Vision Acceleration
Launch Briefing October 2014

Neil Trevett
Vice President Mobile Ecosystem, NVIDIA
President, Khronos Group

Khronos Connects Software to Silicon

Open Consortium creating ROYALTY-FREE, OPEN STANDARD APIs for hardware acceleration

Defining the roadmap for low-level silicon interfaces needed on every platform

Graphics, compute, rich media, vision, sensor and camera processing

Rigorous specifications AND conformance tests for cross-vendor portability

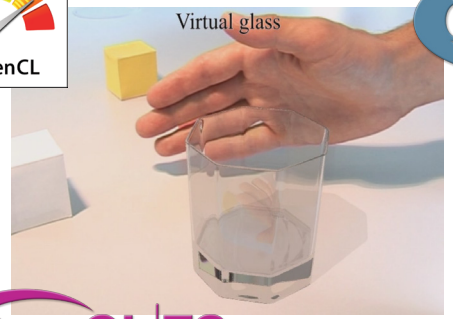
Acceleration APIs
BY the Industry
FOR the Industry

Well over a *BILLION* people use Khronos APIs *Every Day…*

http://accelerateyourworld.org/

Khronos Standards

OpenCL



OpenGL

OpenGL|ES

Visual Computing
- 3D Graphics
- Heterogeneous Parallel Computing

glTF

3D Asset Handling
- 3D authoring asset interchange
- 3D asset transmission format with compression

COLLADA

KHRONOS GROUP

Over 100 companies defining royalty-free APIs to connect software to silicon

EGL

OpenKCam

OpenVX

Sensor Processing
- Vision Acceleration
- Camera Control
- Sensor Fusion

StreamInput

WebGL

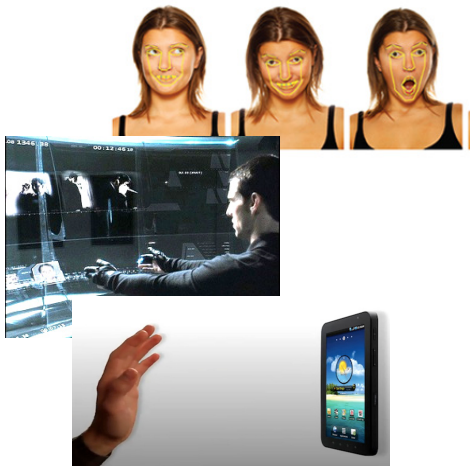
Acceleration in HTML5
- 3D in browser – no Plug-in
- Heterogeneous computing for JavaScript

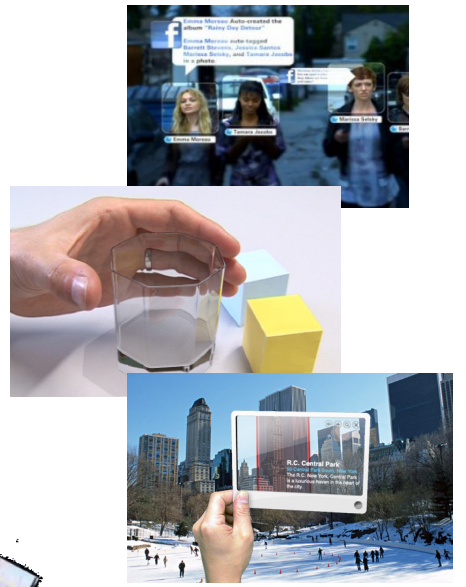
WebCL

KHRONOS GROUP

Mobile Vision Acceleration = New Experiences

Need for advanced sensors and the acceleration to process them





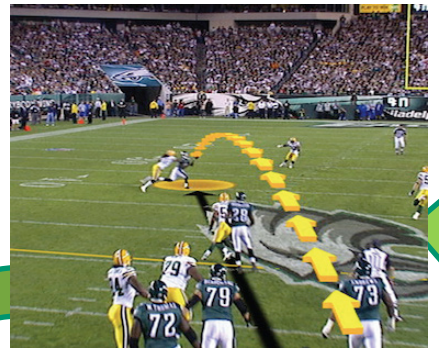
Computational Photography and Videography

