Real-Time Algorithm Enabling High Dynamic Range Imaging and High Frame Rate Exploitation for Custom CMOS Image Sensor System Implemented by FPGA with Co-Processor

Blake C. Jacquot*, Nathan Johnson-Williams
The Aerospace Corporation, 2310 E. El Segundo Blvd., El Segundo, CA 90245-4609

ABSTRACT

We present results from a prototype CMOS camera system implementing a multiple sampled pixel level algorithm ("Last Sample Before Saturation") in real-time to create High-Dynamic Range (HDR) images that approach the dynamic range of CCDs. The system is built around a commercial 1280 × 1024 CMOS image sensor with 10-bits per pixel and up to 500 Hz full frame rate with higher frame rates available through windowing. We provide details of system architecture and present images collected with the system.

Keywords: real-time algorithm, CMOS image sensor, wide dynamic range, day/night vision, FPGA

1. INTRODUCTION

1.1 Motivation

Prior generations of space systems typically used CCDs for visible imaging, but CMOS imagers are becoming the dominant visible image sensor for space-based applications as CCD industrial base declines. Slow CMOS adoption, relative to other industries, is driven largely by CCDs having characteristics that fit well with space surveillance needs. Such capabilities include time delay and integrate (TDI), low fixed pattern noise (FPN), and high dynamic range. The generally lower dynamic range of CMOS imagers can be partly addressed by techniques and algorithms for High/Wide Dynamic Range CMOS (HDR/WDR CMOS). While algorithmic HDR imaging helps bridge the gap between CCDs and CMOS imagers, it is either computationally expensive or requires high bandwidth data transmission.

Real-time algorithms help address computation and bandwidth issues, but implementing real-time algorithms still requires significant computing demands. The system must capture, analyze, and store every pixel according to algorithm constraints without dropping frames.

1.2 Contributions

This paper documents implementation of a real-time multiple samples algorithm to extend dynamic range for a custom-built camera module with commercial CMOS image sensor. It expands on our prior work.[1] Though used in ground-based applications, this algorithm is new to space systems, which typically use well adjustment for dynamic range extension. The multiple samples technique (in model form) allows arbitrarily large dynamic range with appropriately scaled computational demands. The algorithm is implemented by FPGA to relieve burdens on the host processer.

These methods support applications such as motion tracking in day/night vision environments and aim to help close the gap between CCD and CMOS imagers. Additionally, they address sensor data generation rates and efficiently using available bandwidth for satellite imaging.

1.3 Related work

There are roughly eight commonly-used ways to enable HDR in CMOS image sensors:

Dual column level amplifier, Logarithmic pixel, Multiple sampled pixel, Pixel level Sigma Delta ADC with residue readout, Time to saturation with residue readout, Multiple wells, Well adjustment by adjusting reset signal, and Spatially varying exposure.[2–9]

* Blake.C.Jacquot@aero.org, blakejacquot@gmail.com; phone 1 (310) 336-1510; http://www.aerospace.org/

Real-Time Image and Video Processing 2015, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9400, 940004 · © 2015 SPIE
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2077727

An early use of a multiple sample scheme used two column parallel readouts to enable a short and long exposure.[10] Another similar implementation used dual capture[11] where a user stitched two images together to create an HDR image. This approach is not easily scalable to more than two samples due to area constraints. Other approaches used pixel-level analog-to-digital converters (ADC) to enable an arbitrary number of samples.[12] Later work applied multiple samples to reduce read noise at low illumination by "averaging" the captured images to reduce read noise.[13] An integrated "system-on-chip" approach enables multiple samples by including a microcontroller.[14] There is also more recent work using the multiple samples approach.[15–20]

1.4 Limitations and benefits

The multiple samples method implemented in real time by FPGA can require substantial computational complexity, memory, and power. Additionally, FPGA configuration and debugging takes significantly more time than constructing the equivalent operations in post-processing. Another limitation of real-time processing is that much of the data is thrown away after constructing the HDR image, which eliminates the user's ability to examine individual frames. Finally, the unit cell used for multiple samples must have adequate anti-blooming or else surrounding pixels can be contaminated from charge spill over.

However, despite these limitations, real-time HDR has significant benefits for bandwidth constrained environments. Data that may need to be thrown away due to bandwidth restrictions can be retained in some form. Information from these discarded images is included in the HDR frame. The user may decide that having some information from the sub-sampled images may be better than having none.

2. METHOD

2.1 Multiple samples approach

The multiple samples algorithm records each pixel at higher rates than intended for the final HDR image and then processes data according to algorithm constraints. Each pixel value is recorded at evenly spaced intervals in time. These pixel values are analyzed at the end of the frame cycle to determine if it has reached full well and is saturated or not. If not saturated, the last recorded value is stored for the HDR image if not saturated. However, if saturated, the algorithm backtracks along the pixel samples until it finds the last sample before saturation (LSBS). The algorithm extrapolates based on a linear fit to what value the pixel would have if it did not reach full well at frame end with this LSBS sample and record of the origin,. This extrapolated value is stored for the HDR frame. After the frame time, all pixels are reset and the process repeats. A conceptual representation is provided in Figure 1.

