Post-Capture Synthesis of Images Using Manipulable Integration Functions

A presentation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in Computer Science in the Stanley and
Karen Pigman College of Engineering
at the University of Kentucky
By

Paul Selegue Eberhart

Personal Introduction

- Around academia/UK forever
- BSEE, BSCPE, BSCS, MSEE … PhD CS?

≈

Computer Engineer

- Hacker who found out I like teaching
- Interested in tools and useful artifacts

This Project

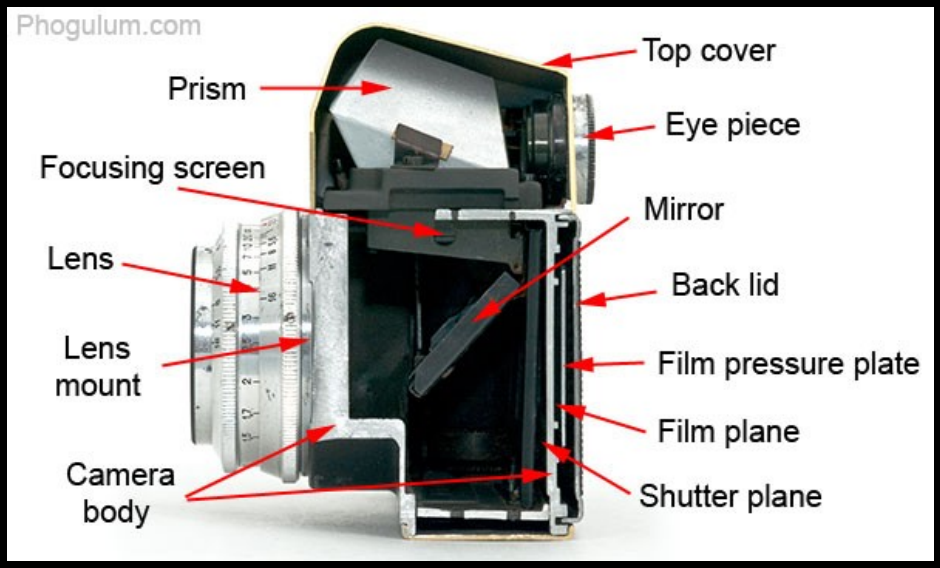
- Analyze traditional practices around cameras
- Analyze how modern cameras behave
- Develop a model that better leverages the nature of digital cameras
- Analyze ongoing/upcoming camera developments in light of that model

- Good things
- Bad things
- Complications
- Insights
- Papers published along the way

Things are being presented in _logical_ rather than _chronological_ order, for narrative reasons.

I want to but am currently not:
...Because, as everyone in CS knows, anything typeset in computer modern is automatically more credible
[Flash up chicken paper sight gag]

First...



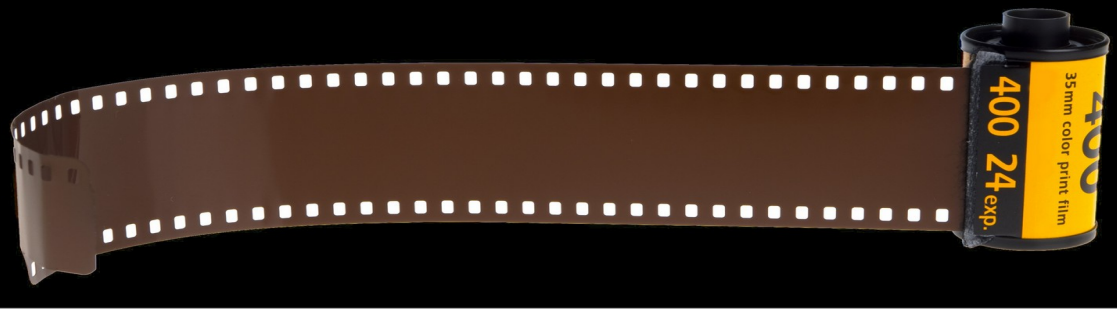
Since we're talking about assumptions from the 140-odd year film area, we need to know what they are and where they come from.

(Boring for old people, photographers, and especially old photographers (Sorry hank))

Go over this bisected SLR in broad terms
-Lens
-Mechanical Gadgets direct light
    - SLR= Flippy Mirror, Might be Pellicle
    - Viewfinder w/ focus assistance
-Film at the back

Film

- Photochemical emulsion
- Chemical + optical processing



Classically silver halides that get easier to crash to metal when struck by photons
Some kind of substrate – glass, polymer film, paper etc.

Delayed gratification
Nasty chemicals – develop, wash
A sophisticated user can push process/dodge/burn and manipulate in the develop and print process.

Shake it like a polaroid picture

Early digital cameras also didn’t have instant feedback

3 Exposure Controls

The human operator configures:

1. Film Speed
2. Shutter Speed
3. Aperture

(Focus/Focal Length intentionally excluded)

Film Speed

- Essentially “gain” or “sensitivity”
- Set when you load the film
- Usually coarsely graduated
  - ISO: 80,100,125,160,200,...5000,6400
  - Or some crustier system; DIN, GOST, vendor specs

Doubling arithmetic ISO = doubling sensitivity.
A little weird because humans perceive brightness log-ish.
Also a log scale where every 3° is a doubling, ISO 100/21°, ISO 200/24° etc.

Shutter Speed

- Time the film is exposed to the light projected by the lens
- Set for the whole scene
- Set no later than time of exposure
- Usually coarsely graduated in 2:1 scale
  - 1,1/2,1/8,1/15(lol),1/30,1/60,1/125,1/250…
- Motion Blur

Make focal planes with hands
- Traveling slit

Not infinitely fast or close to the film plane

Aperture

- Opening through which light passes
  - Ratio of focal length to pupil diameter
- Set for the whole scene
- Set no later than time of exposure
- Usually coarsely graduated in “Stops”
  - Powers of $1/\sqrt{2}$: f/1.4, f/1.8, f/2, f/2.8, f/3.5, f/4, f/5.6
- Side effects: DoF, Sharpness

Grab a lens and wiggle the aperture while talking.

Not just how much, but also angle

APEX

- Log_2, algebraic rearrangement
- $E_v=A_v+T_v=B_v+S_v$
  - Exposure = Aperture + Time = Sensitivity + Brightness
- Let human operators quickly compute exposure
- Reciprocity Failure

Even more coarse, things like

Sunny 16 Rule: On a sunny day set aperture to f/16 and shutter speed to the reciprocal of the film speed for a subject in direct sunlight.

