Accelerating Mobile Augmented Reality
Neil Trevett
VP Mobile Content, NVIDIA
President, Khronos Group

Topics
- Why mobile devices will be ideal for Augmented Reality
- The challenges for compelling mobile Augmented Reality
- Trends in mobile SOCs acceleration and associated APIs
- Emerging trends and directions in the future of mobile AR
- How to start developing mobile AR on Tegra devices
Personal Computing Redefined

Global Smartphone Market Share
Vision-based Augmented Reality

- Camera video stream sent to the compositor
- Camera-to-scene transform locks the 3D rendering to the real world
- 3D augmentations composited with video stream

AR = Input AND Output Processing

- Augmented Reality is uniquely challenging
- Needs sophisticated INPUT AND OUTPUT processing
- *In perfect harmony*

Detecting user gestures and actions

Sensing and tracking the context and the scene around the user → Augmented Reality Application → Generating 3D Graphical Augmentations

Compositing real and synthetic elements to delight and inform
How Many Sensors are in a Smartphone?

- Light
- Proximity
- 2 cameras
- 3 microphones (ultrasound)
- Touch
- Position
  - GPS
  - WiFi (fingerprint)
  - Cellular (tri-lateration)
  - NFC, Bluetooth (beacons)
- Accelerometer
- Magnetometer
- Gyroscope
- Pressure
- Temperature
- Humidity

The Vision of Wearable Computing - Tomorrow

- Google Glass concept video Looks Great! BUT...
- ... Google Glass hardware does not provide full FOV augmentations
- ... Not using vision tracking, just GPS
- But wearable computers will be essential to consumer adoption of Augmented Reality.

Courtesy Google
http://www.youtube.com/watch?v=9c6W4CCU9M4
In the meantime a lot of development is happening with todays devices...

Combining GPS, accelerometer, gyro and camera tracking

Lots of current research into best feature tracking algorithms - all of them very compute intensive

Beachhead industrial applications could use todays devices...

http://www.youtube.com/watch?v=xw3M-TNOo44&feature=related

Courtesey Metaio

Accelerating AR to Meet User Expectations

Ingredients for great mobile AR...

- Cameras with advanced controls
- Lots of secondary sensors - GPS, sensor, gyro etc.
- High performance image and vision processing
- Fast 3D processing

Mobile is an enabling platform for Augmented Reality

- Mobile SOC and sensor capabilities are expanding quickly

But we need mobile AR to be 60Hz buttery smooth AND low power

- Power is now the main challenge to increasing quality of the AR experience

SOC = 'System On Chip'

Complete compute system minus memory and some peripherals
Tegra 3
The World’s First Mobile Quad Core, with 5th Companion Core for Low Power

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>GPU</th>
<th>VIDEO</th>
<th>AUDIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quad Core, with 5th Companion Core</td>
<td>Up to 3x Higher GPU Performance</td>
<td>Blu-Ray Quality Video</td>
<td>HD Audio, 7.1 channel surround</td>
</tr>
<tr>
<td></td>
<td>Up to 1.4GHz Single Core, 1.3GHz Quad Core</td>
<td>12 Core GeForce GPU</td>
<td>1080p High Profile @ 40Mbps</td>
<td></td>
</tr>
</tbody>
</table>

Mobile SOC Performance Increases

- 1100
- 100
- 10
- 1

25x perf increase in next three years
Power is the New Design Limit

- The Process Fairy keeps bringing more transistors
  - Transistors are getting cheaper
- The End of Voltage Scaling
  - The Process Fairy isn’t helping as much on power as in the past

In the Good Old Days
Leakage was not important, and voltage scaled with feature size

- \( L' = L/2 \)
- \( V' = V/2 \)
- \( E' = CV^2 = E/8 \)
- \( f' = 2f \)
- \( D' = 1/L^2 = 4D \)
- \( P' = P \)

Halve L and get 4x the transistors and 8x the capability for the same power

The New Reality
Leakage has limited threshold voltage, largely ending voltage scaling

- \( L' = L/2 \)
- \( V' = V \)
- \( E' = CV^2 = E/2 \)
- \( f' = ~2f \)
- \( D' = 1/L^2 = 4D \)
- \( P' = 4P \)

Halve L and get 4x the transistors and 8x the capability for 4x the power

Mobile Thermal Design Point

Leakage current in latest silicon geometries means that Moore’s Law no longer delivers more performance without increasing power

- 4-5” Screen takes 250-500mW
- 7” Screen takes 1W
- 10” Screen takes 1-2W
- Resolution makes a difference! The iPad3 screen takes up to 8W

Typical max system power levels before thermal failure
Even as battery technology improves - these thermal limits remain
How to Save Power?