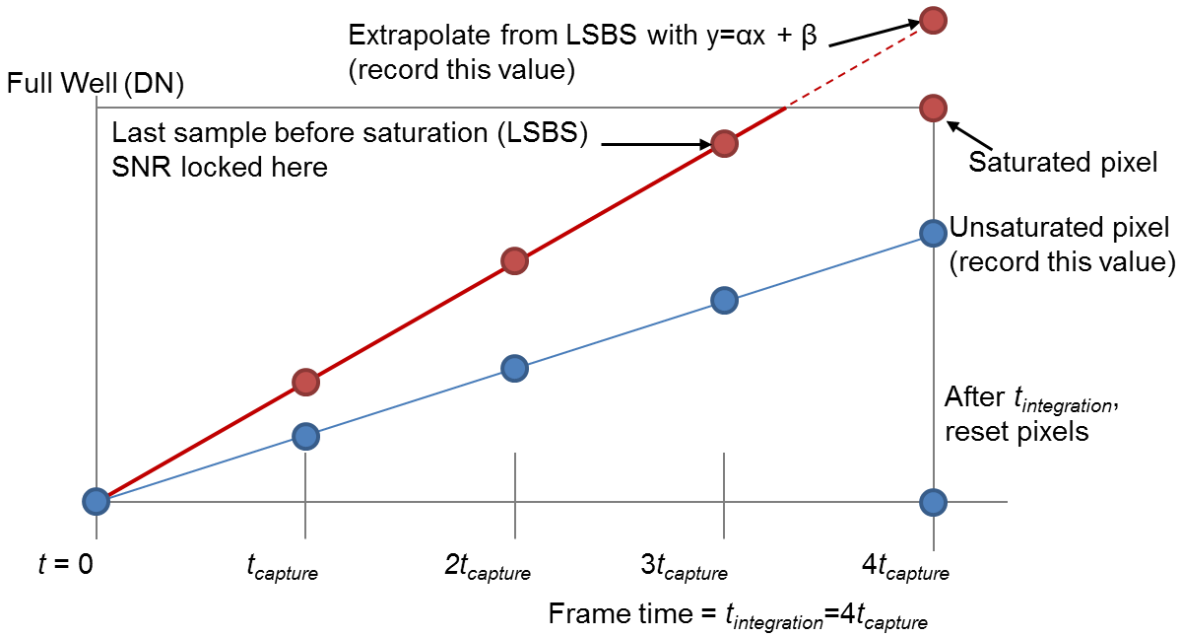

Figure 1. Conceptual representation of multiple sampled pixel method (also known as last sample before saturation, LSBS).

3. IMPLEMENTATION

3.1 Image sensor and camera

The CMOS image sensor chosen for this demonstration is the SXGA (1280 × 1024) LUPA1300-2 from ON Semiconductor in 168-Pin µPGA Package[21]. It accommodates 500 frames per second (fps) at full frame readout, but can reach 102,249 fps with extreme windowing. We used a 325 × 600 window at 1228.4 fps. The sensor has non-destructive readout and a global shutter. Internal requirements stipulated relatively large pixel pitch and bit depth of at least 10 bits. The LUPA1300-2 is one of the few sensors meeting all requirements and also available in small purchase quantities for prototyping. Key settings and specifications for the sensor are shown in Table 1.

Table 1. Key image sensor settings and specifications.

Image Sensor Settings and Specifications

User-Selected		Fixed	
Window width	1280 pixels	Pixel pitch	14 µm
Window height	325 pixels	ADC bit depth	10 bits
Frame rate	1307 Hz	Pixel rate	630 Mbps per channel
$t_{capture}$	765 µsec	Output	Monochrome
Shutter	Global	Pixel architecture	6T
Video channels used	12	CDS	On-chip
Synch channel used	1		
Non-destructive readout	Enabled		
FPN correction	Enabled		

The camera board is shown in Figure 2. The board integrates the microcontroller, image sensor module, mode selection switches, and LVDS input/output pairs.

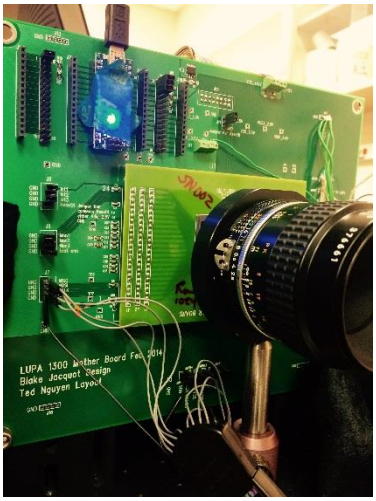

Figure 2. Camera board used for data collection.

3.2 Sensor programming

A microcontroller drove the sensor's input serial peripheral interface (SPI) for programming. The microcontroller was directed via software interface from a host PC. A level shifter converted the 5 V microcontroller logic levels to the 3.3 V levels required by the sensor.

3.3 Data acquisition

The image sensor transmits data over 13 LVDS lines (12 data, 1 synch) at 630 Mbps per line (double data rate). The data is first processed by a deserializer module in the FPGA, which deserializes by a factor of 10. The deserialized data is fed to a synchronizer which looks for known patterns in the image sensor's synch data (e.g. frame start 10-bit signal) and

applies a bit offset to adjust data to an expected configuration. The synchronized data either enters or bypasses the real-time module, depending on user selection. Despite whether the real time module is used, all data are held in a 1 GByte circular buffer within the Xilinx KC705 development board's DDR prior to transmission over a USB 2.0 interface to the host computer. A block diagram of this design is shown in Figure 3.

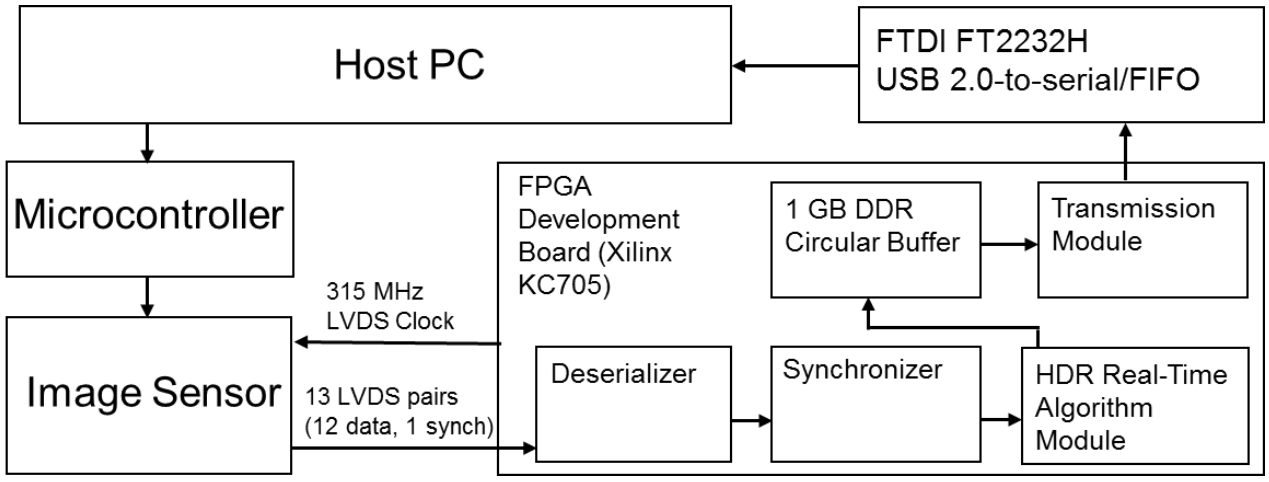

Figure 3. Block diagram of data acquisition system.