Face, Body and Gesture Tracking

3D Scene/Object Reconstruction

Augmented Reality

Visual Computing = Graphics PLUS Vision



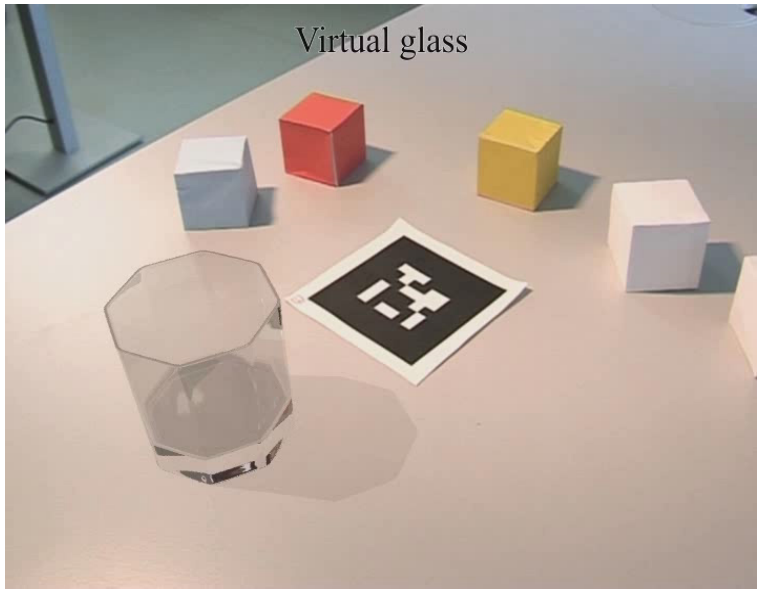
Vision Processing

Imagery

Data

Graphics Processing

Enhanced sensor and vision capability deepens the interaction between real and virtual worlds



Real-time GPU Compute

Research project on GPU-accelerated laptop

High-Quality Reflections, Refractions, and Caustics in Augmented Reality and their Contribution to Visual Coherence

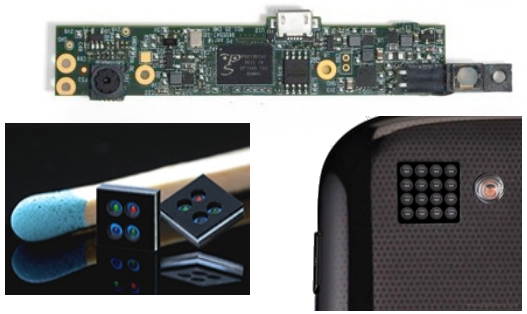
P. Kán, H. Kaufmann, Institute of Software Technology and Interactive Systems, Vienna University of Technology, Vienna, Austria

https://www.youtube.com/watch?v=i2MEwVZzDaA

Vision Pipeline Challenges and Opportunities

Growing Camera Diversity

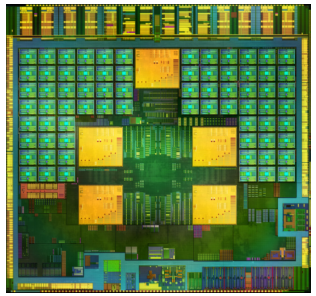
Capturing color, range and lightfields



- Camera sensors >20MPix
- Novel sensor configurations
- Stereo pairs
- Plenoptic Arrays
- Active Structured Light
- Active TOF

Diverse Vision Processors

Driving for high performance and low power



- Multi-core CPUs
- Programmable GPUs
- DSPs and DSP arrays
- Camera ISPs
- Dedicated vision IP blocks

Sensor Proliferation

Diverse sensor awareness of the user and surroundings

- Light / Proximity
- 2 cameras
- 3 microphones
- Touch
- Position
  - GPS
  - WiFi (fingerprint)
  - Cellular trilateration
  - NFC/Bluetooth Beacons
- Accelerometer
- Magnetometer
- Gyroscope
- Pressure / Temp / Humidity

19

EGL

Flexible sensor and camera control to generate required image stream

OpenKCam

EGL

Use best processing available for image stream processing - with code portability

OpenVX

Control/fuse vision data by/with all other sensor data on device

StreamInput

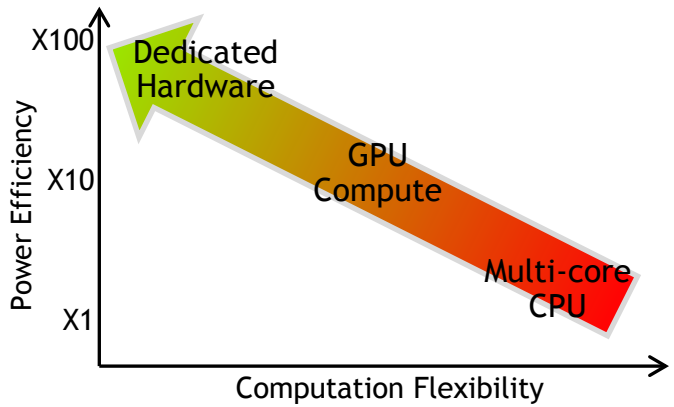
Vision Processing Power Efficiency

- **Depth sensors = significant processing**
  - Generate/use environmental information
- **Wearables will need 'always-on' vision**
  - With smaller thermal limit / battery than phones!
- **GPUs has x10 CPU imaging power efficiency**
  - GPUs architected for efficient pixel handling
- **Traditional cameras have dedicated hardware**
  - ISP = Image Signal Processor – on all SOCs today
- **SOCs have space for more transistors**
  - But can't turn on at same time = Dark Silicon
- **Potential for dedicated sensor/vision silicon**
  - Can trigger full CPU/GPU complex

