

Computational photography & the Stanford Frankencamera

CS 178, Spring 2011

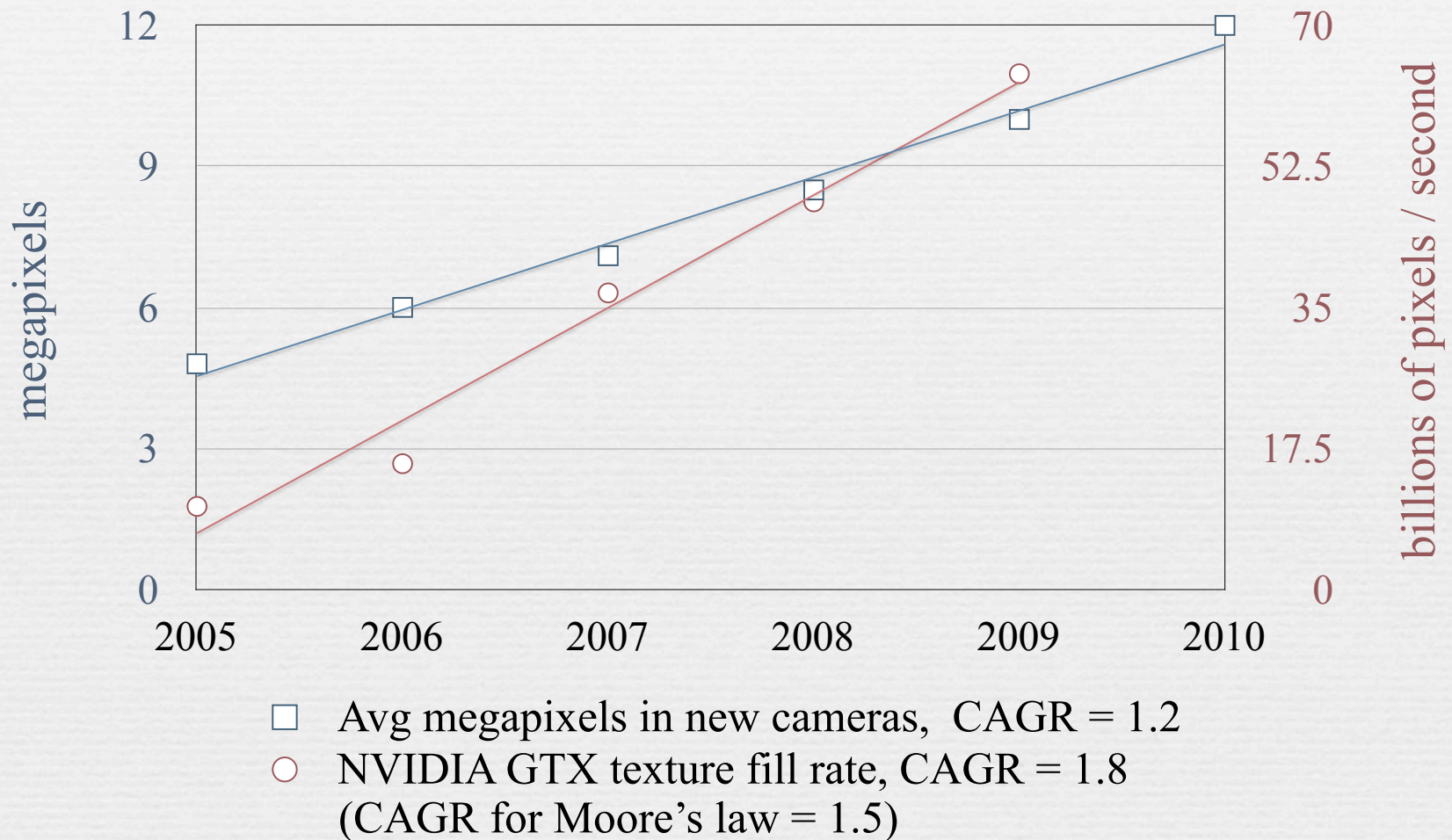


Marc Levoy
Computer Science Department
Stanford University

The future of digital photography

- ◆ the megapixel wars are over (and it's about time)
- ◆ computational photography is the next battleground in the camera industry (it's already starting)

