8

Image Systems Simulation

Joyce E. Farrell and Brian A. Wandell
Stanford University, Stanford, CA, USA

1 Introduction

Imaging systems are designed by a team of engineers, each with a different set of skills and expertise. The team members may work in separate organizations that specialize in different imaging system components, such as optical lenses, filters, sensors, processors, or displays. They may use different analytical tools and language to characterize the imaging component that they design. Engineering teams must make decisions together about the costs and benefits of a design change. Image systems simulations can enhance communication and collaboration between people with different types of expertise. A simulation environment helps engineers to (i) communicate effectively across different realms of expertise, (ii) predict the effect that changes to individual imaging components will have upon system performance, and (iii) experiment with new designs without incurring the cost of building a physical device.

Image systems simulations make it possible to both visualize and quantify how changes in system components will influence image quality. For example, one can evaluate the effects that different optical and sensor components have upon image quality, or how these changes will impact image-processing algorithms. Moreover, simulations make it possible to evaluate the performance of an imaging system under conditions that are difficult to recreate in the laboratory, including high dynamic range images, low light level images, object or camera motion, and so forth.

The importance of simulations has been recognized since the early days of imaging. For example, software simulations played a key role in the design and evaluation of the imaging system used in the first Mars Landing mission in the mid-1970s [1, 2]. Members of the engineering and scientific team that worked on this mission designed a spectrophotometric stereo imaging system that would capture and transmit data from an automated rover vehicle when it landed on Mars [1]. This team not only had to imagine the atmosphere, terrain and possible objects on Mars, but also had to anticipate problems that could arise from radiation

Handbook of Digital Imaging. Edited by Michael Kriss.
ISBN 978-0-470-51059-9.

encountered during the year it took for the imaging system to arrive on Mars. In addition, scientists needed to consider possible effects of lighting, geometry, dust, and abrasions on the quality of the images that the Mars Viking Landing rover would capture on Mars and transmit back to earth.

In remote sensing, image systems simulation is referred to as *end-to-end* simulation [3, 4] and also as image chain analysis [5]. These terms emphasize the importance of evaluating individual imaging components in the context of a complete image systems simulation. Modeling the complete system for remote imaging systems [6] requires (i) characterizing spectral properties of possible targets, (ii) modeling atmospheric conditions, (iii) characterizing the spectral transmissivity of filters, the sensitivity, noise, and spatial resolution of imaging sensors, and (iv) implementing image processing operations, including exposure control, quantization, and detection algorithms.

We developed image systems simulation software for consumer imaging that parallels this methodology [7, 8]. The software (i) represents the radiometric properties of scenes and illuminants, (ii) models image formation through the main lens, (iii) characterizes the sensitivity and noise of sensors, including spatial and spectral sampling of color filter arrays (CFAs), (iv) includes image processing algorithms, and (v) outputs a calibrated (radiometric) representation of the rendered image. In this chapter, we describe this approach to integrate the component models into an image system simulation, and we illustrate how to use image systems simulation software to evaluate design tradeoffs, invent new imaging systems, and optimize image-processing algorithms.

2 Image Systems Simulation Software

Image systems simulation software should clarify how the system components work together to produce the final result; insights from the software should allow for continuing improvements to the system components and create the opportunity to experiment with new designs and components. With these goals in mind, we developed a simulation environment comprising a set of distinct software objects that capture the variety of imaging components and how these objects transform data along the imaging pipeline [7, 8]. The most important objects are the scene, optics, sensor, processor, and display (Figure 1). The *scene* is a radiometric description of the input data. The *optics* object defines the lens properties that convert the scene into an irradiance image at the sensor surface. The *sensor* defines the properties of the pixels and sensor array that govern how the irradiance image is converted into electrons. The *image processor* (*IP*) object is a collection of algorithms that define how sensor data are transformed into display values. The *display* object is a radiometric description of the final image for any calibrated display. The software functions in the simulation environment act on these objects and the associated data. For example, there are functions that combine the scene and optics objects to calculate the irradiance at the sensor. Other functions combine the irradiance with the sensor object to calculate the sensor data.

The software is designed to support calculations of different degrees of complexity. For example, if the optics is a simple diffraction-limited lens, image formation functions convert the scene radiance to the irradiance at the sensor using established closed-form equations. If the optics is a multicomponent lens that includes information about geometric distortion and space-varying point spread functions from a ray-tracing program, then functions are invoked that use more complex computational methods. A flexible image systems simulation

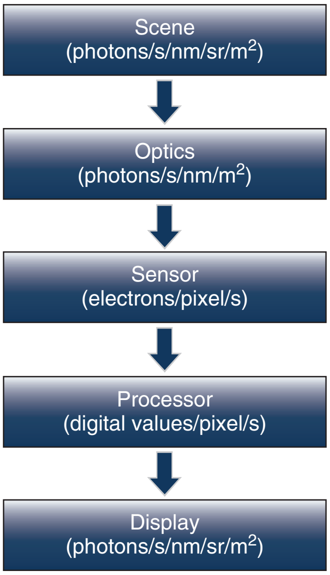

Figure 1 **An image systems simulation environment**. The software is organized around objects and associated data representing the scene, optics, sensor, processor, and display

environment should allow the user to insert new ideas and models into the pipeline while preserving compatibility with the existing computational framework.

In the following sections, we describe the general principles and some specific formulae that one expects to be implemented in an image systems simulation.

3 Scene

The image systems simulation begins with a radiometric description of the light in the scene (Figure 2). To account for the effects of optics (e.g., defocus, chromatic aberration), filters [CFA and infrared (IR) blocking filters], and photodetectors (spectral quantum efficiency), it is necessary to begin with the spectral radiance. However, there are various degrees of radiometric completeness; many useful calculations can be performed even with only a partial description of the scene spectral radiance.

A complete scene radiometric description, L, describes the rate of photons in every scene position (x, y, z), direction (θ, ϕ), wavelength (λ), and moment in time [9]. In addition, one might specify the polarization, although, in this chapter, we ignore polarization and time henceforth. The units of radiance are normalized per time interval (s), solid angle (sr), and area of the point source (photons/s/nm/sr/m^2).

The complete scene radiometric function, $L(x, y, z, \theta, \phi, \lambda)$, is rarely used, and for image systems simulation it is not generally needed. The most common simplification is a static, two-dimensional synthetic scene of a Lambertian surface. Examples are the Macbeth ColorChecker, spatial test patterns, luminance ramps, and uniform fields. These targets have a scene radiance that depends only on $L(x, y, \lambda)$, because they are restricted to a single depth (z), and emit equally in all directions (θ, ϕ), and are constant across time (t). When

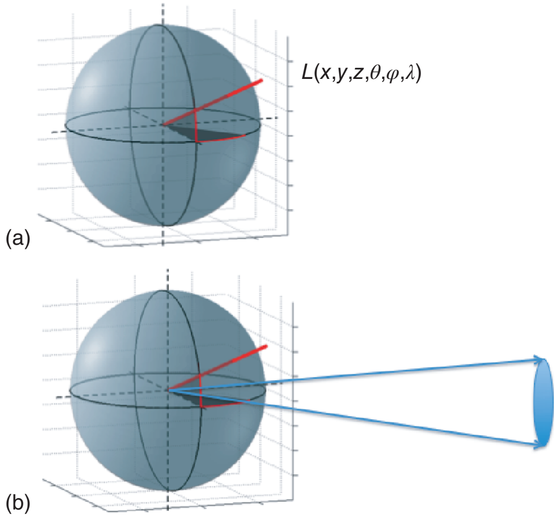

Figure 2 Radiometric description of the scene. The scene spectral radiance, L, is described for each point (x, y, z) in the scene and each direction (θ, φ) and for each wavelength, λ. (a) The red line denotes the direction of a ray from a single point. (b) The red arcs indicate the angle of the ray. In most imaging applications, only a small portion of the scene radiance arrives at the lens. In this case, only rays that fall within the cone defined by the blue lines enter the lens; the red ray and many others do not enter the lens

used in combination with image quality metrics, these synthetic target scenes are useful for evaluating specific features of the system, such as color accuracy, spatial resolution, intensity quantization, and noise.

A simple representation of scene radiance is also sufficient for analyzing most aspects of sensor and illuminant correction algorithms. It is possible to create approximations to spectral representations of natural scenes from RGB images by using a model of a standard display (e.g., sRGB) and assuming that the display white point is the illuminant [10]. More accurate spectral representations of natural scenes can be generated using hyperspectral and multispectral imaging methods [11–19]. These data do not provide in-depth information, but they do provide insights about the typical dynamic range and spectral characteristics of the likely scenes.

A more extensive description of the scene radiance is required for analyzing other aspects of system performance. For example, to analyze the depth of field, the simulation must include information about the distance of the scene point from the camera, $L(x, y, z, \lambda)$.

An important and related representation is to specify the irradiance at the first aperture (Figure 3), which is the only part of the scene radiance that the camera can acquire. This irradiance, called the *plenoptic function* [20] specifies for each point in the plane, (s, t), the photons arriving from each direction. The plenoptic function, $P(s,\ t,\ \theta,\ \phi,\ \lambda)$, has units of spectral irradiance, (photons/s/nm/m^2). In computer graphics, an alternative parameterization is commonly used, $P(s, t, u, v, \lambda)$, where the angles are replaced by (u, v), the coordinates in a plane parallel to the aperture but positioned in the scene.

