Improving Mobile Device Battery Life in Imaging Scenarios

Brent Chartrand

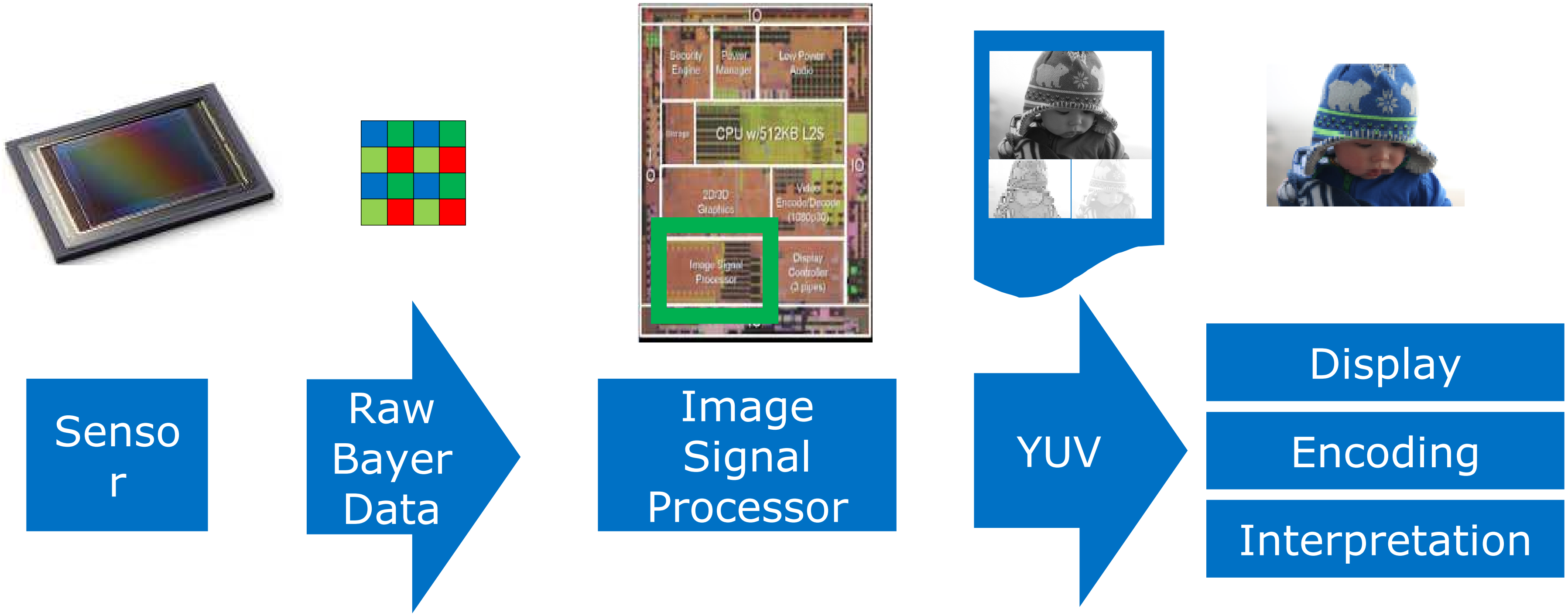
Intel Corporation

April, 2016

Imaging in Mobile is Everywhere

- Camera image quality is a significant component of user perception of device quality
  - Many, many marketing dollars spent on claims of superior image quality
- Five most popular cameras on Flickr are all phones
  (https://www.flickr.com/cameras/, April 8, 2016)
- Moving beyond simple snapshots
  - Video and still
  - Computational: HDR, panoramas, and more
  - Analysis: Face detection and recognition, scene analysis, security, …

What is imaging?



