Computer Graphics on Mobile Devices

Bruno Tunjic*
Vienna University of Technology

Figure 1: Different mobile devices available on the market today. Image courtesy of ASU [ASU 2011].

Abstract

Computer graphics hardware acceleration and rendering techniques have improved significantly in recent years. These improvements are particularly noticeable in mobile devices that are produced in great amounts and developed by different manufacturers. New technologies are constantly developed and this extends the capabilities of such devices correspondingly.

In this paper, a review about the existing and new hardware and software, as well as a closer look into some of the most important revolutionary technologies, is given. Special emphasis is given on new Application Programming Interfaces (API) and rendering techniques that were developed in recent years. A review of limitations that developers have to overcome when bringing graphics to mobile devices is also provided.

Keywords: mobile device, graphics, gpu, API, hardware

1 Introduction

Under the term mobile device we understand any device designed for use in mobile context [Marcial 2010]. In other words this term is used for devices that are battery-powered and therefore physically movable. This group of devices includes mobile (cellular) phones, personal media players (PMP), personal navigation devices (PND), personal digital assistants (PDA), smartphones, tablet personal computers, notebooks, digital cameras, hand-held game consoles and mobile internet devices (MID). Figure 1 shows different mobile devices available on the market today. Traditional mobile phones are aimed at making and receiving telephone calls over a radio link. PDAs are personal organizers that later evolved into devices with advanced units communication, entertainment and wireless capabilities [Wiggins 2004]. Smartphones can be seen as a next generation of PDAs since they incorporate all its features but with significant improvements in screen size and resolution, battery life, memory size, graphics processing and processing in general. Tablet personal computers can be described as lightweight notebook computers that have touch sensitive displays for input rather than a keyboard. A mobile internet device (MID) is a mobile device that provides wireless internet access and fills the void regarding size between smartphones and tablet personal computers. All these devices are very different in many aspects from each other. Most noticeable differences being screen size, processing speed, storage space and size of the devices itself.

According to the Nielsen Media Industry Fact Sheet [Nielsen 2010] there were 223 Million mobile phone users in the United States of America in 2010. Almost 18 percent of the mobile devices on the

*e-mail: e0708079@student.tuwien.ac.at

market are smartphones and overall usage of these devices compared to the usage in 2008 has increased by 5 percent. Due to these events a huge competition between developers and companies is more apparent than ever. Companies like Apple, Samsung, Motorola, Nokia and others are developing new technologies and solutions which make it possible to increase the storage size, speed and overall performance. Therefore today's smartphones and tablets support 3D games and applications and have integrated cameras with up to 10 Megapixels. These examples, amongst many other new features, just prove that graphics on mobile devices have improved in many ways.

As mentioned before, smaller storage, low speed processing and rendering, as well as physical size of the screen were great limitations that in recent years are becoming more and more irrelevant. Even with technologies as System-on-Chip (SoC), where the whole processing units are on a single chip, these limitations are becoming already much less apparent. Since today's tablets and smartphones also have graphical processing units (GPU) that are operating almost in the same manner as the ones in personal desktop computers, screen resolutions as well as rendering are significantly improved.

The motivation for these steady improvements regarding graphics, as explained in a report by T.J. Olson [Olson 2008], is clearly because of the different user expectations, new applications that are emerging and increasing display sizes. This means that new and less expensive technologies are expected. This on the other hand alsoy guarantee to manufactures that they will succeed and sell the product. New applications that are developed can also benefit from better graphics. This means that games, graphic user interfaces and multimedia in general not only look better, but the whole usability and functionality of the application is improved. Last important factor is the need for higher resolutions and screen sizes. Explanation can be found in the advantages of greater resolutions that simply make it possible to work and interact easier with the device.

In the next chapter hardware basics, limitations, as well as some specific products, are explained. This section is followed by an overview of various APIs available and conclusion regarding the whole topic.

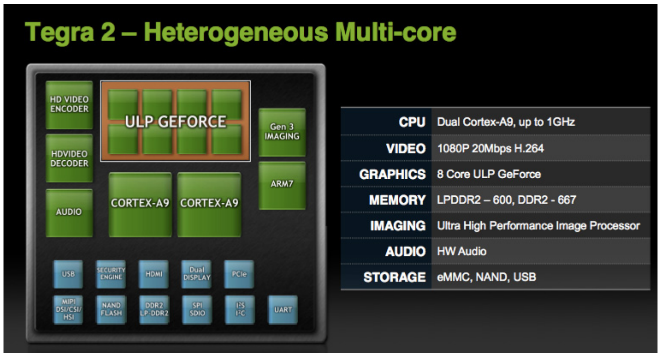
2 Hardware

Some of the key factors that contributed to the success of mobile devices on the market are the dramatic improvements in display technologies [Capin et al. 2008] in the last ten years. Mobile phones like Nokia 3310, which was released in 2000, had a monochromatic display with resolution of 48 x 84 pixels. In recent years a standard smartphone like HTC Desire has a 24-Bit (16.8 million colors) display with a 480 x 800 pixels resolution.

Hardware-wise a lot has changed in recent years. These changes have to be observed from separate views because of the different technologies and mobile devices itself. This means that there is no classification of devices in a way that we distinguish between tablet personal computer and smartphone or any other device, because similar graphics processing units are used in every mobile device.

As mentioned in the introduction, smartphones sales are getting significantly higher every year. This creates higher competition between manufacturers and as a result smartphones are becoming available in mid-range prices. Information that these devices can process is comparable to personal desktop computers on every level. These devices are extremely small but have great processing capabilities. This, however, is a result of new lightweight and micro technologies that are incorporated in them. Tablet personal computers are also small and portable but today very efficient and sophisticated mobile computers that have enjoyed great success with users of all ages and genders. Key features are of course the size and the weight of the device. They are very thin and lightweight but at the same time they offer wide range of functionalities that can be compared to those of personal desktop computers.

Every smartphone, as well as tablet personal computer, uses technology known as System-on-Chip, where all the components are integrated in one chip. This is the reason that it is possible to keep the physical size and weight on a minimum. More about this technology and design of a SoC can be found in the book by S.B. Furber [Furber 2000]. Figure 2 shows an example architecture of a SoC.


Tegra 2 – Heterogeneous Multi-core

CPU	Dual Cortex-A9, up to 1GHz
VIDEO	1080P 20Mbps H.264
GRAPHICS	8 Core ULP GeForce
MEMORY	LPDDR2 – 600, DDR2 - 667
IMAGING	Ultra High Performance Image Processor
AUDIO	HW Audio
STORAGE	eMMC, NAND, USB

Figure 2: NVIDA Tegra 2 SoC architecture. Image courtesy of AnandTech [Shimpi 2011].