Not enough incident light, aperture starts acting like a pinhole, etc.

Control

- Before capture:
  - Human operator makes determinations
  - Early automation: human operator queues priorities
- What if we could defer the decision?

Comparison

- Yi-Ren Ng, Lytro
- 2006 PhD “Light field photography with a hand-held plenoptic camera”
- Tries to defer focus like this work tires to defer exposure
  - By imposing a lenticular array in the optical path
- Method proposed in 1908

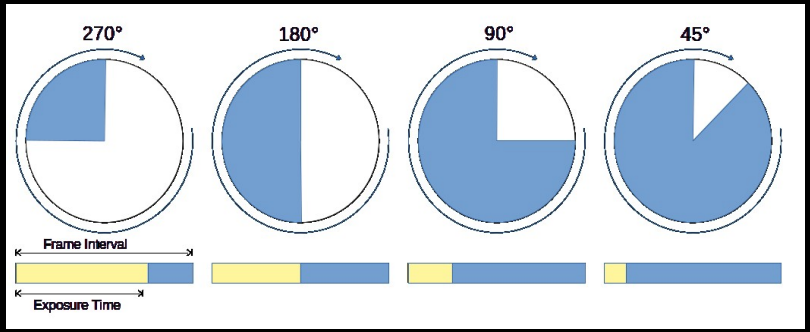
Gabriel Lippmann

- Winning the nobel prize for color-from-interference at the time
- Called it “Integral Photography”
- Couldn’t build it

- 100 years of precedent
  - Early implementations by the 1920s
  - launched a thousand landfill-ready lenticular postcards
- 20+ years of computational precursors

What About Video?

- Machine drags film through the camera
  - In a synchronized way
- Still just frames
- Shutter angle



Exposure time is bounded by framerate

Digital Cameras

- Optical system is similar
- Sensor is an array of photoelectric sensels
  - With a readout mechanism
  - Since the late 1960s
- Under computer control
  - Independent from the sensor
  - Since the mid 1970s

First digital sensors were some folks at bell labs
- Not widespread commercial for another 15-20 years

Independent meaning: mid 1970s, Canon AE-1 gets a microprocessor in 1976

~~Film Speed~~ Sensitivity

- ISO 12232:20076 and amendments
- Configured on-the-fly
- Nothing says it has to be uniform
  - Except the configuration device

Chicanery: sensitivity varies by band, IR is easy, etc.

ADCs aren’t fully independent (lines)

Canon Dual-ISO alternating lines

Still has to be correlated (gain constant during exposure)

ISOLess?

- How “real” is that ISO setting?
  - A change in analog sensitivity?
  - A change in digital post-processing?
    - This would allow manipulation after exposure
  - A mixture of the two?
- Find out by bracketing ISO with everything else fixed

How much of that ISO setting is Analog and how much is digital post-processing?

Studied and published:
H. G. Dietz and P. S. Eberhart, “Iso-less?” EI2014

Spread of Cameras, several vendors, several ages, mix of CCD and CMOS sensors, different market segments, etc.

JPEG Crops from Canon A4000 at ISO 1600, 16x ISO 100, and 16x ISO 100 Filtered

How much of that ISO setting is Analog and how much is digital post-processing?

Studied and published:
H. G. Dietz and P. S. Eberhart, “Iso-less?” EI2014

ISOLess?

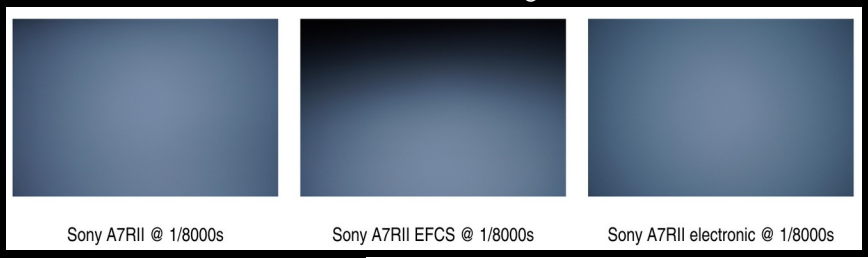
- It Depends!
  - Some cameras have many analog gain settings
  - Some cameras have one analog gain setting
  - Most cameras do at least some digital gain
- Reprogrammed cameras where possible
  - CHDK
  - Successive addition is like multiplying by 2^n...

Sensor gain will tolerate least a degree of digital post processing without much (or any) loss of quality/information

Shutter (and Readout) Speed

- Start (or stay) open
- Electronic First Curtain
- No/Rolling Shutter
- Timing Artifacts

Shuttering Methods and the Artifacts they Produce



H. Dietz and P. Eberhart, “Shuttering methods and the artifacts they produce,” EI2019

First picture shows that the electronic shutter is in-plane and the mechanical is not => shading from off-angle light

Second picture shows banding from AC lighting (center) with EFCS and geometric distortion from rolling readout (right) => readout time is NOT CORRELATED

The light integrated into a frame is not as correlated as one might imagine

Aka. Structure in the time dimension

Aperture

- Not much change from conventional
- The blades are servo controlled so it easily could be moved continuously, even during exposure
  - Your firmware *probably* won't let you

During exposure for Apodization

This … isn't actually a digital thing, just a computer control thing.

Minolta Maxxum/alpha/Dynax 7 STF by twitching the aperture in a 35mm film camera ca. 2000

Multiple sub-exposures, not truly continuous

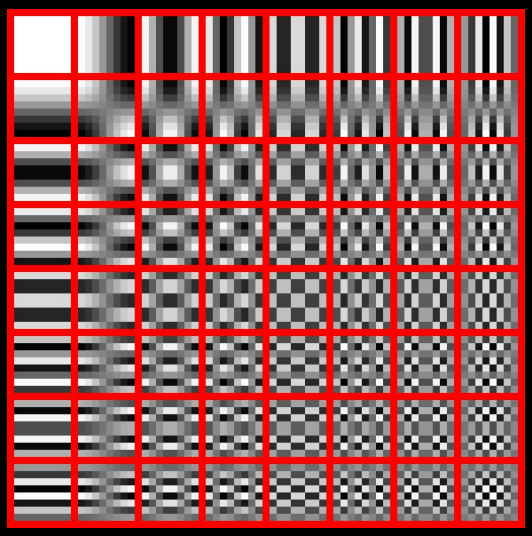
Control

- Programmed shooting modes
- Automation
  - Bracketing
  - Faster than any human operator
- Even computers might be feeble
  - APEX96

Priorities. .. Magic, and More Magic

Encoding & Compression

- Agree on a representation
  - Simple: BMP, PNM
- DCT
- Entropy Encoding
- Not all the time: RAW
  - But usually even then



Discrete Cosine Transform: Decompose (spatial) frequency components into linear combination of references

JPEG does 8x8

Entropy Encoding: RLE, Huffman, Arithmetic Etc.

