

Digital Camera Image Formation: Processing and Storage

Aaron Deever, Mrityunjay Kumar and Bruce Pillman

Abstract This chapter presents a high-level overview of image formation in a digital camera, highlighting aspects of potential interest in forensic applications. The discussion here focuses on image processing, especially processing steps related to concealing artifacts caused by camera hardware or that tend to create artifacts themselves. Image storage format issues are also discussed.

1 Introduction

The hardware of a digital camera was discussed in the previous chapter. This chapter describes image processing operations used with digital cameras. Each operation can introduce characteristic artifacts under some conditions. The purpose of this chapter is to mention them so researchers can expect and recognize them when found.

These processing operations can take place either in a camera or on a computer, depending upon the chosen camera and workflow. Most processing steps are largely the same whether performed in a camera or on a computer, although increased complexity is often allowed when processing on a computer, rather than in a portable camera. The discussion here will describe the processing chain step by step, mentioning when computer processing is more likely to differ from processing in a camera.

Most of the discussion in this chapter focuses upon routine processing performed on most images from digital cameras. Naturally, more complex processing is occa-

A. Deever (✉) · M. Kumar · B. Pillman
Corporate Research and Engineering, Eastman Kodak Company,
Rochester, NY, USA
e-mail: aaron.deever@kodak.com

M. Kumar
e-mail: mrityunjay.kumar@kodak.com

B. Pillman
e-mail: bruce.pillman@kodak.com

sionally used for some images and applications. These applications and processing techniques will be discussed in Sect. 4.

Images from digital cameras are stored in files, normally in one of several standard formats. The information included in the file can affect image use, especially in forensic applications, so image storage formats are briefly discussed in Sect. 3.

Finally, Sect. 5 discusses some of the characteristics of processing chains for video capture.

2 Nominal Image Processing Chain

To provide an order for the discussion of routine camera image processing, the processing steps are presented in the sequence shown in Fig. 1. The ordering shown in the figure is reasonable, although in practice steps are often moved, combined, or split up and applied in several locations in the chain. In Fig. 1, ellipses represent the image at key points of interest along the chain, while rectangles represent processing blocks. The blocks in this chain will be discussed in the following sections.

One reason for split or redundant operations is the tendency for noise to be amplified during the processing chain. White balancing, color correction, tone and gamma correction, and edge enhancement are all operations that tend to increase noise or at least increase visibility of the noise. Because noise is amplified during processing, it is common for several operations to take steps to reduce noise or at least limit its amplification. Also, the chain is often altered to meet different expectations for image quality, performance, and cost. For example, color correction usually works more accurately when performed on linear data, but lower cost image chains will often apply gamma correction fairly early in the processing chain to reduce bit depth, and apply color correction on the gamma-corrected data.

In general, image processing chain design is fairly complex, with tradeoffs in the use of cache, buffer memory, computing operations, image quality, and flexibility. This chapter will discuss some of the more common processing chain operations, but the reader is advised to consult [2] for further discussion of the design of camera processing chains.

2.1 Raw CFA Image

The first image is the raw image as read from the sensor through the analog signal processing chain. It is a single channel image in which different pixels sense different colors through a color filter array (CFA), as discussed in the previous chapter.

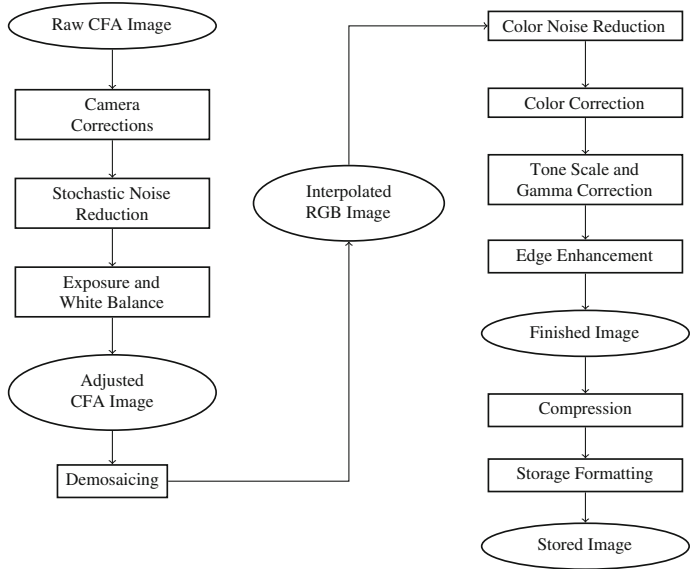


Fig. 1 Nominal flow for a digital camera processing chain

2.2 Camera Corrections

Most of the processing blocks in the nominal processing chain are much simpler to implement if the incoming image has a known response to light exposure, a known offset for the dark signal level, and has no camera-induced artifacts. These ideal conditions are essentially never met with real hardware. At the very least, sensors essentially always have defective pixels and have a dark signal level that varies somewhat with integration time and temperature. Often, other artifacts are also present and require correction or concealment.

The first processing block in the chain of Fig. 1 is more precisely a collection of blocks, illustrated in Fig. 2, designed to convert the acquired raw CFA image into a more idealized “raw” image. The processing blocks required for a specific camera vary depending upon the hardware and the user expectations. Lower cost hardware typically leaves more artifacts in the raw image to be corrected, but the user expectations are often lower as well, so the choice of correction blocks used with a particular camera is the result of a number of system engineering and budget decisions. Few, if any, cameras use all of the processing blocks shown in Fig. 2. In some cases, users save images to a raw capture format and use sophisticated desktop software for processing, enabling a degree of control over the processing chain not usually exercised by the casual user.

While these blocks are presented in a specific order in this discussion, the ordering chosen for a specific camera is dependent upon the causes of the artifacts needing correction and interactions between the effects. Usually, the preferred order of cor-

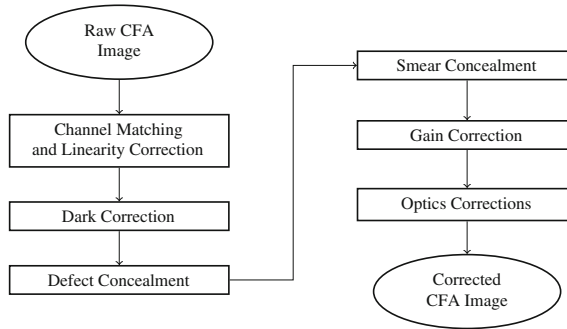


Fig. 2 Typical flow for digital camera corrections

rection blocks is roughly the inverse of the order in which the artifacts are caused. For example, gain artifacts are mostly caused by optics and interactions between the taking lens and the sensor, while linearity problems usually occur primarily in the analog signal processing chain after collection of charge in the pixels. Therefore, linearity correction would typically be applied to the image before gain correction is applied. Each of these correction blocks is complicated by the artifacts that have not yet been corrected. For example, if a dark correction is computed before defect concealment is completed, care should be taken to avoid using defective pixels in calculation of statistics for dark correction.

2.2.1 Channel Matching and Linearity Correction

The first correction discussed here is to match the response of multiple outputs or analog signal processing chains, such as with the dual output sensor shown in Fig. 3. Because the artifacts due to channel mismatch are highly structured, usually a seam in the middle of the image or a periodic column pattern, the responses for the multiple outputs must match very closely. The most common form of this correction is to adaptively compute a dark offset correction for each output that will bring similar pixels from each output to match a common value, using reference dark pixels. More complex algorithms involve sampling pixels from the image area in order to perform gain or linearity matching as well as dark level matching. This requires either controlled capture of calibration images or the ability to estimate channel mismatches in the presence of scene content variation [68]. The key to successful matching of multiple output channels is to take advantage of the knowledge of which image pixels came from which output.

2.2.2 Dark Correction

Dark correction is always necessary, since the analog output from the image sensor is rarely precisely “zero” for a zero light condition. As mentioned in the previous

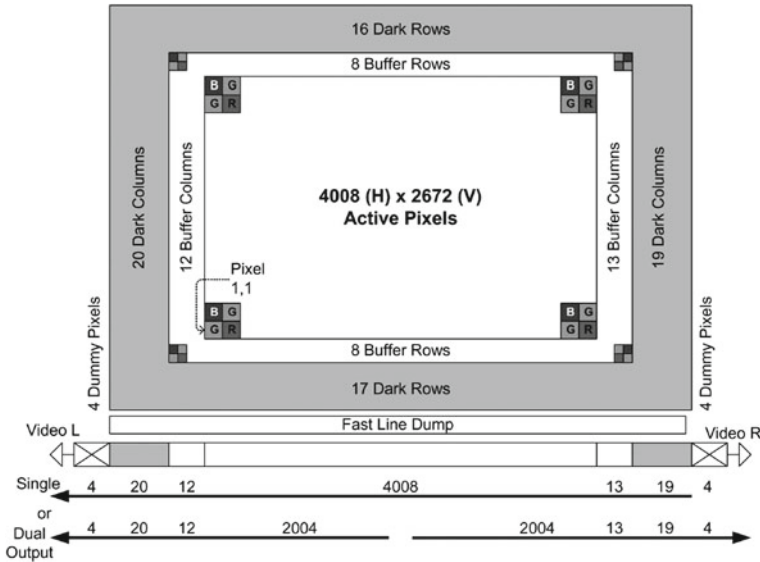


Fig. 3 Example layout for a multiple output interline CCD

chapter, dark current is collected within pixels as well as light-induced charge. In addition, slight drifts in analog offsets mean the dark level for an image will usually drift by a small percentage of the full-scale signal. The nonlinearity of human perception and image tone processing means this kind of drift causes fairly obvious changes in the shadows of an image. This is illustrated in Fig. 4, showing a plot of the conversion from linear relative exposure to CIE L^* , a standard perceptually even measure of lightness [18]. As shown in Fig. 4, the slope increases substantially in the shadows, below a midtone gray near 50 L^* . To examine slope in the shadows more closely, Fig. 5 plots the change in L^* due to a change in relative exposure of 0.01, plotting for a range from 0 to 0.2. A one-unit change in CIE L^* is approximately one just-noticeable difference (JND). As shown in the diagram, a 0.01 change in exposure produces a much larger change in L^* as relative exposure goes toward zero. Normal scene shadow exposures range from roughly 0.03 (corresponding to a typical diffuse black patch on a test chart) to 0.2 (corresponding to a midtone gray). In the figure, the dash-dotted line connects the black patch exposure of 0.03 with the slope at that exposure, roughly four ΔL^* , or JNDs.

