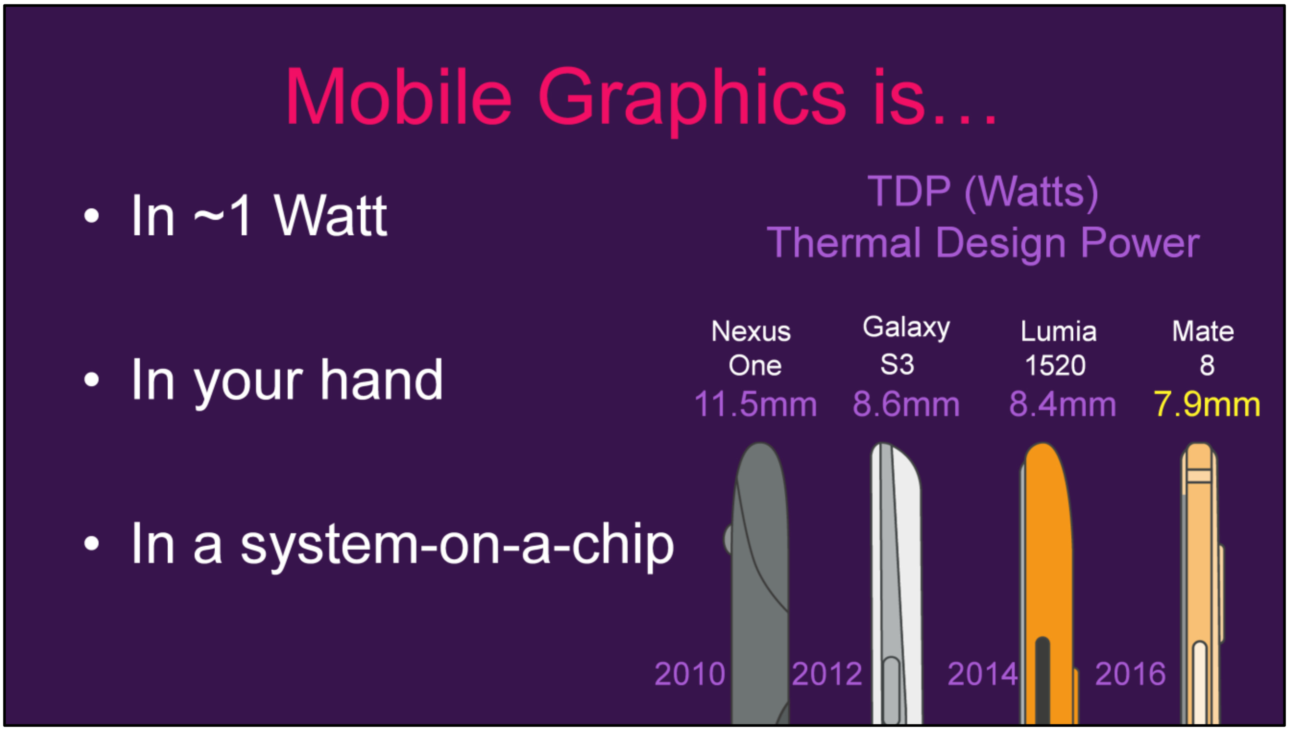
To properly measuring success in mobile it is insufficient to just measure performance. Measuring performance is easy, but doing so without reference to other metrics can give you a rather distorted view of success.

A key measurement to take into account is power, which gives you a more sensible Performance / Watt metric. Pretty much anything on a phone will improve if you provide more power (this is also known as cheating ☺). Similarly, for power constrained applications, like VR, reducing power will actually make things go faster.

The third key measurement is area. Now this is something that only hardware designers and silicon providers have any control over, and in practice increasing area also tends to increase power, but perhaps a useful way of thinking about this is as cost. Die area costs money. If you really want an idea to be applicable to large number of people you need to get this down – recall that wealth distribution is a limiting factor. Sometimes reducing area is a hardware designer managing to make things smaller, but sometimes it's intentionally doing more in software, or reusing other existing hardware blocks, in order trade some power for cost.

