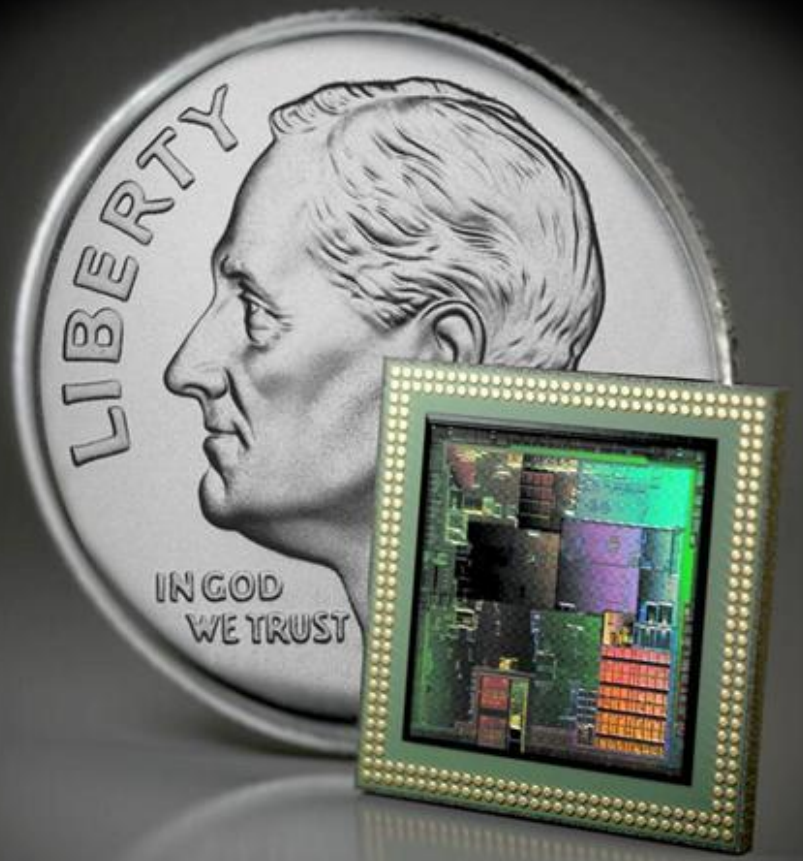
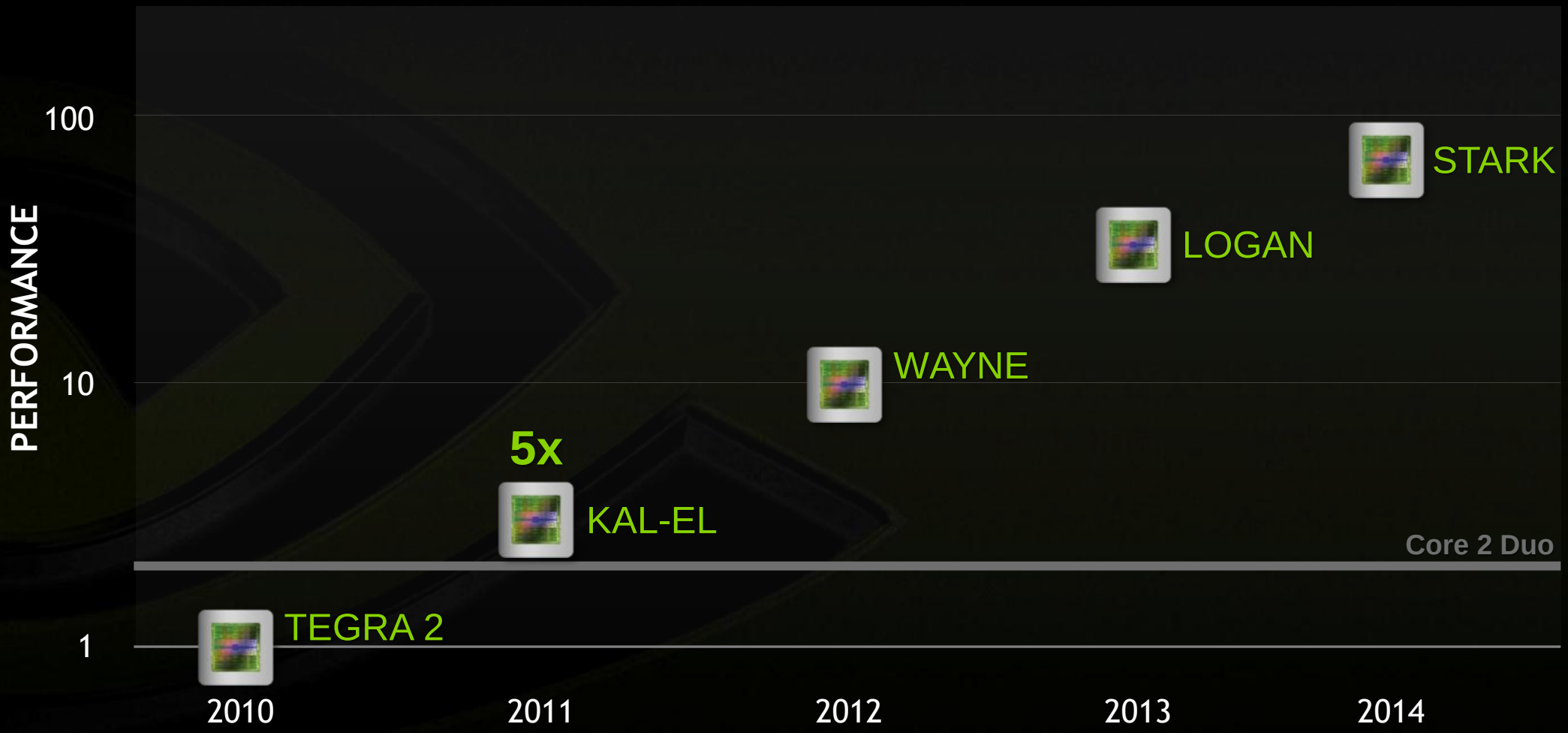
Project Kal-El update: Middleware for Mobile Visual Computing

