University of Kentucky

UKnowledge

Electrical and Computer Engineering Faculty Publications

Electrical and Computer Engineering

3-4-2015

Frameless Representation and Manipulation of Image Data

Henry G. Dietz
University of Kentucky, henry.dietz@uky.edu

Follow this and additional works at: https://uknowledge.uky.edu/ece_facpub

 Part of the Electrical and Computer Engineering Commons

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Repository Citation

Dietz, Henry G., "Frameless Representation and Manipulation of Image Data" (2015). *Electrical and Computer Engineering Faculty Publications*. 7.
https://uknowledge.uky.edu/ece_facpub/7

This Article is brought to you for free and open access by the Electrical and Computer Engineering at UKnowledge. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Publications by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

Frameless Representation and Manipulation of Image Data

Digital Object Identifier (DOI)
http://dx.doi.org/10.1117/12.2083468

Notes/Citation Information

Published in *Proceedings of SPIE*, v. 9410, Visual Information Processing and Communication VI, article 94100R, p. 1-10.

Copyright 2015 Society of Photo Optical Instrumentation Engineers. One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Henry G. Dietz "Frameless representation and manipulation of image data", *Proceedings of SPIE,* 9410, Visual Information Processing and Communication VI, 94100R (March 4, 2015).

http://dx.doi.org/10.1117/12.2083468

This article is available at UKnowledge: https://uknowledge.uky.edu/ece_facpub/7

Frameless representation and manipulation of image data

Henry Gordon Dietz*
University of Kentucky, Lexington, Kentucky, USA

Abstract. *Most image sensors mimic film, integrating light during an exposure interval and then reading the "latent" image as a complete frame. In contrast, frameless image capture attempts to construct a continuous waveform for each sensel describing how the Ev (exposure value required at each pixel) changes over time. This allows great flexibility in computationally extracting frames after exposure. An overview of how this could be accomplished was presented at EI2014, with an emphasis on frameless sensor technology. In contrast, the current work centers on deriving frameless data from sequences of conventionally captured frames.*

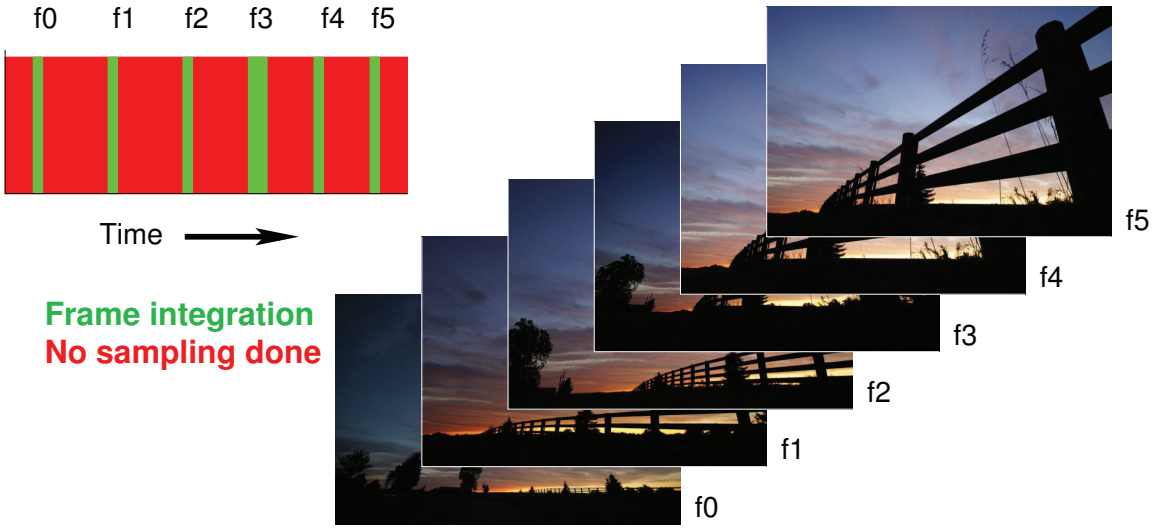


Figure 1. Traditional Image Capture

1. REVIEW OF PRIOR WORK

Capture of a sequence of images, or frames in a movie, is normally accomplished as shown in Figure 1. Each image is captured by exposing the film or sensor for a specific photon integration period of a duration often referred to as the *shutter speed* (S) or exposure time, *Tv*. This duration is primarily a function of the luminance *Bv*, sensitivity *Sv*, and aperture of the lens (either f/number or transmission-efficiency-adjusted T number) *Av*. Tv also is constrained by the range of available shutter speeds and cannot be longer than the reciprocal of the desired frame rate. Using a log scale, such as the *additive system of photographic exposure* (APEX)[4] commonly applied inside cameras, exposure value $Ev = \mathrm{Av} + \mathrm{Tv} = \mathrm{Bv} + \mathrm{Sv}$. The problem is that any scene contains a range of luminances, and thus any Ev selected will be a compromise – degrading image quality in areas for which a different Ev would be appropriate. Further, under many circumstances the Ev will necessitate a Tv that is not 1/frame rate, resulting in either underexposure or potentially large temporal gaps between the images.

