Scene Appearance Change As Framerate Approaches Infinity

Henry Dietz, Zachary Snyder, John Fike, and Pablo Quevedo; Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky

Abstract

The primary goal in most uses of a camera is not to capture properties of light, but to use light to construct a model of the appearance of the scene being photographed. That model should change over time as the scene changes, but how does it change over different timescales? At low framerates, there often are large changes between temporally adjacent images, and many are attributed to motion. However, as the scene appearance is sampled in ever finer time intervals, the changes in the scene become simpler and eventually insignificant compared to noise in the sampling process (e.g., photon shot noise). Thus, increasing the temporal resolution of the scene model can be expected to produce a decreasing amount of additional data. This property can be leveraged to allow virtual still exposures, or video at other framerates, to be computationally extracted after capture of a high-temporal-resolution scene model, providing a variety of benefits. The current work attempts to quantify how scene appearance models change over time by examining properties of high-framerate video, with the goal of characterizing the relationship between temporal resolution and effectiveness of data compression.

Introduction

There are now many consumer cameras capable of 120 frames per second (FPS) or faster video at modest resolution. The Digital Photography Review website has a camera feature search facility[1] that allows searching for cameras supporting high-speed video capture; the results of searching there are summarized in Figure 1. Although data does not seem to have been consistently entered after 2013 and there are some errors in the older data (such as incorrectly crediting four Canon cameras with 480FPS and failing to list Sony models at all levels), it is obvious that higher FPS video capture is becoming an increasingly popular feature. This is an industry-wide trend; in order of first model introduction, the feature search data includes high-framerate models from most major camera manufacturers, including Casio, Fujifilm, Canon, Nikon, Panasonic, Samsung, Leica, Kodak, Olympus, and Pentax.

The key to supporting high framerates has been providing sufficient bandwidth for getting image data off the sensor, processed, and stored. As these capabilities were enhanced to support video capture at 2K HD (1920x1080 pixels) and now 4K UHD (3840x2160 pixels), many cameras have become capable of higher framerates at somewhat reduced resolutions. A list of under-$8000 cameras capable of some form of slow motion video capture[2] lists 59 models. This trend affects consumer, professional movie, and industrial cameras alike. For example, various RED movie cameras are able to record raw sensor

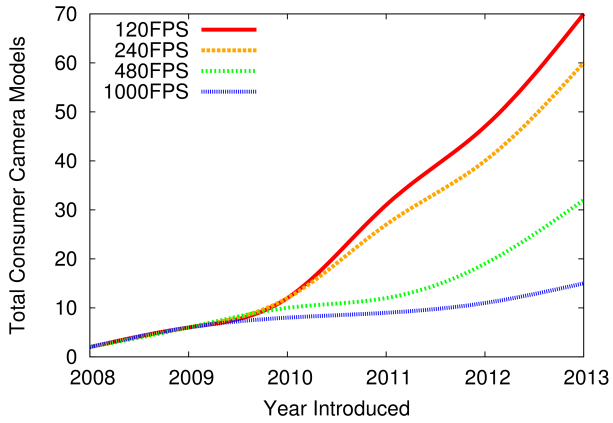

Figure 1. *High-framerate consumer cameras*

data at up to 100FPS at 6K (6144x3160 pixels) native resolution and 300FPS at 2K. Recently, the traditional high-speed industrial camera brands have been joined by Kickstarter-spawned edgertronic and FPS1000 trying to bring down the price of cameras capable of up to about 18,000 FPS.

Of course, the obvious catch is that resolution must be greatly reduced to achieve these high video framerates *within the system bandwidth available*. However, it is precisely that premise which the current paper questions – at least when framerates go to 1000FPS and beyond.

