As Time Goes By

Hank Dietz (and MANY students!)

ECE Seminar, 2:00PM Sept. 29, 2017

University of Kentucky
Electrical & Computer Engineering

University of Kentucky

Abstract

Using film, there wasn't much choice but to capture and display visual data as either a static image or a time sequence of such images – frames. However, some things change quickly with the passage of time while others do not. Why not take advantage of this property by using frameless computational rendering and image capture? This talk will discuss some of the basic concepts, advantages, and implementation methods. Several examples will be given of things TDCI (Time Domain Continuous Imaging) can do that conventional imaging cannot.

This day and age we're living in
gives cause for apprehension
With speed and new invention
and things like fourth dimension.
Yet we get a trifle weary
with Mr. Einstein's theory. ...
The fundamental things apply
as time goes by.

University of Kentucky

As Time Goes By?

- Time is the fourth dimension in the song
- Einstein's relativistic world is unnerving; we don't see photons and shot noise (which is why I'm not a believer in QIS)
- The fundamental thing is what we see, and ***it doesn't change so much*** ... as time goes by
- **Model what we see changing**

University of Kentucky

Rendering Output

- Mostly done a frame at a time
- Bishop, Fuchs, McMillan, and Zager, *Frameless Rendering: Double Buffering Considered Harmful*, SigGraph '94
- Why not use continuous, randomly-ordered, update of display rather than double-buffer with synchronized frame buffer swap?

Display Paint Order Update

- Frame sequence at 25FPS

University of Kentucky

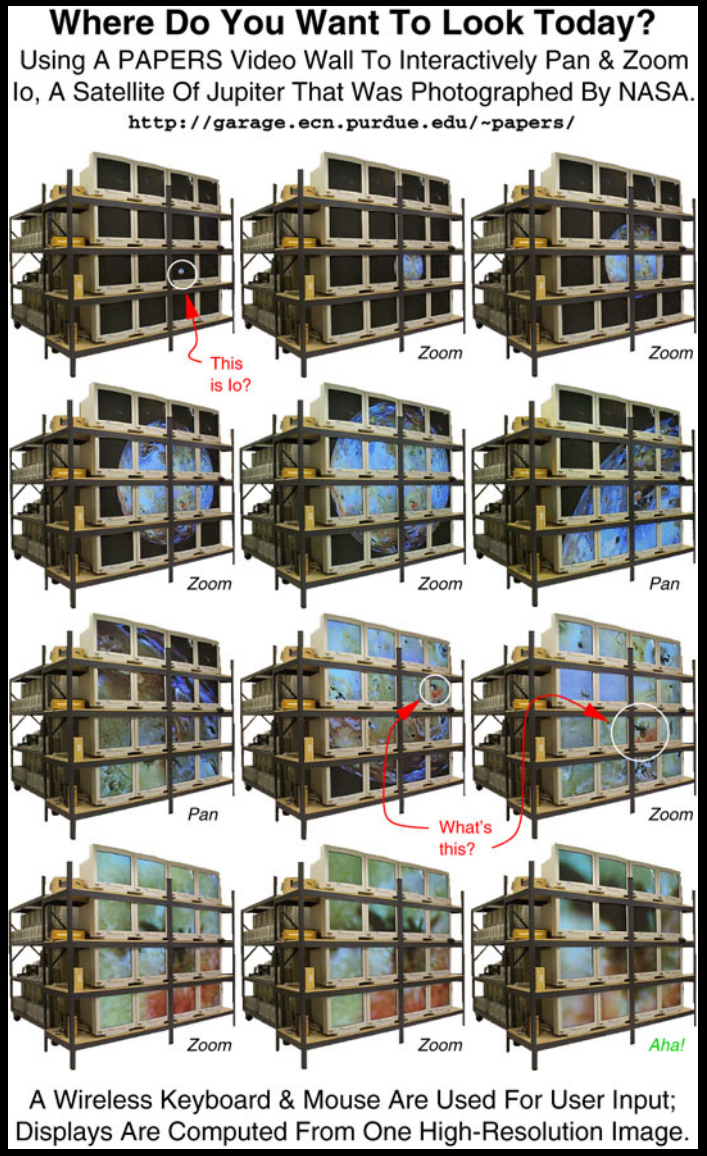
Random Update

- Frame sequence at 25FPS

University of Kentucky

Video Walls

- Linux cluster video walls, 1996 Dietz @ Purdue
- 16 1600x1200 monitors
- Unsync'd, slow, video cards
- Stationary random dither for update and shading avoids flicker
- In VWLib, uses MMX



University of Kentucky

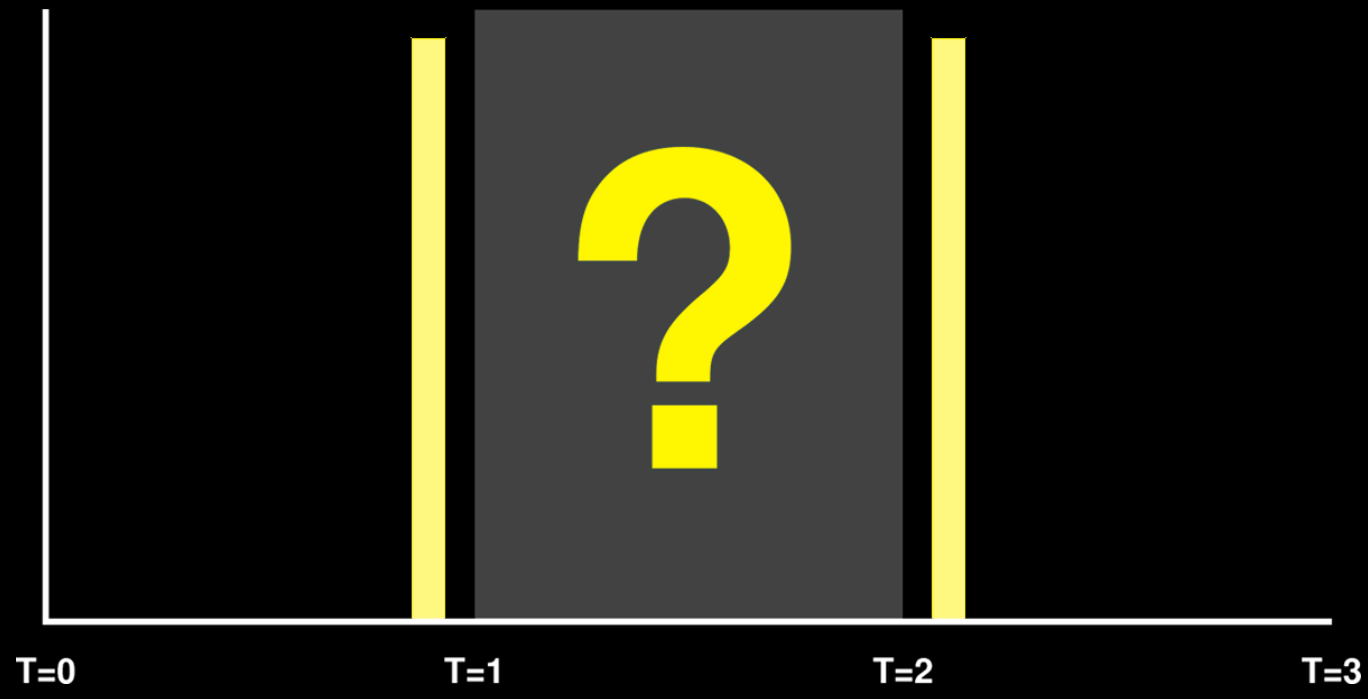
Update Only What Changed

- UDP versions of VWLib used "*inverse raindrop*" intellegent update of the display for memory-mapped dynamic images
- Dayal, Wooley, Watson, and Lubke, *Adaptive frameless rendering*, SigGraph `05
- Oddly, LCD and OLED displays – which are inherently randomly accesible – still are fed a **complete frame at a time**!
- Should really only send *unexpected changes*

