Chapter 2
Image Demosaicing

Ruiwen Zhen and Robert L. Stevenson

2.1 Introduction

Digital cameras are extremely popular and have replaced traditional film-based cameras in most applications. To produce a color image in a digital camera, there should be at least three color components at each pixel location. This can be achieved by three CCD (Charge-Coupled Devices) or CMOS (Complementary Metal-Oxide Semiconductor) sensors, each of which receives a specific primary color. However, the associated cost and space is prohibited in many situations. As a result, most digital cameras on the market use a single sensor covered by a color filter array (CFA) to reduce the cost and size.

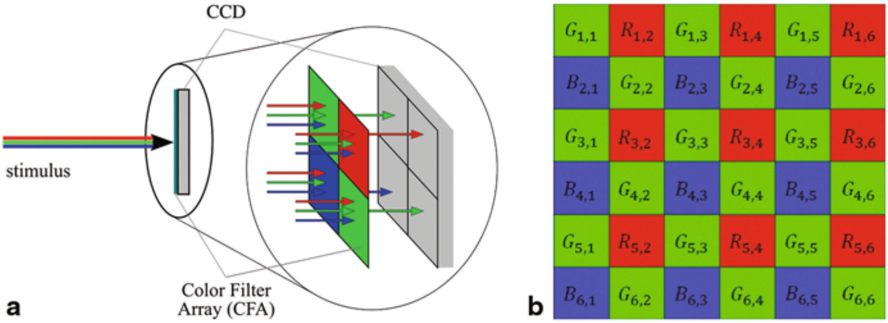
The CFA consists of a set of spectrally selective filters that are arranged in an interleaving pattern so that each sensor pixel samples one of the three primary color components (Fig. 2.1a). These sparsely sampled color values are termed mosaic images or CFA images. To render a full-color image from the CFA samples, an image reconstruction process, commonly known as CFA demosaicing, is required to estimate the other two missing color values for each pixel. Among many possible CFA patterns, we focus on the widely used Bayer CFA pattern [8] shown in Fig. 2.1b. The Bayer pattern samples the green band using a quincunx grid, while red and blue are obtained by a rectangular grid. The green pixels are sampled at a higher rate since the green color approximates the brightness perceived by human eyes.

Before fully exploring the various demosaicing algorithms, we will introduce the basic knowledge about demosaicing in this section. We start from the formalism for demosaicing process and the simplest demosaicing method, bilinear interpolation, which allows us to introduce the demosaicing color artifacts, and major principles adopted by most demosaicing algorithms. After that, we show how to evaluate

R. Zhen (✉) · R. L. Stevenson
University of Notre Dame, 275 Fitzpatrick Hall, Notre Dame, IN 46556, USA
e-mail: rzhen@nd.edu

R. L. Stevenson
e-mail: rls@nd.edu

M. E. Celebi et al. (eds.), *Color Image and Video Enhancement*,
DOI 10.1007/978-3-319-09363-5_2



$G_{1,1}$	$R_{1,2}$	$G_{1,3}$	$R_{1,4}$	$G_{1,5}$	$R_{1,6}$
$B_{2,1}$	$G_{2,2}$	$B_{2,3}$	$G_{2,4}$	$B_{2,5}$	$G_{2,6}$
$G_{3,1}$	$R_{3,2}$	$G_{3,3}$	$R_{3,4}$	$G_{3,5}$	$R_{3,6}$
$B_{4,1}$	$G_{4,2}$	$B_{4,3}$	$G_{4,4}$	$B_{4,5}$	$G_{4,6}$
$G_{5,1}$	$R_{5,2}$	$G_{5,3}$	$R_{5,4}$	$G_{5,5}$	$R_{5,6}$
$B_{6,1}$	$G_{6,2}$	$B_{6,3}$	$G_{6,4}$	$B_{6,5}$	$G_{6,6}$

Fig. 2.1 **a** Single CCD sensor covered by a CFA [35], **b** Bayer CFA

demosaicing algorithms, including the test image database, objective measure, and subjective measure of quality.