3.4 Real-time algorithm module

The real time algorithm is a multistate pipelined architecture. It is configured to process pixels from 12 channels at a time in parallel and pipelined so that it processes a new set of 12 pixels every clock cycle. A state machine for the real-time algorithm is shown in Figure 4.

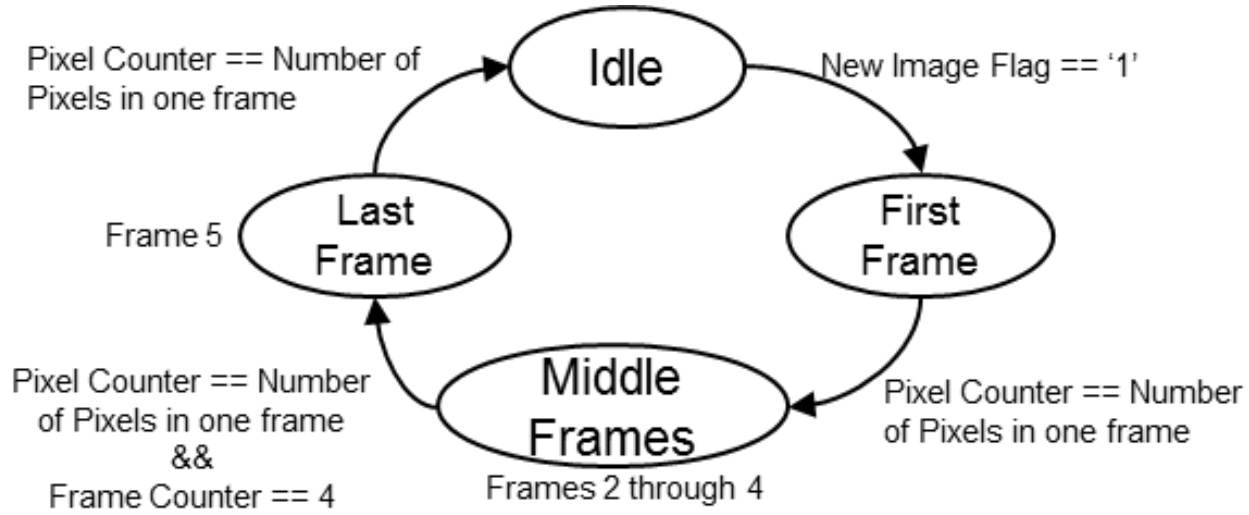

Figure 4. Block diagram of real-time algorithm module.

Each pixel of the first frame is stored in a memory location corresponding to its geographic location on the image sensor and tagged as a saturated/unsaturated pixel based on a programmable threshold. Earlier frames entering the system have shorter integration time than later frames. For subsequent frames, each pixel is first checked for saturation and compared to the corresponding stored pixel. If a pixel is newly saturated, the prior value of the pixel is maintained in memory and the frame number where it was first saturated is updated in that same memory address. If the pixel is unsaturated, the value in memory is updated and tagged as unsaturated. For pixels that are saturated in the first frame (always saturated) the value from frame 1 is stored. On the last frame, unsaturated pixels are output without extrapolation and saturated pixels use

extrapolated values based on the last reading before saturation and the frame where that pixel was first saturated. More details on the algorithm are shown in Figure 5.

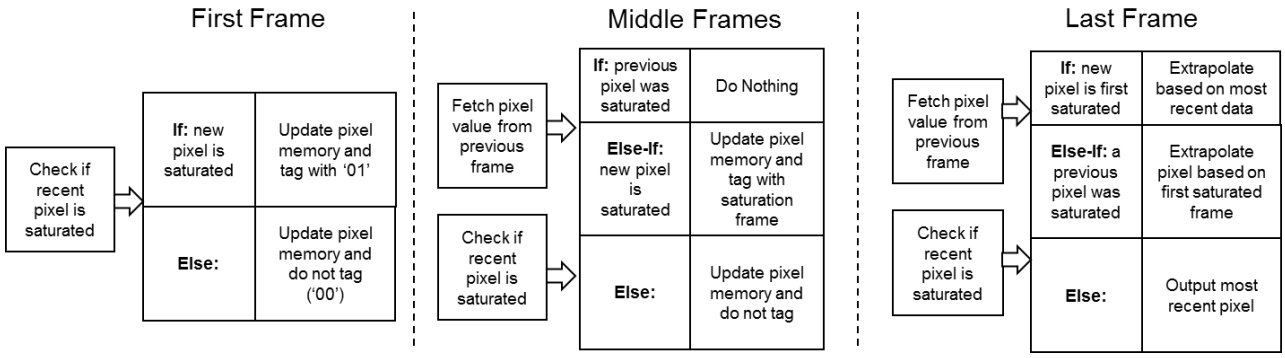

Figure 5. Pixel processing method.

