

## Chapter 16

# Single-Sensor Camera Image Processing

RASTISLAV LUKAC and KONSTANTINOS N. PLATANIOTIS

---

### 16.1 Introduction

In recent years a massive research and development effort has been witnessed in color imaging technologies in both the industry and the ordinary life. Color is commonly used in television, computer displays, cinema motion pictures, print, and photographs. In all these application areas the perception of color is paramount for the correct understanding and dissemination of the visual information. Recent technological advances have reduced the complexity and the cost of color devices, such as monitors, printers, scanners, and copiers, thus allowing their use in the office and home environment. However, it is the extreme and still increasing popularity of the consumer, single-sensor digital cameras which boosts today the research activities in the field of digital color image acquisition, processing, and storage. Single-sensor camera image processing methods are becoming increasingly important due to the development and proliferation of emerging digital camera-based applications and commercial devices, such as imaging enabled mobile phones and personal digital assistants, sensor networks, surveillance, and automotive apparatus.

This chapter focuses on single-sensor camera image processing techniques with particular emphasis on image interpolation based solutions. The chapter surveys in a systematic and comprehensive manner demosaicking, demosaicked image postprocessing and camera image zooming solutions which utilize data-adaptive and spectral modelling principles to produce camera images with an enhanced visual quality. The existence of realistic and efficient design procedures and the variety of processing solutions developed through the presented data-adaptive, spectral model-based processing framework make this family of processing methodologies an indispensable tool for single-sensor imaging.

The chapter begins with Section 16.2 which briefly discusses the digital camera solutions, emphasizing cost-effective hardware architecture for a consumer-grade camera equipped with a color filter array placed on the top of a single image sensor. The most common color filter array layouts are also introduced and commented upon.

The next part of the chapter, Section 16.3, describes the work-flow in the single-sensor cameras. Particular emphasis is placed on the essential camera image processing of the sensor data. The second part of this section presents various image processing paradigms which are taxonomized according to their ability to follow the structural and the spectral characteristics of the acquired image. This section also introduces the so-called generalized camera image processing solution and lists the spatial, structural, and spectral constraints imposed on such a solution in practical applications. This part of the chapter also includes Section 16.4 which focuses on edge-sensing processing mechanism and Section 16.5 which targets the essential spectral model for single-sensor image processing. The omission of either of these elements during the single-sensor imaging operations results in significant degradation of the visual quality of the full-color camera image.

The main part of the chapter is devoted to single-sensor imaging solutions developed using the concept of image interpolation. The connection between image interpolation operations and the single-sensor imaging solutions which use the edge-sensing mechanism and spectral model is highlighted. Examples and experimental results included in the chapter indicate that the framework is computationally attractive, yields good performance and produces images of reasonable visual quality. In this part of the chapter, Section 16.6 focuses on the demosaicking process or spectral interpolation which is used to generate a color image from a single-sensor reading and is an integral processing step in a single-sensor camera pipeline. Since demosaicked images often suffer from reduced sharpness, color shifts and visual artifacts, the use of demosaicked image postprocessing or full-color image enhancement is often utilized in order to enhance fine details and produce natural colors. This processing step is extensively analyzed and documented in Section 16.7. Finally, image spatial resolution expansion is often implemented in digital cameras with limited optical zooming capabilities. To this end, Section 16.8 focuses on camera image zooming or spatial interpolation which is to be performed either in the demosaicked image domain or directly on the sensor data. Although based on the same principle, these zooming approaches produce the different visual quality and have the different computational complexity.

The chapter concludes with Section 16.9. The section summarizes the main single-sensor camera image processing ideas.

## 16.2 Digital Camera Architectures

Digital color cameras [1], [2], [3] capture color images of real-life scenes electronically using an image sensor, usually a charge-coupled device (CCD) [4], [5] or complementary metal oxide semiconductor (CMOS) [6], [7] sensor, instead of the film used in the conventional analog cameras. Therefore, captured photos can be immediately viewed by the user on the digital camera's display, and immediately stored, processed, or transmitted. Without any doubt this is one of the most attractive features of digital imaging.

The captured photo can be considered a  $K_1 \times K_2$  Red-Green-Blue (RGB) digital color image  $\mathbf{x} : Z^2 \rightarrow Z^3$  representing a two-dimensional matrix of three-component samples (pixels)  $\mathbf{x}_{(r,s)} = [x_{(r,s)1}, x_{(r,s)2}, x_{(r,s)3}]^T$ , where each individual channel of  $\mathbf{x}$  can be considered a  $K_1 \times K_2$  monochrome image  $x_k : Z^2 \rightarrow Z$ , for  $k = 1, 2, 3$ . The pixel  $\mathbf{x}_{(r,s)}$  represents the color vector [8] occupying the spatial location  $(r, s)$ , with  $r = 1, 2, \dots, K_1$  and  $s = 1, 2, \dots, K_2$  denoting the image row and column, respectively. The value of the R ( $k = 1$ ), G ( $k = 2$ ), and B ( $k = 3$ ) component  $x_{(r,s)k}$  defined in the integer domain  $Z$  is equal to an arbitrary integer value ranged from 0 to 255 in a standard 8-bits per component representation and denotes the contribution of the  $k$ -th primary in the color vector  $\mathbf{x}_{(r,s)}$ . The process of displaying an image creates a graphical representation of the image matrix where the pixel values represent particular colors in the visible spectrum (Figure 16.1) [8].<sup>1</sup>

Due to the monochromatic nature of the image sensor, digital camera manufacturers implement several solutions to capture the color nature of the true visual scene. The following three are the most popular designs currently in use.

- **Three-sensor device** (Figure 16.2) [1], [2], [3]: This architecture acquires color information using a beam splitter to separate incoming light onto three optical paths. Each path has its own Red, Green or Blue color filter having different spectral transmittances and sensor for sampling the filtered light. Since the camera color image is obtained by registering the signals from three sensors, precise mechanical and optical alignment is necessary to maintain correspondence among the images obtained from the different channels [1]. Besides the difficulties of maintaining image registration, the high cost of the sensor<sup>2</sup> and the use of a beam splitter make the three-sensor architecture available only for some professional digital cameras.
- **X3 technology-based device** (Figure 16.3) [9], [10], [11]: This architecture uses a layered single image sensor which directly captures the complete color information at each spatial location in an image during a single exposure. Color filters are stacked vertically and ordered according to the energy of the photons absorbed by silicon. Taking advantage of the natural light absorbing characteristics of

<sup>1</sup>The figures in this chapter can be seen in color at <http://www.dsp.utoronto.ca/~lukacr/CRCcip/CRCchapLukac.pdf>

<sup>2</sup>The sensor is the most expensive component of the digital camera and usually takes from 10% to 25% of the total cost [3].

silicon, each sensor cell records color channels depth-wise in silicon.<sup>3</sup> The drawbacks of the sensor are its noisy behavior (especially for the layers in the base of the chip) and relatively low sensitivity. Furthermore, the sensor usually suffers from intensive optical/electrical cross talk which, however, can be processed out for natural images by clever signal processing. Specialized X3 technology-based sensors have been used in various medical, scientific, and industrial applications.

- **Single-sensor device** (Figure 16.4) [2], [3]: This architecture reduces cost by placing a color filter array (CFA), which is a mosaic of color filters, on top of the conventional single CCD/CMOS image sensor to capture all three primary (RGB) colors at the same time. Each sensor cell has its own spectrally selective filter and thus, it stores only a single measurement. Therefore, the CFA image constitutes a mosaic-like gray-scale image with only one color element available in each pixel location. The two missing colors must be determined from the adjacent pixels using a digital processing solution called the demosaicking process [2], [12], [13]. Such an architecture represents the most cost-effective method currently in use for color imaging and for this reason is almost universally utilized in consumer-grade digital cameras.

### 16.2.1 Consumer-Grade Camera Hardware Architecture

Following the block scheme shown in Figure 16.5, the camera acquires the scene by first focusing and transmitting light through the CFA. To reduce the various artifacts present in the demosaicked image due to down-sampling the color information performed by the CFA, a blurring filter is placed in the optical path [14]. Because of the presence of blurring filters in a blocking system, both sharpness and resolution of the image captured using consumer-grade camera is usually lower compared with the architectures depicted in Figures 16.2 and 16.3, [10], [14].

The acquired visual information is sampled using the image sensor and an analog-to-digital (A/D) converter. The DRAM buffer temporally stores the digital data from the A/D converter and then passes them to the application-specific integrated circuit (ASIC) which together with the microprocessor realize the digital data processing operations, such as demosaicking and image resizing. The firmware memory holds the set of instructions for the microprocessor, and along with the ASIC they are most distinctive elements between the camera manufactures.

After demosaicking has been completed, the digital image is displayed and stored in memory, often the 16 MB built-in memory or the optional memory supplied by the manufacturers usually in the form of memory stick cards of various types and storage capacities. The end-user is often offered the option to

---

<sup>3</sup>The light directed to the sensor is absorbed first by the Blue layer, then by the Green, and finally by the Red layer placed deepest in silicon [9].

re-process stored images by passing them back to the ASIC unit. The interested reader should refer to [14], [15], [16] for additional information on the camera optical and hardware components.

### 16.2.2 Color Filter Array (CFA)

Apart from camera firmware, another distinctive element of the consumer-grade digital cameras is the CFA. The type of a color system and the arrangements of the color filters in the CFA significantly vary depending on the manufacturer.

The color systems used in the various CFA designs can be divided into [17]: i) tri-stimulus (RGB, YMC) systems, ii) systems with mixed primary/complementary colors (e.g. MGCY pattern), and iii) four and more color systems (e.g. those constructed using white and/or colors with shifted spectral sensitivity). Although the last two variants may produce more accurate hue gamut compared to tri-stimulus systems, they often limit the useful range of the darker colors [18]. In addition, the utilization of four or more colors in the CFA layout increases computational complexity of the demosaicking operations. For this reason and due to the fact that color images are commonly stored in RGB color format the tri-stimulus RGB based CFAs are widely used by camera manufacturers.

A number of RGB CFAs with the varied arrangement of color filters in the array are used in practice [17], [18], [19], [20], [21]. Key factors which influence this design issue relate to [17]: i) cost-effective image reconstruction, ii) immunity to color artifacts and color moiré, iii) reaction of the array to image sensor imperfections, and iv) immunity to optical/electrical cross talk between neighboring pixels.

The first criterion is essential because of the real-time processing constraints imposed on the digital camera. Among the various CFAs shown in Figure 16.6, periodic CFAs such as the Bayer and Yamanaka patterns enjoy a distinct advantage over pseudo-random (or random) CFAs, since the aperiodic nature of the later makes the demosaicking process more complex [22]. To reduce the array's sensitivity to color artifacts in the reconstructed image, each pixel of the CFA image should be surrounded in its neighborhood by all three primary colors [18], as it can be seen in the arrays shown in Figures 16.6(c)-(e). On the other hand, images captured using periodic CFAs (e.g. Bayer and Yamanaka patterns) usually suffer from color moiré effects, which may be reduced by using pseudo-random (or random) CFAs instead [17]. It should be also mentioned that the wavelength of the G color band is close to the peak of the human luminance frequency response and thus, some CFAs [Figures 16.6(a), (b) and (e)] use the double number of G color filters than that of the R and B filters to reduce the amount of spectral artifacts in the outputted, reconstructed image [23], [24].

CFA designs, such as the diagonal stripe pattern and the diagonal Bayer pattern are considered to be the best in terms of their robustness against image sensor imperfections since defects are typically observed

along rows or columns of the sensor cells. With respect to immunity to optical/electrical cross talk between neighboring pixels,<sup>4</sup> CFAs with the fixed number of neighbors corresponding to each of the three primary colors significantly outperform pseudo-random CFA layouts.

As it can be seen from the above discussion, there is no CFA which satisfies all design constraints. Therefore, camera manufacturers usually select a CFA layout by taking into consideration cost, compatibility with other processing elements, and hardware constraints. Particular attention is devoted to the type and resolution of the image sensor, camera optical system, and image processing capabilities of the device. The intended application (e.g. consumer photography, surveillance, astronomy), in which the single-sensor device will be used, is also an important factor to be considered.

### 16.3 Camera Image Processing

After acquiring the raw sensor data, the image is preprocessed, processed and postprocessed to produce a faithful digital representation of the captured scene. Figure 16.7 depicts an example flow of the processing steps performed in the consumer-grade camera pipeline. Note that the depicted flow may significantly vary depending on the manufacturer [14], [15].

The preprocessing starts with defective pixel detection and correction of missing pixels by interpolating using the neighboring data. In this way, impairments caused by the failure of certain photo-elements in the sensor can be rectified. If the captured data resides in a nonlinear space, often the case due to the electronics involved, then a linearization step is usually implemented. The subsequent dark current compensation reduces the level of dark current noise introduced into the signal through thermally generated electrons in the sensor substrate. This processing step is essential in low exposure images where both signal and noise levels may be comparable. The last preprocessing step is the so-called white balance which is used to correct the captured image for the scene illuminant by adjusting the image values, thus recovering the true scene coloration. The interested reader can find an overview of the preprocessing steps in [14], [15], [16].

After preprocessing, the consumer-grade digital camera performs intensive image processing. Apart from demosaicking which is a mandatory and integral step in the processing pipeline needed in restoring the color information originally present in the captured image, camera image processing includes some optional steps. For instance, the visual quality of the restored color image can be improved by postprocessing the demosaicked image, whereas the spatial resolution of the captured image can be enhanced by zooming the CFA image and/or (postprocessed) demosaicked image. It will be shown in this chapter that these four camera image processing steps are fundamentally different, although they employ similar, if not identical, digital signal processing concepts.

---

<sup>4</sup>Diagonally located neighbors have a lower crosstalk contribution than the vertically/horizontally located neighbors [18].

Once the restoration of the full-color information is completed, the captured image enters the postprocessing stage in order to enhance its coloration and transform it to an output color encoding, appropriate for displaying and printing purposes [14]. Since the spectral sensitivities of the camera are not identical to the human color matching function [16], the color correction process adjusts the values of color pixels from those corresponding to accurate scene reproduction to those corresponding to visually pleasing scene reproduction. The subsequent tone scale rendering process transforms the color image from the unrendered spaces where a 12-16 bit representation was used for calculations to a rendered (mostly sRGB [25]) space with 8-bit representation, as it is required by most output media. The obtained image is then enhanced by sharpening/denoising in order to reduce the low-frequency content in the image and remove insignificant, noise-like details. An overview of the postprocessing steps can be found in [14], [15], [16].