Because the dark level must be controlled more precisely than low-cost analog electronics can provide, the analog dark level is chosen to be several percentage points into the range of the A/D converter, followed by a digital dark floor subtraction. If the dark floor of the image is uniform enough, dark subtraction is simply the subtraction of a global value from the image, usually based upon simple statistics from the light-shielded dark pixels during a capture.

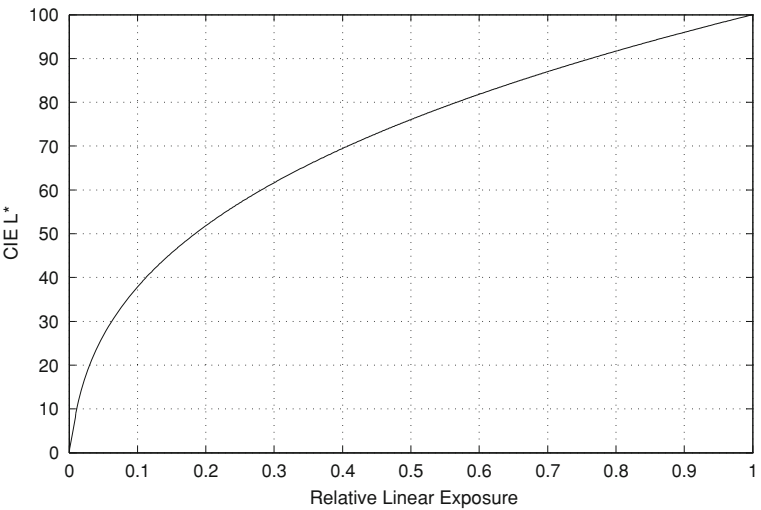


Fig. 4 CIE L* versus relative exposure

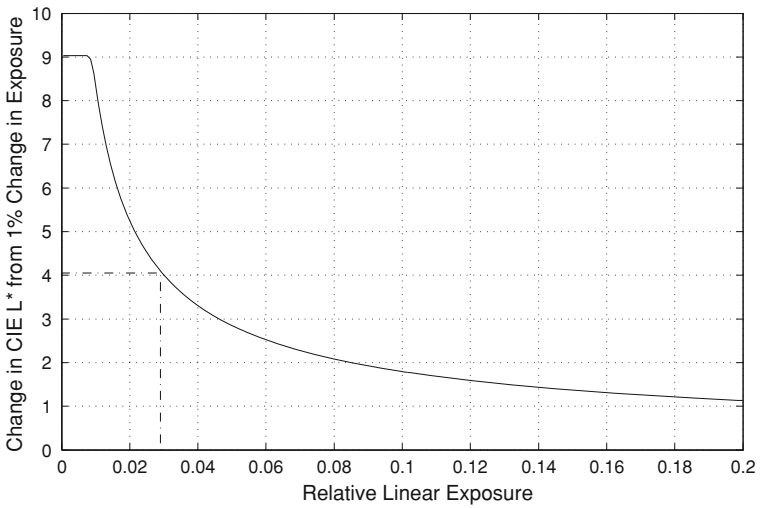


Fig. 5 Change in CIE L* due to a 0.01 change in relative exposure

If the dark floor of the sensor is not uniform enough, then a full dark floor image is subtracted from a captured image to remove the fixed pattern. One extremely simple approach is to capture a second image immediately after capturing the scene image. The second image is captured with no exposure, providing a full image of dark pixels. This works most accurately if an optical shutter is closed and the integration time for the dark capture matches the integration time for the scene capture. This approach

reduces fixed patterns in the dark floor of the image, but increases the random noise. Because of this, the dark capture technique is most often used with long integration times (such as 1/2 s or more), when the appearance of noise arising from the dark fixed pattern is clearly greater than the noise increase from the frame subtraction.

A second approach is to model the dark floor fixed patterns. Most often, this is done using light-shielded dark pixels from the border around the sensor. One common approach is to create a dark floor using dark pixels from the left and right of the image sensor, estimating a dark floor value for each row from the dark pixels for that row. This value can be corrupted by noise in the dark pixels, so some smoothing may be used to reduce the dark floor estimation error. Depending upon the sensor capabilities and the camera architecture, dark rows above and below the image may also be used. Dark pixels on the left, right, top, and bottom of the image provide enough data for a complete separable dark floor model. This one- or two-dimensional approach is very fast and is especially effective at correcting row and column patterns in a CMOS sensor.

In some cases, the dark floor is modeled using data from multiple dark captures. By averaging multiple dark captures, the impact of temporal noise on the dark floor estimate is minimized. This technique is still affected by changes in sensor temperature and integration time. Although changes in dark current caused by changes in temperature and integration time are well understood, other factors affecting the dark level may not be modeled as simply. This approach is quite rare in portable color cameras. Astronomical and other scientific applications, especially ones using a temperature-controlled sensor, routinely use this technique, made easier by the controlled temperature.

2.2.3 Defect Concealment

Sections 2.2.1 and 2.2.2 both referred to correction, because those artifacts can be essentially removed with no significant loss of data. Sensor defects are somewhat problematic, since they indicate lost data that simply was not sensed. Algorithms for treating defects interpolate the missing data. This process is referred to here as concealment rather than correction, to emphasize that it can only interpolate the missing data.

As mentioned in the previous chapter, the most common defects are isolated single pixel defects. Concealment of isolated pixels is usually done with a linear interpolation from the nearest adjacent pixels of the same color sensitivity. Some cameras at a low enough price point treat these defects with an impulse noise filter rather than maintaining a map of defective pixels. This tends to (inappropriately) filter out high-contrast details such as stars, lights, or specular reflections. With low-cost optics, the taking lens may spread fine detail from the scene over enough pixels to reduce confusion between scene detail and sensor noise.

Bright pixel defects caused by cosmic ray damage must be concealed without depending upon a map from the sensor or camera manufacturer. If at a low enough price point, the impulse removal approach works well. For higher user expectations,

a camera can implement a dark image capture and bright defect detection scan in firmware, usually done at startup. New defects found in the dark image are added to the defect map. Because cosmic ray damage tends to produce bright points rather than marginal defects, detecting these defects is relatively easy.

Sensor column defects present a much greater concealment challenge, because the human visual system is extremely sensitive to correlated features such as lines. Concealment of defective columns has been approached as an extended CFA interpolation problem with some success [34]. As with single pixel defects, a broader taking lens point spread function (PSF) will prevent the highest frequency detail from being imaged on the sensor, making defect concealment easier. The general goal for column concealment has been to ensure that concealment artifacts are subtle enough to be masked by noise or scene content. Concealment algorithms may produce columns with slightly different noise characteristics even if the mean value is well estimated. Concealment of defects including two or more adjacent columns is much more challenging and is an area of current research, although it has been approached with some success [33].

As mentioned previously, some sensor flaws create a defect in several adjacent pixels, here termed a cluster defect. The difficulty of concealing one of these defects increases dramatically with the size of the defect. Methods usually involve filling in from the boundary of the defect with a variety of adaptive approaches. The advantage is that these defects are rare enough that complex processing for the defect concealment is relatively insignificant compared to the processing time for the rest of the image.

Dirt on the cover glass of the sensor creates a much more complex defect, since it varies in size depending upon the f /number of the lens and the distance from the exit pupil to the sensor. Concealment of these defects must take the variable size into account, usually using a combination of gain correction and pixel interpolation.

2.2.4 Smear Correction

Interline smear is a challenging artifact to correct or conceal because the artifacts vary with scene content. It is manifested as an offset added to some of the columns in the captured image. Since the added signal will usually vary from column to column, the effect will vary with the original scene content. If a small amount of charge is added to pixels that are well below saturation, the artifact is manifested as a column that is brighter and lower in contrast than normal. If the sum of scene charge and smear charge saturates the pixels in the column, then the column looks like a bright defective column. Smear usually affects several adjacent columns, so saturated columns become difficult to conceal well. Concealment approaches start with the use of dark rows or overclocked rows to estimate the smear signal that should be subtracted from each column. One example from the patent literature is [52], which subtracts a smear signal from each column and applies a gain adjustment after the subtraction. The gain adjustment prevents bringing saturated columns down below the maximum code value, but adds gain variations to each column. In order to

control the gain variation, the amount of smear signal subtracted from each column is limited. In cases where the columns are saturated and thus effectively defective, Yoshida [90], and Kim [49] describe ways to approach concealment. Since the smear artifacts can be so wide, concealment may be of limited quality. Because smear artifacts are caused by the scene, imperfect concealment is usually more acceptable than with sensor defects. Because rapid movement can cause smear artifacts that appear as jagged diagonals, there is some recent art that seeks to correct even these artifacts [71].

2.2.5 Gain Nonuniformity Correction

As discussed in the previous chapter, gain nonuniformities are caused by several sources in digital cameras. The correction is essentially a multiplication of each pixel with a gain map. The variation in each implementation is the model used to determine the gain to be applied to each pixel. Early implementations, with very limited memory for storing gain corrections, used simple separable polynomials. Later implementations stored small images, with a gain value for each color channel for small tiles of the image, such as 4×16 , 8×8 , 16×16 , and so forth. These maps were often created to make the sensor response to a uniform illumination completely flat, which left taking lens effects and interactions uncompensated.

With increasing adoption of CMOS sensors and evolution to smaller pixels, gain corrections now usually include lens interactions. For a camera with a fixed lens, these are relatively simple. For cameras with interchangeable lenses, this creates new overhead to combine a sensor gain map with a lens interaction gain map. When lens effects get too severe (such as 50% falloff in the corners of the image), gain correction is usually limited to minimize noise amplification. This results as yet another system optimization, trading off darkness versus noisiness in the corners. With suitable noise reduction algorithms, the increase in noise can be mitigated, although usually with an increase in noise reduction artifacts.

Banding artifacts can be caused by interaction of a flickering illuminant with a rolling shutter. These are noticed more often with video than still captures and are discussed in Sect. 5.

2.2.6 Optics Corrections

In addition to illumination falloff, the taking lens can also produce other effects, particularly chromatic aberrations. The main aberrations that are corrected are geometric distortion, longitudinal color, lateral color, and spatially varying PSF. Smith contains a more complete discussion of geometric distortion and chromatic aberrations [84].

