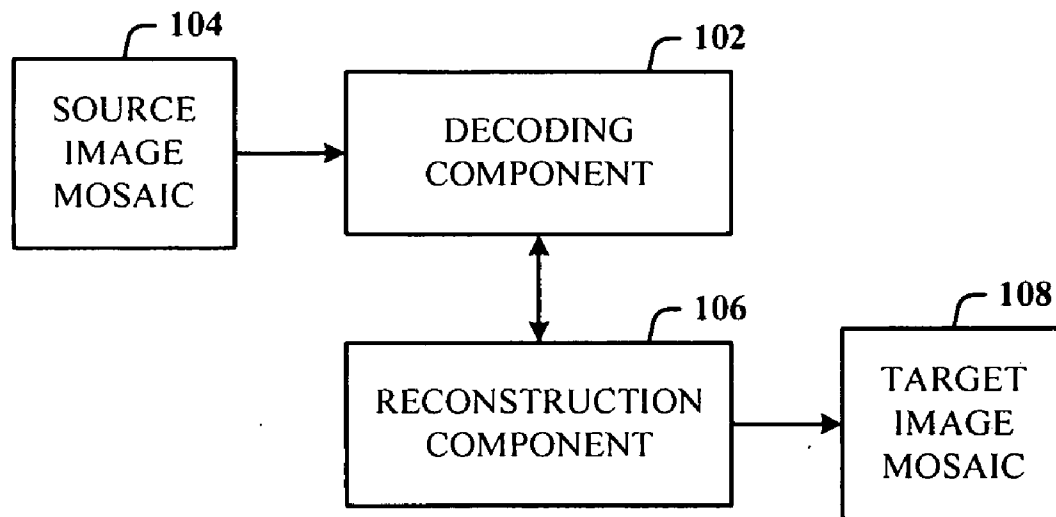


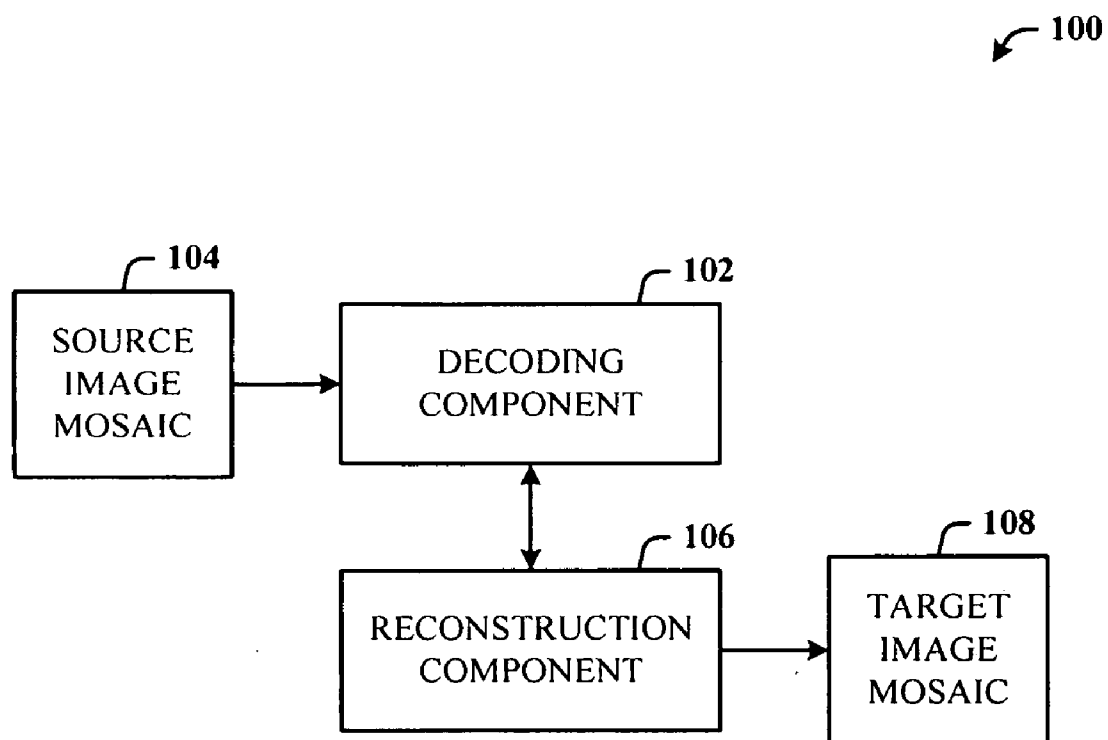


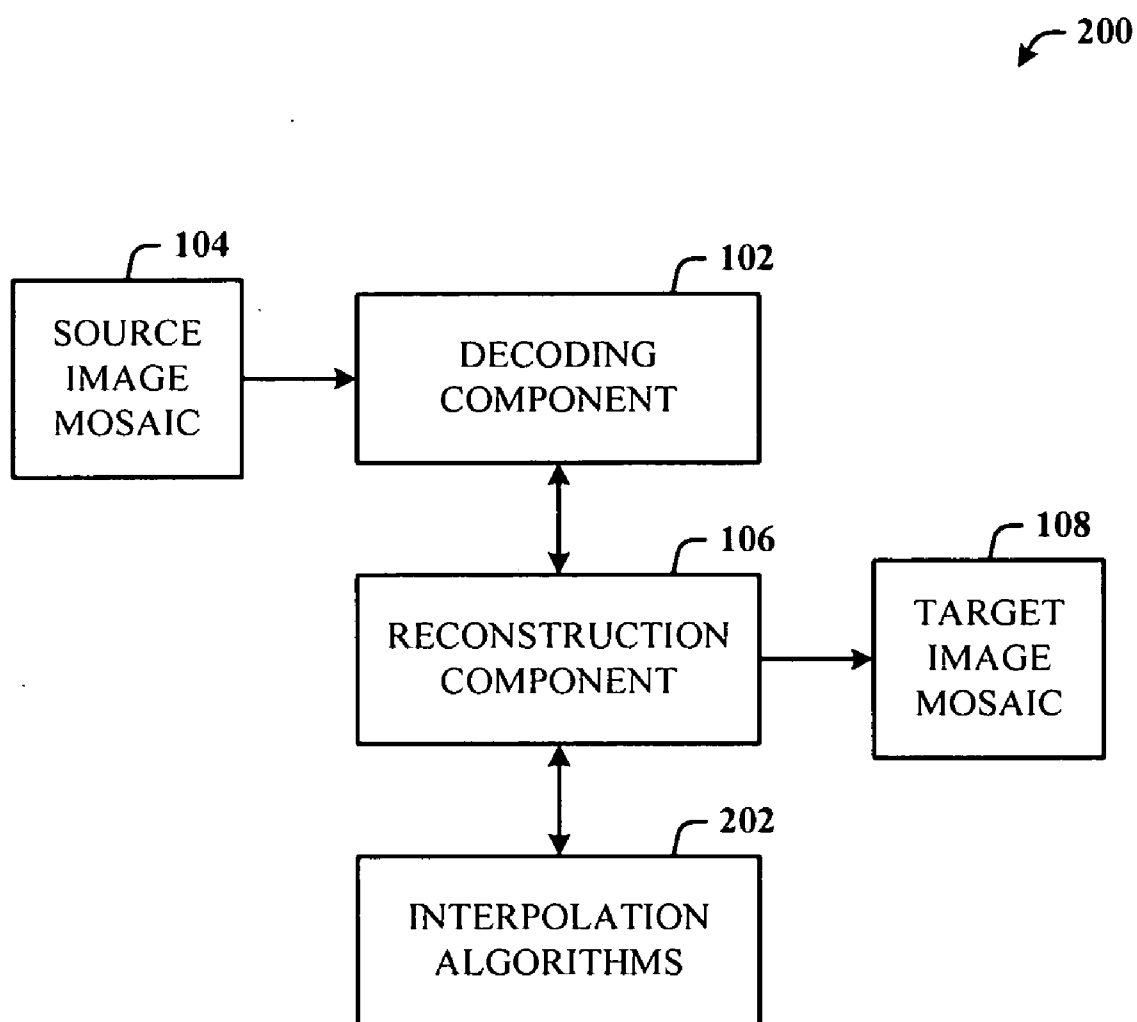
US 20100067789A1

(19) **United States**(12) **Patent Application Publication**
Cai et al.(10) **Pub. No.: US 2010/0067789 A1**(43) **Pub. Date: Mar. 18, 2010**(54) **RECONSTRUCTION OF IMAGE IN A BAYER PATTERN**(22) Filed: **Sep. 18, 2008**(75) Inventors: **Xiongfei Cai**, Beijing (CN);
Feng-Hsiung Hsu, Cupertino, CA (US)**Publication Classification**(51) **Int. Cl.**
G06K 9/36 (2006.01)(52) **U.S. Cl.** **382/166**Correspondence Address:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052 (US)(57) **ABSTRACT**

Architecture for decoding (demosaicing) a source image and performing reconstruction directly from the Bayer pattern to reduce memory size and improve communication bandwidth. The architecture can be easily implemented in hardware such as in field programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs).

(73) Assignee: **MICROSOFT CORPORATION**,
Redmond, WA (US)(21) Appl. No.: **12/212,792**↖ **100**

**FIG. 1**

**FIG. 2**

300

302

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \frac{\Delta x}{2} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} R(X, Y) \\ R(X+2, Y) \\ R(X, Y+2) \\ R(X+2, Y+2) \end{bmatrix}$$

304

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{2} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} + \frac{\Delta x}{2} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} R(X-1, Y) \\ R(X+1, Y) \\ R(X-1, Y+2) \\ R(X+1, Y+2) \end{bmatrix}$$

306

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{4} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} - \frac{\Delta x}{4} + \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} R(X, Y-1) \\ R(X+2, Y-1) \\ R(X, Y+1) \\ R(X+2, Y+1) \end{bmatrix}$$

308

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} \frac{1}{4} - \frac{\Delta x}{4} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} - \frac{\Delta x}{4} + \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} + \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} R(X-1, Y-1) \\ R(X+1, Y-1) \\ R(X-1, Y+1) \\ R(X+1, Y+1) \end{bmatrix}$$