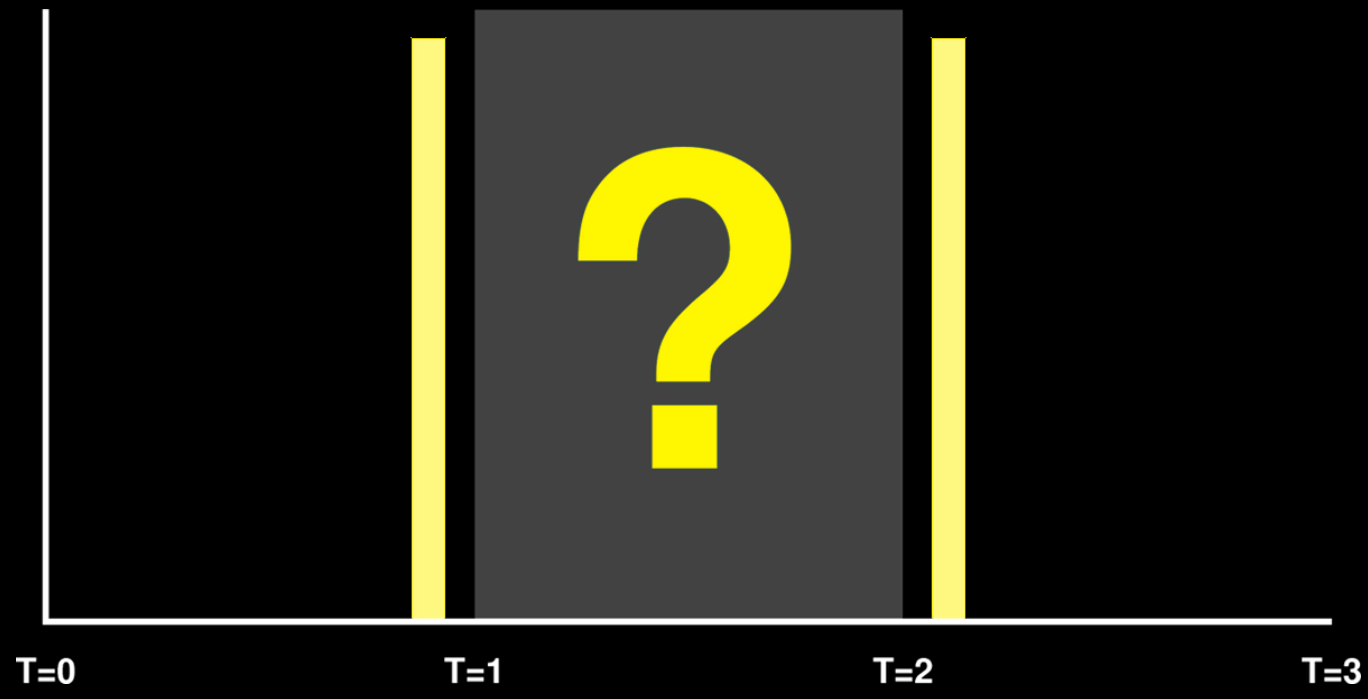
Capturing Scenes

- Cameras create scene appearance models
- Photons are *just* the sampling mechanism
- Scene appearance changes slowly over time

E.g., Fossum's Quanta Image Sensor (QIS) is trying to capture images fast enough to record each photon... but that's not necessary because **one photon doesn't tell you anything**

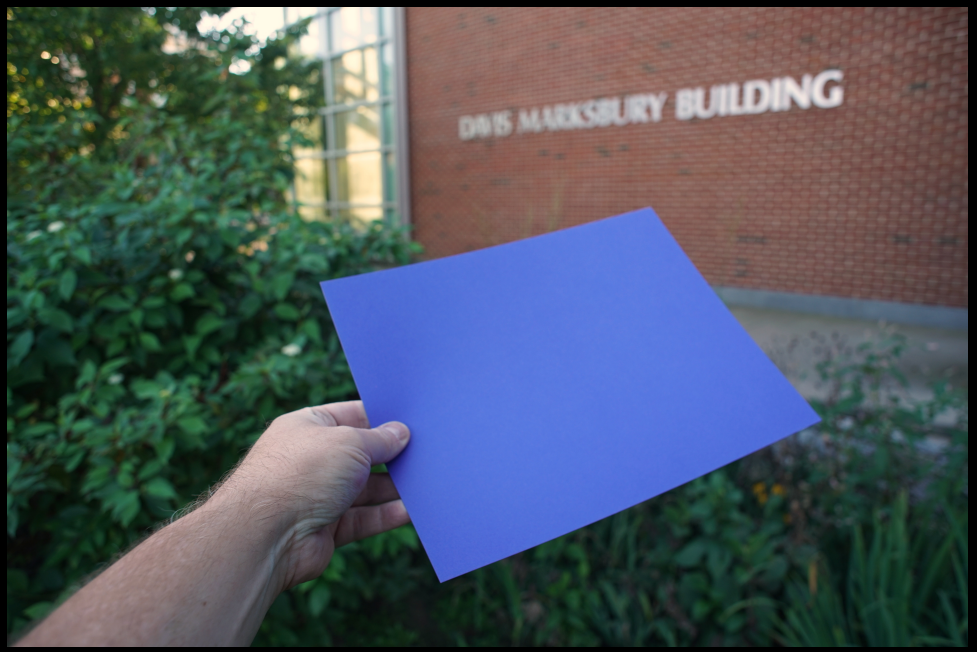


What is the value of a pixel in T=[1..2]
if in T=[0..3] photons hit at T=0.99, T=2.01?