Geometric distortion is caused by magnification varying across the image and is usually described as being pincushion or barrel distortion. Magnification can be modeled as a low-order polynomial (5th order or less) function of distance from the center of the image. Geometric distortion is corrected by warping the image to invert

the change in magnification. The warping is usually done with a bilinear or other relatively simple interpolation, using a low-order function to represent a correction as a function of radial position [91]. The warping does not always completely correct the distortion, because variation during lens assembly causes the geometric distortion for specific cameras to vary slightly.

Lateral color occurs when magnification varies with color channel. Lateral color aberrations in the lens are corrected via a mechanism similar to correction of geometric distortion. Since the aberration is essentially a magnification that is slightly different for each color channel, resizing one or more color channels to achieve a common magnification can be folded in with the correction of geometric distortion.

Longitudinal color aberrations are caused when the different color channels are focused at different distances from the lens. For example, if the lens focus position is set so the green channel is in best focus, the red and blue channels may be slightly out of focus. Lens design can control this variation, but at increased lens cost. Usually, an autofocus system will bring the green channel into best focus and the red channel or blue channel will tend to be farther out of focus. The treatment for this artifact is to sharpen one or more color channels with a spatial filter. If this kind of correction is required, the geometric corrections can be included in the creation of these filters, by allowing use of asymmetric kernels and by allowing the filter kernels to vary with position in the image. Because distortion correction may spatially resample the color channels individually, it is often included in the processing chain after demosaicing.

Correction for a spatially varying PSF is similar to the correction for longitudinal color. Convolution with a spatially varying kernel is used, although usually only on a luma channel or on all three color channels.

Optics corrections can be particularly complex, especially if correcting for spatially varying PSF or longitudinal color aberrations. These corrections are especially common in more sophisticated desktop software for processing images from raw files, allowing the user to tune adjustments for the specific image and application.

2.3 *Stochastic Noise Reduction*

In most digital camera image processing chains, noise reduction is a critical operation. The main problem is that compact digital cameras usually operate with limited signal and significant noise. As mentioned at the beginning of Sect. 2, noise reduction is often addressed in several places in the processing chain. All noise reduction operations seek to preserve as much scene information as possible while smoothing noise. To achieve this efficiently, it is important to use relatively simple models to discriminate between scene modulation and noise modulation. Before the demosaicing step in Fig. 1, it is somewhat difficult to exploit inter-channel correlations for noise reduction. In the stochastic noise reduction block, grayscale techniques for noise reduction are usually applied to each color channel individually. There are many possible approaches, but two families of filtering, based upon different models of the image capture process, are discussed here.

The first of these models represents the random noise in the sensor capture and is used for range-based filtering, such as in a sigma filter [57] or the range component of a bilateral filter [85]. Early in a digital camera processing chain, a fairly simple model for noise variance is effective. There are two primary sources of random noise in the capture chain. The first is Poisson-distributed noise associated with the random process of photons being absorbed and converted into charge within a pixel. The second is electronic read noise, modeled with a Gaussian distribution. These two processes are independent, so a pixel value Q may be modeled as $Q = k_Q(q + g)$, where k_Q is the amplifier gain, q is a Poisson random variable with mean m_q and variance σ_q^2 , and g is a Gaussian random variable with mean m_g and variance σ_g^2 . Because q is a Poisson variable, $\sigma_q^2 = m_q$, and the total variance for a digitized pixel Q is

$$\sigma_Q^2 = k_Q^2(m_q + \sigma_g^2). \quad (1)$$

where m_q is the mean original signal level (captured photocharge), and σ_g^2 is the read noise. This relationship, that signal variance has a simple linear relationship with code value and a positive offset, allows a very compact parameterization of σ_Q^2 based upon a limited number of tests to characterize capture noise. Noise reduction based upon the noise levels in captured images allows smoothing of modulations with a high probability of being noise, while not smoothing over (larger) modulations that have a low probability of being noise.

Range-based filtering can introduce two main artifacts in the processed image. The most obvious is loss of fine texture. Since the noise reduction is based on smoothing small modulations and retaining large modulation, textures and edges with low contrast tend to get over-smoothed. The second artifact is the tendency to switch from smoothing to preservation when modulation gets larger. This results in a very nonuniform appearance in textured fields or edges, with portions of the texture being smoothed and other portions being much sharper. In some cases, this can lead to contouring artifacts as well.

The second simple model is an approximation of the capture PSF. Any point in the scene is spread over a finite area on the sensor in a spatially bandlimited capture. Thus, single-pixel outliers, or impulses, are more likely to be caused by sensor noise than scene content. While range-based filtering handles signal-dependent noise fairly well, it is prone to leave outliers unfiltered and it tends to increase the kurtosis of the distribution since it smooths small modulations more than larger modulations. The likelihood that impulses are noise leads to use of impulse filtering noise reduction. A standard center-weighted median filter can be effective, especially with lower cost cameras that have a large enough PSF to guarantee any scene detail will be spread over several pixels in the capture, thus preventing it from appearing as an impulse. More sophisticated approaches may be used for cameras with smaller or variable PSFs, such as digital SLR cameras.

The characteristic artifact caused by impulse filtering is elimination of small details from the scene, especially specular reflections from eyes and small lights. When applying impulse filters to CFA data, the filtering is particularly vulnerable to

creating colored highlights, if an impulse is filtered out of one or two color channel(s), but left in the remaining channel(s).

2.4 Exposure and White Balance Correction

The human visual system automatically adapts in complex ways when viewing scenes with different illumination. Research in the areas of color appearance models and color constancy continue to focus on developing models for how different scenes appear to human observers under different conditions. Because the human visual system generally has nonlinear responses and operates over a wide range of conditions, this process is extremely complex. A more restricted form of the problem is normally addressed in digital cameras. The goal is to capture neutral scene content with equal responses in all color channels ($R=G=B$), with the midtones being rendered near the middle of the tone scale, regardless of the illuminant or content of the scene.

White balance adjustment is accomplished by multiplying pixels in each color channel by a different gain factor that compensates for a non-neutral camera response and illuminant imbalance. In a digital camera, exposure adjustment is usually done primarily by controlling exposure time, analog gain, and f /number, but sometimes a digital exposure adjustment is used as well. This is done by scaling all three gain factors by a common factor. Application of the gain factors to the CFA data before demosaicing may be preferred, since some demosaicing algorithms may presume equal responses for the different color channels.

The other part of the exposure and white balance task is estimation or selection of appropriate gain factors to correct for illumination imbalance. Knowledge of the illuminant (and the camera's response to it) is critical. The camera's response to typical illuminants, such as daylight, incandescent, and fluorescent, is easily stored in the camera. Illuminant identification can be done manually through a user interface, which then drives selection of the stored gains for the illuminant. This is generally quite simple for the camera, but more tedious for the user. Another common approach is to allow the user to capture an image of a neutral (gray) under the scene illuminant and have the camera compute gain factors from that image.

The classic and most ill-posed form of the estimation problem is to analyze the image data to estimate the gains automatically, responding to the illuminant, regardless of the scene content. A classic difficult example is a featureless flat image with a reddish cast. Is it a gray card under incandescent illumination, or a reddish sheet of paper under daylight? Fortunately, normal scenes contain more information than a flat field.

Current cameras approach this estimation problem with different algorithms having different responses to scene content and illuminants. Camera manufacturers usually have somewhat different preferences, for example, biasing white balance to render images warmer or cooler, as well as different approaches to estimating the scene illuminant.

Most automatic white balance and exposure algorithms are based on some extension of the gray world model, that images of many different scenes will average out to 18% gray (a midtone gray). Unfortunately, this says very little about a specific image, and the algorithm must work well for individual images. Most extensions of the gray world model try to discount large areas of single colors, to avoid having the balance driven one way or another by red buildings, blue skies, or green foliage. There is also a tendency to more heavily weight colors closer to neutral more than colors far from neutral, but this complicates the algorithm, since the notion of neutral is not well-defined before performing illuminant estimation. This can be approached by applying a calibrated daylight balance to the image for analysis, then analyzing colors in the resulting image [25]. Another extension is to consider several possible illuminant classes and estimate the probability of each illuminant being the actual scene illuminant [24, 65].

Sometimes, highlight colors are given special consideration, based on the theory that highlights are specular reflections that are the color of the illuminant [56, 62, 64]. This breaks down for scenes that have no truly specular highlights. Some approaches also consider the color of particular scene content. The most common of these is using face detection and adjusting balance to provide a reasonable color for the face(s). This has other challenges, because faces themselves vary in color.

Using exposure control information such as scene brightness can help with the illuminant estimation. For example, a reddish scene at high illumination levels is more likely to be an outdoor scene near sunset, while the same scene with dim illumination is somewhat more likely to be indoor illumination. Another example is flash information. If an image is captured primarily with flash illumination, then the illuminant is largely known.

2.5 Adjusted CFA Image

The adjusted CFA image shown in Fig. 1 is still a single-channel image in which different pixels represent different color channels. It is now conditioned to more cleanly represent the scene, with few sensor-imposed artifacts. It also has less noise than the original capture, and is adjusted to represent the scene, with roughly equal red, green, and blue responses for neutral scene content, correcting out any illuminant imbalance.

2.6 Demosaicing

Capturing color images using a digital camera requires sensing at least three colors at each pixel location. One of the approaches to capture multiple colors at each pixel is to use a set of imaging sensors and project the scene onto each one of these sensors [58]. However, this increases the cost of the device and also requires careful alignment of

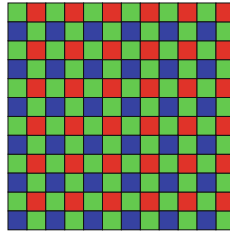


Fig. 6 Bayer CFA pattern

the sensors to produce a visually pleasing color image. Therefore, to reduce cost and complexity, most digital cameras are designed using a single monochrome CCD or CMOS image sensor with a CFA laid on top of the sensor [31, 58]. The CFA is a set of color filters that samples only one color at each pixel location, and the missing colors are estimated using interpolation algorithms. The CFA interpolation algorithms are also widely referred to as CFA demosaicing or demosaicing algorithms [28, 51, 87].

Among many CFA patterns, the Bayer CFA pattern [15] in Fig. 6 is one of the most commonly used CFA patterns in digital cameras. Since the human visual system is more sensitive to the green portion of the visual spectrum, the Bayer pattern consists of 50% green filters with the remaining 50% filters assigned equally to red and blue colors. The red, green, and blue pixels in the Bayer pattern are arranged as a 2×2 periodic minimal repeating unit, where each unit consists of two green filters, one red, and one blue filter. An example of a Bayer CFA image and the corresponding CFA interpolated color image is shown in Fig. 7.