FIG. 3

400

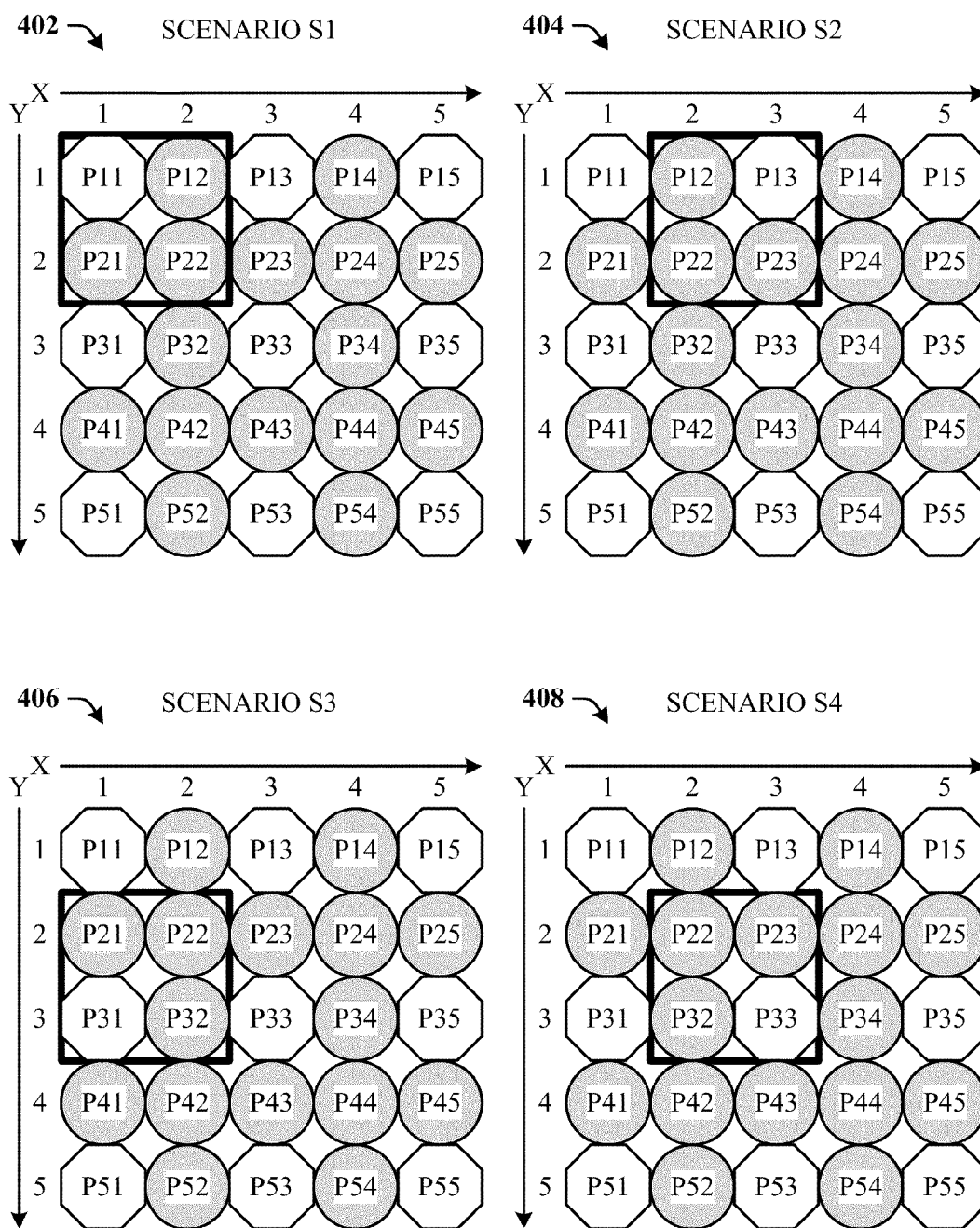