High-framerate photography generally has been associated with capture of scenes containing very fast-moving objects (e.g., a bullet in flight) or other rapidly changing phenomena (e.g., fast chemical reactions). Naturally, being able to use a fast enough shutter speed generally has required specialized lighting. High-speed strobes and extremely bright lights have enabled framerates as high as a million frames per second. The question this paper poses, and attempts to answer, is simply *what happens if the framerate is increased to capture ordinary scenes without using special lighting?*

Back to the future

To understand why one would want to capture ordinary scenes at very high framerates, it is useful to go back to the digital camera that probably deserves primary credit for introducing consumers to high-FPS photography: Casio's Exilim Pro EX-F1[3], which was introduced in January 2008. This 12X superzoom camera could record full HD video (1920x1080 pixels) at 60 fields/second and up to 1200FPS video at 336x96 pixel resolution. It also offered features for relatively high-speed continuous

capture of still images at the full 2816x2112 pixel native sensor resolution. When capturing full-resolution still images at even a few FPS was a good rate, it could handle 60FPS, including the ability to continuously "prerecord" images and save those starting a specified time before the shutter button was fully depressed. This prerecord feature was explicitly marketed as allowing the user to select when the shutter was fired after the event had occurred, so that one could always be certain of capturing the scene at precisely the desired moment. In other words, the Casio Exilim Pro EX-F1 did not just have support for high-framerate movies – *it suggested that perhaps a camera should simply record as fast as it can and allow the user to defer picking the exposure interval until after the event has happened.*

To infinity and before

Where Casio limited the choice of exposure interval to selecting *one* of the frames captured, our research group has been investigating a similar, but more extreme, approach called Time Domain Continuous Imaging (TDCI)[4]. TDCI replaces the entire concept of capturing frames with the idea of each pixel independently capturing samples from which a continuous waveform is created to describe how its value varies over time. Using these waveforms, a still image can be synthesized for any virtual exposure interval by simply computationally integrating the area under each pixel's waveform for the designated interval. The potential advantages are huge:

- Exposure interval can be *smoothly* adjusted after capture: virtual shutter speed is independent of exposure, and the user can nudge exposure interval forward/backward to get the precise moment with zero "shutter lag"
- HDR (High Dynamic Range) with integration period <, =, or > exposure interval: never lose data to overexposure, temporally interpolate underexposed pixels
- Framerate-independent movies: no more "stutter" in displaying at cinematic (24FPS), PAL (25FPS), and NTSC (59.94 fields/s) framerates
- Artifact-free movie pans and motion in general: computationally integrating means no temporal gaps between frames (e.g., no "jumping" objects in movie pans)

However, we have not yet been able to construct the new type of sensor needed to directly perform TDCI capture. As a first step, we constructed a simple multi-camera system[5] that uses deliberate skewing of the exposures of the four Canon PowerShot N component cameras, all sharing the same point of view, to synthesize a TDCI representation. The current prototype, FourSee, is shown in Figure 2.

Still, using multiple consumer cameras is more awkward than using just one. With inexpensive cameras producing high quality video at up to 1000FPS, and higher framerates sure to come, we began to investigate the idea of synthesizing TDCI data from a single high-speed video capture. If the temporal gaps between video frames are small compared to 1/framerate, then merging data from a sequence of high-framerate frames can be used to synthesize a nearly-continuous waveform for each pixel. Alternatively, the set of frames within the desired time window can simply be "stacked" – combined to produce a

Figure 2. *FourSee multi-camera TDCI prototype*

lower-noise, larger dynamic range, image much as is done for astrophotography[6].

With the above processing intent, the primary advantage in using a higher capture framerate is an improvement in temporal accuracy, and one would also expect a modest improvement in signal/noise ratio. Unfortunately, the bandwidth, processing, and storage cost would seem to increase dramatically with framerate, making this approach impractical for high framerates... or is it? There are three fundamental properties that should work to reduce the volume of new data per frame as framerate is increased:

1. Most of the pixels in real-world scenes do not change appearance arbitrarily quickly. Thus, once the framerate has exceeded that speed, further increases in framerate do not produce any data about scene change, although they might slightly improve sampling signal/noise ratio.
2. Photon shot noise, statistical variation in photon emission rate, is always present and the number of photons counted by a pixel for a given interval is essentially independent of framerate applied within that interval. Thus, once the framerate has become high enough to exceed the fastest statistically-significant rate of change of photon arrival rate, additional temporal resolution delivers almost no additional useful information: the variations seen are simply random noise.
3. Some scene components might be changing very quickly, but if there are not enough photons to sample that change, the scene change cannot be reliably recorded no matter how high the framerate. This is why high-speed video is commonly associated with intense lighting; without it, faster changes in scene content cannot add any information content as framerate is increased.