But how to program specialized processors?
Performance and Functional Portability

Advanced Sensors

Wearables

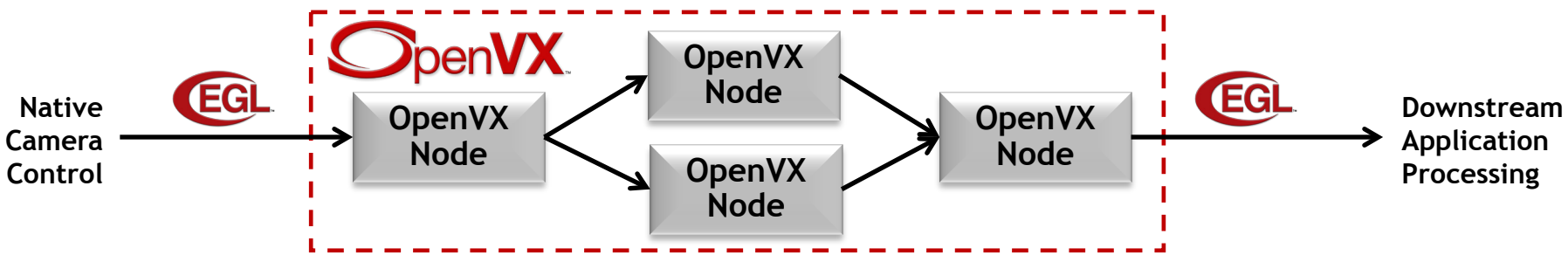


OpenVX - Power Efficient Vision Acceleration

- **Out-of-the-Box vision acceleration framework**
  - Enables low-power, real-time applications
  - Targeted at mobile and embedded platforms
- **Functional Portability**
  - Tightly defined specification
  - Full conformance tests
- **Performance portability across diverse HW**
  - Higher-level abstraction hides hardware details
    - ISPs, Dedicated hardware, DSPs and DSP arrays, GPUs, Multi-core CPUs …
- **Enables low-power, always-on acceleration**
  - Can run solely on dedicated vision hardware
  - Does not require full SOC CPU/GPU complex to be powered on

OpenVX Graphs - The Key to Efficiency

- **Vision processing directed graphs for power and performance efficiency**
  - Each Node can be implemented in software or accelerated hardware
  - Nodes may be fused by the implementation to eliminate memory transfers
  - Processing can be tiled to keep data entirely in local memory/cache
- **VXU Utility Library for access to single nodes**
  - Easy way to start using OpenVX by calling each node independently
- **EGLStreams can provide data and event interop with other Khronos APIs**
  - BUT use of other Khronos APIs are not mandated