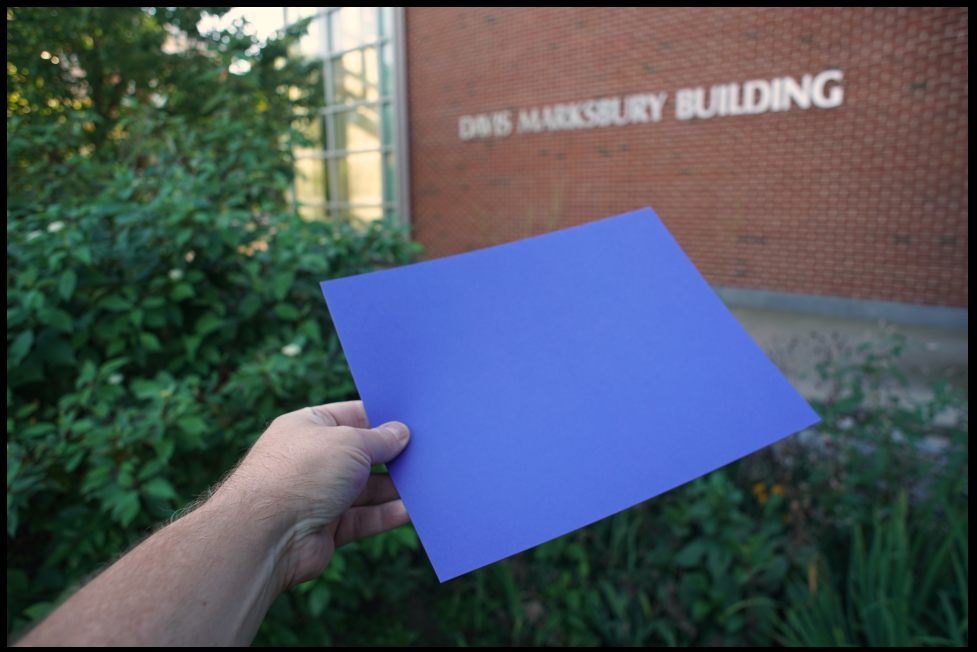
What is the value of a pixel in T=[1..2]
if in T=[0..3] photons hit at T=0.99, T=2.01?

- Value is 0 because no photons hit in [1..2]
- Value is 2/3 photon – photon arrival **rate**



What color is a blue sheet of paper when **NO** photons are sampling it?

University of Kentucky



What color is a blue sheet of paper when **NO** photons are sampling it?

- Black
- Probably still blue… I just can't prove it

University of Kentucky

Time Domain Continuous Imaging (TDCI)

- Allows frameless capture
- TDCI representation: a continuous waveform per pixel, compressed (mostly) in time domain
  - E.g., 8MP sensor produces 8M waveforms
  - Each waveform is sequence of pixel value change records
- Can use TDCI representation for processing
- Can use TDCI representation for display

Benefits of TDCI

- Suggests better ways to make cameras
- Rendering virtual exposures enables:
  - High dynamic range (HDR), improved SNR
  - Rendering a virtual exposure for any time interval (start time, shutter speed)
  - Rendering a conventional video at any FPS and shutter angle (temporal weighting)
- Could also directly use TDCI representation for general image processing and display

The FourSee TDCI Multicamera

- Four Canon PowerShot N photograph image projected by central lens
- Software under CHDK controls timing: 240FPS, 240FPS (1/480s late), 24FPS, stills

University of Kentucky

TDCI Inside *One* Canon PowerShot via CHDK

A Canon Hack Development Kit implementation of Time Domain Continuous Imaging

Katie Long, Henry Dietz, & Clark Demaree

University of Kentucky

Aggregate.Org UNBRIDLED COMPUTING

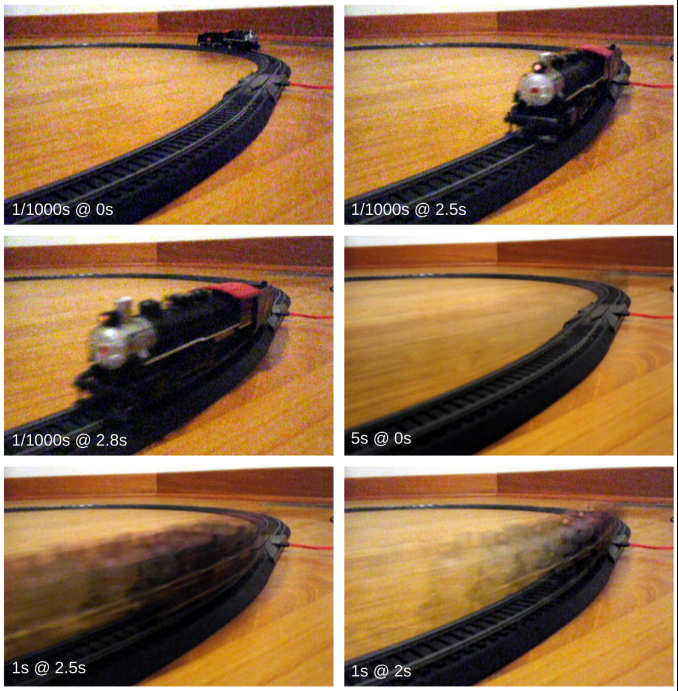
***Time Domain Continuous Imaging (TDCI)* is frameless capture: a continuous waveform is recorded representing each pixel's value over time.**

We used the Canon Hack Development Kit (CHDK) to directly implement TDCI capture *inside* Canon PowerShots. Each capture is extracted from the live view stream and encoded in the new `.tik` (Temporal Image Kontainer) file format.

All the images to the right are virtual exposures computed from a single `.tik` capture:

Temporal Image Kontainer capture in a $100 PowerShot:

- Extends camera functionality without making permanent changes
- CHDK Lua script triggers `.tik` capture when shutter button pressed
- Compressed capture is compiled C code integrated into CHDK
  - Uses a simple pixel value error model to determine when a pixel's value has changed beyond noise variations
  - SIMD algorithm detects change on 4-pixel **UYVYYY** blocks
  - Outputs a pixel change record only as needed; each change record holds spatio-temporal distance from last change record and value of the new pixel block
- File is saved so a PGM preview can be seen without decoding `.tik`
- Can render virtual exposures using either postprocessing with the **`tik`** program or (unfortunately blind) in-camera processing



This work was supported in part under NSF CNS Award #1422811, *CSR: Small: Computational Support for Time Domain Continuous Imaging*

University of Kentucky

TIK
(Temporal Imaging from KY)