Pixel extrapolation is performed conceptually by first dividing the stored pixel by the frame number where it first saturated and then multiplying by the total number of frames used to construct the HDR image (5 in our case). However, since FPGAs are not ideally suited for division, we approach the problem differently. To make the calculation easier to implement on a pipelined FPGA, a single multiplication is used based on pre-calculated values (5/1, 5/2, 5/3, 5/4 or 5/5). In order to use fixed point math, all values are multiplied by 12 (becoming 60/12, 30/12, 20/12, 15/12, or 12/12). The denominator is dropped, essentially making units "1/12ths" but allows the output to be an integer. To maintain consistency, unsaturated values are multiplied by 12 (since the units are 1/12ths, it is equivalent to multiplying by 1). Essentially, the LSBS pixel value is multiplied by an integer (e.g. 60, 30, 20, 15, or 12). The output from the real-time algorithm is 16 bits per pixel and the value is never rounded.

The real time algorithm uses a 13 bit word in memory. The most significant 3 bits are used to store the frame number in which the pixel was first saturated (e.g. "001" indicates first frame saturation, "002" indicates second frame saturation, "000" indicates the pixel has never saturated). The remaining 10 bits are used to store the pixel value (either the last value before saturation or current pixel value as appropriate. The address scheme reflects the order in which the pixels are read out: address 0 is the first pixel, address 1 is the second pixel, and so on. The order in which pixels are read out is consistent. The memory scheme for the real-time algorithm is shown in Figure 6.

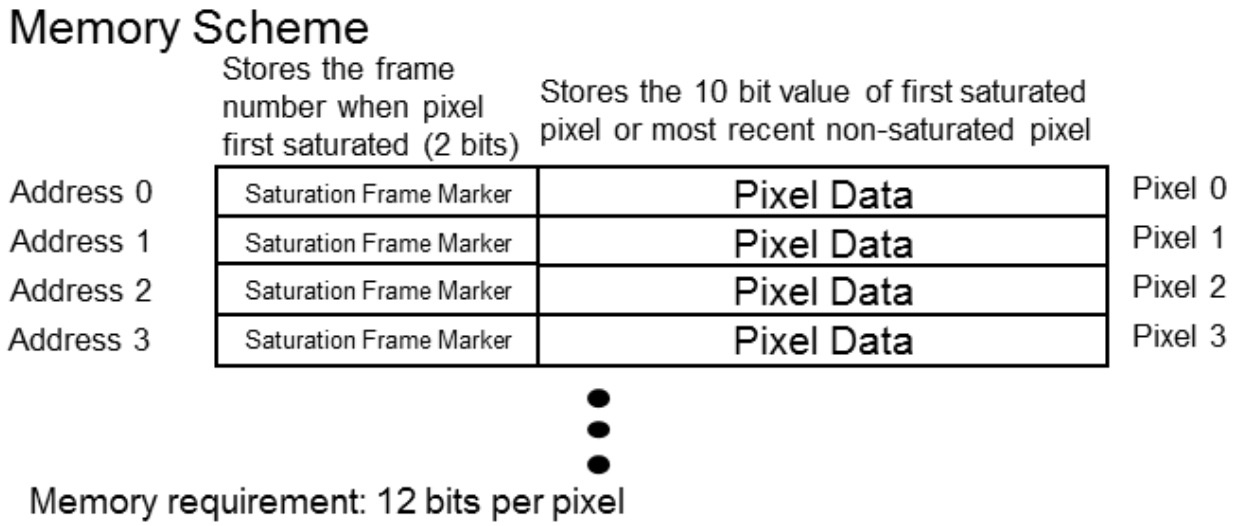

Figure 6. Memory scheme for real-time algorithm.

4. RESULTS

4.1 Validating real-time processing against post processing for HDR image

To aid in reducing the time to optimize the algorithm and gain visibility into sub-sampled images, we did most initial work in post-processing and applied best parameters to the real-time module in FPGA. Development with FPGA results in slower design cycles and is not an ideal platform for algorithm optimization. In contrast, post-processing with scripts on a desktop can result in faster design cycles. Additionally, post-processing allows visibility into the sub-sampled images that the real-time algorithm throws away after calculating the HDR image.

4.2 Sample number selection and pixel value with post-processing

We used 5 non-destructive reads in our implementation. Figure 7 shows images of which frame was selected by the Last Sample Before Saturation (LSBS) algorithm (e.g. frame 1, 2, 3, 4, or 5) and the resulting unsaturated pixel value at that frame. Both images show significant detail. The LSBS Sample image shows some pixels for which sample 0 was chosen, indicating it was saturated in the first frame. Since all samples are saturated, these pixels are saturated in the final image. The LSBS value image shows that values selected for extrapolation are skewed toward the higher end of the distribution, consistent with expectations. The threshold value selected for saturation was 768 DN which roughly corresponds to bloomed full well measured on the chip.

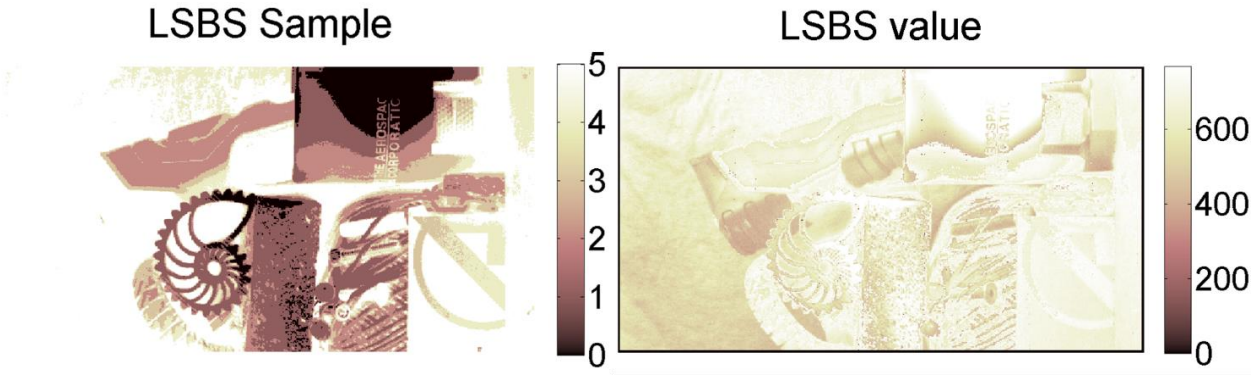

Figure 7. LSBS sample selection and pixel value at that sample.