Hot3D
High Performance Graphics 2011
David Luebke

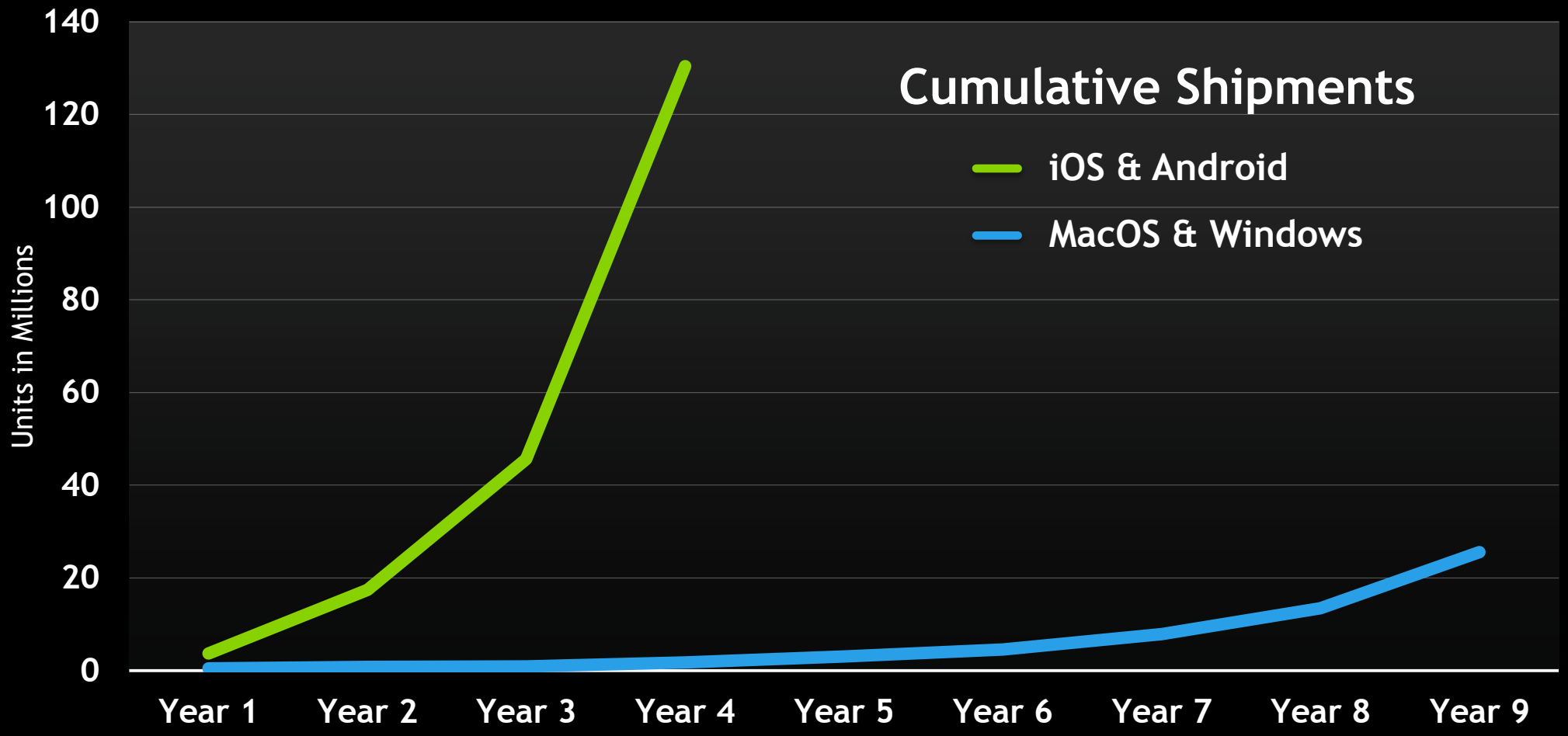
TEGRA
Highly Detailed Block Diagram



TEGRA: Highly Detailed Roadmap



Mobile Computing Momentum



Source: Gartner, Apple, NVIDIA

NVIDIA Mobile Computer Vision

- Focus on Mobile Vision Applications
- Optimize core algorithms for Tegra hardware

