System support for efficient multi-resolution visual computing on mobile systems

Robert LiKamWa
http://meteor.ame.asu.edu
Arizona State University
Vision is power hungry especially at high resolutions
Vision doesn’t always need high-resolution images

0.1 MP suffices for object recognition (e.g., GoogleNet)
We can exploit this if image sensing is energy-proportional.

Energy-per-frame

Resolution

(0.1 MP, 3 fps)

(1 MP, 3 fps)
We can exploit this if image sensing is energy-proportional

But it’s not.

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Image sensor power breakdown

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys '13]
Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys '13]
Idle power limits energy-proportionality

Average Power

(0.1 MP, 3 fps)

280 mW

(1 MP, 3 fps)

320 mW

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Driver-based power optimization:

(1) Aggressive power management

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys '13]
Driver-based power optimization:

(2) Pixel clock frequency optimization

\[ f_{\text{best}} = \sqrt{\frac{c_1 \times N}{c_2 \times T_{\text{exp}}}} \]

- Slower clock
- Faster clock

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys '13]
Energy-proportionality before driver-based management

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Energy-proportionality after driver-based management

Energy characterization and optimization of image sensing toward continuous mobile vision [MobiSys ’13]
Energy-proportionality after driver-based management

![3D Graph]

- Power (mW): 20 mW at (0.1 MP, 3 FPS)
- Frame rate (FPS): 280 mW at (0.1 MP, 3 FPS)
- Pixel Count (x10^5)
Vision is power hungry, especially at high resolutions

Energy consumption of augmented reality marker detection
On Jetson Tegra X2 board
However, resolution reconfiguration incurs latency penalty

Linux V4L2 (Video4Linux2) + Jetson TX2

Android 8.0 + Nexus 5X

iOS 12 + iPhone X

Notice the visible gap during reconfiguration

Where does resolution reconfiguration latency come from?

Hardware? Operating system?
Hardware is not the culprit

- Hardware register values are effective by the next frame

[From AR0330 datasheet]
In the operating system, resolution reconfiguration undergoes a **sequential procedure** inside the media framework which requires the application to invoke several **expensive system calls**.

- **Open device**
- **Set sensor format**
  - `ioctl(VIDIOC_S_FMT)` sets sensor output format
- **Request/map buffers**
  - `ioctl(VIDIOC_REQBUFS)` and **`mmap`** new sets of buffers
  - `ioctl(VIDIOC_STREAMON)` start new streams
- **Start streaming**
- **Process image**
- **Resolution request?**
  - **No**
  - **Yes**
    - **Application sends a resolution request**
    - `ioctl(VIDIOC_STREAMOFF)` **turns off current streams**
    - **`munmap` and free buffers**
- **Stop streaming**
- **Release buffers**
Aspirations for a reconfigurable media framework

1: Preserve the pipeline of existing frames

2: Change resolution immediately, effective in the next capture

3: Minimize format synchronizations across the video system stack
We introduce the Banner media framework

Applications can reconfigure sensor resolution through only one `ioctl()` call with Banner.
Banner avoids repeated reconfiguration procedure

Format-oblivious memory management

Parallel reconfiguration

Reconfiguration in Banner

Parallel reconfiguration

- Enact thread-level concurrency
- Time request to happen within reconfiguration timing budget

\[ T_{\text{budget}} = T_{\text{interval}} - T_{\text{capture}} \]
Format-oblivious memory management

- One time buffer allocation
- Format-oblivious frame delivery
Banner avoids repeated reconfiguration procedure

Format-oblivious memory management

Parallel reconfiguration

Reconfiguration in Banner

Banner realizes seamless resolution reconfiguration

Legacy V4L2

Banner

Object moves closer. Switching to lower resolution...

Banner media framework for seamless resolution reconfiguration

Format-oblivious memory management

Parallel reconfiguration

Driver-based management for energy-proportional image capture

Multi-resolution visual computing
Dr. Robert LiKamWa
Ongoing efforts in multi-resolution visual computing systems

Contextual use cases
- Augmented Reality/Mixed Reality foveated rendering, foveated sensing
- Adaptive neural networks to work on variable-size image streams

Software-defined imaging systems
- Variable bit depth
- Region-of-interest
- Sensor control loops
- Cloud integration (5G)

Fine-grained reconfigurability
- Explore multi-resolution not just across frames, but within frames
- Explore variable temporal resolution

Reconfigurable resolution framework support

http://meteor.ame.asu.edu