More sophisticated options available… rarely used

Interchange, patents, etc.

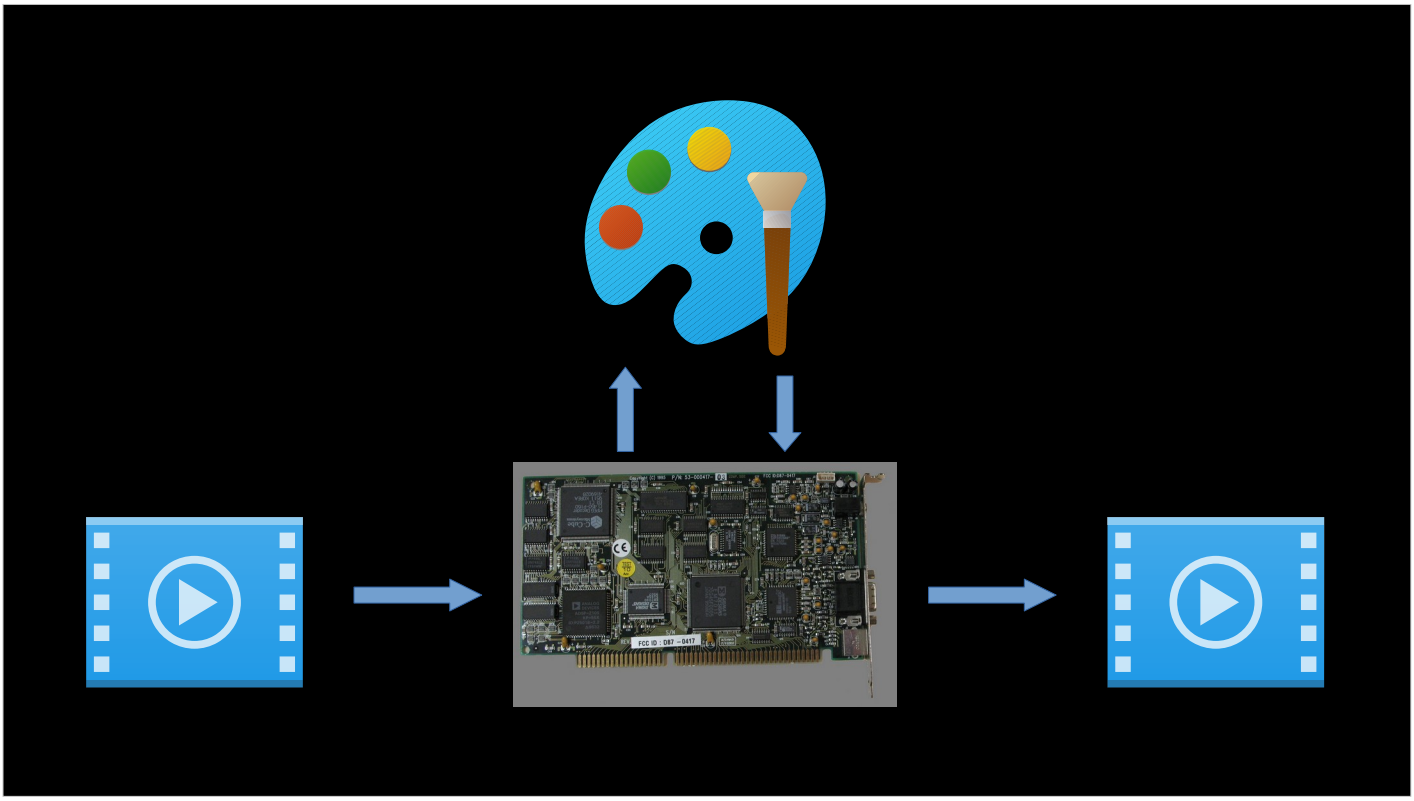
What About Digital Video?

- “Box of Frames” is too big
- DCT + [Motion Estimation] = Compressed Video Format
- DCT more-or-less like images
- Motion Estimation/Optical Flow models describe changes to scene appearance over time