In contrast, frameless image capture attempts to construct a continuous waveform for each sensel describing how the Ev (exposure value required at each sensel) changes over time, as shown in Figure 2. We call this *Time Domain Continuous Imaging* – TDCI. Still images are computationally extracted after capture using the average

* University of Kentucky Department of Electrical and Computer Engineering and the Center for Visualization and Virtual Environments. E-mail: hankd@engr.uky.edu

Visual Information Processing and Communication VI, edited by Amir Said, Onur G. Guleryuz, Robert L. Stevenson, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9410, 94100R
© 2015 SPIE-IS&T · CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2083468

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

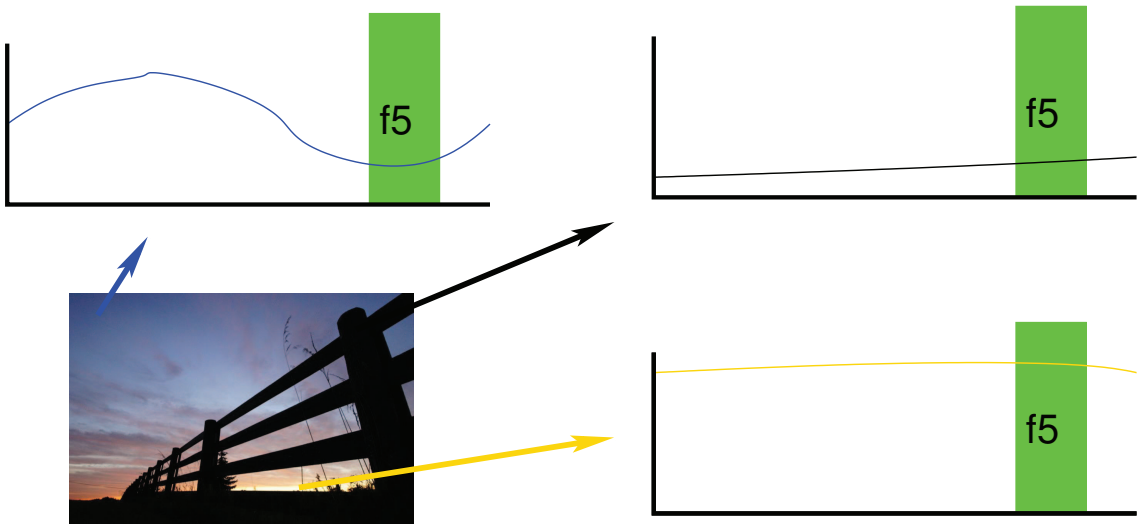


Figure 2. Ideal Time Domain Continuous Imaging (TDCI)

value of each sensel's waveform over the desired interval. Thus, image frames can be extracted to represent any interval(s) within the captured period.

Decoupling aperture and shutter settings from scene brightness has been the dream behind "auto ISO" settings in digital cameras, but making significant ISO changes seriously degrades image quality. TDCI's ability to directly manipulate the interval represented by an image, without being directly constrained by the Ev, represents a fundamental change in how one thinks about imaging. For still images, this means that the virtual shutter speed may be selected freely, even after capture. It is also possible to nudge the virtual exposure interval forward or backward in time to capture precisely the intended action with zero "shutter lag."

For movies, the implications are even greater. The movie industry has suffered a variety of incompatible framerates: cinematic 24FPS, PAL 25FPS, and NTSC 59.94 fields/second. Converting between framerates inevitably causes "stutter" or at least a significant loss in quality. In contrast, the continuous waveforms are essentially a framerate-independent encoding from which essentially perfect frames may be extracted at any framerate significantly below the temporal resolution of the waveforms. Not only can the frames be sampled from the waveforms at any framerate, but there need not be any temporal gaps between frames. The "jumping" objects in pans, and discontinuities in motion in general, occur because conventional movie frames often represent exposure intervals that are significantly shorter than 1/framerate.