Figure 8 shows histograms of the sample number and the associated pixel value for the images. As can be seen, sample 5 is particularly represented due to the dark nature of most of the scene, especially the background.

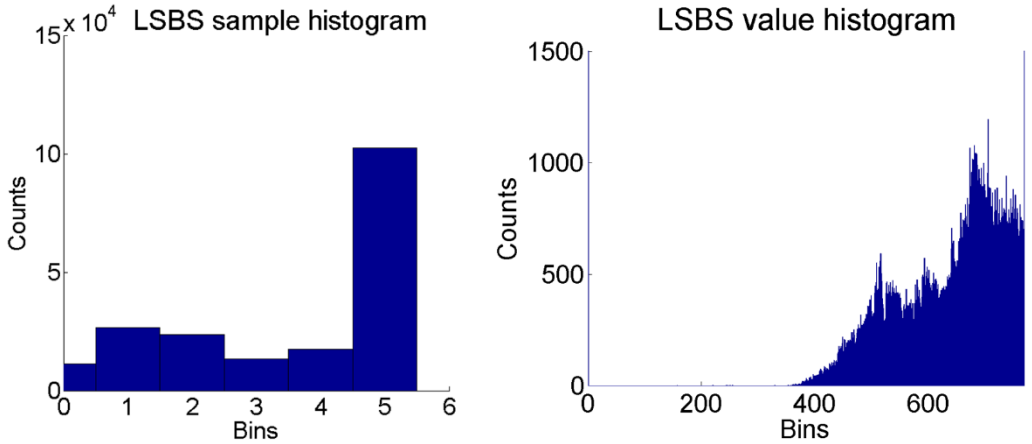

Figure 8. Histograms of sample number and associated pixel value.

4.3 Sampled images for constructing and HDR frame

Figure 9 shows all subsamples used to construct a single HDR frame

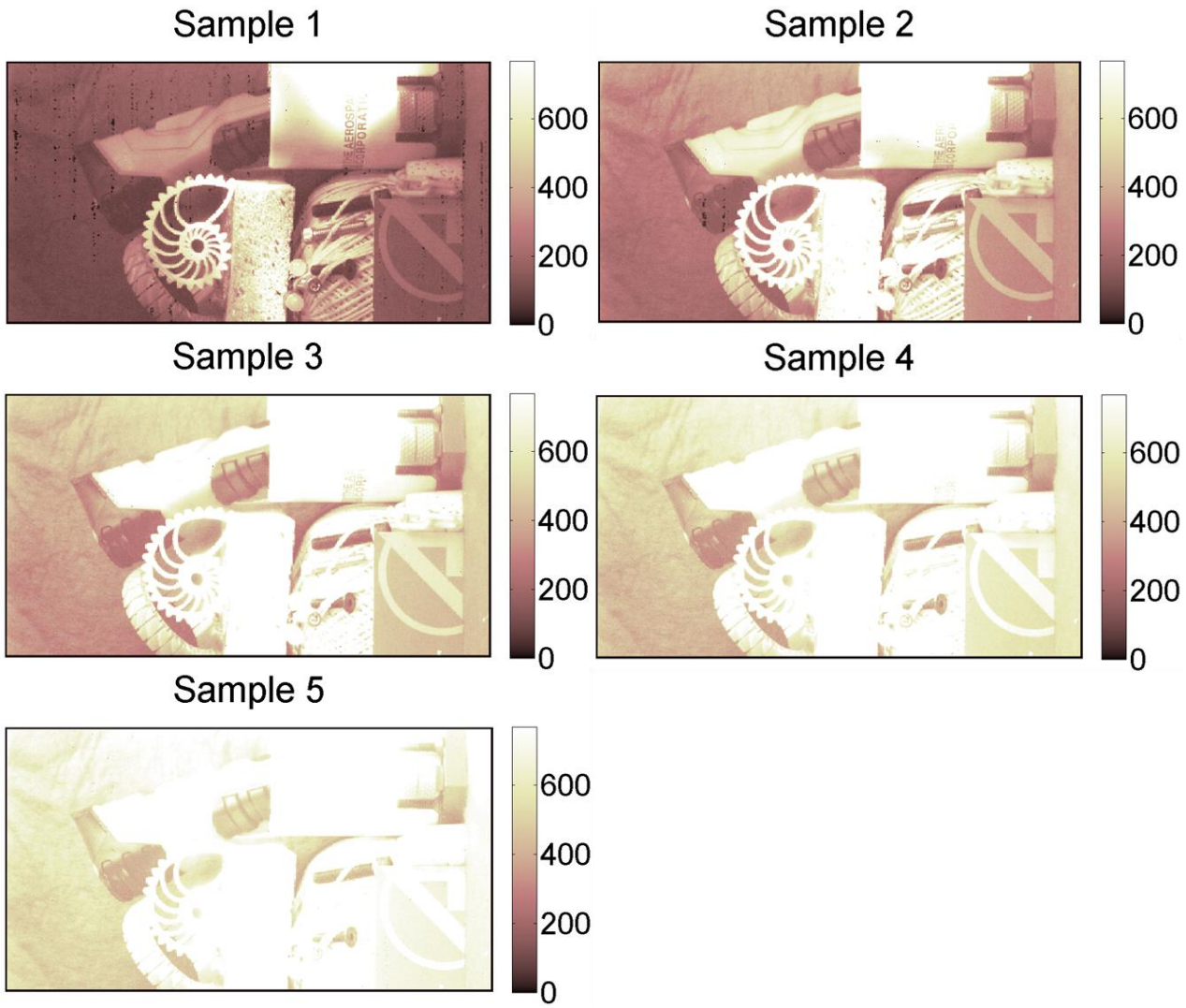

Figure 9. Subsampled images that feed HDR processing algorithm.

Figure 10 shows histograms of all sub-sampled images. As expected earlier samples have histograms skewed towards low DN values than later samples. Bloomed full well occurs around 800 DN and saturated full well around 900 DN.

