TDCI QE Presentation

Paul S. Eberhart

2020-11-18

TDCI QE Presentation

Paul S. Eberhart

2020-11-18

Me.

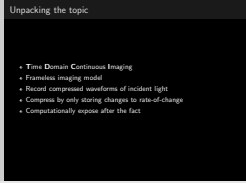
- Tool Nerd
- BS EE, BS CoE, BS CS, MS EE
- Too Long in Grad School
- Likes teaching and building prototypes ... to a fault

Me.
- Tool Nerd
- BS EE, BS CoE, BS CS, MS EE
- Too Long in Grad School
- Likes teaching and building prototypes ... to a fault

- Signed up for CS PhD to take classes (possibly for transfer), decided I might as well finish

Unpacking the topic

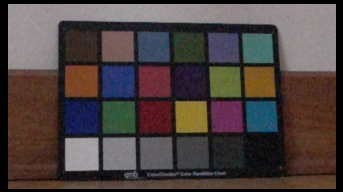
- **T**ime **D**omain **C**ontinuous **I**maging
- Frameless imaging model
- Record compressed waveforms of incident light
- Compress by only storing changes to rate-of-change
- Computationally expose after the fact



- Making a scene model by photon arrival rate
- Can model certainty/reliability as well as the scene

Rendering Example

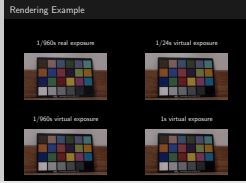
1/960s real exposure



1/24s virtual exposure

1/960s virtual exposure

1s virtual exposure



- From original TIK paper: 960FPS video on RX100, rendered through TIK to make exposures

Motivating Example

...So you want to take a picture of a running child

- Casio started the burst-shooting-then-pick thing - partial solution
- Some devices do simultaneous shooting - bad pictures, it's just video framegrabbing
- Some Sony bodies can burst and piece together, which gets closer
- Same game as HDR, spoiled by scene motion and lighting changes because the multiples aren't centered.

TDCI History

- In-House project of the KAOS group
- Publication record starting in 2014
- My first related co-author the same year
- NSF 1422811, "Computational Support for Time Domain Continuous Imaging" 2014-2018
- TIK toolchain and format exist

TDCI History
- In-House project of the KAOS group
- Publication record starting in 2014
- My first related co-author the same year
- NSF 1422811, "Computational Support for Time Domain Continuous Imaging" 2014-2018
- TIK toolchain and format exist

- My first in-field paper was "ISO-Less" (2014), on expressly TDCI papers by 2017 (TIK)
- MANY Subisquent related(-ish) papers; mostly doing tool-building
- TIK = Temporal Imaging from Kentucky (File format AND toolchain)

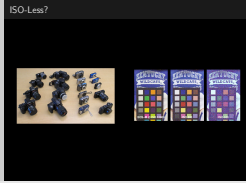
Coauthored Related Publications

- Iso-less?, EI2014
- Use of flawed and ideal image pairs to drive filter creation by genetic programming, EI2016
- Temporal super-resolution for time domain continuous imaging, EI2017
- Tik: a time domain continuous imaging testbed using conventional still images and video, EI2017
- Multispectral, high dynamic range, time domain continuous imaging, EI2018
- Lessons from design, construction, and use of various multicameras, EI2018
- Shuttering methods and the artifacts they produce, EI2019
- **Non-uniform integration of TDCI captures, EI2020**
- Senscape: Modeling and presentation of uncertainty in fused sensor data live image streams, EI2020
- Characterization of camera shake, EI2020
- An Ultra-Low-Cost Large-Format Wireless IoT Camera, EI2021

Coauthored Related Publications
- Iso-less?, EI2014
- Use of flawed and ideal image pairs to drive filter creation by genetic programming, EI2016
- Temporal super-resolution for time domain continuous imaging, EI2017
- Tik: a time domain continuous imaging testbed using conventional still images and video, EI2017
- Multispectral, high dynamic range, time domain continuous imaging, EI2018
- Lessons from design, construction, and use of various multicameras, EI2018
- Shuttering methods and the artifacts they produce, EI2019
- Non-uniform integration of TDCI captures, EI2020
- Senscape: Modeling and presentation of uncertainty in fused sensor data live image streams, EI2020
- Characterization of camera shake, EI2020
- An Ultra-Low-Cost Large-Format Wireless IoT Camera, EI2021