The consumer-grade cameras (Figure 16.7) commonly store the rendered/enhanced color image in a compressed format using the Joint Photographic Experts Group (JPEG) standard [14]. However in recent years, the Exchangeable Image File (EXIF) format [26], [27] has been popularized due to its convenient implementation and the possibility to store additional (metadata) information regarding the camera and the environment. High-end single-sensor digital cameras (Figure 16.8) apply image compression onto the preprocessed image, and then the compressed image data are formatted and stored in a Tagged Image File Format for Electronic Photography (TIFF-EP) [14], [16]. In this format, the image is stored along with additional information, such as the details about camera setting, spectral sensitivities, and illuminant used. Additional information about the methods used for compression and storage of camera images can be found in [28], [29], [30], [31], [32], [33].

As it can be seen in Figures 16.7 and 16.8, the camera image processing can be implemented in i) a conventional digital camera which stores the demosaicked (RGB) output, or in ii) a companion personal computer (PC) which interfaces with the digital camera which stores the images in the CFA-like format. Both above processing pipelines can use the same processing solution. However, the approach depicted in Figure 16.8 allows for the utilization of sophisticated solutions which cannot, due to their complexity, be embedded in the conventional camera image processing pipeline (Figure 16.7) which has to operate under real-time constraints. In addition, due to the utilization of the companion PC in the pipeline the end-user can select different settings of the processing solutions and re-process the image until certain quality criteria are met. Finally, it should be mentioned that Figures 16.7 and 16.8 depict the basic camera work-flow. Both pipelines are flexible enough to accommodate various image, video and multimedia processing operations, such as video demosaicking [34], CFA video compression [35], and digital rights management (e.g. image indexing by embedding the metadata information into the camera image to allow for securing, and easy organization/retrieval of the captured images in the personal or public databases) [36], [37].

### 16.3.1 CFA Data Imaging

For the sake of clarity in the terminology used in the rest of the chapter it should be emphasized here that the image outputted by the preprocessing module in Figures 16.7 and 16.8 is called hereafter the CFA image, which is a  $K_1 \times K_2$  gray-scale mosaic-like image  $z : Z^2 \rightarrow Z$  with the CFA (scalar) data  $z_{(r,s)}$ . Since information about the arrangements of color filters in the actual CFA is readily available either from the camera manufacturer (Figure 16.7) or obtained from the TIFF-EP format (Figure 16.8), the CFA image  $z$  shown in Figure 16.9(a) can be transformed to a  $K_1 \times K_2$  three-channel (color) image  $\mathbf{x} : Z^2 \rightarrow Z^3$  shown in Figure 16.9(b) with the RGB pixels  $\mathbf{x}_{(r,s)} = [x_{(r,s)1}, x_{(r,s)2}, x_{(r,s)3}]^T$ .

The correspondence between the  $z_{(r,s)}$  values and the R ( $k = 1$ ), G ( $k = 2$ ), or B ( $k = 3$ ) color filters at the  $(r, s)$  CFA locations can be indicated using spatial location flags [22]. Following the dimensions of the CFA image  $z$ , a  $K_1 \times K_2$  vectorial field  $\mathbf{d} : Z^2 \rightarrow Z^3$  of the corresponding location flags  $d_{(r,s)k}$  is initialized using the default value  $d_{(r,s)k} = 1$  to indicate the presence of a CFA value  $z_{(r,s)}$  in the color vector  $\mathbf{x}_{(r,s)}$  for the proper value of  $k$ . For example, if  $(r, s)$  corresponds to a G CFA location in the image  $z$ , then  $\mathbf{x}_{(r,s)} = [0, z_{(r,s)}, 0]^T$  and  $d_{(r,s)k} = 1$  for  $k = 2$  should be used. If  $(r, s)$  corresponds to a R (or B) CFA location, then  $\mathbf{x}_{(r,s)} = [z_{(r,s)}, 0, 0]^T$  (or  $\mathbf{x}_{(r,s)} = [0, 0, z_{(r,s)}]^T$ ) and  $d_{(r,s)k} = 1$  for  $k = 1$  (or  $k = 3$ ) should be utilized. In all other cases, the flags are set to  $d_{(r,s)k} = 0$  indicating the two missing components in  $\mathbf{x}_{(r,s)}$  are set equal to zero to denote their portion to the coloration of the image  $\mathbf{x}$  shown in Figure 16.9(b). Note that the presented approach is independent from the CFA structure and thus suitable for an arbitrary CFA shown in Figure 16.6, [22].

### 16.3.2 Structural Information-Based Image Processing

Operating on the CFA image  $z$  or its colored variant  $\mathbf{x}$ , processing solutions included in the camera image processing module in Figures 16.7 and 16.8 can be classified considering the image content (structure) into the two basic paradigms shown in Figure 16.10: (a) non-adaptive processing, and (b) data-adaptive processing .

The former paradigm [see Figure 16.10(a)] uses no data-adaptive control to follow the structural content of the captured image [38], [39], [40], [41], [42], thus often reducing to a linear processing cycle which is easy to implement. However, it is well-known that most image processing tasks cannot be efficiently accomplished by linear techniques. Image signals are nonlinear in nature due to the presence of edges and thus, most of the linear and non-adaptive techniques tend to blur structural elements such as fine image details [43]. It should also be mentioned that images are perceived through the human visual system, which has strong nonlinear characteristics [44].

The latter paradigm (see Figure 16.10(b)) uses the so-called edge-sensing weights to follow structural

content of the image [45], [46], [47]. Such a processing solution, often called nonlinear processing, usually results in enhanced performance. Nonlinear, data-adaptive methods are able to preserve important color structural elements and produce sharply looking images.

### 16.3.3 Spectral Information-Based Image Processing

With respect to spectral characteristics, camera image processing can be divided into (Figure 16.11): (a) component-wise processing, (b) spectral model-based processing, and (c) vector processing.

Component-wise methods [Figure 16.11(a)], directly adopted from the gray-scale imaging, process each channel of the color image separately [38], [39], [48]. By omitting the essential spectral information and thus introducing a certain inaccuracy in the component-wise estimates, the projection of the outputted color components into the restored RGB image often produces color artifacts (a new color quite different from the neighbors) [8]. On the other hand, component-wise processing methods are usually fast and easy to implement.

Spectral model-based processing methods [Figure 16.11(b)] use the essential spectral information from the input camera image to eliminate color shifts and artifacts in the image to be outputted [2], [49], [50], [51]. This processing paradigm assumes vector (multi-channel) samples as the input and generates a single-channel output. Therefore, the procedure has to be repeated for each color channel in order to generate the full-color output. Due to its computational simplicity, the spectral model-based solutions are the most widely used in camera image processing among the paradigms shown in Figure 16.11.

Similarly to the spectral model-based paradigm, vector processing methods [Figure 16.11(c)] utilize the inherent correlation amongst the color channels and process the color image pixels as vectors [8], [43]. In this way, color artifacts in the output image are greatly reduced [8]. However, since vector processing methods generate the output color image in a single pass, they are usually computationally expensive compared with the solutions depicted in Figures 16.11(a) and (b). Therefore, the use of vector processing in camera image processing is rather limited at the moment.

### 16.3.4 Generalized Camera Image Processing Solution

From the discussion in Sections 16.3.2-16.3.3 is evident that an ideal processing solution should be able to follow both the structural content of the image and the spectral characteristics of its color content. To produce visually pleasing output images, it is essential to overcome both [52]: i) the CFA limitations and ii) the spatial, structural and spectral constraints imposed on the processing solution.

Figure 16.12 depicts the block diagram of the so-called generalized processing solution [2], [24], [36]. The characteristics of such a solution are essentially determined by the edge-sensing mechanism (ESM) and

the spectral model (SM) which are implemented through the operators  $\Lambda$  and  $\Psi$ , respectively. The ESM operator  $\Lambda$  uses both the structural and the spatial characteristics to preserve the sharpness and structural information of the captured image. The SM operator  $\Psi$  uses both the spectral and the spatial characteristics of the neighboring color pixels to eliminate spectral artifacts in the outputted signal. Thus, both the ESM and SM are based on the local neighborhood area which is determined by parameter  $\zeta$ .

Spatial constraints imposed on the processing solution relate to the size of an area of support and the form of the shape-mask utilized in processing. By denoting  $(r, s)$  as the location under consideration, due to the strong spatial correlation among the neighboring image samples of the natural image it is commonly expected that the utilization of the closest  $3 \times 3$  neighborhood  $\{(r \pm u, s \pm v); u, v \in \{-1, 0, 1\}\}$  ensures the faithful reconstruction of the color vector  $\mathbf{x}_{(r,s)}$ . Since the CFA image has a mosaic-like structure and not all of the neighboring locations correspond to a color channel being reconstructed, operating on the image  $\mathbf{x}$  obtained using the well-known Bayer CFA [see Figure 16.13(a)] the local neighborhood described by  $\zeta$  is most commonly limited to the shape-masks shown in Figures 16.13(b)-(e), [45], [47], [53].

To quantify the contributions of the adjacent color vectors  $\mathbf{x}_{(i,j)} = [x_{(i,j)1}, x_{(i,j)2}, x_{(i,j)3}]^T$  to the vector  $\mathbf{x}_{(r,s)}$  under consideration, the so-called data-adaptive concept is used as follows [2], [51], [52]:

$$\mathbf{x}_{(r,s)} = \sum_{(i,j) \in \zeta} \{w'_{(i,j)} \Psi(\mathbf{x}_{(i,j)}, \mathbf{x}_{(r,s)})\} \quad (16.1)$$

where  $(i, j) \in \zeta$  denotes the spatial location arrangements on the image lattice, e.g.  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  shown in Figures 16.13(b),(d),(e) and  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  shown in Figure 16.13(c). Each of the available color vectors  $\mathbf{x}_{(i,j)}$  inside  $\zeta$  is associated with the normalized weighting coefficient  $w'_{(i,j)}$  defined as:

$$w'_{(i,j)} = w_{(i,j)} / \sum_{(i,j) \in \zeta} w_{(i,j)} \quad (16.2)$$

where  $w_{(i,j)}$  are the so-called edge-sensing weights. By emphasizing inputs which are not positioned across an edge and directing the processing along the natural edges in the true image, the use of the edge information in the data-adaptive formula in (16.1) preserves the structural image contents, and thus ensures a sharply formed output image.

## 16.4 Edge-Sensing Mechanism (ESM)

Structural constraints of (16.1) relate to the form of the ESM operator  $\Lambda$  used to generate the weights in (16.2) as  $\{w_{(i,j)}; (i, j) \in \zeta\} = \Lambda(\mathbf{x}, \zeta)$ . If the  $w_{(i,j)}$  values are not obtained through the function  $\Lambda(\cdot)$  of the color components available in the image  $\mathbf{x}$ , i.e. the fixed setting  $\{w_{(i,j)} = \xi; (i, j) \in \zeta\}$  to a non-zero constant  $\xi$  is used, the data-adaptive solution in (16.1) reduces to non-adaptive schemes, such as those listed in [38], [40], [42], [49].

However, such an omission of the structural information in (16.1) leads to the significant degradation of the visual quality [2], [47]. Therefore, most of the available designs employ the data-dependent ESM operator  $\Lambda(\mathbf{x}, \zeta)$  which uses some form of inverse gradient of the samples in the image  $\mathbf{x}$ . Large image gradients usually indicate that the corresponding vectors  $\mathbf{x}_{(i,j)}$  are located across edges. Weights  $\{w_{(i,j)}; (i,j) \in \zeta\} = \Lambda(\mathbf{x}, \zeta)$  are inversely proportional to gradient values, thus penalizing the corresponding inputs  $\mathbf{x}_{(i,j)}$  in (16.1). Since the original CFA components are readably available in the color image  $\mathbf{x}$  as explained in Section 16.3.1, in order to increase the accuracy of the adaptive design in (16.1) it is common to restrict  $\Lambda(\cdot)$  only on the acquired CFA data  $z_{(\cdot,\cdot)}$ , i.e. edge-sensing weights are obtained as  $\{w_{(i,j)}; (i,j) \in \zeta\} = \Lambda(z, \zeta)$ .

#### 16.4.1 Aggregation Concept-Based ESM

Cost considerations and real-time constraints in digital apparatus necessitate the utilization of a simple and easy to implement ESM  $\Lambda(z, \zeta)$ , such as the one defined as follows [2], [54]:

$$w_{(i,j)} = 1 / \sum_{(g,h) \in \zeta} |z_{(i,j)} - z_{(g,h)}| \quad (16.3)$$

where  $\sum_{(g,h) \in \zeta} |z_{(i,j)} - z_{(g,h)}|$  is the aggregated absolute difference (Figure 16.14) between the CFA components  $z_{(\cdot,\cdot)}$  inside the neighborhood  $\zeta$ . This approach significantly reduces memory requirements since both the ESM operator  $\Lambda$  and the data-adaptive estimator (16.1) use the same inputs occupying the spatial locations  $(i,j) \in \zeta$ . This also makes the ESM implementation in (16.3) independent from the CFA structure and thus flexible for the use in the imaging pipeline equipped with the CFA other than the Bayer CFA [22]. Thinking in these dimensions, different forms of the ESM  $\Lambda(z, \zeta)$  have been proposed in [2].

The framework allows for obtaining the edge-sensing weights in an automated manner (Figure 16.7), with the example of the ESM formulated in (16.3). In addition to this design, the weights can be controlled by the end-user, as shown in Figure 16.8 [2]. One of the possible solutions is the utilization of the well-known sigmoidal function. As empirical evidence suggests that the relationship between perception and distances measured in physical units is exponential in nature [55], the weights calculated using

$$w_{(i,j)} = \beta_{(i,j)} (1 + \exp\{\sum_{(g,h) \in \zeta} |z_{(i,j)} - z_{(g,h)}|\})^{-r} \quad (16.4)$$

can lead to some performance improvements in terms of the visual quality [2]. In the definition above,  $r$  is a parameter adjusting the weighting effect of the membership function and  $\beta_{(i,j)}$  is a normalizing constant. Within the framework, numerous solutions may be constructed by changing the way the weights are calculated as well as the way the available color components from different color channels are treated. The choice of these parameters determines the characteristics and influences the efficiency of the processing solution.

