Computational Sensors

Suren Jayasuriya

Postdoctoral Fellow,
The Robotics Institute,
Carnegie Mellon University
1) Vote on this poll about project checkpoint date on Piazza: https://piazza.com/class/j6dop76al46ao?cid=126

1) Reminder: Start HW5 this first week – if you haven’t started by Wednesday, you should hurry up.
Computational Photography

optics to focus light on an image plane

digital sensor to capture focused light (electrical process)

arbitrary computation between sensor and image
Computational Photography

- Optics to focus light on an image plane
- Digital sensor to capture focused light (electrical process)
- Arbitrary computation between sensor and image

Examples include:

- Coded Apertures

Light Fields

Slide courtesy of Ioannis Gkioulakis
Computational Photography

- Optics to focus light on an image plane
- Digital sensor to capture focused light (electrical process)
- Arbitrary computation between sensor and image

Examples include:

- Panorama Stitching
- HDR Imaging

Slide courtesy of Ioannis Gkioulekas
Computational Photography

- Optics to focus light on an image plane
- Digital sensor to capture focused light (electrical process)
- Arbitrary computation between sensor and image

This lecture will cover recent developments in computational sensors

Slide courtesy of Ioannis Gkioulekas
Review: Traditional CMOS Image Sensors

Pixel stack:
- Microlens
- Color Filter
- Photodiode
- Readout Circuitry

Slide courtesy of Ioannis Gkioulekas
3T (3 Transistors) Pixel:
- Each pixel has a reset, source follower (or amplifier) and row select transistor
- The relative timings between turning on/off the transistors = exposure and readout of the pixel
RST = ON

- The photodiode is charged to a high voltage (between 1-3.3 volts for modern technologies)
RST = OFF

- The photodiode is now integrated photocurrent onto its own (internal) capacitor

- Voltage is decreased across the capacitor as (negative) charge accumulates

- Eventually, the pixel will saturate or voltage = 0

Photocurrent flows from cathode to anode
ROW = ON

- Transistors Msf and Msel are turned on

- The voltage is read out to the column, where it is sent to a column amplifier, then an ADC to be digitized
Rolling Shutter

CMOS Image Sensor

Row Address Decoder

Start Address

Stop Address

Sample and Hold Column ADC

Column Scanner

Output

Slide courtesy of Jinwei Gu
Rolling Shutter

Pixel Array

Row Address Decoder

Start Address

Stop Address

Address Generator

Sample and Hold Column ADC

Column Scanner

Image Rows

Time

Slide courtesy of Jinwei Gu
Rolling Shutter

Pixel Array

Row Address Decoder

Start Address

Stop Address

Address Generator

Sample and Hold Column ADC

Column Scanner

Exposure Readout

Image Rows

Time

Slide courtesy of Jinwei Gu
Advantages and Disadvantages of Rolling Shutter

Advantages:
- Easy to read out the image sensor, space-efficient in column parallel readout
- No need for extra memory to store pixel voltage (unlike global shutter)

Disadvantages:
- Rolling shutter effect
- Can effect performance of computer vision algorithms such as structure-from-motion, SLAM, and stereo if not careful
Computational Sensors

But what is different about computational sensors vs. regular sensors?

Some options:

1. Change the readout/timing
2. Change the pixel design itself
3. Change the on-chip processing
Computational Sensors

But what is different about computational sensors vs. regular sensors?

Some options:

1. Change the readout/timing
2. Change the pixel design itself
3. Change the on-chip processing
Change the Readout and Timing

Camera Shutter: Space-Time Sampling

Rolling Shutter (CMOS)

Global Shutter (CCD)

Coded Rolling Shutter

Flutter Shutter

[Raskar et al., 06]

Gu et al, “Coded Rolling Shutter Photography: Flexible Space-Time Sampling” ICCP 2010
Gu et al, “Coded Rolling Shutter Photography: Flexible Space-Time Sampling” ICCP 2010
Change the Readout and Timing

Coded Exposure Photography: Assisting Motion Deblurring using Fluttered Shutter
Raskar, Agrawal, Tumblin (Siggraph 2006)

<table>
<thead>
<tr>
<th>Short Exposure</th>
<th>Traditional</th>
<th>Coded</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

- **Shutter**
- **Captured Photos**
- **Deblurred Results**

*Image is dark and noisy*
*Result has Banding Artifacts and some spatial frequencies are lost*
*Decoded image is as good as image of a static scene*
But what is different about computational sensors vs. regular sensors?