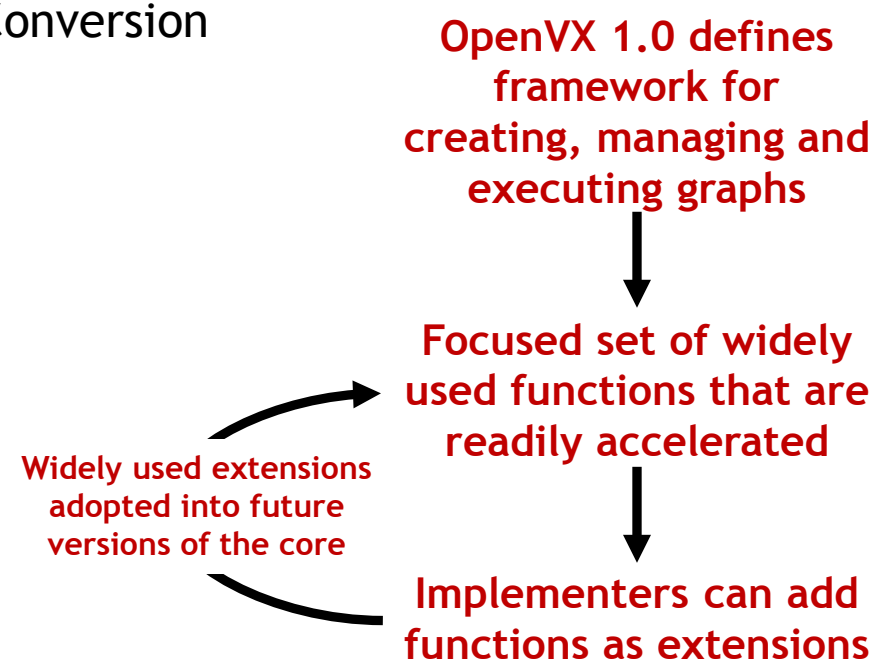
Example OpenVX Graph

OpenVX 1.0 Function Overview

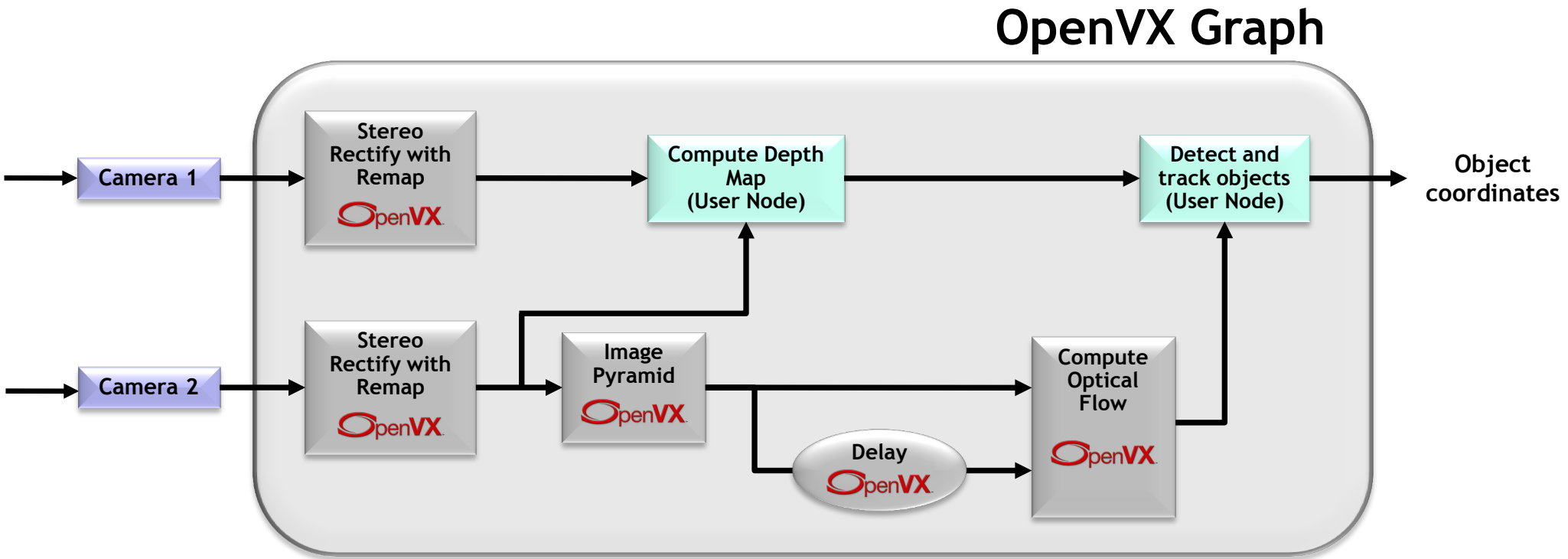
- **Core data structures**
  - Images and Image Pyramids
  - Processing Graphs, Kernels, Parameters
- **Image Processing**
  - Arithmetic, Logical, and statistical operations
  - Multichannel Color and BitDepth Extraction and Conversion
  - 2D Filtering and Morphological operations
  - Image Resizing and Warping
- **Core Computer Vision**
  - Pyramid computation
  - Integral Image computation
- **Feature Extraction and Tracking**
  - Histogram Computation and Equalization
  - Canny Edge Detection
  - Harris and FAST Corner detection
  - Sparse Optical Flow

OpenVX Specification Is Extensible

Khronos maintains extension registry



Example Graph - Stereo Machine Vision



Tiling extension enables user nodes (extensions) to also optimally run in local memory

OpenVX and OpenCV are Complementary

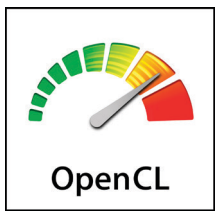
	OpenCV	OpenVX
Governance	Community driven open source with no formal specification	Formal specification defined and implemented by hardware vendors
Conformance	No conformance tests for consistency and every vendor implements different subset	Full conformance test suite / process creates a reliable acceleration platform
Portability	APIs can vary depending on processor	Hardware abstracted for portability
Scope	Very wide 1000s of imaging and vision functions Multiple camera APIs/interfaces	Tight focus on hardware accelerated functions for mobile vision Use external camera API
Efficiency	Memory-based architecture Each operation reads and writes memory	Graph-based execution Optimizable computation, data transfer
Use Case	Rapid experimentation	Production development & deployment