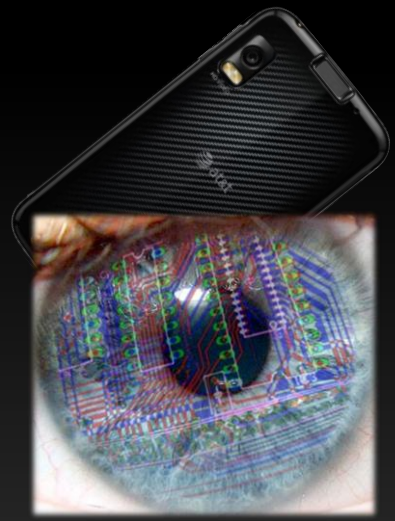
Graphics

Render Images From Scenes

Inverse Problems

Massively Parallel

Computer Vision



Understand Scenes From Images

Perception

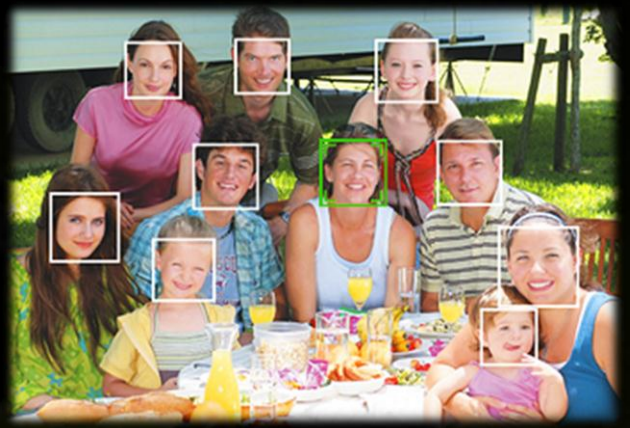
- *Where is the device?*
- *What's nearby?*
- *Who's nearby?*
- *What is the user doing?*

