A simulation tool for evaluating digital camera image quality

Joyce E. Farrell[a], Feng Xiao[b], Peter B. Catrysse[c], Brian A. Wandell[bc]

[a]ImagEval Consulting LLC, P.O. Box 1648, Palo Alto, CA 94302-1648
[b]Department of Psychology, Stanford University, Stanford, CA 94305
[c]Department of Electrical Engineering, Stanford University, Stanford, CA 94305

ABSTRACT

The *Image Systems Evaluation Toolkit (ISET)* is an integrated suite of software routines that simulate the capture and processing of visual scenes. *ISET* includes a graphical user interface (GUI) for users to control the physical characteristics of the scene and many parameters of the optics, sensor electronics and image processing-pipeline. *ISET* also includes color tools and metrics based on international standards (chromaticity coordinates, CIELAB and others) that assist the engineer in evaluating the color accuracy and quality of the rendered image.

Keywords: Digital camera simulation, virtual camera, image processing pipeline, image quality evaluation

INTRODUCTION

ISET is a software package designed to assist engineers in image quality evaluation and imaging system design. Imaging systems systems, including cameras and displays, are complex and require transforming signals through a number of different devices. Engineers typically evaluate isolated components of such systems. Customers judge imaging systems, however, by viewing the final output rendered on a display or on a printer, and this output depends on the integration of all the system components. Consequently, understanding components in isolation, without reference to the characteristics of the other system components, provides only a limited view. In these types of complex systems, a controlled simulation environment can provide the engineer with useful guidance that improves the understanding of design considerations for individual parts and algorithms.

In this paper, we describe the first major application of *ISET*: a virtual camera simulator. This virtual camera simulator was designed to help users evaluate how image capture components and algorithms influence image quality. *ISET* combines optical modeling [1] and sensor simulation technology [2] developed at Stanford University (Stanford Docket S00-032) with non-proprietary image processing algorithms and display models to simulate the entire image processing pipeline of a digital camera.

A true simulation must begin with a complete representation of the physical stimulus, and without such data sets, accurate simulation is impossible. Obtaining accurate input data has been a major limitation in developing a digital camera simulator and only a few radiometric image data sets are available [3,4,5]. *ISET* includes a set of radiometric input stimuli and a methodology for acquiring additional stimuli based on off-the-shelf hardware. The availability of such images permits the user to simulate a complete imaging pipeline, including a radiometric description of the original scene, optical transformations to irradiance signals, sensor capture, and digital image processing for display. The simulation also includes image quality metrics for evaluating and comparing different rendered outputs. The tool provides an integrated programming environment that enables users with different backgrounds to experiment with various components of the imaging pipeline and measure the final result in terms of image quality.

SIMULATOR OVERVIEW

The digital camera component of the *ISET* software is organized around four key modules: **Scene**, **Optics, Sensor**, and **Processor**. Each module includes a variety of specialized tools and functions that help the user set parameters, experiment with alternative designs and component properties, and calculate relevant metrics. The software package

uses a Graphical User Interface (GUI) to assist the user in experimenting with the parameters and components within the image systems pipeline. The software is based on Matlab and an open architecture, making it possible to access the intermediate data and apply proprietary functions or metrics.

The input to the virtual camera simulation is a radiometric scene description that is managed by the **Scene** module (Figure 1). The scene radiance is converted into an irradiance image at the sensor by the algorithms contained within the **Optics** module (Figure 2). The conversion from radiance to irradiance is determined by the properties of the simulated optics. The sensor irradiance image is converted, in turn, into electron counts within each pixel of the image sensor. The **Sensor** module (Figure 3) manages the transformation from irradiance to sensor signal. This transformation includes an extensive model of both optical and electrical properties of the sensor and pixel. The **Processor** module (Figure 4) transforms the electron count into a digital image that is rendered for a simulated color display. This module includes algorithms for demosaicing, color conversion to a calibrated color space, and color balancing.