Premise: available computing power in cameras is rising faster than megapixels



◆ this “headroom” permits more computation per pixel, or more frames per second, or less custom hardware

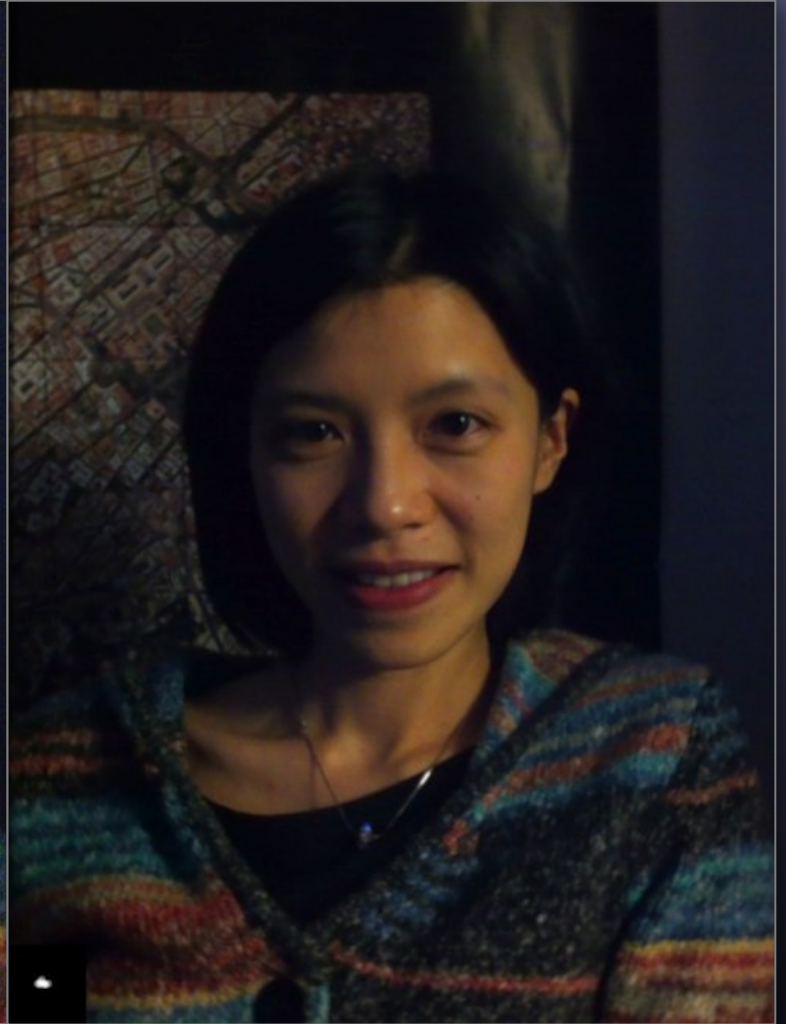
The future of digital photography

- ◆ the megapixel wars are over (long overdue)
- ◆ computational photography is the next battleground in the camera industry (it's already starting)
- ◆ how will these features appear to consumers?
 - standard and invisible
 - standard and visible (and disable-able)
 - aftermarket plugins and apps for your camera