There are two key technical challenges while designing a GPU for mobile devices [Olson 2008]:

- Keeping the power consumption as low as possible. This is an apparent problem because every mobile device is powered by a battery and consumers are always demanding longer battery life. This leads to a conflict since the power needed for graphics has to be reduced greatly, in contrast to the graphic cards in personal desktop computers for general use where sometimes expansion busses have to deliver additional power. Some methods that help in solving this challenge are clock gating, dynamic voltage scaling and power gating. Clock gating disables simply the sections that are idle and reduces the power needed for switching. On the other hand dynamic voltage scaling and power gating are responsible for static power. More about power consumption and its analysis can be read in the report by B.Mochocki, K.Lahiri and S.Cadambi [Mochocki et al. 2006].
- Graphics processors designers must overcome storage limitations. This problem is noticeable also on personal desktop computers, but can be in some amount resolved by an increased budget. However there are still limitations in sense of technologies available even for personal desktop computers. Because of this constraint, many graphics processing units used in mobile devices use special rendering techniques, as for example tile-based rendering [Gu et al. 2007] where an image is subdivided and then rendered. This solution, however, in some scenes with a lot of geometry can actually fill out the memory bandwidth. Another proposed solution is the real-time remote rendering [Yoo et al. 2010]. It is a technique that is used when a local device does not have sufficient rendering capabilities. Therefore it is a very useful approach for mobile devices. A proxy server records a 3D scene as an

2D image and the mobile device displays the image using 3D warping [McMillan and Bishop 1995] which allows very efficient warping of the received 2D images to the new rendering viewpoint.

There are many different manufacturers of GPUs that are integrated in smartphones and tablet personal computers, and there are different approaches and technologies that are used. Some of the manufacturers with significant market share in mobile device GPU development are ARM Holdings, PowerVR (a division of Imagination Technologies Grop plc), Qualcomm and NVIDIA. Some of the products designed by these manufacturers are reviewed in the following sections.

2.1 PowerVR graphics

PowerVR graphics IP [Imagination 2011a] is a group of graphics hardware accelerators that have a wide range of usage, but are designed mainly for mobile devices. Power consumption is therefore kept on a minimum using clock gating techniques, whereas there are some differences depending on the family. Their architecture is fully scalable, which makes them usable for low cost as well as very high performance computer devices. General-purpose computing on graphics processing units (GPGPU) technologies enable that the GPU is not only used for handling the computation of computer graphics but also to perform computations which would typically be performed by the CPU. A special tiling technology is used to process the hidden surface removal and pixel blending. This technology uses a rasterizer consisting of 32x32 tile into which the polygons are rasterized. This way the deferred shading is enabled and it is assured that only visible shader effects are gone through by the shader. Performance is characterized with clock frequencies of 400 MHz and more. There are also pre-verified system level cache configurations available (32, 64 and 128 kb) that reduce the memory bandwidth. Application areas are: smartphones, mobile internet devices (MIDs), personal media players, games consoles, high-definition television (HDTV), personal navigation devices (PND), in-car navigation and information, electronic dashboards, ultra-mobile personal computers (UMPC), notebooks and netbooks. There are four families of the intellectual property (IP) cores that PowerVR Graphics offers:

- PowerVR SGX Series5 IP Core Family
- PowerVR SGX Series5XT IP Core Family
- PowerVR MBX IP Core Family
- PowerVR VGX150 IP Core

Almost all products support all of the major APIs, which are OpenGL ES 1.1 and 2.0, OpenVG 1.1, OpenGL 2.0 and 3.0 and DirectX9 and 10.1 and OpenCL.

Apple manufactures a personal computer tablet called iPad. The most recent version of this device is the iPad2 which uses an System-on-Chip (SoC) called Apple A5 that is designed by Apple and manufactured by Samsung. This chip has a dual core GPU called PowerVR SGX543MP2 from the Series5XT incorporated in itself. This same technology is used in another famous Apple product, iPhone 4S, whereas the older products used an earlier version of the A5 SoC called Apple A4 that also had an older version of a GPU from PowerVR SGX Series5. Samsung also has a device that uses this technology and has a respective place in the world of tablet personal computers, namely Samsung GALAXY Tablet which has the same GPU as Apple devices with Apple A4 SoC.

PowerVR's next generation processing units, announced in November 2011, will be the Series 6 with codename Rogue. New architecture and features have not been released nor has any specification or factsheet.

2.1.1 The PowerVR SGX Series5

Some of the GPU models from this series are: SGX520, 530, 531, 535, 540, 545. The newest OpenGL ES 2.0 and DirectX 10.1 APIs are supported. This unit is incorporated in various SoCs like: Apple A4, Texas Instruments' OMAP Series 3 and 4, Samsung's Hummingbird. The devices that use these chipsets are, as mentioned before, Samsung Galaxy S, Galaxy Tab, Wave I and II, Apple iPhone 4, iPad and iPod Touch.

A USSE (Universal Scalable Shader Engine) is incorporated, enabling 2D, 3D and GPGPU processing in a single core. This enables advanced geometry and pixel processing including advanced lighting effects and other related GPGPU tasks. Power consumption is kept on the minimum with the low-power architecture that uses the special clock gating techniques. Figure 3 shows the simplified architecture of a chip from this series.

2.1.2 The PowerVR SGX Series5XT

The PowerVR SGX Series5XT is the next generation of the older Series5. It brings new dedicated YUV and Color Space Accelerators. Products from this series are SGX543MP1-16, SGX544MP1-16, SGX554MP1-16.

A USSE2 (Universal Scalable Shader Engine), as opposed to the USSE in the earlier Series5, is incorporated doubling the floating point in comparison to its predecessor. This enables the best possible image quality and effective GPGPU processing. This family offers Multi-Processor (MP) support for 2 to 16 cores. This is the most important and evident difference from its predecessor. The GPU used by iPhone 4S and iPad2, as mentioned before, therefore has 2 cores since its graphics unit is the SGX543MP2. When it comes to power consumption, the SGX Series5XT architecture offers even greater performance while keeping the dissipation on a minimum. This was achieved by the use of special clock gating techniques, just like in the previous Series5, specific for this family of GPUs. Some other features are the upgraded shader-driven tile-based deferred rendering (TBDR) architecture, support for all standard mobile and desktop APIs and full backwards compatibility with PowerVR MBX and SGX Series5. Simple diagram representing architecture of a chip from this series is shown in figure 4.

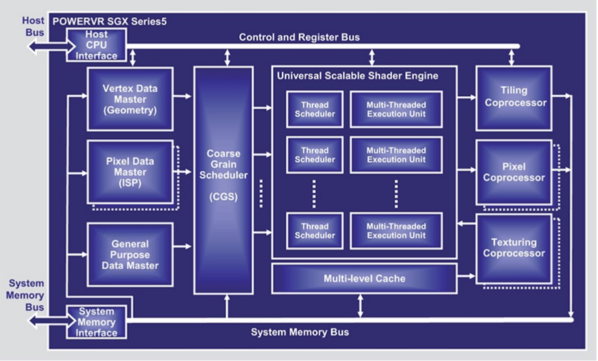


Figure 3: PowerVR SGX Series5 Block Diagram (Single-core). Image courtesy of PowerVR [Imagination 2011b].