2.1.1 Demosaicing

Let $I^{CFA}:\mathbb{Z}^2 \to \mathbb{Z}$ denote a $M \times N$ Bayer CFA image. Each pixel $I^{CFA}(i, j)$ with coordinates $i = 1, 2...M$, $j = 1, 2...N$ in the image I^{CFA} corresponds to a single color component. Assuming the sampling pattern is as Fig. 2.1b, then

$$I^{CFA}(i, j) = \begin{cases} R_{i,j} & \text{for } i \text{ odd and } j \text{ even} \\ B_{i,j} & \text{for } i \text{ even and } j \text{ odd} \\ G_{i,j} & \text{otherwise} \end{cases} \tag{2.1}$$

where $R_{i,j}, B_{i,j}, G_{i,j}$ values range from 0 to 255 if the image is quantized with 8-bit for each color channel. The demosaicing process is to estimate the missing two color values at each pixel location (i, j) for rendering a full color image $\hat{I}:\mathbb{Z}^2 \to \mathbb{Z}^3$:

$$\hat{I}(i, j) = \begin{cases} (R_{i,j}, \hat{G}_{i,j}, \hat{B}_{i,j}) & \text{for } i \text{ odd and } j \text{ even} \\ (\hat{R}_{i,j}, \hat{G}_{i,j}, B_{i,j}) & \text{for } i \text{ even and } j \text{ odd} \\ (\hat{R}_{i,j}, G_{i,j}, \hat{B}_{i,j}) & \text{otherwise} \end{cases} \tag{2.2}$$

Each triplet in Eq. (2.2) represents a color vector, in which $R_{i,j}, B_{i,j}, G_{i,j}$ are color components available in the CFA image I^{CFA} and $\hat{R}_{i,j}, \hat{B}_{i,j}, \hat{G}_{i,j}$ are estimated missing color components [35]. For use in later discussion, we also define the original full-color image as:

$$I(i, j) = (R_{i,j}, G_{i,j}, B_{i,j}) \quad \text{for } \forall i \text{ and } \forall j \tag{2.3}$$

Many algorithms have been proposed for CFA image demosaicing. The simplest demosaicing methods apply well-known interpolation techniques, such as nearest-neighbor replication, bilinear interpolation, and cubic spline interpolation, to each color plane separately. The remaining part of this subsection will introduce the bilinear interpolation demosaicing method. The bilinear approach is useful to understand since many advanced algorithms still adopt bilinear interpolation as an initial step; additionally, these algorithms usually use the results of bilinear interpolation for performance comparison.

The bilinear interpolation method fills the missing color values with weighted averages of their neighboring pixel values. Considering the CFA pattern in Fig. 2.1b, the missing blue and green values at pixel $R_{3,4}$ are estimated thanks to the following equations:

$$\hat{B}_{3,4} = \frac{1}{4}(B_{2,3} + B_{2,5} + B_{4,3} + B_{4,5}) \tag{2.4}$$

$$\hat{G}_{3,4} = \frac{1}{4}(G_{3,3} + G_{2,4} + G_{3,5} + G_{4,4}) \tag{2.5}$$

Similarly, the red and green components can be estimated at blue pixel locations. As for the green pixel location, for example $G_{3,3}$, the blue and red values are calculated as:

$$\hat{R}_{3,3} = \frac{1}{2}(R_{3,2} + R_{3,4}) \tag{2.6}$$

$$\hat{B}_{3,3} = \frac{1}{2}(B_{2,3} + B_{4,3}) \tag{2.7}$$

These interpolation operations can be easily implemented by convolution [6]. If we decompose the CFA image into three color planes, I_R^{CFA}, I_G^{CFA} , and I_B^{CFA}, as shown in Fig. 2.2, the convolution kernels for bilinear interpolating of each color plane are:

$$K_B = K_R = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.8}$$

$$K_G = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.9}$$

Figure 2.3 shows an example of bilinear interpolating of the image *lighthouse* with the above convolution kernels. Though the bilinear interpolation method is computationally efficient and easy to implement, we see that the demosaiced image in Fig. 2.3b suffers from severe visible artifacts, especially the image regions with high-frequency content.