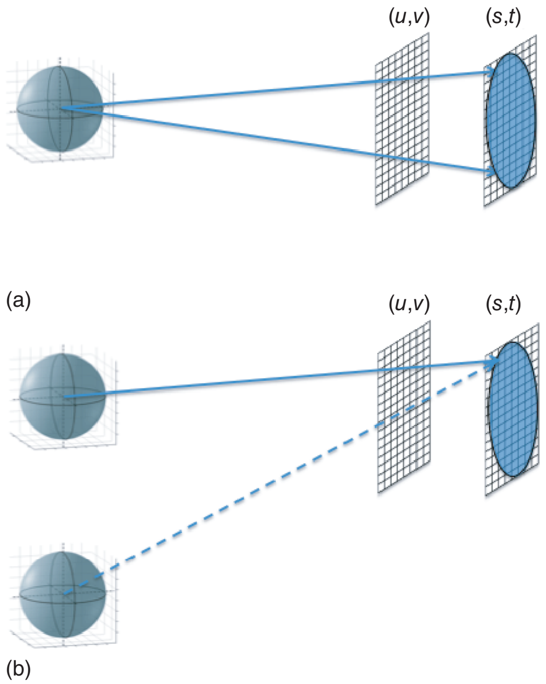

Figure 3 The plenoptic function (light field). While the scene spectral radiance is defined by a six-dimensional parameterization (Figure 2), the light field describes only those rays that are incident at the surface of the lens. We can parameterize all the rays that arrive at the lens by defining their position in two planes – one in the plane of the lens aperture (s, t), and a second parallel plane slightly outside the aperture plane (u, v). Each ray incident at the lens can be uniquely defined from its position in these two planes (u, v, s, t). The light field incident at the lens, therefore, can be defined as $P(u, v, s, t, \lambda)$. This light field is a full description of all the rays that enter the lens. Many key effects of the optics can be modeled using linear operators that operate on the light field [21]

In image systems software, a particularly important example of the plenoptic function is the radiance from the plane of the lens exit aperture, (s, t), to the sensor plane, (u, v). This representation is called the *light field* [22] or *lumigraph* [22, 23]. Light field cameras are designed to estimate this function; knowledge of this plenoptic function permits users to manipulate the image rendering in useful ways, such as refocusing the depth plane or depth of field [24].

The complexity of the scene representation should match the simulation goals. A simple representation of scene radiance, $L(x, y, \lambda)$, is sufficient to predict the visibility of blur, noise, or color differences. A more complex representation, $L(x, y, z, \theta, \phi, \lambda)$, is necessary to analyze transparency, depth of field, and synthetic apertures.

When possible, it is useful to separate the scene radiance into an illuminant term and a surface reflectance term. For some purposes- it is enough to represent the illuminant as a single vector that is constant across the scene. In other cases, it is best to represent the illuminant as a multidimensional array that varies with scene position (Figure 4). In either case, dividing the scene radiance by the scene illumination at a position approximates the surface reflectance.

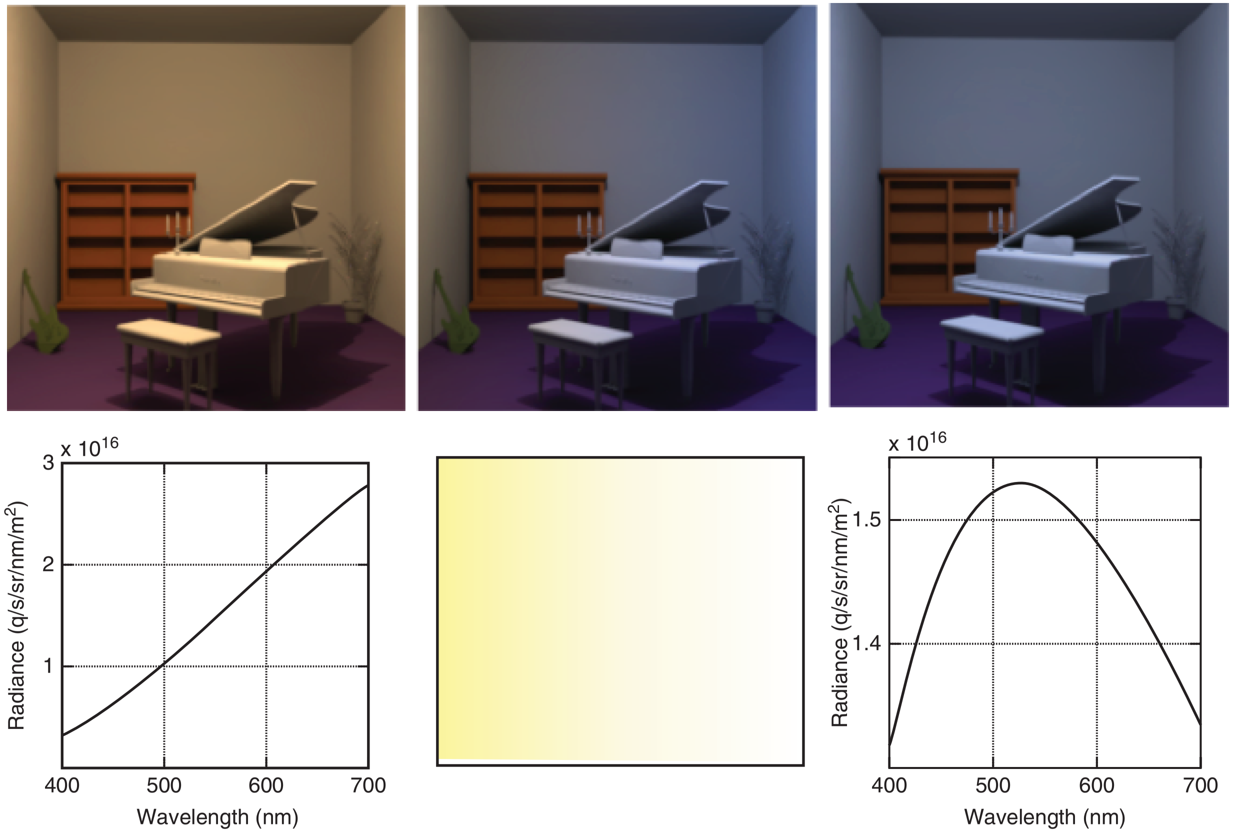

Figure 4 **Spatial variation in the scene illumination**. A synthetic scene generated from computer graphics data is shown. The three scenes show the same surfaces rendered using different illuminant spectral power distributions. The scene on the left is rendered with a 3500 K blackbody radiator and on the right with a 7000 K blackbody. The middle is rendered with a space-varying illuminant that is 3500 K on the left and 7000 K on the right. As the scene is generated using computer graphics, the depth map is known exactly. This information can be used to account for depth of field effects when using the optics to compute the sensor irradiance

3.1 *Efficient Scene Representations*

The size of the spectral representations can be significant, particularly when both the surface and illuminant vary across the scene. For example, hyperspectral imagers capture a sequence of images of the same scene finely sampled over a wide range of wavelengths. Data from hyperspectral imagers that use CCD (charge coupled device) or CMOS (complementary metal oxide semiconductor) sensors can capture spectral data between 400 and 1000 nm. Hyperspectral imagers that use indium gallium arsenide (InGaAs) or mercury cadmium telluride (HgCdTe) sensors can capture between 900 and 1700 nm and 1000 and 2500 nm, respectively. Representing scene radiance with such high spectral resolution can require significant amounts of memory. For example, a 1 K × 1 K image represented from 400 to 770 nm at 4 nm spacing (93 samples) at double precision is about 750 MB.

It is possible to reduce the dimensionality of the spectral data by using linear models for surfaces and illuminants [19]. First, scene radiance data can be expressed as the linear combination of separate functions describing the spectral reflectance of surfaces and the spectral power of scene illumination. Second, these functions can themselves be described by linear combination of a smaller set of spectral basis functions. For example, the spectral power distribution of natural daylights has been measured by various groups around the world, and there is consensus that daylights can be represented by a weighted combination of three spectral basis functions [25]. It is less certain how to randomly sample natural surfaces, but over the last few decades multiple groups have sampled surface reflectance functions of natural objects and reached a consensus that these can be represented to an accuracy of 1–2% using a linear model comprising as few as six or seven spectral basis functions [19, 26, 27]. Such linear models provide a compact representation: rather than representing the data with 93 wavelength samples, it is adequate to use only the seven coefficients of the spectral basis functions, reducing the memory requirement by about a factor of nearly 15 (Figure 5).

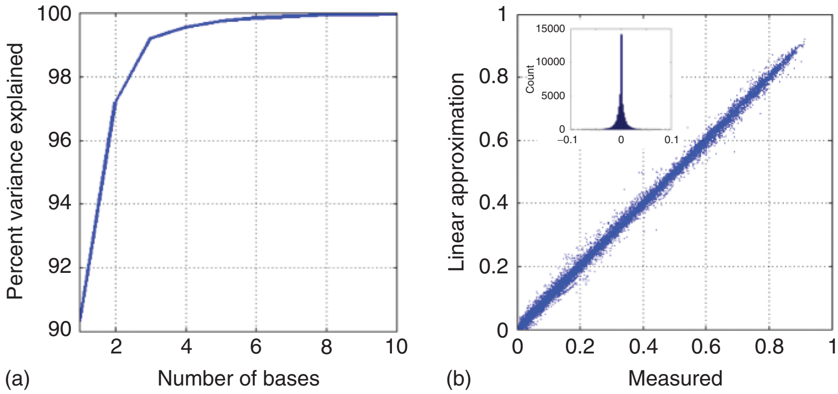

Figure 5 Linear model of surface reflectance. Using the singular value decomposition, we calculated the optimal linear basis functions for predicting the spectral reflectance of a set of containing 700 surfaces. (a) The percent variance accounted for increases with the number of bases. The first 10 bases explain the data quite well, beyond the accuracy of the original measurements. (b) The scatter plot shows the prediction accuracy for a model with seven bases. The inset histogram shows the error distribution, which has a 1% standard deviation

4 Optics and Sensor Irradiance

The simulation of imaging optics converts the scene radiance data into an irradiance image that represents the sum of the rays incident at a region on the sensor surface (photons/s/nm/m^2). Just as there are useful simplifications of the scene radiance, optics simulation can be carried out with varying degrees of complexity. Traditional modeling based on scene radiance representations separates optical factors into several terms: lens shading (vignetting), geometric distortions, and blur. More complex modeling can be applied to the light field representation [21].