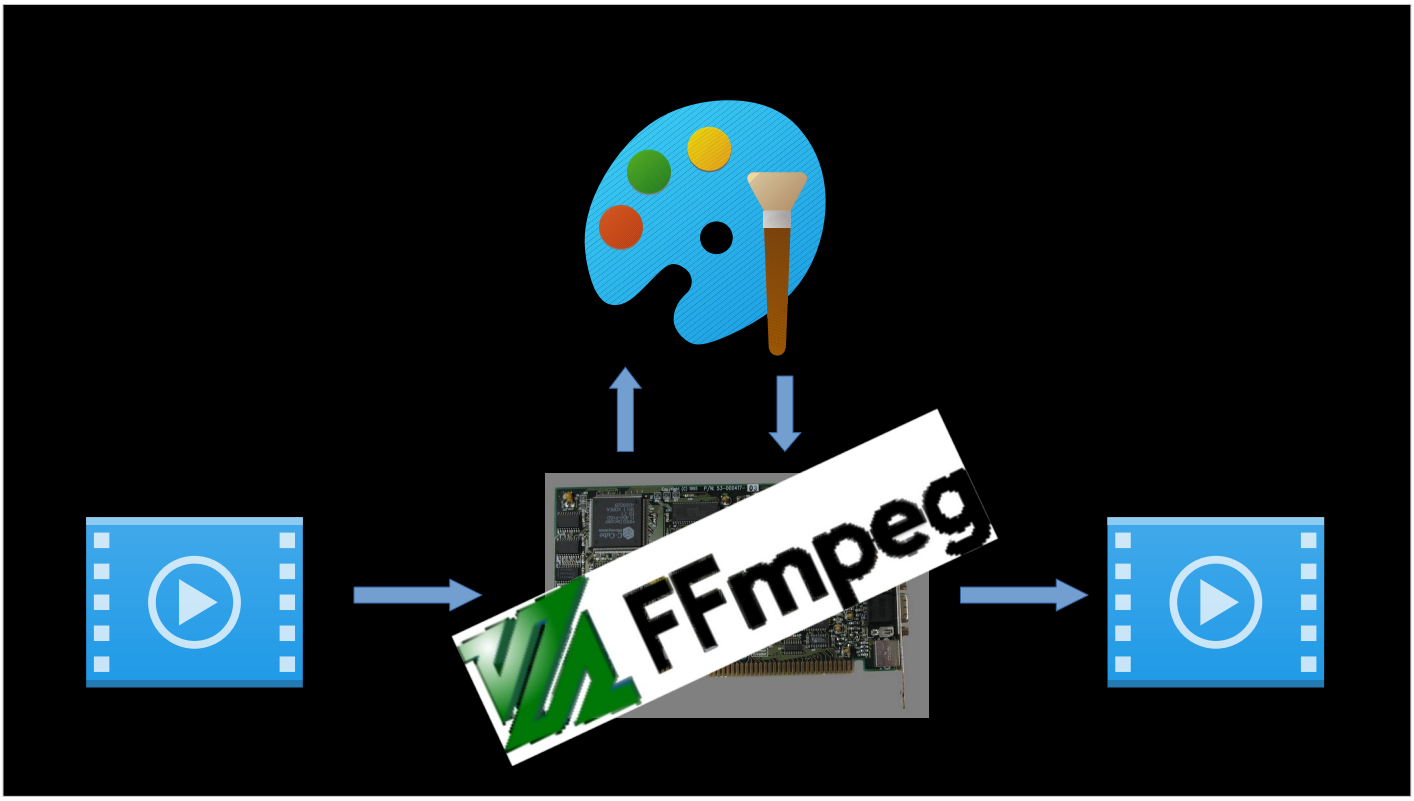
H261 → MPEG-1 in ‘92

There is even MJPEG that only does a box of DCT’d frames. Compression is 1:20 instead of like 1:50.

T




Computers are fast enough to operate directly on compressed video now
So we're going to do homomorphic editing, right?
Right?


FFmpeg

- I thought I was setting up an engineering project
  - Modify a digital camera to operate in a way more consistent with it’s nature
  - Expose and recombine existing behavior
  - UI design
- Got a lot more philosophical
  - A digital native mode of operation for cameras

On the way to Violate Assumptions

Having Violated Assumptions

Interfaces

- Digital cameras are film camera simulacra
  - Everything is frame oriented
  - Whole frame sensitivity and shutter
  - Parameterize for pleasing images at time of capture
  - Even the physical layout

Designed to simulate film, then replaced film, still simulating.

Lytro Camera (2012)

Lytro Illum (2014)

Everyone who builds a camera that *isn't* a Film Simulacra runs into this – the Lytro Camera didn't "Look like a camera" and the later Lytro Illium is a stylized skeuomorph of an SLR.

MILC

- Now we’re talking digital native!
- Use the exposed main sensor for:
  - Focus, Exposure, Video, etc.
  - EVF
- Eating the market

A few exceptions to the simulacra rule
- EVF means it’s doing decent exposure recomputation _on the fly_ to display the preview
- Funny tales of early vendors, placing mirrorless between point-n-shoot/phone and srs bsns SLRs.
  Nikon 1 vs Z

Event Cameras, QIS

- Event Cameras: Capture changes only
  - Asynchronously
- QIS: Count photons
- Data Rate Limited
  - Especially saturation due to correlated changes

Not (yet?) mass market

Event cameras are “first derivative” if you will

QIS = Quanta Image Sensor
- “Next step” by Eric R. Fossum, known for enhancements to make practical CMOS sensors

TDCI

- “Time Domain Continuous Imaging”
- First publications ca. 2014
- Capture scene model
  - By logging changes to the rate of change of incident light
  - Second Derivative camera
- Computationally render frames

Capture Maximum Scene Data

- Saturation is the enemy!
  - Within noise of [Min/Max] adds no information
- Expose to minimize out-of-range
- Redundant data is wasted bandwidth
- Correlated sampling makes those harder

Correlated is that frame assumption
- Traps you to exposing specific areas properly, the rest is along for the ride.
- Bandwidth used on redundant readings is wasted. (Scene Constancy again)

Computational Integration

- Separate capture and integration
- Σ incident light over an interval ≈ exposure
- Integration after capture
  - Over and over, until you get it right
- Shutter angles > 360

Simplest motivating example:
Trying to take a cute picture of a toddler.
One shot!
Time wrong = less cute picture falling on face
Interval wrong = blur
Sensitivity wrong = dark or washed

- No Real TDCI Cameras
  - But a few hacked approximations
- TIK: a time domain continuous imaging testbed using conventional still images and video
  - Software testbed
  - Synthesize from various sources

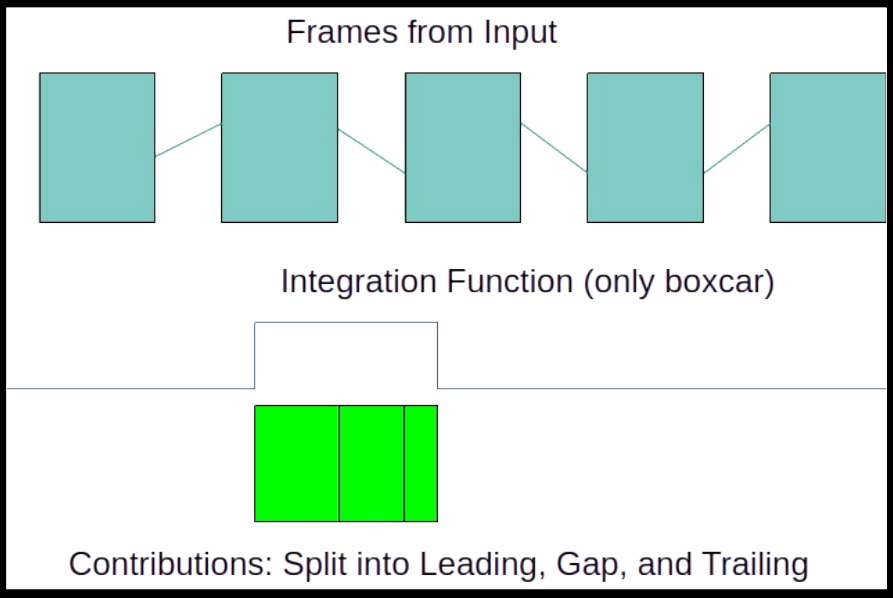
Katie's EI2016 CHDK TIK
Scrapes the viewfinder feed (lol)
720x240, motion detect trigger
Camera can't play back the stream