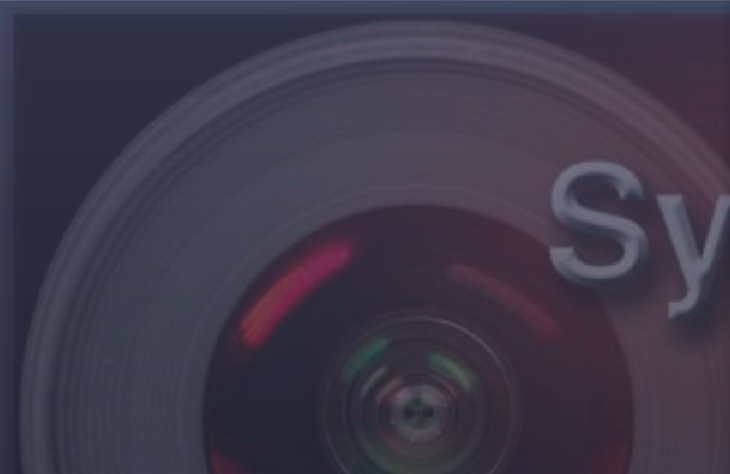
The future of digital photography

- ◆ the megapixel wars are over (long overdue)
- ◆ computational photography is the next battleground in the camera industry (it's already starting)
- ◆ how will these features appear to consumers?
 - standard and invisible
 - standard and visible (and disable-able)
 - aftermarket plugins and apps for your camera
- ◆ traditional camera makers won't get it right
 - they'll bury it on page 120 of the manual (like Scene Modes)
 - the mobile industry will get it right (indie developers will help)

SynthCam



Navigation and control icons: a refresh symbol, the number '2', a Japanese flag, a download icon, and a settings gear.



SynthCam is an app for the

Price: \$0.99

Video explanation of SynthCam

If you're a first-time user of SynthCam, start with this video. It explains what SynthCam does and how to use it. If the video below doesn't play correctly on this web page, or if you want to view it at full resolution, you can find it at <http://www.youtube.com/watch?v=b0zLgCF42Vk>. Note that the video is based on version 1.0; the user interface has changed slightly since then.



Multi-point focusing in SynthCam Version 2.0

This additional video explains how to use the multi-point focusing capabilities of Version 2.0, and how to use them to create a tilt-shift photograph that makes the world look like a miniature model. If the video doesn't play or if you want to see it at full resolution, go to <http://www.youtube.com/watch?v=S1tLoFVI6a8>.



Why have traditional camera makers been so slow to embrace computational photography?

(soapbox mode ON)

- the camera industry is secretive
 - no flow of workers between companies and universities
 - few publications, no open source software community
- camera companies sell hardware, not software
 - many are not comfortable with Internet ecosystems
- some computational techniques are still not robust
 - partly because researchers can't test them in the field

(soapbox mode OFF)

Camera 2.0

Marc Levoy
Computer Science Department
Stanford University



Camera 2.0

Marc Levoy
Computer Science Department
Stanford University



Camera 2.0

Marc Levoy
Computer Science Department
Stanford University



Camera 2.0

Marc Levoy
Computer Science Department
Stanford University



The Stanford Frankencameras



Frankencamera F2

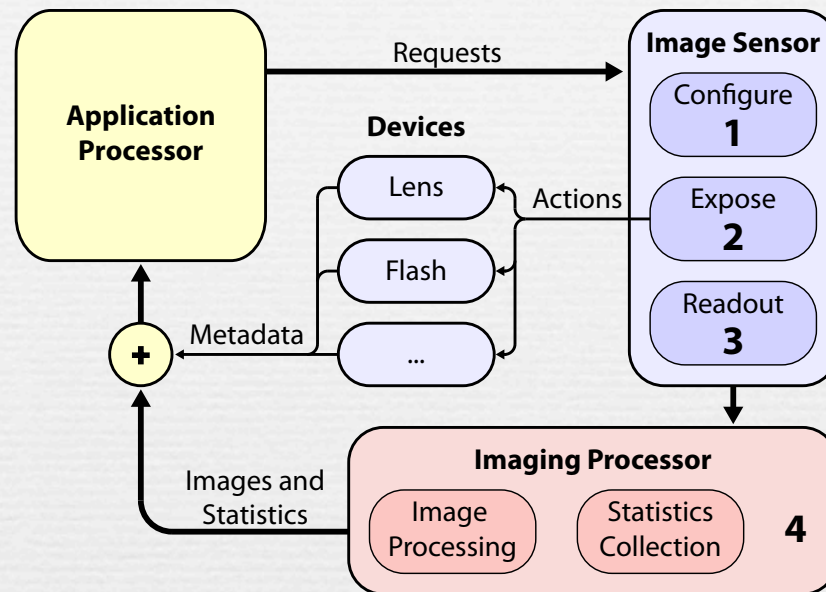


Nokia N900 "F"