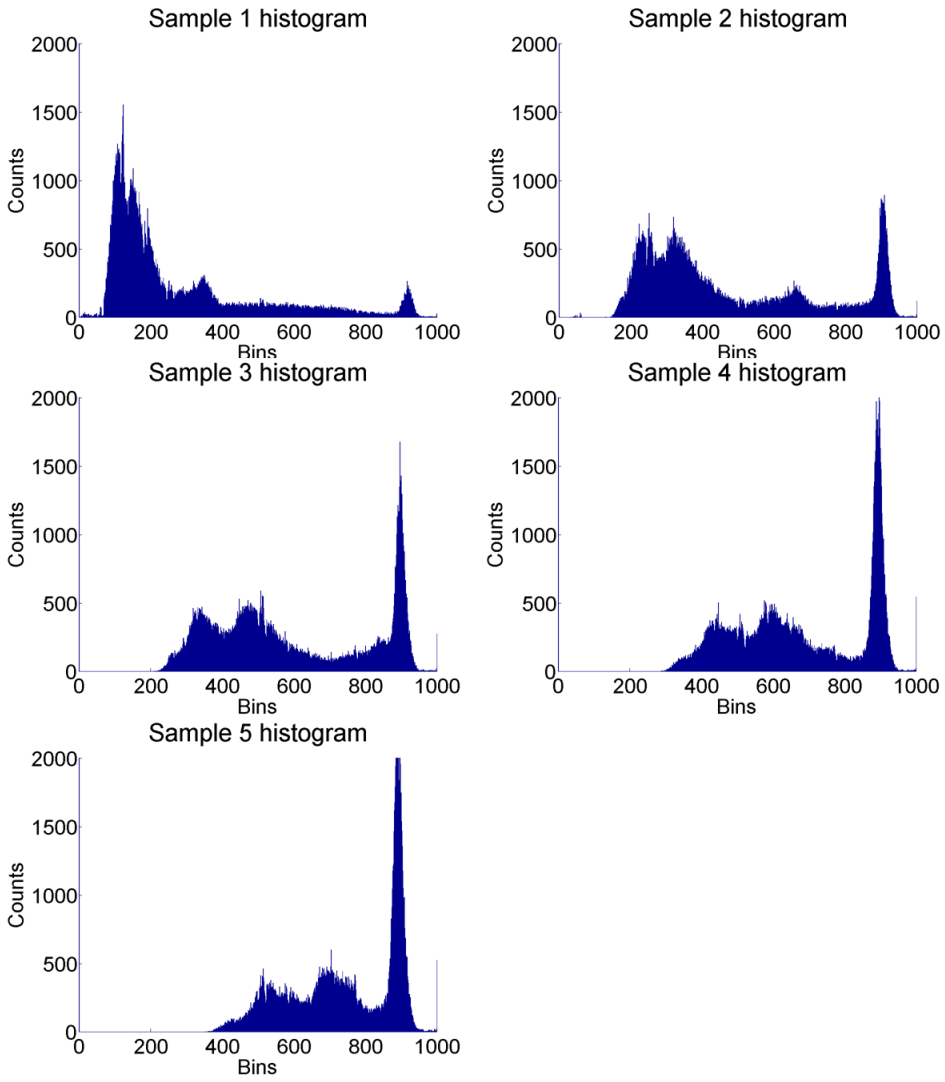

Figure 10. Histograms of sub-sampled frames.

4.4 HDR image from post processing

Figure 11 shows the HDR image constructed from the above sub-sampled frames using the LSBS algorithm. Detail is much greater for the HDR frame than for any individual sub-sampled frame. In particular, background details of the dark cloth are easily visible simultaneously with letters on the coffee mug. None of the subsampled images contain both details.

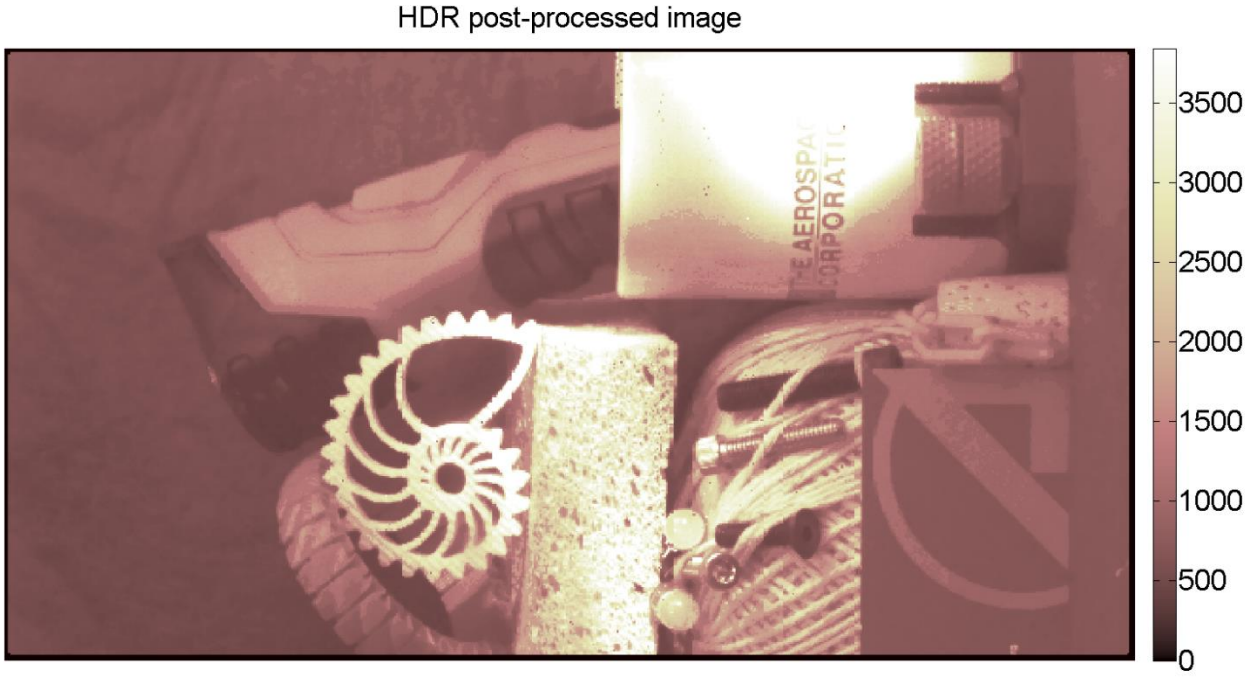

Figure 11. Resulting image with HDR algorithm applied during post-processing

Figure 12 shows the histogram for the post-processed HDR image.