Interaction

Computer Vision = Smart Photography

Get the right shot, automatically

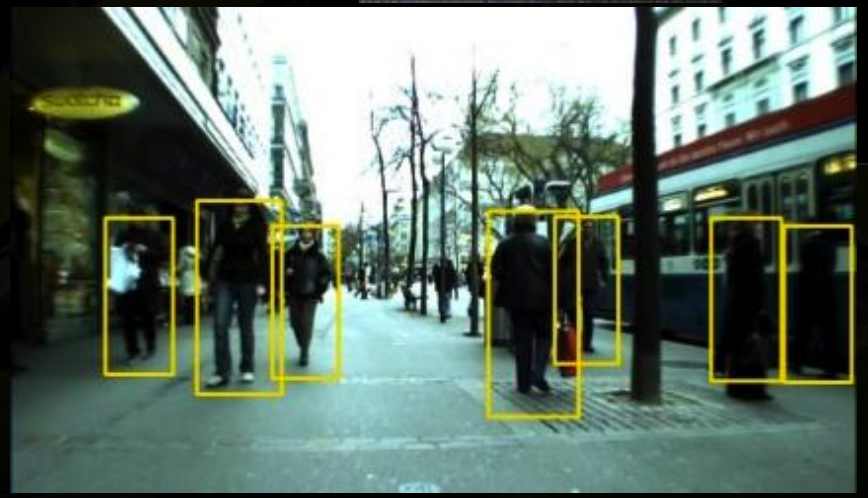
Face Detection



Scene Classification

Stabilization

Computer Vision = Vehicle Safety



Computer Vision = New Applications

Augmented Reality

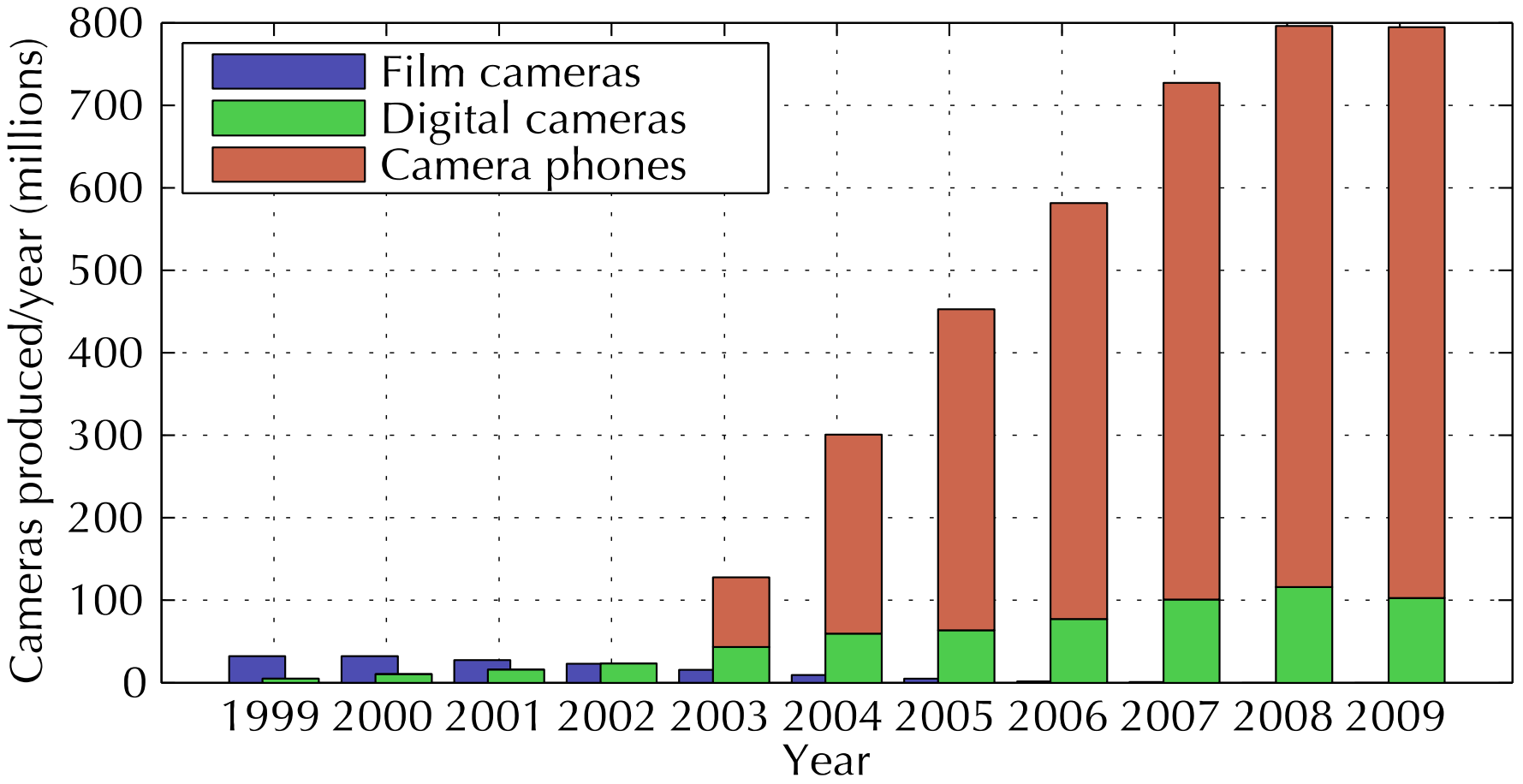
Augmented Reality Ghost Hunter (Argh)