- ◆ facilitate research in experimental computational photography
- ◆ for students in computational photography courses worldwide
- ◆ proving ground for plugins and apps for future cameras

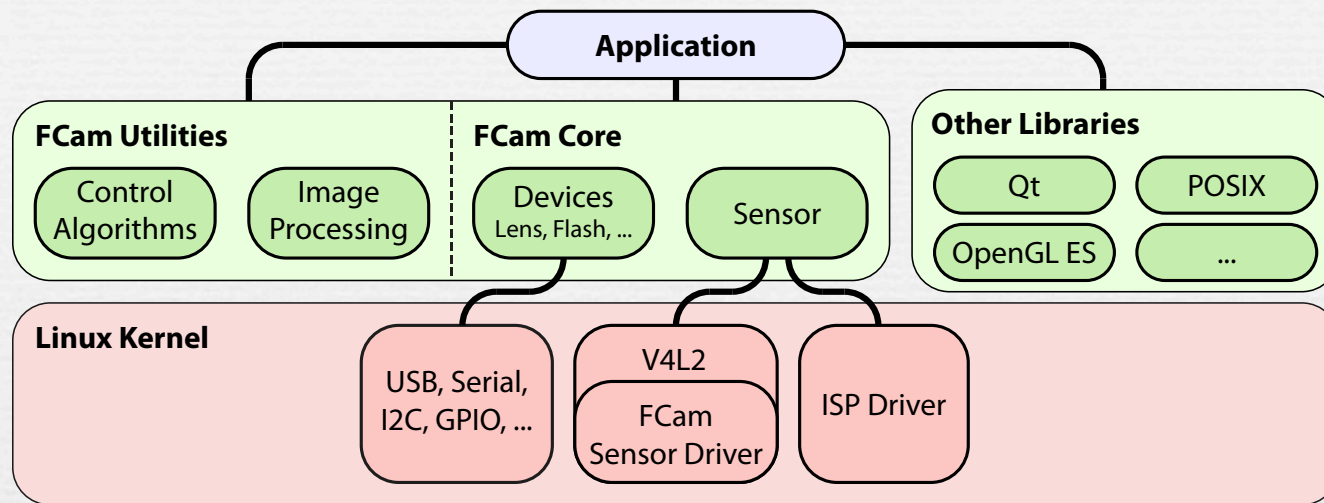
Frankencamera architecture:

how do you allow different camera settings on each frame?



- ◆ there's isn't a notion of *current camera settings*
- ◆ instead, a *pipeline* converts requests into frames
- ◆ a *request* includes all settings for one frame
 - exposure, ISO, zoom, focus, white balance, resolution, ROI, flash
- ◆ a returned *frame* contains an image, a frame ID, and the settings used to capture that frame

Frankencamera software: the FCam API



- ◆ lets *developers* write to a common abstraction across all cameras
- ◆ lets *camera designers* innovate and accelerate under the hood
- ◆ standard C++, cross-compiled using `gcc`, loaded using `ssh`, debugged using `gdb`, etc.
- ◆ principle of least surprise

Example #1: capture an HDR stack

```
Sensor sensor;  
Shot low,med,high;  
  
low.exposure = 1/80.;  
med.exposure = 1/20.;  
high.exposure = 1/5.;  
  
sensor.capture(low);  
sensor.capture(med);  
sensor.capture(high);  
  
Frame frames[3];  
frames[0] = sensor.getFrame();  
frames[1] = sensor.getFrame();  
frames[2] = sensor.getFrame();  
  
fused = mergeHDR(frames);
```

Ex #2: strobing flash-noflash

```
Sensor sensor;  
Flash flash;  
vector<Shot> burst(2);
```

```
burst[0].exposure = 1/200.;  
burst[1].exposure = 1/30.;
```