### 16.4.2 Neighborhood Expansion-Based ESM

On the other hand, the constraint imposed on the size of  $\zeta$  in  $\Lambda(z, \zeta)$  may reduce the edge-sensing capability of the solution. To overcome the limitation, unfortunately at the expense of an increased implementation complexity and memory requirements, the ESM operator  $\Lambda(z, \zeta)$  is modified to  $\Lambda(z, \zeta')$  by considering a larger, typically  $5 \times 5$  or  $7 \times 7$  neighborhood  $\zeta' = \{(r \pm u, s \pm v); u, v \in \{-2, 0, 2\}\}$  or  $\zeta' = \{(r \pm u, s \pm v); u, v \in \{-3, 0, 3\}\}$ , respectively. In this case, the weights  $\{w_{(i,j)}; (i, j) \in \zeta\} = \Lambda(z, \zeta')$ , for  $\zeta \subset \zeta'$ , are calculated using the components  $z_{(\cdot, \cdot)}$  in  $\zeta'$ . The described concept has been employed in various processing solutions, for example in [12], [45], [47], [46], [56], [57]. It should be mentioned that these ESMs may not be directly applicable to non-Bayer CFAs [22] as they implicitly differentiate amongst the color components corresponding to different color channels inside  $\zeta'$ .

To demonstrate the concept, the ESMs used in [52], [57] is described below. Employing a diamond-shape mask  $\zeta = \{(r-1, s), (r, s-1), (p, q+1), (p+1, q)\}$  in (16.1), the weights  $w_{(i,j)}$  for  $(i, j) \in \zeta$  are calculated using the original CFA components  $z_{(i', j')}$  for  $(i', j') \in \zeta'$  as follows [see Figure 16.15(a)]:

$$\begin{aligned} w_{(r-1,s)} &= 1/(1 + |z_{(r-2,s)} - z_{(r,s)}| + |z_{(r-1,s)} - z_{(r+1,s)}|) \\ w_{(r,s-1)} &= 1/(1 + |z_{(r,s-2)} - z_{(r,s)}| + |z_{(r,s-1)} - z_{(r,s+1)}|) \\ w_{(r,s+1)} &= 1/(1 + |z_{(r,s+2)} - z_{(r,s)}| + |z_{(r,s+1)} - z_{(r,s-1)}|) \\ w_{(r+1,s)} &= 1/(1 + |z_{(r+2,s)} - z_{(r,s)}| + |z_{(r+1,s)} - z_{(r-1,s)}|) \end{aligned} \quad (16.5)$$

Otherwise, a square-shape mask  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  is considered denoting the weights are obtained as [see Figure 16.15(b)]:

$$\begin{aligned} w_{(r-1,s-1)} &= 1/(1 + |z_{(r-2,s-2)} - z_{(r,s)}| + |z_{(r-1,s-1)} - z_{(r+1,s+1)}|) \\ w_{(r-1,s+1)} &= 1/(1 + |z_{(r-2,s+2)} - z_{(r,s)}| + |z_{(r-1,s+1)} - z_{(r+1,s-1)}|) \\ w_{(r+1,s-1)} &= 1/(1 + |z_{(r+2,s-2)} - z_{(r,s)}| + |z_{(r+1,s-1)} - z_{(r-1,s+1)}|) \\ w_{(r+1,s+1)} &= 1/(1 + |z_{(r+2,s+2)} - z_{(r,s)}| + |z_{(r+1,s+1)} - z_{(r-1,s-1)}|) \end{aligned} \quad (16.6)$$

At this point it should be mentioned that although a significant research effort has been devoted to the development of various ESMs, the latest advances in camera image processing clearly indicate that high-quality camera images are obtained only if the edge-sensing camera image processing solution employs a proper SM [2], [53].

## 16.5 Spectral Model (SM)

In the camera image processing environment, spectral constraints relate to the utilization of the essential spectral characteristics of the captured image during the image interpolation process [52]. Based on the assumption that a typical natural image exhibits significant spectral correlation among its RGB color planes

and spatial correlation among the neighboring pixels, a SM is used to support any interpolation operations performed in the single-sensor imaging pipeline.

Following the data-adaptive formula (16.1), the spectral characteristics of the two neighboring vectors  $\mathbf{x}_{(i,j)}$  and  $\mathbf{x}_{(r,s)}$  are incorporated into the interpolation process through the SM operator  $\Psi(\mathbf{x}_{(i,j)}, \mathbf{x}_{(r,s)})$ . If no spectral information is considered during processing, the spectral estimator in (16.1) reduces to component-wise processing:

$$x_{(r,s)k} = \sum_{(i,j) \in \zeta} \{w'_{(i,j)} x_{(i,j)k}\} \quad (16.7)$$

where  $x_{(\cdot,\cdot)k}$  denotes the  $k$ -th components of the color vector  $\mathbf{x}_{(\cdot,\cdot)}$ . By omitting the essential spectral information, the component-wise solutions in [38], [39], [48] produce various color shifts and spectral artifacts.

### 16.5.1 Modelling Assumption

The rationale behind the SM construction can be easily explained using the basic characteristics of the color vectors. According to the tristimulus theory of color representation, the three-dimensional RGB vector  $\mathbf{x}_{(r,s)} = [x_{(r,s)1}, x_{(r,s)2}, x_{(r,s)3}]^T$  is uniquely defined [8] by its length (magnitude)  $M_{\mathbf{x}_{(r,s)}} = \|\mathbf{x}_{(r,s)}\| = \sqrt{x_{(r,s)1}^2 + x_{(r,s)2}^2 + x_{(r,s)3}^2}$  and orientation (direction)  $D_{\mathbf{x}_{(r,s)}} = \mathbf{x}_{(r,s)} / \|\mathbf{x}_{(r,s)}\| = \mathbf{x}_{(r,s)} / M_{\mathbf{x}_{(r,s)}}$ , where  $\|D_{\mathbf{x}_{(r,s)}}\| = 1$  denotes the unit sphere defined in the vector space. Thus, any color image can be considered a vector field where each vector's direction and length are related to the pixel's color characteristics and influence significantly its perception by the human observer [8].

It is well-known that natural images consist of small regions which exhibit similar, if not identical, color-chromaticity properties [58], [59]. Since color-chromaticity relates to the color vectors' directional characteristics determined using  $D_{\mathbf{x}_{(\cdot,\cdot)}}$ , it is reasonable to assume that two color vectors  $\mathbf{x}_{(r,s)}$  and  $\mathbf{x}_{(i,j)}$  occupying spatially neighboring locations  $(r, s)$  and  $(i, j)$  have the same chromaticity characteristics if they are collinear in the RGB color space [52]. Based on the definition of dot product  $\mathbf{x}_{(r,s)} \cdot \mathbf{x}_{(i,j)} = \|\mathbf{x}_{(r,s)}\| \|\mathbf{x}_{(i,j)}\| \cos(\langle \mathbf{x}_{(r,s)}, \mathbf{x}_{(i,j)} \rangle)$ , where  $\|\mathbf{x}_{(\cdot,\cdot)}\|$  denotes the length of  $\mathbf{x}_{(\cdot,\cdot)}$  and  $\langle \mathbf{x}_{(r,s)}, \mathbf{x}_{(i,j)} \rangle$  denotes the angle between three-component color vectors  $\mathbf{x}_{(r,s)}$  and  $\mathbf{x}_{(i,j)}$ , the enforcement of orientation constraints via the SM operator  $\Psi(\mathbf{x}_{(i,j)}, \mathbf{x}_{(r,s)})$  in (16.1) implies that the following condition holds [52]:

$$\langle \mathbf{x}_{(r,s)}, \mathbf{x}_{(i,j)} \rangle = 0 \Leftrightarrow \frac{\sum_{k=1}^3 x_{(r,s)k} x_{(i,j)k}}{\sqrt{\sum_{k=1}^3 x_{(r,s)k}^2} \sqrt{\sum_{k=1}^3 x_{(i,j)k}^2}} = 1 \quad (16.8)$$

Since both the magnitude and directional characteristics of the color vectors are essential for the human perception, the above concept should be extended by incorporating the magnitude information  $M_{\mathbf{x}_{(\cdot,\cdot)}}$  into the modelling assumption. Using color vectors  $\mathbf{x}_{(i,j)}$  and  $\mathbf{x}_{(r,s)}$  as inputs to the so-called generalized vector SM [52], the underlying modelling principle of identical color chromaticity enforces that their linearly

shifted variants  $[\mathbf{x}_{(r,s)} + \gamma\mathbf{I}]$  and  $[\mathbf{x}_{(i,j)} + \gamma\mathbf{I}]$  are collinear vectors:

$$\langle \mathbf{x}_{(r,s)} + \gamma\mathbf{I}, \mathbf{x}_{(i,j)} + \gamma\mathbf{I} \rangle = 0 \Leftrightarrow \frac{\sum_{k=1}^3 (x_{(r,s)k} + \gamma)(x_{(i,j)k} + \gamma)}{\sqrt{\sum_{k=1}^3 (x_{(r,s)k} + \gamma)^2} \sqrt{\sum_{k=1}^3 (x_{(i,j)k} + \gamma)^2}} = 1 \quad (16.9)$$

where  $\mathbf{I}$  is a unity vector of proper dimensions and  $x_{(\cdot,\cdot)k} + \gamma$  is the  $k$ -th component of the linearly-shifted vector  $[\mathbf{x}_{(\cdot,\cdot)} + \gamma\mathbf{I}] = [x_{(\cdot,\cdot)1} + \gamma, x_{(\cdot,\cdot)2} + \gamma, x_{(\cdot,\cdot)3} + \gamma]^T$ .

### 16.5.2 Advanced Design and Performance Characteristics

Through the parameter  $\gamma$ , the model controls the influence of both the directional and the magnitude characteristics of the neighboring vectors in the camera image processing. The scale shift introduced by the parameter  $\gamma$  prevents color shifts in flat areas as well as near edge transitions. A number of processing solutions, with different design characteristics and performance, can be thus obtained by modifying the  $\gamma$  parameter. The interested reader can find extensive analysis and experimentation, as well as detailed guidelines for the use of the vector SMs in the single-sensor imaging pipeline in [52].

It is also proven in the same work [52] that the vector model defined over the two-dimensional vectors generalizes the previous SMs. Namely, for the parameter  $\gamma = 0$  the vector SM reduces to

$$\frac{x_{(r,s)k}}{x_{(i,j)k}} = \frac{x_{(r,s)2}}{x_{(i,j)2}} \quad (16.10)$$

which implements the basic modelling assumption behind the color-ratio model [49]. Using the ratio values in (16.10), the model enforces hue uniformity in localized image areas.

If  $\gamma \rightarrow \infty$  is considered, the vector SM generalizes the color-difference model which implies uniform image intensity in localized image areas [50]. In this case, the modelling assumption of uniform image intensity is enforced by constraining the component-wise magnitude differences between spatially neighboring color vectors as follows [52]:

$$x_{(p,q)k} - x_{(i,j)k} = x_{(p,q)2} - x_{(i,j)2} \quad (16.11)$$

Finally, through the simple vectors' dimensionality reduction the vector generalizes the normalized color-ratio model [51] defined as follows:

$$\frac{x_{(r,s)k} + \gamma}{x_{(i,j)k} + \gamma} = \frac{x_{(r,s)2} + \gamma}{x_{(i,j)2} + \gamma} \quad (16.12)$$

The model above enforces the basic modelling assumption of constant hue both near edge transitions and flat image areas, thus leading to the enhanced performance. For example, Figure 16.16 shows that the significant improvements of the visual quality are obtained when the color-ratio model is replaced in the well-known Kimmel demosaicking algorithm with the normalized color-ratio model [51]. Similar conclusions have been drawn in [2] where various SMs have been tested using the different data-adaptive demosaicking solutions.

## 16.6 Demosaicking

As it was mentioned in Section 16.5, demosaicking is an integral and probably the most common processing step used in single-sensor digital cameras [2], [12]. The objective of the demosaicking process is to estimate the missing color components of the vectorial pixels  $\mathbf{x}_{(r,s)}$  obtained using the intermediate step described in Section 16.3.1 and produce the demosaicked full-color image. Demosaicking performs spectral interpolation  $\mathbf{x} = f_\varphi(z)$  which transforms a  $K_1 \times K_2$  gray-scale image  $z : Z^2 \rightarrow Z$  shown in Figure 16.9(a) to a  $K_1 \times K_2$  three-channel, full color image  $\mathbf{x} : Z^2 \rightarrow Z^3$  depicted in Figure 16.17(a), [24], [60].

It is well-known that [8], [43]: i) natural images are non-stationary due to the edges and fine details, ii) a typical natural image exhibits significant (spectral) correlation among its RGB color planes, and iii) the spatially neighboring pixels are usually highly correlated. Thus, by operating in small localized image areas, each of which can be treated as stationary, a camera image processing solution can minimize local distortion in the outputted image by utilizing the spatial, structural and spectral characteristics during processing. Re-formulating the spectral interpolation function  $f_\varphi(z)$  as  $f_\varphi(\Lambda, \Psi, \zeta, z)$  with  $z$  denoting the acquired CFA image and  $\zeta$  determining the local neighborhood area [Figures 16.13(b)-(e)], the performance of  $f_\varphi(\cdot)$  critically depends on the choice of the ESM and SM operators  $\Lambda$  and  $\Psi$ , respectively [2], [52]. As shown in Figure 16.18, the omission of any of the operators  $\Lambda$  and  $\Psi$  during demosaicking results in excessive blur, color shifts and visible aliasing effects [53].

