https://doi.org/10.2352/ISSN.2470-1173.2019.4.PMII-589

Fast Restoring of High Dynamic Range Image Appearance for Multi-Partial Reset Sensor

Ziad Youssfi, Firas Hassan; Ohio Northern University; Ada, Ohio, USA

Abstract

Improving image quality by capturing high dynamic range (HDR) in a scene remains a technological challenge for CMOS sensors. The multi partial reset is a simple technique that allows a sensor to capture HDR inexpensively with high frame rate, low noise floor, and high signal to noise ratio. However, it flattens images due to compression of highlights, rending it useful only for machine vision and automotive applications.

In this work, we present an inverse function that restores HDR image appearance specifically for a multi partial reset sensor (MPRS). This function can be applied in software or firmware before demosaicing. Results show that the function automatically enliven images with more depth and saturation that suit general purpose photography. Moreover, latency results show that it can be applied for real-time videography of high frame rate. These results would be computationally much more expensive to achieve using general image enhancement techniques, i.e. not specific to MPRS, especially for high definition, high frame rate, real-time video.

Introduction

Dynamic range of a digital camera is one aspect, among many, that contributes to image quality; it is defined as the ratio of the luminance that just saturates the camera sensor to the luminance that registers just above its noise floor by one standard deviation [1]. Since the luminance in our real-world environment can vary by eight orders of magnitude, e.g. between starlight and sunlight, capturing this wide range presents a technological challenge.

In a normal CMOS active pixel sensor (APS), each pixel's photodiode converts light photons into electrons, which are collected into a diffusion capacitor during exposure or integration time [2, 3]. These electrons are converted into voltage, which is then converted into a digital value for each pixel at the end of exposure time. If exposure time for a scene is set to capture details in the shadows, and if the light intensity in the highlights for that same scene generates more electrons than a pixel diffusion capacitor can handle, the extra electrons for that pixel are drained and not converted into voltage; i.e. that pixel reaches saturation before the end of integration time, which translates into clipping or loss of details in the highlight in the final image. This saturation limits the sensor's dynamic range. For instance, in Figure 1-(a), lines 1 and 2 represent light intensities whose integration keep rising linearly until the end of integration time T; however, line 3 represents a pixel light intensity that reaches 100% saturation before the end of integration time.

Many techniques have been proposed to increase the dynamic range of CMOS image sensors [4]. The MPRS technique achieves HDR by dividing integration into multiple phases that have different saturation levels [5-7]. In our example sensor, we have three phases:

1. In phase 1, from 0 to T1, the maximum saturation level is set to SL1, which is lower than 100%, Figure 1-(b). If light integration reaches SL1 before T1, integration stops at SL1 until T1. Line 2 and 3 in Figure 1-(b) show this scenario, while the integration in line 1 is unaffected.

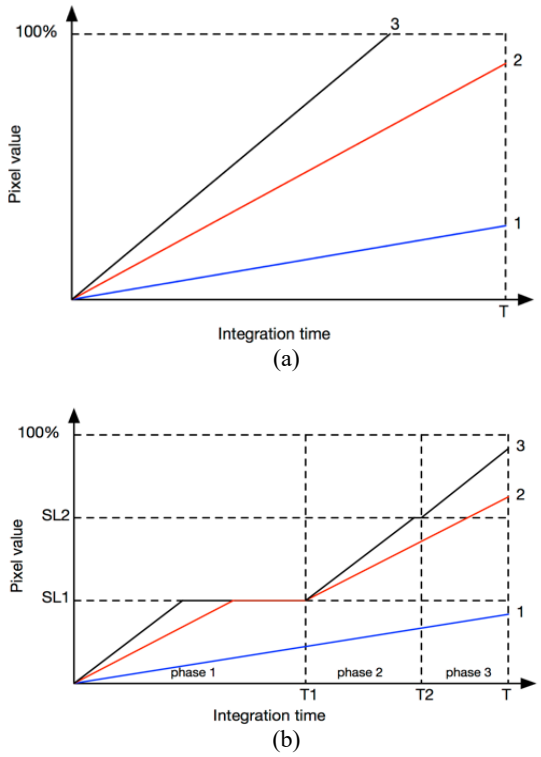


Figure 1. (a) Normal integration time. (b) Multi-partial reset integration time with three phases.