- Works with ***any camera***

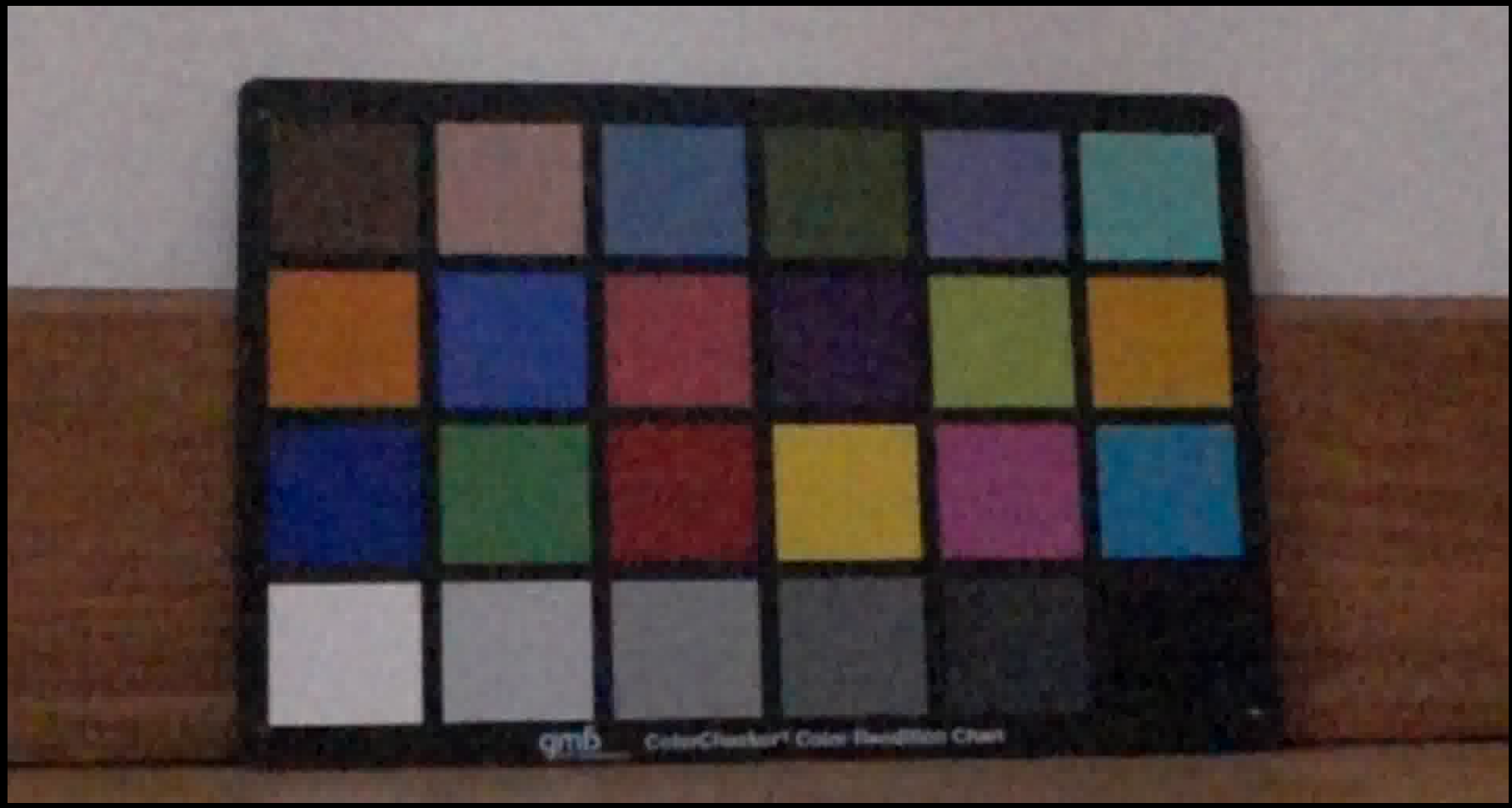
- **`tik`** is an open-source C program
- **`tik`** can construct/use a **value error model**
- **`tik`** can derive a TDCI model of scene appearance from timestamped images:
  - Still images with temporal annotation
  - Video frames with FPS, shutter angle
- **`tik`** can render **virtual exposures**, **video**

University of Kentucky

Creation of an Error Model

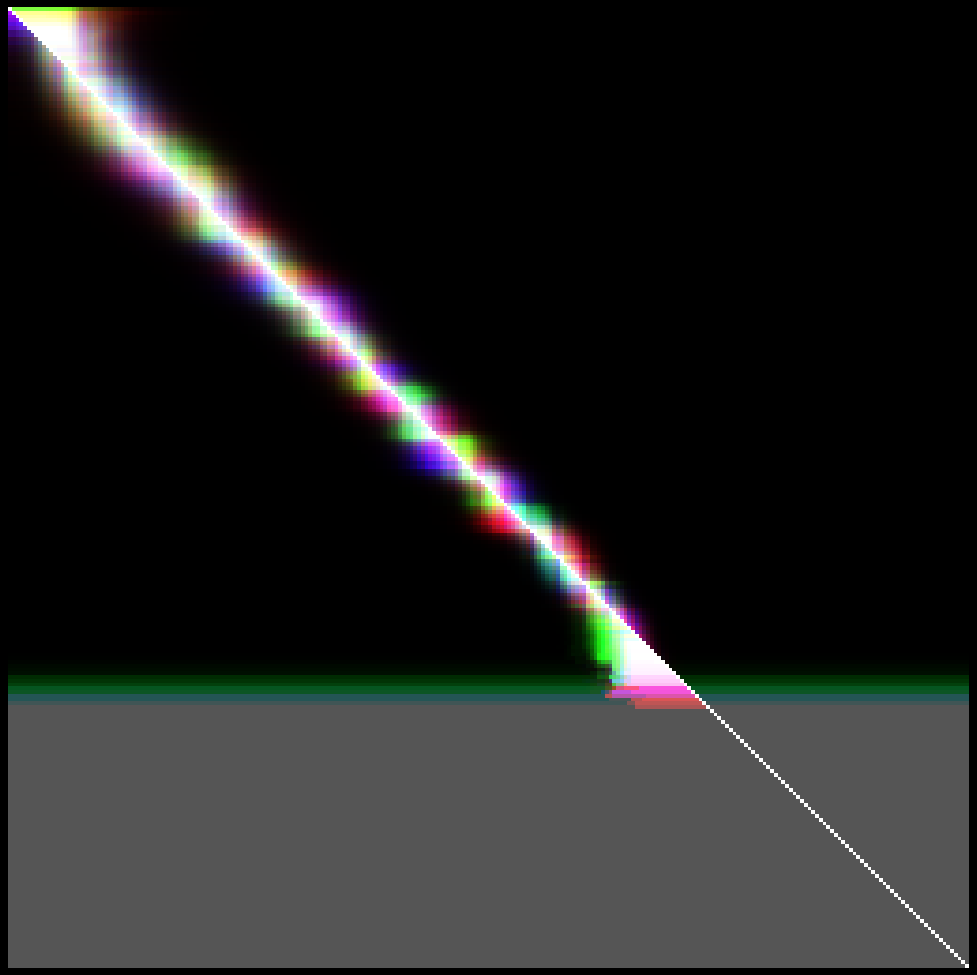
1. Capture video or still sequence of a completely static scene
2. For each pixel value in each color channel, create a histogram of values it transitions to
3. Convert the histogram into probabilities scaled into the range [0..255]
4. Make monotonically non-decreasing
5. Create 256x256 map as X is previous value, Y is next value, RGB are scaled probabilities