FIG. 4

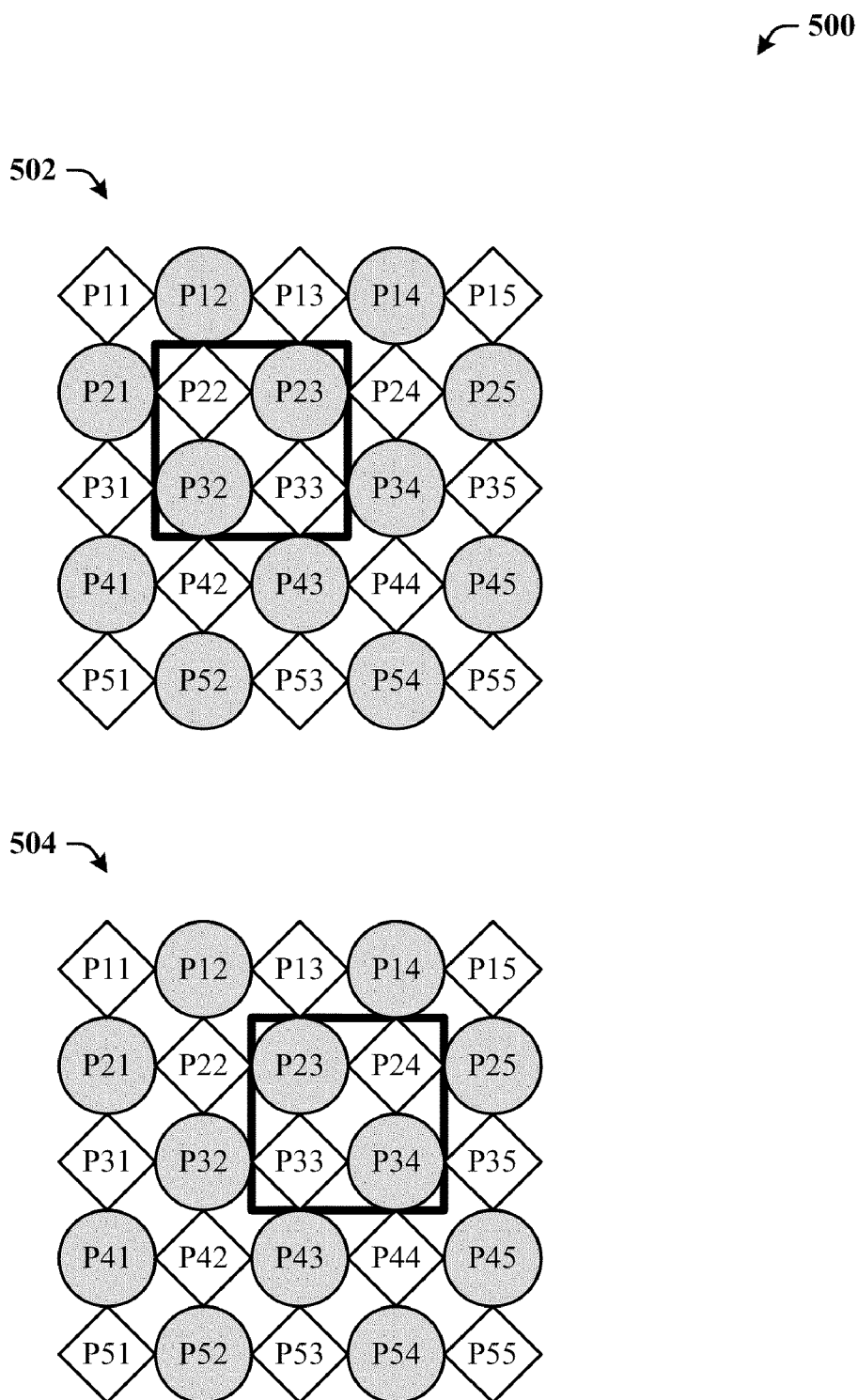