Some options:

1. Change the readout/timing
2. Change the pixel design itself
3. Change the on-chip processing
ASP: A New Type of Pixel

Angle Sensitive Pixel Structure

Grating pitch, \( d \)
Grating separation, \( z \)

Diffraction grating
Analyzer grating
Photodiode

Capturing Light Fields using ASPs

Capturing Light Fields using ASPs

Capturing Light Fields using ASPs

Angle Sensitive Pixel Structure

Grating pitch, $d$

Grating separation, $z$

Diffraction grating

Analyzer grating

n-well

Photodiode

Operating Principle: Talbot Effect

Plane wave on grating generates periodic diffraction pattern

~ 1 µm
Incident Angle shifts

Intensity, a.u.

Depth, microns

0 5

0 5 10 15

x, microns
Add an Analyzer Grating

0 degrees

10 degrees
Angle Response
Angle Response
Angle Response

Pixel Output (V)

Incident Angle, $\theta$ (deg)
Angle Response
Angle Response

\[ V_{out} = I_0 A(\theta)(1 + m \cos(\beta \theta + \alpha)) \]
Quadrature Inversion

\[ V_0 = I_0 A(\theta)(1 + m \cos(\beta \theta)) \]
\[ V_{\pi/2} = I_0 A(\theta)(1 - m \sin(\beta \theta)) \]
\[ V_{\pi} = I_0 A(\theta)(1 - m \cos(\beta \theta)) \]
\[ V_{3\pi/2} = I_0 A(\theta)(1 + m \sin(\beta \theta)) \]

\[ I_0 A(\theta) = \frac{V_0 + V_{\pi}}{2} = \frac{V_{\pi/2} + V_{3\pi/2}}{2} \]

\[ \theta = \frac{1}{\beta} \tan^{-1}\left( \frac{V_0 - V_{\pi}}{V_{3\pi/2} - V_{\pi/2}} \right) \]
\[ \rho^{(\alpha, \beta, \gamma)}(\theta) = \frac{1}{2} + \frac{m}{2} \cos(\beta \cos \gamma \theta_x + \beta \sin \gamma \theta_y + \alpha) \]
ASP Camera

We tile the entire image sensor with this repeated pattern of different ASP pixels.

The sensor is fabricated in an unmodified CMOS process.
Experimental Setup

- ASP Sensor
- Chip Package
- Main Lens
- Prototype Setup
ASP Light Field Capture

\[ i(x, y) = \iiint l(x, y, q, f) \, dq \, df \]

**Physical Layout**

<table>
<thead>
<tr>
<th>Physical Layout</th>
<th>Impulse Response (2D)</th>
</tr>
</thead>
</table>

- Each pixel modulates the light field with a different angular response function

\[ \rho^{(\alpha, \beta, \gamma)}(\theta) = \frac{1}{2} + \frac{m}{2} \cos(\beta \cos \gamma \theta_x + \beta \sin \gamma \theta_y + \alpha) \]
ASP Light Field Capture

- Model the image capture process: \( i = l \)

\[
\rho^{(\alpha, \beta, \gamma)}(\theta) = \frac{1}{2} + \frac{m}{2} \cos(\beta \cos \gamma \theta_x + \beta \sin \gamma \theta_y + \alpha)
\]
• Linear Reconstruction: $l_{downsampled} = \frac{1}{i}$

• We can invert this equation using linear methods by reducing the resolution of the 4D light field.