### 16.6.1 Bayer CFA-Based Demosaicking Procedure

Because of the double number of G elements in the Bayer CFA [Figure 16.13(a)], the conventional practice is to start the demosaicking process by estimating the missing G components in the pixel array [2], [12], [47], [42]. Following the discussion in Section 16.3.1, the  $(r, s)$  location to be demosaicked are readily indicated by the corresponding location flags  $d_{(r,s)k} = 0$ , with  $k = 2$  for the G components. In all such locations, the demosaicking solution updates the color vector  $\mathbf{x}_{(r,s)}$  using the component-wise data-adaptive formula (16.7) with  $k = 2$  denoting the G plane and  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  indicating the available neighboring locations [Figure 16.13(b)]. In the same time, the solution also updates the location flag for  $k = 2$  to  $d_{(r,s)k} = 2$  differentiating thus between the original CFA locations ( $d_{(r,s)k} = 1$ ), the demosaicked locations ( $d_{(r,s)k} = 2$ ), and the locations to be demosaicked ( $d_{(r,s)k} = 0$ ). The procedure is repeated until a fully populated G plane is obtained, i.e. until all the location flags are set to  $d_{(\cdot,\cdot)2} \neq 0$ .

Since the above processing step makes the G components available at the R and B CFA locations, the missing R and B components can be demosaicked using the spectral model based formula (16.1). For example, by adopting the color difference model (16.11), the generalized spectral interpolator in (16.1) is modified to demosaick the R ( $k = 1$ ) or B ( $k = 3$ ) component  $x_{(r,s)k}$  occupying the location  $(r, s)$  with

$d_{(r,s)k} = 0$  as follows:

$$x_{(r,s)k} = x_{(r,s)2} + \sum_{(i,j) \in \zeta} \{w'_{(i,j)}(x_{(i,j)k} - x_{(i,j)2})\} \quad (16.13)$$

where  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  denotes the available locations as shown in Figure 16.13(c). The procedure successively updates the flags to  $d_{(r,s)k} = 2$  for  $k = 1$  and  $k = 3$  at each location  $(r, s)$  which is being demosaicked via (16.13). In the sequence, (16.13) should be re-applied with  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  shown in Figures 16.13(d) and (e) to fully populate both the R and B channels. After demosaicking all available locations and updating all the corresponding flags for  $k = 1$  and  $k = 3$  to  $d_{(\cdot,\cdot)k} \neq 0$  the demosaicking step produces a reconstructed color image, such as the one shown in Figure 16.17(a).

If the higher visual quality is required [Figure 16.17(b)], then both the sharpness and coloration of the demosaicked image can be enhanced by employing a postprocessor in the processing pipeline [60], [61]. Since the postprocessing step can be viewed as a stand-alone solution, as it is demonstrated in Fig. 16.7, the demosaicked image postprocessing (full-color image enhancement) related issues are discussed in Section 16.7.

## 16.6.2 Universal Demosaicking

By taking advantage of the CFA structure when the demosaicking solution is constructed, the solution becomes too dependent on the particular CFA used, such as the Bayer CFA [Figure 16.6(a)] in the procedure described above. The dependency mostly relates to the shape masks [Figures 16.13(b)-(e)] and the form of the ESM used during the processing, and it significantly reduces, if not removes altogether, the flexibility of the solution to be employed in the pipeline equipped with a non-Bayer CFA. For instance, the traditional forms of the ESM listed in [12] or the one described in Section 16.4.2 cannot be used in conjunction with the CFAs shown in Figures 16.6(c)-(h) since each subtraction operation in the gradient calculations requires the presence of the two components from the same color channel. To eliminate these drawbacks, a processing solution must be independent from the arrangement of the color filters in the CFA. This constraint has been addressed in the framework first introduced and analyzed in [22].

The framework employs a  $3 \times 3$  sliding window instead of the specialized shape masks [Figures 16.13(b)-(e)] known from the Bayer CFA-based demosaicking procedures. In addition, demosaicking the  $k$ -th channel of the color image  $\mathbf{x}$ , the framework uses the control mechanism over the demosaicking steps in (16.7) and (16.13) to prevent from operating in areas which lack adequate input information [22]. Note that the number of color components corresponding to particular color channels and their locations in the processing window vary not only between the CFAs, but often inside the CFA itself (e.g. for pseudo-random CFAs). Therefore, the framework tracks the structural image characteristics using the aggregation concept-based ESM in (16.3)

which obviously constitutes a flexible ESM. The interested reader can find the detailed description of the universal demosaicking framework in [22].

Figure 16.19 demonstrates that the choice of the CFA may be one of the most important factors in designing the single-sensor imaging pipeline. Although the same processing solution has been employed to process the CFA image, results corresponding to the different CFAs differ significantly in terms of aliasing artifacts present in the demosaicked image.

## 16.7 Demosaicked Image Postprocessing

As it was mentioned in Section 16.6.1 and demonstrated in Figure 16.17, both the color appearance and the sharpness of the demosaicked image can be further improved by employing a postprocessor in the pipeline to localize and eliminate false colors and other impairments created during demosaicking [60], [61]. Postprocessing the demosaicked image is an optional step, implemented mainly in software and activated by the end-user. It performs full color image enhancement  $f_\eta(\Lambda, \Psi, \zeta, \mathbf{x})$ , since the input of the solution is a fully restored RGB color image  $\mathbf{x}$  and the output is an enhanced RGB color image. Unlike demosaicking, postprocessing can be applied iteratively until certain quality criteria are met.

Since there is no method to objectively determine whether or not a color component is inaccurate, the postprocessing framework should utilize the differences between the color components generated by a demosaicking solution and the original CFA components included in the restored color vector  $\mathbf{x}_{(r,s)}$  of the demosaicked image  $\mathbf{x}$ , [60], [61]. If the demosaicked image postprocessing directly follows the demosaicking step in the imaging pipeline, then the location flag values updated during demosaicking to  $d_{(r,s)k} = 2$  are restored to  $d_{(r,s)k} = 0$ , for  $r = 1, 2, \dots, K_1$ ,  $s = 1, 2, \dots, K_2$ , and  $k = 1, 2, 3$ , to guide the demosaicked image postprocessing step. If the camera stores the demosaicked image and the demosaicked image postprocessing is to be performed independently from the demosaicking step, then the proper location flags  $d_{(r,s)k}$  can be obtained from the metadata information stored in the EXIF format. Finally, it should be mentioned that only the components obtained using the demosaicking process are enhanced by the postprocessing process, i.e. original CFA components are kept unchanged.

Following the Bayer CFA-based demosaicking procedure in Section 16.6.1, postprocessing of the G color plane in all locations  $(r, s)$  with the constraints  $d_{(r,s)2} = 0$  can be realized using R or B components as follows [60]:

$$x_{(r,s)2} = x_{(r,s)k} + \sum_{(i,j) \in \zeta} \{w'_{(i,j)}(x_{(i,j)2} - x_{(i,j)k})\} \quad (16.14)$$

where  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  denotes the available G CFA locations [Figure 16.13(b)]. If  $(r, s)$  corresponds to the R CFA location ( $d_{(r,s)1} = 1$ ), then the parameter  $k = 1$  should be used in (16.14). Otherwise, (16.14) is used for the B CFA location ( $d_{(r,s)3} = 1$ ) and the pertinent pa-

parameter is  $k = 3$ . The weights  $w'_{(i,j)}$  can be obtained in (16.2) using an arbitrary ESM listed in [12] or in Sections 16.4.1 and 16.4.2.

After the G plane is enhanced, the postprocessing step is completed by enhancing the demosaicked R ( $k = 1$ ) and B ( $k = 3$ ) components using (16.13), first with  $\zeta = \{(r - 1, s - 1), (r - 1, s + 1), (r + 1, s - 1), (r + 1, s + 1)\}$  shown in Figure 16.13(c) and then with  $\zeta = \{(r - 1, s), (r, s - 1), (r, s + 1), (r + 1, s)\}$  shown in Figures 16.13(d) and (e). This step is performed for R (or B) components only in locations corresponding to G and B (or G and R) CFA values, i.e.  $d_{(r,s)1} = 0$  (or  $d_{(r,s)3} = 0$ ). The postprocessing process can lead to some significant improvements of the visual quality for most, if not all, demosaicking solutions (Figure 16.20) [60], [61]. The Bayer CFA-based postprocessing concepts can be easily extended using the control mechanisms described in Section 16.6.2 to complete the demosaicking process for an arbitrary CFA. Such a universal postprocessing solution has been introduced in [22].

## 16.8 Camera Image Zooming

Cost-effectiveness considerations in imaging-enabled mobile phones, surveillance, and automotive apparatus often lead to image outputs which do not have the sufficient spatial resolution for subsequent image analysis tasks, such as object recognition and scene interpretation [62]. Therefore, digital image zooming is required. In addition, along with the demosaicking, image zooming is the most commonly performed processing operation in single-sensor digital cameras. Image zooming or spatial interpolation of a digital image is the process of increasing the number of pixels representing the natural scene [8], [63]. Unlike spectral interpolation (demosaicking)  $f_\varphi(\cdot)$ , spatial interpolation  $f_\phi(\cdot)$  preserves the spectral representation of the input. Operating on the spatial domain of a digital image, spatial interpolation transforms [24]: i) a color image  $\mathbf{x}$  to an enlarged color image  $\mathbf{x}' = f_\phi(\mathbf{x})$ , or ii) a gray-scale image  $z$  to an enlarged gray-scale image  $z' = f_\phi(z)$ .

As it is shown in Figure 16.7, the spatial resolution of captured camera images can be increased by performing the spatial interpolation  $f_\phi(\cdot)$  using: i) the demosaicked image  $\mathbf{x}$  [see Figure 16.21(a)], or ii) the acquired, grayscale CFA image  $z$  [see Figure 16.21(b)]. The different order of demosaicking and zooming operations employed in the process significantly influences both the performance (Figure 16.22) and the cost of the processing pipeline [24], [62], although from the camera-user's perspective both cascades  $f_\phi^\tau(f_\varphi(z))$  and  $f_\varphi(f_\phi^\tau(z))$  transform a  $K_1 \times K_2$  CFA image  $z : Z^2 \rightarrow Z$  to an enlarged  $\tau K_1 \times \tau K_2$  demosaicked image  $\mathbf{x}' : Z^2 \rightarrow Z^3$ , where  $\tau \in Z$  is an integer zooming factor. Other differences result from the capability of a zooming solution to follow the structural content of the captured image and the way how the essential spectral information is treated during the processing.

### 16.8.1 Spatial Interpolation of Demosaicked Images

Since the consumer-grade cameras conventionally store the image in the demosaicked full-color format, the spatial resolution of the captured image is most often increased using a color image zooming technique operating on the demosaicked RGB color vectors. For example, assuming the zooming factor  $\tau = 2$ , the zooming procedure maps the demosaicked color vectors  $\mathbf{x}_{(\cdot,\cdot)}$  into the enlarged image  $\mathbf{x}'$  with the pixels  $\mathbf{x}'_{(\cdot,\cdot)}$  using  $\mathbf{x}'_{(2r-1,2s-1)} = \mathbf{x}_{(r,s)}$ , with  $r$  and  $s$  denoting the spatial coordinates in the demosaicked image [64]. Thus, the pixels  $\mathbf{x}'_{(2r,2s)}$  denote the new rows and columns (e.g. of zeros) added to the demosaicked image data  $\mathbf{x}_{(r,s)}$ , for  $r = 1, 2, \dots, K_1$  and  $s = 1, 2, \dots, K_2$ .

The procedure produces the three pixel configurations shown in Figures 16.23(a)-(c) [8], [64], however configurations depicted in Figures 16.23(b) and (c) lack the available components needed for faithful reconstruction of the enlarged image  $\mathbf{x}'$  with the pixels  $\mathbf{x}'_{(r,s)}$ , for  $r = 1, 2, \dots, 2K_1$  and  $s = 1, 2, \dots, 2K_2$ . Therefore, the procedure first interpolates the vector  $\mathbf{x}'_{(r,s)}$  located in the center of the four available neighboring vectors  $\{\mathbf{x}'_{(i,j)}, \text{ for } (i,j) \in \zeta\}$  with  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$ , as shown in Figure 16.23(a). Then, all configurations become similar to the one depicted in Figure 16.23(d), and the procedure interpolates the vector  $\mathbf{x}'_{(r,s)}$  situated in the center of the four neighboring vectors  $\{\mathbf{x}'_{(i,j)}, \text{ for } (i,j) \in \zeta\}$  with  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$ . The process results in the fully-populated, enlarged color image  $\mathbf{x}'$ .

To the date, the known color image spatial interpolation methods can be categorized into two classes [24]: i) component-wise solutions [Figure 16.11(a)], and ii) vector solutions [Figure 16.11(c)]. As shown in Figure 16.24, the same signal processing concept employed in the both component-wise and vector processing paradigms can lead to different visual quality of the enlarged images, especially in terms of color artifacts and zipper effects.

The component-wise zooming solutions, such as those presented in [65], [66], [67], process each color channel of the enlarged image  $\mathbf{x}'$  separately as follows :

$$x'_{(r,s)k} = f \left( \sum_{(i,j) \in \zeta} \{w'_{(i,j)} x'_{(i,j)k}\} \right) \quad (16.15)$$

where  $x'_{(\cdot,\cdot)k}$  denotes the  $k$ -th components (i.e.  $k = 1$  for R,  $k = 2$  for G, and  $k = 3$  for B) of the color vector  $\mathbf{x}'_{(\cdot,\cdot)}$  and  $f(\cdot)$  is a nonlinear function that operates over the weighted average of the input components. The weights  $w'_{(i,j)}$  calculated in (16.2) can be obtained using the approaches described in [68], [69], [70], [71]. Due to the component-wise nature of a zooming solution, the enlarged images obtained using (16.15) often suffer from color shifts and artifacts [64]. Moreover, the lack of edge-sensing combined with the omission of the spectral information during the processing often results in aliasing, edge blurring, jagged lines or blockiness [63].

Vector techniques process the available color pixels  $\mathbf{x}'_{(i,j)}$  as set of vectors :

$$\mathbf{x}'_{(r,s)} = f \left( \sum_{(i,j) \in \zeta} \{w'_{(i,j)} \mathbf{x}'_{(i,j)}\} \right) \quad (16.16)$$

preserving thus the spectral correlation of the enlarged image's color channels, reducing the presence of most color artifacts, and eliminating shifted color edges [64]. The most popular vector approaches are based on theory of robust order-statistics [8], [63], [64], and data-adaptive concepts [8], [64].