- Much more expensive to MOVE data than COMPUTE data
- Energy efficiency must now be key metric during silicon AND software design
  - Awareness of where data lives, where computation happens, how is it scheduled
- Need to use hardware acceleration
  - Lots of processing in parallel
  - Efficient caching and memory usage
  - Reduces data movement

Example - Typical Camera ISP

- Camera ISP (Image Signal Processor) has little or no programmability
  - Scan-line-based, Data flows through compact hardware pipe
  - No global memory used to minimize power
- BUT… computational photography apps now want to mix non-programmable ISP processing with more flexible GPU or CPU processing
  - ISP pipelines will provide tap/insertion points to/from CPU/GPU at critical points
Engineering Software Apps for Power

- Use Dark Silicon - specialized hardware only turned on when needed
  - SoCs lots of space for transistors - just can’t turn them all on at same time!
  - Dedicated units increase locality and parallelism of computation to save power compared to programmable processors
- Be power smart when using programmable processors
  - Instrumentation will soon appear for energy-aware compilers and profilers
  - Dynamic and feedback-driven software power optimization
  - Power optimizing compiler back-end compilers / installers
- Smart, holistic use of sensors and peripherals
  - Wireless modems and networks
  - Motion sensors, cameras, networking, GPS

Programmers View of Typical SOC c. 2012
Khronos Connects Software to Silicon

ROYALTY-FREE, OPEN STANDARD APIs for advanced hardware acceleration
- Graphics, video, audio, compute, visual and sensor processing
- Low level silicon to software interface needed on every platform
- Defines the forward looking roadmap for the silicon community
- Shipping on billions of devices across multiple operating systems
- Rigorous conformance tests for cross-vendor consistency
- Khronos is OPEN for any company to join and participate
- Acceleration APIs BY the Industry FOR the Industry

Mobile Platform Innovation Vectors

Console-Class 3D
- Performance, Quality, Controllers and TV connectivity

OpenGL ES

OpenVX
- Vision
- Cameras as sensors, Computational Photography, Gesture Processing

Sensor Fusion
- Devices become ‘magically’ context aware – location, usage, position

StreamInput

HTML5 and WebGL
- Web Apps that can be discovered on the Net and run on any platform

New platform capabilities being driven by SILICON and APIs
Native APIs for Augmented Reality

FCAM - Open Source Project
- Capture of stream of camera images with precision control
  - A pipeline that converts requests into images
  - All parameters packed into the requests - no visible state
  - Programmer has full control over sensor settings for each stream frame
- Control over focus and flash
  - No hidden daemon running autofocus/metering
- Control ISP
  - Can access supplemental statistics from ISP
**Requested Camera Extensions for AR**

- Enhanced exposure parameters in FCAM single or burst mode
  - ISO, white balance, frame rate, focus modes, resolution
- Synchronization with other system sensors
  - And with output displayed frame
- ROI extraction
  - From wide angle and fish-eye lenses
- Data output format control
  - Grayscale, RGB(A), YUV
  - Access to the raw data e.g. Bayer pattern
- Query camera information
  - Focal length (fx, fy), principal point (cx, cy), skew (s), image resolution (h, w)
  - Spatial information of how cameras and sensors are placed on device
  - Calibration and lens distortion

---

**OpenCV - Open Source Project**

- Computer vision open source project
  - Excellent functionality - widely used in academia, fast prototyping, some products
  - Not an API definition and not managed by Khronos
- Extensive functionality >1,000 functions
  - Difficult for silicon vendors to provide complete acceleration
- Traditionally runs on a single CPU
  - Some partial acceleration projects underway: OpenCL, CUDA, Neon ...
OpenCV4Tegra

- Tegra-specific acceleration for OpenCV Library
  - OpenGL ES GLSL, ARM Multithreading and NEON optimizations
  - Available on Tegra Android Development Platform Toolkit (TADP)
- Develop native vision functionality in the NDK
  - Can access from Java applications using JNI

**Functions**

**Core**
- absdiff, add, bitwise_and, bitwise_not, bitwise_or, bitwise_xor, compare, countNonZero, cvRound, dot, max, mean, meanStdDev, min, minMaxLoc, norm2, norm, phase32f, reduceC, reduceR, subtract, sum

**Imgproc**
- Blur3x3, blur5x5, canny, cvtColorGray2RGBA, cvtColorRGBA2Gray, cvtColorYCrCb, cvtColorYUV420, dilate, erode, gaussianBlur3x3, gaussianBlur5x5, integral, medianBlur3x3, pyrDown, pyrUp, resizeAreaDownBy2, resizeAreaDownBy4, resizeDownLinear, resizeUpLinear, scharrFilter, sobelFilter, threshold

**Calib3d**
- Features2d (FAST detector)
- Stitching (warppers)

---

**OpenCV4Tegra - Neon Speedups**

![NEON Speedup on Tegra](chart.png)
OpenCV4Tegra - GPU Speedups

Tegra GPU Speedup

OpenVX

Vision Hardware Acceleration Layer
- Enables hardware vendors to implement accelerated imaging and vision algorithms
- For use by high-level libraries or apps

Focus on enabling real-time vision
- On mobile and embedded systems

Diversity of efficient implementations
- From programmable processors, through GPUs to dedicated hardware pipelines

OpenVX does not duplicate OpenCV functionality, just provides essential acceleration.