At any stage of the process, the user can extract intermediate data, apply experimental algorithms, replace the data, and then continue the simulation. The simulator uses the convenient and open Matlab architecture; the software includes a suite of tools for extracting and substituting data without requiring the user to have intimate knowledge of the internal workings of the simulator itself.

In the following section, we describe some of the main properties of each of the modules. Then we describe some applications of the virtual camera simulator in *ISET*.

Scene

Digital camera simulation requires a physically accurate description of the light incident on the imaging sensor. *ISET* represents image scenes as a multidimensional array describing the spectral radiance (photons/sec/nm/sr/m^2) at each pixel in the sampled scene. The spectral radiance image data are assumed to arise from a single image plane at a specified distance from the optics. Variables such as the spectral wavelength sampling, distance to the plane, mean scene luminance, and spatial sampling density can be interactively set by the user. The Scene window is illustrated in Figure 1, which shows how the user can investigate the image data by selecting a region of the image and plotting the scene radiance. Different scenes can be created using options in the pull-down menus, and the properties of the current scene, such as its' luminance, spatial sampling density, and field of view, can be adjusted interactively as well. Multiple scenes can be loaded into a single session to make it easier to compare the effects of different scene properties on the imaging pipeline.

There are several different sources of scene data. For example, there are synthetic scenes, such as the Macbeth ColorChecker, spatial frequency sweep patterns, intensity ramps and uniform fields. When used in combination with image quality metrics, these synthetic target scenes are useful for evaluating the color accuracy, spatial resolution, intensity quantization, noise and other properties of the imaging system.

Another important source of scene data are calibrated representations of natural scenes. The scene data are acquired using a high-resolution CCD camera with external multiple filters and multiple exposure durations. Eighteen images of each scene are captured with a high-resolution (6 megapixels) RGB CCD digital camera using six exposure durations and three external color filter conditions. The multiple filters expand the spectral dimension of the scene data; the multiple exposures expand the dynamic range of the scene data. We estimate the effective dynamic range of this multi-capture imaging system to be greater than 1000,000: 1. The scene data from these multiple captures are then combined, using linear models of surfaces and lights, into a final high dynamic range multispectral scene image [6]. Details about the method and accuracy of the multi-capture imaging system are described at http://www.imageval.com/ISET.htm. Finally, scene radiance image data can be generated from RGB data. The user can either select or provide information about the source of the RGB data (e.g. CRT display phosphors and gamma) and *ISET* uses linear models to estimate the spectral radiance image data.
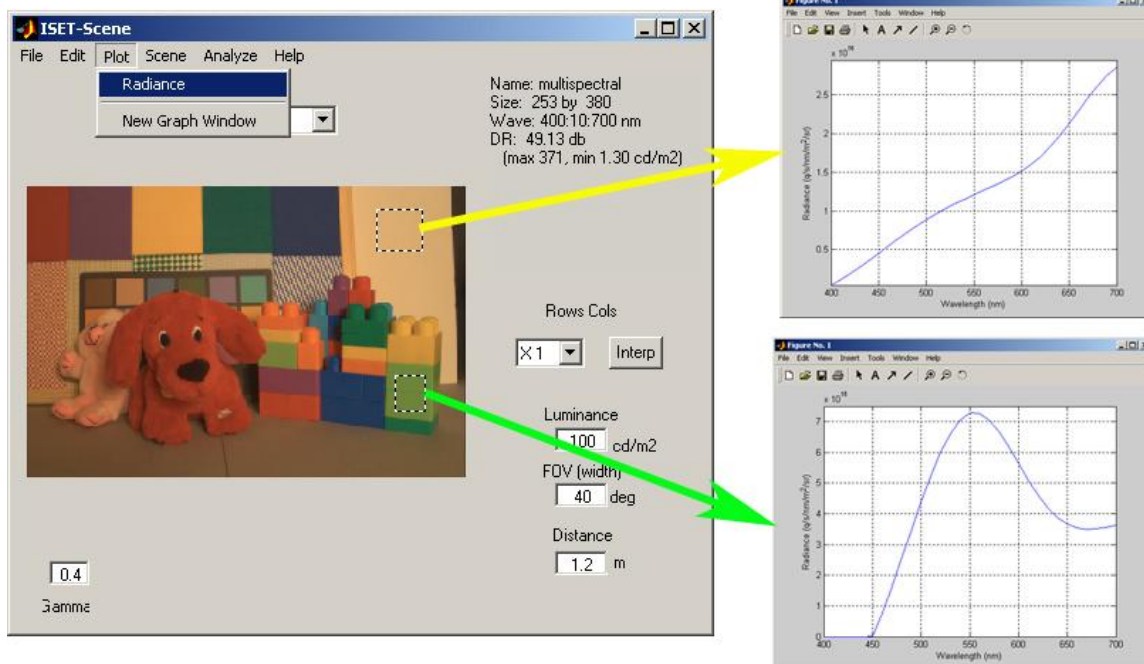
Figure 1. The Scene Window displays a representation of the scene spectral radiance. The user can view the spectral radiance data by selecting any region in the image scene representation, as indicated by the dashed squares above. On the right, the top graph shows the spectral power distribution of a tungsten light reflected from a white surface. The lower graph shows the combined effect of the tungsten illuminant and the spectral reflectance of a green surface.