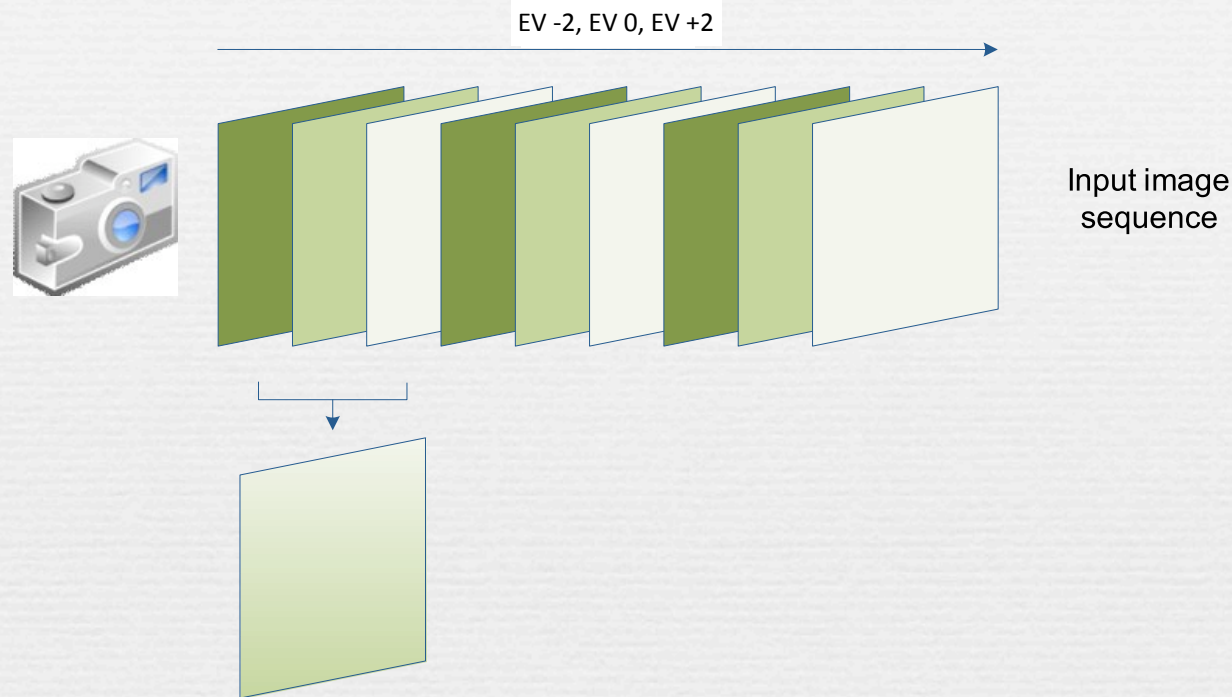
```
Flash::FireAction fire(&flash);  
fire.time = burst[0].exposure/2;  
burst[0].actions.insert(fire);
```

```
sensor.stream(burst);
```

```
while (1) {  
    Frame flashFrame =  
        sensor.getFrame();  
    Frame noflashFrame =  
        sensor.getFrame();  
}
```