Senso
r

Raw
Bayer
Data

Image
Signal
Processor

YUV

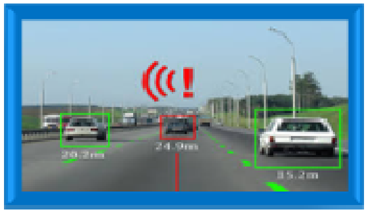
Display

Encoding

Interpretation

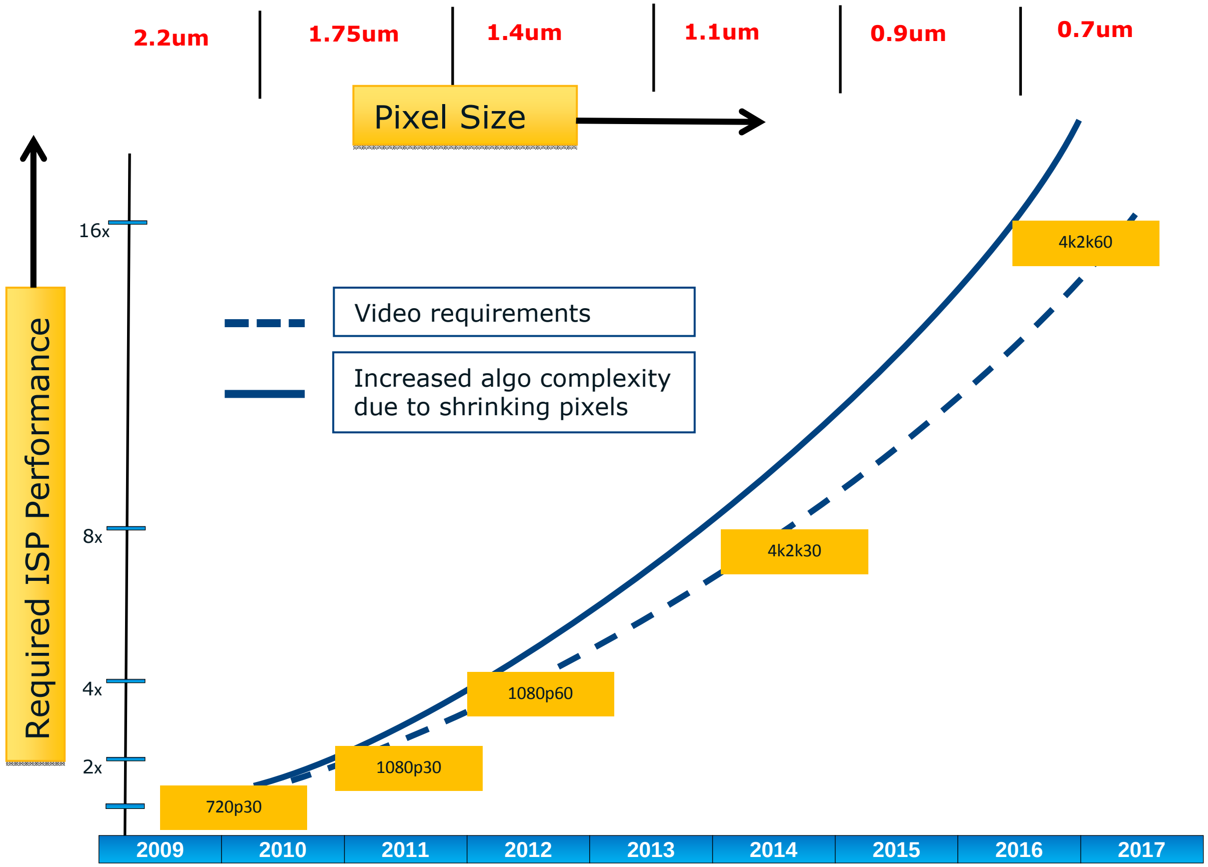
Taking unprocessed sensor data and turning it into

something a human likes to see



something a machine can interpret

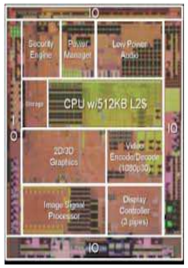
Resolutions are Growing
Pixels are shrinking

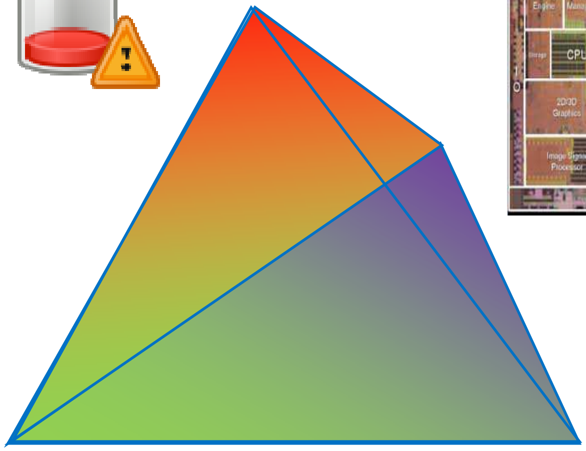


Challenges

Minimum Power

Minimum Silicon Area





Features

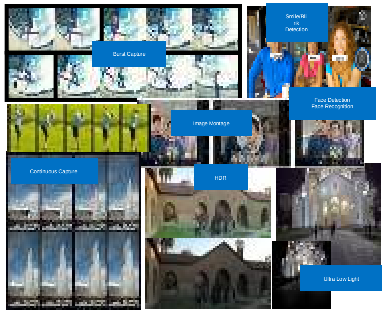


Image Quality

Users want:

- High resolution, high quality imaging
- With many features
- In low cost devices
- Without draining the battery

What affects imaging power

Imaging Parameters

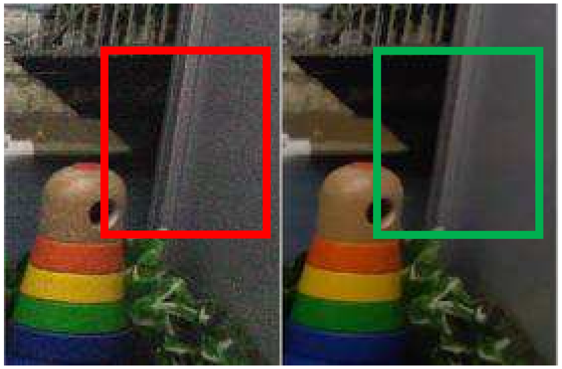
- Image resolution
- Bit depth

Imaging Hardware Architecture choices

- Pixels-per-clock
- Data transfer back and forth to system DRAM
- Algorithm implementation and hardware resource usage

Typical hardware design optimization

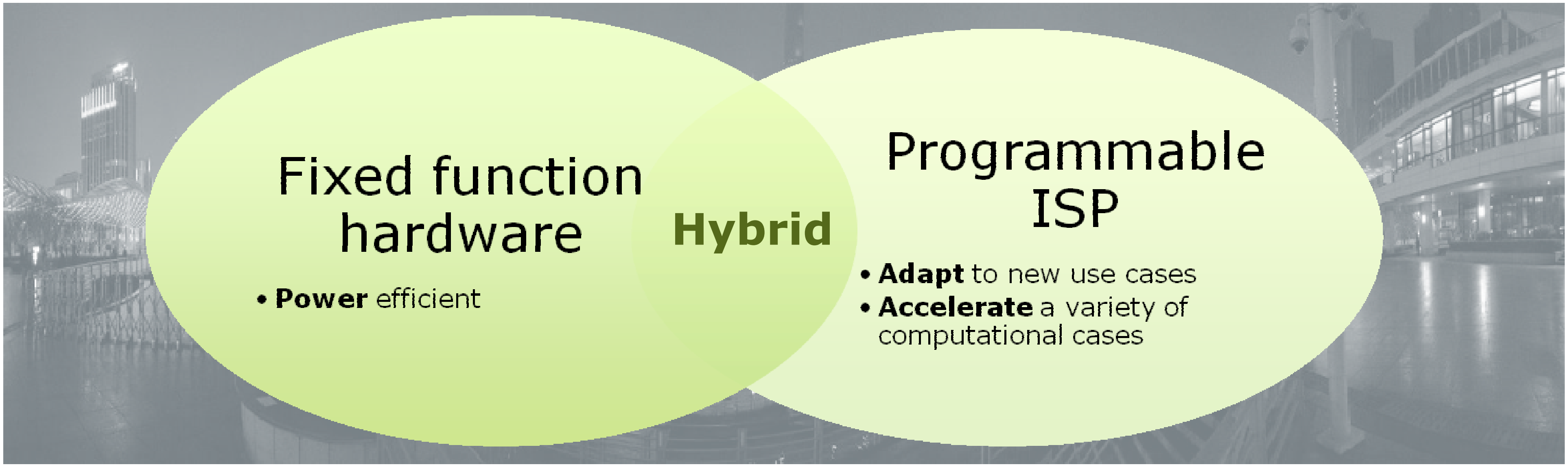
- Clock and power gating
- Process, library, and operating point choices