4.1 Conversion of Units

The traditional camera equation converts radiance to irradiance for an ideal lens with a circular aperture [8, 28]. The equation is accurate in the center of the image (i.e., on the optical axis). T(λ) is the lens spectral transmissivity and f/# is the lens focal length divided by diameter of the lens aperture.

$$I(x, y, \lambda) = \frac{\pi T(\lambda)}{4 \times (f/\#)^2} L(x, y, \lambda)$$

The relative illumination declines with distance from the image center (lens vignetting). The relative illumination may differ between lenses, but for many purposes, a simple formula (cos-fourth law) relates the relative fall off to the visual field angle θ when using a thin lens.

$$R(\theta) = \cos^4(\theta)$$

The relative illumination can also be written as a formula relating distance to the lens, d and field height, $r = (x^2 + y^2)^{\frac{1}{2}}$.

$$R(S) = \left(\frac{d}{r}\right)^4$$

4.2 Geometric Distortion

The geometric distortion describes how the lens maps object coordinates, (x, y), into the sensor coordinates $(\hat{x}, \hat{y})$. The distortion can be calculated using lens design software, or it can be measured empirically using a printed or displayed grid. The distortion is typically radially symmetric, and its size depends on the lens settings such as aperture size and distance from the sensor. Using the radial symmetry, the polynomial is often represented as a polynomial function of field height, such as

$$\hat{x} = x + x(k_1 r + k_2 r^2)$$

where $\hat{x}$ is the real (distorted) position at field height r for a point whose ideal position is x. A similar equation is used for the y-dimension. If the parameters k_i are both zero, there is no distortion. Other geometric formulae have been proposed and analyzed, but this simple polynomial often does well [29–31] (Figure 6).

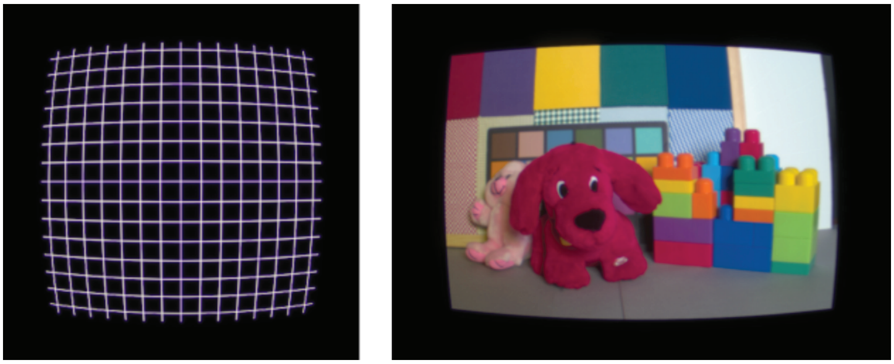

Figure 6 Geometric distortion. These images represent the irradiance at the sensor after the scene has passed through a lens (R54-852 Finite Conjugate Micro Video Imaging Lens). The geometric distortion and point spread functions were calculated from the lens prescription file using optical design software (Zemax). See [32]

4.3 Spatial Blur

The generalized point spread function of an optical system depends on field position, depth, and wavelength. Suppose the sensor coordinate system is represented as (u, v). The spread over the sensor surface from a scene point at (x, y, z) and wavelength λ can be expressed as the function $P(u, v \mid x, y, z, \lambda)$. The general point spread accounts for lens blurring as a function of field height, depth of field, and chromatic aberration. To calculate how the image is blurred using this point spread, knowledge of the image field height (x, y), the object distance at that field height, $z(x, y)$, and the spectral irradiance are required. The blurred irradiance is the sum of the point spread functions, weighted by the irradiance.

In general systems, the point spread function shape varies substantially with field height, depth, and wavelength (Figure 7). The extent of the blurring can depend on the lens setting. If all of the scene objects are far, or the depth of field of the optics is large (small aperture), the point spread functions are effectively independent of scene depth and can be expressed as $P(u, v \mid x, y, \lambda)$. This collection of point-spread functions captures field height dependent lens imperfections, diffraction, and chromatic aberration.

If the simulation is performed over a small field of view, the variation with field position may become negligible as well. In this case, the collection of point spread functions is simply $P(u, v \mid \lambda)$ and the blur is shift-invariant. An image region in which the point spread is shift-invariant is called *isoplanatic*. A shift-invariant simulation can be computed efficiently using the fast Fourier transform. Within a waveband, shift-invariance transformations are acceptable approximation for objects that are effectively far away, or if the calculation is confined to a paraxial region for objects with a narrow depth range.

An ideal (diffraction-limited) lens is a useful model for many calculations: such a model sets an upper bound on spatial acuity. An ideal lens with a circular aperture has a radially symmetric, shift-invariant point spread function whose spread depends only on the ratio of lens focal length and aperture (*f*/#), and the irradiance wavelength. There is a closed-form

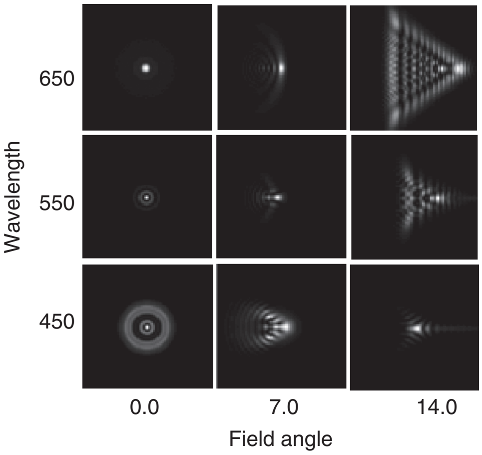

Figure 7 Point-spread functions vary with wavelength, field height, and depth. The point spread functions shown were calculated for the same lens used in Figure 6

solution for this wavelength-dependent point spread

$$P(v \mid \lambda) = I_0 \left(\frac{2J_1(v)}{v} \right)^2, v = \frac{\pi x}{\lambda f/\#}$$

(J_1 is a first-order Bessel function; I_0 is the irradiance at wavelength λ). The point spread of the ideal lens and circular aperture is called the *Airy pattern*; it comprises a bright central region that falls off to a minimum (dark ring), and followed by a sequence of alternating light and dark rings. For typical imaging systems, in which the spread of the light is small compared to the focal length of the lens, the radius of the first ring is $\sin(\theta) = 1.22\lambda/d$, or given that the angle is small simply $\theta = 1.22\ \lambda/d$, (θ is in radians, λ is the wavelength of the light, and d is the size of the aperture). The distance to the first minimum can also be written as $\Delta = 1.22 \times \lambda \times (f/\#)$. For example, given an ideal lens with an *f*/# similar to that of the human eye (5.6), and light in the middle of the visible wavelengths (550 nm), the radius of the first Airy ring is 3.76 microns.

4.4 Optics Operation

In many imaging systems, the optical parameters are adjusted in response to changes in the viewing condition. For example, many cameras implement autofocus algorithms to bring objects at a specific distance into sharp focus on the sensor surface. In addition, the user often has control of the size of the aperture, which determines both the amount of light at the sensor and the depth of field.

Autofocus algorithms differ in the area of the sensor image that is analyzed, the image statistic that is calculated, and the predetermined target value of the statistic. Autofocus algorithms are driven either by an error signal that is derived from a separate subsystem or data obtained through the lens (TTL) from the sensor itself. The contrast of an edge is a typical image statistic for focusing [33–36]. Software simulations can test autofocus algorithms under a range of conditions [37]. The algorithms tend to work better with good lenses and high light levels, with performance speed and accuracy falling off beyond a normal compliance range.

As an application, consider that image systems simulations can establish manufacturing tolerances on the placement of the sensor with respect to the lens aperture (Figure 8). To provide optimal focus for distant objects, the distance between the sensor and lens should equal the focal length. Using the image systems simulation, we can calculate the defocus [38] and render the image for relatively small misalignments.

4.5 Extended Optical Designs

Optical design is an active field, and there is considerable interest in designing systems that capture and analyze information about the light-field [23]. Systems are being designed with special pupil functions (coded apertures) that can be useful in making measurements such as object distance [40–42]. In addition, a number of groups have implemented systems with lenslet arrays that are placed behind the optics but in front of the sensor [24, 43]. This design enables measurement of the light-field. With this information, one can render images that simulate acquisitions using a range of pupil apertures and sensor positions. In this way, one can change the depth of field and focal plane after the image has been captured. Simulations based on light-field representations extend the traditional model of scene radiance to a more complex but richer set of possibilities.

5 Sensor

Image systems simulations use the concept of a generalized sensor: a software object that includes descriptions of several system components that are integrated with the silicon chip. The generalized sensor includes the IR blocking filter, diffuser, microlens array, CFA, pixel geometry, photodetector geometry, and sensor circuitry. Some of the sensor components, such as the pixel, are complex enough to be represented as independent objects that are a software class within the generalized sensor object.

The essence of the generalized sensor is a *phenomenological model* of how irradiance is converted to sensor outputs. To predict the sensor output, the simulation does not need to explicitly model the current flow at every transistor, the charge at every capacitor, or the material properties of every resistor. Attempts to incorporate this level of detail make the simulation impractical. A phenomenological model is a set of mathematical formula that predicts properties, such as light sensitivity, spatial and temporal integration, fixed and temporal noise, and circuit properties (e.g., quantization). The parameters of the model depend on system components that are closely coupled to the sensor – the CFA, anti-aliasing filter, and microlens array – although these components are not part of the chip itself. The generalized sensor object includes the image systems components whose properties are essential for an accurate phenomenological model.