2. In phase 2, from T1 to T2, the maximum saturation level is moved up from SL1 to SL2, which is still lower than 100%. If integration reaches SL2 before T2, integration stops at SL2 until T2. Line 3 in Figure 1-(b) represents this scenario, while the integration in line 1 and 2 are unaffected.
3. In phase 3, from T2 to T, the maximum saturation level is moved to the default 100%. Integration in this phase stops only if it reaches 100% before time T.

This three-phase process of MPRS allows pixels to tolerate high luminance intensities without saturation. It also allows pixel values to be read and converted into digital values immediately at the of end of integration time. Since there is no processing overhead for HDR, MPRS lends itself to high frame rate video applications (e.g. 30 frames per second or higher), such as in high definition video. Moreover, due to its simplicity, it offers low noise floor and high signal to noise ratio [4].

The main disadvantage of MPRS is that highlight pixel values that would not saturate without MPRS end up closer to shadow range. This compression effect can be seen in Figure 1-b where the

pixel value for line 2 ends up closer to the pixel value of line 1 than in Figure 1-a. Visually, this compression "flattens" images for an unappealing feel. Although machine vision and automotive applications can still benefit from the HDR in MPRS, this compression presents a challenge for general purpose applications.

In this work, we present a simple inverse restoring function to adjust the HDR output of MPRS so that highlights regain their original distance from shadows. The function can be implemented in software before demosaicing. Results show that the function automatically enliven images with more depth and saturation that suit general purpose photography and videography. These results would be computationally much more expensive to achieve using general image enhancement techniques, i.e. not specific to MPRS, such as in [1], especially for high definition, high frame rate, real-time video.

The rest of the paper is organized as follows: Section II presents our inverse restoring function and its analysis; Section III presents our results and discussion of distance adjustment between highlights and shadows, as well as adjustment in color saturation. It also discusses latency result; and Section IV presents our conclusion.

MPRS Inverse Restoring Function

The inverse function restores MPRS pixels to their original values before compression using knowledge of the phase parameters. For simplicity, we assume that all input parameters $SL1$, $SL2$, T_1, and T_2 are given as ratios of one. We also normalize the integration time T to unity and the pixel value x from the ADC into $p \leq 1$, by dividing by 2^n, where n is the number of bits. Since the integration time is normalized, pixel values on the y-axis can be expressed as the slope of a line going through the origin. The inverse restoring function divides the range of the compressed pixel value p into shadow, midrange, and highlight regions based on the maximum line slopes for these regions.

The maximum slope for the shadows region that does not saturate during the integration time $t \leq T_1$ is the slope that touches SL1 at T_1. Hence, at the end of the integration time $T = 1$, the maximum value for the shadow region is $\frac{SL1 \times 1}{T_1}$. Any pixel value p below this value does not need restoring as the integration goes uninterrupted without any reset.

The maximum slope for the midrange region that does not saturate while $t \geq T_1$ is a slope that will start at SL1 and touches SL2 at T_2. Given those two points, we can calculate the maximum value p for the midrange region to be $\frac{SL2-SL1}{T_2-T_1} \times (1 - T_1) + SL1$. Hence, the midrange region is between pixel values $\frac{SL1}{T_1}$ and $\frac{SL2-SL1}{T_2-T_1} \times (1 - T_1) + SL1$. Any compressed pixel value p within this region needs to be expanded into $\hat{s} = \frac{p-SL1}{1-T_1}$.

Finally, the highlights region is for pixel value $p \geq \frac{SL2-SL1}{T_2-T_1} \times (1 - T_1) + SL1$. Any compressed pixel value within this region needs to be expanded into $\hat{s} = \frac{p-SL2}{1-T_2}$.