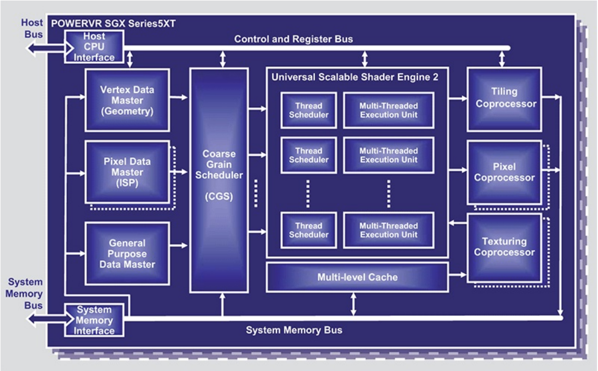


Figure 4: PowerVR SGX Series5XT Block Diagram (Multi-core). Image courtesy of PowerVR [Imagination 2011c].

2.2 ARM Mali Graphics

Mali embedded graphics intellectual property (IP) is just like any other GPU designed to be efficient and used in a wide range of devices. APIs supported include Kronos OpenVG 1.1, OpenGL ES 1.1 and 2.0, and OpenCL, as well as Microsoft's DirectX. Just like any family of graphics processing units its main goal is achieving great performance while keeping the power consumption on a minimum. This company however is not only specialized for designing the GPUs for mobile devices but rather has a wide range of products used by desktop personal computers. Five different GPUs are manufactured in the ARM Mali series, these being T658, T604, 400 MP, Mali-300 and Mali-200.

One of the best sold smartphones today according to [Samsung 2011], Samsung Galaxy S2, incorporates a Mali-400 MP GPU that gives this phone its biggest advantages when compared to other devices in this price range.

2.2.1 ARM Mali-400 MP

Mali-400 MP is a multi-core graphics processing unit, that is scalable from 1 to 4 cores. This unit supports playback at 1080p through OpenGL 2.0. Maximum resolution for texture supported is 4096 pixel along any axis, and for screen resolution is 4096 x 4096 pixels. Integration of this GPU itself is very simple because of the Advanced Microcontroller Bus Architecture (AMBA) and Advanced eXtensible Interface (AXI) that are standards in this field today. One big advantage is also the possibility to integrate it with other ARM Mali products, that are all summarized under the name ARM Mali ecosystem program. The product offers an advanced tile-based deferred shading and full scene anti-aliasing (FSAA) or also called super sampling anti-aliasing (SSAA) that makes the images seem smoother and more realistic through avoiding aliasing on full-screen images. It has an integrated Memory Management Unit (MMU) that reduces the memory bandwidth. One of big advantages of this product is also that the CPU and GPU power management is independent from each other. This minimizes the power consumption and therefore maximizes the battery life. A simplified and short overview of components and architecture can be seen in figure 5.

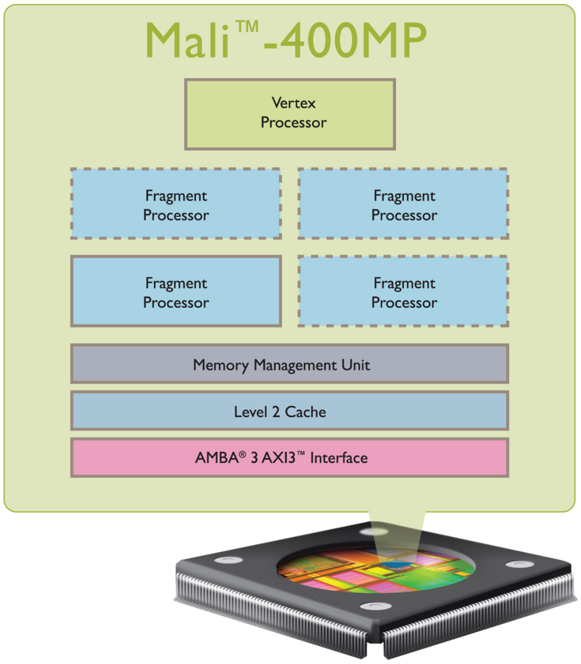


Figure 5: ARM Mali-400 MP structure. Image courtesy of ARM [ARM 2011].

2.3 Qualcomm Adreno Graphics

Qualcomm company (formerly Imageon) designed a whole System-on-Chip series called Snapdragon. Four generations (S1, S2, S3, S4) have been created so far, whereas only the first two are basically used in smartphones and tablets. Only few devices use the third family and the fourth is not aimed at mobile devices at all. Every Snapdragon SoC uses a technology called Adreno for graphics processing [Qualcomm 2011]. The advantages of Qualcomm's Adreno GPUs are described also in figure 6. Adreno uses up the cycles that other GPU architectures leave unusable. The different products in detail are:

- Adreno 130 - included in the Snapdragon S1 chipsets.
- Adreno 200 - available in some Snapdragon S1 chipsets. It has an improved and flexible unified shader architecture and a programmable Function Pipeline.
- Adreno 205 - Snapdragon S2, or the second generation of Snapdragon chipsets, includes this GPU. It features a dedicated 2D OpenVG graphics hardware, 3D hardware that speeds up and smoothens the 2D rendering. It also has more then twice the graphics performance of its predecessor (Adreno 200).
- Adreno 220 - available in S3 Snapdragon chipsests. Key feature is the enhanced level of 3D graphics performance, that enables many improvements in gaming experience. Testings show that it has five times the performance of Adreno 200.
- Adreno 225 - technology not yet available in smartphones or tablets.

Supported APIs are OpenGL ES 2.0 and 1.1, OpenVG 1.1, EGL 1.3, DirectX, SVGT 1.2, Direct Draw, GDI and Concurrent CPU, DSP, graphics and MDP.

Most of the smartphones designed by HTC use the Snapdragon SoCs and therefore also the Adreno technology for graphics processing unit. HTC Desire has the S1 Snapdragon and the Andreno 200 GPU integrated. HTC Evo 4G uses the same chipset, whereas for example the HTC Evo Shift 4G has the Snapdragon S2 chipset with an Adreno 205 GPU on it. The same combination is used by HTC Desire S and HD. S3 generation of the Snapdragon SoC is also used by the HTC Sensation and HTC Sensation XE model.

Some other products that use the first generation technology from Qualcomm are Sony Ericsson Xperia X10, LG Optimus Q, Nexus One, Samsung focus and others. This list includes the devices that use Android OS, but Windows Phone (or its older version called Windows Mobile) as well. Second generation or the S2 is also incorporated in some Sony Ericsson, Motorola and Fujitsu Toshiba smartphones. However the third generation can also be found in the HP TouchPad that has an Adreno 220 integrated.