Optics

The imaging optics are modeled using a wave-optics approach which takes into account the finite resolution obtained with finite size optics. The user can vary the size of the aperture of the imaging optics by changing the f-number, which will automatically result in an adjustment of the image irradiance and resolution. Finite resolution is calculated using an optical transfer function (OTF) approach, which is based on the finite aperture as determined by the f-number. To account for wavelength dependent behavior, the OTF is implemented in a spectral manner.

Image irradiance (photons/sec/nm/m^2) is determined using radiometric concepts and includes the effect of the off-axis cos-4th effect, which results in a darkening of the corners with respect to the center of the image when a uniform object is imaged.

Metrics for analysis of the optics include plots of the pointspread function (PSF), linespread function (LSF), and optical modulation function. Figure 2 illustrates the PSF in two optics configurations, in which the f-number of the lens is changed. The size of the image, the image magnification, and the irradiance in the image plane are all represented in the data structures and are all described interactively in the upper right hand portion of the Optics window.
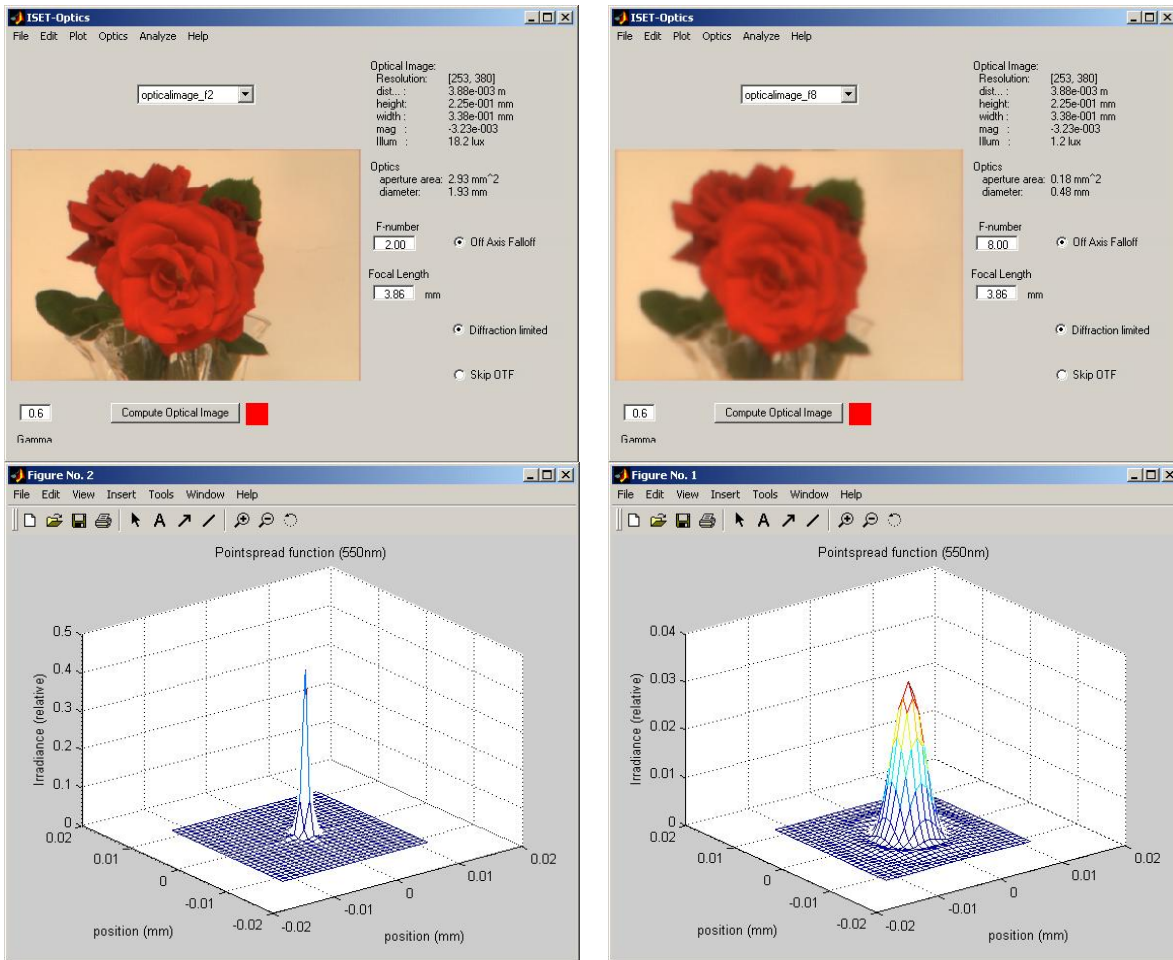
Figure 2. The Optics module transforms the scene data into an irradiance image in the sensor plane. By interactively selecting image regions, the user can plot the irradiance data. A variety of standard optical formats can be chosen and their properties can be analyzed. The images on the left show a standard optics configuration using a low f-number (2) and the pair of images on the right show the same scene imaged using a large f-number (8). The PSF for each of the optics, measured at 550nm, is shown in the panels below.
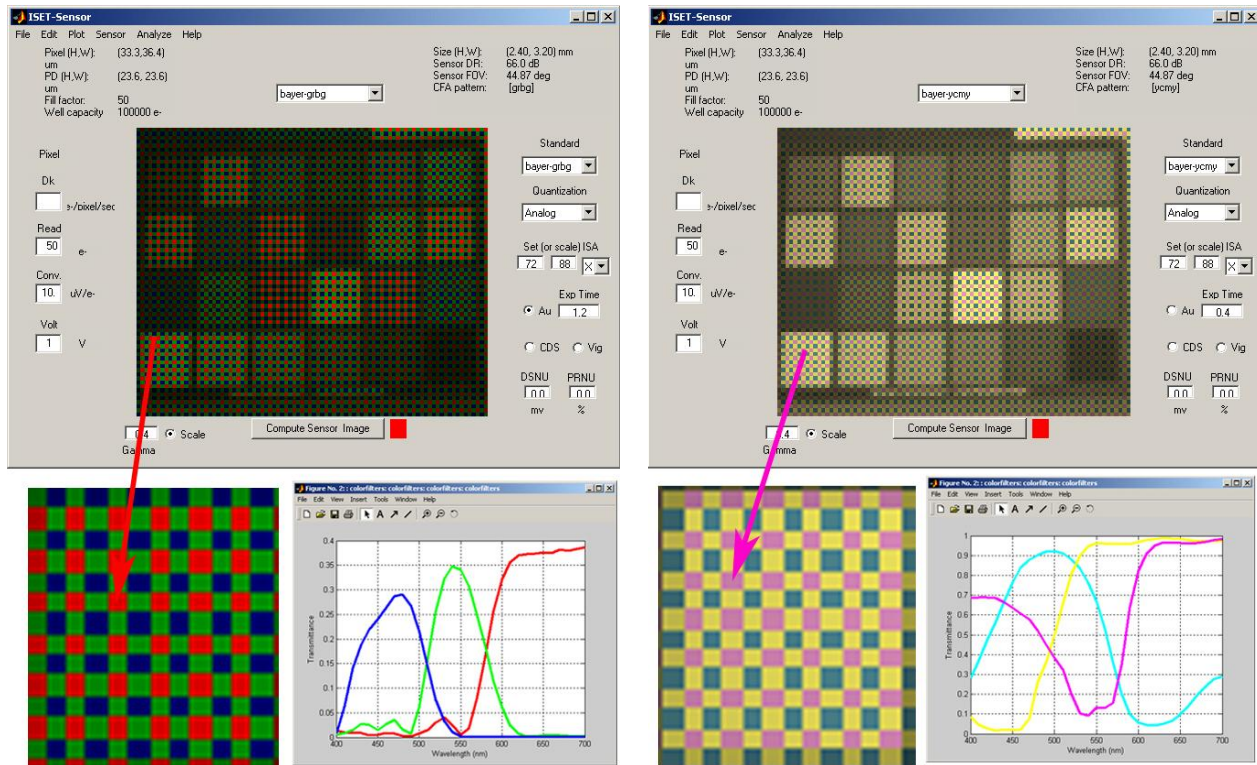
Sensor