To renormalize the pixel value $\hat{s}$, we multiply it by 2^n and round to the nearest integer.

The inverse function algorithm expressed in pseudocode follows:

INVERSE-FUNCTION (x)

$p = \frac{x}{2^n}$

if $p \leq \frac{SL1}{T_1}$ **then**

$\hat{s} = p$

else if $p \leq \frac{SL2 - SL1}{T_2 - T_1} \times (1 - T_1) + SL1$ **then**

$\hat{s} = \frac{p - SL1}{1 - T_1}$

else

$\hat{s} = \frac{p - SL2}{1 - T_2}$

return $y = [2^n \times \hat{s}]$

For MPRS to work properly, the parameters should be selected carefully such that the slopes gradually increase from one region to the second. As mentioned before, the largest slopes in the three different regions are $\frac{SL1-0}{T_1-0}$, $\frac{SL2-SL1}{T_2-T_1}$, and $\frac{1-SL2}{1-T_2}$ representing the shadows, mid lights and high lights, respectively. The dynamic range of the image is defined by the ratio between the largest slope in the high lights region and the smallest slope in the shadow region. Hence, to increase the dynamic range, we should decrease the maximum slope of the shadows region as much as possible and increase the maximum slope of the highlights region as much as possible. To make the maximum slope of the shadows region as small as possible, we should decrease its numerator and increase its denominator. In this work, the chosen camera has limitation on its parameters. The smallest SL1 was 0.2 or 20% of total saturation level. The largest T_1 was 0.7 or 70% of the total integration time. To increase the maximum slope of the highlights region, we picked SL2 to be as small as possible, which was again limited to 0.4 or 40% of total saturation level. Hence, our choice was limited to T_2. Moving T_2 as close as possible to 1 will increase the dynamic range, but the trade-off is in the reduction of the mid lights region. A good compromise is to set T_2 to 0.9 or 90% of total integration time. These are the parameters that we used throughout this work. Also, the resolution of the ADC of the camera that we worked with is $n = 10\ bits$.

Based on the selected parameters, we can calculate the largest slope that does not saturate in the high lights region to be $s_{max} = \frac{1-0.4}{1-0.9} = 6$. The smallest non-zero slope is $s_{min} = p_{min} = \frac{1}{1024}$. Therefore, the dynamic range of the sensor using our parameters is $20log_{10}\left(\frac{s_{max}}{s_{min}}\right) = 75.77dB$. Notice that if the parameters are set to the following: $SL1 = \frac{1}{3}$, $SL2 = \frac{2}{3}$, $T_1 = \frac{1}{3}$, and $T_2 = \frac{2}{3}$ then the MPRS will act as a normal CMOS sensor where $s_{max} = 1$ and $s_{min} = \frac{1}{1024}$ and the dynamic range is 60.2 dB. Therefore, based on these parameters and after applying our proposed inverse restoring function, we achieved an restoration in the dynamic range of 15.57dB.

Implementation and Results

To acquire MPRS HDR images, we employed a camera from XIMEA Inc. (model xiQ MQ022CG-CM). This camera model captures images with up to 90 dB dynamic range, owing to its sensor made by CMOSIS Inc. The 2/3" color CMOS active pixel sensor has 2 MP with a 10-bit ADC output resolution. The XIMEA camera interfaces through USB 3.0 to a PC for data and control.

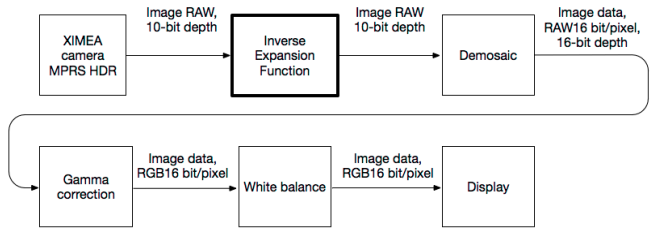


Figure 2. Software pipeline implementation

To facilitate our software pipeline implementation, we leveraged XIMEA APIs and OpenCV C++ libraries. Figure 2 shows our pipeline diagram.