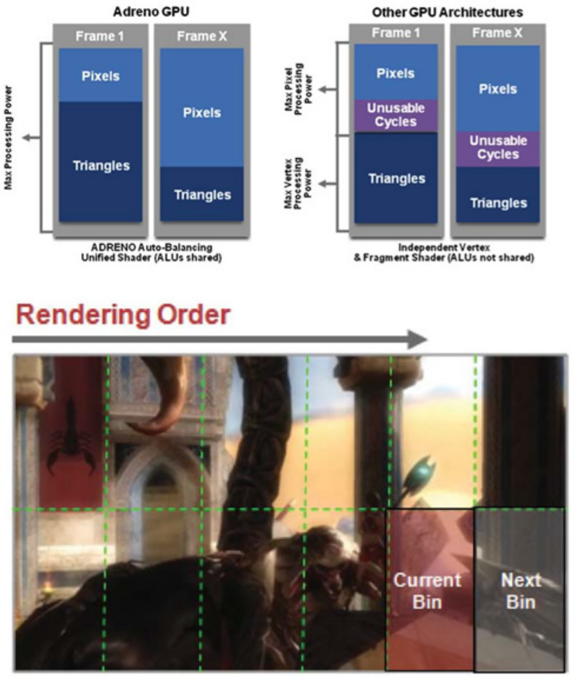


Figure 6: Adreno GPU advantages illustration. Image courtesy of CDRInfo.com [CDRInfo 2011].

2.4 NVIDIA Graphics

NVIDIA is a well known manufacturer of graphics processor units for personal desktop computers, but this company also designed the before mentioned Tegra series of System-on-Chips that are used in smartphones, PDAs and tablet personal computers. Some of the newer SoCs available from NVIDIA are:

- Tegra APX series - Processor used is the ARM11 600 MHz multicore processor (MPCore), it features the NOR or NAND Flash and Mobile DDR memory. It provides up to 12 megapixels camera support and includes the GeForce ULV (Ultra Low Voltage) support for OpenGL ES 2.0 and Direct3D Mobile.
- Tegra 6xx series - ARM11 700 or 800 MHz MPCore processor and low-power DDR memory are featured, as well as support for 720p HD Camera and display up to 1080p.
- Tegra 2 series - Dual-core ARM Cortex-A9 CPU and ULP (Ultra Low Power) GeForce GPU are integrated. Interesting is that there is a version of this SoC that supports 3D displays, however then it uses higher clocked CPU and GPU. Figure 2 shows the chip architecture.
- Tegra 3 series - Newest SoC from NVIDIA. Processor used is the Quad-core ARM Cortex-A9 MPCore with up to 1.4 GHz single core mode and 1.3 GHz multi-core mode. GPU is the 12-Core Nvidia with support for 3D stereo. Supported video output goes up to 2560x1600 pixels. This chip is about five times faster than its predecessor Tegra 2.

SoCs from NVIDIA that can be found in many recent tablets and smartphones are the ones from Tegra 2 series. Some example devices are Motorola Xoom, Samsung Galaxy R, LG Optimus 2X, Tesla Model S, Lenovo ThinkPad Tablet K1 etc. These chipsets use the Ultra Low Power (ULP) GeForce GPU that is further explained in the next subsection. Figure 7 shows architecture diagram of ULP GeForce.

2.4.1 Ultra Low Power GeForce

The Ultra Low Power GeForce GPU is derived from the desktop GeForce GPU architecture, but is adjusted to consume very little power like every other GPU for mobile devices.

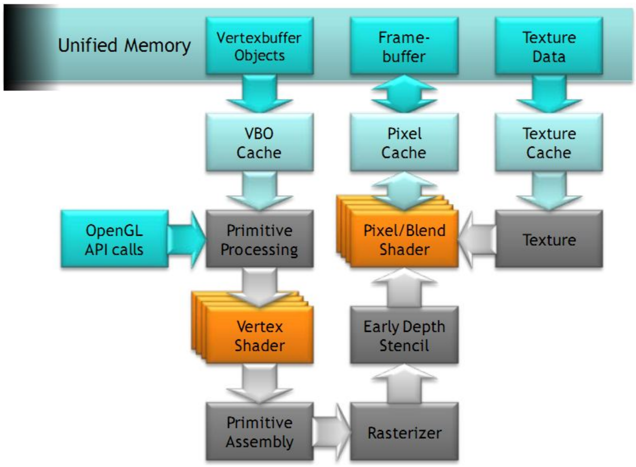


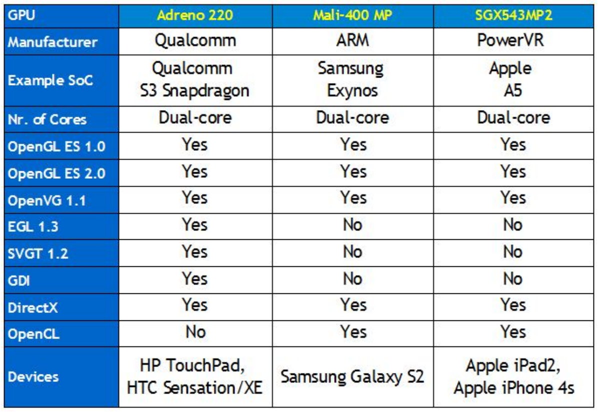
Figure 7: ULP GeForce GPU Architecture. Image courtesy of NVIDIA [NVIDIA 2011].

As seen in Figure 7 the GeForce GPU has four pixel shader cores and four vertex shader cores. This enables high speed image processing and rendering. ULP GeForce GPU provides some additional features to the OpenGL ES 2.0 standards as for example [NVIDIA 2011]:

- Early-Z support - Z-buffer (depth buffer) is used in GPUs to determine which pixels are visible for the observer and which pixels are hidden. This process is very time and power consuming since great memory bandwidth is used to store the depth information for each pixel. For every pixel this value has to be checked to determine if the corresponding pixel is visible. In OpenGL ES 2.0 pipeline this is done after the pixels are processed by pixel shader. GeForce GPU has an advantage that this is done before the pixel shading. This reduces the memory bandwidth between GPU and chip and therefore saves power.

- Pixel cache, Texture cache and Vertex cache - pixel, texture and vertex information are cached, therefore memory accesses are reduced.
- Integrated Pixel Shader and Blend Unit - The pixel shading and blending are not separated as in OpenGL ES 2.0, and this can help implement blendings that OpenGL ES does not support, for example some Adobe Flash blending modes.
- Coverage sampling anti-aliasing (CSAA) - This technique of anti-aliasing minimizes the memory accesses to system memory drastically compared to the traditional multi-sample snti-aliasing (MSAA) and super-sample anti-aliasing (SSAA).
- Advanced Anisotropic Filtering (AF) - The texture details are improved even at extreme view angles.
- Numerous features that regulate power management and reduce power consumption.