Making a novel sensor is a years and M$ proposition, look at QIS

TIK: a time domain continuous imaging testbed using conventional still images and video,
Henry Dietz, Paul Eberhart, John Fike, Katie Long, Clark Demaree, Jong Wu, EI2017

TIK is "a stack of first pancakes"

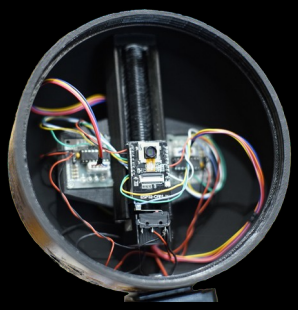
The TIK Model



The Ministry of Funny Cameras





Alternate title “Weird Cameras Я Us”

Survey of our own work published at EI2018 as “Lessons from design, construction, and use of various multicameras”

Co-witnessing, synchronization (or not), global time, etc.

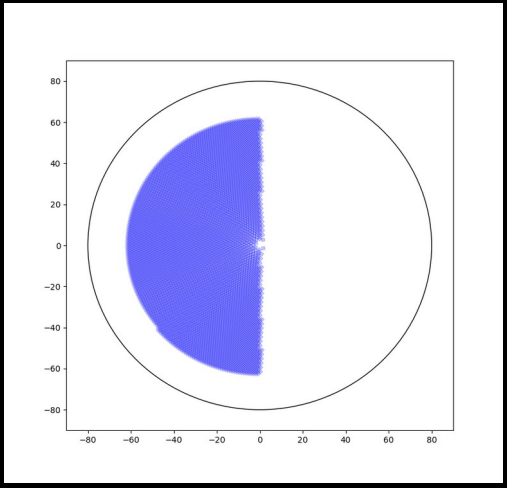
Also EI2022 ESP32-CAM as a programmable camera research platform

I’ve built parts or set up tooling of all of the pictured systems:

FourSEE, KREight, KREighteen, KameraflY, Kodama, Lafodis

An Ultra-Low-Cost Large-Format Wireless IoT Camera

- LAFODIS160
- Exercise in Non-Uniform Sampling
- Swing a 1.5mm^2 sensor around in a 160mmD large format projection



LAFODIS160
- H. Dietz and P. Eberhart, “An ultra-low-cost large-format wireless iot camera,” EI2021

LAFODIS=**LA**rge **FO**rmat **DI**gital **S**canning
- Reprogrammable camera!
- Exercise in non-uniform sampling!
    - Mostly a thought experiment
- Excuse to work on a Polar Robot!
    - Not as much because we were working on it high-pandemic, so I mostly only got to play with code.

- Naive around ~500MP 4x5, fancy full 160mm ~2.6GP.. at about 1MP/s so minutes per frame
- Useful experience in Python/MatPlotLib

Sane Next Steps in TDCI

- Design a better TDCI Capture Device
  - Hack a “prosumer” camera
  - Extract a better scene model from a conventional camera
  - Exclude: Ground Up Semiconductor Design
- Make better exposure tools
- Address coordinated scene change swamping

Finished course work, commit to a PhD project
Kibitzing in this digital camera stuff

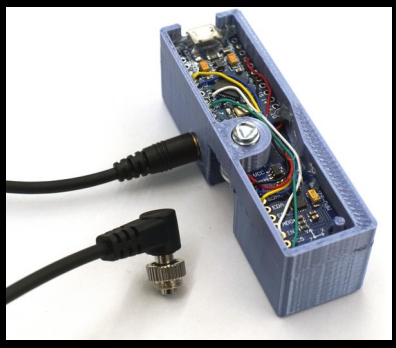
Also, an Intrusive Thought

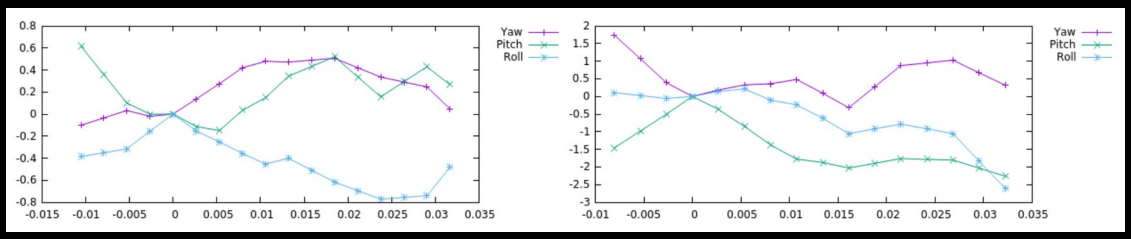
- In a TDCI system:
- Time is a complete, continuous dimension to be manipulated after the fact
- Gain is a complete, continuous dimension to be manipulated after the fact

Wibbly-Wobbly

You ever have one of those ideas that you *know* is going to be trouble the instant it passes through your head, but you can’t stop thinking about it?
My thesis title is one of those.

Characterization of Camera Shake





- Common but not consistent or visible: IBIS,OIS
- EI2020: Characterization of camera shake

- Hank, William Davis (undergraduate researcher) , and Me!
- I mostly built the test electronics package, Will focused on the test procedure, Hank built the analysis code.
- Gadget is called ShAKY
-9 axis = 3x (Magnetometer, Accelerometer, Gyro)
- Lessons:
    - Expected: viewfinder>screen,
      no mechanical shutter = less motion
    - Not: 2 hands = more rotation
- Not much more TDCI mileage
    -Hank's burst super-resolution tricks

Better TDCI Capture

- Nicer Optics
- More Spatial Resolution
- More Temporal Resolution
- RAW?

- Sony α6000
- Mirrorless, 24.3 MP APS-C sensor, EVF, etc.

Play MemoriesCameraApps → OpenMemories

Sony α6000

- Running Linux
  - Display layer similar to Android
- Programmable via PlayMemories
  - Jailed
- Hacked; Ma1co provides OpenMemories and OpenMemories Tweak

The Computer in the Camera

- `...Linux 3.0.27_nl-rt106 SMP PREEMPT RT...`
- 4 core Arm v7l
- 200MB of main memory
- Black box kernel modules
- Coprocessors

(24MP*12BPP)/8=36MB per raw buffer

- Takes about 3 to really do TDCI diffing

- Blackbox Kernel Modules
  - `osal_utm, osal_uipc`, etc.
  - `liro`
  - `dmac, lld, ldec`
- PlayMemories Discontinued

Touch random values at random memory locations
Defy static analysis because they work in concert
Probably talking to fixed function hardware?