Computations using the generalized sensor object can be separated into an estimate of the mean response followed by addition of the noise. The mean signal accounts for the scene radiance, passing through the optics, and the conversion to the image sensor response. The mean calculation can be computationally expensive and a clean copy is worth saving. The noise model is relatively inexpensive to compute. Separating the mean signal and noise calculations

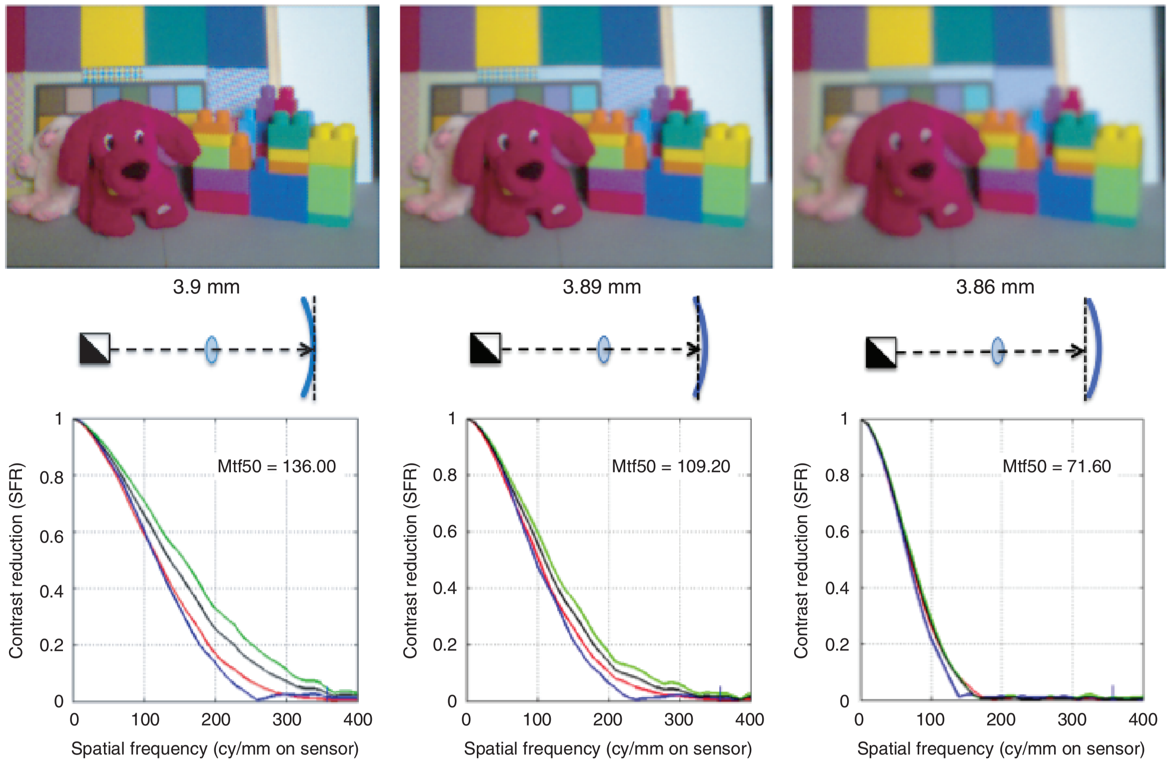

Figure 8 **Sensor misalignment**. For the best focus, the sensor must be positioned accurately behind the lens. This simulation illustrates the consequences of placing the sensor at two distances near the optimal position. The simulation is for a camera with a diffraction-limited lens with f-number of 2.4 and focal length of 3.9 mm. The pixel size is 1.4 microns, the pixel fill factor is 75%, and the sensors images were processed using bilinear demosaicking. The top leftmost image is the output of the system when the sensor is placed at the focal length distance of 3.9 mm. The middle and right images were produced when the sensor was placed 10 and 40 microns from the optimal position. The graphs show the modulation transfer functions calculated using the ISO 12233 method [39]. The inset text shows the spatial frequency on the sensor surface at which the image contrast is reduced by 50%

is useful for many applications in which one would like to analyze the signal to noise across multiple captures of a fixed scene (e.g., video frames).

5.1 Signal Transduction

The conversion from photons to electrons is linear over most of the operating range: the number of electrons scales with intensity and sums across wavelength. The mean response of the photodetector to an irradiance image ($I(x, y, \lambda)$, photons/s/nm/m^2) is determined by the aperture function across space, $A(x, y)$, exposure time (T, s), and photodetector spectral quantum efficiency ($S(\lambda)$, e$^-$/photon). Material properties of the silicon substrate, such as thickness and doping concentrations, influence the photodetector spectral efficiency [44, 45]. The number of electrons in a pixel can be calculated as the sum across the aperture and wavelength range

$$e = \int_{x,y}\int_{\lambda} S(\lambda)A(x, y)I(x, y, \lambda)\mathrm{d}x\mathrm{d}y\mathrm{d}\lambda$$

Deviations from this linear equation arise under very low light because of transistor thresholds and under very high light levels when the ability to store electrons is exceeded (well capacity).

5.2 Pixel Geometry

The first generation of CMOS imagers placed the photodetector in the silicon substrate, below the color filters and metal layers. In this geometry, the detector is at the bottom of a tube; the position and width of the opening to the tube determine the pixel aperture [46, 47], and the length of the tube can be as long as the pixel aperture. This geometry limits how efficiently photons find their way from the imaging lens to the detector, particularly for pixels at the edge of the sensor (pixel vignetting).

To compensate for the loss of sensitivity created by this arrangement, sensors include microlenses positioned above the color filters. For on-axis pixels, the microlens concentrates the light onto the portion of the substrate containing the detector. For pixels located at the edge of the sensor, the microlenses serve mainly to redirect the light from the lens so that more photons reach the photodetector. To appropriately redirect the rays, the position of the microlens depends on the location of the pixel within the sensor array. The microlens is directly over on-axis pixels, but it is substantially displaced for pixels at the edge of the sensor array.

The combination of the pixel geometry, materials, and microlens is sometimes called the *pixel optics* [47]. In systems simulation, these can be grouped together into a single phenomenological spectral efficiency function that varies from the center to the edge of the sensor array.

5.3 Sensor Noise

Noise factors are grouped into two types: temporal and fixed-pattern noise. Temporal noise fluctuates with each acquisition. One inescapable source of temporal noise is the Poisson distribution of incident photons (photon noise) and the corresponding noise in the sensor electrons (shot noise). Reading the stored electrons is a noisy process (read noise), and there is some variation when resetting the pixel to an initial state (reset noise).

Fixed-pattern noise refers to variations that are present in every acquisition. For example, the surface area of the individual pixels may vary across the sensor array. A consequence of this variation is that the signal gain will be greater in some pixels than in others. This unwanted random gain variations is fixed across time (photoresponse nonuniformity, PRNU). A second example of fixed pattern noise is a difference in the pixel reset circuitry. The circuitry may be more effective in some pixels than others, producing a reliable mean difference in the dark level voltage across the sensor array (dark signal nonuniformity, DSNU).

Other properties of the generalized sensor may contribute to the PRNU and DSNU, including variations in the placement of the microlens array, local current leakage, and so forth. The software summarizes the system performance using parameters that define the variance of the response gain and offset without specifying which component causes this variation. The phenomenological parameters can be estimated by experiments with an intact device. Experimental methods for estimating these quantities, and more details on implementing the calculations, are described elsewhere [8].

Figure 9 illustrates how photon noise and sensor noise (dark voltage, read noise, PRNU, and DSNU) affect the final image. It is impossible to diagnose the source of image noise by simply looking at single images. There has been great progress over the last 15 years in

(a) Photon noise, 20 cd/m^2

(b) Read noise (15 electrons), 80 cd/m^2

(c) PRNU (10%), 80 cd/m^2

(d) DSNU (2 mV), 80 cd/m^2

Figure 9 Sources of noise. Processed images from a sensor with the Bayer CFA, 1.7 micron pixel and 15 ms exposure duration. Photon noise is visible when the mean scene luminance is 20 cd/m^2 (a). Image noise due to read noise (b), PRNU (c), and DSNU (d) when the scene luminance is 80 cd/m^2 is also visible

(a) 1.2 micron pixel, 10 cd/m^2, 15 msec (b) 1.2 micron pixel, 100 cd/m^2, 15 msec

(c) 2.4 micron pixel, 10 cd/m^2, 15 msec (d) 2.4 micron pixel, 100 cd/m^2, 15 msec

Figure 10 **Photon noise**. The processed images shown in (a,b) were calculated using a simulated imaging sensor that has a 1.2-micron pixel and a 15-ms exposure duration. The images shown in (c,d) were calculated using a simulated imaging sensor that has a 2.4-micron pixel and a 15-ms exposure

reducing sensor noise; but photon noise will always be present and particularly visible at low light levels (Figure 10) [48, 49]. Image systems simulations allow designers to manipulate different sources of noise and visualize the impact on perceived image quality.

5.4 Global Wavelength Management: Lens and Infrared (IR) Cut Filters

Consumer photography seeks to reproduce a color image whose appearance resembles what the person saw at the time of acquisition. The first step in making an accurate color reproduction is to exclude irradiance wavelengths that are outside of the visible range. The optical glass in the camera lens typically reduces short wavelength (below 400 nm) transmission. A filter placed on the sensor surface reduces the long-wavelength and near IR transmission. This filter typically blocks wavelengths beginning at 650–680 nm. The combination of UV blocking and IR cutoff produces an irradiance image at the sensor surface that contains energy in the wavelength range that influences human vision.

5.5 Local Wavelength Management: Color Filter Array

Under moderate to high illumination levels, the human visual system encodes the scene using three types of light sensitive detectors (cone photoreceptors) that sample the spectral irradiance. To provide enough information to match the scene color appearance, the sensor must measure the same three-dimensional space of the spectral irradiance sampled by the human eye.

Sensors sample the spectral irradiance by placing an array of color filters over the pixel array. Classically, the CFA consists of three spectral types: a long-wavelength (red), short-wavelength (blue), and middle-wavelength (green). The most common spatial format is a repeating 2×2 pattern (super-pixel) that includes one red, one blue, and two green samples (Bayer array [50]). To reduce spatial aliasing, an optical low pass filter spreads a point of light across the super-pixel. This filter is placed in the optical path between the lens and sensor.

Although the Bayer CFA has been predominant, many other arrays have been suggested. The simulation environment should enable the use of arbitrary CFAs, including ones with much larger super-pixels and many different types of CFAs, IR cutoff filters, and lens transmission types.

5.6 Pixel Spatial Sampling

Most sensors transform the spectral irradiance image into a two-dimensional array of voltage samples, one sample from each pixel. Even so, the simulation software should allow for pixel sampling positions beyond the usual, two-dimensional sampling grid. For example, the Foveon sensor has three spectral samples at each pixel [51], the Fuji "Super CCD" sensor has a non-rectangular sampling arrangement [52]. Implementing the generalized sensor that enables arbitrary pixel-positions entails significant computational overhead; it is more efficient to create irregular representations by simulating several acquisitions and then selecting the pixel responses. For example, the Foveon sensor can be implemented as three sequential captures with different color filters; the Fuji sensor can be simulated by two captures with spatially displaced copies of the sensor.

5.7 Sensor Operation

Before image acquisition, software adjusts the sensor parameters so that the sensor data occupy a large portion of the sensor's range. The two most important sensor parameters are the integration time and the response gain.