To test the effect of the inverse function, we employed a calibration target used by photographers to help set exposure and white balance. The target consists of fabric with three regions of specific light reflectance: a black region representing shadows of near 5% reflectance, a middle grey region of about 18% reflectance, and white region of about 90-100% reflectance, as shown in Figure 3. Using images from this calibration target, we computed four histograms as shown in Figure 4. Each of the white, grey, and black regions in the calibration target show up as two humps in the histograms because the images are raw data from the color filter array. Because there are as many green pixels as red and blue ones, and since the green has higher luminance absorption, there is one hump for greens, and another for the reds plus blues combined.

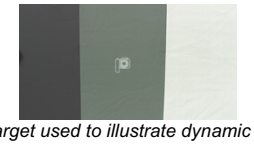


Figure 3. Calibration target used to illustrate dynamic range restoration for histogram data

It is important to note the compression effect by the MPRS HDR on the distance between the highlights and midrange regions between histogram (a) and (b) in Figure 4. This compression is responsible for "flattening" the image. However, when applying the inverse function, the distance between highlights and midrange is restored in histogram (c) in Figure 4. This gives more latitude for a later stage of gamma correction or compression to move up the midrange, as is usually done for displaying images. If a gamma correction is applied on histogram (b), it would compress the midrange and highlights even further.

The histogram in Figure 4 (d) shows the result of our attempt to "unflatten" the image using inverse gamma adjustment instead of our proposed restoring function. The histogram clearly shows that this gamma adjustment does not restore the distance between the highlights and the midrange, as the regions are moved together by the inverse gamma curve.

Figure 5 shows a set of test images, taken on the beach in La Jolla, CA. As can be seen in Figure 5 (a), the lighting in this scene represents a challenge for any sensor without HDR due to the variation of light intensity. Some of the harsh highlight reflections in the foreground (cactus), as well as most of the background clouds, are clipped. Figure 5 (b) shows the same scene but with the sensor HDR feature enabled and using the same parameters that were given in section 2. The highlights in the foreground and the background clouds are no longer clipped. However, tonal values are compressed,