960FPS from Sony RX100 IV



University of Kentucky

Empirical Noise Map for 960FPS Sony RX100 IV Video



University of Kentucky

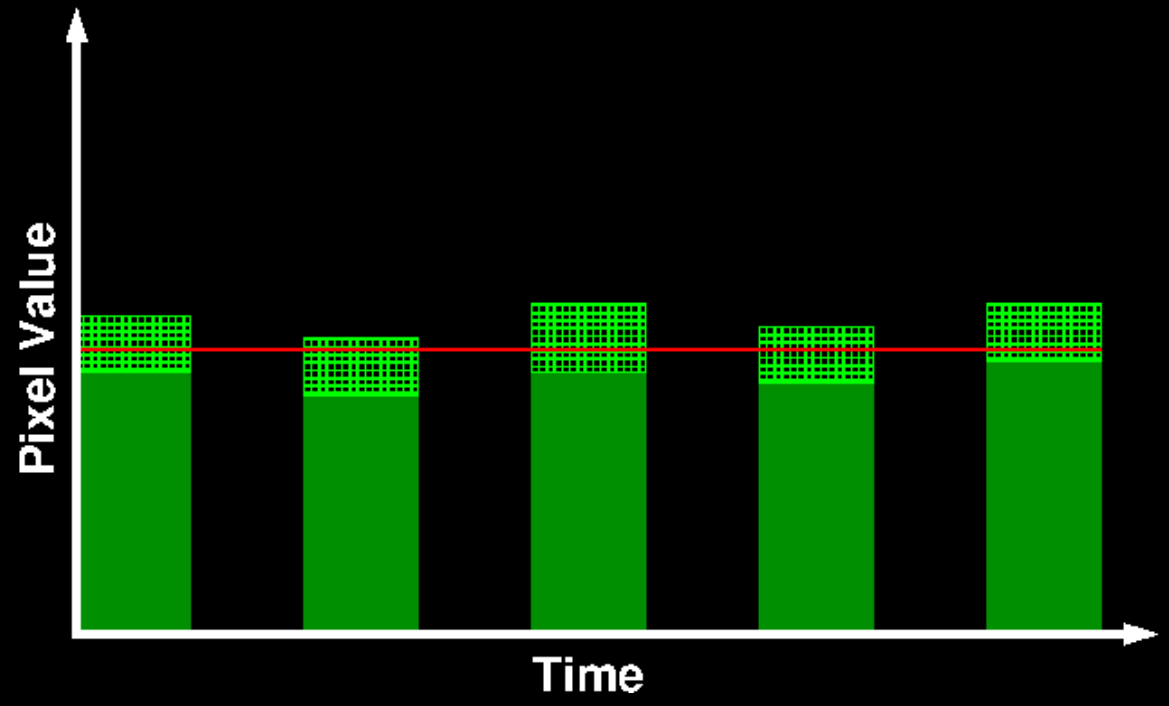
Creation of RGB `.tik`

1. Examine a video frame / still image at a time
2. Scan each image (in roll order, if specified)
3. Check if R,G,B pixel value is same within error model as previous change record for this pixel
4. If pixel changed, write pixel-change record: {spatio-temporal distance to last change record for any pixel, new pixel value};
   If not, improve previous change record value

Temporal Super-Resolution

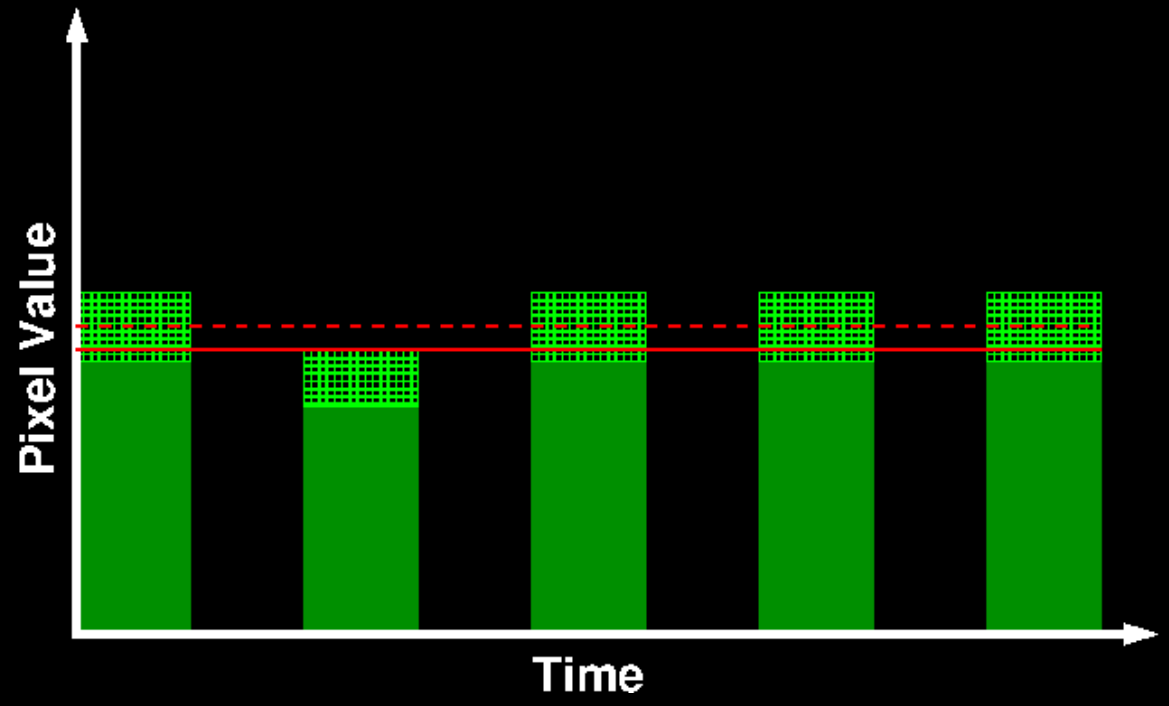
- Super-resolution (SR) means resolving details smaller than sample period (e.g., pixel size)
- Temporal SR (TSR) means resolving time in units smaller than the frame time
- Usually, this is done assuming:
  - Pixel values in each frame are correct
  - Majority of change between frames is from scene changes (scene faster than lighting)
  - Result is multiplied framerate

Changes That Aren't Changes



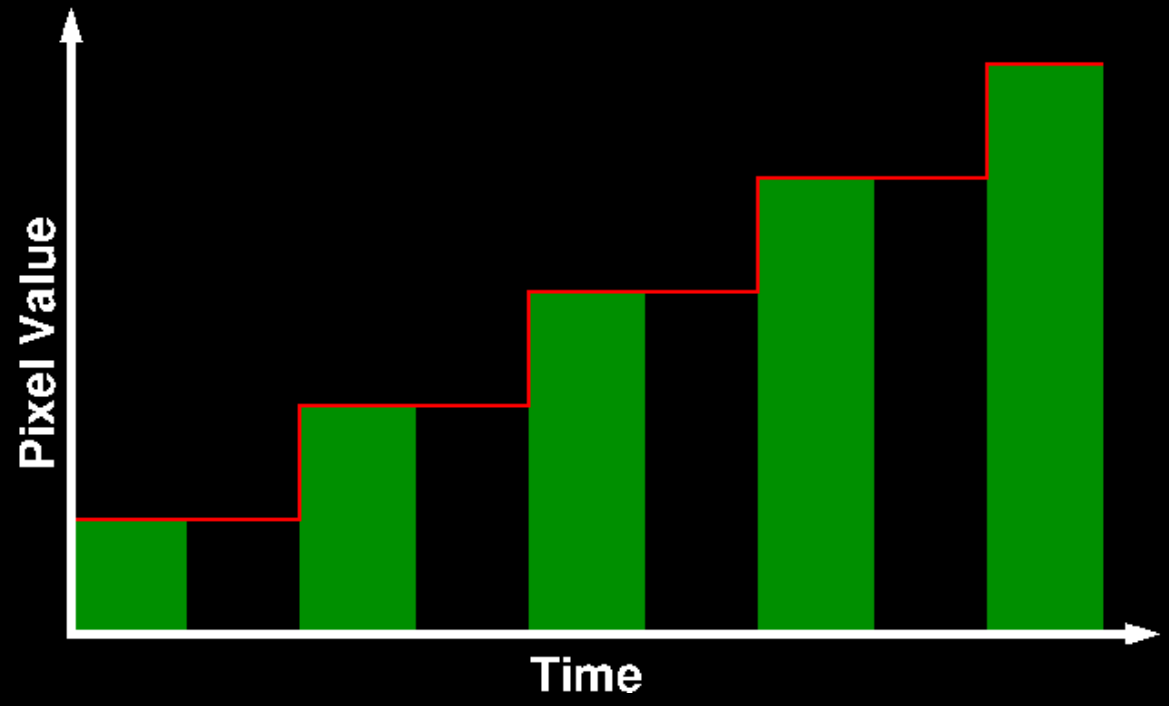
- Within error bounds is probably constant