2.5 Comparison



GPU	Adreno 220	Mali-400 MP	SGX543MP2
Manufacturer	Qualcomm	ARM	PowerVR
Example SoC	Qualcomm S3 Snapdragon	Samsung Exynos	Apple A5
Nr. of Cores	Dual-core	Dual-core	Dual-core
OpenGL ES 1.0	Yes	Yes	Yes
OpenGL ES 2.0	Yes	Yes	Yes
OpenVG 1.1	Yes	Yes	Yes
EGL 1.3	Yes	No	No
SVGT 1.2	Yes	No	No
GDI	Yes	No	No
DirectX	Yes	Yes	Yes
OpenCL	No	Yes	Yes
Devices	HP TouchPad, HTC Sensation/XE	Samsung Galaxy S2	Apple iPad2, Apple iPhone 4s

Figure 8: Short overview of most up-to-date GPUs for mobile devices.

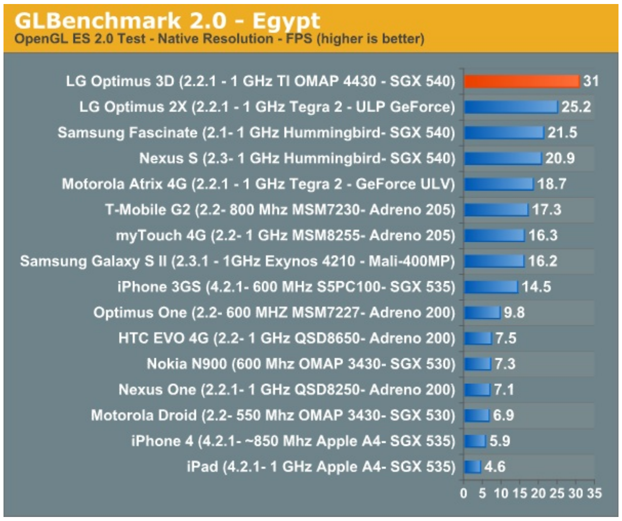


Figure 9: FPS speed comparison of different GPUs. Image courtesy of AnandTech [AnandTech 2011].

Every GPU reviewed in this paper has its advantages, as well as disadvantages. In figure 9 a testing conducted by AnandTech [AnandTech 2011] is presented. GLBenchmark 2.0 [Khronos 2011a] was used to test the performance of different GPUs running OpenGL ES 2.0. The comparison was based on maximum frames per second of each chip. As seen in figure 9, NVIDA Tegra 2 performed significantly better in this test than Mali-400 MP, PowerVR SXG535, as well as Qualcomm Adreno 205. According to the same source, Tegra 2 also has slight advantage based on flash performance. When it comes to API support, almost every newer GPU supports all the major APIs. Detailed explanation of API support by some GPUs is shown in figure 8.

To sum up one could say that all the hardware components described in this paper perform very similar if tested in same conditions. Based on GLBenchmark, Tegra 2 is slightly in advantage, but the fact that has to be considered is that the graphics performance depends also on other components integrated in a mobile device.

3 APIs

An Application Programming Interface(API) is an interface that is used for communication between different applications. Their biggest contribution is that they make it possible to port the code to different accelerators [Olson 2008]. This of course increases the number of different platforms on which a certain application or code can be used. Most famous APIs for desktop personal computers are OpenGL and DirectX. OpenGL is a cross-platform API that is managed by the Khronos Group and it is the main 3D API on MacOS, Linux, a lot of other Unix-based OS and it is also available on Windows. On the other hand, DirectX is an API exclusive for Microsoft platforms and developed by Microsoft.

When 3D graphics for mobile devices resurfaced a sudden need of APIs for mobile devices appeared. This was mostly due to the fact that standard OpenGL and DirectX were too extensive. These standard APIs were designed and developed for desktop computers and therefore the power consumption and memory bandwidth were not taken in consideration. Another reason for appearance of APIs specific for mobile devices was that a standardization in certain amount was needed [Akenine-Moller and Strom 2008]. Mobile3D Graphics was created for Java and a native standard called OpenGL ES (Embedded System) was developed by Khronos Group. Since then, many different APIs appeared and disappeared from the market. An overview of the most important ones is given in the furter section.

3.1 OpenGL ES

OpenGL ES (Embedded System) [Khronos 2011b] in general represents more or less a subset of desktop OpenGL, where parts that were too expensive or inutile were simply removed to accommodate to the capabilities of mobile devices. It supports 2D as well as 3D graphics programming. It is a low-level API between software applications and hardware or software graphics engines. More about the difference between the Low Level API and High Level API can be read in the 3.1 Chapter of Hardware 3D Graphics Acceleration for Mobile Devices by T.J. Olson [Olson 2008]. Important fact is that since it is based on the OpenGL no new technologies or knowledge are needed.

OpenGL ES can be easily adjusted for older mobile phones or today's smartphones. Devices that use it, or may use it, can range from devices with just a few MHz to the most up-to-date devices that have processor speeds up to 1-2 GHz. The overall ease-of-use is very high since it is based on OpenGL so it follows an intuitive design and logical commands. Great advantage is also that it is open source and writing the extensions is encouraged, which is the reason why this technology is constantly improving. There are different approaches for enhancing the whole performance of graphics that use this API. An example is proposed in the paper by Q.P. Zhang and L.L. Lai [Zhang and Lai 2007]. Different versions of OpenGL ES have been released throughout the years:

- OpenGL ES 1.0 - released in 2003.
- OpenGL ES 1.1 - released in 2004.
- OpenGL ES 2.0 - released in 2007.

3.1.1 OpenGL ES 1.0

This version of OpenGL ES was released in 2003 and was the first version of OpenGL ES. It is a subset of OpenGL 1.3 with many for mobile devices irrelevant and expensive functionalities eliminated. OpenGL ES 1.0 API uses a fixed-point arithmetic unit that is more efficient in the devices that do not have the floating-point unit. For more information on this topic see [Nadalutti et al. 2006].

OpenGL ES 1.0 is the official 3D graphics of the operating systems Android and Symbian, and is also used by Sony Playstation 3 along with the 2.0 version of OpenGL ES.

3.1.2 OpenGL ES 1.1

OpenGL ES 1.1 was first released in 2004 and it concentrated itself even more on the hardware acceleration of the API, but at the same time being fully backwards compatible to its predecessor. OpenGL 1.5 was the reference for this version. Improvements included:

- Buffer Objects - New concept that enabled simple vertex array storing and element index data.
- Auto Mipmap Generation - Application does not have to generate the mip-levels anymore. More about Mipmaping in OpenGL can be found in the specifications of OpenGL 1.1 [Khronos 1997].
- Enhanced Texture Processing - This enables the effects as bump-mapping and per-pixel lighting.
- Vertex Skinning Functionality - Smoothens the animation of complex figures.
- User defined Clip-Planes - allows the culling of non-visible polygons in early stage, increasing the overall performance.
- Enhanced Point Sprites and Point Sprite Arrays - This way a realistic particle effect is enabled, because the particles are drawn using points.
- Static and Dynamic State Queries - OpenGL ES can be used in layered software environment.
- Draw Texture - A mechanism for writing pixel rectangles from one or more textures to a rectangular region of the screen was integrated.
- New Core Additions and Profile Extensions - Several improvements regarding OES were incorporated.