Wordlens

Google Goggles

Gesture interfaces

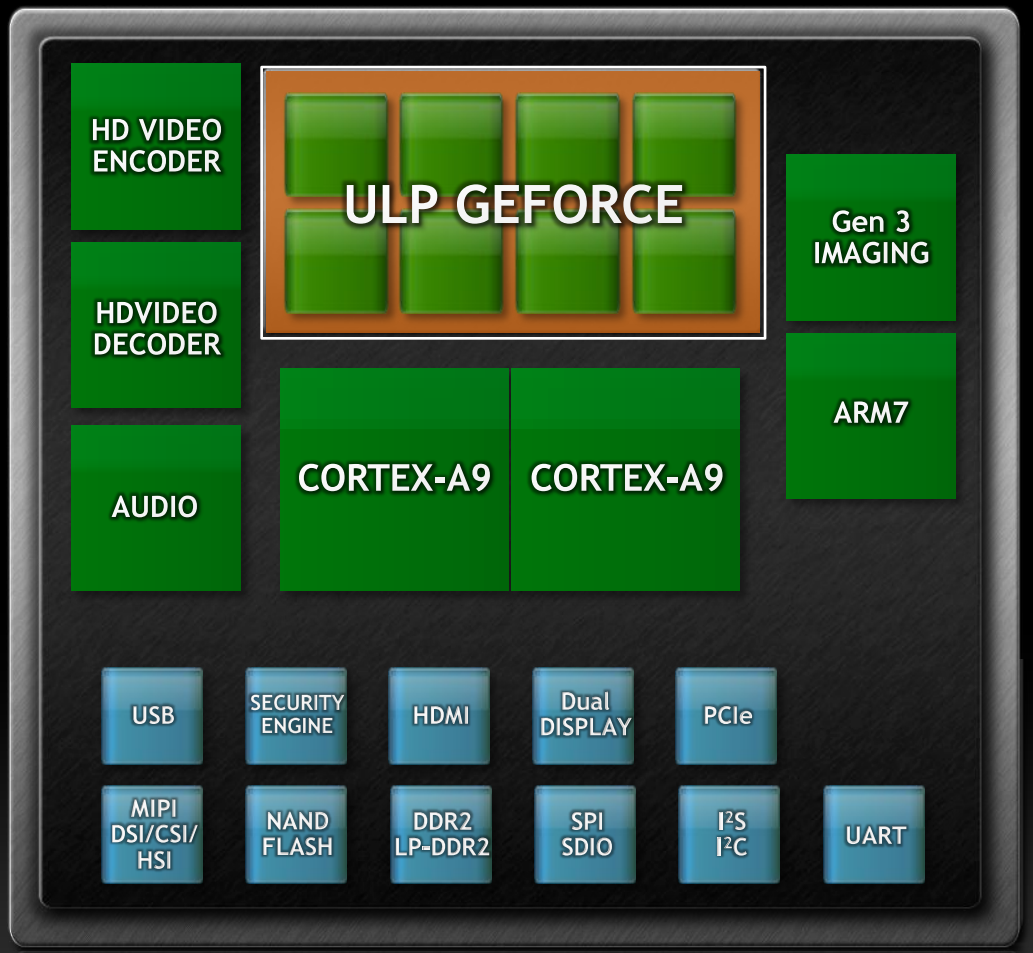
Traditional cameras / camera phones



Trends in camera phone sales

- Sales keep growing
  - 2003 — 85 million — 16% of the phones
  - 2010 — 805 million — 65% of the phones
  - 2014 — 1.3 billion — 85% of the phones
- Average resolution grows too
  - 2008 — 1 MP
  - 2009 — 2 MP
    - 2010 5+MP: ~ 13%, >100 million
  - 2014 — 5 MP
    - 2014 5+MP: ~ 42%, >550 million

Tegra 2 – Heterogeneous Multi-core



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

PROJECT KAL-EL

- World's first mobile quad-core CPU with NEON
- New 12-Core NVIDIA GPU
  with support for 3D stereo
- Extreme HD – 2560x1600
- 5X Tegra 2

DEMO

- “Glowball” demo
- Showing shading & geometry horsepower
- Not discussed here: Tegra-accelerated PhysX

Kal-el Reference Design

- Stereo 5 MP back Camera
- Front 2 MP camera

Want One?
Submit a project proposal!
http://research.nvidia.com/cvpr11

Android

- **Fastest Growing OS Ever**
- **36% of mobile market**
  (vs. iOS 26%)*

Use readily available consumer devices

*Source: http://blog.nielsen.com/nielsenwire/consumer/android-leads-u-s-in-smartphone-market-share-and-data-usage/

OpenCV

Thousands of Developers, Cross Platform API

- Open standard for Compute Vision
- Analogous to OpenGL for Graphics
- 12 years old, professionally developed
- Optimized for x86 SSE, CUDA GPU
- Over 3 Million Downloads!
- > 500 Algorithms

Common API for Server, Workstation, Desktop and now Mobile Platforms!

OpenCV Functionality Overview

Image processing

General Image Processing

Segmentation

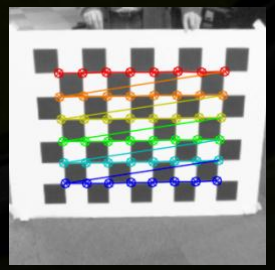
Machine Learning, Detection

Image Pyramids

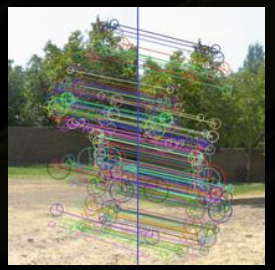
Transforms

Fitting

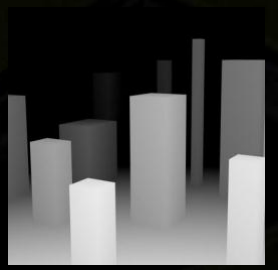
Video, Stereo, and 3D



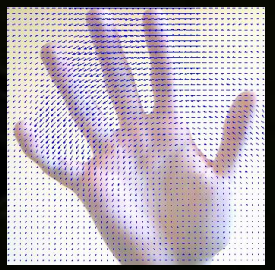
Camera Calibration



Features

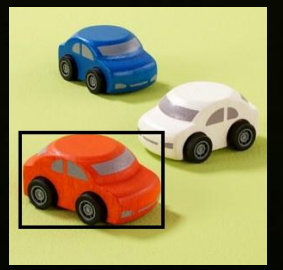


Depth Maps



Optical Flow

Inpainting



Tracking

OpenCV on Tegra

Optimized for ARM, Tegra & Android

Bringing the most popular Computer Vision library to the worlds most popular processor architecture on the fastest growing OS