In summary, although higher framerates certainly will appear to carry more information, the fraction of that information that is purely noise dramatically increases. It is expected that the amount of additional useful information obtained as framerate is increased will approach a constant determined by the above three effects. This implies that, with an appropriate model allowing the frame data to be filtered to remove noise that does not contribute to the recording of the scene, the additional data incurred

by huge, or even infinite, increases in framerate can be expected to be finite and relatively small. This paper reports on a subset of the experiments that we conducted to determine if this prediction is consistent with real-world behavior.

Experimental Procedure

Although high-framerate video found on the Internet was also used, the experiments conducted center on high-speed video that was captured to answer this question using two cameras:

- Canon PowerShot N: This is a very inexpensive camera (cost was approximately $130 new) capable of up to 240FPS for 30 seconds, but at a mere 320x240 pixel resolution and suffering significant compression artifacts. The low-quality video encoding limits absolute quality of stacked frames because the artifacts are largely in fixed positions across many frames, and thus are not entirely removed by stacking.
- Sony Cyber-shot DSC-RX100 IV: This $1000 camera uses arguably the most advanced sensor technology in a consumer camera, an Exmor RS stacked CMOS sensor that is bonded to a large buffer memory to provide very high bandwidth for short bursts. It is capable of up to 1000FPS (960FPS in NTSC mode) for 2 seconds at a capture resolution of 1136x384, but upscales that to 2K HD video, which has the effect of reducing encoding artifacts somewhat. The camera also allows exposure control during high-framerate video recording, which we employed to set a shutter speed of 1/1000s for 960FPS video so that the inter-frame temporal gap was negligible.

The test procedure was:

1. Record a normal scene using the fastest framerate mode with ordinary lighting.
2. Convert the video to a still image sequence at the full framerate. Tonal linearity of each frame may also be corrected.
3. For each potential framerate, create a still image sequence at the desired framerate by simple stacking[6] of the inherently aligned frames; for example, converting 960FPS video into 240FPS would mean stacking each sequence of four images to create one resulting image.
4. For each potential framerate, encode the video as a compressed TDCI stream and record the size of the final stream.

The TDCI encoding process used is basic, but does incorporate a model that allows it to avoid encoding information-free noise. At the front of the TDCI file, there is a header recording the size of the frames. Each pixel is assigned an ID based on location. The remainder of the file consists of pixel update records and time markers. Although TDCI usually records time in a more precise unit, time here was counted one "tick" per frame. Every pixel update record is a pixel ID followed by a new value. The time markers simply indicate when the next pixel update record has a different time from the one previously recorded.

Each pixel update record should be thought of as not only a current pixel value, but also an expected value for the future with implicit error bounds based on a noise model. A pixel update record is produced only when the pixel value is not within the error bounds for the value expected – and this is the sole form of compression used in constructing the TDCI version of the video. The particular noise model used here was an overly simple model that almost certainly underestimated the noise in all videos, allowing *maximum(5/128*expected, 5)* noise in values recorded with 8-bit precision. A more sophisticated noise model would recognize that noise is higher on higher-framerate samples, and would thus be expected to compress TDCI encodings at higher framerates much more aggressively.

Figure 3. *Canon PowerShot N 240FPS pink video*

Results

The first video tested was one shot to have action about as fast as would be likely to occur in human movement, but with very strong color variations to make the movement obvious and less compressible. The scene was a pink dragon puppet, shown in Figure 3, rapidly walked, in a wildly exaggerated motion, in front of a blank wall. This would be expected to compress very close to linearly as framerate is increased because there are fast-moving high-contrast edges. This scene was shot in normal room lighting at 240FPS using a Canon PowerShot N.