OpenVX Announcement

- **Finalized OpenVX 1.0 specification released October 2014**
  - www.khronos.org/openvx
- **Full conformance test suite and Adopters Program immediately available**
  - $20K Adopters fee ($15K for members) – working group reviews submitted results
  - Test suite exercises graph framework and functionality of each OpenVX 1.0 node
  - Approved Conformant implementations can use the OpenVX trademark
- **Khronos working on open source sample implementation of OpenVX 1.0**
  - Expected release on GitHub by end of 2014

Khronos APIs for Vision Processing

GPU Compute Shaders (OpenGL 4.X and OpenGL ES 3.1)
Pervasively available on almost any mobile device or OS
Easy integration into graphics apps – no vision/compute API interop needed
Program in GLSL not C
Limited to acceleration on a single GPU



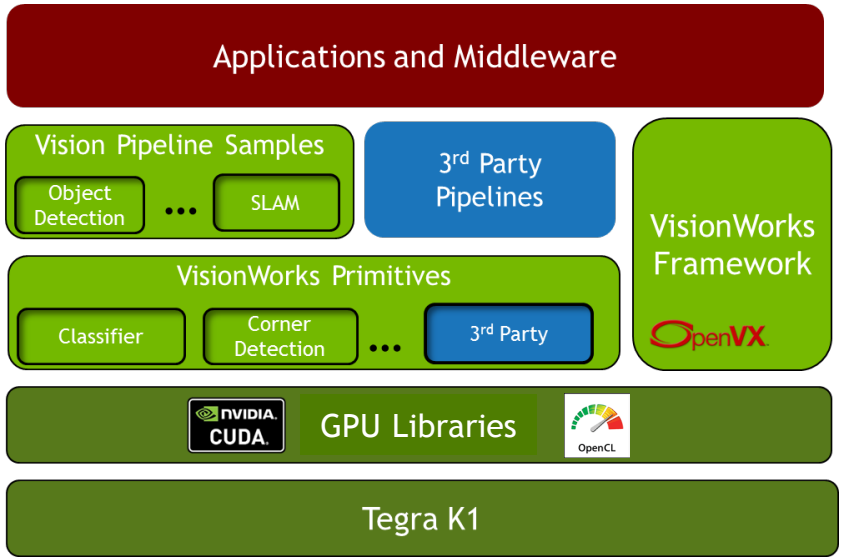
General Purpose Heterogeneous Programming Framework
Flexible, low-level access to any devices with OpenCL compiler
Single programming and run-time framework for CPUs, GPUs, DSPs, hardware
Open standard for any device or OS – being used as backed by many languages and frameworks
Needs full compiler stack and IEEE precision

Out of the Box Vision Framework - Operators and graph framework library
Can run some or all modes on dedicated hardware – no compiler needed
Higher-level abstraction means easier performance portability to diverse hardware
Graph optimization opens up possibility of low-power, always-on vision acceleration
Fixed set of operators – but can be extended

It is possible to use OpenCL or GLSL to build OpenVX Nodes on programmable devices!

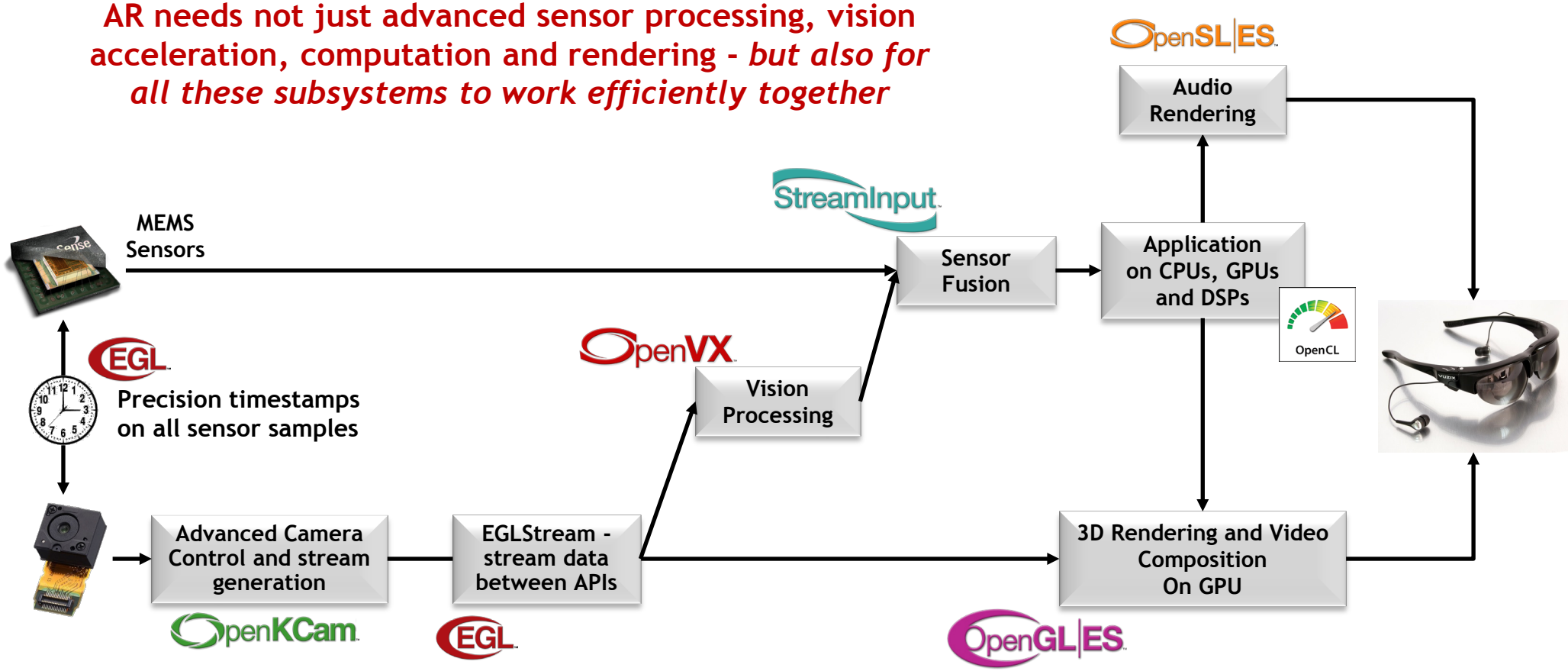
NVIDIA VisionWorks is Integrating OpenVX