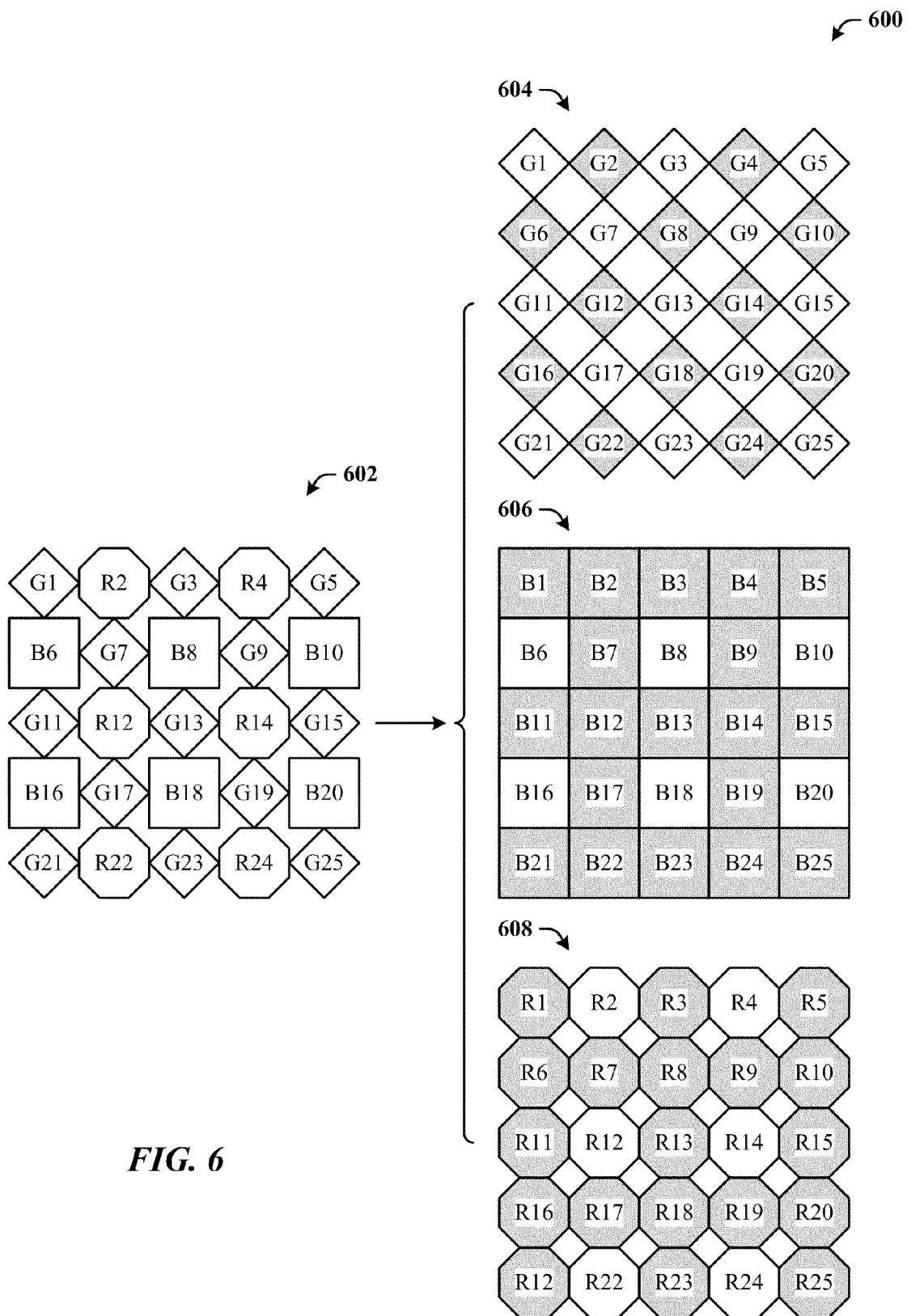