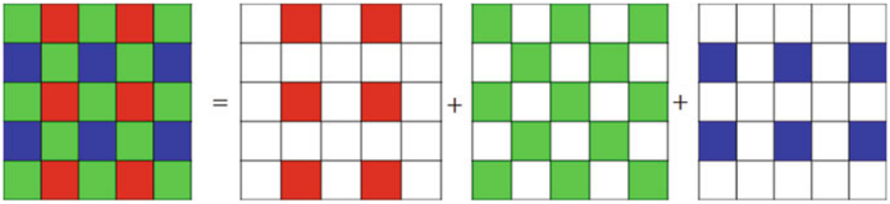


Fig. 2.2 CFA color plane decomposition

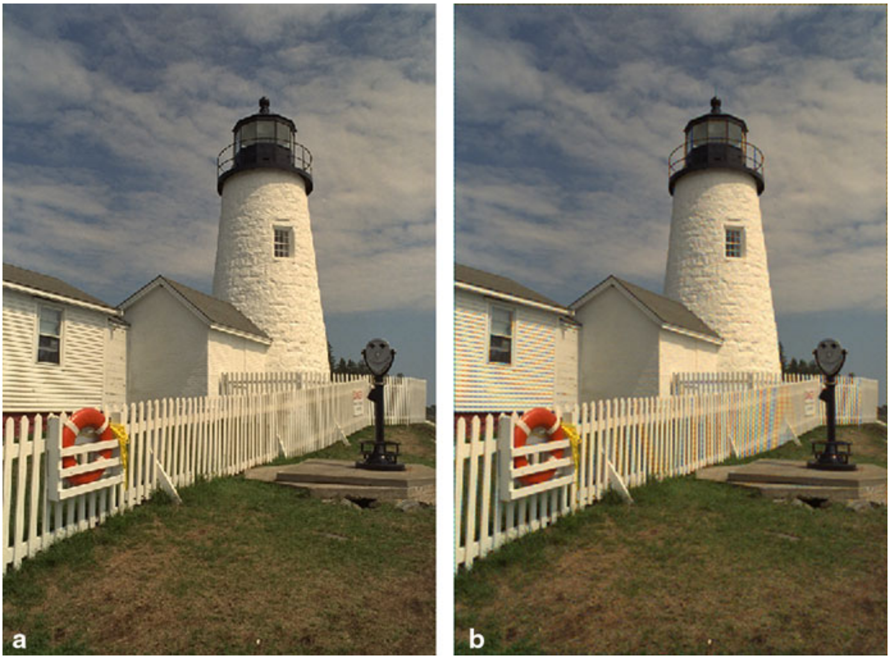


Fig. 2.3 **a** Original image, **b** Demosaiced image by bilinear interpolation

2.1.2 Demosaicing Artifacts

To analyze the demosaicing artifacts introduced by bilinear interpolation, Chang et al. [11] synthesized an image with a vertical edge (Fig. 2.4a) and obtained the corresponding bilinear interpolated result (Fig. 2.4c). The synthesized image has two homogeneous areas with different gray levels L and H $(L < H)$, and the three color components in each gray area are equal. Figure 2.4b shows the Bayer CFA image yielded by sampling the synthesized image. The results of bilinear interpolating of each color plane are displayed in Figs. 2.4d–2.4f.