Computational photography

- Overcome limitations of normal cameras
  - Usually by combining several images
- HDR imaging

- Panorama stitching

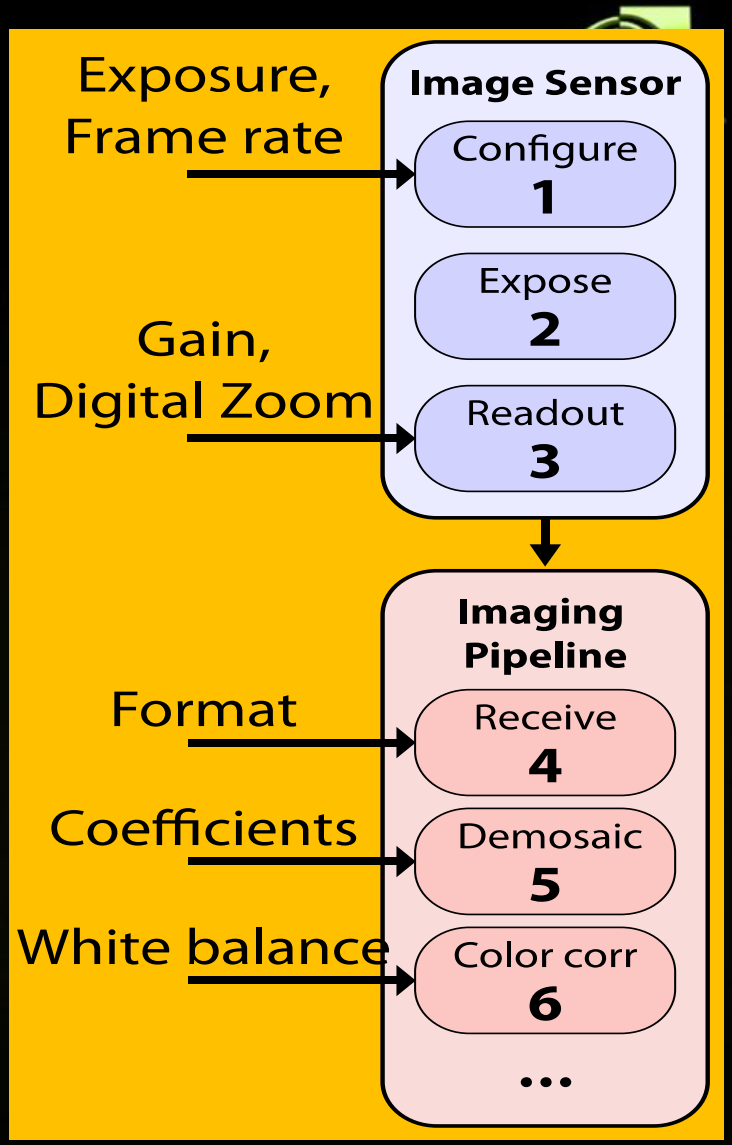
- Flash / no-flash imaging

Problem 1: Lack of access

- Camera control loops run in hardware or OS libraries
  - Cannot be disabled - surprises abound
  - No direct access to sensor, lens, flash
- Available settings limited
  - Often only exposure compensation
- Enough to write a Point & Shoot app, no more