### 16.8.2 Spatial Interpolation of CFA Images

CFA image zooming solutions [24], [62], [72] constitute a novel way of spatial interpolation in single-sensor cameras. By using the zooming solution prior to the demosaicking solution (see Figure 16.7), the zooming approach prevents from amplifying the imperfections and visual impairments introduced in the demosaicking step [24]. Moreover, since the zooming operations are performed on the gray-scale CFA image instead of the demosaicked full-color image, the CFA zooming approach is obviously computationally simple and attractive to be embedded in cost-effective devices, such as imaging-enabled mobile phones and pocket-size digital cameras.

The CFA zooming procedure first maps the CFA pixels  $z_{(\cdot,\cdot)}$  into the enlarged gray-scale image  $z'$  with the pixels  $z'_{(\cdot,\cdot)}$  and then interpolates the empty positions using the available values. From the first view the procedure is quite similar to the one used in demosaicked image zooming. However, it is shown in [24], [62] that the common mapping step  $z'_{(2r-1,2s-1)} = z_{(r,s)}$  used in zooming the demosaicked image with a factor  $\tau = 2$  discards the essential mosaic-like structure of the enlarged Bayer CFA image and prohibits the correct use of the demosaicking step. Therefore, zooming the CFA image requires the special, CFA-based approach which maps the CFA values  $z_{(r,s)}$  with  $r$  and  $s$  denoting the spatial coordinates in the CFA image  $z$  to  $z'_{(2r-1,2s)} = z_{(r,s)}$  for  $r$  odd and  $s$  even, to  $z'_{(2r,2s-1)} = z_{(r,s)}$  for  $r$  even and  $s$  odd, and  $z'_{(2p-1,2q-1)} = z_{(r,s)}$  otherwise [62]. The remaining positions in the enlarged image  $z'$  are filled-in by zeros.

After the mapping step is completed [Figure 16.25(a)], the procedure continues by interpolating the pixels corresponding to G CFA locations using  $z'_{(r,s)} = \sum_{(i,j) \in \zeta} \{w'_{(i,j)} z'_{(i,j)}\}$  in the two steps due to the lack of spatial information. Thus,  $z'_{(r,s)}$  is the center of the four neighboring values  $z'_{(i,j)}$  described first by  $\zeta = \{(r-2, s), (r, s-2), (r, s+2), (r+2, s)\}$  depicted in Figure 16.25(a), and then by  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  in the updated image [Figure 16.25(b)].

By populating all G locations in the enlarged image  $z'$ , the obtained information along with the original CFA values can be used as the basis for the spectral-model based interpolation operations used to populate R and B locations in the image  $z'$ . If the color-difference model in (16.11) is employed, then the values

corresponding to R locations in the image  $z'$  can be obtained as follows:

$$z_{(r,s)} = z_{(r,s-1)} + \sum_{(i,j) \in \zeta} \{w'_{(i,j)}(z'_{(i,j)} - z'_{(i,j-1)})\} \quad (16.17)$$

whereas the values corresponding to B locations are obtained using

$$z_{(r,s)} = z_{(r-1,s)} + \sum_{(i,j) \in \zeta} \{w'_{(i,j)}(z'_{(i,j)} - z'_{(i-1,j)})\} \quad (16.18)$$

In both above equations, the values  $z'_{(i,j-1)}$  and  $z'_{(i-1,j)}$  represent the spatially shifted neighboring locations used to overcome the lack of spectral information in the locations  $(i, j)$ . Additional information about these spatial interpolation steps can be found in [24], [62]. The weights  $w'_{(i,j)}$  in (16.2) can be calculated using the concepts listed in Section 16.4.1. An overview of the other ESM solutions suitable for CFA zooming can be found in [24].

By populating R and B CFA locations using (16.17) and (16.18), respectively, the procedure interpolates the center of the four  $z'_{(i,j)}$  values described by  $\zeta = \{(r-2, s-2), (r-2, s+2), (r+2, s-2), (r+2, s+2)\}$  shown in Figure 16.25(c) and then by  $\zeta = \{(r-2, s), (r, s-2), (r, s+2), (r+2, s)\}$  in the updated image shown in Figure 16.25(d). The process results in the enlarged, mosaic-like image  $z'$  [see Fig.16.25(e)], which is a gray-scale image similarly to the one shown in Figure 16.21(b).

## 16.9 Conclusion

The digital image processing operations which support digital cameras equipped with a color filter array placed on top of a single image sensor were the main focus of this chapter. These consumer-grade cameras capture the natural scene first by producing the gray-scale, mosaic-like image and then by using extensive calculations based on the concept of image interpolation to output the full-color, visually pleasing image. Taking into the consideration the way the structural content and the spectral characteristics of the captured image are treated during the processing, as well as the nature of the interpolation operations, the chapter provided a taxonomy of single-sensor camera image processing solutions.

Since the edges and fine details (i.e. structural content) are essential for image understanding and color (i.e. spectral information) plays a significant role in the perception of edges or boundaries between two surfaces, data-adaptive spectral model-based solutions were the main focus of the chapter. Building on the refined edge-sensing and spectral modelling principles, these solutions constitute a rich and expanding class of tools for demosaicking, demosaicked image postprocessing and camera image zooming. By utilizing both the structural and spectral characteristics of the samples within the localized image area in cascaded sophisticated processing steps, the presented framework can: i) overcome the spatial, structural and spectral constraints imposed on the solution, ii) produce sharply-looking, naturally colored camera images, and iii) mimic the human perception of the visual environment.

As shown in this work, image interpolation operators based on the edge-sensing and spectral modelling principles constitute a basis for demosaicking which is the mandatory processing step in single-sensor imaging to obtain the full-color image. The same processing tools can be used to enhance the visual quality of the demosaicked image and to increase the spatial resolution of the captured images. Therefore, it is not difficult to see that single-sensor camera image processing techniques have an extremely valuable position in digital color imaging and will be also essential for new-generation imaging-enabled consumer electronics as well as novel multimedia applications.

## Bibliography

- [1] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Transactions on Image Processing*, vol. 6, pp. 901–932, July 1997.
- [2] R. Lukac and K. N. Plataniotis, "Data-adaptive filters for demosaicking: A framework," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 560–570, May 2005.
- [3] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, pp. 20–30, November 1998.
- [4] P. L. P. Dillon, D. M. Lewis, and F. G. Kaspar, "Color imaging system using a single CCD area array," *IEEE Journal of Solid-State Circuits*, vol. 13, pp. 28–33, February 1978.
- [5] B. T. Turko and G. J. Yates, "Low smear CCD camera for high frame rates," *IEEE Transactions on Nuclear Science*, vol. 36, pp. 165–169, February 1989.
- [6] T. Lule, S. Benthien, H. Keller, F. Mutze, P. Rieve, K. Seibel, M. Sommer, and M. Bohm, "Sensitivity of CMOS based imagers and scaling perspectives," *IEEE Transactions on Electron Devices*, vol. 47, pp. 2110–2122, November 2000.
- [7] A. J. Blanksby and M. J. Loinaz, "Performance analysis of a color CMOS photogate image sensor," *IEEE Transactions on Electron Devices*, vol. 47, pp. 55–64, January 2000.
- [8] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Processing Magazine; Special Issue on Color Image Processing*, vol. 22, pp. 74–86, January 2005.
- [9] R. F. Lyon and P. M. Hubel, "Eying the camera: Into the next century," in *Proc. IS&TSID Tenth Color Imaging Conference*, (Scottsdale, USA), pp. 349–355, November 2002.
- [10] R. J. Guttosch, "Investigation of color aliasing of high spatial frequencies and edges for Bayer-pattern sensors and Foveon X3 direct image sensor," tech. rep., Foveon, 2002.
- [11] P. M. Hubel, J. Liu, and R. J. Guttosch, "Spatial frequency response of color image sensors: Bayer color filters and Foveon X3," Tech. Rep. ID 6502, Foveon, San Antonio, USA, March 2002.

- [12] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schaffer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, pp. 44–54, January 2005.
- [13] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Transactions on Image Processing*, vol. 13, pp. 1263–1274, September 2004.
- [14] K. Parulski and K. Spaulding, *Digital Color Imaging Handbook*, ch. Color image processing for digital cameras, pp. 728–757. Boca Raton, FL., 2002.
- [15] J. Holm, I. Tastl, L. Hanlon, and P. Hubel, *Colour Engineering: Achieving Device Independent Colour*, ch. Color processing for digital photography, pp. 179–220. New York: Wiley, 2002.
- [16] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine; Special Issue on Color Image Processing*, vol. 22, pp. 34–43, January 2005.
- [17] R. Lukac and K. N. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 1260–1267, November 2005.
- [18] FillFactory, "Technology - image sensor: the color filter array faq," tech. rep., Available Online: <http://www.fillfactory.com/htm/technology/htm/rgbfaq.htm>.
- [19] B. E. Bayer, "Color imaging array." U.S. Patent 3 971 065, July 1976.
- [20] S. Yamanaka, "Solid state camera." U.S. Patent 4 054 906, November 1977.
- [21] M. Parmar and S. J. Reeves, "A perceptually based design methodology for color filter arrays," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, vol. 3, (Montreal, Canada), pp. 473–476, May 2004.
- [22] R. Lukac and K. N. Plataniotis, "Universal demosaicking for imaging pipelines with an RGB color filter array," *Pattern Recognition*, vol. 38, pp. 2208–2212, November 2005.
- [23] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, pp. 997–1013, September 2002.
- [24] R. Lukac, K. N. Plataniotis, and D. Hatzinakos, "Color image zooming on the Bayer pattern," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 15, pp. 1475–1492, November 2005.
- [25] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet - sRGB," tech. rep., Our University, [www.w3.org/Graphics/Color/sRGB.html](http://www.w3.org/Graphics/Color/sRGB.html), 1996.
- [26] T. S. C. on AV & IT Storage Systems and Equipment, "Exchangeable image file format for digital still cameras: Exif Version 2.2," Tech. Rep. JEITA CP-3451, Japan Electronics and Information Technology Industries Association, April 2002.
- [27] J. Tesic, "Metadata practices for consumer photo," *IEEE MultiMedia*, vol. 12, pp. 86–92, July-September 2005.
- [28] Y. T. Tsai, "Color image compression for single-chip cameras," *IEEE Transactions on Electron Devices*, vol. 38, pp. 1226–1232, May 1991.

- [29] T. Toi and M. Ohita, "A subband coding technique for image compression in single CCD cameras with Bayer color filter arrays," *IEEE Transactions on Consumer Electronics*, vol. 45, pp. 176–180, February 1999.
- [30] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 1448–1456, November 2003.
- [31] S. Battiato, A. R. Bruna, A. Buemi, and A. Castorina, "Analysis and characterization of JPEG 2000 standard for imaging devices," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 773–779, November 2003.
- [32] X. Xie, G. L. Li, X. W. Li, D. M. L. Z. H. Wang, C. Zhang, and L. Zhang, "A new approach for near-lossless and lossless image compression with Bayer color filter arrays," in *Proceedings of the Third International Conference on Image and Graphics (ICIG'04)*, (Hong Kong, China), pp. 357–360, December 2004.
- [33] A. Bazhyna, A. Gotchev, and K. Egiazarian, "Near-lossless compression algorithm for Bayer pattern color filter arrays," in *Proc. SPIE-IS&T Electronic Imaging*, vol. SPIE 5678, pp. 198–209, 2005.
- [34] R. Lukac and K. N. Plataniotis, "Fast video demosaicking solution for mobile phone imaging applications," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 675–681, May 2005.
- [35] L. Zhang, X. Wu, and P. Bao, "Real-time lossless compression of mosaic video sequences," *Real-Time Imaging; Special Issue on Multi-Dimensional Image Processing*, vol. 11, pp. 370–377, October-December 2005.
- [36] R. Lukac and K. N. Plataniotis, "Digital image indexing using secret sharing schemes: A unified framework for single-sensor consumer electronics," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 908–916, August 2005.
- [37] R. Lukac and K. N. Plataniotis, "On a generalized demosaicking procedure: A taxonomy of single-sensor imaging solutions," *Lecture Notes in Computer Science*, vol. 3514, pp. 687–694, May 2005.
- [38] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU," *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 1342–1352, November 1998.
- [39] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proceedings of the IEEE*, vol. 90, pp. 123–132, January 2002.
- [40] D. Alleyson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system," *IEEE Transactions on Image Processing*, vol. 14, pp. 439–449, April 2005.
- [41] W. T. Freeman, "Median filter for reconstructing missing color samples." U.S. Patent 4 724 395, 1988.
- [42] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 503–513, June 2003.
- [43] K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Springer Verlag, 2000.
- [44] O. Faugeras, "Digital color image processing within the framework of a human visual model," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 380–393, August 1979.
- [45] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Transactions on Image Processing*, vol. 8, pp. 1221–1228, September 1999.

- [46] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicing approach," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 256–261, February 2004.
- [47] W. Lu and Y. P. Tang, "Color filter array demosaicking: New method and performance measures," *IEEE Transactions on Image Processing*, vol. 12, pp. 1194–1210, October 2003.
- [48] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. S. III, "Demosaicking methods for Bayer color arrays," *Journal of Electronic Imaging*, vol. 11, pp. 306–315, July 2002.
- [49] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal." U.S. Patent 4 642 678, February 1987.
- [50] J. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. SPIE*, vol. 3028, pp. 117–125, February 1997.
- [51] R. Lukac and K. N. Plataniotis, "Normalized color-ratio modeling for CFA interpolation," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 737–745, May 2004.
- [52] R. Lukac and K. N. Plataniotis, "A vector spectral model for digital camera image processing," *IEEE Transactions on Circuit and Systems for Video Technology*, submitted.
- [53] R. Lukac and K. N. Plataniotis, "A new approach to CFA image indexing," *Lecture Notes in Computer Science*, vol. 3691, pp. 137–144, September 2005.
- [54] R. Lukac, K. Martin, and K. N. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 15–24, February 2004.
- [55] K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Adaptive fuzzy systems for multichannel signal processing," *Proceedings of the IEEE*, vol. 87, pp. 1601–1622, September 1999.
- [56] N. Kehtarnavaz, H. J. Oh, and Y. Yoo, "Color filter array interpolation using color correlations and directional derivatives," *Journal of Electronic Imaging*, vol. 12, pp. 621–632, October 2003.
- [57] L. Chang and Y. P. Tang, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 355–365, May 2004.
- [58] B. Tang, G. Sapiro, and V. Caselles, "Color image enhancement via chromaticity diffusion," *IEEE Transactions on Image Processing*, vol. 10, pp. 701–707, May 2001.
- [59] P. E. Trahanias, D. Karakos, and A. N. Venetsanopoulos, "Directional processing of color images: Theory and experimental results," *IEEE Transactions on Image Processing*, vol. 5, pp. 868–881, June 1996.
- [60] R. Lukac and K. N. Plataniotis, "A robust, cost-effective postprocessor for enhancing demosaicked camera images," *Real-Time Imaging, Special Issue on Spectral Imaging II*, vol. 11, pp. 139–150, April 2005.
- [61] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image postprocessing using local color ratios," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 14, pp. 914–920, June 2004.
- [62] R. Lukac and K. N. Plataniotis, "Digital zooming for color filter array based image sensors," *Real-Time Imaging, Special Issue on Spectral Imaging II*, vol. 11, pp. 129–138, April 2005.