Changes That Aren't Changes



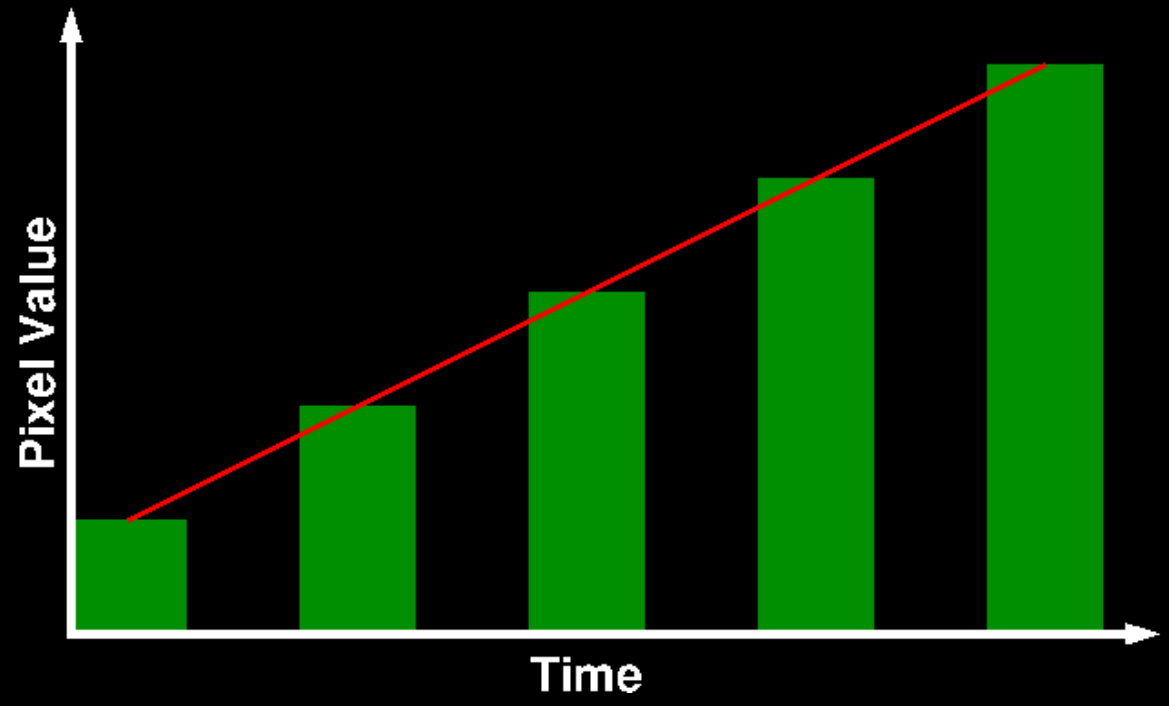
- Not just average; constrain to error bounds

Slopes – Traditional Video



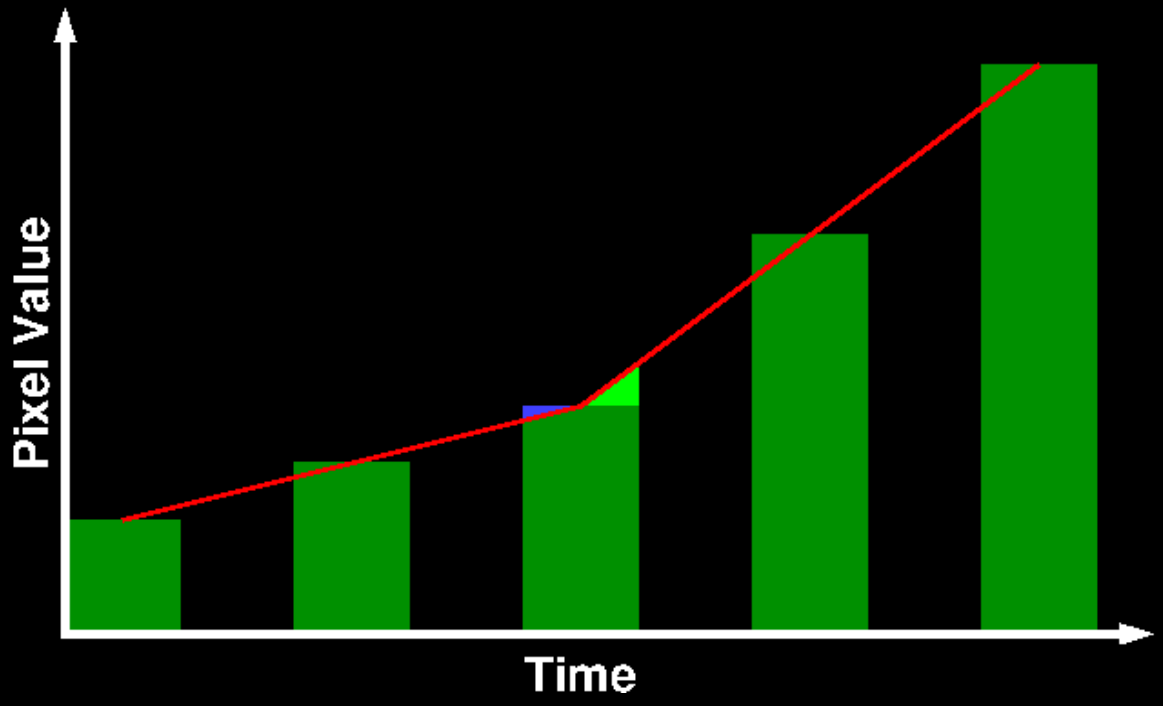
- Ignores difference between integration interval and 1/framerate