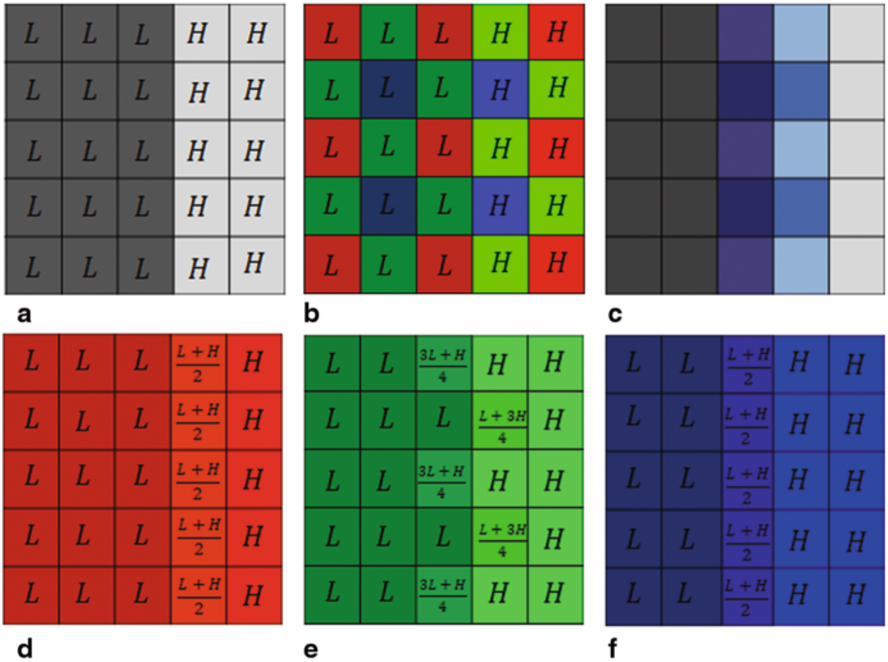


Fig. 2.4 **a** Synthesized gray image, **b** CFA samples of **a**, **c** Bilinear interpolation result, **d** Bilinear interpolated red plane, **e** Bilinear interpolated green plane, **f** Bilinear interpolated blue plane

We can see that the three interpolated color planes suffer from different errors due to their different sampling patterns. The green plane gives rise to the obvious grid error pattern while the red and blue planes produce an intermediate level between low and high intensity levels. Visually, two types of artifacts are generated in the demosaiced image: one is the pattern of alternating colors along the edge, called zipper effect, and the other is the noticeable color errors (the bluish tint in this example), called false color.

The zipper effect refers to the abrupt or unnatural changes of intensities over a number of neighboring pixels, manifesting as an "on–off" pattern in regions around edges [11]. Figure 2.5b shows that the fence bars in the bilinear interpolated *lighthouse* are corrupted by the zipper effects. They are primarily caused by improper averaging of neighboring color values across edges. Interpolation along an object boundary is always preferable to interpolation across it because the discontinuity of the signal at the boundary contains high-frequency components that are difficult to estimate. If an image is interpolated in the direction orthogonal to the orientation of the object boundary, the color that appears at the pixel of interest is unrelated to the physical objects represented in the image [27]. For this reason, many proposed demosaicing algorithms are edge-sensitive. Another reason that could influence zipper effects is the quincunx structure of the CFA green samples. According to Chang's experimental results, the zipper effects are more likely to occur around

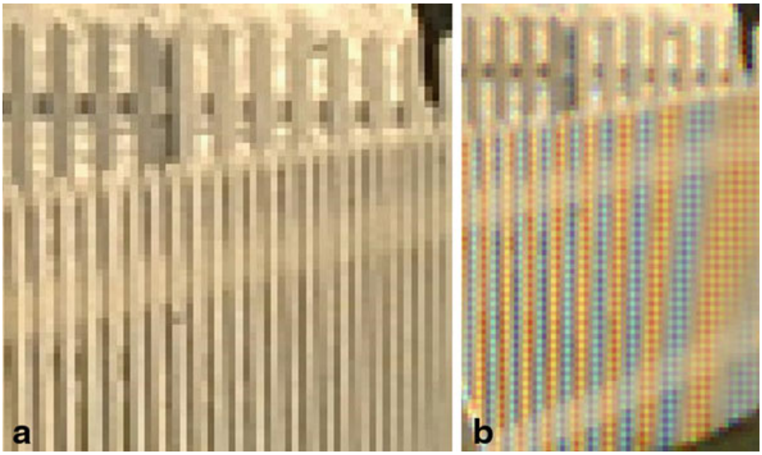


Fig. 2.5 Zipper effect. **a** Fence bars in the original image, **b** Fence bars in the bilinear interpolated image

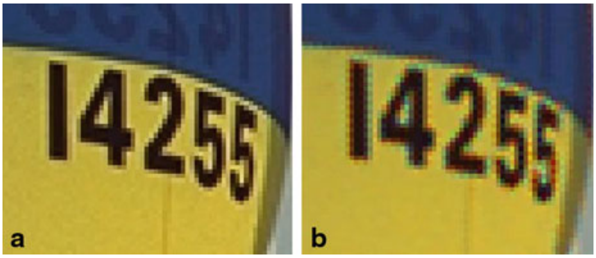


Fig. 2.6 False color. **a** Numbers in the original image, **b** Numbers in the bilinear interpolated image

edges not aligned in the diagonal direction along which the green values are fully sampled.