All in all, the OpenGL ES 1.1 improved the image quality, increased performance and reduced power consumption significantly, compared to the OpenGL ES 1.0. This version is supported by Android 1.6, iOS, iPad, iPhone, BlackBerry 5.0 operating system, Nintendo 3DS and other.

3.1.3 OpenGL ES 2.0

OpenGL ES 2.0 was developed and designed as a subset of OpenGL 2.0 and the key new feature is the programmable 3D graphics pipeline. Figure 10 shows the digram of fixed function pipeline, whereas the new programmable pipeline can be seen in figure 11. New features integrated in this version are:

- Programmable 3D graphics pipeline - The fixed function transformation and fragment pipeline was replaced by this new feature.
- OpenGL ES Shading Language - It also introduced the possibility to create shader and program objects as well as the ability to write vertex and fragment shaders in OpenGL ES Shading Language that is provided.
- Wider range of precision options is offered for embedded devices.
- Frame Buffer Objects - This way the surface management was simplified.

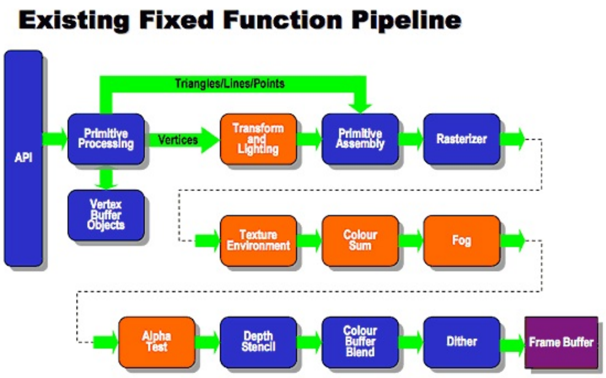


Figure 10: Fixed function pipeline used in OpenGL 1.X. Image courtesy of the Khronos Group [Khronos 2011c].

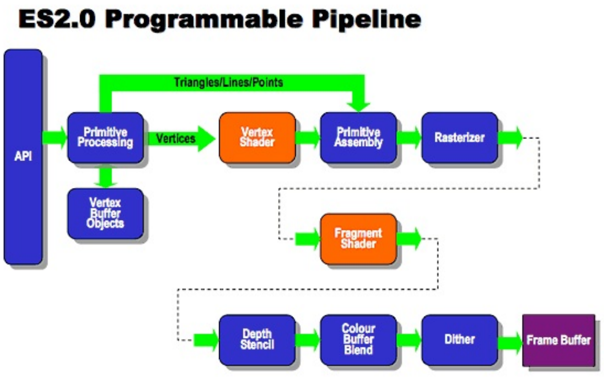


Figure 11: New programmable pipeline introduced in OpenGL ES 2.0. Image courtesy of the Khronos Group [Khronos 2011d].

Only disadvantage of OpenGL ES 2.0 is that because of these changes it is not backwards compatible with its predecessor. Devices and software that use or support OpenGL ES 2.0 are iPad, iPhone 3GS and later, iPodTouch, Android platform from 2.2 version, BlackBerry devices with BlackBerry OS 7.0, WebGL, some new Nokia phones, Palm webOS and others. More about this API can be found in the report by A.Colubri and J.Kim [Colubri and Kim 2010].

3.2 OpenVG

Another API for mobile devices but aimed at vector graphics is the OpenVG 1.0 that was released in 2005. It is actually a match of OpenGL ES but for 2D graphics. Just like other mentioned OpenGL APIs it is a low level and cross-platform API. It targets primarily mobile devices and vector graphics libraries like Flash and Scalable Vector Graphics (SVG). However this API is not exclusively limited to these kind of devices and can be used also on desktop or server machines.

In late 2008 a Open VG 1.1 API was released. This version included some new features as:

- Glyph API added - Very important for hardware accelerated text rendering.
- Support for Adobe Flash and Flash Lite 3 technologies.
- Multi-sampled anti-aliasing (MSAA).

One of the key solutions that this API provides is the reduction of power consumption needed for high quality vector graphics. Scalability is also an advantage, making it possible to adapt to the screen size without multiple bitmaps. Formats that are accelerated with OpenVG are formats like Flash, SVG, PDF, Postscript, Vector fonts etc. Just like any other OpenGL product it is portable and device independent.

3.3 Mobile 3D Graphics

Since first mobile phones that had more complex graphics did not allow to install new C/C++ applications, there was a need to develop a 3D API for Java, because almost every phone producer before 2006 supported Java Mobile Edition (JME or later J2ME). One possible solution was to create a wrapper that enabled almost direct access to OpenGL ES. This project is led under the name JSR 239, however that standard never lived up to its expectations, and this is the reason that Mobile 3D Graphics API was created.

Mobile 3D Graphics (M3G) API is an example of a high-level API for mobile devices. This is a result of the problem that Java is much slower than C or assembly code. This means that only the control logic is given to Java and API is implemented in assembly or C code. First idea was, however, to use the Java 3D API for this purpose [Pulli et al. 2005a]. This idea turned out to be impossible since so many changes would have had to be taken that it did not make sense. This is the reason why a completely new API was created.

Key feature of every 3D content is the animation and M3G API provides a very good support for it [Pulli 2006]. Another important feature is that a programmer has two drawing modes, namely immediate and retained mode. In the immediate mode the graphics commands are issued directly into the graphics pipeline and they are executed immediately. In the retained mode the whole scene graph is rendered at once. Important is that it is always recommended to implement the M3G on top of the OpenGL ES. The biggest advantage of M3G is that it is easy to create and animate a scene. However great disadvantage lies in the fact that this API runs very slow on many mobile phones. Figure 12 shows a M3G game developed in 2003 running on a mobile phone.

Figure 12: Screen shots from a simple M3G game running on a mobile phone already in 2003. Image courtesy of K.Pulli [Pulli et al. 2005a].

More about Mobile 3D Graphics API can be read in the work from K.Pulli and J.Vaarala [Pulli et al. 2005b]. More about the architecture of M3G can by found in the report by M. Andreia F. Rodrigues, R.G. Barbosa and N.C. Mendonca [Andreia et al. 2006].

3.4 Microsoft DirectX for mobile devices

API specific for Microsoft products and operating systems is the well known DirectX. It is actually a collection of APIs that consists of Direct3D, DirectDraw, DirectMusic, DirectSound etc. The corresponding versions of graphics APIs for mobile devices are the Direct3D Mobile and DirectDraw for Windows Embedded CE.

3.4.1 Direct3D Mobile

As mentioned before, the Direct3D Mobile is the application programming interface that provides the support for 3D graphics on Windows CE (Embedded) platforms. Direct3D Mobile compares very closely to Direct3D 8 but has also some elements of Direct3D 9. Its special architecture enables the implementation of driver solutions based on software, hardware or an combination of both. In addition to floating point values this API supports also fixed point values in form of the 16.16 fixed point format, meaning that 16 bits before and after the decimal point are possible.