- ...Oh, it's up to 11
- Green is directly TDCI-related, bold is first author
- Cyan is precursor work
  - ISO-Less extended ex.
  - shuttering study then generating exposures with shutter behavior the next year
  - Shake for filtering correlated changes due to camera motion
  - Current weird camera does non-uniform readout experiments
- Most papers are Dr. Dietz + Me + Our flock of undergrads

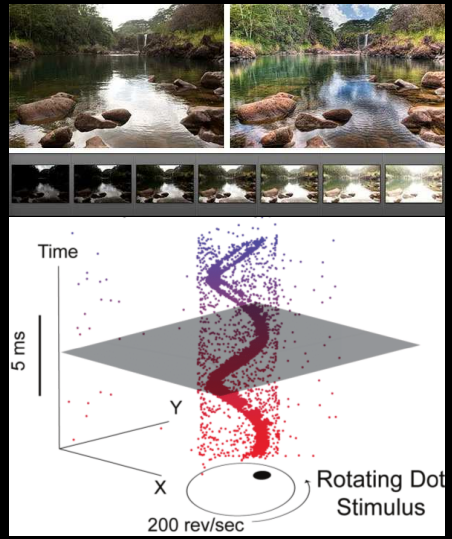
ISO-Less?

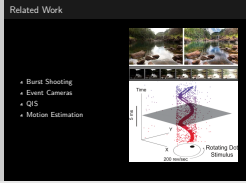


- Do cameras actually change their analog sensitivity when you set the ISO (Film-speed) or just apply digital gain? - YES
- Necessary that they be sort-of ISOless for TDCI to work on conventional sensor.
- Determination: Most cameras aren't, but APEX manipulation to maximize information content is possible (eg. don't clip).

Related Work

- Burst Shooting
- Event Cameras
- QIS
- Motion Estimation





- Burst discussed earlier in example.
- Mentioning for useful comparisons to TDCI
- Event cameras - Neuromorphic. Only edges (=polarity change. no constant data). Correlated change swamps data rate.
- Quanta Image Sensor - "jot" charge wells. req. high density. Count instead of Rate = no scene model.
- Both still frame-based
- Motion Estimation, larger granularity, deep buffers, threshold popping.

New Stuff (PhD Deliverables)

1. Non-Uniform
2. Pro-Grade/RAW/In-Camera
3. Tools for Mortals

New Stuff (PhD Deliverables)
Non-Uniform
Pro-Grade/RAW/In-Camera
Tools for Mortals

- I'd been kibitzing in tooling for TDCI and couldn't find a human-factors/software tools type project like I'd been thinking, so I sliced off a piece of this that hit my interests as a PhD.

(1) Non-Uniform

- Vary the interval, shape, and gain of integration
- Simulate arbitrary sensor and shutter behavior
- Do it again and again on the same data
- PoC published as EI2020 in "Non-Uniform Integration of TDCI Captures"

(1) Non-Uniform
- Vary the interval, shape, and gain of integration
- Simulate arbitrary sensor and shutter behavior
- Do it again and again on the same data
- PoC published as EI2020 in "Non-Uniform Integration of TDCI Captures"

-

Non-Uniform PoC

- Nasty Octave script to prove algorithms
- Ingest from video frames (only linear interpolation)
- Dog slow
- Works.

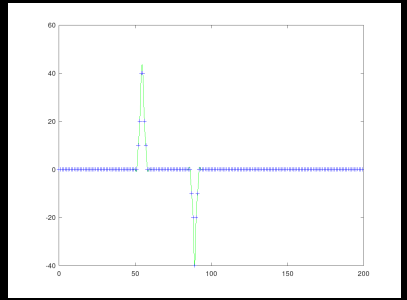
Non-Uniform PoC
- Nasty Octave script to prove algorithms
- Ingest from video frames (only linear interpolation)
- Dog slow
- Works.