- VisionWorks library contains diverse vision and imaging primitives
- Will leverage OpenVX for optimized primitive execution
- Can extend VisionWorks nodes through GPU-accelerated primitives
- Provided with sample library of fully accelerated pipelines



Khronos APIs for Augmented Reality

AR needs not just advanced sensor processing, vision acceleration, computation and rendering - *but also for all these subsystems to work efficiently together*



Summary

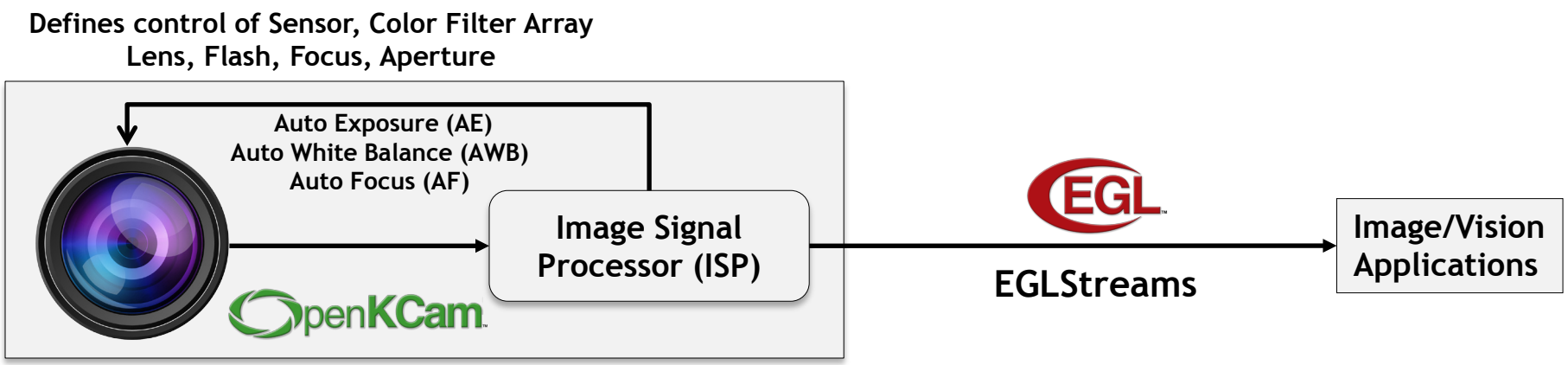
- **Khronos is building a trio of interoperating APIs for portable / power-efficient vision and sensor processing**
- **OpenVX 1.0 specification is now finalized and released**
  - Full conformance tests and Adopters program immediately available
  - Khronos open source sample implementation by end of 2014
  - First commercial implementations already close to shipping
- **Any company is welcome to join Khronos to influence the direction of mobile and embedded vision processing!**
  - $15K annual membership fee for access to all Khronos API working groups
  - Well-defined IP framework protects your IP and conformant implementations
- **More Information**
  - www.khronos.org
  - ntrevett@nvidia.com
  - @neilt3d