Because the quality of the CFA interpolated image largely depends on the accuracy of the demosaicing algorithm, a great deal of attention has been paid to the demosaicing problem. Although simple non-adaptive image interpolation techniques (e.g., nearest neighborhood, bilinear interpolation, etc.) can be used to interpolate the CFA image, demosaicing algorithms designed to exploit inter-pixel and inter-channel correlations outperform non-adaptive interpolation algorithms as illustrated in Fig. 8. The original (ground truth) and the corresponding Bayer CFA images are shown in Figs. 8a, b, respectively. Three different demosaicing algorithms, namely, (i) nearest-neighborhood interpolation [32], (ii) bilinear interpolation [7], and (iii) directional linear minimum mean-square-error estimation (DLMMSE) [94] were applied to the CFA image and the corresponding demosaiced color images are shown in Fig. 8c–e. Both the nearest-neighborhood and bilinear interpolation algorithms are non-adaptive in nature and as a result, produced aliasing artifacts in the high-frequency regions. However, DLMMSE, due to its adaptive design, reconstructed the color image almost perfectly. For more details on the design and performance of various adaptive and non-adaptive CFA demosaicing algorithms, see [37, 60, 67, 93].

Placement of the CFA on top of the image sensor is essentially a downsampling operation. Therefore, the overall quality of color images (e.g., spatial resolution, color fidelity, etc.) produced by demosaicing algorithms not only depends upon

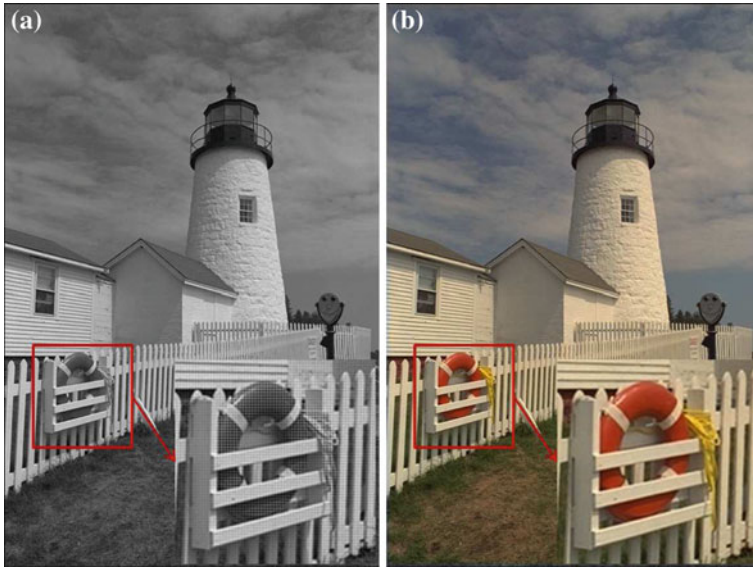


Fig. 7 Example color image reconstruction from Bayer pattern. **a** Bayer CFA image, **b** full-resolution CFA interpolated color image

the accuracy of the demosaicing algorithm but is also influenced significantly by the underlying CFA layout. Careful selection of a CFA pattern and corresponding demosaicing algorithm leads to high-quality color image reconstruction.

Although the Bayer pattern is one of the most commonly used CFA patterns, many others such as GGRB, RGBE, CYMM, CYGM, etc., [36, 63, 88, 89] also have been suggested for consumer digital cameras. Primarily influenced by manufacturing constraints and implementation costs, these CFA constructions and the corresponding demosaicing algorithms have been researched extensively. However, a systematic research framework for designing optimal CFA patterns is still a fairly new research direction. Some of the state-of-the-art developments in this area include Kodak's panchromatic CFA [53, 77], second-generation CFA [38], etc. A detailed review of these and other similar CFA patterns is beyond the scope of this chapter and readers are encouraged to refer to [14, 16, 61, 66, 82] for more details.

2.7 Interpolated RGB Image

After demosaicing, the interpolated image has three channels, each fully populated. The demosaicing process may have amplified noise in some of the color channels. The white balance process almost assuredly has as well, since gain factors greater than unity are normally used.

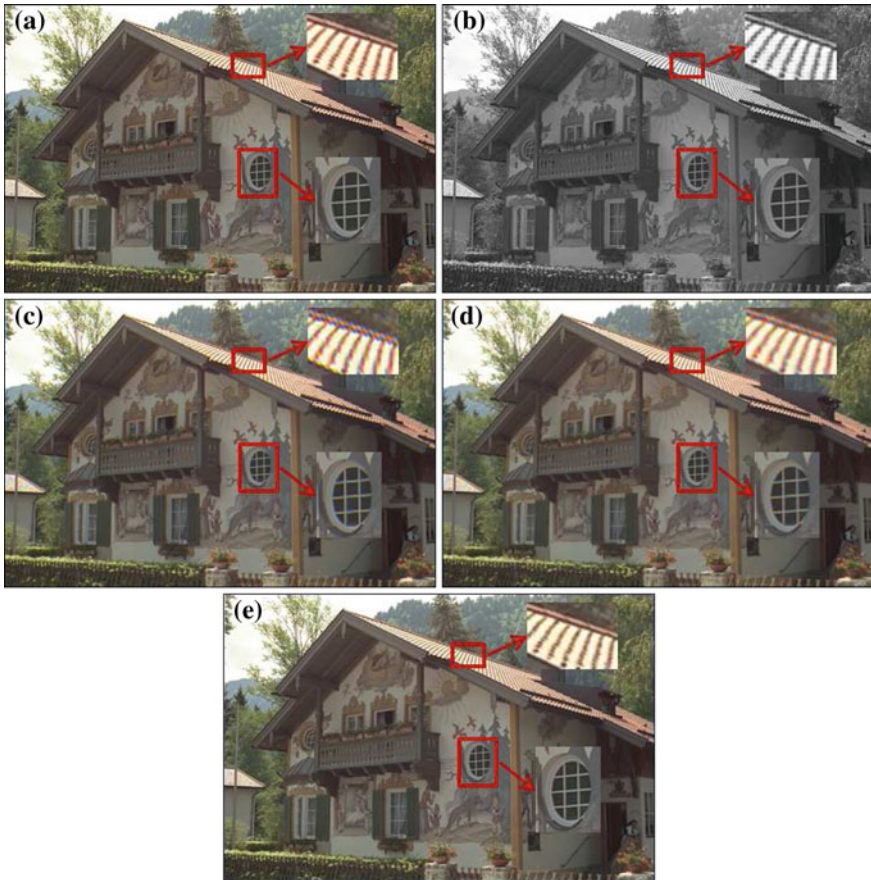


Fig. 8 Demosaicing algorithm comparisons: **a** original (ground truth) image, **b** Bayer CFA image, **c** nearest-neighbor interpolation, **d** bilinear interpolation and **e** DLMMSE [94]

2.8 Color Noise Reduction

Section 2.3 discussed noise reduction based upon simple single-channel noise models. This section discusses exploitation of inter-channel correlation and knowledge of the human visual system for further noise reduction. As mentioned before, these concepts are easier to apply after demosaicing, although application of these concepts before and during demosaicing remains a research opportunity.

Once three fully populated color channels are present, it is straightforward to rotate the color image to a luma-chroma color space. Because color correction has not yet been performed, this luma-chroma space will typically not be colorimetrically accurate. Still, a simple uncalibrated rotation such as in (2), similar to one proposed by Ohta [70], suffices to get most of the scene detail into the luma channel and most

of the color information into the two chroma channels.

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ -1 & 2 & -1 \\ -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Once separated, each channel is cleaned based upon the sensitivity of the human visual system. In particular, the luma channel is cleaned carefully, trying to preserve as much sharpness and detail as possible while providing adequate noise reduction. Digital cameras operate over a wide range of gain values, and the optimum balance of noise reduction and texture preservation typically varies with the input noise level. The chroma channels are cleaned more aggressively for several reasons. One reason is that sharp edges in the chroma channels may be color interpolation or aliasing artifacts left over from demosaicing [35]. The second is that viewers are less sensitive to chroma edges and especially sensitive to colored noise. The overall quality of the image is improved by emphasizing smoothness of the chroma channels rather than sharpness.

After noise reduction, this rotation is inverted as in (3).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} \quad (3)$$

Chroma-based noise reduction can produce two signature artifacts, color blobs and color bleed. Color blobs are caused by chroma noise that is smoothed and pushed to lower spatial frequencies in the process, without being eliminated. Successful cleaning at lower spatial frequencies requires relatively large filter kernels or iterative filtering. Implementations with constraints on available memory or processing power tend to leave low-frequency noise unfiltered.

Desktop implementations, with fewer constraints, can use pyramid [5] or wavelet decomposition [17] to reach the lowest frequencies. Chroma-based noise reduction is also prone to color bleed artifacts, caused by substantial mismatch in edge sharpness between the luma and chroma channels. Adaptive techniques that smooth the chroma channels using edge detection techniques such as [1] reduce the color bleeding problem by avoiding smoothing across edges. The visibility of color bleed artifacts depends in part upon the luma-chroma color space used for noise reduction. If the color space is less accurate in separating colorimetric luminance from chrominance data, color bleed artifacts will also affect the lightness of the final image, increasing the visibility of the artifacts. Adaptive chroma noise cleaning that can clean to very low frequencies while avoiding significant color bleeding is an open research question.

Color moiré patterns are sometimes addressed in a processing chain, often treated as a form of color noise. The usual approach is a variation of chroma noise reduction, including an additional test to check for high-frequency textures [3, 4].

2.9 Color Correction

After suitable noise reduction, colors are corrected, converting them from (white balanced) camera responses into a set of color primaries appropriate for the finished image. This is usually accomplished through multiplication with a color correction matrix, as in (4).

$$\begin{bmatrix} R_S \\ G_S \\ B_S \end{bmatrix} = \mathbf{C} \begin{bmatrix} R_C \\ G_C \\ B_C \end{bmatrix} \quad (4)$$

In this equation, \mathbf{C} is a 3×3 matrix with coefficients determined to convert from the camera's white balanced native sensitivity into a standard set of color primaries, such as sRGB [8], ROMM [81], or Adobe RGB (1998) [6].