The function of the Sensor module is to simulate the transformation of irradiance (photons/sec/nm/m^2) into an electrical signal. The image sensor model includes a great many design parameters, only some of which can be discussed here. Among the various factors accounted for are the spatial sampling of the optical image by the image sensor, the size and position of pixels and their fill-factor. The wavelength selectivity of color filters, intervening filters (such as an infrared filter), and the photodetector spectral quantum efficiency are also included in the simulation. The sensor current (electrons/s) is converted into a voltage (V) through direct integration for a given exposure time (sec) and using a user-specified conversion gain (μV/electron). Various sources of noise (read noise, dark current, dark signal non-uniformity (DSNU), photoresponse non-uniformity (PRNU), photon noise) are all included in the simulation. Finally, to complete the physical signal pipeline, the analog voltage (V) is converted into a digital signal (DN) according to the specifications of the user (analog-to-digital step size in V/DN).

The graphical user interface of the Sensor module permits the user to select a variety of color filters and alter their arrangement. The user can also set the size of individual pixels, the fill-factor of the photodetector within the pixel, and the thickness of different optical layers. The geometrical effects of pixel vignetting can be estimated [7], and the

combined effects of all of the wavelength dependent functions can be plotted and summarized in a graph of the sensor QE.

The interface includes a variety of metrics that summarize the quality of the sensor, including the ISO saturation of the sensor, the signal-to-noise (SNR) of individual pixels as well as the SNR of the entire sensor array. Individual noise parameters, such as the dark current, read noise, conversion gain, and voltage swing can all be set interactively. Finally, the simulated sensor output can include the effects of vignetting, correlated double-sampling (CDS).

Figure 3 illustrates two sensors using different color filter arrays. The scene on the left is a simulation of the capture using an RGB sensor mosaic (no infrared filter) and the scene on the right illustrates a CMY filter pattern (again, no infrared filter). The Sensor window renders the current at each pixel using a color that represents the pixel spectral quantum efficiency.