The direct capture of continuous waveforms is problematic, but there are a variety of methods by which continuous waveforms of varying quality can be synthesized from appropriately structured discrete samples. Because the extraction of a frame is done using waveforms that are continuous time-varying functions, an Ev estimate is always available, even if a particular sensel was not actually sampled during the desired interval. The result is HDR (high dynamic range) with a low and directly controllable Ev noise level. Neither overexposure nor underexposure occur per se; bright sensels simply provide waveforms with potentially finer temporal accuracy than dark sensels.

The bulk of the EI2014 paper[2] provides an overview of the goals and potential approaches for building sensors with these capabilities. In contrast, the current paper is concerned with TDCI as a model for handling image data that might even come from conventional sensors, and describes the first prototype multi-camera constructed to collect such data. This work is supported in part under NSF grant CNS-1422811, *CSR: Small: Computational Support for Time Domain Continuous Imaging.*

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

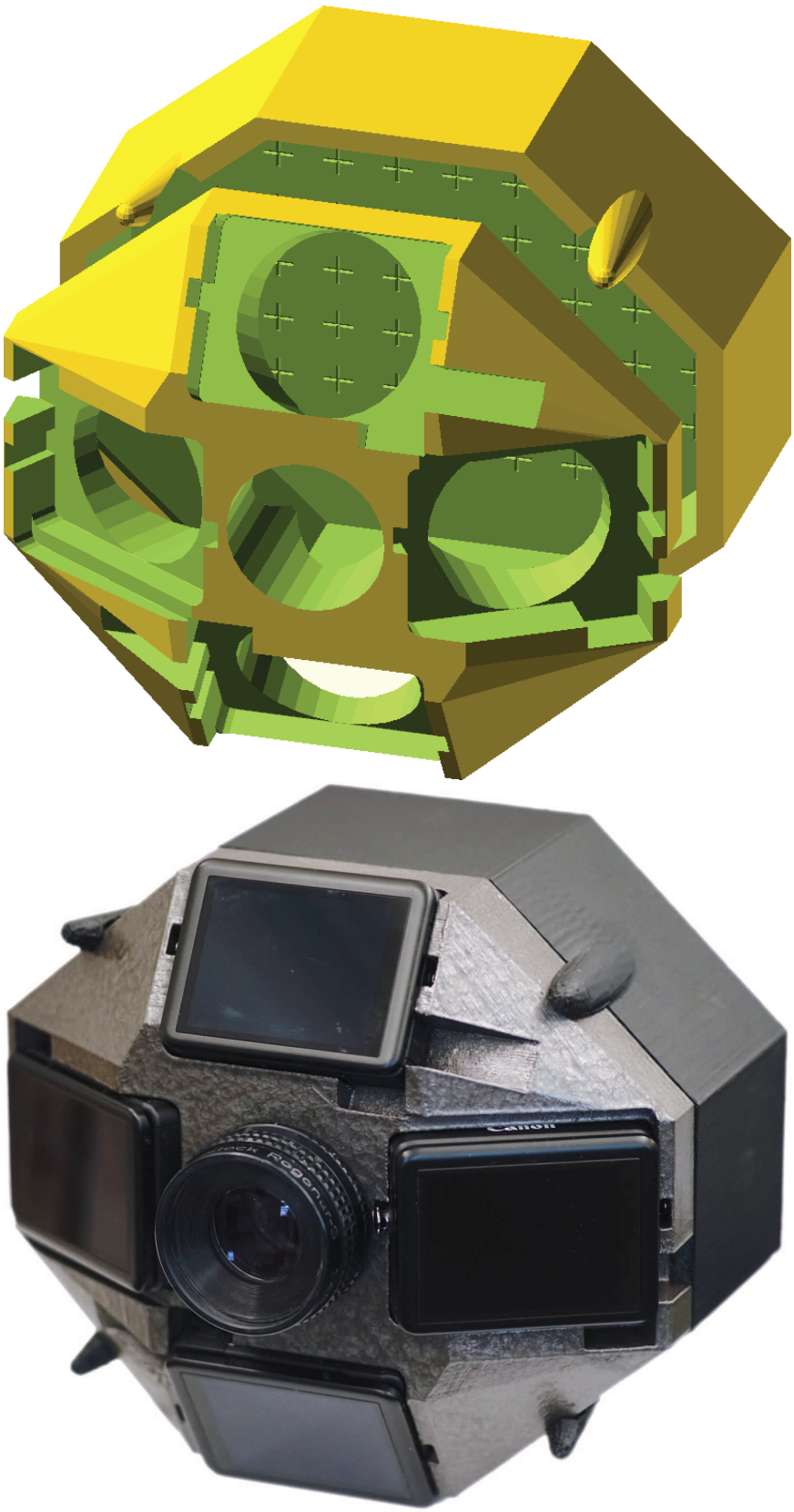


Figure 3. 3D-Printed Multi-Camera Frame Design And Prototype

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

2. MULTI-CAMERA PROTOTYPE

TDCI suffers a "chicken and egg" problem: which should come first – the TDCI stream processing or a sensor that can capture data in the proposed way? Because the stream processing is critical to showing benefits to the approach, and also offers the potential to be applied to image data collected from more conventional sensors, it has been given preference. However, to test processing algorithms it is necessary to have realistic data streams. There are many cameras capable of framerates reasonably approximating the 1/1000s or better temporal resolution of the proposed TDCI sensor. There also are many cameras capable of recording a fairly large dynamic range, at least 13 stops. Unfortunately, none of the cameras available combines both properties. Thus, a prototype camera has been constructed to collect image stream data with approximately the correct temporal and dynamic range properties. Surprisingly, the multi-camera described here accomplishes this at a total cost well under $1000 by using multiple consumer cameras and 3D printing technology.

2.1 The Multi-Camera Design

Over the past decade, it has become standard practice to combine multiple cameras into a single camera system that provides attributes such as a wider view angle or higher spatial resolution.[5,6] The TDCI prototype multi-camera, shown in Figure 3, uses much the same concept, but to instead enhance temporal resolution and dynamic range.

Theoretically, the ideal structure for this multi-camera would be a collection of 1000FPS-capable cameras arranged with beam splitters to all see precisely the same view through a single lens. However, the beam splitter arrangement is mechanically complex, and very awkward to construct using commodity cameras that have permanently-attached lenses. The alternative described here simply arranges multiple commodity cameras photograph an image projected by a single lens – essentially the image formed by a camera obscura on a high-quality screen.