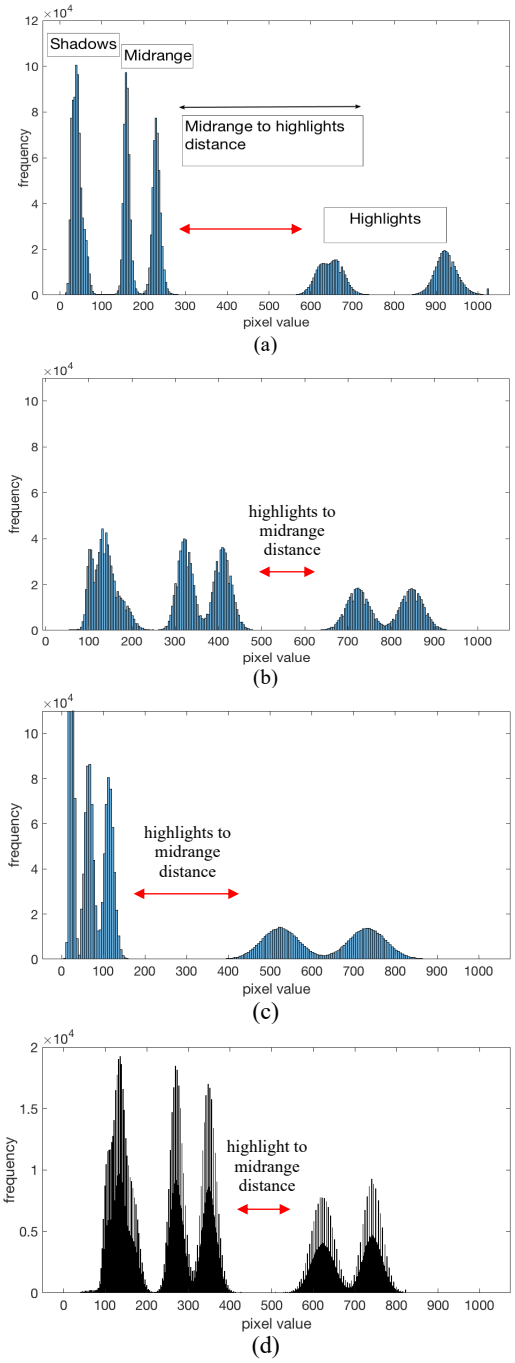


Figure 4. Histogram of raw data images of the calibration target: (a) without HDR. (b) with HDR. (c) with the proposed inverse restoring function. (d) with an attempt with reverse gamma adjustment.

leading to a rather "flat" looking image. Figure 5 (c) shows the same scene using the HDR sensor in conjunction with the proposed inverse function. Both the foreground and background show much better tonal range variation while keeping the details visible for a more pleasing "look."

A Region of Interest (ROI) of the image that shows the blue sky is highlighted in red in the three different images. The size of

this region is 50×50 pixels. A zoomed-in version of this region in the three different images is shown side by side in Figure 6. It shows the shade of blue improving gradually between the three different images.

We calculated the median of the three-color components inside this region for the three different images. Then we measured the percentage relative difference between blue and red ($d1$) and blue and green ($d2$) for the three different images, where

$$d1 = \frac{median(blue) - median(red)}{median(blue)} \times 100$$

$$d2 = \frac{median(blue) - median(green)}{median(blue)} \times 100$$

The results of this experiment are shown in Table 1.

Table 1. Relative difference between blue & red, and blue & green in the ROI of the test images

Percentage relative difference	d1	d2
Without HDR	18.43%	-0.93%
With HDR	21.04%	6.80%
With HDR and inverse function	40.03%	22.79%

As shown in the Table 1, for the image without HDR, the median of blue is only about 18% higher than the median of red and a little bit less than the median of green. In fact, due to saturation issues, the sky as a whole look a little bit greenish in this image. For the image with HDR, the percentages improve slightly to 21 % and 6%, respectively. These improvements change the color of the sky to light blue. Finally, for the image with both HDR and inverse restoring function, the percentages increase to about 40% and 23%, respectively. These improvements result in the strong blue color of the sky. All these improvements are directly related to the HDR sensor and the proposed inverse restoring function suggested in this paper. Figure 7 shows the effect of our inverse function on another set of images taken in La Jolla, CA.

Latency Results

To estimate frame rate playback, we measured the computation latency for each of stage between image acquisition and display in our software pipeline; i.e. stage latency for the inverse function, demosaicing, gamma correction and white balance. The measurements were taken for a single thread implementation without any acceleration such as vectorization or parallelization. Therefore, these measurement present estimates that are conservative for current computing systems. The implementation system we tested runs with a 2.9 GHz Intel Core i7 with 16 GB of RAM memory and Mac OS. Table 2 shows the stage latencies.

Table 2. Pipeline stage latency

Stage	Latency
Inverse function	26.6 ms
Demosacing (OpenCV)	12.3 ms
Gamma correction	3.7 ms
White balance	2.1 ms

(a)

(b)

(c)

Figure 5. Comparative test images: (a) Image without HDR. (b) Image MPRS HDR. (c) Image with our HDR restoration inverse function

Figure 6. A zoomed-in views corresponding from left to right to the three ROI's (red squares) in Figure 5. (a), (b), and (c) respectively.

Since accelerating the implementation is beyond the current scope of this work, we intend in future work to accelerate the stages through parallelization with multiple threads and using vector SIMD

instructions (single instruction multiple data). Other alternatives are multithreading on GPU or FPGA.

In any case, we believe the fastest and most efficient implementation might be in firmware on the MPRS sensor itself right after the ADC stage.