Figure 3. The Sensor Window displays the electron count for each pixel in the sensor image. The pixels are coded in terms of the color filter placed in front of each pixel in the photodetector array. The images on the left side of the figure display data for a sensor with a GBRG (green-blue-red-green) color filter array (CFA) and the images on the right correspond to data for a YCMY (yellow-cyan-magenta-yellow) CFA. The bottom images illustrate the CFA by enlarging the sensor data for a white surface in the Macbeth ColorChecker. The bottom graphs show the spectral transmittance of the RGB sensors (left) and CMY sensors (right).
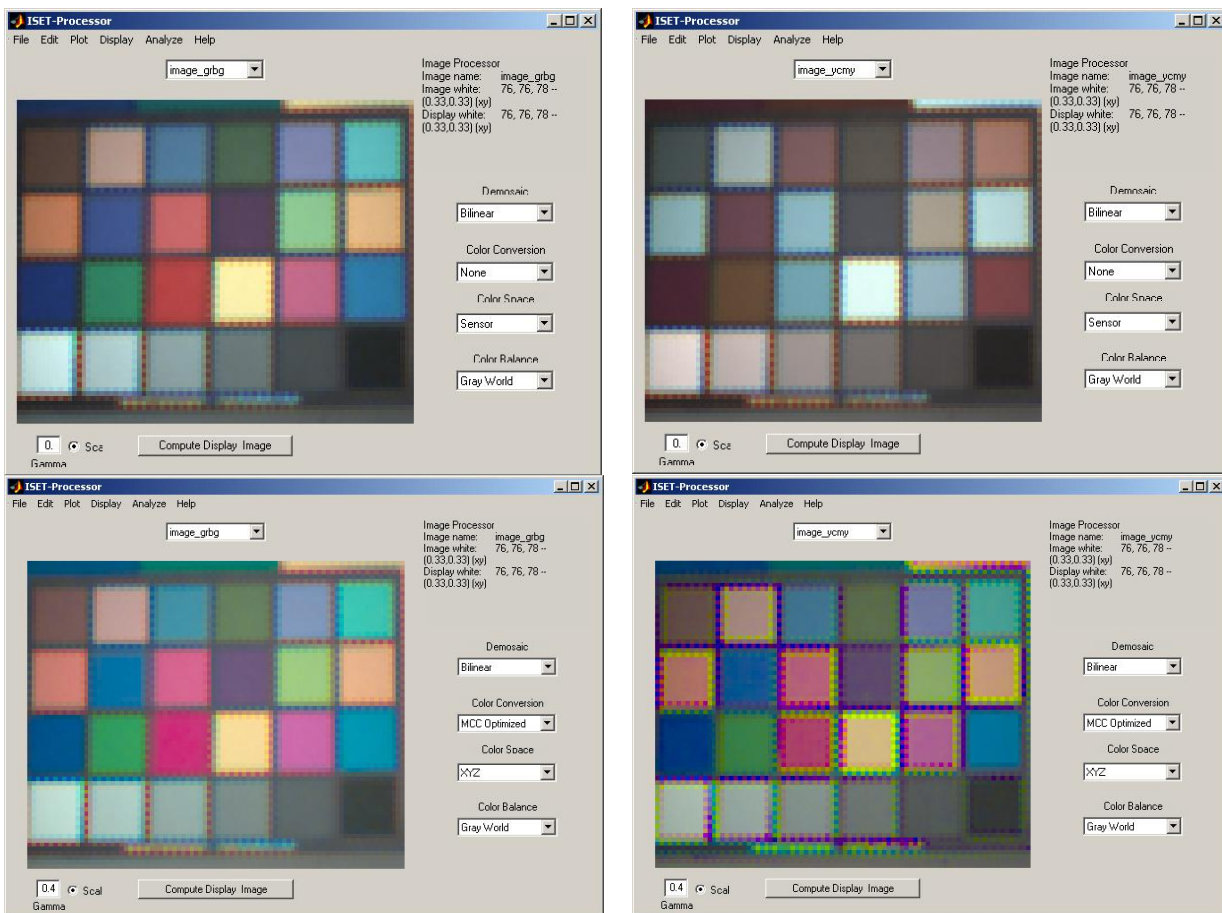
Processor

The Image-Processing Module transforms the linear sensor values into an image for display. The module includes several standard algorithms for basic camera operation. These include algorithms for interpolating missing RGB sensor values (demosaicing) and transforming sensor RGB values into an internal color space for encoding and display (color-balancing, color-rendering and color-conversion). Because of the simulator's extensible and open-source architecture

(Matlab) the user can also insert proprietary image-processing algorithms in the pipeline and evaluate how these algorithms will perform under various imaging conditions or with various types of sensors. The user can also specify the properties of the target display device, including its white point and maximum luminance.