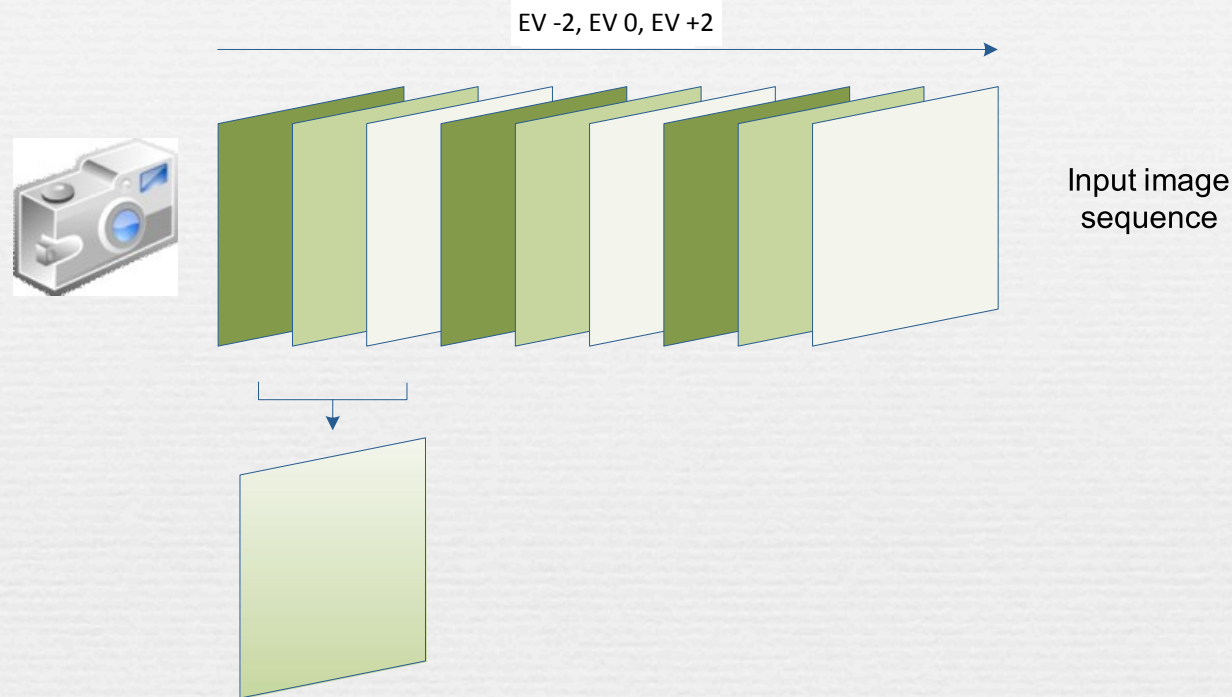


Application #1: real-time HDR viewfinder



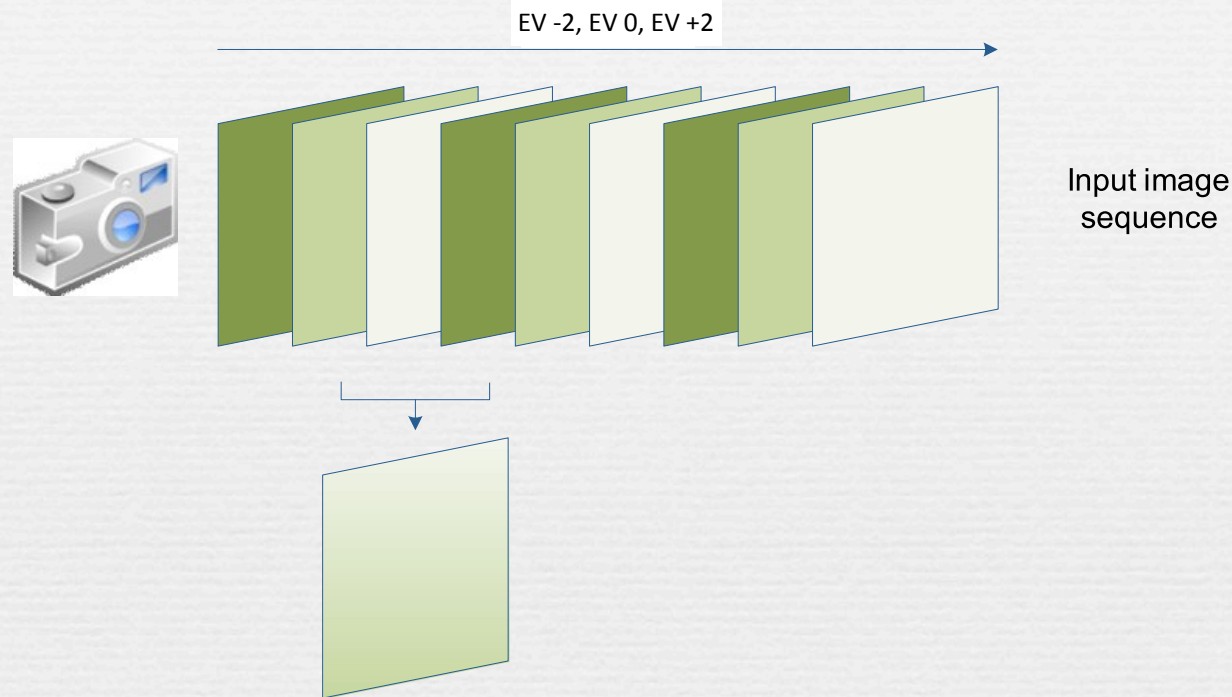
- ◆ cycles through three different exposure times at 40fps
- ◆ moving 3-frame window merged to HDR and tone mapped