Noise Reduction

Low light processing

Flexibility versus Power



Fixed function hardware

- **Power** efficient

Hybrid

Programmable ISP

- **Adapt** to new use cases
- **Accelerate** a variety of computational cases

Hyper-spectral imaging

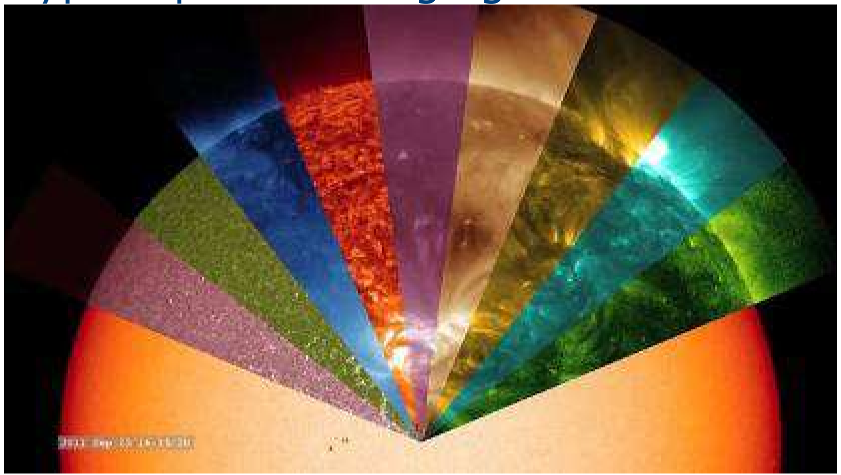
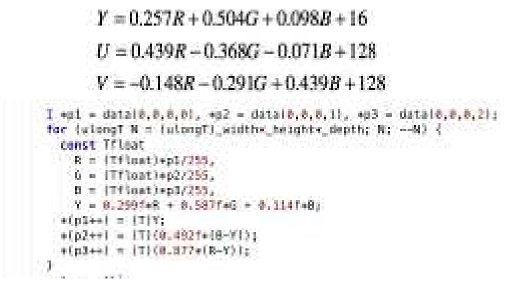


Image credit: NASA

Power Considerations in Imaging

- Pre-silicon
  - Advanced tools to quickly turn mathematical algorithms into efficient DSP code, and fixed-function gates
  - Modeling to examine tradeoffs between size, flexibility, and power



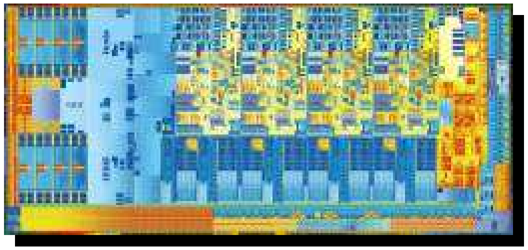
$$Y = 0.257R + 0.504G + 0.098B + 16$$

$$U = 0.439R - 0.368G - 0.071B + 128$$

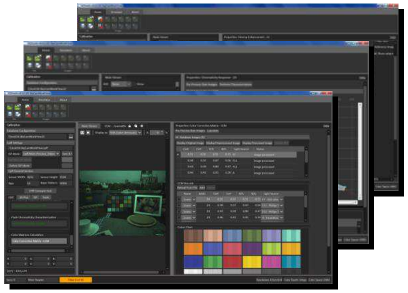
$$V = -0.148R - 0.291G + 0.439B + 128$$

```
T *p1 = data(0,0,0,0), *p2 = data(0,0,0,1), *p3 = data(0,0,0,2);
for (ulongT N = (ulongT)_width*_height*_depth; N; --N) {
  const Tfloat
    R = (Tfloat)*p1/255,
    G = (Tfloat)*p2/255,
    B = (Tfloat)*p3/255,
    Y = 0.299f*R + 0.587f*G + 0.114f*B;
  *(p1++) = (T)Y;
  *(p2++) = (T)(0.492f*(B-Y));
  *(p3++) = (T)(0.877*(R-Y));
}
```

- In silicon
  - Power efficient logic design



- Platform
  - Low power, low latency fabric and memory
  - Other active components can interfere
- Developers
  - Tools to efficiently use the hardware blocks available to produce complete solutions



Summary

Imaging

- Is a high value feature on the platform
- Continues to evolve rapidly

Imaging Power

- Can be high if not architected and implemented well
- Has many possible optimization points
- Demands many tradeoffs
- Requires tools, models, and expertise in many domains to analyze