- [63] N. Herodotou and A. N. Venetsanopoulos, "Colour image interpolation for high resolution acquisition and display devices," *IEEE Transactions on Consumer Electronics*, vol. 41, pp. 1118–1126, November 1995.
- [64] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "Vector operators for color image zooming," in *Proc. IEEE International Symposium on Industrial Electronics (ISIE'05)*, vol. III, (Dubrovnik, Croatia), pp. 1273–1277, June 2005.
- [65] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, pp. 1153–1160, December 1981.
- [66] R. R. Schultz and R. L. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Transactions on Acoustics, Speech and Image Processing*, vol. 3, pp. 233–242, May 1994.
- [67] S. E. Reichenbach and F. Geng, "Two-dimensional cubic convolution," *IEEE Transactions on Image Processing*, vol. 12, pp. 857–865, August 2003.
- [68] K. Jensen and D. Anastasiou, "Subpixel edge localization and interpolation of still images," *IEEE Transactions on Image Processing*, vol. 4, pp. 285–295, March 1995.
- [69] S. Thurnhofer and S. K. Mitra, "Edge-enhanced image zooming," *Optical Engineering*, vol. 35, pp. 1862–1869, July 1996.
- [70] A. M. Darwish, M. S. Bedair, and S. I. Shaheen, "Adaptive resampling algorithm for image zooming," *IEE Proceedings - Vision, Image, Signal Processing*, vol. 144, pp. 207–212, August 1997.
- [71] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," *IEEE Signal Processing Letters*, vol. 11, pp. 359–362, March 2004.
- [72] G. G. S. Battiato and F. Stanco, "A locally adaptive zooming algorithm for digital images," *Image and Vision Computing*, vol. 20, pp. 805–812, September 2002.
- [73] R. Kakarala and Z. Baharav, "Adaptive demosaicing with the principal vector method," *IEEE Transactions on Consumer Electronics*, vol. 48, pp. 932–937, November 2002.
- [74] J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera." U.S. Patent 5 629 734, March 1997.
- [75] C. Cai, T. H. Yu, and S. K. Mitra, "Saturation-based adaptive inverse gradient interpolation for Bayer pattern images," *IEE Proceedings - Vision, Image, Signal Processing*, vol. 148, pp. 202–208, June 2001.

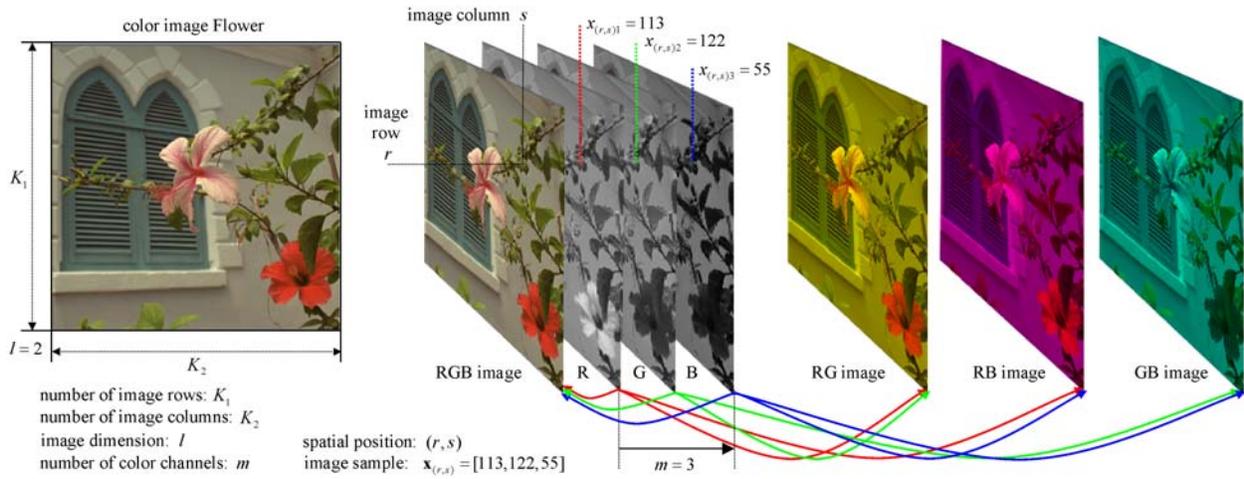


Figure 16.1: Color image representation in the RGB color domain.

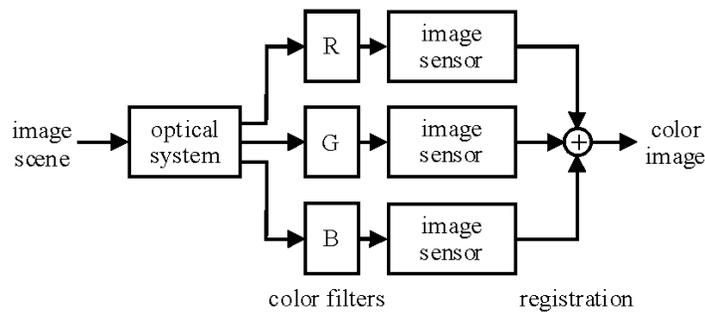


Figure 16.2: Three-sensor device.

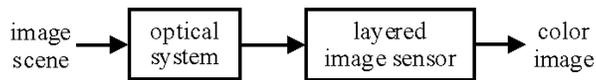


Figure 16.3: X3 technology based device.

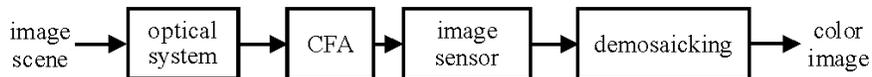


Figure 16.4: Single-sensor device.

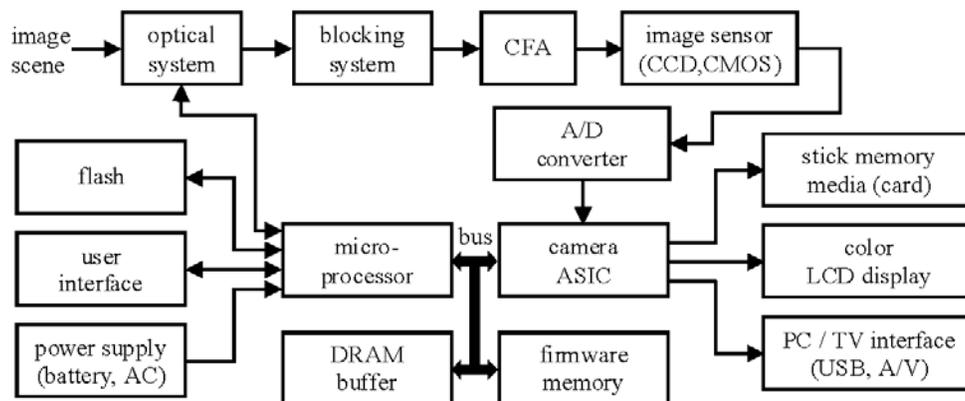


Figure 16.5: Consumer camera hardware architecture.

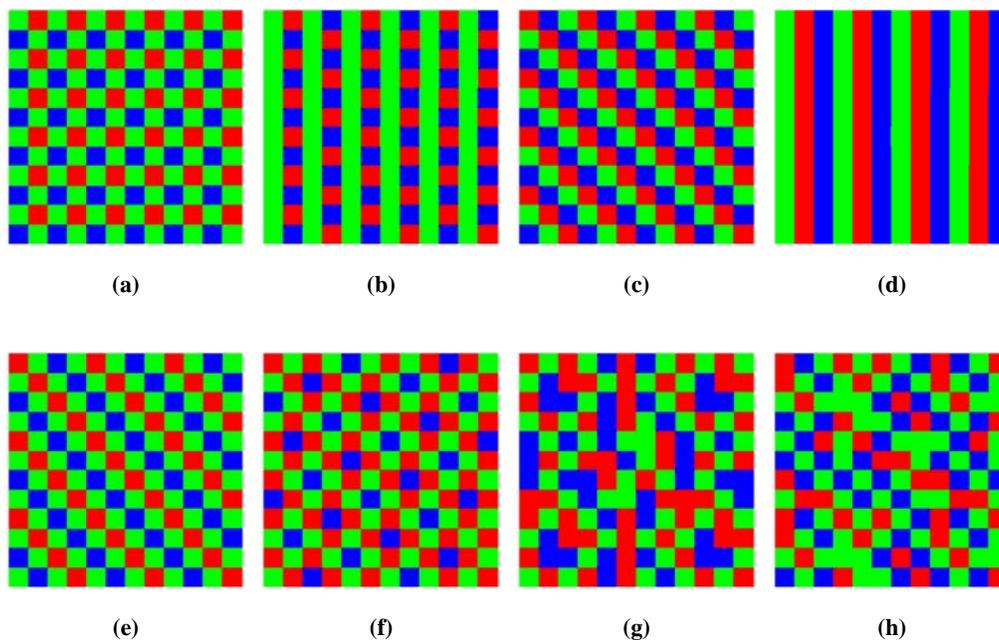
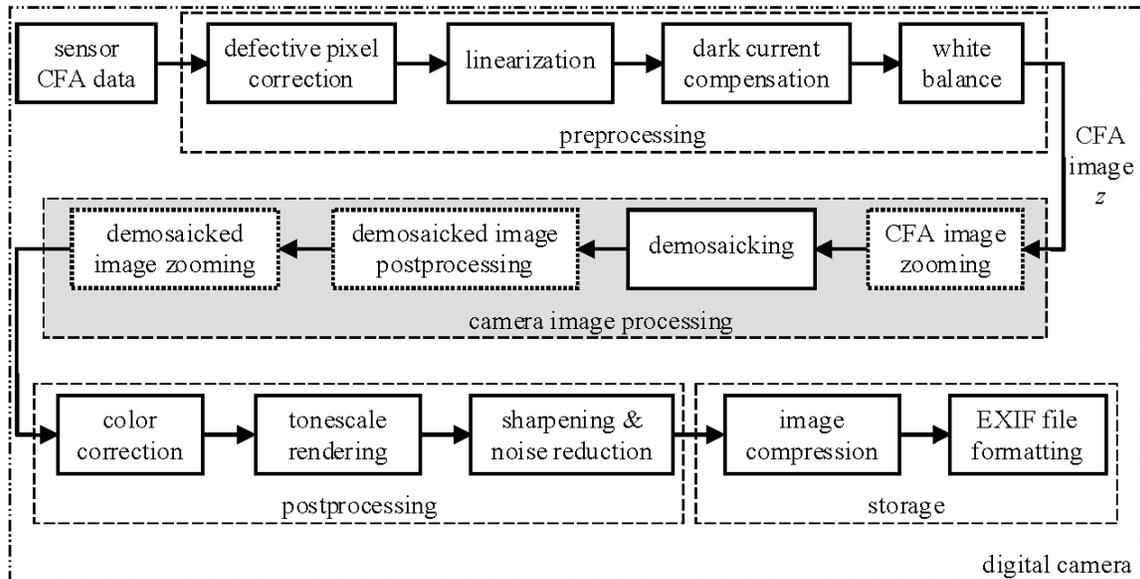
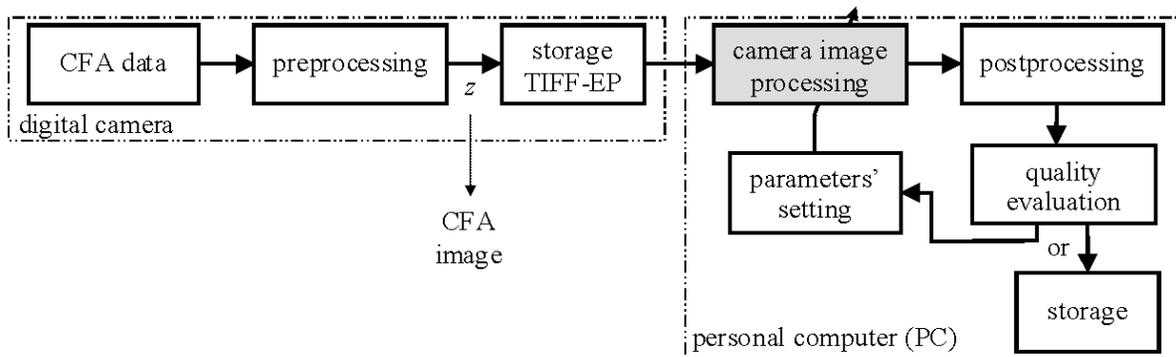


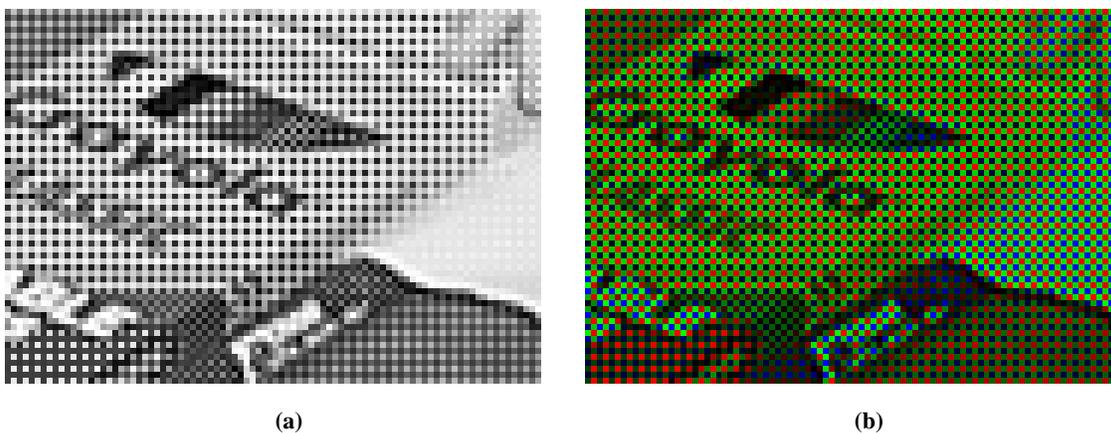
Figure 16.6: Examples of RGB CFAs: (a) Bayer pattern, (b) Yamanaka pattern, (c) diagonal stripe pattern, (d) vertical stripe pattern, (e) diagonal Bayer pattern, (f,g) pseudo-random patterns, (h) HVS-based pattern.