The component cameras in the multi-camera cannot all be optically aligned with the obscura image; in fact, at most one could be. However, angling a component camera to point at the obscura image causes two potentially significant problems:

- The angle imposes an easily-corrected perspective distortion. This distortion does not constitute a different point of view for the scene being photographed, because that point of view is defined by the obscura's lens. Correcting this distortion does imply that some pixels of the component camera will be capturing areas beyond the edges of the obscura image and are thus wasted, but at small angles the distortion and loss in resolution are both minor.

- The angle means that the focus plane of the component camera is not parallel to the obscura image plane. Thus, some portions of the obscura image might not be in focus. This can be remedied by ensuring that there is sufficient depth of field (DoF) to cover the obscura image. Small-sensor compact cameras generally have enough DoF for at least several degrees of angle misalignment.

The 3D-printed frame shown in Figure 3 forms a camera obscura using a large-format enlarger lens, a Rodenstock Rogonar-S 135mm $f/4.5$. This lens is mounted in a focusing tube in the front of the multi-camera and projects its image onto a sheet of high-quality inkjet photo paper which is used as the obscura screen (the choice of screen material limits image quality, and photo paper was excellent except for a specular reflection using glossy paper). Around that lens, four compact cameras are arranged pointing at the obscura screen.

All the multi-camera parts had to be designed and built specially for the particular obscura lens and compact camera combination to be used. This was done using free 3D modeling software (OpenSCAD, Cura, and Pronterface) to drive a MakerGear M2 3D printer. All three 3D-printed parts, back, front plate, and lens focusing mount, were made out of PLA plastic, which is very dimensionally stable. Printed black PLA is not completely opaque and can be somewhat reflective, so the back and rear-facing side of the front plate were painted with flat black latex paint to ensure a good light seal. The outside of the front plate was painted with a hammered-finish metallic bronze spray paint to provide an attractive and more durable surface.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx



Figure 4. Canon PowerShot N, Back And Front Views

2.2 The Component Cameras (Canon PowerShot N)

Very few consumer-oriented cameras support 1000FPS, but a surprisingly large and increasing number of compact, low-cost, consumer cameras now have support for capture of video sequences at framerates of 120FPS or higher. It is not difficult to make camera hardware that can support 1920x1080 recording at 60FPS also provide 1280x720 recording at 120FPS or even higher framerates at further reduced resolution. Many compact cameras support framerates up to 240FPS, but typically at resolutions approximating the CIF (Common Intermediate Format) 352x288 standard that was developed to support applications like video conferencing with easy conversion to both NTSC and PAL formats. The reduction in resolution is necessary to keep the data rate viable, as is fairly aggressive video compression, but neither of these compromises is a fatal problem in producing TDCI test streams.