A bulky, but potentially useful way to understand why compression effectiveness might increase with framerate is the histogram of inter-frame pixel value differences given in Figure 4. In this figure, the pixel value in one frame determines the X coordinate and the corresponding pixel's relative value in the next frame determines the Y coordinate. The central gap represents pixel values that have not significantly changed according to the (here, very crude) noise model. Black pixels are from the 240FPS frames while red pixels are from those sampled at 24FPS. Clearly, the value spread is far less at higher framerates, so it would be natural to obtain greater compression.

TDCI compression plots

A more concise numerical summary is given by simply plotting the compressed file sizes for the exact same scene at various framerates. For the pink test, this is shown in Figure 5. The red

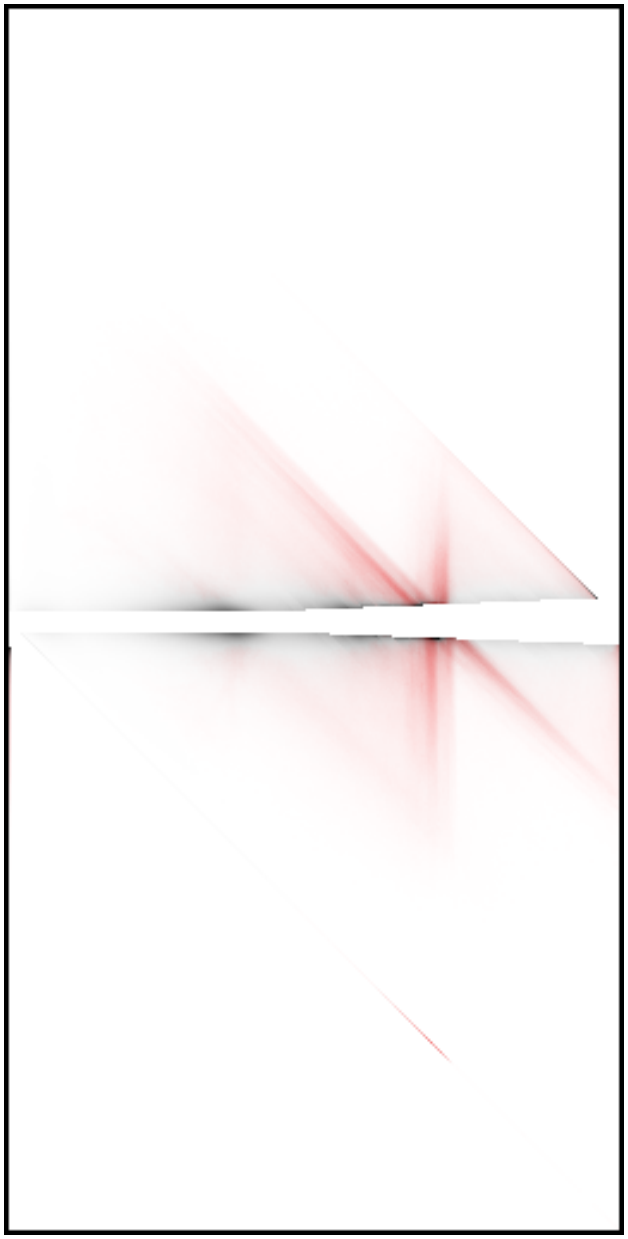

Figure 4. *Histogram of inter-frame pixel differences*

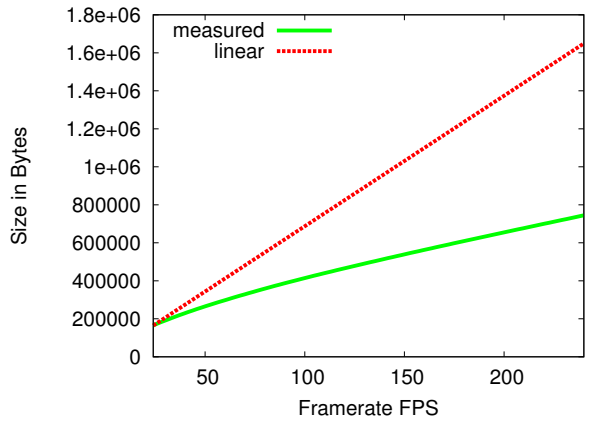

Figure 5. *TDCI compression of pink video*