Conclusion

The multi-partial reset sensor has multiple advantages including HDR and low noise, low cost, and high frame rate. However, it suffers from rather "flat" looking images. We address this limitation with an inverse function. This function not only restores the HDR appearance of MPRS images for a more pleasing feel by improving the contrast and color content of the image, but it also can do so with a small latency. This latency makes our inverse function appealing for high frame rate, real-time videography applications. Our inverse function can be implemented in software of on the sensor firmware.

References

[1] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski, *High Dynamic Range Imaging*, 2nd ed. Morgan Kaufman, 2010.

[2] E. Fossum. (2008). *CMOS Active Pixel Image Sensors: Past, Present, and Future*. Available: http://ericfossum.com/Presentations/2008%20Jan%20CMOS%20Image%20Sensors%20Past%20Present%20and%20Future.pdf

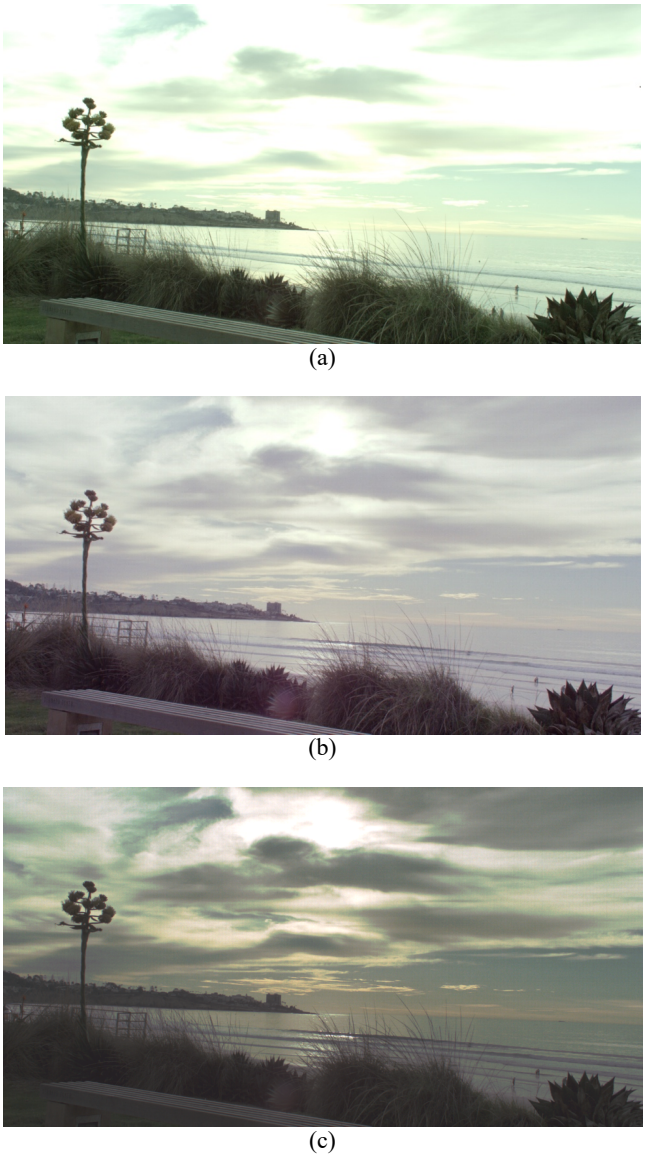
[3] E. R. Fossum and D. B. Hondongwa, "A Review of the Pinned Photodiode for CCD and CMOS Image Sensors," *IEEE Journal of the Electron Devices Society,* vol. 2, no. 3, pp. 33-43, 2014.

[4] A. Spivak, A. Belenky, A. Fish, and O. Yadid-Pecht, "Wide-Dynamic-Range CMOS Image Sensors—Comparative Performance Analysis," *IEEE Transactions on Electron Devices,* vol. 56, no. 11, pp. 2446-2461, 2009.