• The resulting reconstruction is low resolution due to the spatio-angular tradeoff.
Compressive Light Field Photography

Captured 2D Image

ASP Projection

\[ i = \Phi l \]

4D Light Field
Compressive Light Field Photography

Captured 2D Image

ASP Projection

\[ i = \Phi l = \Phi D \alpha \]

Overcomplete dictionary

Dictionary

Sparse Coefficients
Decomposing light fields into sparse representations

\[ \mathbf{l} = \mathbf{D} \alpha \quad \text{s.t.} \quad \alpha \text{ is sparse} \]

- **Original Light Field**
- **Dictionary**
- **Coefficient vector**

Overcomplete dictionary

Can lead to fewer non-zero coefficients
Dictionary Learning

\[ l = \alpha \text{ s.t. } \alpha \text{ is sparse} \]

Training light fields

Sample 1,000,000 random 4D patches from training light fields

Use K-SVD algorithm to solve this problem

[Marwah et al. 2013]
Compressive Light Field Reconstruction

Captured 2D Image

ASPI Projection

Overcomplete dictionary

Basis Pursuit Denoise:

\[
\begin{align*}
\text{minimize} & \quad \|\alpha\|_1 \\
\text{subject to} & \quad \|i - \Phi D\alpha\|_2 \leq \epsilon
\end{align*}
\]
Experimental results

Nonlinear Light Field Reconstruction

Light Field  Light Field View
Comparison of reconstruction methods
Digital refocusing after the picture has been taken

Focused on Swan

Focused on Knight
But what is different about computational sensors vs. regular sensors?

Some options:

(1) Change the readout/timing

(2) Change the pixel design itself

(3) Change the on-chip processing
Event-Based Cameras (also called Dynamic Vision Sensors)

Key techniques -

- Active logarithmic front end
- Self-clocked switch-cap differentiator

Dynamic Vision Sensor
## Event-Based Cameras (also called Dynamic Vision Sensors)

### Concept Figure for Event Based Camera
https://www.youtube.com/watch?v=kPCZESVfHoQ

### High Speed Output on a Quadcopter
https://www.youtube.com/watch?v=LauQ6LWTkxM

<table>
<thead>
<tr>
<th></th>
<th>CMOS (Grasshopper3)</th>
<th>DVS (DVS128)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>2048 x 2048</td>
<td>128 x 128</td>
</tr>
<tr>
<td>Power</td>
<td>4.5 W</td>
<td>23 mW</td>
</tr>
<tr>
<td>Frame Rate</td>
<td>90 Hz</td>
<td>3-10 kHz</td>
</tr>
<tr>
<td>Dynamic Range</td>
<td>52.87 dB</td>
<td>120 dB</td>
</tr>
<tr>
<td>Data Rate</td>
<td>Very High</td>
<td>Very Low</td>
</tr>
</tbody>
</table>
Applications of Event Based Cameras

Kim et al, “Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera” ECCV 2016 (Best Paper Award!)
Time-of-Flight (TOF) Imaging and Transient Imaging

Microsoft Kinect V2

Streak Cameras

cathode

Laser

Photo switch

CCD Detector

Demodulating Pixel

Single Photon Avalanche Diodes (SPAD)

Q/R

I_A

V_TH

V_CTRL

V_x

digital pulse

OUT

Gate 1 = High

Gate 2 = Low

Reflected Light

Differential output voltage

Gate 1 zone of influence (large)

Gate 2 zone of influence (small)

(unmodified CMOS process) psub
More on TOF and transient imaging in future lectures
Computational Sensors

But what is different about computational sensors vs. regular sensors?

Some options:

(1) Change the readout/timing

(2) Change the pixel design itself

(3) Change the on-chip processing
On-Chip Image Compression

Fig. 5. (a) Architecture of the overall imager including the sensor and the processor. (b) Corresponding microphotograph of the chip implemented in Alcatel 0.35-μm CMOS technology with the main building blocks highlighted. (c) Layout of the pixel.

Chen et al, A CMOS Image Sensor with On-Chip Image Compression Based on Predictive Boundary Adaptation and Memoryless QTD Algorithm, VLSI 2011
On-chip CNNs

Design of on-chip mixed-signal ADC for implementing a CNN on chip

Goal: Energy-efficient computer vision
Future of Computational Image Sensors

• Tighter integration of hardware and software, spanning programming languages to computer architecture to circuits to optics

• Image sensors custom for specific applications (machine vision, scientific imaging, etc)

• New pixel/sensing technologies: MEMS, Photonics, 3D stacking, etc.

• What do you predict?
Basic Reading:

Additional Readings:
• J. Gu et al, “Coded Rolling Shutter Photography: Flexible Space-Time Sampling”, ICCP 2010
• H. Kim et al, “Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera” ECCV 2016