Figure 6. *Canon PowerShot N 240FPS grass video*

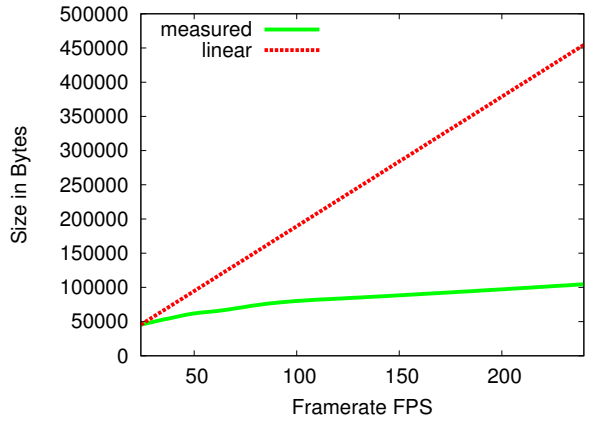

Figure 7. *TDCI compression of grass video*

line shows what would be a linear compression rate – compression effectiveness being independent of framerate. However, the green line showing the measured file sizes falls well below the red, and appears to be slowly converging toward a fixed maximum compressed size. In other words, this supports our thesis that an arbitrary framerate could be supported with finite bandwidth and storage capacity.

The second test scene, shown in Figure 6, also was captured at 240FPS using a Canon PowerShot N. However, it is a largely stationary outdoor scene, with only a modest fraction of the frame moving – the grass blades, which move *very* quickly on that windy day. As can be seen in Figure 7, the fact that so little is moving outweighs other factors and the compressed file size converges much more quickly than for the pink video. A high framerate is needed to eliminate the motion of the grass from one frame to the next, but 240FPS appears close to sufficient.