[5] D. Hertel, A. Betts, R. Hicks, and M. t. Brinke, "An adaptive multiplereset CMOS wide dynamic range imager for automotive vision applications," presented at the Proc. IEEE Intell. Veh. Symp., 2008.

[6] X. Wang, G. Meynants, and B. Wolfs, "Pixel structure with multiple transfer gates," United States Patent 9,001,245, 2015. Available: https://patents.google.com/patent/US9001245B2/en.

[7] D. M. Steven Decker, Kevin Brehmer, Charles G. Sodini, "A 256 x 256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-Parallel Digital Output " *IEEE Journal of Solid-State Circuits,* vol. 33, no. 12, 1998.



(a)

(b)

(c)

Figure 7. (a) Image without HDR, (b) Image with MPRS HDR, and (c) Image with HDR and our inverse function.

Author Biographies

Ziad Youssfi received his BS, MS and PhD in Electrical and Computer Engineering from Michigan State University, East Lansing, Michigan. His current research focuses on parallel image processing, deep learning, and optimizing computer architecture. Before completing his Ph.D., he joined Intel Corporation to work on the P6 line of processors and chipsets. When not working, he can be found enjoying his time with his wife and practicing fine art photography.

Firas Hassan received his BE in Electrical Engineering from Beirut Arab University, his ME in Computer & Communication from the American University of Beirut, and his PhD in Electrical & Computer Engineering from The University of Akron, OH. Since 2009, he has been an Assistant then Associate Professor at Ohio Northern University. His areas of research include image processing, information theory, and encryption. When not working, he can be found enjoying him time with his family.

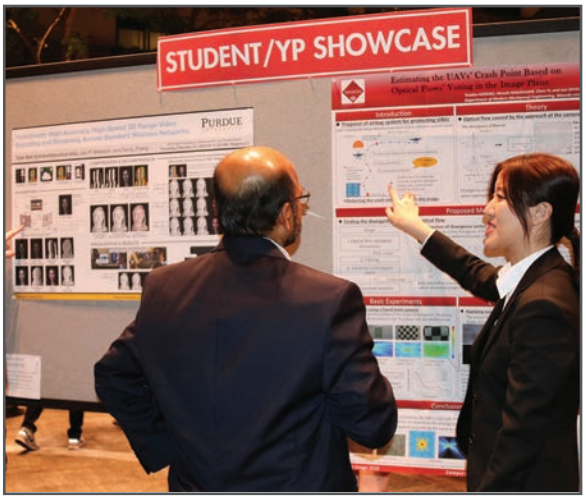
JOIN US AT THE NEXT EI!

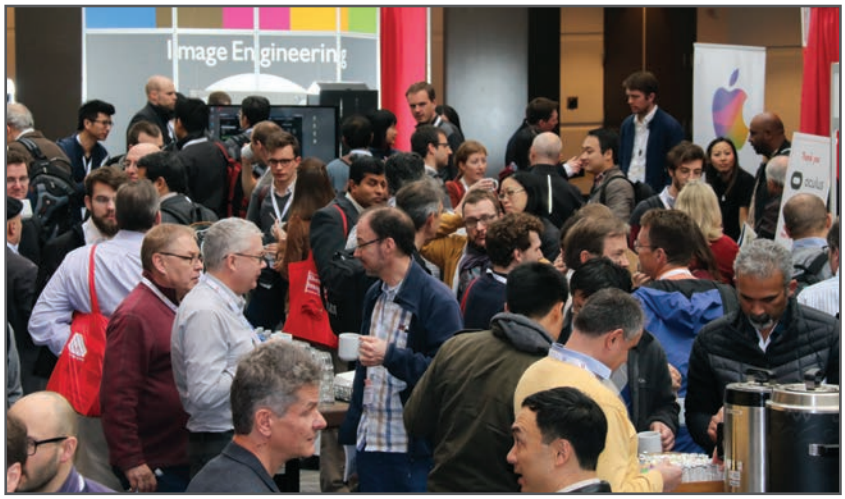
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org