These 3 are common and important enough you may hear them abbreviated as PPA.

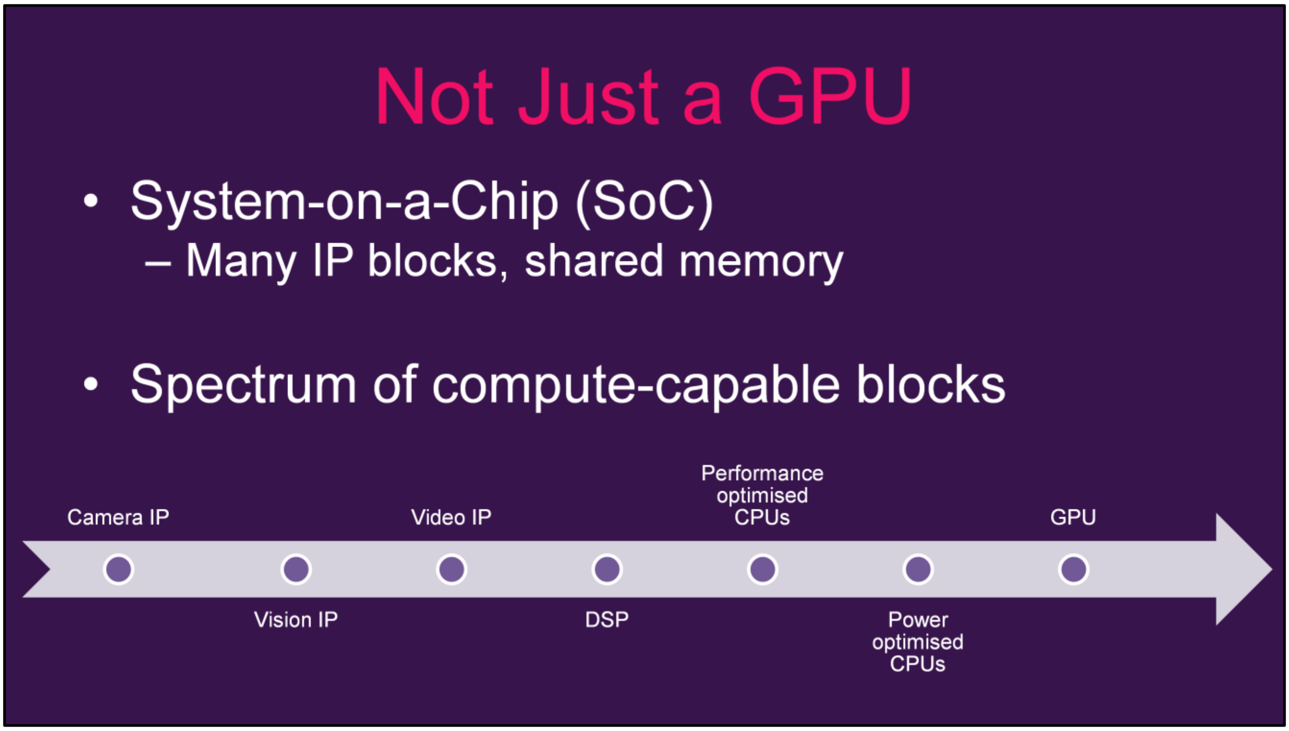
Further measures are bandwidth and latency. Bandwidth costs a lot of power, and often impacts directly on the thermal headroom of the SoC, so it is something you can’t afford to be wasteful with.



So, I would define mobile graphics as something running in around 1 Watt, which is a generally sustainable figure, not including power for display/memory/etc. This can be much more for short periods, but not for anything more than a few seconds.

The 1 Watt cap comes from the fact it has to work in your hand, and is limited by the devices capacity to dissipate heat, expressed as TDP. Phones don't have fans or super-fancy cooling systems, with heat often dissipating back in to the PCB. Bigger things tend to fair better but phones have actually got smaller over time.