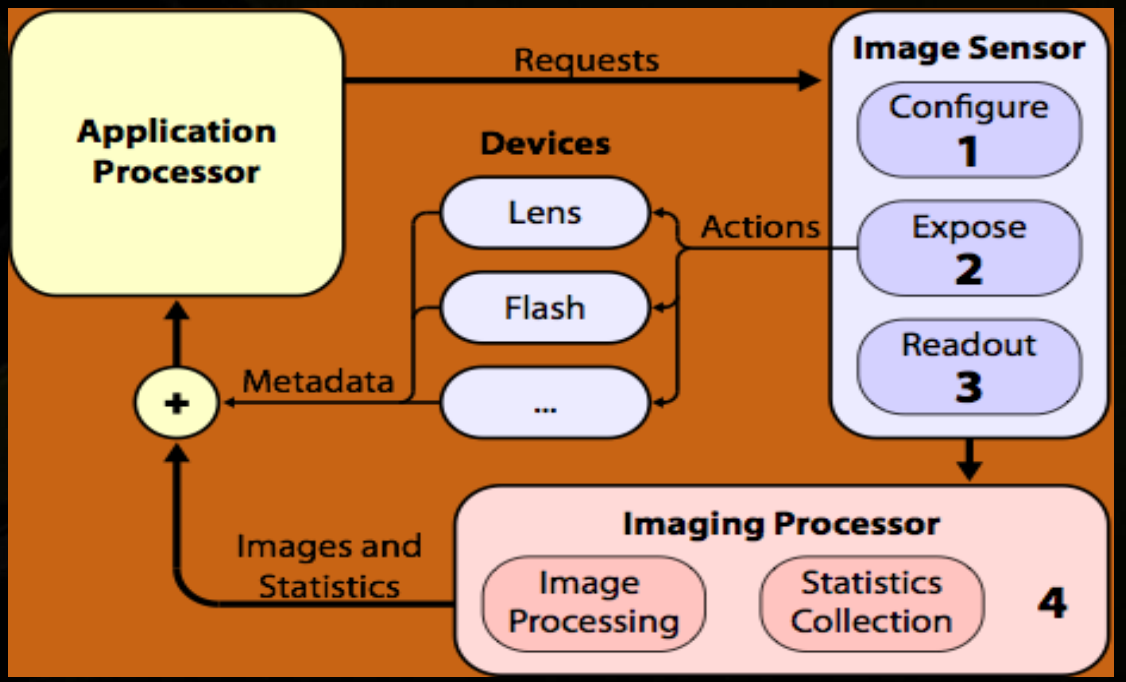
Problem 2: Wrong model

- Cameras are pipelines:
  - sensors have at least 3 stages
  - many post-processing stages, HW or SW
    - ➢ can have many images in flight
    - ➢ substantial latency
- Configuration is spread over the entire pipeline
  - if you change settings, they may affect different images!



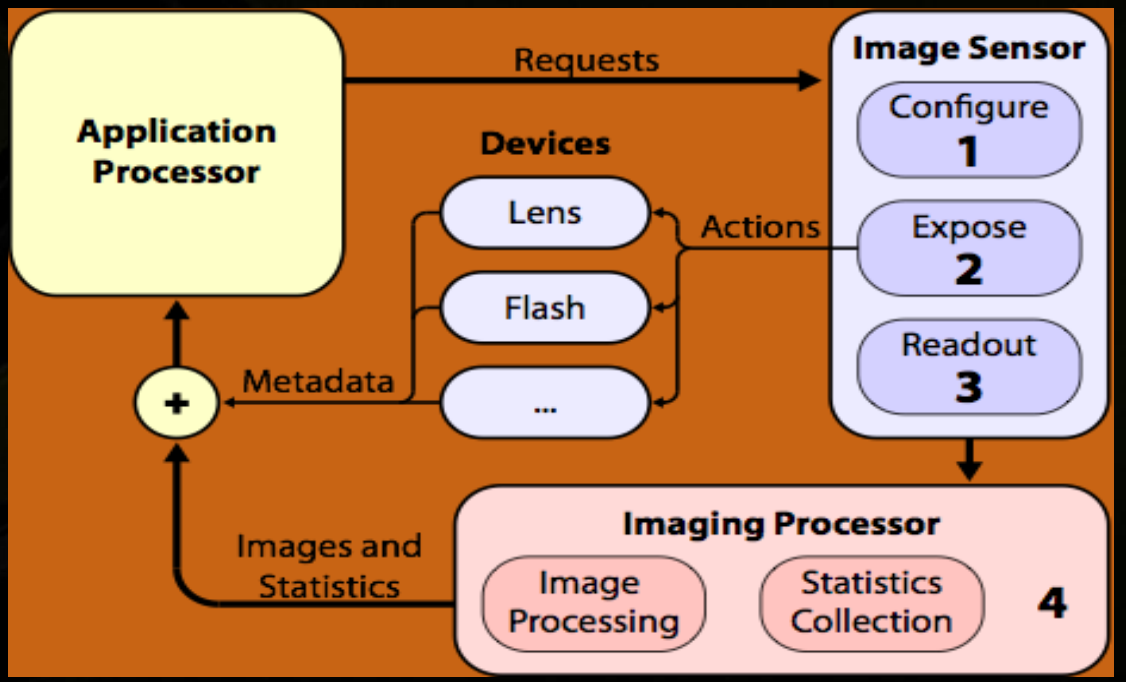
The FCam Architecture

- A software architecture for programmable cameras
  - that attempts to expose the maximum device capabilities
  - while remaining easy to program



Full control for the programmer

- Programmer has full control over sensor settings
  - and access to the supplemental statistics from ISP
- No hidden daemon running autofocus/metering
  - nobody changes the settings under you