One characteristic of color correction is the location of colors in the finished image color space. For example, the color of blue sky, skin, and foliage in the rendered image can vary from manufacturer to manufacturer. Different camera designers usually choose different objectives when doing this primary conversion; this can be thought of as combining color correction and preferred color rendering. Considerations affecting the determination of the color matrix are discussed in more depth by Hunt [40] and also Giorgianni and Madden [27]. If a standard color space with relatively small gamut, such as sRGB, is chosen, then many saturated colors will be outside the gamut. Depending upon the gamut-mapping strategy chosen, this can introduce clipping or gamut-mapping artifacts. The color correction matrix will usually amplify noise in the image. Sometimes, color correction is deliberately desaturated to reduce the noise in the rendered image.

If a more complex color correction transform is chosen, such as one that increases color saturation but preserves flesh tones at a less saturated position, then some nonuniform noise characteristics and even contouring may be observable.

2.10 Tone Scale and Gamma Correction

After color correction, a tone scale is applied to convert the image, still linear with respect to scene exposure, into a final color space. Sometimes, this is referred to as gamma correction, although most processing chains apply additional contrast adjustment beyond simple gamma correction. Along with color correction, the choice of the tone scale for rendering a reproduction of a scene is complex. As with color correction, issues involved in this selection are discussed in more depth by Hunt [40] and Giorgianni and Madden [27].

The most common processing chains simply apply a tone scale to all pixels in the image as a look-up table operation, regardless of scene content. Consumer preference is usually for a higher contrast tone scale, as long as no significant scene information is lost in the process. This is somewhat dependent upon scene and user expectation;

professional portraits are an example where the preference is for a lower contrast look.

More recently, processing chains are using adaptive tone scales that are adjusted for each scene. When such adjustments become aggressive (bringing up shadow detail, compressing highlight range), image texture becomes unnatural. If the contrast in the shadows is stretched, noise is amplified, leading to image quality degradation. If the highlights are compressed, texture in the highlights is flattened, leading to a different quality degradation.

The more sophisticated processing chains apply the adaptive tone scale with spatial processing. The approach is usually to use a multilevel decomposition to transform the image into a base image containing low spatial frequencies and a detail or texture image [26, 30]. Once the image is decomposed into base and detail images, the tone scale adjustments are applied to the base image with no changes or with controlled changes in detail. When the decomposition into base and texture images is imperfect, this approach can cause halo effects near high-contrast edges where the base image is adjusted extensively. In extreme cases, the image is rendered with an artificial decoupling of base image and texture, looking more like an animation rather than a photographic image. Image decomposition and derivation of spatially adaptive tone scales for optimal rendering of images are open areas of research [10, 20, 22].

2.11 Edge Enhancement

During image capture, edge detail is lost through optical and other effects. Most processing operations, especially noise reduction, further reduce high spatial frequency content. Display devices, printers, and the human visual system also attenuate system response. Edge enhancement, also known as sharpening, is a relatively simple spatial operation to improve the appearance of images, making it appear sharper and partially compensating for these losses.

The core of routine edge enhancement is a convolution operation to obtain an edge image, which is then scaled and added to the original image, as in (5).

$$\mathbf{A}' = \mathbf{A} + k\mathbf{E} \quad (5)$$

In this equation, \mathbf{A}' is the enhanced image, \mathbf{A} is the color-corrected image from the previous processing stage, k is a scalar gain, and \mathbf{E} is the edge enhancement image. The key variations in the process lie in the creation of \mathbf{E} . This is normally done with a standard spatial convolution, as in (6).

$$\mathbf{E} = \mathbf{A} * \mathbf{h} \quad (6)$$

In this equation, \mathbf{h} is a high pass convolution kernel. In other implementations, an unsharp mask formulation is chosen, as in (7).

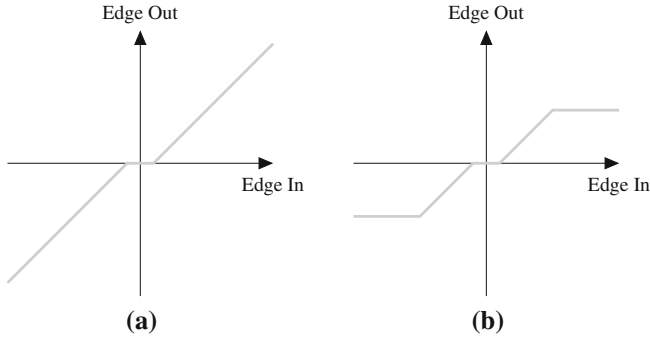


Fig. 9 Example edge enhancement nonlinearities: **a** soft thresholding, **b** edge limiting

$$\mathbf{E} = \mathbf{A} - \mathbf{A} * \mathbf{b} \quad (7)$$

In this equation, \mathbf{b} is a low-pass convolution kernel. If \mathbf{h} in (6) is chosen to be $\mathbf{h} = \mathbf{I} - \mathbf{b}$, then (6) and (7) are identical.

In both implementations, the design of the convolution kernel and the choice of k are the main tuning parameters. The design of the kernel controls which spatial frequencies to enhance, while k controls the magnitude of the enhancement. Often, the kernel is designed to produce a band-pass edge image, providing limited gain or even zero gain at the highest spatial frequencies. In most practical implementations, the size of the kernel is relatively small, with 5×5 being a common size.

Even with a band-pass kernel, this formulation amplifies noise and can produce significant halo or ringing artifacts at high-contrast edges. Both these problems can be treated by applying a nonlinearity to the edge image before scaling and adding to the original image, as in (8) or (9).

$$\mathbf{E} = L(\mathbf{A} * \mathbf{h}) \quad (8)$$

$$\mathbf{E} = L(\mathbf{A} - \mathbf{A} * \mathbf{b}) \quad (9)$$

Figure 9 shows two example nonlinearities, both based upon the magnitude of the edge value. The edge values with the smallest magnitude are most likely to be the result of noise, while larger edge values are likely to come from scene edges. The soft thresholding function shown in Fig. 9a reduces noise amplification by reducing the magnitude of all edge values by a constant, and is widely used for noise reduction, such as in [17]. Soft thresholding eliminates edge enhancement for small modulations, while continuing to enhance larger modulations. The edge-limiting nonlinearity shown in Fig. 9b also limits halo artifacts by limiting the edge value at the largest edge values, since high-contrast edges are the most likely to exhibit halo artifacts after edge enhancement.

Application of edge enhancement in an RGB color space will tend to amplify colored edges caused by any capture or processing artifacts earlier in the capture

chain, as well as colored noise. For this reason, edge enhancement is often applied to the luma channel of an image rather than all three color channels, for the same reasons that noise reduction is often applied in a luma-chroma color space. With more aggressive edge enhancement, different artifacts can be caused by the choice of different color spaces. Selection of a carefully chosen luma-chroma space tends to minimize artifacts, although aggressive edge enhancement can still lead to color bleeding problems if luma edges are enhanced too much more than chroma edges.

Selection of a color space for edge enhancement can depend upon several factors. For chains planning to apply JPEG compression in a YCbCr color space, the Y channel is a natural choice. In other cases, the green channel may be an acceptable luma channel—it is often the channel with the lowest noise and highest captured spatial resolution. This choice can produce artifacts with edges of different colors, however, since some luma edges will not appear in the green channel.

2.12 Finished Image

The finished image is ready for viewing, in a standard color space such as sRGB. It is possible to store this image in a file with no compression, such as in a TIFF file. In practice, this is very uncommon, since images from digital cameras are precisely the sort of natural photographic image for which lossy compression techniques such as JPEG were defined.

2.13 Compression

Once an image is fully processed, it is often compressed to reduce the amount of physical storage space required to represent the image data. Image compression algorithms can be divided into two categories: lossy and lossless. Lossless image compression algorithms are reversible, meaning that the exact original image data can be recovered from the compressed image data. This characteristic limits the amount of compression that is possible. Lossy image compression algorithms allow some of the original image data to be discarded, and only an approximation to the original image is recovered from the compressed image data. Many image compression algorithms have been proposed both academically and commercially, but digital cameras predominantly utilize the JPEG image compression standard, and in particular a baseline implementation of the JPEG lossy image compression standard [75].

The fundamental building blocks of a lossy JPEG image compression algorithm are illustrated in Fig. 10. Decompression can be achieved by inverting the operations and performing them in the reverse order. These building blocks, along with a common preprocessing step, are described below.

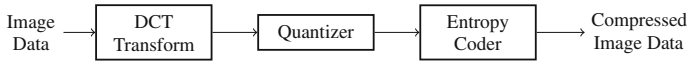


Fig. 10 JPEG lossy encoder building blocks

2.13.1 Preprocessing

JPEG can be used for single-component (channel) images as well as multi-component images. Each component is compressed separately, however, and thus it is advantageous when compressing a three-component RGB image to first reduce the correlation among the components by converting to a luma-chroma space, such as YCbCr. This conversion allows for more efficient compression since the components have less redundancy among them. Additionally, the JPEG standard allows some variability in the size of each component, and commonly the Cb and Cr components are subsampled by a factor of 2 horizontally and vertically as a preprocessing step.

2.13.2 DCT Transform

Each component is divided into a grid of 8×8 blocks, and a two-dimensional discrete cosine transform (DCT) is applied to each 8×8 block of image data. This operation, shown in (10), converts pixel values into transform coefficients corresponding to spatial frequencies.

$$X(u, v) = \frac{C(u)C(v)}{4} \sum_{m=0}^7 \sum_{n=0}^7 x(m, n) \cos \frac{(2m+1)u\pi}{16} \times \cos \frac{(2n+1)v\pi}{16}, \quad (10)$$

where

$$C(u) = \begin{cases} 1/\sqrt{2} & u = 0 \\ 1 & 1 \leq u \leq 7. \end{cases} \quad (11)$$

In (10), $x(m, n)$ are the pixel values for a given block, $X(u, v)$ are the corresponding DCT transform coefficients, and $0 \leq u, v \leq 7$.

For smoothly varying natural imagery composed mostly of low frequencies, the DCT compacts the majority of the energy for a given block into a small subset of the transform coefficients corresponding to low spatial frequencies (small values of u and v in (10)). Given infinite data precision, the DCT is invertible and the inverse DCT is applied during decompression to recover pixel values from the transform coefficients.

The block-based nature of the DCT used in JPEG can lead to a specific type of artifact known as a blocking artifact. Each 8×8 block of image data is compressed separately, and at high levels of compression, artificial transitions can appear at the boundary between two blocks. Since the image data is partitioned into a uniform grid of 8×8 blocks, the location of potential blocking artifacts is known in a JPEG

image. The presence of blocking artifacts at locations other than on the boundaries of 8×8 blocks suggests that an image has been modified in some way, such as by cropping. Blocking artifacts are a well-known characteristic of JPEG images, and many post processing techniques have been proposed to address them ([86] and references therein).