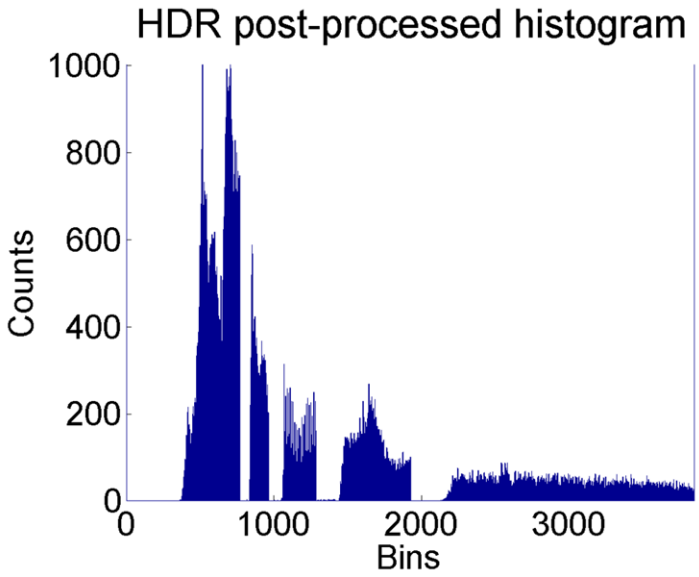

Figure 12. Histogram of post-processed HDR frame

4.5 HDR image from real-time processing

Figure 13 shows an HDR frame resulting from real-time processing. This image compares well with the post-processed image. Again, black background details are visible in the same image as letters on the coffee mug.

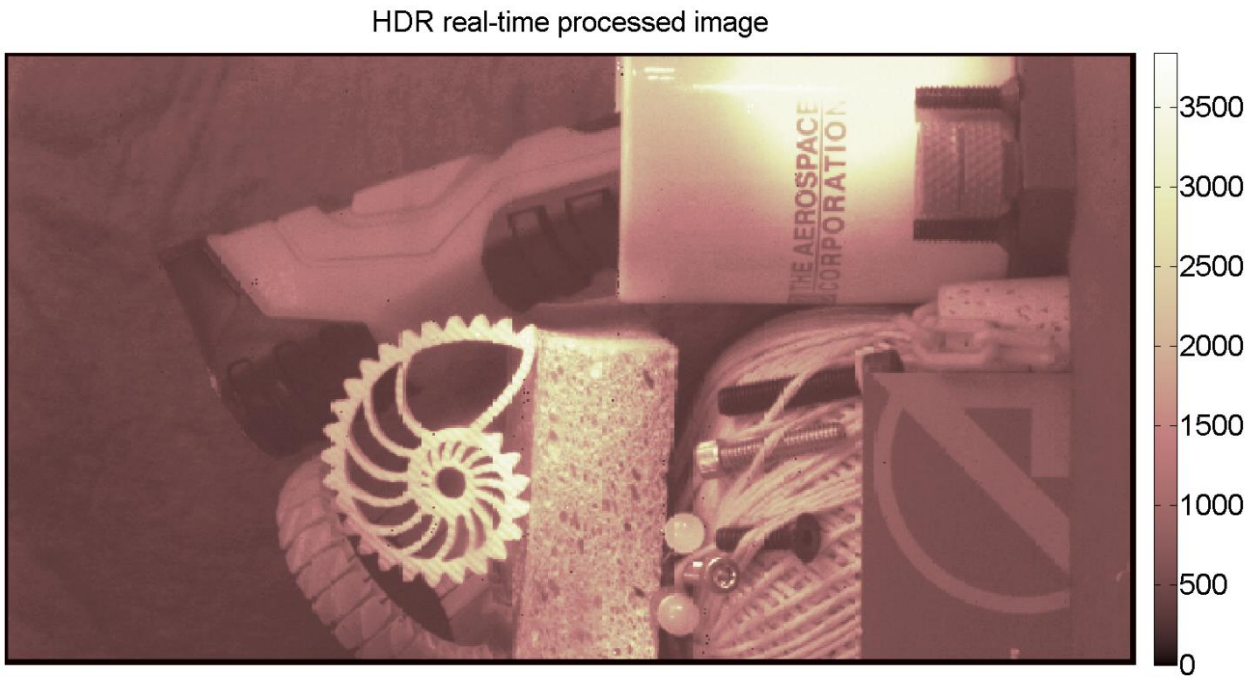

Figure 13. Resulting image with HDR processing applied in real-time.

Figure 14 shows the histogram of the real-time processed HDR image.

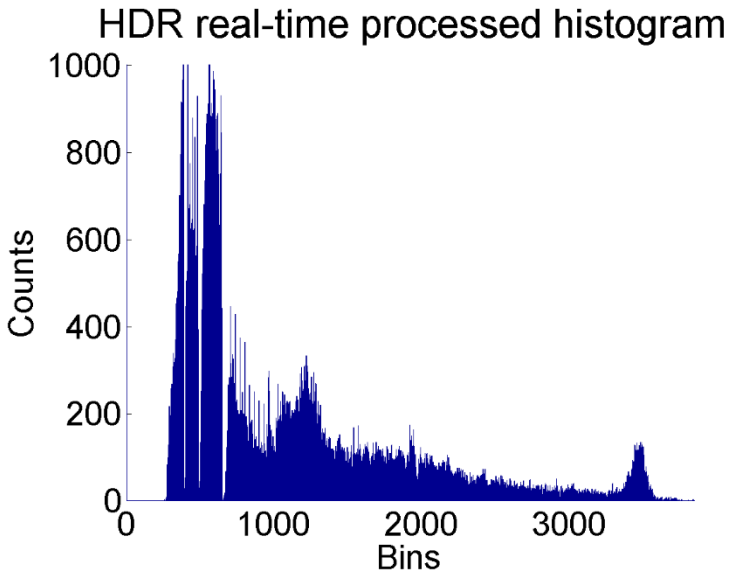

Figure 14. Histogram of real-time processed HDR image.