Aiming for provisional specification in 1H 2013 and final specification 2H13
**OpenVX Execution Flow**

- **OpenVX Graph for efficient execution**
  - Each Node can be implemented in software or accelerated hardware
- **EGL provides data and event interop - with streaming**
  - BUT use of other Khronos APIs are not mandated
- **VXU Utility Library provides efficient access to single nodes**
  - Open source implementation

---

**Market Demand for Sensor Fusion API**

- Innovative use of growing sensor diversity
- Synchronized use of multiple interoperating sensors in one app
- PORTABLE apps need to be isolated from sensor and OS details
- Application developers do not wish to be Sensor Fusion experts

---

**StreamInput**

A High-level Sensor Fusion API

- Do NOT force the application developer to access individual sensors (unlike almost all other sensor APIs)
- High-level API enables sensor vendors to drive and deliver competitive sensor fusion innovation
StreamInput - Portable Access to Sensor Fusion

Apps request semantic sensor information
StreamInput defines possible requests, e.g.
“Provide Skeleton Position” “Am I in an elevator?”

Advanced Sensors Everywhere
RGB and depth cameras, multi-axis motion/position, touch and gestures, microphones, wireless controllers, haptics keyboards, mice, track pads

Processing graph provides sensor data stream
Utilizes optimized, smart, sensor middleware
Apps can gain ‘magical’ situational awareness

Apps Need Sophisticated Access to Sensor Data
Without coding to specific sensor hardware

Apps request semantic sensor information
StreamInput defines possible requests, e.g.
“Provide Skeleton Position” “Am I in an elevator?”

Processing graph provides sensor data stream
Utilizes optimized, smart, sensor middleware
Apps can gain ‘magical’ situational awareness

Using Android Java for AR

Limited fusion. No cross sensor synch. No virtual sensors

Java Callback
Preview frames
OpenCV through JNI

30FPS but 2-3 frames latency. Framerate jitter
Need synth between input and composited frames

No advanced camera controls: ROI, burst parameter control, focus modes, format choices etc.

Java provides access to EGL-type and OpenGL ES functionality

No programmable, close to sensor, image processing
Tegra Android Development Pack

- GET STARTED in minutes NOT hours
- INSTALLS all tools required for Tegra Android
- CPU DEBUGGING with Nsight Tegra
- GPU DEBUGGING with PerfHUD ES
- OPTIMIZE applications with Tegra Profiler
- REFERENCE docs, samples & tutorials
- OPTIMIZED for Tegra Android development
- FLASHESS Tegra DevKit with OS Image
- CONFIGURED for debugging and profiling
- INCLUDES Kernel symbols and DS-5 support

http://developer.nvidia.com/develop4tegra

AR Portability with Middleware Engines

- Middleware SDKs and tools for application developers
  - Hides OS details - provides OS portability
- Porting and optimization through close cooperation with silicon vendors
  - NVIDIA has close relationships with Metaio and Unity
  - Optimizing for Tegra power/performance

Applications

Apps Engines

Native APIs and Java APIs
OpenGL ES, EGLStreams, StreamInput

Partners
Future AR Tracking

- The industry has just started figuring out how to do highly realistic Augmented Reality
- Research today using CUDA-accelerated PCs

Real-Time Surface Light-field Capture for Augmentation of Planar Specular Surfaces
Jan Jachnik, Richard A. Newcombe, Andrew J. Davison
Imperial College, UK

More Hyper Realistic Augmentations

High-Quality Reflections, Refractions, and Caustics in Augmented Reality and their Contribution to Visual Coherence
P. Kán, H. Kaufmann
Institute of Software Technology and Interactive Systems, Vienna University of Technology, Vienna, Austria
CUDA on Tegra

- Tegra + discrete GPU development platforms
  - Available to select developers
- CUDA 5.0
  - Kepler support
- Enables early development of ARM-based applications with desktop-class graphics and compute

See me after the presentation if you are interested
- Or email ntrevett@nvidia.com

In Summary

- Augmented Reality will change the way we use computing - again - but needs significantly more improvement
- Multiple APIs and standards being put into place to deliver the needed performance at low power
- You can get started today with AR on NVIDIA Tegra
  - Today - Optimized Android Java APIs, OpenCV4Tegra, OpenGL ES, Unity
  - Talk to me if you have interest ARM/desktop GPU development systems
  - Soon - FCAM, GPU Compute, OpenVX, Metaio AR SDK