The false colors are spurious colors which are not present in the original image, as in Figs. 2.5b and 2.6b. They appear as sudden hue changes due to inconsistency among the three color planes. Such inconsistency usually results in the large intensity changes in the color difference planes [11]. Based on this observation, many algorithms attempt to utilize the spectral correlation between different planes and ensure that the hue or color difference plane is slowly varying.

Both the zipper effect and the false color are referred to as misguidance color artifacts, which are mainly caused by erroneous interpolation direction. These artifacts affect the regions with high-frequency content most. However, even with correct interpolation direction, the reconstructed image may still contain several errors called interpolation artifacts, and it is associated with limitations in the interpolation [27]. Normally, interpolation artifacts are far less noticeable than misguidance color artifacts.

2.1.3 Demosaicing Principles

The drawbacks brought by simple interpolation in separate planes motivated the appearance of more advanced algorithms specifically designed for the reconstruction of CFA images to improve the overall demosaicing performance. An excellent review of the demosaicing algorithms proposed in the past several decades can be found in [31,35,47]. In order to reduce the misguidance color artifacts, most of them are developed based on three principles: spectral correlation, spatial correlation, and green-plane-first rule.

The most popular principle in the demosiacing literature appears to be the green-plane-first rule, that is to interpolate the green plane first. The key motivation behind this principle is that the green component is less aliased than the other two. Thus, having a full-resolution green plane could facilitate the recovery of blue and red planes. In addition, human eyes are more sensitive to the change of the luminance component (green) than that of the chrominance components. The interpolation accuracy of the green plane is critical to the quality of the demosaiced image.

The spectral correlation of a color image dictates that there is a strong dependency among the pixel values of different color planes, especially in areas with high spatial frequencies [11]. This correlation is usually exploited by using the assumption that the differences (or ratios) between the pixel values in two color planes are likely to be constant within a local image region. In 1987, Cok [15] first proposed interpolation based on color hue constancy. Hue is understood as the ratio between chrominance and luminance, i.e., R/G and B/G. Following his work, several schemes [2, 29] were devised to estimate the missing color values with the aid of other color planes. The formal statement of the hue constancy is given below:

- The color ratios between green and red/blue channels satisfy:

$$\begin{aligned} \frac{R_{i,j}}{G_{i,j}} &= C^{rg}_{i,j} \text{ and} \\ \frac{B_{i,j}}{G_{i,j}} &= C^{bg}_{i,j} \end{aligned} \tag{2.10}$$

where $C^{rg}_{i,j}$ and $C^{bg}_{i,j}$ are piecewise constant within the boundary of a given object.

However, later work asserted that the differences instead of the ratios between green and red/blue planes are slowly varying [3, 21, 22, 49, 59], i.e.,

- The color differences between green and red/blue channels satisfy:

$$\begin{aligned} R_{i,j} &= G_{i,j} + A^{rg}_{i,j} \text{ and} \\ B_{i,j} &= G_{i,j} + A^{bg}_{i,j} \end{aligned} \tag{2.11}$$

where $A^{rg}_{i,j}$ and $A^{bg}_{i,j}$ are piecewise constant within the boundary of a given object.