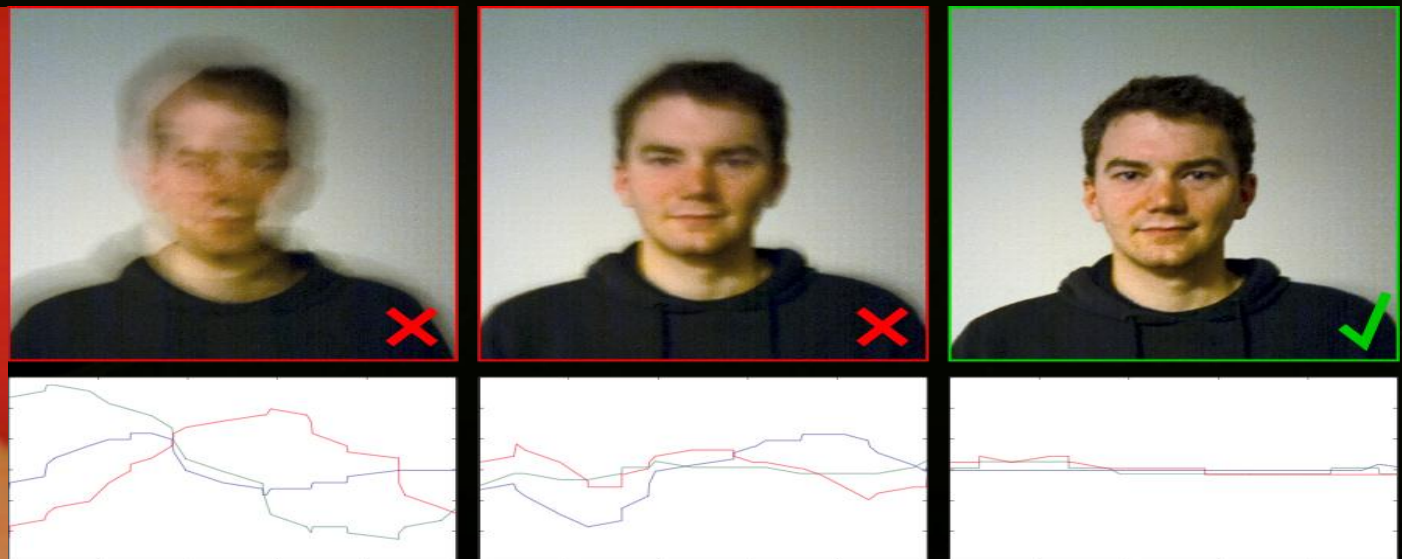


Double-flash example

- Using the F2 Frankencamera and two Canon flash units
  - control the cameras during the exposure
  - low intensity strobing followed by second curtain flash

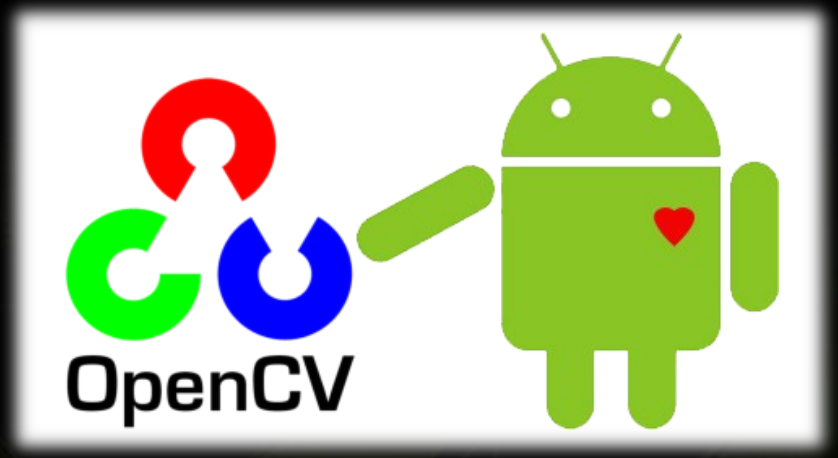
Lucky Imaging: Hand-held long exposures

- Holding camera steady for a long exposure is difficult
  - but sometimes you get lucky and hold it steady for a while
- We attached a 3-axis gyro to the N900
  - estimate if a captured image suffers from handshake
    - keep capturing if it does



FCam on Tegra: Baby steps

Using OpenCV for Android



- **OpenCV 2.3 for Android:**
  - **Native Android Camera Support**
  - **Multithreading**
  - **Java API (soon)**
  - **Tegra HW Optimizations (soon)**

Wiki with the latest information:

http://opencv.willowgarage.com/wiki/Android

Support/discussion

group: https://groups.google.com/group/android-opencv

THANK YOU

Thank You to

For continuous support and innovation in OpenCV

GPU TECHNOLOGY CONFERENCE

GPU Technology Conference Worldwide Events

GTC Workshop Japan, Tokyo, July 22, 2011

Co-hosted with the Tokyo Institute of Technology and bringing together top researchers, scientists and industry leaders to focus on critical research, trends and opportunities in GPU computing.

GTC China, Beijing, December 15-16, 2011

Focusing on the very latest scientific research and commercial applications in GPU computing.

GTC 2012, San Jose, CA, May 14-17, 2012

Advancing awareness of High Performance Computing and the transformational impact of GPUs.

www.gputechconf.com