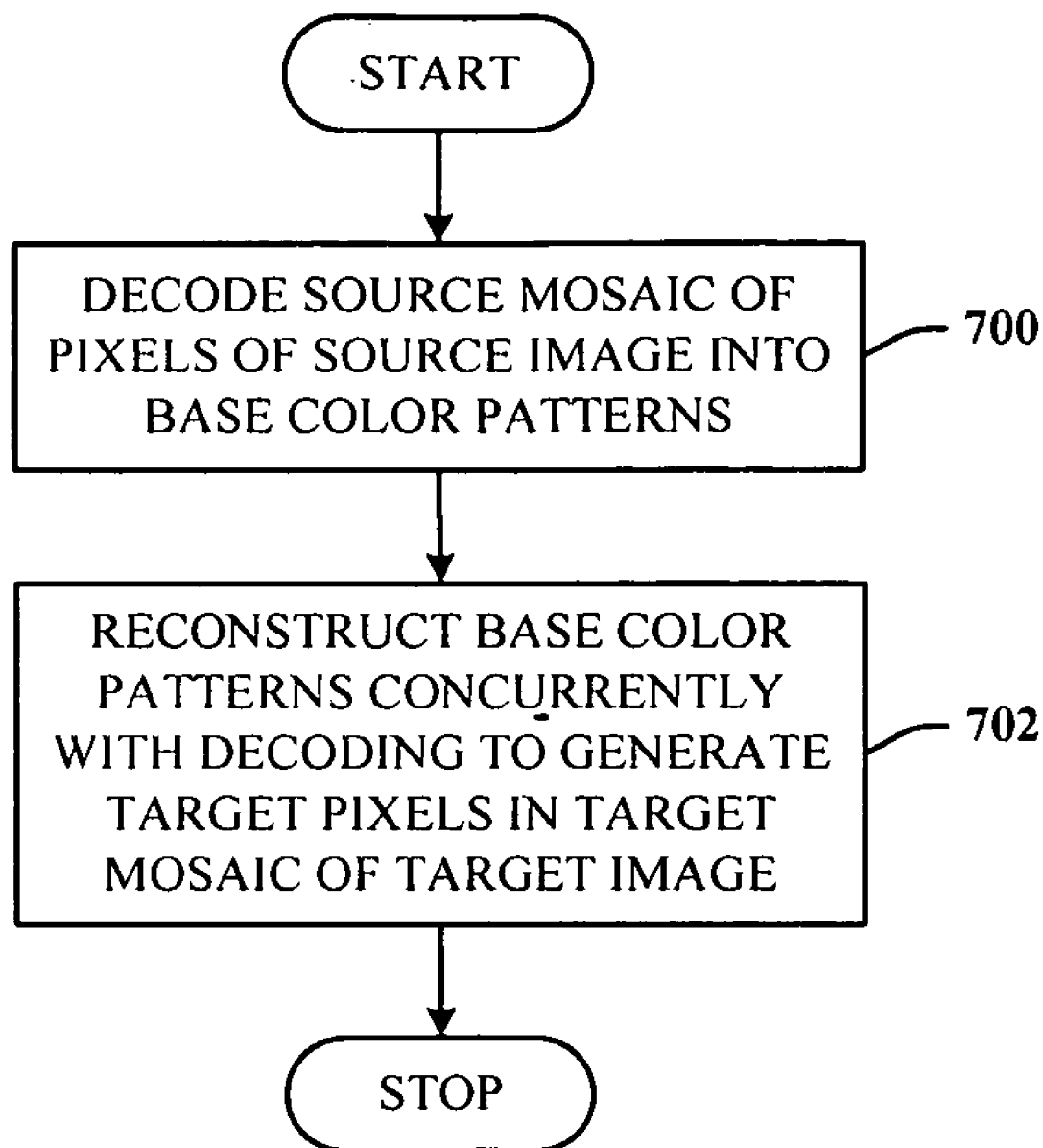
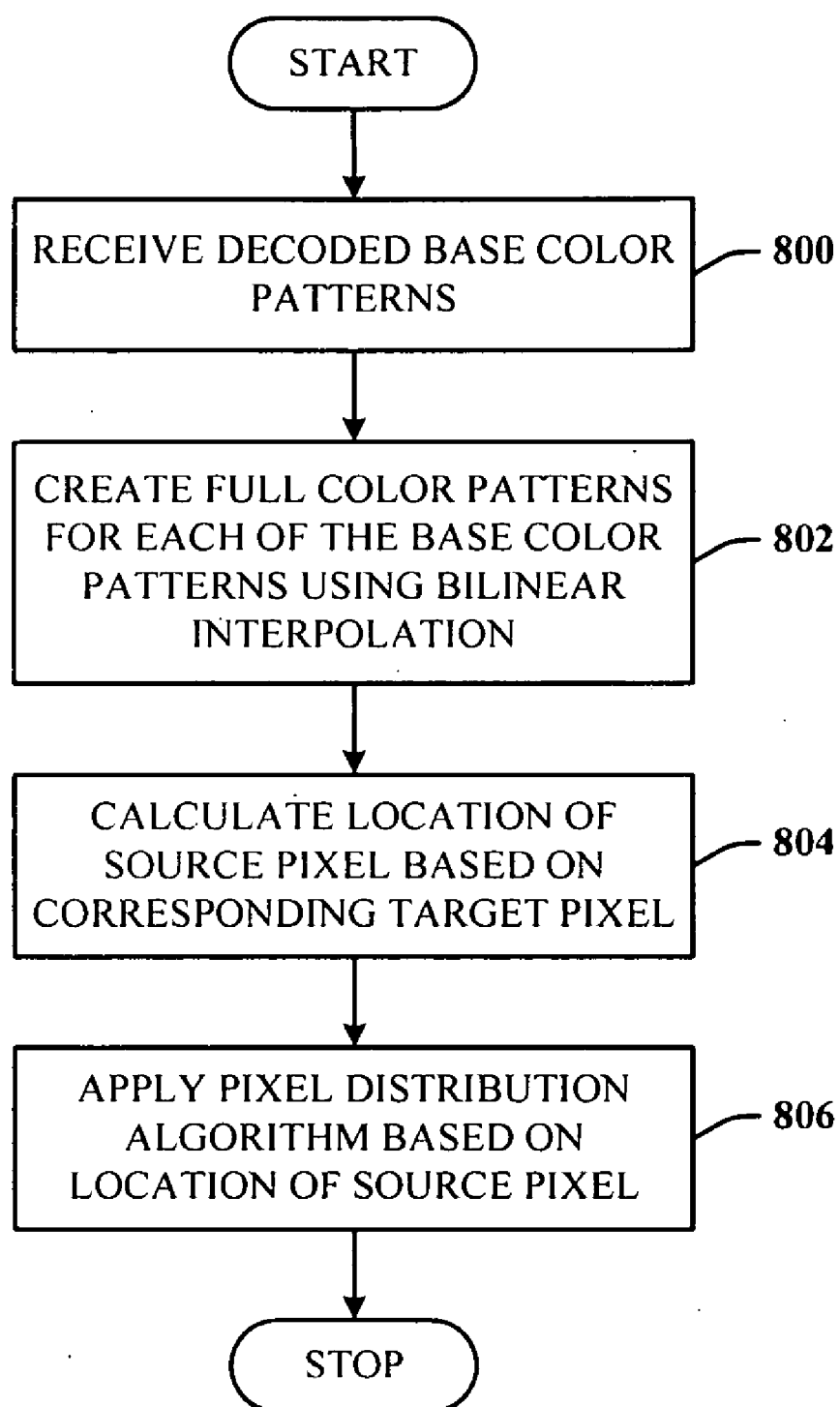