The other prevalently property of mobile graphics is that the GPU is always part of a SoC, or "system on a chip".



SoCs are not unique to mobile but are central to the mobile industry. It's a simple but effective idea – take many components from multiple vendors and put them all into a single chip.

On a single chip then benefit from efficient communication and shared resources, such as shared caches and a shared memory system, which nowadays support things like coherent access and hardware protected regions.

The interesting thing for graphics is that graphics is not just about GPUs. There is a wide spectrum of compute-capable blocks, plus some configurable but fixed function blocks focused at images and video.

It's not uncommon to see 8 cpu cores, with some optimised for peak performance (like a webpage load) and others optimised for power (for long running things).

If you don't use it, it's a fair amount of compute just hanging around, consuming area.

Sensors & Connectivity

- NFC, Bluetooth, GPS, Radio, Wi-Fi, Gyroscope,…
  - → availability of media
  - → ability to share and communicate
- Cameras
  - → Computational photography, computer vision
- Pokemon Go
  - → Pokemon + GeoCaching + AR

Mobile graphics also sits amongst an unprecedented array of sensors and communication systems, giving us the necessary raw materials – the availability of media to work with, as well as the ability to create, share and communicate.

Phone cameras are particularly important. They capture our lives and advances in computational photography, or lightfield cameras, or computer vision, or any number of related topics will directly impact this if they can be made relevant to the smartphone. I would single out computer vision as something with the potential to radically change the way we interact with the world.

These sensors also sometimes lead to surprise advances, such as Pokemon Go. Although in this case, the key components are really “pokemon” and “geocaching” rather than the graphics ☺.

Modern APIs

- Modern, low-level ‘explicit’ APIs
  – Vulkan, Metal, DX12
- Tiling GPUs as “first class” citizens
  – Concept of a ‘renderpass’
  – Low overhead, multi-threaded
  – Low level
- Basis for further innovation

Another thing happening right now on mobile is the advent of a new generation of low level, low overhead APIs. Notably Vulkan on Android and Metal on iOS.

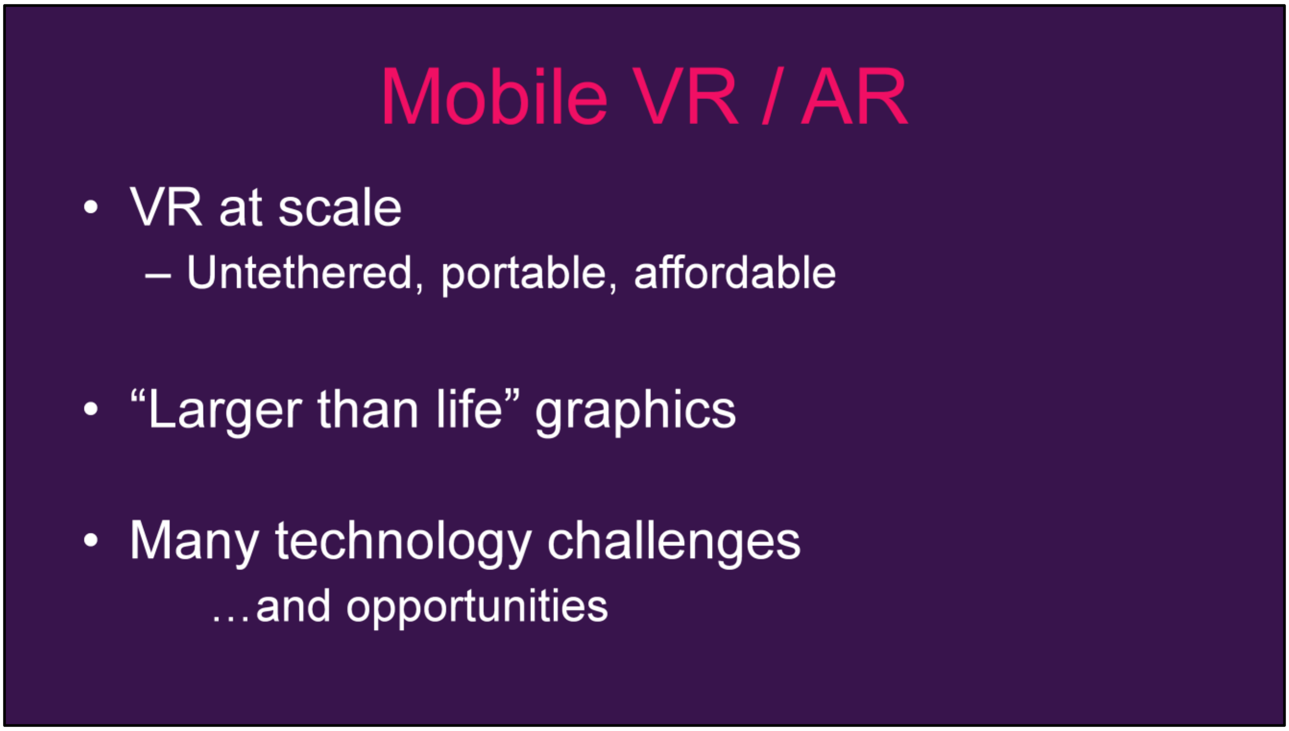
They represent a long-overdue cleanup of graphics APIs, particularly on mobile where OpenGL ES has dominated, and despite it’s clear success as an API, had also dragged a lot of baggage along with it that was getting in the way.