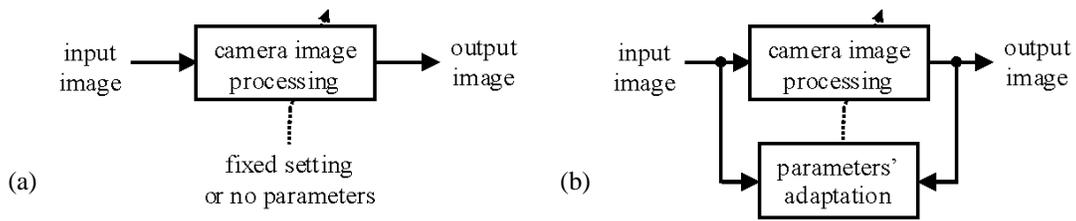
**Figure 16.7:** Example image processing flow in a consumer-grade camera.



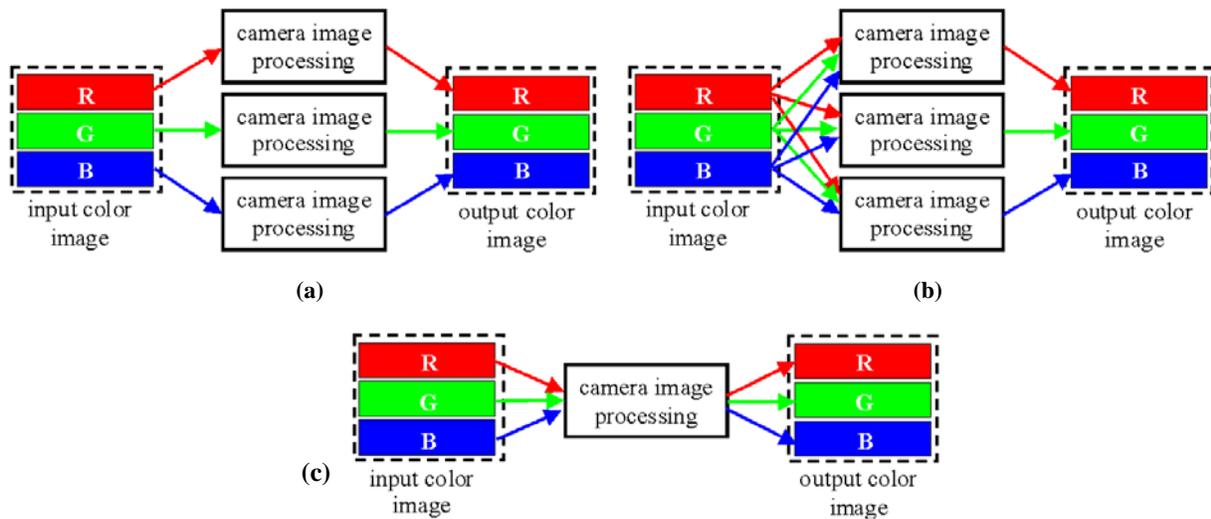
**Figure 16.8:** Example image processing flow in a camera interfacing with a companion personal computer.



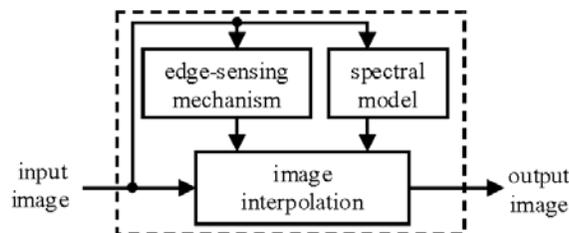
**Figure 16.9:** CFA image obtained using a well-known Bayer CFA with the GRGR phase in the first row: (a) acquired gray-scale CFA image, (b) CFA image re-arranged as a color image.



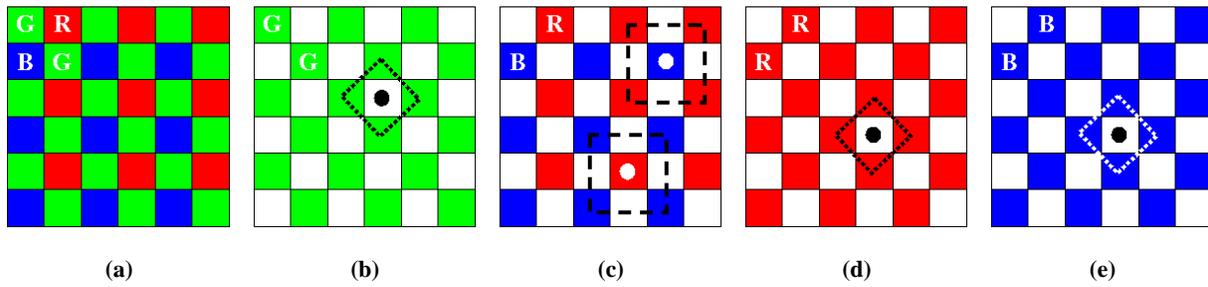
**Figure 16.10:** Image processing paradigms divided considering the structural characteristics: (a) non-adaptive processing, (b) data-adaptive processing.



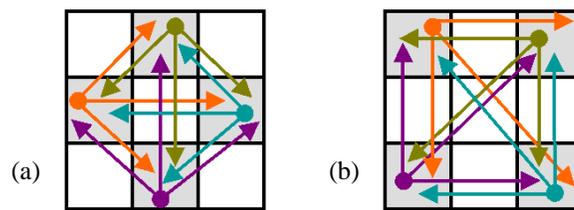
**Figure 16.11:** Image processing paradigms divided considering the spectral characteristics: (a) component-wise processing, (b) spectral model-based processing, (c) vector processing.



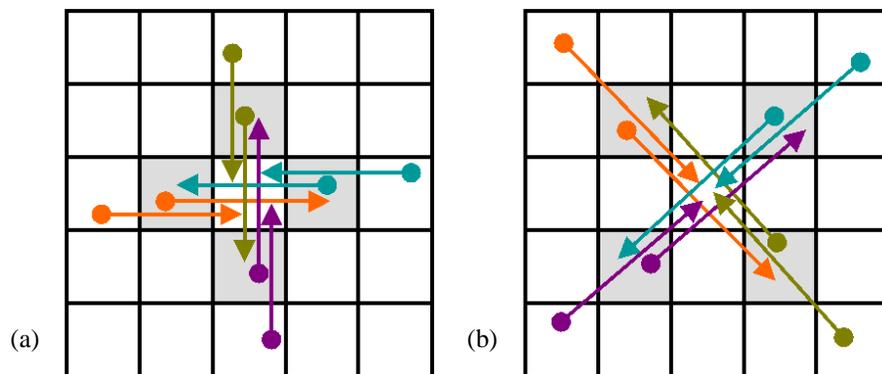
**Figure 16.12:** Generalized camera image processing solution suitable for demosaicking, demosaicked image postprocessing and camera image zooming.



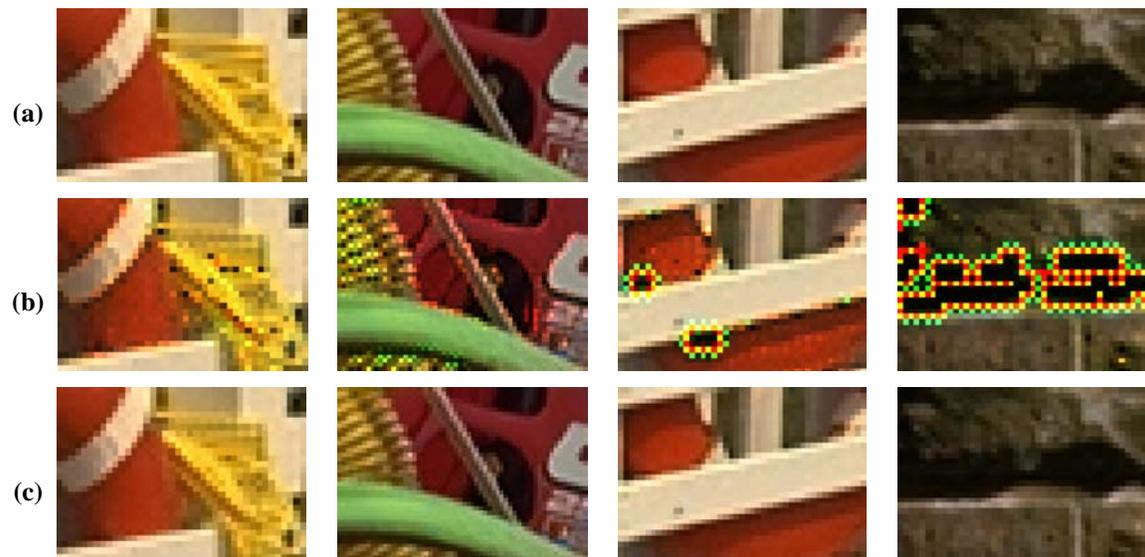
**Figure 16.13:** Shape-masks used in restoring the color information in the Bayer CFA image: (a) Bayer CFA with the GRGR phase in the first row, (b) G channel restoration, (c-e) R and B channel restoration.



**Figure 16.14:** Aggregation concept depicted for the shape masks: (a) the shape mask  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  [see Figures 16.13(b),(d),(e)], and (b)  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  [see Figure 16.13(c)].



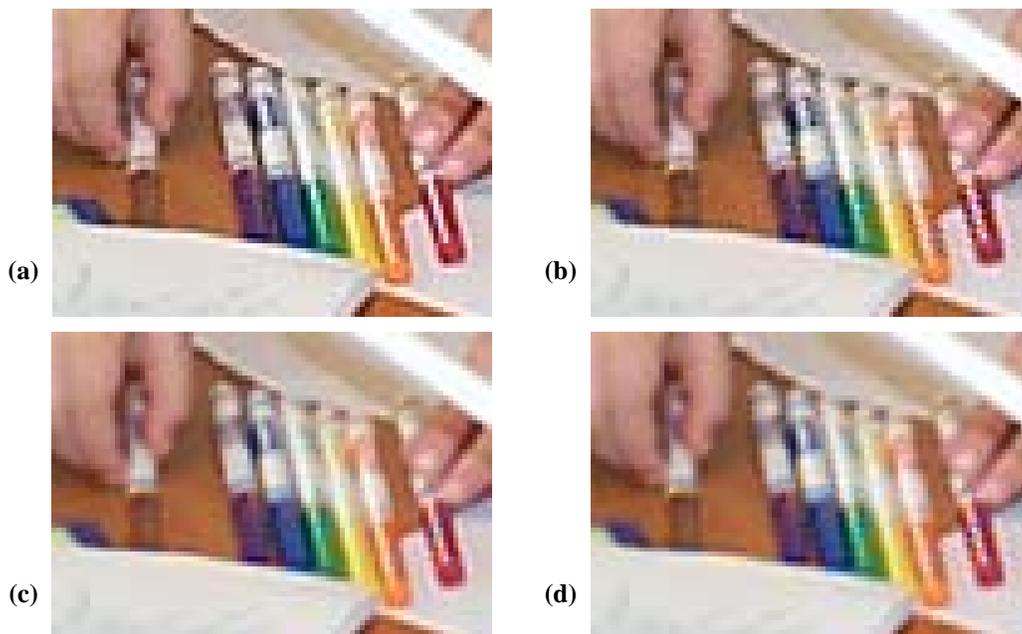
**Figure 16.15:** Neighborhood expansion concept depicted for the shape-masks: (a)  $\zeta = \{(r-1, s), (r, s-1), (r, s+1), (r+1, s)\}$  [see Figures 16.13(b),(d),(e)], and (b)  $\zeta = \{(r-1, s-1), (r-1, s+1), (r+1, s-1), (r+1, s+1)\}$  [see Figure 16.13(c)].



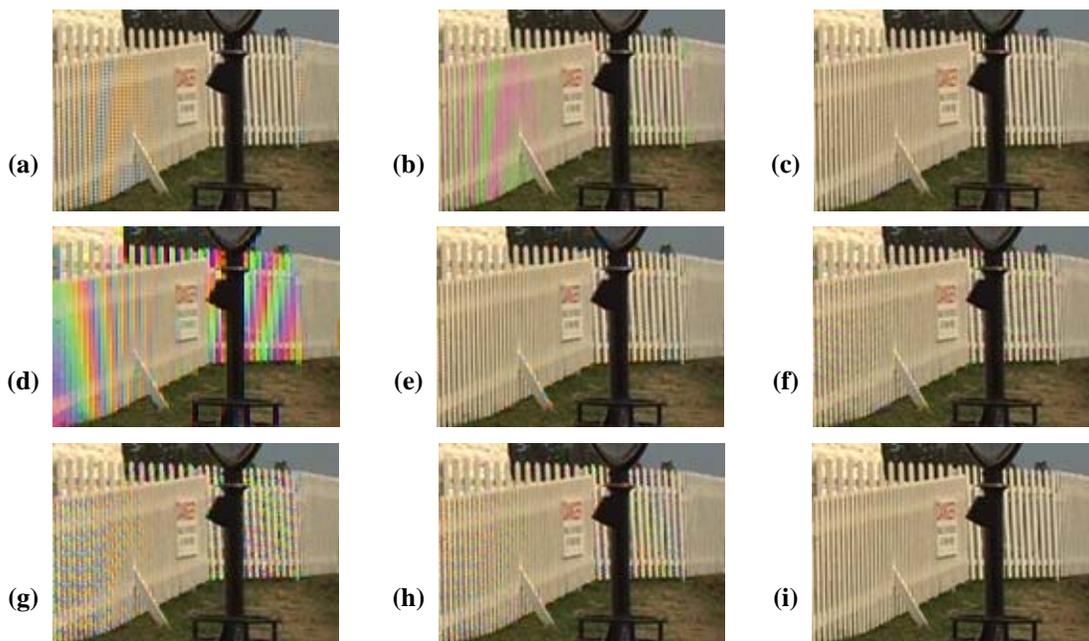
**Figure 16.16:** Performance improvements obtained by changing the SM: (a) original image, (b) Kimmel algorithm based on the color-ratio model [45], (c) Kimmel algorithm based on the normalized color-ratio model [51].