KHRONOS GROUP

Background Material

Need for Camera Control API - OpenKCAM

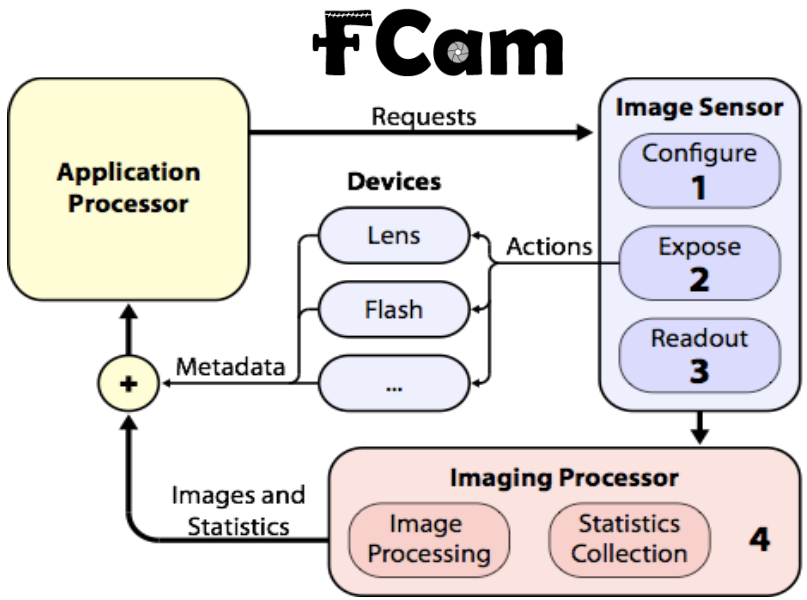
- **Advanced control of ISP and camera subsystem – with cross-platform portability**
  - Generate sophisticated image stream for advanced imaging & vision apps
- **No platform API currently fulfills all developer requirements**
  - Portable access to growing sensor diversity: e.g. depth sensors and sensor arrays
  - Cross sensor synch: e.g. synch of camera and MEMS sensors
  - Advanced, high-frequency per-frame burst control of camera/sensor: e.g. ROI
  - Multiple input, output re-circulating streams with RAW, Bayer or YUV Processing



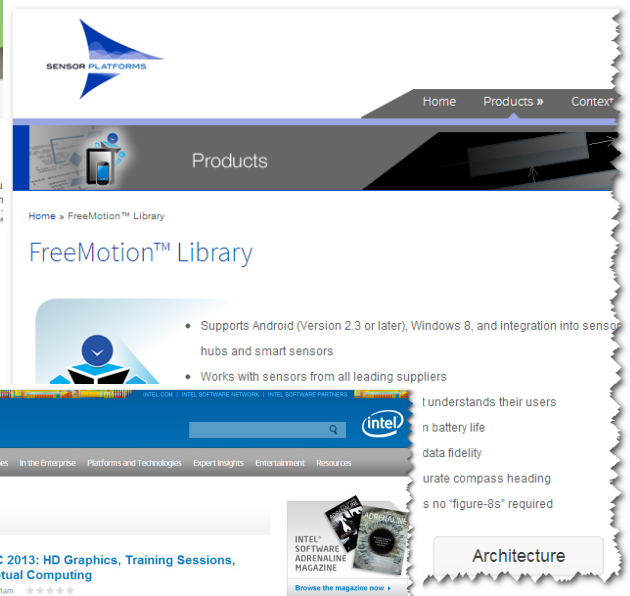
OpenKCAM is FCAM-based

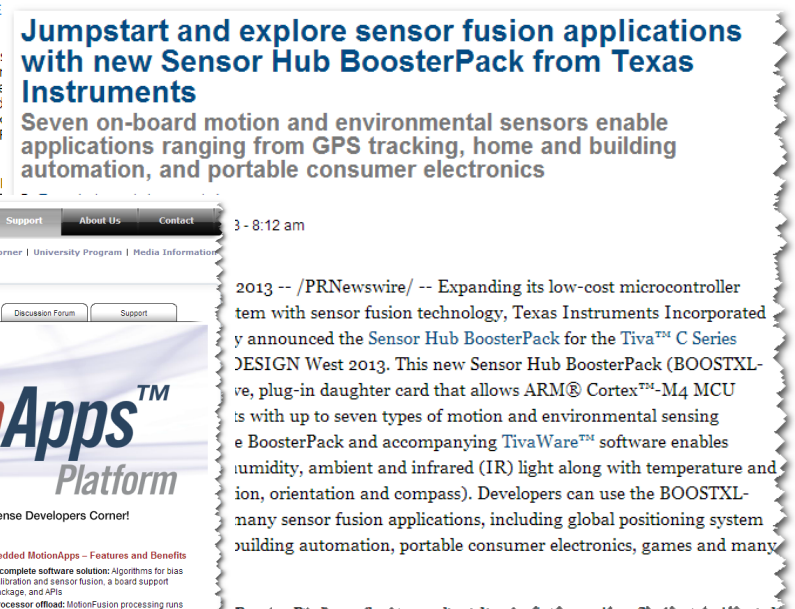
- **FCAM (2010) Stanford/Nokia, open source**
- **Capture stream of camera images with precision control**
  - A pipeline that converts requests into image stream
  - All parameters packed into the requests - no visible state
  - Programmer has full control over sensor settings for each frame in stream
- **Control over focus and flash**
  - No hidden daemon running
- **Control ISP**
  - Can access supplemental statistics from ISP if available
- **No global state**
  - State travels with image requests
  - Every pipeline stage may have different state
  - Enables fast, deterministic state changes