Even a 240FPS sequence of frames might be sufficiently close to 1000FPS for testing TDCI stream processing. However, the dynamic range captured is not sufficient. In part, the dynamic range is limited by the video compression and even more fundamentally by the dynamic range of the sensor itself, but the basic truth is that 1/240s maximum exposure interval simply does not let enough photons through to obtain a large dynamic range above photon shot noise. The solution is to construct a multi-camera in which all cameras see the exact same scene through the same lens and from the same point of view, but to control the cameras such that exposure intervals are deliberately skewed to enhance both temporal resolution and dynamic range.

Very few cheap consumer cameras provide control mechanisms that would allow appropriate relative timing and skewing of exposures across cameras, but Canon PowerShot cameras using CHDK[1] can. Using CHDK, Lua scripts (and even arbitrary C code compiled into native ARM32 instructions) can be used to implement sophisticated camera control and processing inside each camera. There are a number of ways to obtain fairly precise synchronization of cameras; the CHDK-standard way is sensing of when 5V power is applied to the camera USB connector. CHDK also provides access to a camera-internal 1KHz clock, which can be used to time and trigger events.

Unfortunately, most Canon PowerShot cameras supported by CHDK do not provide video framerates above 30FPS. In response to a technical support request, the explanation was given that the majority of PowerShot cameras use CCDs that simply do not have sufficiently high-speed readout. The PowerShots with CMOS imagers capable of higher framerates up to 240FPS are mostly the larger and more expensive models. However, there is an exception: the PowerShot N (shown in Figure 4). The design and user interface of the N is highly unusual. The entire rear of the camera is a tilt-able touch screen, leaving no obvious way to hold the camera. There is a shutter button, but it is disguised as a ring around the lens which one nudges up or down to fire the shutter. However, this very strange styling makes the camera extremely well suited to our application because the body is as small as the rear LCD, which means multiple cameras can be mounted closer together than normally would

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

be possible. The N also has seen major price drops; it was released at $299.99, but at this writing sells for about $130. The relevant specifications of the N are:

- 12.8MP 1/2.3-inch CMOS imager, 12.1MP 4000x3000 effective
- Continuous full-resolution shooting to 2.3FPS
- Video 1920x1080 @24FPS, 640x480 @120FPS, 320x240 @240FPS ... the record time is limited to 30s per clip in the high-speed modes
- Shutter speed 15s-1/2000s (more under CHDK)
- Sensitivity ISO 80-6400 (more under CHDK)
- Optical zoom 5.0-40.0mm f/3.0-f/5.9, can stop down to f/9
- Rear tilt (max 90°) capacitive touch LCD, 2.8" with 461,000 dots
- MicroSD card, USB 2, and 802.11 wireless for saving images, etc.

The dynamic range recorded is not published, but appears to be no more than about 9 stops within a single captured image under ideal circumstances, with little difference between using camera JPEGs or raw capture enabled by CHDK. High-speed video has perhaps 5 stops of dynamic range. However, by varying the exposure time, the usual high-dynamic range (HDR) methods for combining multiple captures could extend dynamic range to 20 or more stops.

2.3 CHDK Multi-Camera Control

Being able to control the component cameras is critical in this approach. Various board or industrial cameras allow direct control from a host computer, but CHDK allows enough freedom in programming so that the cameras themselves can implement nearly all the control required. The primary difficulty is that CHDK does not allow cameras to directly talk with each other, so it is not feasible to invent and implement new actions in response to unforeseen events. Coordination implementing a pre-planned shooting event is relatively simple to implement.

The key to coordination is the establishment of a shared notion of global time. Each Canon PowerShot has a built-in real-time clock that can be accessed using the Lua call `get_time()`, but it is not of sufficiently high resolution to be useful here, ticking in 1s increments. Neither are the real-time clocks in multiple cameras able to be synchronized to within a fraction of 1s. There is a secondary timing measurement used to control real-time scheduling inside a PowerShot: a 1KHz clock and elapsed tick count. The Lua call `get_tick_count()` returns a count, in 1/1000s units, of ticks since camera reset. Although simultaneous reset of multiple cameras might not result in synchronization of their tick counts, CHDK has provision for very accurately detecting when 5V is applied to the USB connector, so software can record that time and use the difference between the current tick count and that remembered tick count as a global time reference. Accuracy obtained in this way theoretically should be better than +/-1 tick, or about 1/500s.