Just like any API for mobile devices it is created in a manner that it uses the desktop version API as a reference but eliminates some of its feature. This elimination includes the support for some 3D graphics that are not available in handheld devices and expensive, power consuming features.

Some of the functionalities not supported are bump mapping, cube maps, edge antialiasing, emissive materials, gamma correction, higher-order primitives, line stippling, mirror-once texture wrapping, multisample masking, non-local video memory, Phong shading, pixel shaders, point sprites, stereoscopic viewing, vertex blending and indexed vertex blending, vertex shaders and volume textures.

Direct3D Mobile also has limited support in some areas of functionality: depth buffer initialization, multisample anti-aliasing, limited

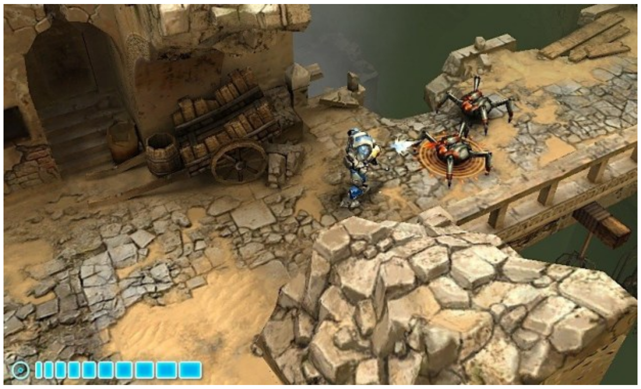


Figure 13: Screenshot from The Harvest, a Direct3D-based mobile game runing on Windows Phone 7 with NVIDIA Tegra SoC. Image courtesy of Techcrunch.com [Coldewey 2011].

to HAL (hardware abstraction layer) devices and support only for single current vertex and index stream.

On the other hand some of the included functionalities are support for 3D Z-buffers, switchable depth buffering, flat and Gouraud shading, multiple light sources, full material and texture support, mipmapping, transformation and clipping, hardware independence, support for page flipping, support for rendering to windowed or full-screen applications, access to image-stretching hardware and exclusive hardware access.

As seen in figure 13 Direct3D-based games also offer high-end graphics. More about the Direct3D Mobile can be read at the Microsoft Direct3D website [Microsoft 2011a].

3.4.2 DirectDraw for Windows Embedded CE

The DirectDraw API provides the hardware acceleration for 2D graphics. This separation of 2D and 3D graphics acceleration can be compared to the same approach that Khronos has, having OpenGL assigned to 3D and OpenVG to 2D graphics hardware acceleration. Mobile version of this API was created, like every other mobile graphics API, by eliminating some expensive features from its desktop version and by expanding some for mobile devices relevant capabilities.

Supported effects are bit-block transfers, page flipping and multiple back buffers overlays, alpha source over destination blending, video YUV pixel formats and color conversion and direct video access to the frame buffer [Microsoft 2011b].

3.5 Collaborative Design Activity

Collaborative Design Activity (COLLADA) developed and maintained by the Khronos Group is an interchange format for 3D assets that contains the information about graphics. This means that it enables communication and transport of data from one digital content creation tool to another or to a completely different application. It uses an open standard XML [W3C 2011] schema for this purpose. This makes it possible to create and read 3D content on different machines and systems. Figure 14 shows how a scene generated on a personal desktop computer can be used on a mobile device, when using COLLADA as the interchange format.

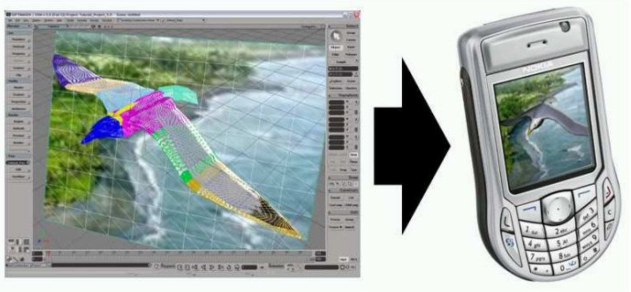


Figure 14: ”Collada facilitates creation of interactive content with digital content creation tools and exporting it to a variety of systems, in this case to an M3G file displayed on a mobile phone.” [Pulli 2006]. Image courtesy of K.Pulli.

3.6 Summary

OpenGL ES and Direct3D Mobile are two main APIs when it comes to new generation mobile devices. A comparison between the two is based mostly on one key factor, which is the number of supported platforms. Therefore the biggest advantage that OpenGL ES has over the Direct3D Mobile is that it is a cross-platform API. As mentioned before, Microsoft's DirectX, and therefore Direct3D Mobile and DirectDraw for Windows Embedded CE, are limited to Windows operating systems. The fact is that there are a great number of Android-based mobile device, as well as devices that run Apple's iOS. Since DirectX does not support these systems, this API is much less spread than OpenGL ES.

When it comes to performance, a plain comparison is not possible. Reason being that APIs on different platforms and hardware give different results. Mobile 3D Graphics is a high-level API which can not be compared to OpenGL ES and Direct 3D Mobile which are low-level APIs. Therefore performance tests between the three are unreliable. COLLADA is more of a content interchange format than API. Because of that it is excluded from any kind of comparison with other APIs mentioned in this paper.

4 Conclusion

Graphics on computers in general have improved in great amount in the last twenty years. This statement also applies to mobile graphics which matured even more in much shorter timespan. One fact that has to be considered is that mobile graphics had a very good background to begin with. The main goal of all APIs and hardware was, and still is, to bring the best existing graphics to these devices. This means that mostly nothing new has to be invented but, more or less, redesigned and adjusted. These adjustments include the memory usage and power consumption reduction.

Smartphones took over the market in just a couple of years and replaced the existing mobile phones in its entirety. Tablets are becoming more and more functional and soon will probably almost completely replace notebook and possibly desktop computers. Apple products like iPod and other multimedia players are something that is, from consumer's point of view, almost irreplaceable and constantly needed. All this leads to a great deal of competition between companies. Therefore the market is flooded with new technologies. This means that market will be expanding in every aspect, bringing always new devices and technologies while keeping them as cheap as possible. Therefore it is guaranteed that graphics APIs and hardware will continue to improve and follow the trends and expansions in the field of computer graphics.

References

AKENINE-MOLLER, T., AND STROM, J. 2008. Graphics Processing units for Handhelds. In *Proceedings of IEEE Conference*, vol. 96, 779–789.

ANANDTECH, 2011. Samsung's Galaxy S II Preliminary Performance: Mali-400MP Benchmarked. http://www.anandtech.com/show/4177/samsungs-galaxy-s-ii-preliminary-performance-mali400-benchmarked. Accessed December 11, 2011.

ANDREIA, M., RODRIGUES, F., BARBOSA, R., AND MENDONCA, N. 2006. Interactive Mobile 3D Graphics for On-the-go Visualization and Walkthroughs. In *SAC '06 Proceedings of the 2006 ACM symposium on Applied computing*, 1002–1007.