As with the other windows, the user can maintain several different images at once making it simple to compare between different algorithms, sensors and scenes. Several metrics for analysis of the processed and rendered image are present in this window, including measurements of CIE XYZ, CIELAB and chromaticity coordinates.

Figure 4 illustrates the transformation of RGB sensor data (on the left) and CMY sensor data (on the right) using the Image Processor module. The images on the top show the rendering when the sensor data are color-balanced using a Gray World algorithm. The images on the bottom show how the colors can be improved by including internal color transformations to estimate the scene properties and color balance the display. The CMY images also show that the color transformations needed to bring the CMY sensor array into an accurate color rendering can amplify the image spatial noise [8].


Figure 4. The Processor Window displays a digital image rendered for a simulated display. In the examples shown above, the sensor images were "demosaiced" using bilinear interpolation to predict the missing sensor values in the GBRG (left) or YCMY (right) sensor arrays. The images in the top row were color-balanced using a Gray World algorithm that operated on the RGB (left) and CMY (right) image data. The images in the bottom row represent sensor image data that were converted to XYZ and then color-balanced using the Gray World algorithm in the XYZ color space [9]. The color-balanced data for all four images were converted to a calibrated sRGB color space for display.

SIMULATOR APPLICATIONS

ISET can be used to evaluate how sensor and processing design decisions will influence the quality of the final rendered image. For example, *ISET* includes tools to quantify the visual impact of different types of sensor noise and rendering algorithms. It is also possible to use ISET to simply look at the visual impact of different algorithm choices. In this section, we illustrate one such application by illustrating the effect of correlated double sampling on perceived image quality. Additional application examples are described at http://www.imageval.com/ISET.htm.

Several variants of CDS have been proposed and implemented [10]. For this example, we use the form of CDS in which the sensor acquires two images: a conventional image (signal plus noise) and a second image with zero exposure duration (noise). The CDS algorithm subtracts these two images to eliminate fixed pattern noise. Other types of non-random noise, however, may still be visible.

The simulations in the top part of Figure 5 illustrate the benefits of including CDS under bright imaging conditions. The mean of the simulated sensor illuminance in the upper two images is 17.4 lux and the two lower images is 1.74 lux. The images on the left are processed without using CDS and the images on the right include CDS. The fixed simulation parameters include the DSNU set at 2 mV standard deviation with a total voltage swing of 1V. The exposure duration was set to 16 ms, similar to a single frame in a 60Hz video capture. The pixel size was 4 micron and the photodetector fill factor was 50%. The dark current was 10 electrons/pixel/second; the read noise was 50 electrons; the conversion gain was 15 μV/electron; and the PRNU was 0. Adaptive demosaicing and Gray World color balancing (performed in CIE 1931 XYZ color space) were used. The optics were assumed to be conventional quarter-inch optics with f# = 2.0
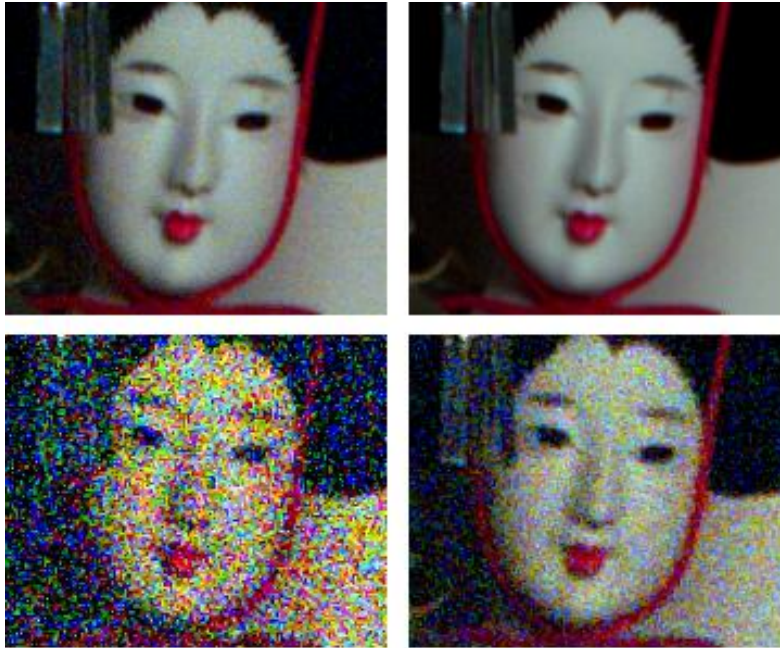
Figure 5. Simulations illustrating the benefits of correlated double sampling (CDS) at different levels of sensor irradiance (see text for details).