Application #1: real-time HDR viewfinder



- ◆ cycles through three different exposure times at 40fps
- ◆ moving 3-frame window merged to HDR and tone mapped

Application #1: real-time HDR viewfinder



- ◆ cycles through three different exposure times at 40fps
- ◆ moving 3-frame window merged to HDR and tone mapped

Application #1: real-time HDR viewfinder



single exposure



3-exposure HDR

Application #2: dual flash units

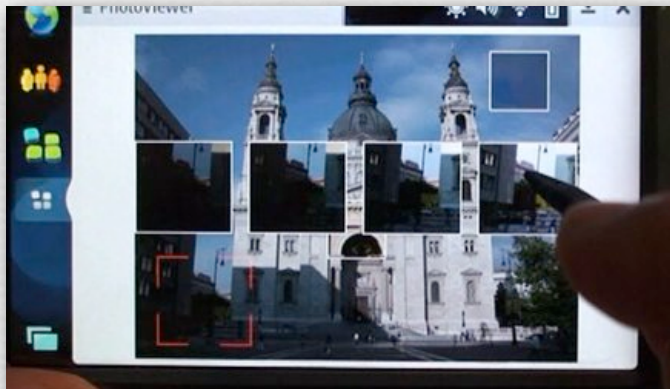


- Canon 430EX (smaller flash) strobed continuously
- Canon 580EX (larger flash) fired once at end of exposure



Rethinking the user interface

- ◆ Don't ask, "Did I get the shot?"
- ◆ Instead ask, "Did I capture enough imagery?"

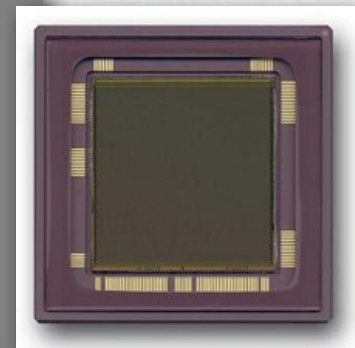
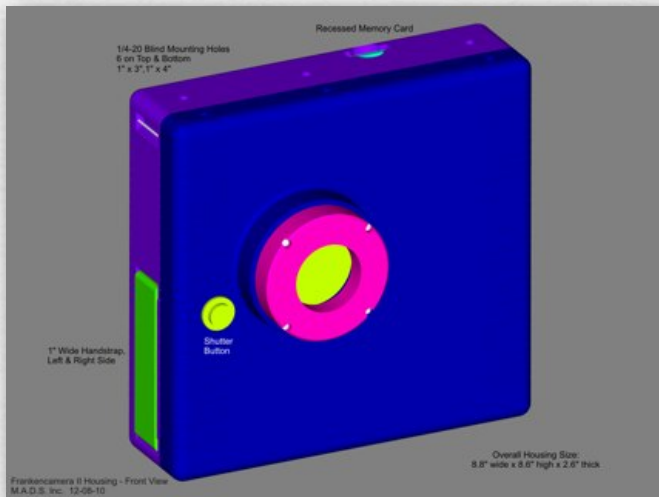


Long-term roadmap

- ◆ distribution to hobbyists, 3rd party developers
 - probably only N900s or equiv.
 - plugins and apps

Long-term roadmap

- ◆ distribution to hobbyists, 3rd party developers
 - probably only N900s or equiv.
 - plugins and apps
- ◆ distribution to researchers and students
 - courseware + Frankencameras/N900s
 - bootstrap open-source community



Cypress LUPA 4000
24x24mm, 4Mpix

Long-term roadmap

- ◆ distribution to hobbyists, 3rd party developers
 - probably only N900s or equiv.
 - plugins and apps
- ◆ distribution to researchers and students
 - courseware + Frankencameras/N900s
 - bootstrap open-source community
- ◆ wish list for makers of camera hardware
 - per-frame resolution switching at video rate
 - fast path into GPU texture memory
 - hardware feature detector