Auto exposure (AE) algorithms determine the time required for the highest scene radiance value to produce a response near the sensor saturation level. This is desirable because high response levels have the best signal-to-noise ratio, and measurements that span the response range have the smallest quantization noise.

If the duration needed to fill up the response range is long, more than a couple of hundred milliseconds, the image content is likely to move and a handheld camera will shake [53–55]. To reduce these undesirable motion effects, the integration time may be shortened and the sensor gain increased. A gain adjustment does not improve the signal-to-noise ratio, but it does reduce the quantization noise. Adjustments of the sensor gain are referred to as changes in the camera speed, in analogy to film speed.

For a given scene radiance, the integration time and lens aperture combine to produce a given response level. The effects of these two camera parameters are often summarized as the exposure value (EV). The EV formula is usually expressed with respect to the relative aperture (*f*/#) and exposure time (*T*)

$$\mathrm{EV} = \log_2\left(\frac{(f/\#)^2}{T}\right)$$

This is equivalent to

$$\mathrm{EV} = 2\log_2(d) - (2\log_2(A) + \log_2(T))$$

where d is the focal length and A is the aperture diameter. Increasing the diameter or increasing the exposure time has the same effect.

In addition to modeling these traditional exposure methods, it is possible to simulate alternative exposure control algorithms, including exposure bracketing and digital pixel systems that repeatedly and nondestructively read the pixel over time [56, 57].

5.8 Novel Sensor Designs

There have been many interesting new developments in sensor technology [58, 59]. Many innovations were driven by the trend toward higher resolution and smaller pixel size. The decrease in pixel size reduces light sensitivity and signal-to-noise.

Back-illuminated sensors have achieved a significant improvement in light sensitivity. Recall that pixel vignetting, because of the presence of metal layers above the photodetector, reduces pixel sensitivity. In back-illuminated sensors, the metal layers are placed behind the photodiode [60]. Sensitivity can be improved by as much as 50% by flipping the silicon wafer during the manufacturing process, and then thinning the reverse side so that light is absorbed into the photodetector [61].

There also has been innovation in the design of color acquisition methods. Several manufacturers have produced sensors that create a high sensitivity pixel by replacing one of the two green pixels in the Bayer array with a clear (white) or relatively clear (emerald) filter. This increases the sensitivity and requires further innovations in the image processing. There have been advances in the color filter materials, including new types of materials based on quantum dots that create a much thinner filter [62, 63]. Spectral responsivity can be controlled by placing fine metal lines in the pixel [64, 65] and by applying voltages within the photodetector substrate [66, 67].

6 Image Processing

The image processor (IP) converts sensor data into an image that can be displayed or printed. The IP accomplishes two critical goals: spatial interpolation and color transformation. First, in most cases, the sensor data are incomplete because each pixel measures only one color channel; to display a color image, one must specify at least three values at each spatial location. The camera IP supplies the missing pixel values. Second, the IP must transform the camera sensor data into a calibrated color representation that can be used for accurate rendering on a display or printer. The color transform used in the IP must adapt to the illumination because the human visual system does so.

6.1 Interpolation

There are two reasons why sensor data must be interpolated. First, when producing a sensor with millions of pixels, some of them will fail (dead pixels). If a sensor has pixels in a cluster or line that fail, the part will be rejected. But if there are relatively few dead pixels, and they are at random and widely spaced locations, the missing data can be inferred from neighboring pixels. Most modern systems have a dead pixel replacement algorithm [68, 69].

The second reason why data are interpolated arises from the widespread use of CFAs. For example, the Bayer CFA [50] is based on a 2×2 super-pixel (RG/GB). The optics includes an anti-aliasing filter so that the irradiance varies little across the super-pixel. Hence, these data are adequate to represent three color channels at the spatial resolution of the super-pixel. Typically, however, manufacturers increase the spatial resolution by algorithms that interpolate the output from the super-pixel resolution to the single pixel resolution.

Spatial interpolation of the color channels is called *demosaicking*, and this component of image processing has attracted widespread interest [70]. Demosaicking algorithms draw on a diverse array of signal processing techniques, for example, inverse problems [71], neural networks [72], wavelets [73], Bayesian statistics [74, 75], and convex optimization [76]. The vast majority of demosaicking algorithms have been optimized for the Bayer CFA.

Two general demosaicking principles have emerged. First, there is a high degree of correlation between the nearby pixel responses across the color channels. This correlation is due in part to (i) image blurring, (ii) the spectral power distributions of natural image data, which tend to change smoothly across wavelength, and (iii) the overlap in the color channel responsivities. Second, the most successful demosaicking algorithms are adaptive, in that they interpolate using rules that identify image spatial structure [77–79].

Image systems simulations make it possible to evaluate algorithms under a wide range of different imaging conditions. This is valuable because demosaicking algorithm performance depends significantly on these conditions. For example, many demosaicking algorithms are designed to use edge information from the image. Yet, when the optical blur spans several pixels, these algorithms find very few sharp image edges. Conversely, under low light conditions, the image data contains a great deal of noise that is easily confused with image edges. In this case, it is important to protect the demosaicking algorithm from interpreting sensor noise as a high contrast edge. The conditions in which specific types of algorithms are helpful can be assessed through simulation.

6.2 Color Transformations

To render a color image properly, we must know the intended effect on the human observer. For example, one might like the rendered image to match the appearance of the scene that was captured. Or, one might wish to render a picture with higher saturation or contrast than the original scene. Whatever the intent, the IP must produce data that can be accurately rendered onto the display or print. The key technology for accurately representing the desired image is to produce output in a calibrated color space, such as display sRGB [80].

The spectral sensitivities of different cameras vary considerably. Even so, we are unaware of cameras whose sensor outputs are in a calibrated space, say within a linear transformation of the *XYZ* system (colorimetric). As camera sensors are not colorimetric, it is mathematically impossible to linearly transform all possible irradiance distributions into a calibrated color representation: there will be pairs of irradiance distributions that produce the same camera responses (camera metamers) and that have different *XYZ* values. Despite the limits of camera metamerism, the IP color transform is typically a linear function that makes a best effort to convert the camera color responses into a calibrated space.

The IP color transform must adapt to the image data because the human visual system adapts in response to changes in the ambient lighting conditions. As a first-order approximation, human adaptation preserves the color appearance of surfaces across lighting conditions. For example, a white shirt retains its appearance whether it is directly illuminated by the sun or indirectly illuminated by the blue sky. This visual adaptation is commonly referred to as *color constancy* [81]. The IP color transform is selected in the same way, in the sense that the transform is selected so that white surfaces are rendered as white in the final image.

For a conventional RGB camera, the IP color transform from sensor data to an output image is represented by a 3×3 linear transformation. There are many ways to select this transformation, but the general principles are clear and can be divided into two parts. An illustrative example is provided here.

Suppose that we want to produce rendered images that appear as if the surfaces are illuminated by a spectral power distribution, d_1. Represent the camera spectral responsivities in the columns of a matrix, C, and represent the CIE *XYZ* functions in the columns of a matrix H. Finally, create a list of surface reflectance functions that are considered important for rendering and place these in the columns of a matrix S.

If the image data are acquired under the desired illuminant, d_1, then the IP color transform is simply the 3×3 matrix, L_{d_1} that solves this linear equation

$$H' \operatorname{diag}(d_1) S = L_{d_1} C' \operatorname{diag}(d_1) S$$

We can find L_{d_1} using an inverse operator

$$L_{d_1} = (H' \operatorname{diag}(d_1)\, S)\, [C' \operatorname{diag}(d_1)\, S]^{-1}$$

(These linear equations are illustrated in Figure 11 as a matrix tableau.)

Now, suppose the data are acquired under a different illuminant, say d_2. We adjust the linear transformation to

$$L_{d_2} = (H' \operatorname{diag}(d_1)\, S)\, [C' \operatorname{diag}(d_2)\, S]^{-1}$$

There are various ways to implement the matrix inverse, such as the pseudo-inverse, or ridge regression. It is also possible to solve for the transform L using a search algorithm that minimizes the CIELAB prediction differences. Some engineers allow L to be a 3×3 matrix, others restrict the search to a diagonal transformation, D, and sometimes the IP is based on a fixed 3×3 transformation, F, followed by a diagonal, $L = \mathrm{DF}$.

The color transformation for different scene illuminants can be computed in advance and stored. Image systems software can be used to create these color transforms for a particular set of illuminants and selection of surfaces, and then to test how well the transformations perform under less common illuminant and surface conditions (Figure 12).

To apply the correct transformation, the IP must estimate the illuminant; several illuminant estimation algorithms are described in the literature [82–90]. Some of these algorithms are based on simple image statistics such as the mean RGB value or the ratio of the red and blue sensors. Others involve more elaborate Bayesian computations [85] or Retinex-style algorithms [91, 92].

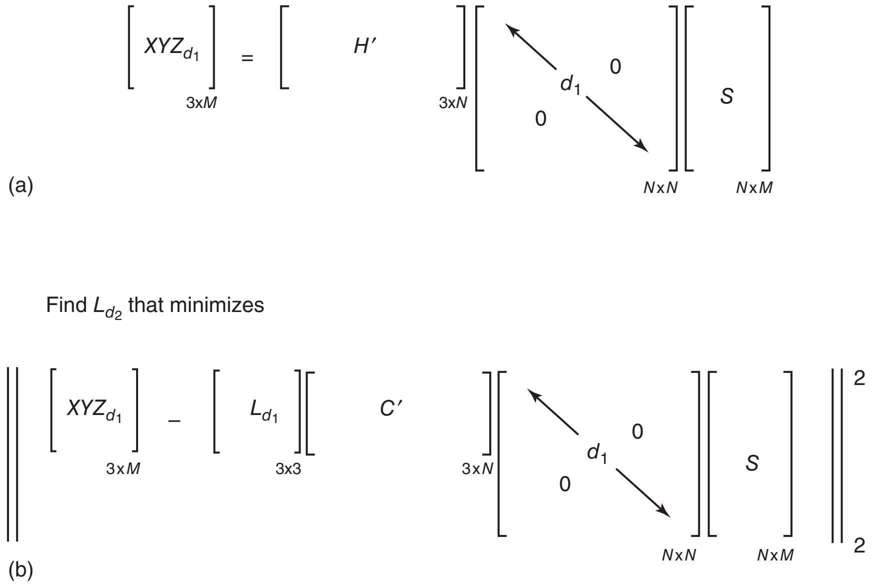

Figure 11 Linear formulation for determining the sensor correction. (a) The CIE *XYZ* values for a set of *M* surfaces under an illuminant are calculated as the product of the surface reflectance (*S*), the illuminant spectral power distribution (d_1), and the human color matching functions (*H*). (b) When the same surfaces are recorded by a camera, we replace the human color matching functions with the camera spectral quantum efficiency (*C*). We try to correct for the difference between *H* and *C* by applying a 3 × 3 transform, L_{d_1}, to the camera data. The same formulation is generalized to compensate for changes in the illuminant

6.3 Novel IP Technologies

Image processing is necessarily coupled with the characteristics of the optics and generalized sensor. Algorithms that are designed to optimize performance for the Bayer CFA dominate the literature. Advances in optics, sensors, and displays require new IP algorithms. These algorithms must account for the new color channels, the arrangement of the color mosaic, sensor noise, and the many different target displays.