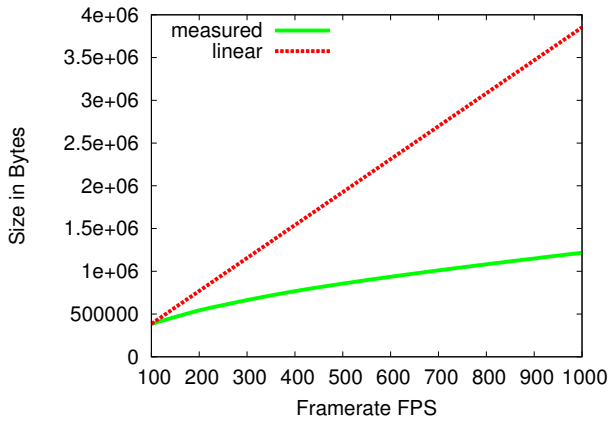

Figure 8. *TDCI compression of baseball video*

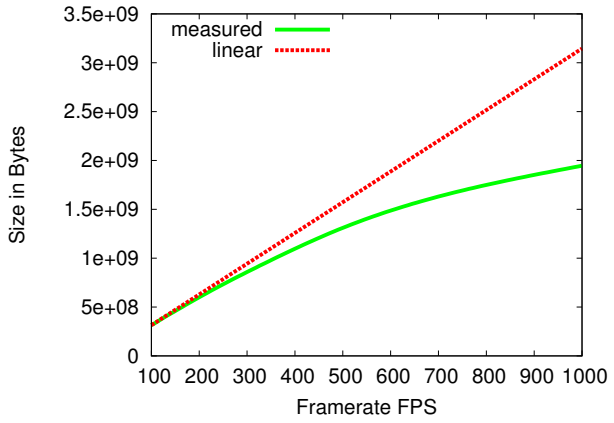

Figure 9. *TDCI compression of circus video*

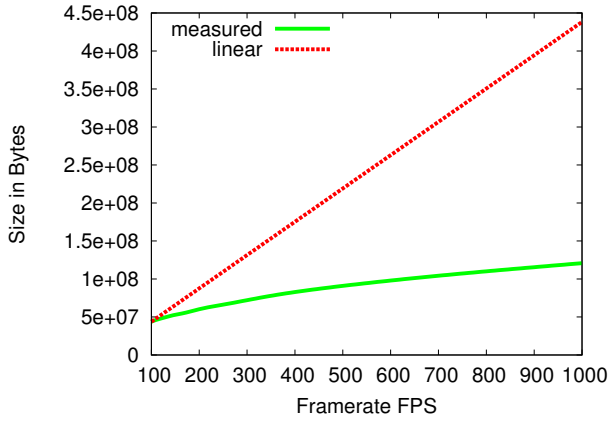

Figure 10. *TDCI compression of jump video*

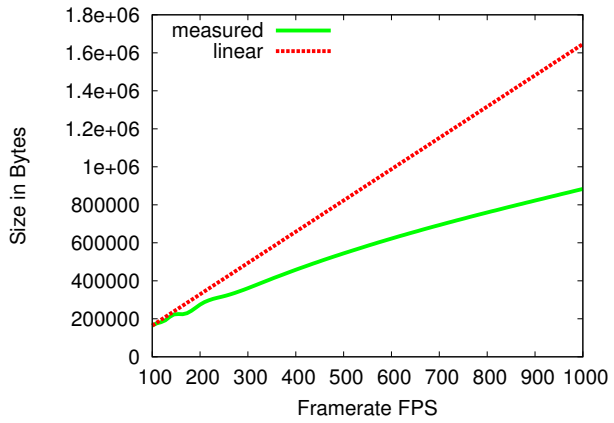

Figure 11. *TDCI compression of skateboard video*

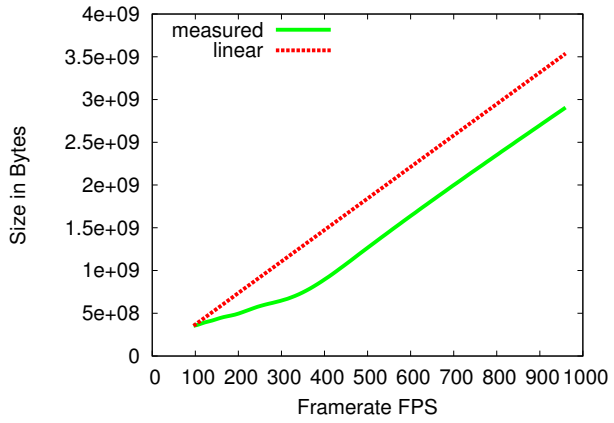

Figure 12. *TDCI compression of turkey video*

Figure 13. *Sony DSC-RX100 IV 960FPS turkey video*

The remainder of the experiments reported here were conducted at a maximum framerate of approximately 1000FPS in bright daylight.

The next four videos are all sports videos found on the internet. Figure 8 is a video of a baseball player at bat, which is a scene with a small amount of very fast motion. Figure 9 is a circus scene with a huge diversity of fairly rapid motions occurring. Figure 10 follows a shockingly athletic basketball jump shot, which involves both fast-moving players and panning of the camera. (Note that the compression used here does not do any type of motion prediction, so panning is actually a very tough case to compress.) Figure 11 shows a person being pulled through gentle waves while standing on his board. Although the rate of conver-