2.13.3 Quantization

Quantization is the lossy step of the JPEG algorithm. It refers to the “many-to-one” mapping of each input transform coefficient into one of a finite number of output levels. This is achieved by dividing each transform coefficient by the corresponding element of a quantization matrix (or quantization table) and rounding the result, as in (12).

$$Q(X(u, v)) = \text{round} \left(\frac{X(u, v)}{q(u, v)} \right). \quad (12)$$

In (12), $q(u, v)$ are the quantization table entries, and $Q(X(u, v))$ are the quantized transform coefficients. Larger values in the quantization table correspond to coarser quantization and greater compression. They also correspond to greater uncertainty and hence greater expected error when reconstructing the transform coefficients from their quantized values during decompression.

Quantization tables can be designed with regard to the human visual system. Because of the decreased sensitivity of the human visual system at high spatial frequencies, transform coefficients corresponding to high spatial frequencies can be quantized more aggressively than transform coefficients corresponding to low spatial frequencies. For multi-component images, JPEG allows multiple quantization tables. In particular, for YCbCr images, it is common to use one quantization table for the luma component (Y), and a separate quantization table for the chroma components (CbCr), exploiting the varying sensitivity of the human visual system to luma and chroma information.

The quantization tables used in the encoding process are included in an image file along with the compressed image data so that a decoder can correctly invert the quantization process. Quantization provides a rich source of information for forensic analysis of digital images. Different camera manufacturers use different quantization tables when generating JPEG images, and thus the values contained in the quantization tables can provide some information about the origin of a digital image. Additionally, requantization of an image, such as can happen when a compressed image is decompressed, modified, and recompressed, can result in unusual statistics in the compressed data [13].

2.13.4 Entropy Coding

The final main building block of a JPEG encoder involves entropy coding, which is a lossless process designed to efficiently encode a collection of symbols. In the case of JPEG, the symbols are related to the quantized transform coefficient values. Huffman codes are used as the entropy codes for the baseline implementation of JPEG lossy compression [39]. Broadly speaking, Huffman codes apply small codewords to represent symbols that occur frequently, and longer codewords to represent symbols that occur infrequently. The quantized transform coefficient values for an 8×8 block very often contain many zero values, in particular for coefficients corresponding to high spatial frequencies, and thus the Huffman codes are designed to efficiently represent these zeros (actually encoded as sequences, or runs, of zeros) with short codewords. During decompression, the entropy coding process can be exactly reversed to recover the quantized transform coefficient values.

3 Storage Formatting

Image storage formats are standardized means for organizing and storing digital images. They provide specifications for including image data and metadata in an image file. Metadata is any type of information that relates to the image, such as the camera model and manufacturer, the image size, and the date and time of the image capture. Widespread adoption of standardized image storage formats has ensured compatibility and interoperability among devices that capture and use digital images.

Current digital cameras almost uniformly use the Exif-JPEG image storage format to store JPEG-compressed image data [46, 47]. In some cases, however, JPEG-compressed data is not sufficient, and other standards, such as the TIFF/EP image storage format, are used to store raw image data [41].

3.1 *Exif-JPEG*

The Exif-JPEG image storage format provides a standard representation of digital images that allows compatibility between digital cameras and the devices and software applications that use the digital images produced by the cameras. The metadata associated with Exif-JPEG files is defined in the Exif specification. This metadata includes general information about the digital camera, as well as specific information about the camera capture settings used for a particular image [73]. Some examples of this metadata are shown in Table 1.

It is possible to have proprietary metadata in an Exif-JPEG image file. Individual camera manufacturers use proprietary metadata to include private information in the image file, such as the serial number of the camera. It may also include information about image processing settings used to generate the final image. Often camera

Table 1 Examples of metadata contained in Exif-JPEG image files

| Metadata field name | Description |
|---------------------|---|
| Make | Name of the manufacturer of the digital camera |
| Model | Model name/number of the digital camera |
| DateTime | Date and time the file was last modified |
| Exposure time | The time duration that the image was exposed on the image sensor |
| Fnumber | The lens f/number used when the image was captured |
| Flash | Provides information about whether the camera flash was used at capture |
| DateTimeOriginal | The data and time the picture was taken by the camera |
| MakerNote | Different camera makers store a variety of custom information |

manufacturers use the “MakerNote” metadata field to store proprietary metadata, and encrypt the contents to protect the proprietary nature of the information.

In general, it can be difficult to identify the source digital camera from which a given Exif-JPEG image file was produced. The camera image processing is designed to remove camera-specific artifacts from the digital image itself. Metadata may indicate the camera make and model, but not necessarily the specific camera used.

Exif-JPEG image files have drawbacks, and are not appropriate in all scenarios. The image is restricted to 24-bit color, using only 8 bits for each color component. The lossy JPEG compression algorithm can introduce artifacts. Additionally, in-camera image processing algorithms used to generate the final image may have sacrificed image quality for computational efficiency.

In some applications, it may be desirable to retain the raw image directly from the image sensor. The raw image data often has greater bit-depth, with 12 or 16 bits per pixel. Additionally, if the raw image is subsequently processed in a more powerful computing environment than available in the digital camera, more complex image processing algorithms can be used to generate the final image.

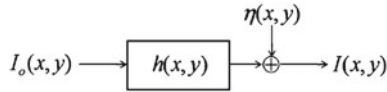
3.2 TIFF/EP

TIFF/EP was the first image format standardized for storing raw image data. In order to be able to properly interpret raw color image data, it is necessary to include metadata along with the image data that describes features such as the CFA pattern, and the color responses of the color channels. Table 2 shows some examples of the metadata associated with a raw color image included in a TIFF/EP file. In addition to the metadata specific to raw color image data, a TIFF/EP image file also contains metadata described previously with respect to Exif-JPEG image files.

Since the image data contained in a TIFF/EP image file has not been fully processed by the digital camera image processing path, it may still retain characteristics and artifacts described previously in this chapter that can be used to identify the specific camera from which the image was produced. Noise patterns and defec-

Table 2 Examples of metadata contained in TIFF/EP image files

| Metadata field name | Description |
|---------------------|--|
| Image width | Width of CFA image data |
| Image length | Length of CFA image data |
| Bits per sample | Bit-depth of each CFA sample |
| CFA pattern | The color filter array pattern |
| ICC color profile | Used to define RGB reference primaries, white point, and opto-electronic conversion function |

**Fig. 11** Image acquisition model for image restoration

tive data in the raw color image can be used to assess the likelihood that an image was generated by a particular camera.

One drawback of the TIFF/EP and other raw image storage formats is that while the format of the raw data is standardized, the image processing algorithms used to generate the final color image are not. Many possible finished images can be produced with different processing. Often, the image processing algorithms are proprietary, and the finished image may be of unknown quality if alternative processing algorithms are used.

4 Post-Processing Enhancement Algorithms

Despite significant advances in the areas of optics and semiconductors, digital cameras do not always provide *picture-perfect* images. Factors that affect image quality include sensor size, limited depth of field of the optical lenses, light condition, relative motion between the camera and the scene, etc. Image restoration techniques are commonly employed to restore the features of interest in the image [12, 29, 83]. However, image restoration techniques are (in general) computationally complex, which makes it impractical to use them during the image acquisition process. Therefore, image restoration is performed as a post processing step.

Classically, image restoration algorithms model the degradation process (such as motion blur, noise, out-of-focus blur, etc.) as a linear, space invariant system [29] as shown in Fig. 11. In this figure, $I_o(x, y)$ and $I(x, y)$ are (high-quality) unknown and (degraded) observed images, respectively, $h(x, y)$ is the system PSF and $\eta(x, y)$ is the measurement noise added during image acquisition.

The relationship between $I_o(x, y)$ and $I(x, y)$ can be expressed as a convolution operation as shown in (13) below,

$$I(x, y) = h(x, y) ** I_o(x, y) + \eta(x, y), \quad (13)$$

where “ $**$ ” denotes the 2- D convolution operation. Typically, inverse techniques are used to estimate $I_o(x, y)$ from (13) and the most commonly used technique is deconvolution [9]. In cases where the degradation PSF $h(x, y)$ is known, a Wiener filter-based approach can be used [44]. In most practical applications, the only known quantity is $I(x, y)$. In such cases, a blind deconvolution technique [55] is necessary to estimate both $I_o(x, y)$ and $h(x, y)$. If the degradation process is spatially variant (e.g., out-of-focus blur, etc.), then (13) is applied locally, *i.e.*, the observed image $I(x, y)$ is divided into several regions and (13) is applied to each region.

In some cases, perfect reconstruction of $I_o(x, y)$ from $I(x, y)$ is difficult even if $h(x, y)$ is known completely. For example, in the case of motion blur, $I_o(x, y)$ cannot be reconstructed completely if zeros of $h(x, y)$ in the frequency domain coincide with the nonzero frequency components of $I_o(x, y)$. Estimation of $I_o(x, y)$ can be improved significantly by using a multichannel approach. In the multichannel setup, multiple images of the same scene with complementary characteristics (e.g., multiple exposures, flash/no-flash, etc.) are acquired and $I_o(x, y)$ is reconstructed by using all of the observed images simultaneously [54, 59, 80].

An example to illustrate the advantages of multichannel image restoration is shown in Fig. 12. The original “cameraman” image in Fig. 12a was blurred using three different types of motion blur filters—(i) motion filter of length 5 and angle 13 (capture 1 in Fig. 12b), (ii) motion filter of length 6 and angle 120 (capture 2 in Fig. 12c), and (iii) motion filter of length 6 and angle 90 (capture 3 in Fig. 12d). The results of the Richardson–Lucy blind deconvolution applied to each capture and the multichannel blind restoration proposed by Kumar et al. [54] are shown in Fig. 12e–h. Clearly, the proposed multichannel algorithm performs better than single-channel deconvolution.

Multichannel image restoration requires complementary images of the same scene. A detailed description of capturing multiple images and processing them simultaneously for various image processing tasks such as deblurring, denoising, etc., can be found in [19, 76, 79, 92].