5. DISCUSSION AND ISSUES

5.1 Comparison between post-processed and real-time processed HDR frames

The resulting images from post-processing and real-time processing compare well against each other. The post-processing histogram has unexplained gaps relative to real-time processing. At time of writing, they are not understood.

5.2 Future directions

With real-time HDR method verified, we will implement additional real-time algorithms. In particular, algorithms exploiting high frame rates are particularly interesting for space applications with limited bandwidth. Additionally, we will consolidate our camera module and prepare for deployment in a small satellite.

6. CONCLUSION

Real-time processing of HDR images compared well against post-processing methods. Both methods show detail of bright and dim details that were either saturated or featureless in subsampled images. The LSBS method is particularly powerful because all samples are correlated, which reduces the impact of reset noise on final images.

ACKNOWLEDGMENT

We thank Eric Herman and Terry Lomheim. Funding was provided by The Aerospace Corporation through the Sustained Experimentation and Research for Program Applications (SERPA) and Internal Research and Development (IR&D) programs.

REFERENCES

[1] Jacquot, B. C., Johnson-Williams, N., "Multiple-samples-method enabling high dynamic range imaging for high frame rate CMOS image sensor by FPGA and co-processor," Proc SPIE 9216, 921609-1 (2014).
[2] Spivak, A., Belenky, A., Fish, A., Yadid-Pecht, O., "Wide-Dynamic-Range CMOS Image Sensors—Comparative Performance Analysis," IEEE Trans. Electron Devices **56**(11), 2446–2461 (2009).
[3] Kavusi, S., Kakavand, H., El Gamal, A., "Quantitative study of high dynamic range sigma-delta-based focal plane array architectures," Def. Secur., 341–350, International Society for Optics and Photonics (2004).
[4] Fowler, B., "High dynamic range image sensor architectures," IS&T/SPIE Electron. Imaging, 787602 (2011).
[5] El Gamal, A., Eltoukhy, H., "CMOS image sensors," IEEE Circuits Devices Mag. **21**(3), 6–20 (2005).
[6] Yang, D. X. D., El Gamal, A., "Comparative analysis of SNR for image sensors with enhanced dynamic range," Electron. Imaging, 197–211, International Society for Optics and Photonics (1999).
[7] Darmont, A., [High Dynamic Range Imaging: Sensors and Architectures], SPIE (2012).
[8] Kavusi, S., "Quantitative study of high-dynamic-range image sensor architectures," Proc. SPIE **5301**, 264–275, SPIE (2004).
[9] Posch, C., Matolin, D., Wohlgenannt, R., "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS," IEEE J. Solid-State Circuits **46**(1), 259–275 (2011).
[10] Yadid-Pecht, O., Fossum, E. R., "Wide intrascene dynamic range CMOS APS using dual sampling," IEEE Trans. Electron Devices **44**(10), 1721–1723 (1997).
[11] Nakamura, T., Saitoh, K., "Recent progress of CMD imaging," IEEE Work. CCDs Adv. Image Sensors, Bruges, Belgium (1997).
[12] Yang, D. X. D., Gamal, A. E., Fowler, B., Tian, H., "A 640×512 CMOS image sensor with ultrawide dynamic range floating-point pixel-level ADC," IEEE J. Solid-State Circuits **34**(12), 1821–1834 (1999).
[13] Liu, X., El Gamal, A., "Synthesis of high dynamic range motion blur free image from multiple captures," Circuits Syst. I Fundam. Theory Appl. IEEE Trans. **50**(4), 530–539, IEEE (2003).
[14] Bidermann, W., El Gamal, A., Ewedemi, S., Reyneri, J., Tian, H., Wile, D., Yang, D., "A 0.18 μm high dynamic range NTSC/PAL imaging system-on-chip with embedded DRAM frame buffer," IEEE Int. Solid-State Circuits Conf.. Dig. Tech. Pap. ISSCC. **1**, 212–488, IEEE (2003).
[15] Sasaki, M., Mase, M., Kawahito, S., Tadokoro, Y., "A Wide-Dynamic-Range CMOS Image Sensor Based on Multiple Short Exposure-Time Readout With Multiple-Resolution Column-Parallel ADC," IEEE Sens. J. **7**(1), 151–158 (2007).
[16] Bilcu, R. C., Burian, A., Knuutila, A., Vehvilainen, M., "High dynamic range imaging on mobile devices," IEEE Int. Conf. Electron. Circuits Syst., 1312–1315, IEEE (2008).
[17] Shang, Y., Guan, Y., Zhang, W., Liu, X., Zhang, S., Zhang, Y., "A high dynamic range complementary metal-oxide-semiconductor (CMOS) camera using multi-slope response and an image reconstruction algorithm," Meas. Sci. Technol. **20**(10), 104005 (2009).
[18] Hasinoff, S. W., Durand, F., Freeman, W. T., "Noise-optimal capture for high dynamic range photography," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 553–560, IEEE (2010).

[19] Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S. K., "Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum.," IEEE Trans. Image Process. **19**(9), 2241–2253 (2010).
[20] Yamada, T., Kasuga, S., Murata, T., Kato, Y., "A 140dB-Dynamic-Range MOS Image Sensor with In-Pixel Multiple-Exposure Synthesis," IEEE Solid-State Circuits Conf. (ISSCC), 50 (2008).
[21] "LUPA 1300.", <http://www.onsemi.com/pub_link/Collateral/NOIL2SM1300A-D.PDF>.