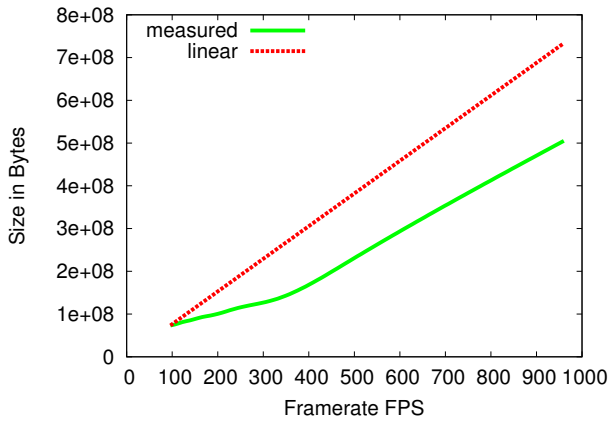

Figure 14. *TDCI compression of turkey at native 1136x384*

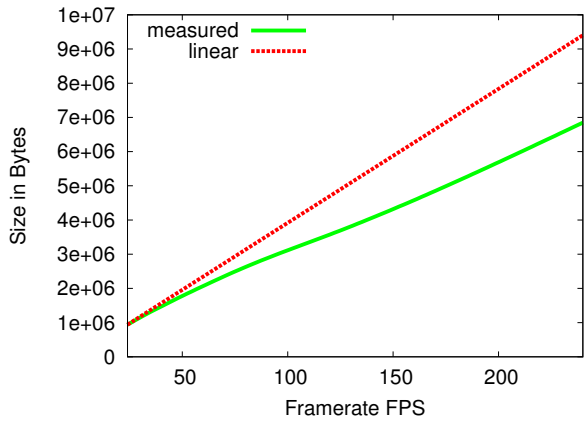

Figure 16. *H.264 compression of pink video*

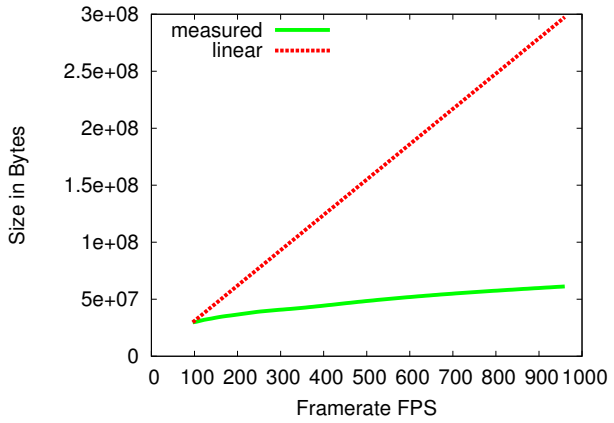

Figure 15. *Turkey video at 1136x384 with higher noise model*

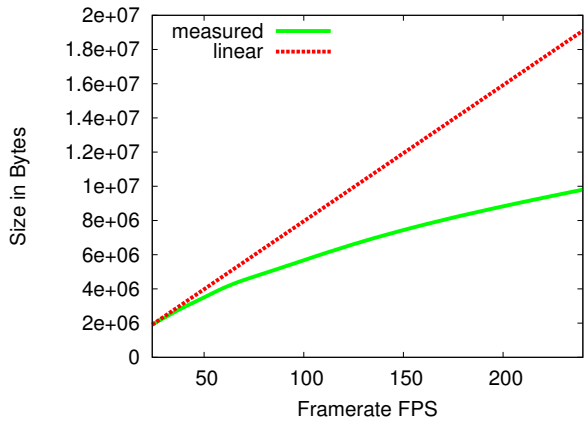

Figure 17. *H.264 compression of grass video*

gence to a constant size varies significantly between these videos, all show the expected characteristic improvement in compression ratio as framerate increases.

However, the last test does not fit this pattern. Figure 12 takes a disturbingly linear path after about 320FPS. Was this a particularly fast-moving scene? No. It was a turkey pretty much standing motionless, as shown in Figure 13. This last video was shot using a Sony DSC-RX100 IV at 960FPS... so perhaps it is a quirk of that camera? In fact, a quirk is involved: this camera records high-framerate video upscaled to 2K HD resolution, but the capture is really performed with 1136x384 pixels. Rescaling the 2K HD frames to the native 1136x384 yields the less-surprising compression curve shown in Figure 14. The shape of this curve is still somewhat unexpected, but it is likely a matter of not modeling noise accurately enough. Figure 15 confirms that compression with a better noise model (twice the noise level) does indeed bring the curve to the expected shape.

H.264 compression plots

As a final test, compression at various framerates was tested using standard H.264 Advanced Video Coding[7]. This compression scheme, widely used with Blu-ray Discs, various streaming internet video sources, and HDTV broadcasts, is much more sophisticated than the purely temporal TDCI compression. It is block-oriented and takes full advantage of motion compensation. This use of spatial information in compressing makes the relationship to photon shot noise information limits for individual pixels much less clear than it is for TDCI, so that it was not clear the same curve shapes would be seen for video compression at various framerates.