Slopes – Other TSR Work



- Assumes precise instantaneous pixel values

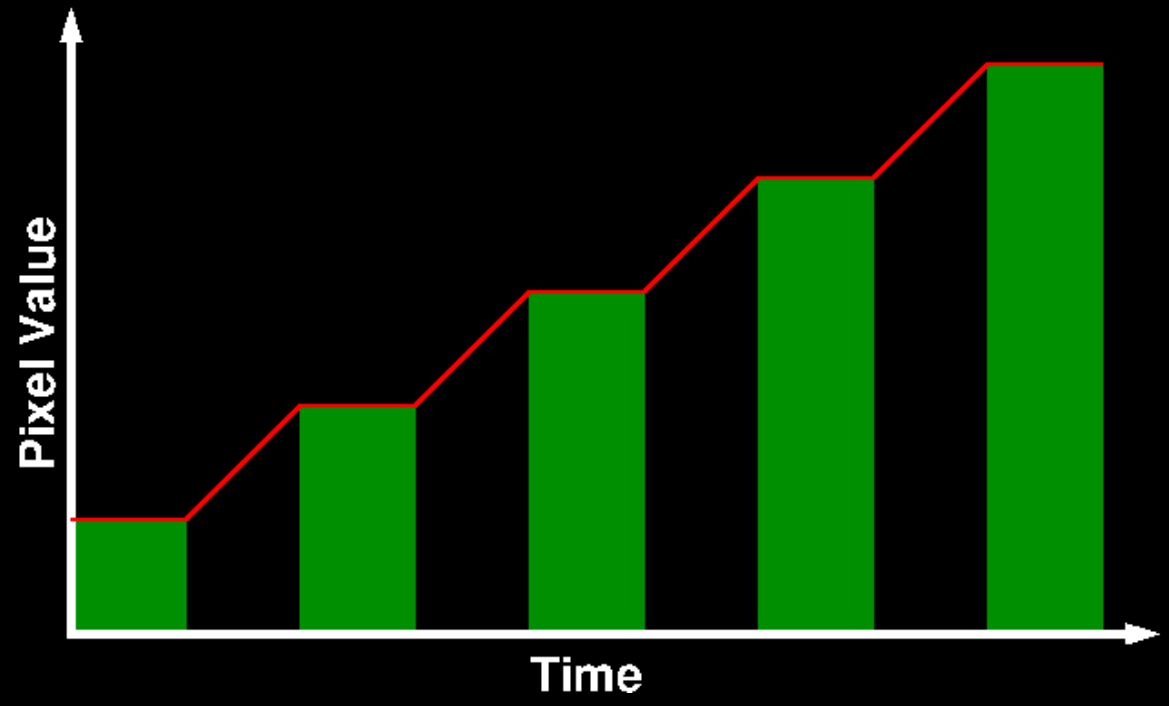
Slopes – Other TSR Work



- Does **not** preserve actual value samples!

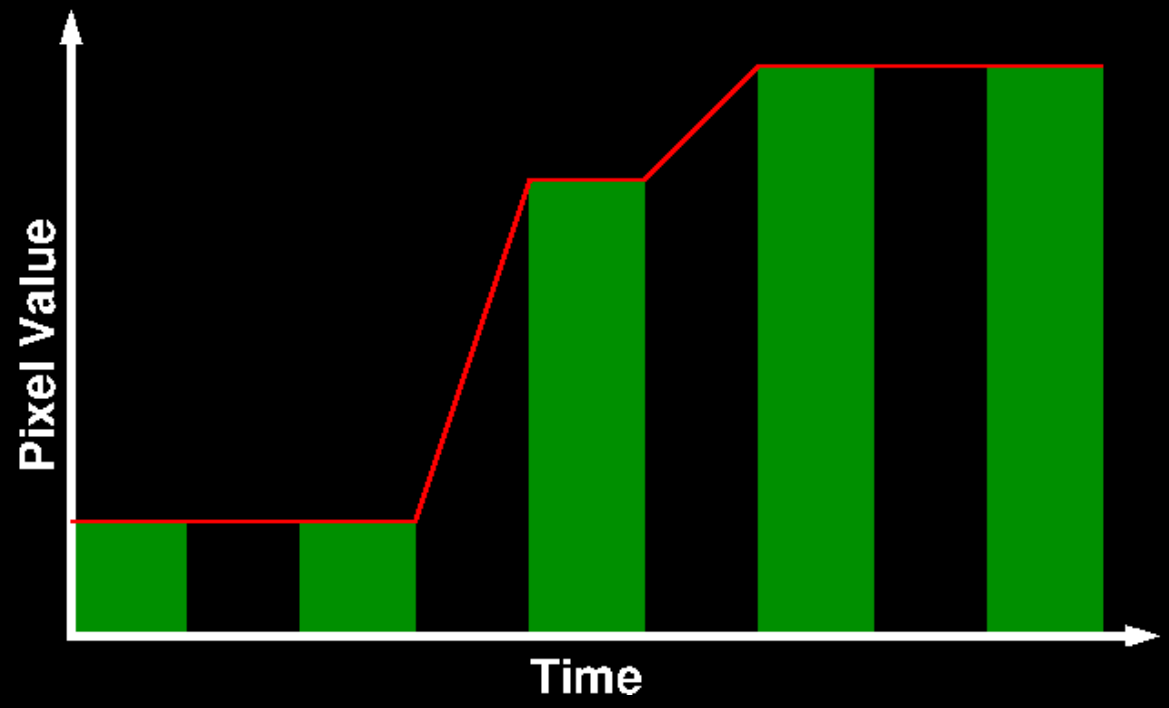
University of Kentucky

Slopes – TDCI TSR



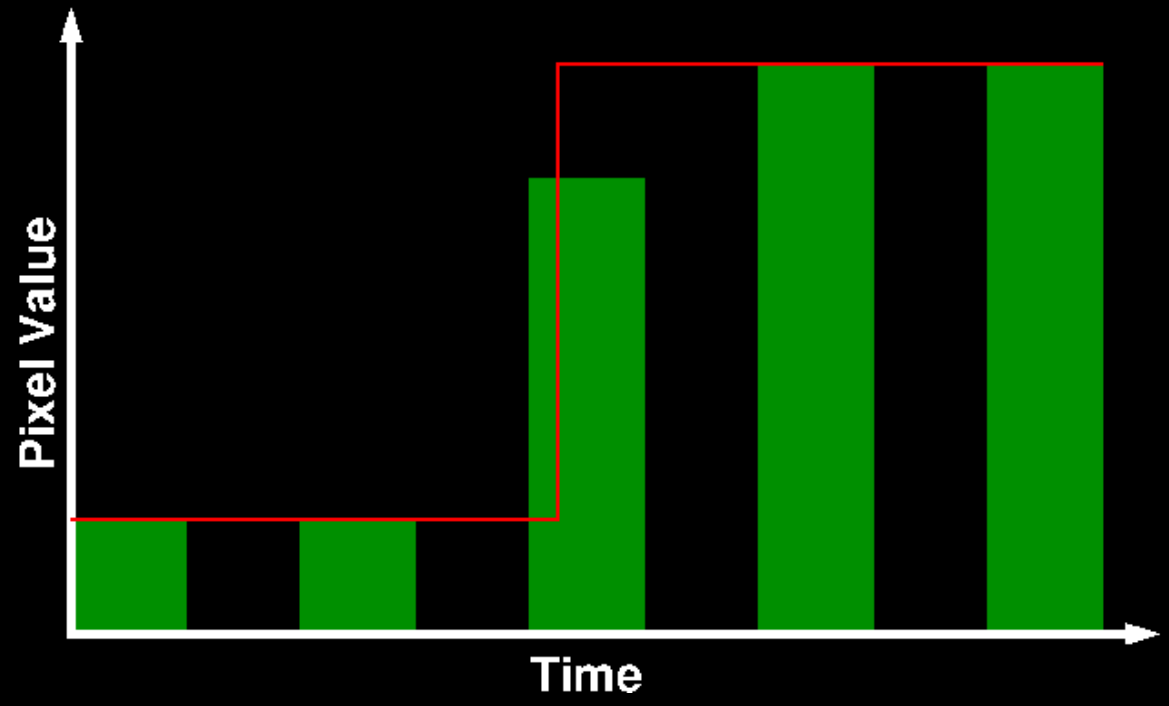
- Pick a curve that preserves value samples