- Octave is Matlab, but not Mathworks-brand-Matlab
- Dog Slow = Minutes per exposure

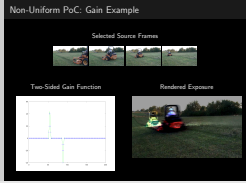
Non-Uniform PoC: Gain Example

Selected Source Frames

Two-Sided Gain Function



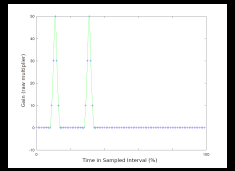
Rendered Exposure



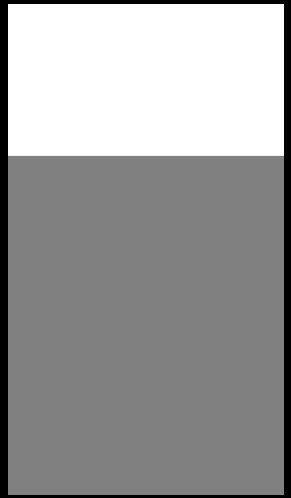
- Input: Video of Dr. Deitz driving a riding mower past
- Two-sided gain function both adds and subtracts incident light
- ringing because of lack of interpolation
- Suppressed everything in the frame that didn't move
- Positive and Negative Images
- This example is slightly brightened for display

Non-Uniform PoC: Multi-Region Example

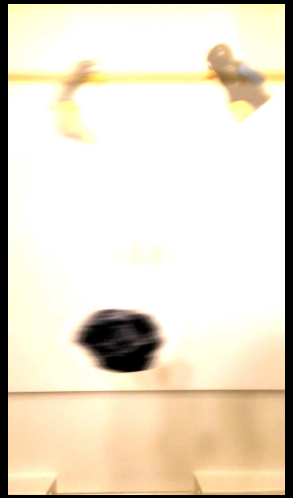
First Gain Function



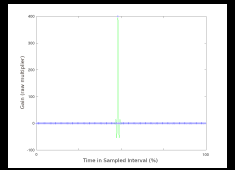
Area Mask



Resulting Exposure



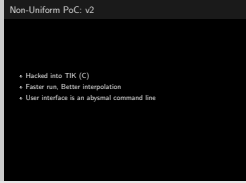
Second Gain Function



- Pendulum containing a penguin and rock, swinging
- Mask selects which parts of image are integrated with which function
- One two-peak function on top part of image
- Single peak on bottom part of image
- Narrower peak = sharper
- Penguins aren't left to right with exposures! Notice the rock is on the return swing.

Non-Uniform PoC: v2

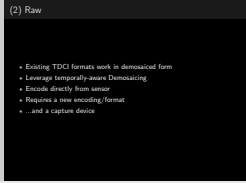
- Hacked into TIK (C)
- Faster run, Better interpolation
- User interface is an abysmal command line



- prototype used Octave features for function splines; having trouble getting dependable solution for that in C
- that last point applies to both versions.

(2) Raw

- Existing TDCI formats work in demosaiced form
- Leverage temporally-aware Demosaicing
- Encode directly from sensor
- Requires a new encoding/format
- ...and a capture device



- Demosaic = interpolate from color filter array pattern to full color pixels
- There are some Temporal + Spatial demosaic schemes,eg. Wu/Zhang from Proc. SPIE, 2003

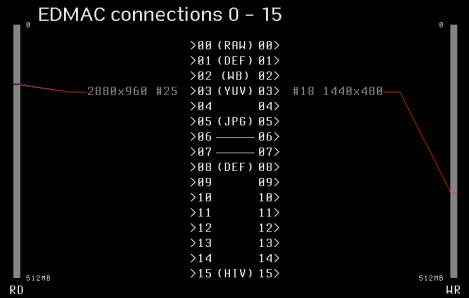
In-Camera

- Other than the (Long 2017) CHDK implementation, all implementations rely on conventional camera capture for reprocessing
- Requires direct access to a camera sensor readout path
- Failed effort: subverting a Sony a6000
- Failed effort: Tethering an industrial camera
- MagicLantern supported Canon bodies have EDMAC access, looks suitable

In-Camera
- Other than the (Long 2017) CHDK implementation, all implementations rely on conventional camera capture for reprocessing
- Requires direct access to a camera sensor readout path
- Failed effort: subverting a Sony a6000
- Failed effort: Tethering an industrial camera
- MagicLantern supported Canon bodies have EDMAC access, looks suitable

- CHDK = Canon Hack Development Kit. Firmware hack for small Canons.
- Long implementation used the live view buffer(limited to 360x240)
- In-camera requirement because bandwidth; can only get the data off the sensor in TDCI form
- Sony cameras run Linux, but sensor readout is via a black-box kernel module, and modern Sonys aren't hack-able the same way we got in, so no forward path
- Basler ACE modules w/ Pylon software look promising, but limitations on number and shape of Through the lens triggering and area readout make it no better than video ingestion
- Industrial camera also doesn't make a good user artifact
- MagicLantern = Firmware hack
- EOS M specs:18MP (5184 x 3456) APS-C sensor, ARM946E-S
- A6000 specs: 4 core ARM-CXD90014, 200MB RAM