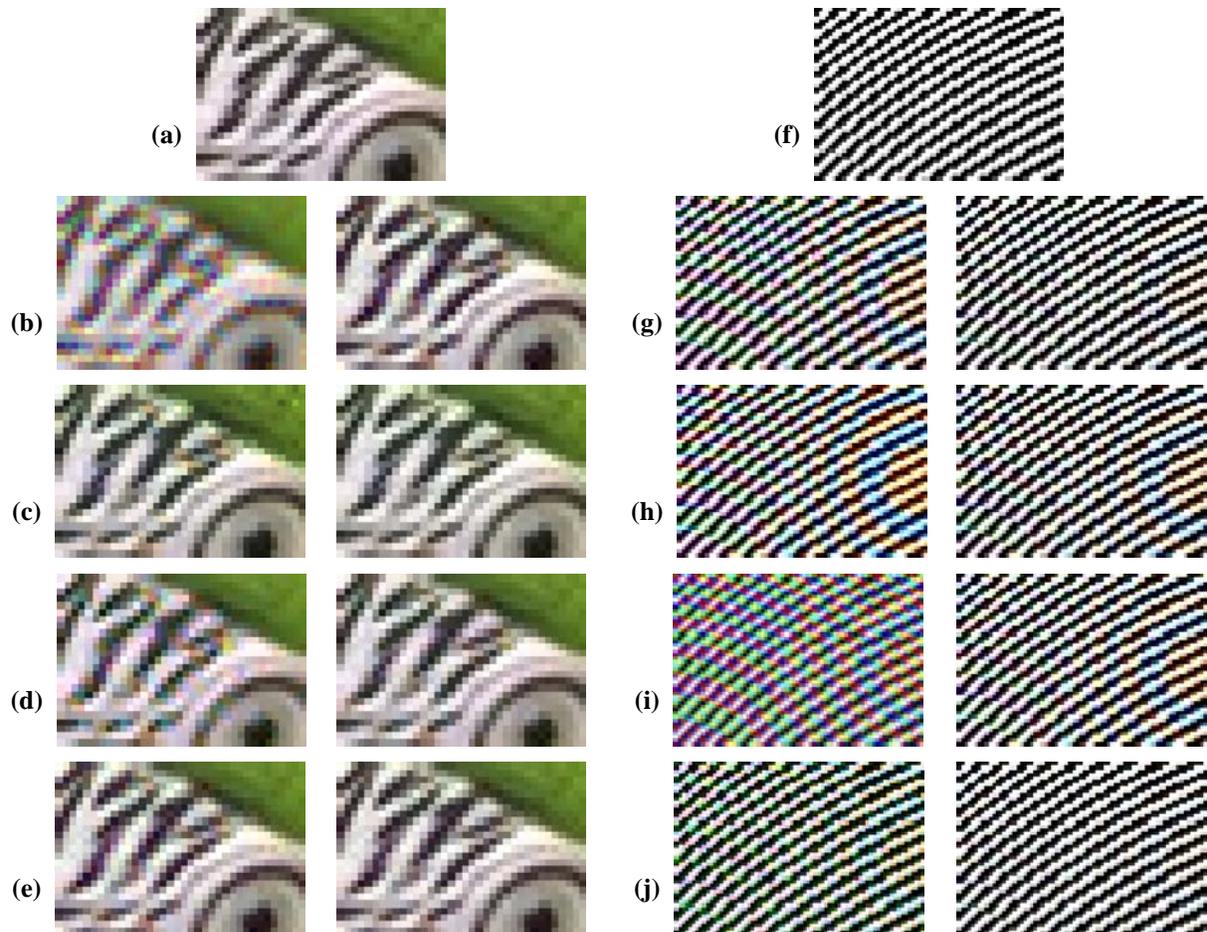
**Figure 16.17:** Demosaicking process: (a) restoration of the image shown in Figure 16.9(a) using the demosaicking solution, (b) image shown in Figure 16.17(a) enhanced using the demosaicked image postprocessor.



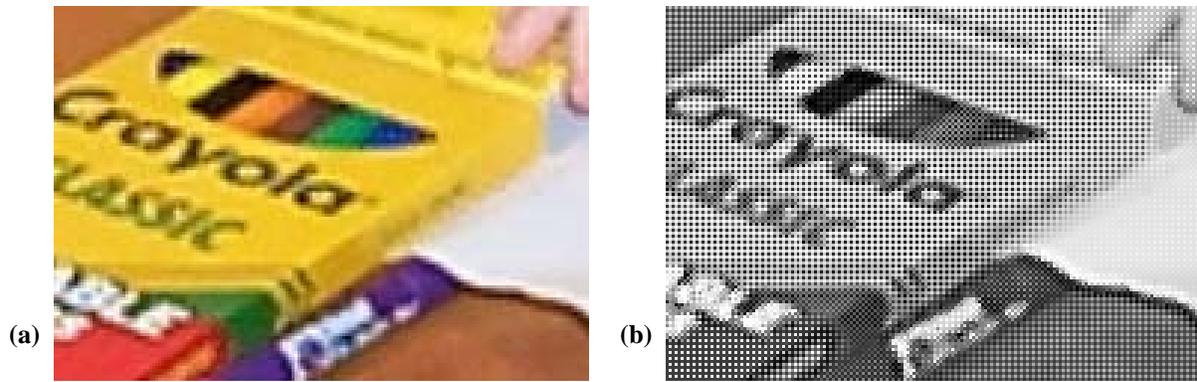
**Figure 16.18:** Influence of the ESM and SM on the quality of the demosaicked image: (a) used both ESM and SM, (b) omitted ESM, (c) omitted SM, (d) omitted both ESM and SM.



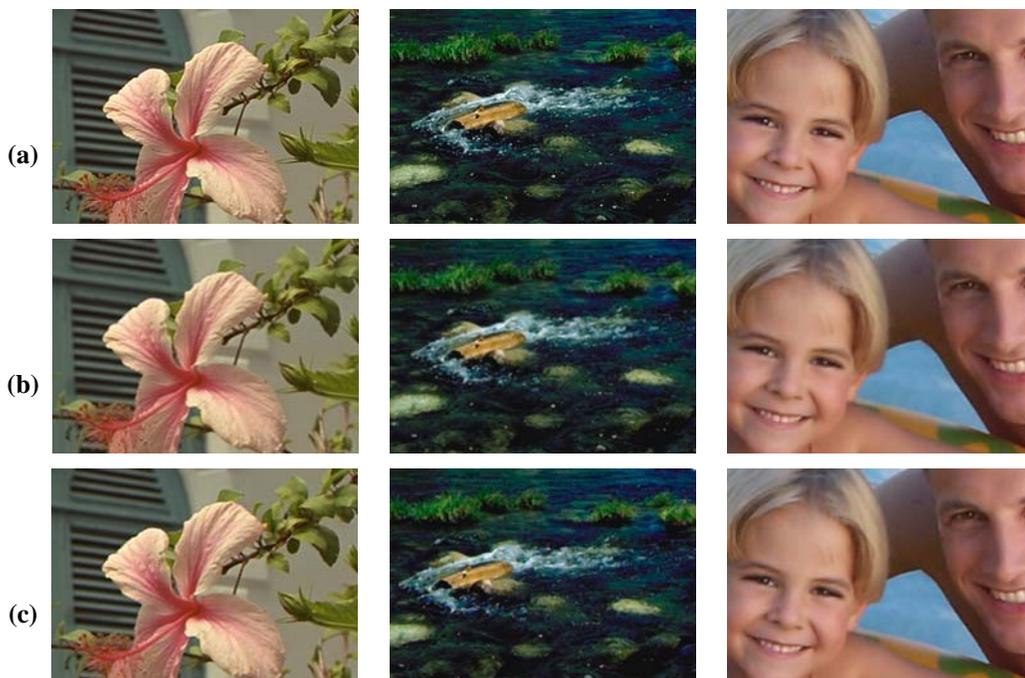
**Figure 16.19:** Influence of the CFA on the quality of the demosaicked image demonstrated using the same processing solution: (a-h) demosaicked image respectively corresponding to the RGB CFAs shown in Figures 16.6(a)-(h); (i) original image.



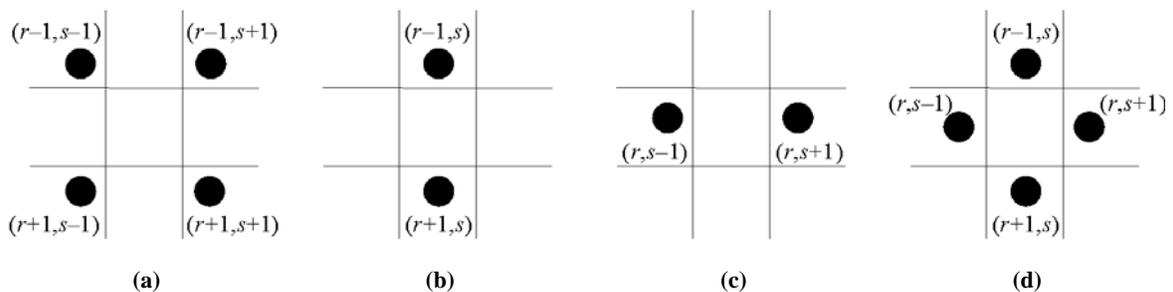
**Figure 16.20:** Demosaicked image postprocessing: (a,f) original images, (left) images demosaicked using the solutions presented in (b) [38], (c) [45], (d) [73], (e) [42], (g) [56], (h) [74], (i) [38], (j) [75]; (right) the corresponding enhanced images obtained by the postprocessing method presented in [60].



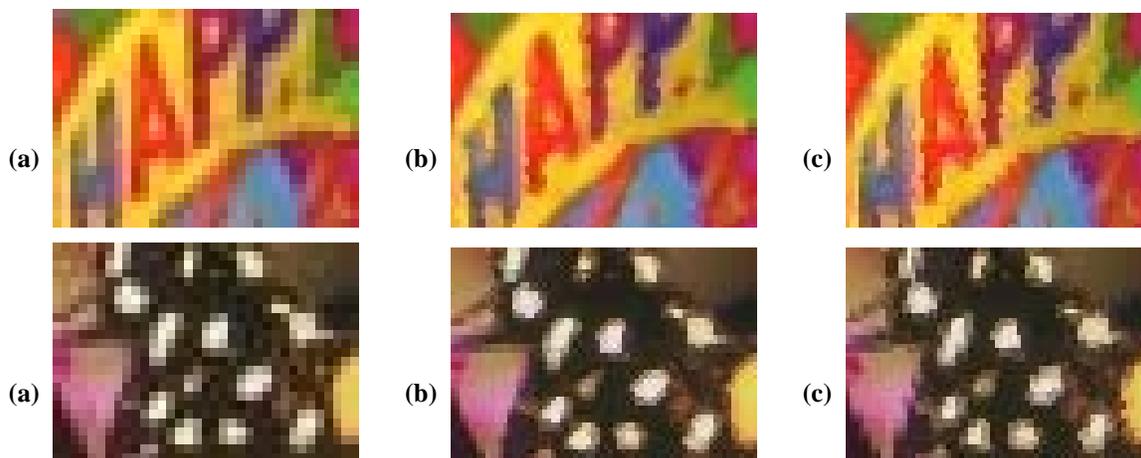
**Figure 16.21:** Spatial interpolation: (a) demosaicked image zooming using the image shown in Figure 16.17(b); (b) CFA image zooming using the image shown in Figure 16.9(a).



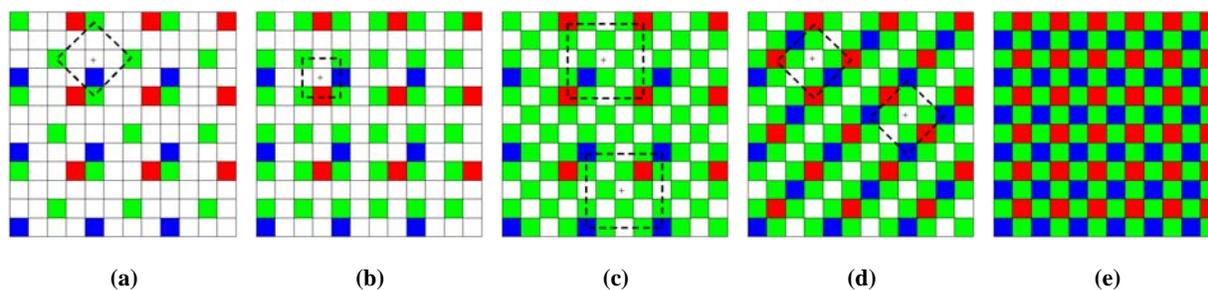
**Figure 16.22:** Camera image zooming: (a) original images, (b) demosaicking followed by demosaicked image zooming, (c) CFA image zooming followed by demosaicking.



**Figure 16.23:** Demosaicked image zooming with a zooming factor  $\tau = 2$ . Spatial arrangements obtained by: (a-c) mapping the demosaicked image  $x$  into the enlarged image  $x'$ , (d) completing the spatial interpolation of  $x'$  using the pattern shown in (a).



**Figure 16.24:** Median filtering based spatial interpolation [64]: (a) original (small) images, and (b,c) up-sampled images obtained using (b) component-wise processing, (c) vector processing.



**Figure 16.25:** CFA image zooming: (a) mapping step using the CFA image  $z$  shown in Figure 16.13(a); (b-d) interpolation steps; (e) enlarged CFA image  $z'$ .