Figure 5 illustrates that, in this simulation, the main source of noise at high intensity is the DSNU. The DSNU is visible in the image (upper left) and this noise is eliminated by the CDS operation (upper right). At low intensity, the photon noise is very high and combines with DSNU (lower left). Subtracting the DSNU improves the image, but the other noise sources are still quite visible (lower right).

CONCLUSIONS

The complexity of the digital imaging pipeline, coupled with a moderate array of image quality metrics, limits our ability to offer closed form mathematical solutions to design questions. In such cases, simulation technology can be a helpful guide for engineers who are selecting parts and designing algorithms. The simulation technology implemented in *ISET* helps in several ways.

First, an engineer working on one part of the system – say, demosaicing algorithms – need not be familiar with the physical simulation of the sensor itself. Similarly, the engineer working to reduce circuit noise need not be an expert in demosaicing algorithms. The simulator provides both users with a framework that implement the less familiar parts of the pipeline. In this way, *ISET* improves collaboration between people with different types of expertise.

Second, with the open-source software architecture, the user can explore and experiment in detail with the portion of the pipeline of major interest.

Third, *ISET* offers a unique set of multispectral input images, correcting a serious deficiency in most attempts to simulate the digital imaging pipeline.

Fourth, ISET includes two methods for evaluating the rendering pipeline. *ISET* produces fully rendered images so that the user can see the effects of parameter changes. *ISET* also includes a suite of image quality tools that can provide quantitative measurements to characterize sensor properties and to quantify color and pattern reproduction errors. The combination of visualization and quantification increases the user's confidence that sensible design and evaluation decisions will be made.

REFERENCES

1. P. B. Catrysse, *The Optics of Image Sensors*, Stanford University doctoral thesis, 2003.

2.   T. Chen, Digital *Camera System Simulator and Applications*, Stanford University doctoral thesis, 2003.

3. Vora, P.L, Farrell, J. E., Tietz, J. D., and Brainard, D. H., Image capture: simulation of sensor responses from hyperspectral images, *IEEE Transactions on Image Processing*, 10, pp. 307-316, 2001.

4. Longere, P. and Brainard, D. H., Simulation of digital camera images from hyperspectral input. In *Vision Models and Applications to Image and Video Processing*. C. van den Branden Lambrecht (ed.), Kluwer. pp. 123-150, 2001.

5. http://color.psych.upenn.edu/hyperspectral/

6. F. Xiao, J. DiCarlo, P. Catrysse and B. Wandell, High dynamic range imaging for natural scenes, In *Tenth Color Imaging Conference: Color Science, Systems, and Applications, 2002*. Scottsdale, AZ.

7. P. B. Catrysse and B. A. Wandell, Optical efficiency of image sensor pixels, *Journal of the Optical Society of America A*, Vol. 19, No. 8, pp. 1610-1620, 2002.

8. U. Barnhofer, J. DiCarlo, Ben Olding and B. Wandell, Color estimation error trade-offs, In *Proceedings of the SPIE,* Vol 5027, pp. 263-273, 2003

9. F. Xiao, J. Farrell, J. DiCarlo and B. Wandell, Preferred color spaces for white balancing, In *Proceedings of the SPIE Electronic Imaging '2003 Conference*, Vol. 5017, Santa Clara, CA, January 2003 , San Jose

10. M. H. White, D. R. Lampe, F. C. Blaha, and I. A. Mack, Characterization of surface channel CCD image arrays at low light levels, *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 1-14, 1974.