Cameras have become increasingly tied to mobile phones, and the computational power of phones is increasing. There is now great interest in placing computer vision algorithms in the IP. For example, many cameras include automatic face identification algorithms [93], red-eye removal, and smile detectors [94]. Mobile phones can combine information from global positioning sensors and imaging systems to identify objects in a scene and search online image databases [95]. Online information can in turn modify how images are acquired.

IP algorithms increasingly interact with the image acquisition. For example, rather than acquiring a single long exposure, processors can increase sensor sensitivity and reduce motion blur by acquiring several shorter exposures, aligning them, and summing the results [96]. IP algorithms also have been extended to acquire an image pair: one with and one without the flash. The processor then uses data from the two images to reduce noise [97] or to estimate the

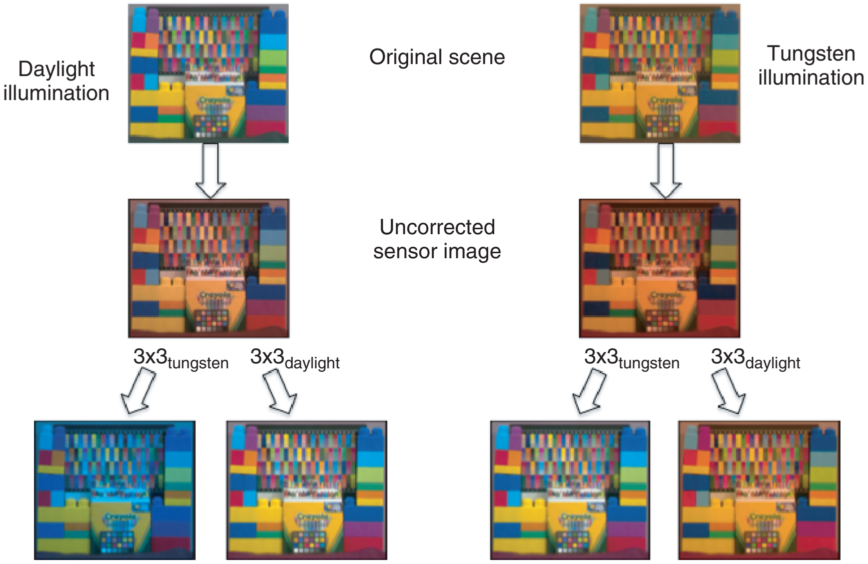

Figure 12 **Illuminant correction transforms**. The IP adaptively selects illuminant transforms that render the surfaces in a scene as if they were illuminated by a standard daylight. These transforms map the sensor values captured under one light into a calibrated color space representing the same surfaces illuminated by the standard daylight. To apply the correct color transformations, the IP must estimate the original scene illumination. The leftmost and rightmost images illustrate the consequences of applying the incorrect color transformation to the sensor image

illuminant [98]. A popular IP algorithm combines a stream of images into a single panorama [99].

As computer power and communications bandwidth increases, the entire notion of an image as a matrix is likely to be replaced by a more general concept: the image data will comprise multiple captures, and the image file will be combination of data and programs that offer the user multiple ways to render the data. In this case, the IP will be part of the image data and not restricted to the camera itself. Image systems simulation can play a very useful role by producing large numbers of test images and evaluating IP performance under a wide range of conditions.

7 Camera System Simulation

To this point, we emphasized how simulations are used to evaluate camera components [49, 100, 101]. Here we explain how image system simulation can be used to design an IP pipeline for a complete camera system. Simulation of the complete system is critical, because hardware and algorithms should be coupled together. For example, it should be possible to increase the dynamic range of imaging sensors by including a clear filter in the CFA. It should also be possible to improve the spectral accuracy of image sensors by increasing the number of different filters. To take advantage of these hardware modifications, new demosaicking, denoising, and color management algorithms are required.

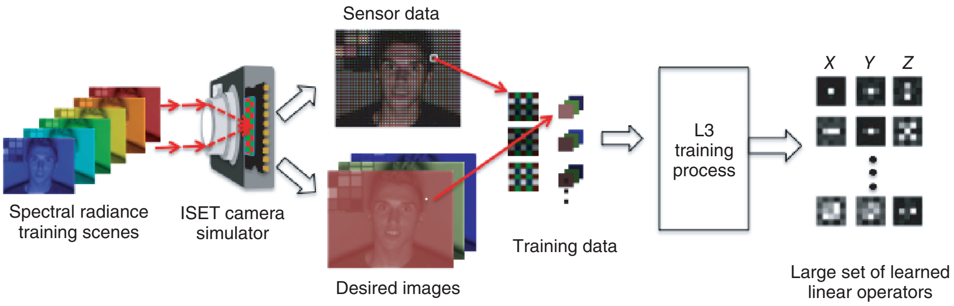

Figure 13 Combining image systems simulation and machine learning. The local, linear, and learned (L3) technology uses image systems simulations to calculate the scene *XYZ* values and the corresponding sensor responses (training data). Using simulations and a large number of training scenes, L3 learns the optimal linear transforms from the sensor responses to the scene *XYZ* for a number of different conditions, including the pixel type (R, G, B, or W), mean response level, and response variance of nearby pixels. These transforms are used to render new sensor data. The transforms comprise an image-processing pipeline (demosaicking, denoising, and color transforms) that is optimized for the simulated camera [103, 104]

As an example, consider designing a camera system that uses a CFA with four color filters: RGB and a clear (W) filter [102]. W-pixels will be more sensitive than RGB-pixels, so the camera will respond under low light levels. However, there is a design challenge: W-pixels saturate at moderate and high light levels, where the RGB pixels provide good information. Hence, to take advantage of this design, we need an adaptive IP algorithm that draws data from the proper pixels at the proper light level. The RGB data should dominate at high light levels, and the W-pixels should dominate at low light levels, and there must be a smooth transition between the illumination levels.

Figure 13 illustrates a method that relies on image systems simulations to create an IP pipeline for a camera with RGBW-pixels. The image system simulation produces responses to a large set of training scenes. Because these are simulated, the calibrated scene *XYZ* values are known. Machine learning is used to discover local linear operators that map the simulated camera responses to the correct scene *XYZ* values. The combination of image system simulation and machine learning is called *L3* (*Local, Linear, and Learned* [103, 104]). L3 calculates and stores optimized linear operator parameters that map camera responses to *XYZ* values for different classes of pixel (RGBW), light level (low to high), and local spatial patterns (smooth and textured). The linear operators, which are applied adaptively, depending on the local image data, combine demosaicking, denoising, and color transforms into a single computational step.

Figure 14 shows images generated using the L3 technology for imaging systems with various types of CFAs – Bayer RGBG, RGBW, and RGBN where N is a neutral density filter that reduces the sensitivity of a W pixel [105–107]. The simulations used a camera with an *f*/4, diffraction-limited lens, 3 mm focal length, 2.2 micron pixel, and 100 ms exposure duration (see [103] for the other simulation parameters). By using image system simulations, we can visualize the results under a wide range of viewing conditions. For example, Figure 14 shows that RGBW sensors have an advantage at low light levels (1 cd/m^2). At higher light levels, the three different CFA types have comparable image quality.

Figure 14 Using image systems simulation to compare camera designs. The L3 technology created an IP pipeline that optimized the performance of camera systems with three different types of CFAs: RGBG (left), RGBW (middle), and RGBN (right). The simulation compares the rendered images for scenes with a mean luminance of 1 cd/m^2 (top row) and 40 cd/m^2 (bottom row)

8 Summary

The design and manufacturing of an imaging system includes contributions from individuals with many different skills and who have the responsibility for selecting and integrating multiple system components. Using image system simulation, engineers can *visualize* how changes in individual components affect the final image. In addition, it is possible to *quantify* the effect that individual imaging components have on the performance of an imaging system. Image systems simulation software provides the engineering team with useful guidance and understanding of how the components will work together across a wide range of imaging conditions.

There remain many opportunities to expand simulations to incorporate more advanced methods for creating realistic spectral data that serve as scenes, advanced optical modeling, and new ideas for the sensor and image processing architectures. Validation using real devices with calibrated scenes is an important aspect of developing image systems simulations [8, 108, 109]. As the systems we simulate become increasingly complex, so too does the task of validation.

The next generation of image systems simulation will need to expand to simulate the effects of combining data from multiple types of sensors, including specialized components that measure depth, motion, and location. As so many modern imagers are integrated into mobile phones, it is likely that image processing will include components that search the Internet to help determine the final rendering [110]. Such a search might use information about the materials of objects (cars, faces, chairs, doors, walls, and buildings) to render the image. In this futuristic vision, the camera module will be part of a larger, networked system that draws on multiple sources of online information. This parallels the human visual system, which makes inferences based on prior knowledge [81].

As image systems evolve in this direction, the expertise of the different teams will broaden to the point where communication during the design process will have to be based on simulations of the output. The team will include software engineers who are expert at Internet database search, and hardware engineers who understand optics and sensor design. Image systems simulation software can provide an effective mechanism for such a diverse team to work together.