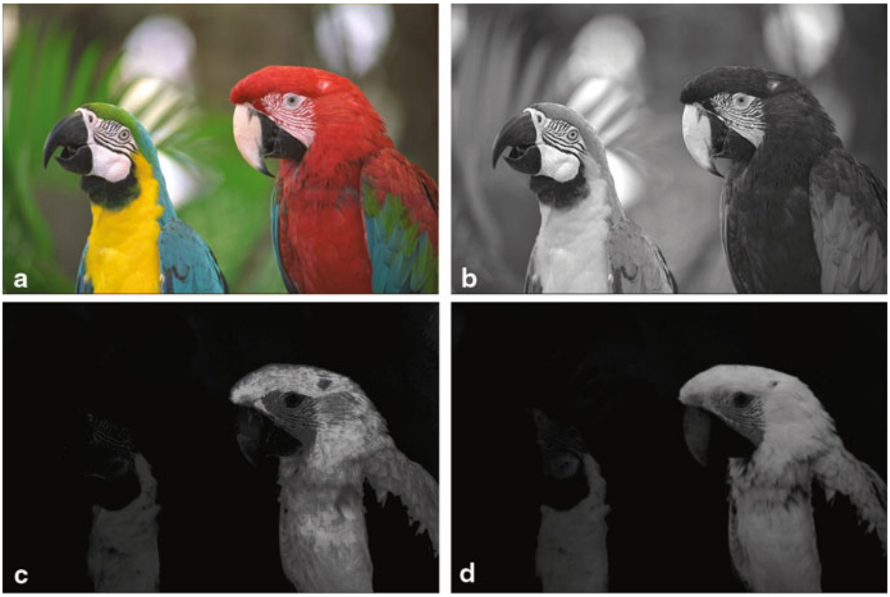


Fig. 2.7 Compare ratio image and difference image. **a** Original image, **b** Green plane, **c** R/G ratio image, **d** $R - G$ difference image

This is because the inter-spectral correlation lies in the high-frequency spectrum and consequently, the difference image of two color planes contains low-frequency components only. Generally, the color difference presents some benefits in comparison to the color ratio. The latter is indeed error-prone when its denominator takes a low value. This happens, for instance, when saturated red/blue components lead to comparatively low values of green, making the ratio very sensitive to the small variations in the red/blue plane. Figure 2.7a shows a natural image which is highly saturated in red. The corresponding green plane G, ratio image R/G and difference image $R - G$ are given in Figs. 2.7b–2.7d respectively. It can be noticed that the ratio and difference images carry out less high-frequency information than the green plane. Moreover, in areas where red is saturated, the ratio image contains more high-frequency information than the difference image, which makes the interpolation result more artifact-prone [35].

The spatial correlation reflects the fact that within a homogeneous image region, neighboring pixels share similar color values [10]. One could use this principle to estimate the missing color components at any pixel location except the pixels near the edge since these pixels have neighbors which do not belong to the same homogeneous region. Therefore, the following assumption is proposed based on the spatial correlation [59]:

- The rate of change of neighboring pixel values along an edge direction is a constant. For example, the pixels along horizontal edges satisfy:

$$\begin{aligned} R_{i,j} - R_{i,j+1} = R_{i,j+1} - R_{i,j+2} = dR \\ G_{i,j} - G_{i,j+1} = G_{i,j+1} - G_{i,j+2} = dG \\ B_{i,j} - B_{i,j+1} = B_{i,j+1} - B_{i,j+2} = dB \end{aligned} \tag{2.12}$$

  where dR, dG and dB are constants.

Following this assumption, many demosaicing methods first analyze the spatial structure of a local image neighborhood and then select a suitable direction for interpolation.

Depending on how the two correlations are exploited, existing demosaicing methods can be grouped into four classes [11]. The methods in the first class exploit neither correlation, applying the same interpolation scheme in each individual color plane, such as bilinear interpolation, nearest-neighbor replication, and cubic spline interpolation. The methods in the second class mainly exploit spatial correlation but little or no spectral correlation; they usually apply some adaptive interpolation scheme in each color plane separately. Examples of this class include Cok's pattern recognition interpolation (PRI) [14] and Adam's edge-sensing (ES) method [2]. Since this class does not fully utilize the spectral correlation, the methods in this class often result in excessive false colors. The methods in the third class mainly exploit image spectral correlation, including Cok's constant-hue interpolation [15], Freeman's median interpolation [20], Pei's effective color interpolation (ECI) [54], and Gunturk's alternating projections method (AP) [24]. Although capable of alleviating false color artifacts, these methods normally produce visible zipper effects around edges and details due to less usage of spatial correlation. The methods of the last class exploit both spatial and spectral correlations. Examples are Li's new edge-directed interpolation [32], Hamilton's adaptive color plane interpolation (ACPI) [5], Wu's primary-consistent soft-decision method (PCSD) [61], Hirakawa's adaptive homogeneity-directed demosaicing algorithm (AHD) [27], and so on.