“OSAL” = “Operating System Abstraction Layer”
“LIRO” = depend of others, spawns hundreds of kernel threads
dmac/lld/ldec = DMA Controller/Load/Decode?

Sony did the PS3 Cell Linux removal after someone jailbroke the VM.
Angry nerds then destroyed the whole security model and sued for a settlement for users.
Sony distributed essentially windows malware on audio CDs to make them inconvenient to rip

- Canon EOS M
- Mirrorless, 18MP APS-C sensor

Canon EOS M

- Running DryOS
  - Used across Canon's line
- No Native Programming Interface
- Hacked: Magic Lantern
- MLV Raw Video

µITRON-interfaces, looks kind of DOSy

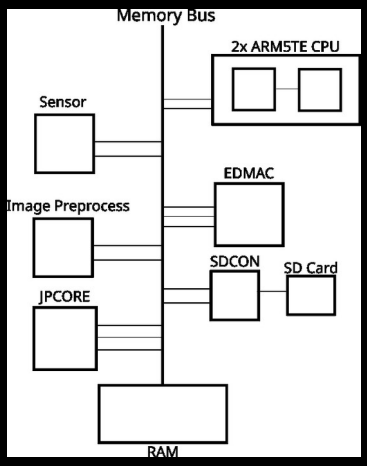
ML:
- Initiated by Trammell Hudson in 2009
- Lead by A1ex since late 2010
- EDMAC reverse engineering back-matched to patent US7817297B2
- Modular

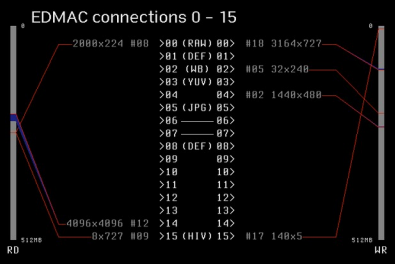
MLV:
- MLVLite implementation by David Milligan
- Also MLVFS
- MLVApp development by Ilia Sibiryakov

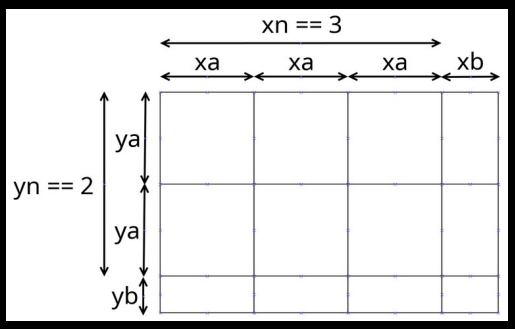
The Computer in the Camera

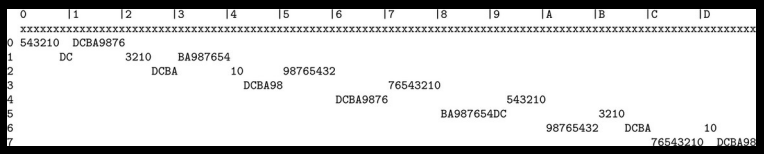
- 2 core ARM5te
- Around 256MB RAM (Split)
- EDMAC

Magic Lantern as a Platform for Digital Photography Research



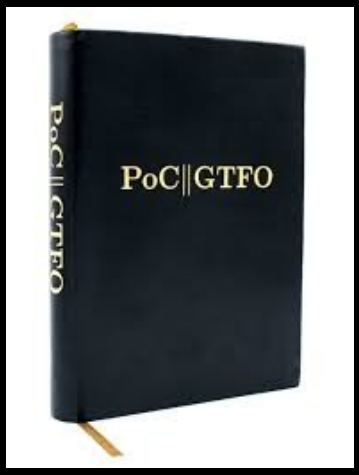






Bus structure
EDMAC Connection Visualizer
EDMAC stride arguments as diagram
14-bit packed RAW format

- No recent cameras supported
- EDMAC is not *that* flexible
  - 2-in-1-out modes not deciphered
  - Subtract mechanism might not be general?
- MLV output of the EOS M tops out at 1728x692 @ 30FPS


PoC||GTFO
PoC||GTFO

Non-Uniform Integration of TDCI Captures

- A PoC prototype to think the newly-freed time and gain dimensions through
  - Is this even possible?
  - What kind of exposure functions make sense?
  - Hack it out in Octave
  - Reprocess video frames

EI2020 Paper

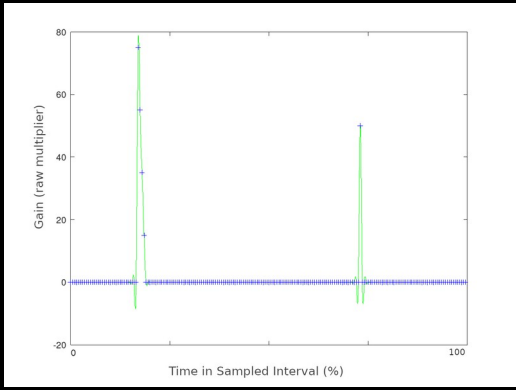
Gain Functions

- Weighted
- Vary over time
- Can be negative
  - “Remove the contributions from this interval”
- Represented as splines

Masks

- Bitmap at same resolution as source
- Value-per-function
- PGMs are already suitable
- Manipulate with conventional bitmap image tools

An Exposure from the PoC



It works...

Findings:

- Splines are not suitable
  - Common cases contain sharp edges
- Slow to render
  - Hard to play if a single low-res exposure takes minutes
- Ringing artifacts
  - Need to use a better scene model

NUTIK

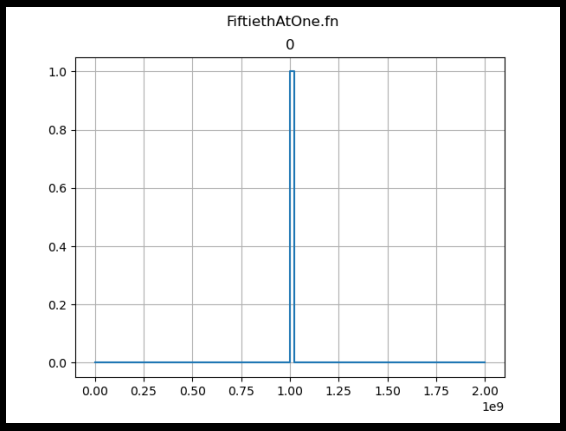
- Build a better software prototype
  - Start from TIK20220707
  - Use OpenCV structures where possible
  - Add Masks and Functions
  - Spec functions in a more suitable way
  - Achieve reasonable performance