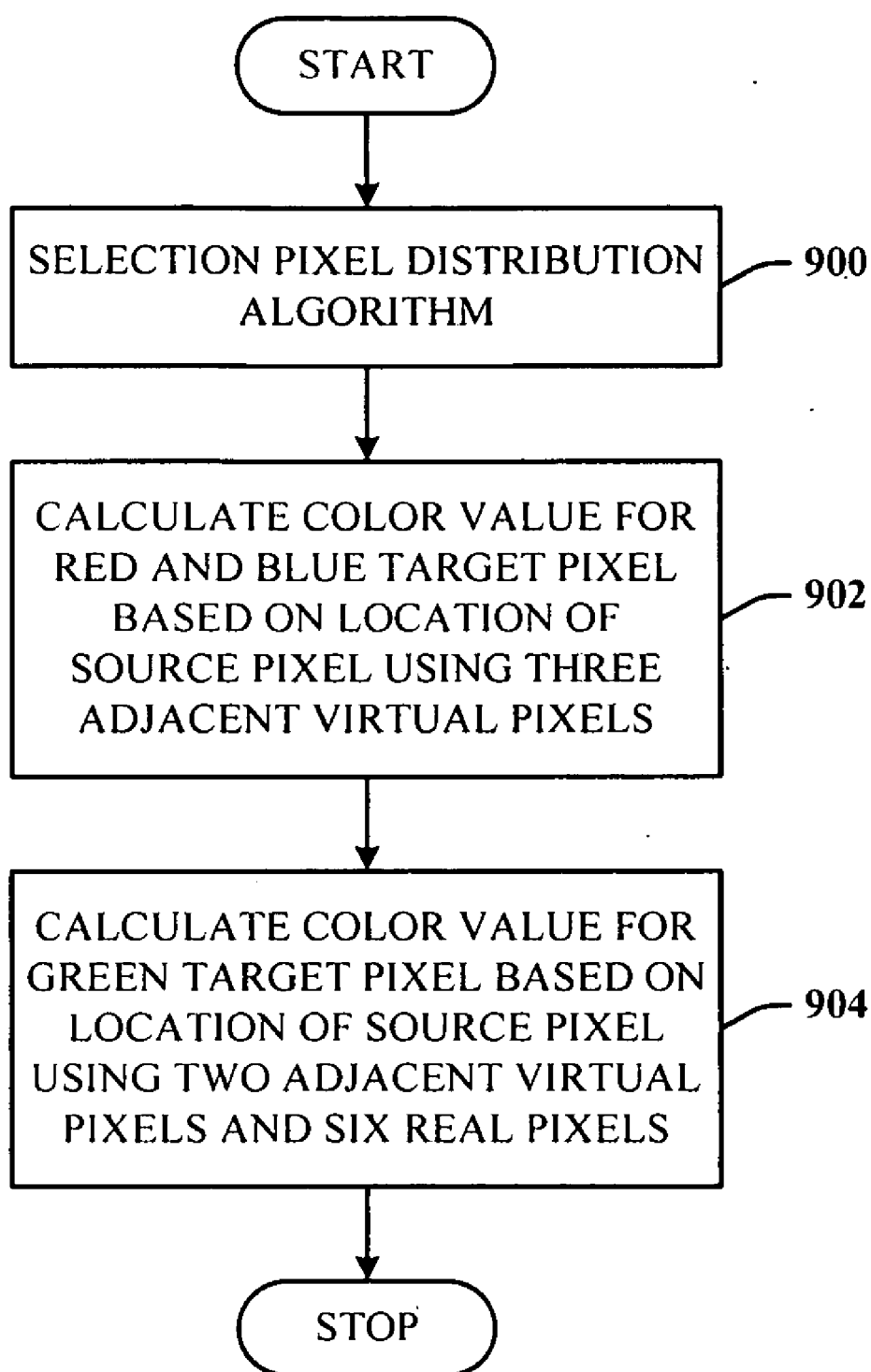