In addition to the above classification, the demosaicing methods could also be divided into frequency-domain and spatial-domain [31], heuristic and nonheuristic [13], iterative and noniterative [57]. These classifications represent most demosaicing algorithms, but they are too general to capture each algorithm's main characteristics. Therefore, in the next section we will learn from Menon [47] and describe five representative methods to give readers a more comprehensive introduction of the existing demosaicing algorithms.

2.1.4 Evaluation Criteria

The common process for evaluating demosaicing algorithms consists of choosing color images that are captured using highly professional three-sensor cameras or

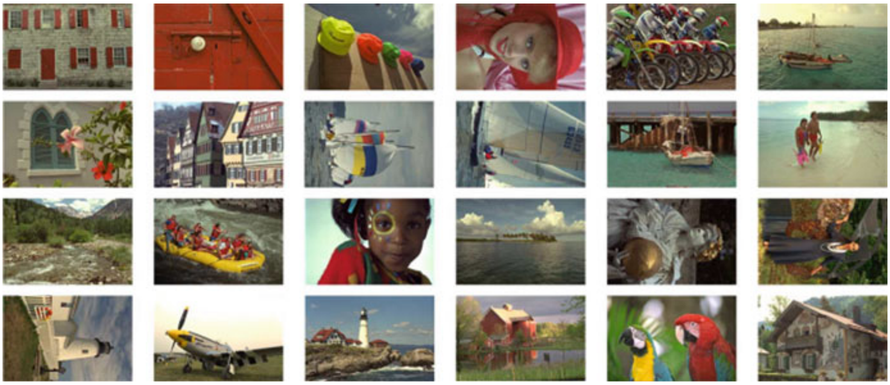


Fig. 2.8 Kodak image database. (These images are referred as Image 1 to Image 24 from *left* to *right* and *top* to *bottom*.)

color scanners, sampling them according to the Bayer CFA pattern to obtain mosaic images, interpolating the mosaic images back to full color images, and comparing the results with the original images [47]. This subsection will discuss the first and last step of the evaluation process.

Most work in the literature uses the Kodak image database [33] shown in Fig. 2.8 as a benchmark for performance comparison. The 24 images in this database are film captured and then digitized at the resolution of 512 × 768 with 8-bit depth per color component. The popularity of the Kodak image database is mainly due to the fact that the database contains natural real-life scenes and varies in complexity and color appearances. To increase the test difficulty, Li et al. [31] included a set of IMAX images with varying-hue and high-saturation edges and Lian et al. [34] added several classical images which are often used in other image processing fields. In addition to the real images, some synthetic images, such as starburst [36] and circular zone plate [39] shown in Figs. 2.9a and 2.9b respectively, were used as well to test the ability of the demosaicing algorithms in handling edges of various orientations and spatial resolutions.

In order to evaluate the demosaiced image, the Mean Square Error (MSE) is widely considered [35, 47, 57]. This criterion measures the mean quadratic error between the original image and the demosaiced image in each color plane. It is defined as:

$$MSE(k) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(\hat{I}_k(i,j) - I_k(i,j))^2 \qquad (2.13)$$

where $I_k(i,j)$ is a color component in the original image and $\hat{I}_k(i,j)$ is the corresponding color component in the demosaiced image, $k = R, G, B$. The MSE

Springer

http://www.springer.com/978-3-319-09362-8

Color Image and Video Enhancement
Celebi, M.E.; Lecca, M.; Smolka, B. (Eds.)
2015, XIII, 393 p. 204 illus., 184 illus. in color.,
Hardcover
ISBN: 978-3-319-09362-8