Exposure Function Spec Files

- Linear interpolate between a list of points
- `Mnn{[t0:g0],[t1:g1],...,[tn:gn]}`
  - `Mnn` = Tag for the area
  - `tn` = Time in ns
  - `gn` = Gain (as a float)

The Simple Case

- Take a 1/50 second unity gain exposure at 1 second
  - `M00{[0,0],`
    `[1000000000,0],`
    `[1000000001,1],`
    `[1020000000,1],`
    `[1020000001,0],`
    `[2000000000,0]}`



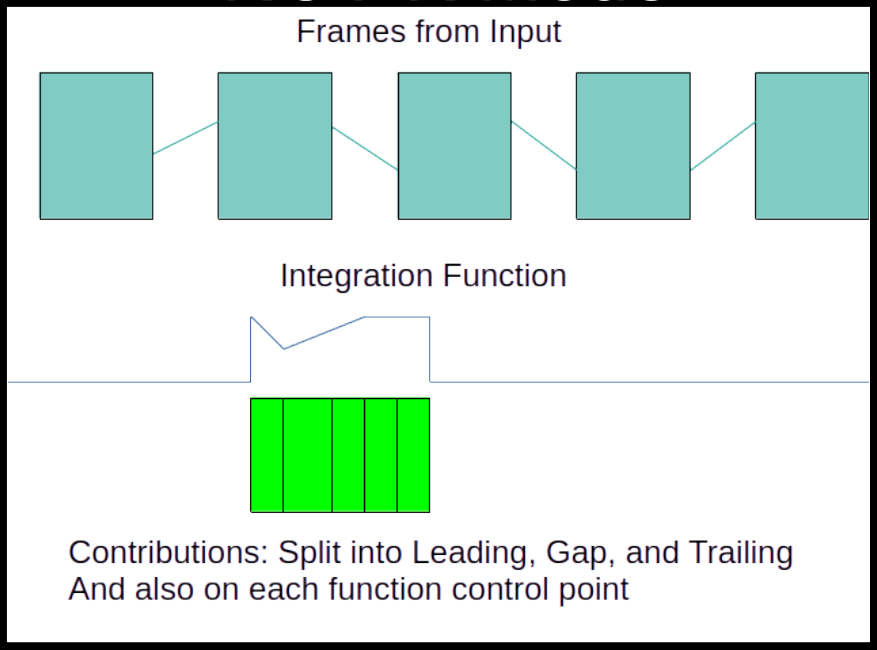
NUTIK Implementation Details

- Add `MaskGain.[cpp,h]`
  - Read and query masks and functions
- Modify `render.cpp`
  - Rendering now uses masks and functions
- Modify `tik.cpp`
  - Add to the command line soup

Segments time
Original Tik: Sum(Part from leading frame, part from gap, part from trailing frame)

NUTIK: Also split on control points!

NUTIK Model


Frames from Input
Integration Function
Contributions: Split into Leading, Gap, and Trailing
And also on each function control point

- Possibly a contribution diagram?

Helpers

- FnPlotter.py
  - Graphs Exposure Function Spec Files
  - Python+MatPlotLib, <100lines
- More needed
  - Thinking about this is “different”

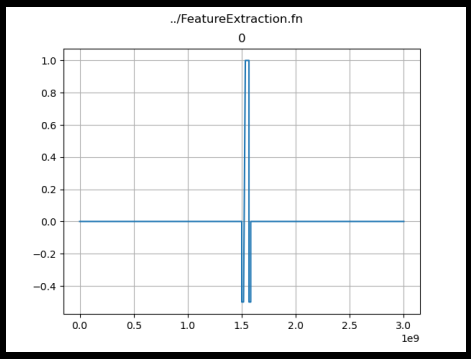
Helper tool for visualizing Exposure Function Spec Files.
Python3 + MatPlotLib, Under 100 lines

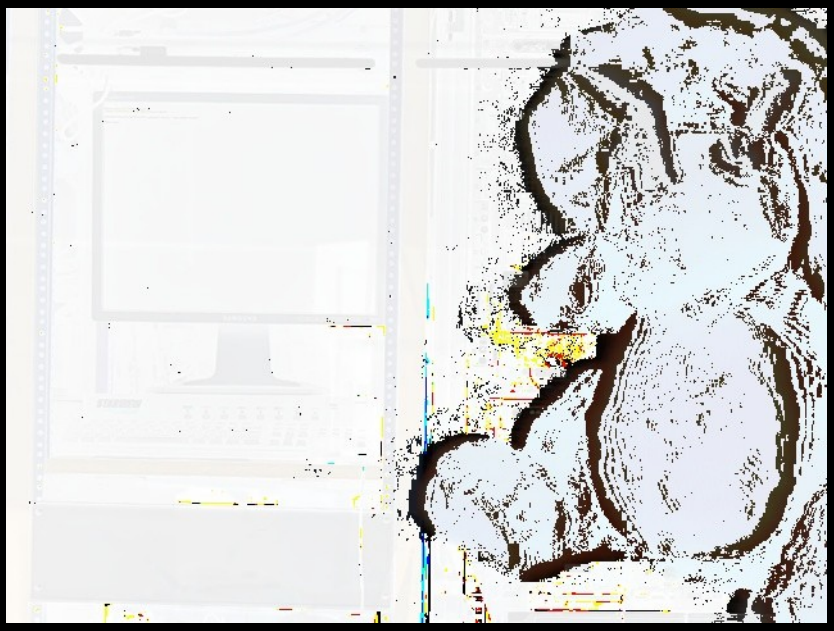
```
./tik -m../AllZero.pgm -k../FiftiethAtThree.fn
Sample.tik
```

Double Exposures

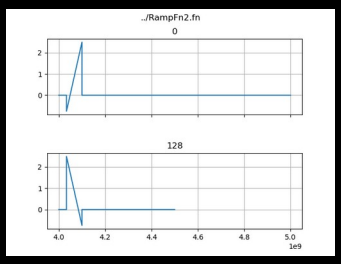
- Don’t work because NUTIK is b0rked

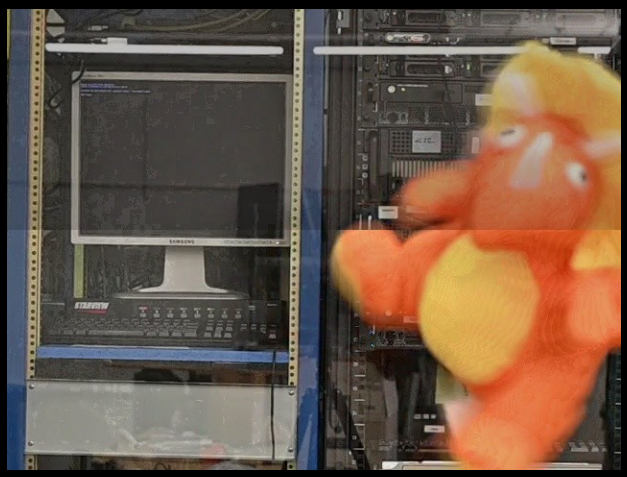
Isolate Differences





Several Ramp Functions(?)