Edges – Linear TDCI TSR



- Edge missed by linear approx. to smoothing

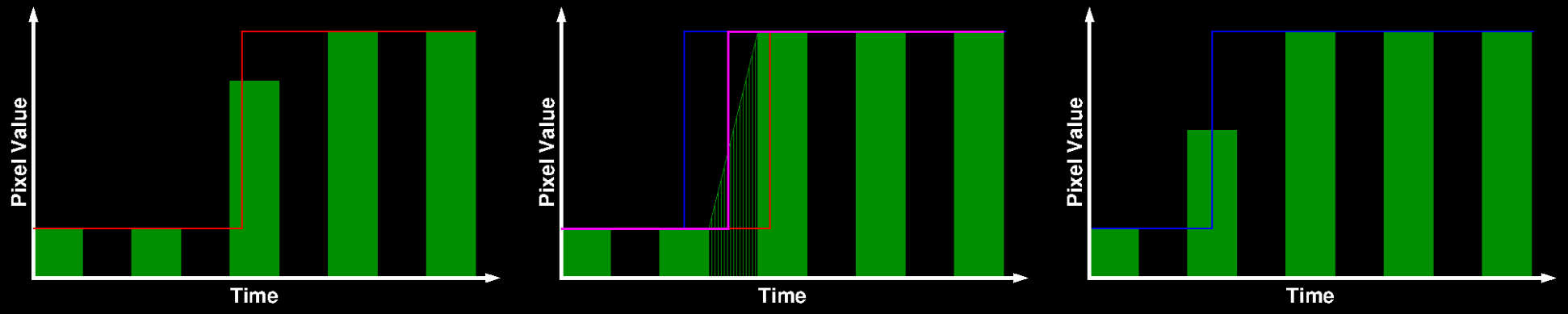
Edges – Edge Localization



- Abrupt transition between stable values allows SR localization of edge!

Temporal Synchronization

- Transitions should be temporally correlated across neighboring pixels – thus, timing can be spatially interpolated
- Consider three neighboring pixels:

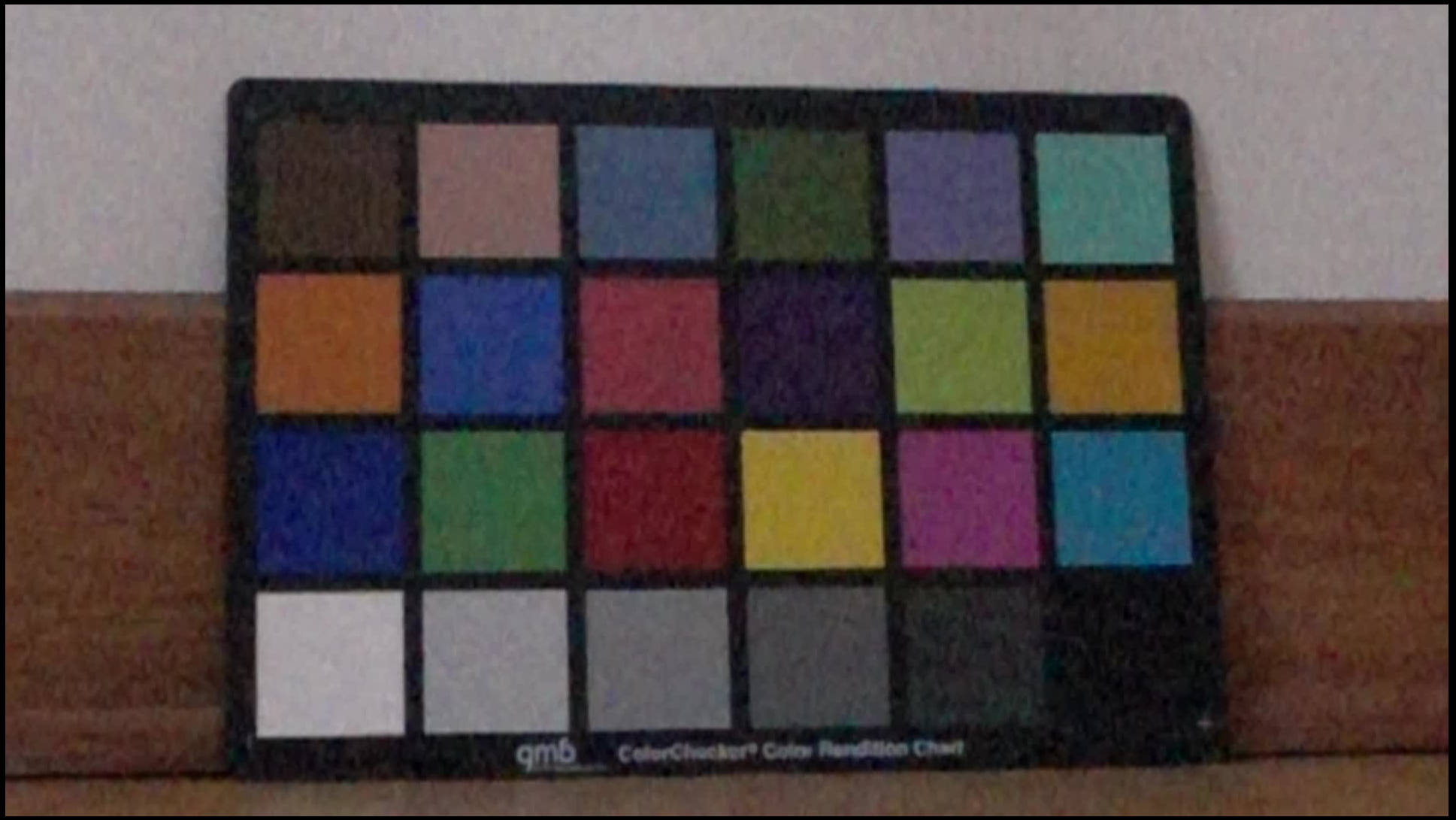


University of Kentucky

Rendering Virtual Exposures

1. Start scanning pixel value change records at beginning of TDCI, tracking current waveform value for each pixel
2. When pixel record time is in virtual exposure interval, integrate under curve
3. Continue scan until all pixels are past interval
4. Map exposure range to output pixel range and output frame
5. Repeat for each virtual exposure

A 960FPS Video Frame



University of Kentucky

A 1/960s Virtual Exposure

University of Kentucky

A 1/24s Virtual Exposure

University of Kentucky

A 1s Virtual Exposure

University of Kentucky

Original Video Capture

- Original 240FPS capture at 1/250s (~346°)

University of Kentucky

Virtual Video Sequence

- Virtual 240FPS video (original rate) with 360°

University of Kentucky

Virtual Video Sequence

- Virtual 24FPS video with 360°

Virtual Video Sequence

- Virtual 100FPS video with 360°

Original Video Capture

- Original 240FPS capture at 1/320s (270°)

University of Kentucky

Virtual Video Sequence

- Virtual 240FPS video with 360°

University of Kentucky

Virtual Video Sequence

- Virtual 25FPS video with 90°

University of Kentucky

Virtual Video Sequence

- Virtual 29.97FPS video with 360°

Virtual Video Sequence

- Virtual 300FPS video with 360°

University of Kentucky

TDCI From A Drone?

University of Kentucky

Virtual Still

- Virtual 1/24s exposure from 24FPS video

University of Kentucky

Virtual Video Sequence

- Virtual 8FPS with 4s shutter (shot @24FPS)

University of Kentucky

Conclusions

- Frame-based imaging is obsolete
  - Too-high data rates for display
  - Unnecessary constraints on capture
- TIK testbed: `.tik` formats + `tik` software proves TDCI feasible & effective
- More to be done on both capture & output...

University of Kentucky