Although the resulting file sizes differ significantly from using TDCI compression, H.264 compression generally resulted in similar curve shapes for all the test high-framerate video sequences. For example, Figures 16 and 17 clearly show an increase in compression ratio as framerate is increased. It is not surprising that in many cases the H.264 encoding is more sensitive to the higher noise levels seen in individual frames captured at high framerates than TDCI encoding is; H.264 compression does not incorporate an explicit model for noise associated with high-framerate capture.

The terminal compressed sizes for TDCI and H.264 appear to often be quite different. However, the point of this paper is not to determine a particular ultimate limit on total information content in a video as framerate approaches infinity – the point is that

such a limit exists as a practical reality. For ordinary scenes using a well-crafted noise model, that limit might often be effectively reached at a framerate of less than 1000FPS.

Conclusion

This paper has attempted to characterize how much additional scene appearance data must be stored when framerate is dramatically increased. The study primarily targets framerates of approximately 1000FPS being used not for slow-motion capture, but for capturing time-varying models of ordinary scenes that can be used to computationally derive still images and videos rendered at arbitrary (lower) framerates.

Deriving lower-framerate sequences by simple stacking of frames from a higher-framerate sequence allows direct comparisons based on size of the data resulting from Time Domain Continuous Imaging (TDCI) encoding. In all cases, the data volume using a higher framerate was significantly less per frame than using a lower framerate. In all but one case, not only was the volume of data lower, but the curve describing the total volume of data appeared to be approaching a finite limit, suggesting that arbitrarily high framerates could be recorded with finite data. The outlying case revealed a similar curve after adjusting the noise model – quality of the noise model is critical.

What framerate is needed to ensure that no additional significant scene appearance information (as opposed to information about photons) can be obtained by increasing the framerate? The answer might often be under 1000FPS, but is highly dependent on the scene, lighting conditions, etc. However, with appropriate compression and a good model of inherent noise, the additional bandwidth and storage required for high framerates is not prohibitive. This justifies use of high-framerate capture not as a way to record unusually fast phenomena with specialized lighting, but as a practical way to allow virtual exposure intervals to be selected after capture. In other words, high-framerate video is a viable method by which TDCI can be approximated using conventional sensors.

Acknowledgments

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

References

[1] Digital Photography Review, Camera feature search, *http://www.DPReview.com/* (2016).

[2] Hi-Speed Cameras Affordable HD Slow Motion Video, HSC Camera Guide, *http://www.hispeedcams.comDPReview.com/* (2016).

[3] Casio, Casio Digital Camera EX-F1 User's Guide, *http://support.casio.com/storage/en/manual/pdf/EN/001/EXF1_MF_FD_EN.pdf* (2008).

[4] Henry Gordon Dietz, Frameless, time domain continuous image capture, *Proc. SPIE 9022, Image Sensors and Imaging Systems 2014, 902207 (March 4, 2014); doi:10.1117/12.2040016.* (2014).

[5] Henry Gordon Dietz, Frameless representation and manipulation of image data, *Proc. SPIE 9410, Visual Information Processing and Communication VI, 94100R (March 4, 2015); doi:10.1117/12.2083468.* (2015).

[6] Keith Wiley and Steve Chambers, Long-Exposure Webcams and Image Stacking Techniques *Digital Astrophotography: The State of the Art, Patrick Moorś Practical Astronomy Series*, Springer, (2005).

[7] Iain E. Richardson, H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. *John Wiley & Sons*, (2004).

Author Biography

Henry (Hank) Dietz is a Professor in the Electrical and Computer Engineering Department of the University of Kentucky. He and the student co-authors of this paper, Zachary Snyder, John Fike, and Pablo Quevedo, have been working to make Time Domain Continuous Image capture and processing practical. See Aggregate.Org for more information about their research on TDCI and a wide range of computer engineering topics.