Given a global timebase, it is possible for each camera to execute a pre-planned sequence of captures. Many such capture schedules are possible. Here are a few issues that suggest specific types of schedules:

- The high-speed video modes use auto exposure with no overrides available. Fortunately, the shutter speed for 240FPS video starts at 1/251s and increases only to avoid overexposure (varying the f/number and ISO first, as shown by the parameters logged in the `.MOV` file EXIF data), integrating light for about 95% of the frame time. Thus, skewing two camera video sequences by 1/500s allows approximately recovering 480FPS by simple differencing. Skewing by 1/1000s with four cameras could allow approximately recovering 960FPS.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

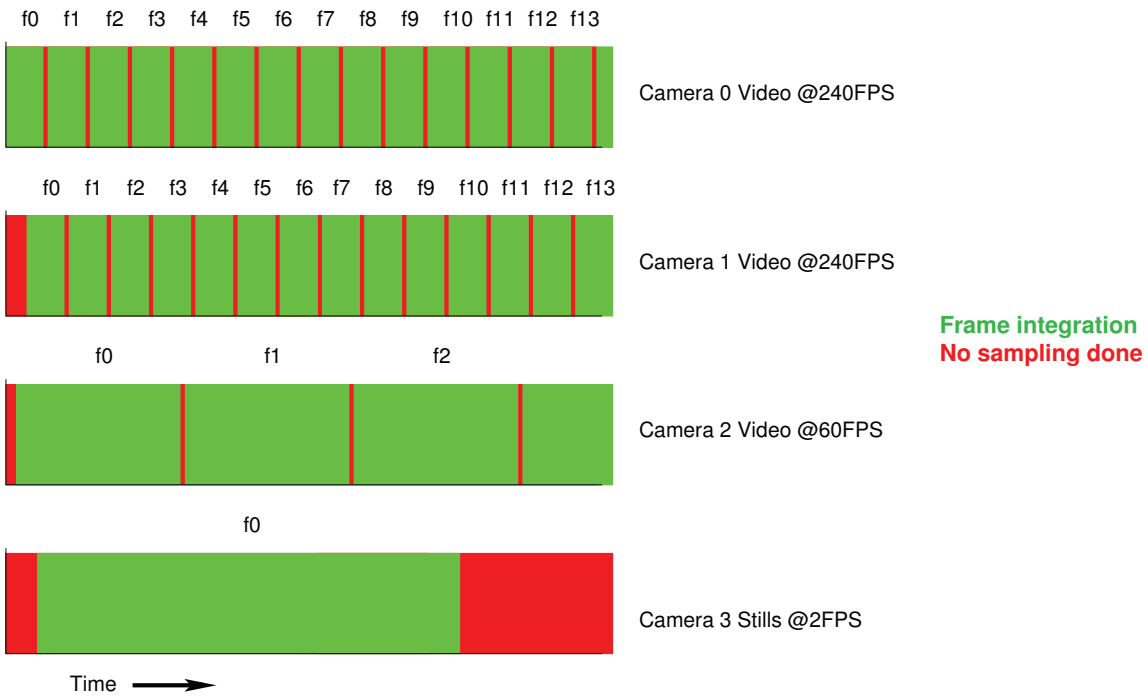


Figure 5. Overview Of Multi-Camera Control

- The high-speed video really captures only the top 4-5 stops of scene content. In order to get higher dynamic range, one or more cameras can be set to record video at a lower framerate or even stills, and exposure settings can be deliberately set in a pattern that favors collecting the shadow detail the high-speed videos lack. The higher-quality low-framerate data is assumed to be correct unless the high framerate data differs by more than noise, proving that the level must have changed.

- The high-speed video modes only can record continuously for about 30s before stopping to complete flushing the video to the microSD card. In order to ensure continuous capture, there should be an overlapping hand-off of video recording from one camera to another.

The above issues suggest an exposure schedule which could be optimized using analysis of signal/noise ratios.[3] A potential schedule is shown in Figure 5. The 240FPS skewed pair are cameras 0 and 1 in the figure. Cameras 2 and 3 simply extend the dynamic range downward with poorer temporal accuracy. Before cameras 0 and 1 reach their 30s video clip capture limit, cameras 2 and 3 would switch to the patterns for 0 and 1. After completing writing their video clips, cameras 0 and 1 would adopt the patterns formerly used by 2 and 3. In this way, 480FPS video can be derived at all times and the dynamic range is improved by lower frequency augmentation of the darker portion of the dynamic range.