Acknowledgments

We thank many colleagues who contributed to our work, including David Brainard, Peter Catrysse, Steve Lansel, Andy Lai Lin, Patrick Maeda, Manu Parmar, Torbjorn Skauli, Qiyuan Tian, and Feng Xiao. We gratefully acknowledge the support from our industry partners, including Canon Research Labs, Hewlett-Packard, Microsoft Corporation, Olympus Corporation, and the Samsung Advanced Institute of Technology.

References

1. Huck, F.O. and Wall, S.D. (1976) Image quality prediction: an aid to the Viking Lander imaging investigation on Mars. *Journal of Applied Optics*, **15** (7), 1748–1766.
2. Huck, F.O., Jobson, D.J., Park, S.K., *et al.* (1977) Spectrophotometric color estimates of the Viking Lander sites. *Journal of Geophysical Research*, **82** (28), 4401–4411.
3. Cathey, W.T., Frieden, B.R., Rhodes, W.T., *et al.* (1984) Image gathering and processing for enhanced resolution. *Journal of the Optical Society of America A*, **1** (2), 241–250.
4. Huck, F.O. and Fales, C.L. (1985) Image gathering and processing: information and fidelity. *Journal of the Optical Society of America A*, **2** (10), 1644–1666.
5. Booth, J.M. and Schroeder, J.B. (1977) Design considerations for digital image processing systems. *Computer*, **10** (8), 15–20.
6. Kerekes, J.P. and Baum, J.E. (2005) Full-spectrum spectral imaging system analytical model. *IEEE Transactions on Geoscience and Remote Sensing*, **43** (3), 571–580.
7. Farrell, J., Xiao, F., Catrysse, P., *et al.* (2004) A simulation tool for evaluating digital camera image quality. *Proceedings of the SPIE*, **5294**, 124–131.
8. Farrell, J.E., Catrysse, P.B., and Wandell, B.A. (2012) Digital camera simulation. *Journal of Applied Optics*, **51** (4), A80–A90.
9. Gershun, A. (1936 Translated by P. Moon and G. Timoshenko in *Journal of Mathematics and Physics*) *The Light Field*, **18**, 55–151, 1939. pp. 51–151.
10. Farrell, J. and Wandell, B. Approximations to Spectral Representations of Natural Scenes from RGB Images, in preparation.
11. Skauli, T. and Farrell, J.E. (2013) A collection of hyperspectral images for imaging systems researchy. *Proceedings of the SPIE Electronic Imaging*.
12. Parmar, M., Imai, F., Park, S.H., *et al.* (2008) A database of high dynamic range visible and near-infrared multispectral images. *Proceedings of the SPIE*, **6817**, 68170N.
13. Parmar, M., Lansel, S., and Farrell, J.. (2012) An LED-based lighting system for acquiring multispectral scenes. *Proceedings of the SPIE*, **8299**, 82990P.
14. Parmar, M., Lansel, S., and Wandell, B. (2008) Spatio-spectral reconstruction of the multispectral datacube using sparse recovery. *Proceedings of the International Conference on Image Processing*.
15. Valero, E.M., Nieves, J.L., Nascimento, S.M.C., *et al.* (2007) Recovering spectral data from natural scenes with an RGB digital camera and color filters. *Color Research and Applications*, **32** (5), 352–360.

16. Foster, D.H., Nascimento, S.M.C., and Amano, K. (2004) Information limits on neural identification of coloured surfaces in natural scenes. *Visual Neuroscience*, **21**, 331–336.
17. Imai, F.H. and Berns, R.S. (1998) High-resolution multispectral image archives: a hybrid approach. *Proceedings of the Sixth Color Imaging Conference*, IST&SID: Scottsdale, Arizona, pp. 224–227.
18. Tominaga, S. (1996) Multichannel vision system for estimating surface and illumination functions. *Journal of the Optical Society of America A*, **13**, 2163–2173.
19. Marimont, D.H. and Wandell, B.A. (1992) Linear models of surface and illuminant spectra. *Journal of the Optical Society of America A*, **9** (11), 1905–1913.
20. Adelson, E.H. and Bergen, J.R. (1991) The plenoptic function and the elements of early vision, in *Computational Models of Image Processing* (ed M.L.a.J.A. Movshon), MIT Press, Cambridge, MA, pp. 3–20.
21. Liang, C., Shi, Y., and Chen, H.H. (2011) *Light field analysis for modeling image formation*. IEEE Transactions on Image Processing, **20** (2), 446–460.
22. Gortler, S., Grzeszczuk, R., Szeliski, R., *et al.* (1996) The lumigraph. *Computer Graphics*, **27**, 43–54.
23. Levoy, M. and Hanrahan, P. (1996) Light field rendering. *Computer Graphics*, **30**, 31–42.
24. Ng, R., Levoy, M., and Bredif, G., *et al.* (2005) Light field Photography with a Hand-Held Plenoptic Camera. Stanford University Computer Science Tech Report CSTR 2005-02.
25. Judd, D.B., MacAdam, D.L., and Wyszeski, G. (1964) Spectral distribution of typical daylight as a function of correlated color temperature. *Journal of the Optical Society of America*, **54**, 1031–1040.
26. Parkkinen, J.P.S., Hallikainen, J., and Jaaskelainen, T. (1989) Characteristic spectra of Munsell colors. *Journal of the Optical Society of America*, **6**, 318–322.
27. Cohen, J. (1964) Dependency of the spectral reflectance curves of the Munsell color chips. *Psychonomic Science*, **1**, 369–370.
28. Schott, J.R., and Barsi, J.A. (2003) Radiometry for Remote Sensing, in *Encyclopedia of Optical Engineering*: Volume 3 (ed R.G. Driggers), CRC Press, 2309–2318.
29. Zhang, Z. (1998) A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71, M. Research, Editor.
30. Zhang, Z. (2000) *A flexible new technique for camera calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **22** (11), 1330–1334.
31. Yoneyama, S., Kitagawa, A., Kitamura, K., *et al.* (2006) In-plane displacement measurement using digital image correlation with lens distortion correction. *JSME International Journal, Series A: Solid Mechanics and Materials Engineering*, **49** (3), 458–467.
32. Maeda, P., Catrysse, P., and Wandell, B.A. (2005) Integrating lens design with digital camera simulation. *SPIE, Electronic Imaging*, **5678** (5), 48–58.
33. Krotkov, E. (1987) Focusing. *International Journal of Computer Vision*, **1**, 223–237.
34. Kao, W.C., Wu, H.H., and Lin, S.Y. (2008) Reusable embedded software platform for versatile single-sensor digital cameras, in *Single-Sensor Imaging: Methods and Applications for Digital Cameras* (ed R. Lukac), CRC Press, Boca Raton.
35. Pillman, B.H. and Adams, J.E.J. (2014) Image quality in consumer digital cameras, in *Academic Press Library in Signal Processing: Volume 4: Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing* (eds R. Chellapa and S. Theodoridis), Elsevier pp. 11–76.
36. Subbarao, M., Choi, T., and Nikzad, A. (1993) Focusing techniques. *Journal of Optical Engineering*, **32** (11), 2824–2836.
37. Tsai, D.-C., Tsai, Z.-M., and Chen, H.H. (2013) A simulation tool for digital autofocus design. *Proceedings of the 2013 IEEE International Conference on Consumer Electronics*.
38. Marimont, D.L. and Wandell, B.A. (1994) Matching color images: the effects of axial chromatic aberration. *Journal of the Optical Society of America*, **11** (12), 1–11.
39. Williams, D. (1988) Benchmarking of the ISO 12233 Slanted-Edge Spatial Frequency Response (SFR) plug-in. *Proceedings of the 51st IS&T PICS Conference*, pp. 133–136.
40. Levin, A., Fergus, R., Durand, F., *et al.* (2007) Image and depth from a conventional camera with a coded aperture. *Jorunal of the ACM Transactions on Graphics*, **26** (3Article No. 70).
41. Veeraraghavan, A., Raskar, R., Agrawal, A.K., *et al.* (2007) Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics*, **26** (3), 69.
42. Zhou, C., Lin, S., and Nayar, S.K. (2011) Coded aperture pairs for depth from defocus and defocus deblurring. *International Journal of Computer Vision*, **93** (1), 53–72.
43. Venkataraman, K., Lelescu, D., Duparré, J., *et al.* (2013) PiCam: an ultra-thin high performance monolithic camera array. *ACM Transactions on Graphics*, **32** (6), 1–13.