[More Interesting Cases Here]

- Why not Zoidberg?

Problems with NUTIK

- Still a “fake”
- Normalization issues
  - Output and Internal
- Awful interface
  - Or just no intuition?
  - Build more chrome

That Lytro Comparison...

- Too weird and janky for consumer market
- Too much compromise on resolution for Pro market
- Out of business in 2018
- Former employees hired en mass by Google
  - ...to make fake depth features
- Ren Ng fully returned to Academia

“Portrait Mode”/”Background Blur” stuff is apparently derived.

Dr. Ng UC Berkeley since 2015, full time after Lytro died.

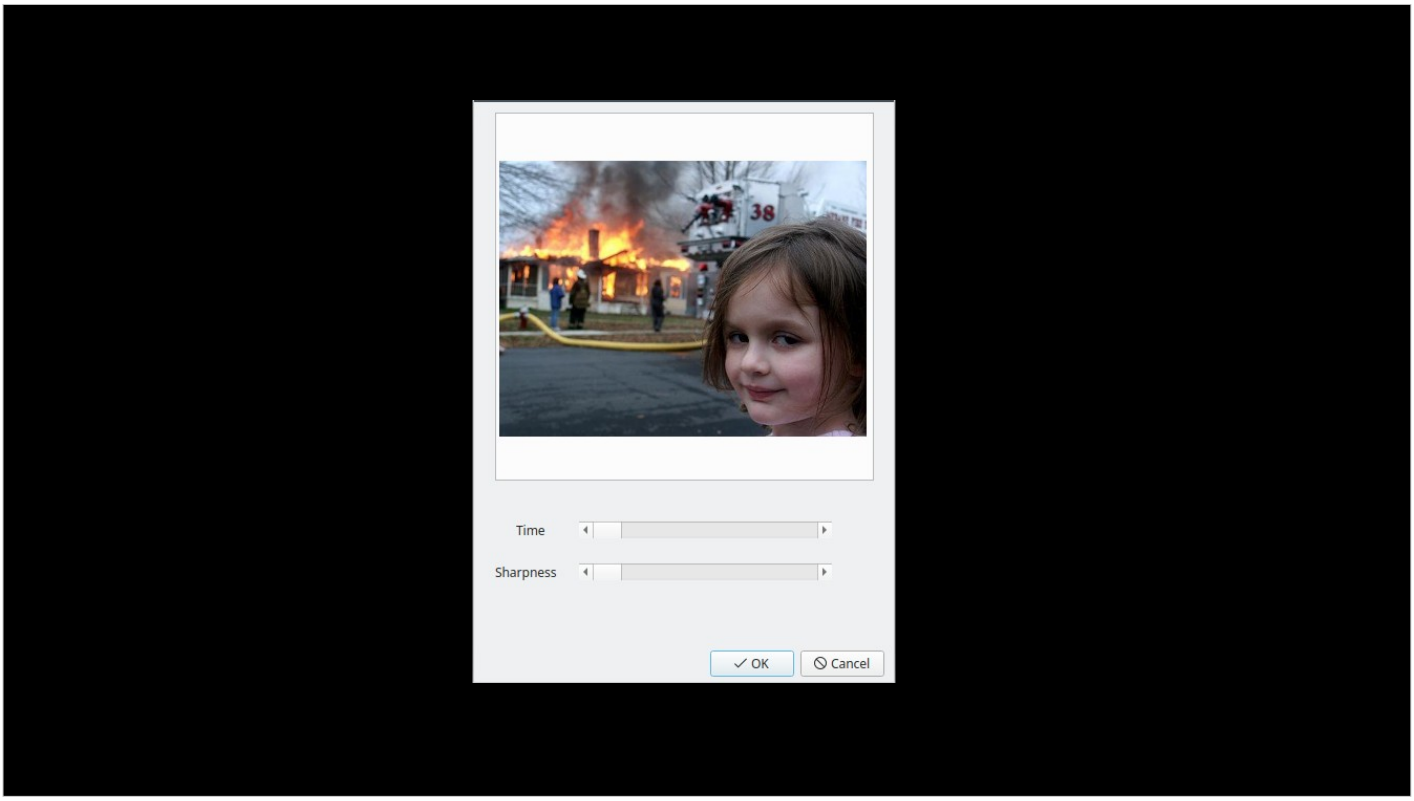
(Non-Uniform) TDCI In the Hands of Users?

I’m going to see something that looks derived from this tech in a commercial product in an ad.

Mechanism hidden, limited controls: “Time” and “Sharpness” maybe some automated subject selection

No one currently involved is likely to come up in the narrative.

Probably a tool that does the generate-model-and-re-expose frame-rate conversion thing


Time
Sharpness
OK
Cancel

Upcoming Developments?

- Bigger Buffers
- Faster Readout
  - More, more independent ADCs
- Credible QIS Sensors
  - Readout Bandwidth Limited

Sony A9 claims “1.6 seconds of 120fps photography at 14-bit Raw quality” = 7.5GB of buffer!

Thanks!

Thanks everyone coming + Listening!

Also: Rafi Finkel and Nathan Jacobs who were on my committee, gave good advice

BACKUP SLIDES

A Different Approach

- UVC Camera + SBC + 3D Printer + TIK = Prototype

IF I'd known what would and wouldn't pan out I might have gone this way.
...but that wouldn't have been research.

Read from a nice UVC camera, hack a UI onto TIK running on the embedded SBC with a little screen, print it a case, etc.

Also: Dr. Jacob's advice to "Minimal demo of some algorithms and move on."