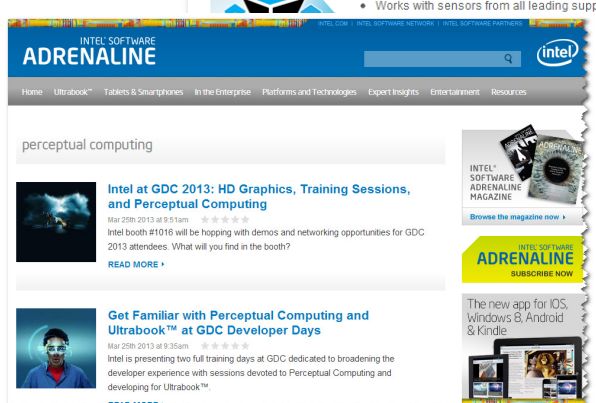
Khronos coordinating with MIPI on camera control and data formats

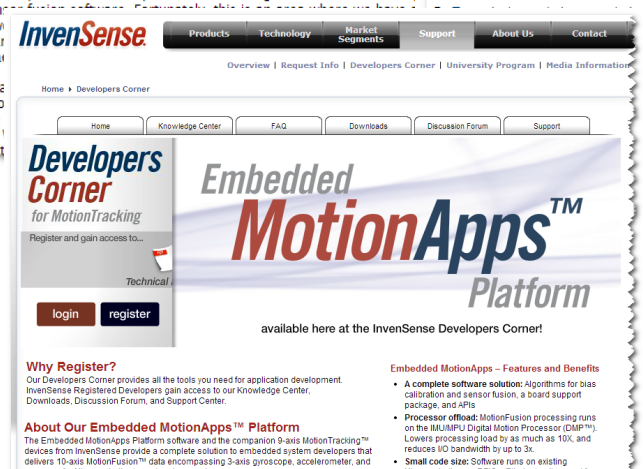


Sensor Industry Fragmentation …









Sensor Data Types

- **Raw sensor data**
  - Acceleration, Magnetic Field, Angular Rates
  - Pressure, Ambient Light, Proximity, Temperature, Humidity, RGB light, UV light
  - Heart rate, Blood Oxygen Level, Skin Hydration, Breathalyzer
- **Fused sensor data**
  - Orientation (Quaternion or Euler Angles)
  - Gravity, Linear Acceleration, Position
- **Contextual awareness**
  - Device Motion: general movement of the device: still, free-fall, …
  - Carry: how the device is being held by a user: in pocket, in hand, …
  - Posture: how the body holding the device is positioned: standing, sitting, step, …
  - Transport: about the environment around the device: in elevator, in car, …

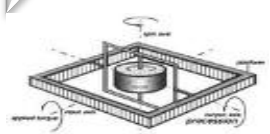
Low-level Sensor Abstraction API

Apps request semantic sensor information
StreamInput defines possible requests, e.g.
Read Physical or Virtual Sensors e.g. “Game Quaternion”
Context detection e.g. “Am I in an elevator?”

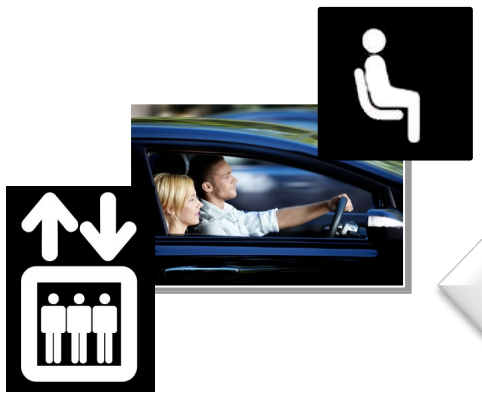
Apps Need Sophisticated Access to Sensor Data
Without coding to specific sensor hardware

StreamInput®

Sensor Discoverability
Sensor Code Portability




Advanced Sensors Everywhere
Multi-axis motion/position, quaternions, context-awareness, gestures, activity monitoring, health and environmental sensors

StreamInput processing graph provides optimized sensor data stream
High-value, smart sensor fusion middleware can connect to apps in a portable way
Apps can gain ‘magical’ situational awareness