ARM, 2011. ARM Mali-400 MP Specifications. http://www.math.drofnats.edu/~gauss. Accessed November 22, 2011.

ASU, 2011. Angelo State University Information Technology. http://www.angelo.edu/services/technology/mobile/. Accessed December 12, 2011.

CAPIN, T., PULLI, K., AND AKENINE-MOLLER, T. 2008. The State of the Art in Mobile Graphics Research. *Computer Graphics and Applications, IEEE 28*, 74–84.

CDRINFO, 2011. Qualcomm Details New Snapdragon S4 SoCs. http://www.cdrinfo.com/sections/news/Details.aspx?NewsId=31301. Accessed November 20, 2011.

COLDEWEY, D., 2011. XNA 4.0 games on Windows Phone 7 Series look awesome (but won't be for Zune HD). http://techcrunch.com/2010/03/09/xna-4-0-games-on-windows-phone-7-series-look-awesome-but-wont-be-for-zune-hd/. Accessed November 28, 2011.

COLUBRI, A., AND KIM, J. 2010. Introduction to Processing for Android. In *SA '10 ACM SIGGRAPH ASIA 2010 Courses*.

FURBER, S. B. 2000. *ARM system-on-chip architecture*. Pearson Education.

GU, R., YEH, T., HUNAG, W., HUANG, T., TSAI, C., LEE, C., CHIANG, M., HSIAO, S., CHANG, Y., AND HUANGI, I. 2007. A Low Cost Tile-based 3D Graphics Full Pipeline with Realtime Performance Monitoring Support for OpenGL ES in Consumer Electronics. *ISCE 2007. IEEE International Symposium on Consumer Electronics*, 1–6.

IMAGINATION, 2011. PowerVR Information and Specifications. http://www.imgtec.com/powervr/powervr-graphics.asp. Accessed November 15, 2011.

IMAGINATION, 2011. PowerVR Information and Specifications. http://www.imgtec.com/factsheets/powervr/POWERVR_SGX_Series5_IP_Core_Family_[3.2].pdf. Accessed November 15, 2011.

IMAGINATION, 2011. PowerVR SGX Series5X Factsheet. http://www.imgtec.com/factsheets/powervr/POWERVR_SGX_Series5XT_IP_Core_Family_[3.1].pdf. Accessed November 15, 2011.

KHRONOS, 1997. Mipmaping in OpenGL 1.1. http://www.opengl.org/documentation/specs/version1.1/glspec1.1/node84.html. Accessed November 27, 2011.

KHRONOS, 2011. GLBenchmark. http://www.glbenchmark.com/. Accessed December 11, 2011.

KHRONOS, 2011. OpenGL. http://www.khronos.org/. Accessed November 22, 2011.

KHRONOS, 2011. OpenGL 1.X. http://www.khronos.org/opengles/1_X/. Accessed November 22, 2011.

KHRONOS, 2011. OpenGL 2.X. http://www.khronos.org/opengles/2_X/. Accessed November 22, 2011.

MARCIAL, L. H., 2010. A comparison of screen size and interaction technique: Examining execution times on the smartphone, tablet and traditional desktop computer. http://marcial.web.unc.edu/files/2011/05/Marcial_lit_review_for_cmte.pdf. Accessed December 11, 2011.

MCMILLAN, L., AND BISHOP, G. 1995. Plenoptic modeling: an image-based rendering system. In *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 39–464.

MICROSOFT, 2011. Direct3D Mobile. http://msdn.microsoft.com/en-us/library/aa452478.aspx. Accessed December 5, 2011.

MICROSOFT, 2011. DirectDraw. http://msdn.microsoft.com/en-us/library/ee491091.aspx. Accessed December 5, 2011.

MOCHOCKI, B., LAHIRI, K., AND CADAMBI, S. 2006. Power Analysis of Mobile 3D Graphics. In *Proceedings Design, Automation and Test in Europe*, 1–6.

NADALUTTI, D., CHITTARO, L., AND BUTTUSSI, F. 2006. Rendering of X3D Content on Mobile Devices with OpenGL ES. In *Web3D '06 Proceedings of the eleventh international conference on 3D web technology*, 19–26.

NIELSEN, 2010. Media Industry Fact Sheet. http://blog.nielsen.com/nielsenwire/press/nielsen-fact-sheet-2010.pdf. Accessed November 15, 2011.

NVIDIA, 2011. Bringing high-end graphics to handheld devices. http://www.nvidia.com/content/PDF/tegra_white_papers/Bringing_High-End_Graphics_to_Handheld_Devices.pdf. Accessed November 22, 2011.

OLSON, T. 2008. Hardware 3D Graphics Acceleration for Mobile Devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 5344 – 5347.

PULLI, K., AARNIO, T., ROIMELA, K., AND VAARALA, J. 2005. Designing Graphics Programming Interfaces for Mobile Devices. *IEEE Computer Graphics and Applications 25*, 6, 66–75.

PULLI, K., VAARALA, J., MIETTINEN, V., AARNIO, T., AND CALLOW, M. 2005. Developing mobile 3D applications with OpenGL ES and M3G. In *Proceedings of SIGGRAPH '05 ACM SIGGRAPH 2005 Courses*, 1–54.

PULLI, K., 2006. New APIs for Mobile Graphics. Nokia Research Center and MIT CSAIL, Cambridge, MA, US. research.nokia.com/files/pulli_spie06.pdf.

QUALCOMM, 2011. Adreno Graphics Processing Units. `https://developer.qualcomm.com/discover/chipsets-and-modems/adreno`. Accessed November 30, 2011.

SAMSUNG, 2011. Samsung GALAXY S II continues success reaching 10 Million in global sales. `http://www.samsung.com/us/news/newsRead.do?news_seq=19920&page=1&gltype=globalnews`. Accessed December 11, 2011.

SHIMPI, A. L., 2011. NVIDIA's Tegra 2 Take Two: More Architectural Details and Design Wins. `{http://www.anandtech.com/show/4098/nvidias-tegra-2-take-two-more-architectural-details-and-design-wins/2}`. Accessed November 15, 2011.

W3C, 2011. Extensible Markup Language (XML). `http://www.w3.org/XML/`. Accessed December 5, 2011.

WIGGINS, R. H. 2004. Personal Digital Assistants. *Journal of digital imaging the official journal of the Society for Computer Applications in Radiology 17*, 5–17.

YOO, W., SHI, S., JEON, W. J., NAHRSTEDT, K., AND CAMPBELL, R. H. 2010. Real-time parallel remote rendering for mobile devices using graphics processing units. In *2010 IEEE International Conference on Multimedia and Expo (ICME)*, 902–907.

ZHANG, Q., AND LAI, L. 2007. Strategies of Enhancing the Performance of Embedded 3D Graphics Applications. *2007 Intenational Conference on Machine Learning and Cybernetics 7*, 4231–4236.