There are a variety of additional tricks that can be used to intelligently adapt exposure to the circumstances. For example, a tiny LED display could be mounted inside the multi-camera to flash gray code values in the view of the cameras to tag frames with very precise timing information. It is also possible to simply tether all the cameras to a notebook or board computer (e.g., Raspberry Pi) and use `chdkptp` to control them. Unlike most tether support, `chdkptp` not only allows moving files back and forth and triggering capture, but also executing and interacting with arbitrary Lua code inside the camera. Standard USB 2 uses a slotted protocol within a 120us window – about 8 times finer than the camera-internal tick resolution – so standard USB messages could be used to synchronize the cameras.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

3. COMBINING FRAME SEQUENCES

The multi-camera recorded data image data inherently consists of multiple frame sequences. This section describes a simple way to convert these frame sequences into a single frame sequence with a higher framerate and increased dynamic range. The conversion works in two stages:

1. Synthesize the low-dynamic-range frames for the highest continuous framerate
2. Augment those frames to enhance the dynamic range using lower-framerate data

With two simplifying assumptions, the synthesis of the highest continuous framerate data is straightforward using just the interleaved samples from the high-speed videos:

- Images are being captured with a shutter speed that is essentially identical to the reciprocal of the framerate. For the PowerShot N at 240FPS, this is ignoring the difference between 1/240s and 1/251s. If significant lighting changes occur in the 1/5476s unsampled intervals (the 1/240-1/251 gaps), they will cause error in the frameless data.
- If there are k cameras capturing data at the high-speed video rate of r FPS, their samples are temporally skewed by precisely $(1/r)/k$. For example, two 240FPS cameras would be temporally skewed such that the sequence of one starts 1/480s after the sequence of the other. Four cameras at 240FPS would have start times skewed by 1/960s, 2/960s, and 3/960s.

If there is only one high-speed video camera, it defines the low-dynamic-range frame sequence. To derive a sequence of frames at $k*r$ FPS from k cameras each recording at r FPS requires differencing frames. The problem is that there is a missing initial value. Consider deriving $2*r$ frames, u, from the frames captured by the first two cameras in Figure 5. Using the frame names to each represent a value linearly proportional (linear gamma) to the number of photons arriving in the interval represented by the frame:

$$\begin{aligned} f0_0 &= u0 + u1 \\ f0_1 &= \quad u1 + u2 \\ f1_0 &= \quad\quad u2 + u3 \end{aligned}$$

The problem is that u0 is unknown. If it were known, then the u frame values could be determined by u1=$f0_0$-u0, u2=$f0_1$-u1, ... This simple differencing works both forward and backward from any known u value. This exact same relationship holds not only for frames as a whole, but for individual pixels in the same position across frames.

Suppose that $f0_0$=$f0_1$. That would imply u0+u1=u1+u2, i.e., u0=u2. Suppose additionally that $f0_1$=$f1_0$; then u1=u3, which further implies u0+u1=u2+u3 because $f0_0$=$f1_0$. This can occur either if u0=u1=u2=u3 or if the u values have a pattern with a frequency of at least r Hz. If we assume that any frequency present would be lower than r Hz, that implies u1=u2=$f0_1/2$. Thus, any set of three temporally-adjacent samples that have identical values for a particular pixel define a point from which differencing can be applied forward and backward. Different pixels may have different reference frames. It would also be possible to force a reference for all pixels by starting or ending the sequence with the obscura lens covered by a diffuser or an opaque cap.

Enhancing the dynamic range of the derived frame sequence, u, is fairly straightforward using processing closely related to standard multi-exposure HDR.[7] The exposure values for the lower-framerate samples need to be appropriately scaled for combining with those from u. Pixels that have values below the noise floor in the u frame sequence simply get their values replaced by the interpolated average values over the corresponding time period for the lower-framerate samples. Pixels that have some quality in both the u frame sequence and the lower-framerate images can be adjusted based on the quality-weighted average value. In Figure 5, frame $f0_2$ spans frames u0 through u9, and if the data were free of noise, $f0_2$=u0+u1+u2+u3+u4+u5+u6+u7+u8+u9. However, we expect more noise in the u values than in $f0_2$, so $f0_2$ should be a more accurate estimate of the sum and the values of u0..u9 should be adjusted to bring the sum closer to $f0_2$. Given that $SNR(x)$ is the estimated

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

Algorithm 1 Constant Expectation Encoding

```
r=0                // record number for this pixel
value[r]=frame[0] // initial pixel value
duration[r]=1     // estimated duration is 1 tick
t=1                // time index
while (t in frame[]) {
  v=frame[t]       // value of pixel at time t
  if (within_noise(v, value[r])) {
    // update current record
    duration[r]=duration[r]+1
  } else {
    r=r+1          // new record
    value[r]=v
    duration[r]=1
  }
  t=t+1             // next frame time
}
```

signal/noise ratio of signal x, and sample f spans a set of u called ui, an improved value for uj in ui can be computed as:

$$uj' = \frac{uj}{\sum ui} * \frac{SNR(f) * f + SNR(uj) * \sum ui}{SNR(f) + SNR(uj)}$$

There are various other viable formulations, some of which could produce higher temporal accuracy. It also would be possible to independently convert each camera's frame sequence into a TDCI stream and then combine the streams. However, the above approach is a reasonable starting point for future research.