Canon EDMAC



- Engine DMA Controller
- Plumbs sensor, encoders, main memory, storage
- Reverse-Engineered from experimental characterization (a1ex) and Patent (US7817297)
- Already in use for RAW Video Capture module (MLV_Lite.mo)

Canon EDMAC
- Engine DMA Controller
- Plumbs sensor, encoders, main memory, storage
- Reverse-Engineered from experimental characterization (a1ex) and Patent (US7817297)
- Already in use for RAW Video Capture module (MLV_Lite.mo)

Proposed RAW Capture

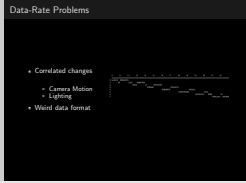
- Code path similar to mlv_lite to redirect raw sensor data to memory
- Target Canon EOS M : Mirrorless design suitable for continuous readout
- Perform TDCI in-body before writing to SD
- Limiting Factor CPU Cycles?
  - RAW Video is IO bound, and `mlv_lite` already does frame-diffing to avoid spurious writes

Proposed RAW Capture
- Code path similar to mlv_lite to redirect raw sensor data to memory
- Target Canon EOS M : Mirrorless design suitable for continuous readout
- Perform TDCI in-body before writing to SD
- Limiting Factor CPU Cycles?
  - RAW Video is IO bound, and mlv_lite already does frame-diffing to avoid spurious writes

- I've hacked MLV_Lite to add bogus arithmetic complexity without dropping framerate so there is headroom

Data-Rate Problems

- Correlated changes
  - Camera Motion
  - Lighting
- Weird data format



- Camera Motion: Do some simple correlation; no fancy motion tracking, just local alignment.
- Lighting: Flicker detection, at least for 50/60Hz
- Not doing the motion stuff in first pass; that's cycles-allowing.
- 14 bit RAW pixels come off packed in groups of 8, as 112bit blobs, gonna have to do some vectorizing.

Proposed RAW Format:TRW

- DNG like: Mutant TIFF container
- mlv like: roll with camera details

Proposed RAW Format:TRW
• DNG like: Mutant TIFF container
• mlv like: roll with camera details

- Tiffs hold anything and give it structured description
- mlv rolls with camera-specific quirks to maximize throughput

(3) Build Better Tools

- TIK is a morass of command line accretion
- Solve with exposure specification DSL
- GUI Front-end to design and preview for usability

(3) Build Better Tools
- TIK is a morass of command line accretion
- Solve with exposure specification DSL
- GUI Front-end to design and preview for usability

- TIK is "research software" - you have to read the source code to figure it out its command line
- Also kind of nasty to hack on due to accretion.

Tool Features

- Read in TRW streams
- Set exposure parameters
- (Re)Expose, including non-uniform (earlier)
- Demosaic
- Generate finished exposures in conventional formats

Tool Features
- Read in TRW streams
- Set exposure parameters
- (Re)Expose, including non-uniform (earlier)
- Demosaic
- Generate finished exposures in conventional formats

- Demosaicing will leverage the scene model and stats for better color
- Similar to some RAW-video-Demosaicing tricks that use temporal referencing in the literature/practice

Domain-Specific Language

- Set up a readily parsed file format that describes an exposure
- Source, Time(s), Exposure Parameters, functions in a standard format
- Thinking INI-Like syntax
- Continue using PGMs for masks - easily embedded in the DSL

Domain-Specific Language
- Set up a readily parsed file format that describes an exposure
- Source, Time(s), Exposure Parameters, functions in a standard format
- Thinking INI-Like syntax
- Continue using PGMs for masks - easily embedded in the DSL

- Allow for scripting, back a front-end, make it reproducible
- DSL is declarative *only*

GUI Frontend

- Give fast preview feedback for parameter tweaking
- Multi-tab drag bar for time specification
- Operate by generating exposure specs in said DSL, then calling the backend tool
- Don't expose all the controls

GUI Frontend
- Give fast preview feedback for parameter tweaking
- Multi-tab drag bar for time specification
- Operate by generating exposure specs in said DSL, then calling the backend tool
- Don't expose all the controls

- Have some Qt experience ... not a lot.
- It has primitives for the image display tools
- GUI is to make it so easy cases aren't paying for extra complexity, not to make everything work.