The new APIs do several key things:

- They expose the ‘renderpass’. This is a concept common to all tiling GPUs, and without it you would lose a lot of opportunities for optimisation
- They allow multithreaded command construction. On devices where multiple CPU cores are not only common but vital to low power operation, this is a long overdue advance.
- And finally they are low level. This does make somethings more complicated (witness your first draw triangle program in vulkan), but it also reduces the amount of driver heuristics required.

And as a clean, low level API, it should serve as the basis for further innovation.

Andrew Garrard will talk about best practice for these APIs momentarily.



And of course, there’s VR.

VR is certainly one of the most interesting use cases for mobile, particularly as it is emerging and growing in tandem with tethered VR. We have two talks on VR today, in the second half of the course.

Mobile VR represents VR at scale. VR that’s affordable, portable and ships in volume.

Mobile VR also does something to your phone – it makes it larger than life. The experience of graphics on phones is generally constrained by the screen size. But plug your phone into a VR headset and you have a full high resolution view.

There are many technology challenges remaining in VR, mobile and otherwise, but this is also a massive opportunity for those of us who like a challenge ☺

Moving Mobile Graphics Pt.1

2:15 Best Practices for Mobile
Andrew Garrard, Samsung R&D UK

2:40 Advanced Real-time Shadowing
Marius Bjørge, ARM

3:00 Video Processing with Mobile GPUs
Jay Yun, Qualcomm

http://community.arm.com/moving-mobile-graphics

Moving Mobile Graphics Pt. 2

3:45 Efficient Multiview Rendering for VR
Cass Everitt, Oculus

4:15 Efficient Use of Vulkan on UE4 Mobile
Niklas (Smedis) Smedberg, Epic

4:40 Making EVE: Gunjack
Ray Tran, CCP Shanghai

http://community.arm.com/moving-mobile-graphics

Thank you! Questions?

- Many thanks to…

Our speakers:

- Andrew Garrard
- Marius Bjørge
- Jay Yun
- Cass Everitt
- Niklas Smedberg
- Ray Tran

Special thanks to:

- Micah Knapp (Qualcomm)
- Tobias Hector (Imagination)
- Chris Hebert (NVIDIA)
- Many of my colleagues at ARM

Feedback, thoughts, ideas:

- sam.martin@arm.com
- @palgorithm

http://community.arm.com/moving-mobile-graphics