4. CONVERSION TO FRAMELESS REPRESENTATION

As the EI2014 paper[2] suggested, given a constant framerate sequence of frames, conversion to a TDCI representation can be relatively straightforward. The processing described in the previous sections produces a frame sequence that is indistinguishable from one produced by a single, imaginary, camera with the higher framerate and increased dynamic range. The simple algorithm described here is viable for regular frame sequences no matter the source.

Storing (or transmitting) a frameless, continuous, waveform for each pixel is prohibitive without highly effective compression. There are many ways to do this, and the research into these representations is at too early a stage to define a standard format. In common to all formats is the basic concept of having a logical stream per pixel with each item in a pixel's stream recording expected behavior and a duration for which that expectation is valid. The alternative representations also share the property that deviation from an expectation causes a new expectation only if the change is statistically significant – i.e., if the change is greater than the variation which could occur by noise (especially photon shot noise) alone.

The simplest processing, Algorithm 1, records the number of frames for which each pixel value is maintained. Only slightly more complex is Algorithm 2, which records when the slope of pixel value change is significantly altered in a new frame. The slope is recorded not as a slope per se, but as a delta being applied over the duration recorded. Once the frame sequence has been transformed into a waveform per pixel by either of the above algorithms, the records can be further compressed and stored interleaved within a file is an order that is sorted first by the frame number in which it becomes active and second by position within a fixed pixel walk order.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx

Algorithm 2 Constant Slope Expectation Encoding

```
r=0               // record number for this pixel
p=0               // previous pixel value = 0
delta[r]=frame[0] // step from previous
duration[r]=1     // duration of step
t=1               // time index
while (t in frame[]) {
  v=frame[t]      // value of pixel at time t
  i=duration[r]+1 // potential new interval
  e=i*delta[r]/duration[r]  // estimated change
  if (within_noise(v, p+e)) {
    // update current record
    delta[r]=v-p  // actual difference
    duration[r]=duration[r]+1
  } else {
    p=p+delta[r]  // new previous value
    r=r+1         // new record
    delta[r]=v-p
    duration[r]=1
  }
  t=t+1            // next frame time
}
```

5. CONCLUSION

This paper has presented a very preliminary overview of how conventional consumer cameras can be used to create good quality TDCI stream data. The key to producing high-framerate data with a high dynamic range is use of a novel multi-camera in which multiple cameras sample the image produced by a camera obscura. The processing needed to combine the multiple frame sequences, and to encode the resulting frame sequence as a TDCI stream, is also discussed.

A first prototype multi-camera has been constructed as discussed here, but has not yet produced sample TDCI streams. It will be shown in the poster presentation of this paper. A revised second prototype is being developed and future work will detail results with it and algorithms using the TDCI streams to synthesize frames, as well as methods to further enhance the TDCI streams (for example, by using information from adjacent pixels).

REFERENCES

1. *Canon Hack Development Kit (CHDK).* http://chdk.wikia.com/wiki/CHDK.
2. Henry Gordon Dietz. Frameless, time domain continuous image capture, 2014.
3. S.W. Hasinoff, F. Durand, and W.T. Freeman. Noise-optimal capture for high dynamic range photography. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 553–560, June 2010.
4. Douglas A Kerr. Apex-additive system of photographic exposure. *Issue*, 7(2007.08):04, 2007.
5. Daniel L. Marks, Hui S. Son, Jungsang Kim, and David J. Brady. Engineering a gigapixel monocentric multiscale camera. *Optical Engineering*, 51(8):083202–1–083202–13, 2012.
6. Jonas Pfeil, Kristian Hildebrand, Carsten Gremzow, Bernd Bickel, and Marc Alexa. Throwable panoramic ball camera. In *SIGGRAPH Asia 2011 Emerging Technologies*, SA '11, pages 4:1–4:1, New York, NY, USA, 2011. ACM.
7. Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting.* Morgan Kaufmann, 2010.

Downloaded From: http://proceedings.spiedigitallibrary.org/ on 12/09/2015 Terms of Use: http://spiedigitallibrary.org/ss/TermsOfUse.aspx