44. Lee, J.S. (2003) Photoresponse of CMOS image sensors, in *Electrical and Computer Engineering*, University of Waterloo, Waterloo.
45. Lee, J.S., Hornsey, R.I., and Renshaw, D. (2003) *Analysis of CMOS photodiodes – part 1: quantum efficiency*. IEEE Transactions on Electron Devices, **50** (5), 1233–1238.
46. Catrysse, P.B. and Wandell, B.A. (2002) Optical efficiency of image sensor pixels. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, **19** (8), 1610–1620.
47. Catrysse, P.B. Imaging optics, in *Handbook of Digital Imaging*. in press.
48. Xiao, F., Farrell, J., and Wandell, B. (2005) Psychophysical thresholds and digital camera sensitivity: the thousand photon limit. *Proceedings of the SPIE 2005 Conference on Electronic Imaging*.
49. Jiajing, X., Bowen, R., Wang, J., *et al.* (2010) Visibility of uncorrelated image noise. *Proceedings of SPIE.*, **7537**.
50. Bayer, B.E. (1976) Color imaging array. Eastman Kodak Company.
51. Merrill, R.B. (1999) Color separation in an active pixel cell imaging array using a triple-well structure. US Patent US 5965875 A
52. Wikipedia (2014) Super CCD. Wikipedia [cited 2014], http://www.dpreview.com/news/1999/10/20/superccd (accessed 17 January 2014).
53. Xiao, F., Fincenti, J., John, G., and Johnson, K. (2007) Camera-motion and mobile imaging. *Proceedings of the SPIE*, **6502**, 650204.
54. Xiao, F., Silverstein, A., and Farrell, J. (2006) Camera motion and effective spatial resolution. *Proceedings of the International Congress of Imaging Science*.
55. Xiao, F., Farrell, J.E., Catrysse, B., and Wandell, B.A. (2009) *Proceedings of the SPIE, 7250, 72500K*.
56. Yang, D., El Gamal, A., Fowler, B., *et al.* (1999) *A 640x512 cmos image sensor with ultrawide dynamic range floating-point pixel-level ADC. IEEE Journal of Solid-State Circuits*, **34** (12), 1821–1834.
57. Yang, D., Fowler, B., and El Gamal, A. (1999) A nyquist rate pixel level ADC for CMOS image sensors. *IEEE Journal of Solid State Circuits*, **34** (7), 348–556.
58. Fossum, E.R. (2011) The Quanta Image Sensor (QIS): Concepts and Challenges. in Imaging and Applied Optics, OSA Technical Digest (CD) Optical Society of America, Toronto, Canada, paper JTuE1. http://www.opticsinfobase.org/abstract.cfm?URI=COSI-2011-JTuE1
59. Ma, J., Masoodian, S., Song, Y. *et al*. (2013) Quanta Image Sensor (QIS): Early Research Progress, in Imaging and Applied Optics, OSA Technical Digest (online) (Optical Society of America), paper JW3B.2. http://www.opticsinfobase.org/abstract.cfm?URI=ISA-2013-JW3B.2
60. Hideo, Y. (2006) Method and apparatus for producing ultra-thin semiconductor chip and method and apparatus for producing ultra-thin back-illuminated solid-state image pickup device. Sony Corp. US Patent 7521335 B2
61. Swain, P.K. and Cheskis, D. (2008) Back-Illuminated Image Sensors Come to the Forefront – Novel Materials and Fabrication Methods Increase Quality and Lower Cost of Sensors for Machine Vision and Industrial Imaging, Photonics Spectra, August 2008. http://www.photonics.com/Article.aspx?AID=34685
62. Sargent, E.H., Clifford, J., Konstantatos, G. *et al*. (2005) Quantum dot optical devices with enhanced gain and sensitivity and methods of making same. US patent US 8102693 B2
63. Koleilat, G.I., Levina, L., Shukla, H., *et al.* (2008) Efficient, stable infrared photovoltaics based on solution-cast colloidal quantum dots. *ACS Nano*, **2** (5), 833–840.
64. Catrysse, P., Wandell, B., and El Gamal, A. (2001) An Integrated Color Pixel in 0.18 mu m CMOS Technology. Technical Digest International Electron Devices Meeting, pp. 24.4.1–24.4.4.
65. Catrysse, P.B. and Wandell, B.A. (2003) *Integrated color pixels in 0.18-um complementary metal oxide semiconductor technology*. *Journal of the Optical Society of America A*, **20** (12), 2293–2306.
66. Langfelder, G. (2012) Spectrally reconfigurable pixels for dual-color-mode imaging sensors. *Journal of Applied Optics*, **51** (4), 91–98.
67. Langfelder, G., Zaraga, F., and Longoni, A. (2009) Tunable spectral responses in a color-sensitive CMOS pixel for imaging applications. *IEEE Transactions on Electron Devices*, **56** (11), 2563–2569.
68. Baharav, I., Kakarala, R., Zhang, X., and Vook, W. (2003) Bad pixel detection and correction in an image sensing device. US Patent US 6737625 B2
69. Stanback, J.H., Zhang, X., Kakarala, R., and Ying, P. (2006) System and method for detecting and correcting defective pixels in a digital image sensor. US Patent US 7460688 B2.
70. Glotzbach, J., Altunbasak, Y., Schafer, R.W., and Mersereau, R.M. (2005) Demosaicking: color filter array interpolation. *IEEE Signal Processing Magazine*, **22** (1), 44–54.
71. Taubman, D. (2000) Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior. *Proceedings of the IEEE International Conference on Image Processing*, **3**, 801–804.

72. Kapah, O. and Hel-Or, H.Z. (2000) Demosaicking using artificial neural networks. *Proceedings of the SPIE*, **3962**, 112–120.
73. Hel-Or, Y. and Keren, D. (2002) Demosaicing of Color Images using Steerable Wavelets. HP Labs Technical Report HPL-2002-206R1.
74. Brainard, D.H. (1994) Bayesian method for reconstructing color images from trichromatic samples. *Proceedings of the IS&T 47th Annual Meeting*, Rochester, NY, pp. 375–380.
75. Mukherjee, J., Parthasarathi, R., and Goyal, S. (2001) Markov random field processing for color demosaicing. *Pattern Recogniton Letters*, **22**, 339–351.
76. Gunturk, B.K., Altunbasak, Y., and Mersereau, R.M. (2002) Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, **11** (9), 997–1013.
77. Buades, A., Coll, B., and Morel, J. (2005) A non-local algorithm for image denoising. *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, 60–65.
78. Buades, A., Coll, B., Morel, J.-M., *et al.* (2009) *Self-similarity driven color demosaicking*. IEEE Transactions on Image Processing, **18** (6), 1192–1202.
79. Mairal, J., *et al*. (2009) Non-local Sparse Models for Image Restoration. IEEE 12th International Conference on Computer Vision (ICCV), pp. 2272–2279.
80. Stokes, M., Anderson, M., Chandrasekar, S., and Motta, R. (1996) *Standard Default Color Space for the Internet – sRGB*, http://www.w3.org/Graphics/Color/sRGB.html (accessed 22 June 2014).
81. Wandell, B.A. (1995) Foundations of vision. Sinauer Associates, Inc.
82. Tominaga, S. and Wandell, B. (1989) Standard surface-reflectance model and illuminant estimation. *Journal of the Optical Society of America A*, **6** (4), 576–584.
83. Tominaga, S. and Wandell, B.A. (2002) Natural scene-illuminant estimation using the sensor correlation. *Proceedings of the IEEE*, **90** (1), 42–56.
84. Tominaga, S., Ebisui, S., and Wandell, B.A. (2001) Scene illuminant classification: brighter is better. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, **18** (1), 55–64.
85. Brainard, D.H. and Freeman, W.T. (1997) Bayesian color constancy. *Journal of the Optical Society of America A*, **14** (7), 1393–1411.
86. DiCarlo, J., Catrysse, P., Xiao, F., and Wandell, B. (2002) System and method for estimating physical properties of objects and illuminants in a scene using modulated light emission. US Patent US 20020171842 A1
87. DiCarlo, J.M. and Wandell, B.A. (2000) Illuminant Estimation: Beyond the Bases. Eighth Color Imaging Conference: Color Science, Systems, and Applications, pp. 91–96.
88. Finlayson, G., Hordley, S.D., and Hubel, P.M. (2001) *Color by correlation: A simple, unifying framework for color constancy*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **23** (11), 1209–1221.
89. Maloney, L.T. and Wandell, B.A. (1986) Color constancy: a method for recovering surface spectral reflectance. *Journal of the Optical Society of America A*, **3** (1), 29–33.
90. Lee, H.-C. (1986) Method for computing the scene-illuminant chromaticity from specular highlights. *Journal of the Optical Society of America A*, **3**, 1694–1699.
91. Brainard, D.H. and Wandell, B.A. (1986) Analysis of the retinex theory of color vision. *Journal Optical Society of America A*, **3** (10), 1651–1661.
92. Kimmel, R., Elad, M., Shaked, D., *et al.* (2003) A variational framework for retinex. *Journal of Computer Vision*, **52**, 7–23.
93. Yang, M.H., Kriegman, D.J., and Ahuja, N. (2002) *Detecting faces in images: a survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **24** (1), 34–58.
94. Whitehill, J., Littlewort, G., Fasel, I., *et al.* (2009) *Toward practical smile detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **31** (11), 2106–2111.
95. Girod, B., Chandrasekhar, V., Chen, D.M., *et al.* (2011) *Mobile visual search: linking the virtual and physical worlds*. IEEE Signal Processing Magazine, **61**, 61–76.
96. Debevec, P. and Malik, J. (1997) Recovering High Dynamic Range Radiance Maps from Photographs. SIGGRAPH.
97. Petschnigg, G., Szeliski, R., Agrawala, M., *et al.* (2005) Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics (TOG)*, **23** (3), 664–672.
98. DiCarlo, J.M., Xiao, F., and Wandell, B.A. (2001) Illuminating illumination. *Proceedings of the Ninth Coloring Imaging Conference*, IS&T: Springfield, VA, pp. 27–34.
99. Szeliski, R. and Shum, H.Y. (1997) Creating full view panoramic image mosaics and environment maps. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., pp. 251–258.

100. Farrell, J., Okincha, M., Parmar, M., *et al.* (2010) Using visible SNR (vSNR) to compare the image quality of pixel binning and digital resizing. *Proceedings of SPIE*, **7537**, 75370C.
101. Farrell, J., Xiao, F., and Kavusi, S. (2006) Resolution and light sensitivity tradeoff with pixel size. *Proceedings of the SPIE*, **6069**.
102. Parmar, M. and Wandell, B.A. (2009) *Interleaved imaging: an imaging system design inspired by rod-cone vision*. SPIE/IS&T Conference on Electronic Imaging, **7250**.
103. Tian, Q., Lansel, S., Farrell, J.E. *et al*. (2014) Automating the design of image processing pipelines for novel color filter arrays: local, linear, learned (L3) method. *Proceedings of the SPIE, 9023, 90230K*
104. Lansel, S.P. (2011) Local Linear Learned Method for Image and Reflectance Estimation, PhD thesis, Stanford University, Stanford, CA.
105. Kijima, T., Nakamura, H., Compton, J.T., *et al*. (2011) Image sensor with improved light sensitivity. Eastman Kodak Company, US Patent 7916363 B2.
106. Gindele, E.B. and Gallagher, A.C. (2002) Sparsely sampled image sensing device with color and luminance photosites, US Patent US 6476865 B1.
107. Yamagami, T., Sasaki, T., and Suga, A. (1994) Image signal processing apparatus having a color filter with offset luminance filter elements, US Patent 5323233 A.
108. Farrell, J., Okincha, M., and Parmar, M. (2008) Sensor calibration and simulation. *Proceedings of the SPIE*, **6817**.
109. Chen, J., Venkataraman, K., Bakin, D., *et al.* (2009) *Digital camera imaging system simulation*. IEEE Transactions on Electron Devices, **56** (11), 2496–2505.
110. Zhang, C., Gao, J., Wang, O., *et al.* (2014) *Personal photo enhancement using internet photo collections*. IEEE Transactions on Visualization and Computer Graphics, **20** (2), 262–275.