Image restoration algorithms enhance salient image features but they are not capable of improving the spatial resolution of an image. In many image processing applications such as image forensics, automatic target recognition (ATR), remote sensing, etc., images with high resolution (HR) are desired and often required. Image super-resolution [21, 23, 69, 72, 74] is a signal processing technique to obtain an HR image (or sequence) from a number of observed low-resolution (LR) images or a LR video sequence captured from the same scene. Image super-resolution is technically possible because of the information contained collectively among the LR images. For example, the spatial misalignment of the LR images, due to spatial sampling on an integer lattice, introduces sub-pixel shifts, from which the lost spatial high-frequency components can be estimated. Additional information can also be incorporated, such as the prior knowledge of a scene and an imaging degradation model. The processed image has a higher spatial resolution and reveals more content details.

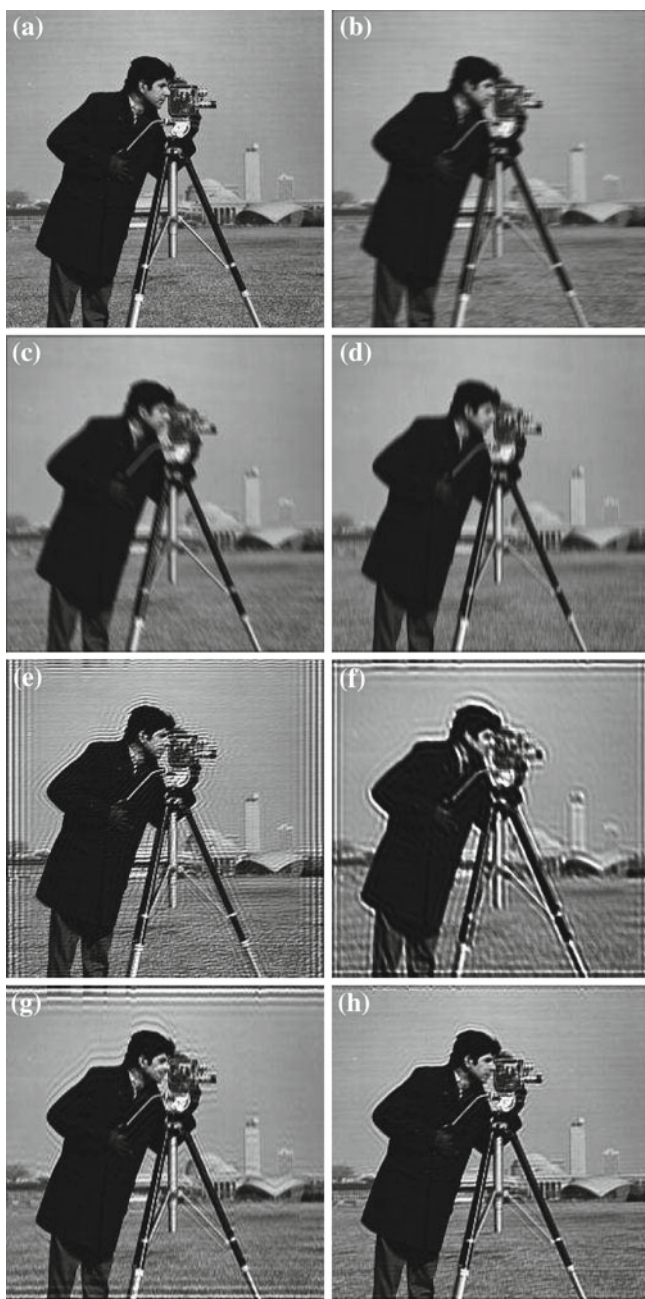


Fig. 12 Cameraman image: **a** original, **b** capture 1, **c** capture 2, **d** capture 3, **e** restored image from capture 1 only, **f** restored image from capture 2 only **g** restored image from capture 3 only and **h** multichannel restoration from all the three captures [54]

5 Video Capture

Processing for video capture is normally a simplified form of the processing used for still image capture. The same basic processing blocks listed in the nominal processing chain in Fig. 1 are still required, although many of them are simplified. The primary reason for this simplification is the time constraint on the operations. Processing for a still capture can take from a small fraction of a second to many seconds, depending upon the size of the image and sophistication of the processing. Even when a rapid sequence of still images is captured, the normal practice is to buffer the raw images in memory and process the images after capture. In contrast, video capture is performed at a fixed frame rate, such as 30 frames per second (fps), for an indefinite period of time. This precludes buffering approaches such as used with still capture, and requires the processing path to keep up with the capture rate.

One advantage for a video processing path is the limited size of the frames, normally two megapixels (1920×1080) or less, compared with five megapixels or much more for still capture. Further, because each image in the video is normally viewed only fleetingly, the image quality of each frame need not be as high.

On the other hand, there are several challenges not present with full-resolution processing. One is the problem of aliasing and colored edge artifacts. Use of binning to limit aliasing in LR readout was discussed in the previous chapter. One side effect of binning is the readout of color pixels whose spatial relationship (non-overlapping, partially overlapping) depends upon the binning pattern. As a result, a video processing path must deal with raw data that can have more complex spatial relationships between pixels than is usual with full-resolution readout.

Several additional issues not relevant to a single image come to light when processing a sequence of images. One is the use of temporal noise reduction. Approaches such as [45, 50] can provide significant benefit, although processing budgets usually constrain them to be relatively simple. In addition, temporal compression techniques, such as MPEG or H.264, are normally used with video, although some processing chains simply use lossy JPEG compression on individual frames [42, 43]. This greatly increases the bit rate for a given level of quality, but saves a great deal of compression processing.

Another processing block of potential interest with video forensics is exposure control. Because exposure control is essentially a feedback control loop, the response of exposure control and automatic white balance to changes in the scene can vary with camera model.

The response of the camera to flickering illuminants can also be revealing, especially if a rolling shutter is employed. Bands caused by illuminant flicker interacting with a rolling shutter exposure are a form of gain artifact that is sometimes treated in the processing path. The standard flickering illuminants oscillate at two times the AC power line frequency. Because the AC frequency is one of two values, flicker detection can be approached as a classification problem, analyzing a video or preview stream for evidence of flicker-induced bands at either 120 or 100 Hz [48, 78].

The most common processing path treatment is really *avoidance* of flicker bands, by using an exposure time that is an integer multiple of the flicker period. If flicker is determined to be a problem and not avoidable through selection of exposure time, it can be mitigated using gain corrections to correct the banding, such as in [11]. The resulting variations in digital gain can lead to noise amplification in bands of the image. As with other gain corrections, suitable noise reduction algorithms allow mitigation of the noise, although usually with an increase in noise reduction artifacts.

Acknowledgments The authors gratefully acknowledge many helpful discussions with James E. Adams, Jr., in the development and review of this chapter. Ken Parulski also provided a great deal of assistance for the discussion of file formats.

References

1. Adams JE Jr, Hamilton JF Jr (2003) Removing chroma noise from digital images by using variable shape pixel neighborhood regions. U.S. Patent 6621937
2. Adams JE Jr, Hamilton JF Jr (2008) Single-sensor imaging: methods and applications for digital cameras. Chapter, digital camera image processing chain design, 1st edn. CRC Press, Boca Raton, pp 67–103.
3. Adams JE Jr, Hamilton JF Jr, Hamilton JA (2004) Removing color aliasing artifacts from color digital images. U.S. Patent 6804392
4. Adams JE Jr, Hamilton JF Jr, Smith CM (2005) Reducing color aliasing artifacts from digital color images. U.S. Patent 6927804
5. Adams JE Jr, Hamilton JF Jr, Williams FC (2007) Noise reduction in color digital images using pyramid decomposition. U.S. Patent 7257271
6. Adobe RGB (1998) Color image encoding. Technical report Adobe Systems, Inc., San Jose. <http://www.adobe.com/adobeRGB>
7. Alleysson D, Susstrunk S, Herault J (2005) Linear demosaicing inspired by the human visual system. *IEEE Trans Image Process* 14(4):439–449
8. Anderson M, Motta R, Chandrasekar S, Stokes M (1995) Proposal for a standard default color space for the internet: sRGB. Fourth IS&T/SID color imaging conference, In, pp 238–245
9. Andrews HC, Hunt BR (1977) Digital image restoration. Prentice Hall, Englewood Cliffs
10. Bae S, Paris S, Durand F (2006) Two-scale tone management for photographic look. *ACM Trans. Graph.* 25(3):637–645 (2006). <http://doi.acm.org/10.1145/1141911.1141935>
11. Baer R (2007) Method and apparatus for removing flicker from images. U.S. Patent 7298401
12. Bando Y, Nishita T (2007) Towards digital refocusing from a single photograph. In: Proceedings of the 15th Pacific conference on computer graphics and applications, pp 363–372.
13. Bauschke HH, Hamilton CH, Macklem MS, McMichael JS, Swart NR (2003) Recompression of JPEG images by requantization. *IEEE Trans Image Process* 12(7):843–849
14. Bawolek E, Li Z, Smith R (1999) Magenta-white-yellow (MWY) color system for digital image sensor applications. U.S. Patent 5914749
15. Bayer BE (1976) Color imaging array. U.S. Patent 3971065
16. Bean J (2003) Cyan-magenta-yellow-blue color filter array. U.S. Patent 6628331
17. Chang S, Yu B, Vetterli M (2000) Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans Image Process* 9(9):1532–1546
18. CIE publ. no. 15.2 (1986) Colorimetry. Technical report, CIE
19. Debevec PE, Malik J (1997) Recovering high dynamic range radiance maps from photographs. *ACM SIGGRAPH*, In, pp 369–378

20. Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. In: SIGGRAPH '02: Proceedings of the 29th annual conference on computer graphics and interactive techniques, pp 257–266. ACM, New York, NY, USA. <http://doi.acm.org/10.1145/566570.566574>
21. Elad M, Feuer A (1997) Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Trans Image Process* 6(12):1646–1658
22. Farbman Z, Fattal R, Lischinski D, Szeliski R (2008) Edge-preserving decompositions for multi-scale tone and detail manipulation. In: SIGGRAPH '08: ACM SIGGRAPH 2008 papers, pp 1–10. ACM, New York, NY, USA. <http://doi.acm.org/10.1145/1399504.1360666>
23. Farsiu S, Robinson MD, Elad M, Milanfar P (2004) Fast and robust multiframe super resolution. *IEEE Trans Image Process* 13(10):1327–1344
24. Finlayson G, Hordley S, Huel P (2001) Color by correlation: a simple, unifying framework for color constancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 23(11):1209–1221
25. Gindele E, Adams JE Jr, Hamilton JF Jr, Pillman BH (2007) Method for automatic white balance of digital images. U.S. Patent 7158174
26. Gindele EB, Gallagher AC (2001) Method for adjusting the tone scale of a digital image. U.S. Patent 6275605
27. Giorgianni EJ, Madden TE (2008) Digital color management encoding solutions, 2nd edn. Wiley, New York
28. Glotzbach J, Schafer R, Illgner K (2001) A method of color filter array interpolation with alias cancellation properties. In: Proceedings of the IEEE International Conference Image Processing, vol 1. pp 141–144
29. Gonzalez RC, Woods RE (2007) Digital image processing, 3rd edn. Prentice Hall, Englewood Cliffs
30. Goodwin RM, Gallagher A (1998) Method and apparatus for areas selective exposure adjustment. U.S. Patent 5818975
31. Gunturk B, Glotzbach J, Altunbasak Y, Schafer R, Mersereau R (2005) Demosaicking: color filter array interpolation. *IEEE Signal Process Mag* 22(1):44–54
32. Gupta M, Chen T (2001) Vector color filter array demosaicing. In: Proceedings of the SPIE, vol 4306. pp 374–382
33. Hamilton JF Jr (2004) Correcting for defects in a digital image taken by an image sensor caused by pre-existing defects in two pixels in adjacent columns of an image sensor. U.S. Patent No. 6741754
34. Hamilton JF Jr (2005) Correcting defect in a digital image caused by a pre-existing defect in a pixel of an image sensor. U.S. Patent No. 6900836
35. Hamilton JF Jr, Adams JE Jr (2003) Correcting for chrominance interpolation artifacts. U.S. Patent 6542187
36. Hamilton JF Jr, Adams JE Jr, Orlicki D (2001) Particular pattern of pixels for a color filter array which is used to derive luminance and chrominance values. U.S. Patent 6330029B1
37. Hirakawa K, Parks T (2005) Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans Image Process* 14(3):360–369
38. Hirakawa K, Wolfe P (2007) Spatio-spectral color filter array design for enhanced image fidelity. In: Proceedings of the ICIP, pp II-81–II-84
39. Huffman DA (1952) A method for the construction of minimum-redundancy codes. *Proc Inst Radio Eng* 40(9):1098–1101
40. Hunt R (1987) The reproduction of colour. Fountain Press, England
41. ISO 12234-2:2001 (2001) Electronic still picture imaging—removable memory—part 2: TIFF/EP image data format
42. ISO/IEC 14496-10:2003 (2003) Information technology—coding of audio-visual objects—part 10: advanced video coding
43. ISO/IEC 14496-2:1999 (1999) Information technology—coding of audio-visual objects—part 2: visual
44. Jain A, Ranganath S (1981) Application of two dimensional spectral estimation in image restoration. In: IEEE International Conference Acoust., Speech, Signal Process (ICASSP), pp 1113–1116

45. Jain C, Sethuraman S (2008) A low-complexity, motion-robust, spatio-temporally adaptive video de-noiser with in-loop noise estimation. *Proceedings of the ICIP*, In, pp 557–560
46. JEITA CP 3451 (2002) Exchangeable image file format for digital still cameras: Exif version 2.2
47. JEITA CP 3451-1 (2003) Amendment 1 exchangeable image file format for digital still cameras: Exif version 2.21 (amendment to version 2.2)
48. Kaplinsky M, Subbotin I (2006) Method for mismatch detection between the frequency of illumination source and the duration of optical integration time for image with rolling shutter. U.S. Patent 7142234
49. Kim HS, Kim BG, Kim CY, Joo YH (2008) Digital photographing apparatus and method for detecting and correcting smear by using the same. U.S. Patent publication no. 2008/0084488 A1
50. Kim YJ, Oh JJ, Choi BT, Choi SJ, Kim ET (2009) Robust noise reduction using a hierarchical motion compensation in noisy image sequences. In *Digest of Technical Papers of the ICCE*, pp 1–2
51. Kimmel R (1999) Demosaicing: image reconstruction from color CCD samples. *IEEE Trans Image Process* 8(9):1221–1228
52. Kondo K (2009) Image sensing apparatus, image sensing method, and program. U.S. Patent 7545420
53. Kumar M, Morales E, Adams JE Jr, Hao W (2009) New digital camera sensor architecture for low light imaging. *Proceedings of the ICIP*, In, pp 2681–2684
54. Kumar M, Ramuhalli P (2005) Dynamic programming based multichannel image restoration. In: *IEEE International Conference Acoustics, Speech, Signal Process (ICASSP)*, vol 2. pp 609–612
55. Kundur D, Hatzinakos D (1996) Blind image deconvolution. *IEEE Signal Process Mag* 13(3):43–64
56. Lee H (1986) Method for computing the scene-illuminant chromaticity from specular highlights. *J Opt Soc Am A* 3:1694–1699
57. Lee J (1983) Digital image smoothing and the sigma filter. *Comput Vis Graph Image Process* 24:255–269
58. Li X, Gunturk B, Zhang L (2008) Image demosaicing: a systematic survey. In: *SPIE Conference on VCIP*, vol 6822, pp. 68, 221J-68, 221J-15
59. Liu X, Gamal AE (2001) Simultaneous image formation and motion blur restoration via multiple capture. In: *IEEE International conference acoustics, speech, signal process (ICASSP)*, vol 3. pp 1841–1844
60. Lu W, Tan YP (2003) Color filter array demosaicing: new method and performance measures. *IEEE Trans Image Process* 12(10):1194–1210
61. Lukac R, Plataniotis K (2005) Color filter arrays: design and performance analysis. *IEEE Trans Consum Electron* 51(4):1260–1267
62. McCann JJ, McKee SP, Taylor TH (1976) Quantitative studies in retinex theory. *Vis Res* 16:445–458
63. Miao L, Qi H, Snyder W (2004) A generic method for generating multispectral filter arrays. In: *Proceedings of the ICIP*, vol 5. pp 3343–3346
64. Miyano T (1997) Auto white adjusting device. U.S. Patent 5659357
65. Miyano T, Shimizu E (1997) Automatic white balance adjusting device. U.S. Patent 5644358
66. Moghadam A, Aghagolzadeh M, Radha H, Kumar M (2010) Compressive demosaicing. In: *Proceedings of the MMSP*, pp 105–110
67. Mukherjee J, Parthasarathi R, Goyal S (2001) Markov random field processing for color demosaicing. *Pattern Recognit. Lett.* 22(3–4):339–351
68. Multiple output sensors seams correction (2009) Technical report application note revision 1.0 MTD/PS-1149, Eastman Kodak image sensor solutions
69. Ng MK, Bose NK (2003) Mathematical analysis of super-resolution methodology. *IEEE Signal Process Mag* 20(3):62–74

70. Ohta YI, Kanade T, Sakai T (1980) Color information for region segmentation. *Comput Graph Image Process* 13(3):222–241
71. Okura Y, Tanizoe Y, Miyake T (2009) CCD signal processing device and image sensing device. U.S. Patent publication no. 2009/0147108 A1
72. Park SG, Park MK, Kang MG (2003) Super-resolution image reconstruction: a technical overview. *IEEE Signal Process Mag* 20(3):21–36
73. Parulski KA, Reisch R (2008) Single-sensor imaging: methods and applications for digital cameras. Chapter, digital camera image storage formats, 1st edn. CRC press, Boca Raton, pp 351–379
74. Patti AJ, Sezan MJ, Tekalp AM (1997) Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans Image Process* 6(8):1064–1076
75. Pennebaker WB (1993) Mitchell JL (1993) JPEG still image data compression standard. Van Nostrand Reinhold, New York
76. Petschnigg G, Szeliski R, Agrawala M, Cohen M, Hoppe H, Toyama K (2004) Digital photography with flash and no-flash image pairs. In: *Proceedings of the ACM SIGGRAPH*, vol 23. pp 664–672
77. Pillman B, Deever A, Kumar M (2010) Flexible readout image capture with a four-channel CFA. In: *Proceedings of the ICIP*, pp 561–564
78. Poplin D (2006) An automatic flicker detection method for embedded camera systems. *IEEE Trans Consum Electron* 52(2):308–311
79. Raskar R, Agrawal A, Tumblin J (2006) Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans Graph* 25:795–804
80. Rav-Acha A, Peleg S (2005) Two motion-blurred images are better than one. *Pattern Recognit Lett* 26(3):311–317
81. Reference output medium metric RGB color space (ROMM RGB) white paper (1999) Technical report version 2.2, accession number 324122H, Eastman Kodak Company
82. Roddy J, Zolla R, Blish N, Horvath L (2006) Four color image sensing apparatus. U.S. Patent 7057654
83. Shan Q, Jia J, Agarwala A (2008) High-quality motion deblurring from a single image. *ACM Trans Graph* 27:1–10
84. Smith WJ (1966) *Modern optical engineering*. McGraw-Hill, San Francisco
85. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: *Proceedings of the 1998 IEEE international conference on computer vision*. IEEE
86. Triantafyllidis GA, Tzovaras D, Strintzis MG (2002) Blocking artifact detection and reduction in compressed data. *IEEE Trans Circuits Syst Video Technol* 12(10):877–890
87. Trussell H, Hartwig R (2002) Mathematics for demosaicking. *IEEE Trans Image Process* 11(4):485–492
88. Yamagami T, Sasaki T, Suga A (1994) Image signal processing apparatus having a color filter with offset luminance filter elements. U.S. Patent 5323233
89. Yamanaka S (1997) Solid state camera. U.S. Patent 4054906
90. Yoshida H (2004) Image pickup apparatus and method of correcting deteriorated pixel signal thereof. U.S. Patent 6809763
91. Yu W (2003) An embedded camera lens distortion correction method for mobile computing applications. *IEEE Trans Consum Electron* 49(4):894–901. doi:[10.1109/TCE.2003.1261171](https://doi.org/10.1109/TCE.2003.1261171)
92. Yuan L, Sun J, Quan L, Shum HY (2007) Image deblurring with blurred/noisy image pairs. *ACM Trans Graph* 26:1–10
93. Zhang F, Wu X, Yang X, Zhang W, Zhang L (2009) Robust color demosaicking with adaptation to varying spectral correlations. *IEEE Trans Image Process* 18(12):2706–2717
94. Zhang L, Wu X (2005) Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Trans Image Process* 14(12):2167–2178

Digital Image Forensics

There is More to a Picture than Meets the Eye

Sencar, H.T.; Memon, N. (Eds.)

2013, VIII, 372 p., Hardcover

ISBN: 978-1-4614-0756-0