FIG. 5



***FIG. 7***

**FIG. 8**

**FIG. 9**

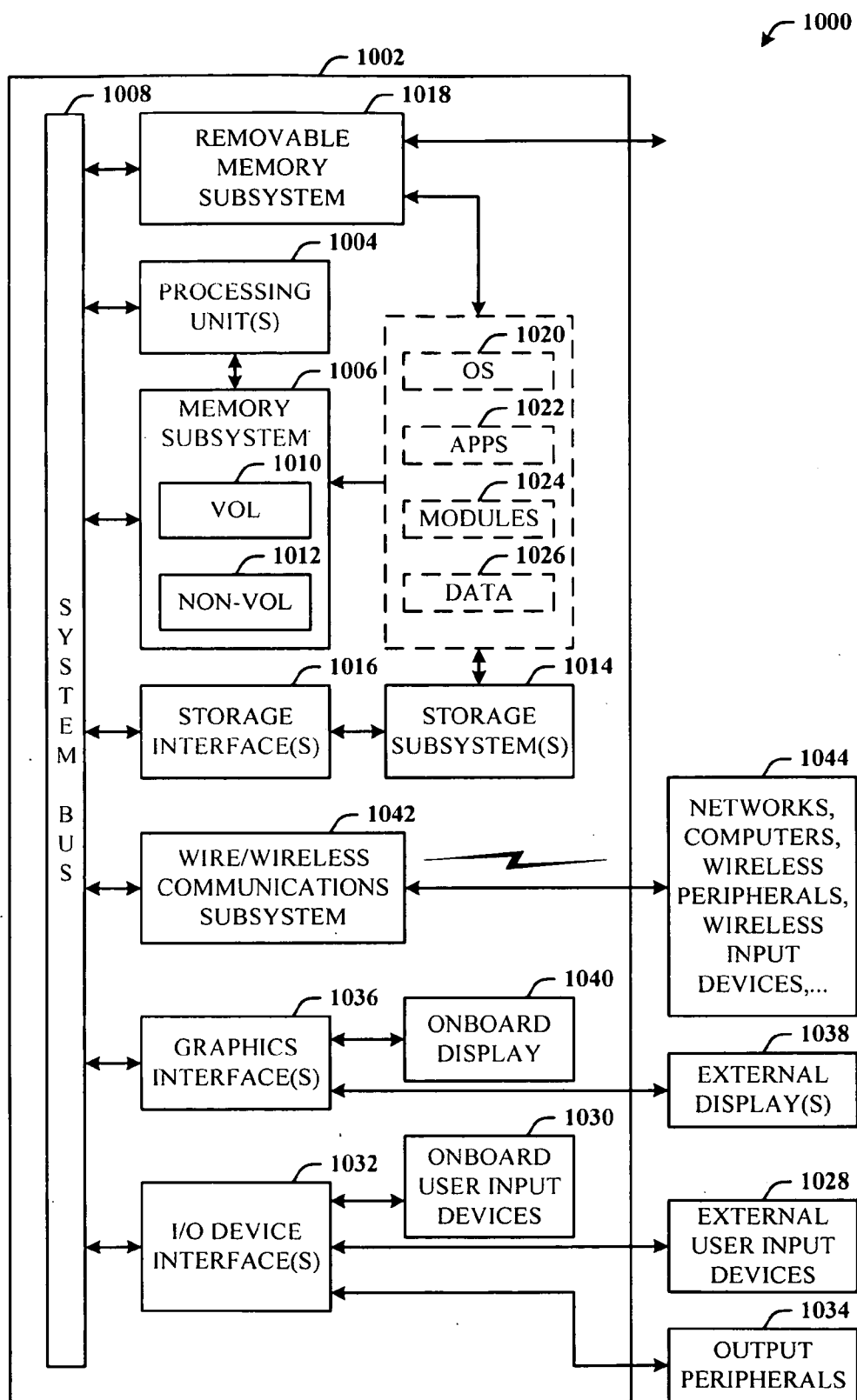


FIG. 10

RECONSTRUCTION OF IMAGE IN A BAYER PATTERN

BACKGROUND

[0001] Single and multi-sensor cameras are predominant in image or video capture systems. The cameras typically employ CCD or CMOS sensors that utilize a Bayer pattern color filter array (where each pixel only contains one color value). When using a Bayer pattern color filter array, the captured image is a mosaic of red, green, and blue (RGB) colors where each pixel is missing two of the three color values. Accordingly, these missing color values need to be interpolated to obtain an approximation of the full RGB values for each pixel to provide a richer output image.

[0002] Vendors can provide different methods to decode the source image to the full RGB patterns in the output image, including bilinear reconstruction, nearest neighbor, cubic, cubic spline, etc., all which have different computational speed and accuracy tradeoffs. As handheld devices become smaller, the amount of available memory and computing power can become limiting factors while consumer demand remains high for better image and video quality.

[0003] However, many of the reconstruction processes for images are based on the full RGB pattern; therefore, memory size and communication bandwidth remain big issues and are not easy to implement in hardware.

SUMMARY

[0004] The following presents a simplified summary in order to provide a basic understanding of some novel embodiments described herein. This summary is not an extensive overview, and it is not intended to identify key/critical elements or to delineate the scope thereof. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

[0005] The disclosed architecture decodes (demosaics) a source image and performs reconstruction directly from the source mosaic pattern (e.g., Bayer) to the target mosaic pattern (e.g., Bayer) to reduce memory size and improve communication bandwidth. Moreover, the architecture can be easily implemented in hardware such as in field programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs).

[0006] The architecture employs multiple different algorithms for calculating the target pixel based on real and virtual source pixels based on the location of the source pixel. The algorithms apply to full red, blue and green base color patterns.

[0007] To the accomplishment of the foregoing and related ends, certain illustrative aspects are described herein in connection with the following description and the annexed drawings. These aspects are indicative of the various ways in which the principles disclosed herein can be practiced, all aspects and equivalents of which are intended to be within the scope of the claimed subject matter. Other advantages and novel features will become apparent from the following detailed description when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 illustrates a computer-implemented image processing system in accordance with the disclosed architecture.

[0009] FIG. 2 illustrates a more detailed image processing system that employs interpolation algorithms for specific interpolation based on the position of the real pixels in the source mosaic.

[0010] FIG. 3 illustrates the resulting equations of the above scenarios for red pixels.

[0011] FIG. 4 illustrates the scenarios of FIG. 3 on pixel sets.

[0012] FIG. 5 illustrates the scenarios for the green pixel reconstruction in a Bayer pattern.

[0013] FIG. 6 illustrates a demosaicing process that decodes a mosaic and the equations for generating interpolated pixels from the real pixels.

[0014] FIG. 7 illustrates a method of processing an image.

[0015] FIG. 8 illustrates a method of reconstructing a mosaic pattern.

[0016] FIG. 9 illustrates a method of applying pixel distribution algorithms.

[0017] FIG. 10 illustrates a block diagram of a computing system operable to execute mosaic reconstruction in accordance with the disclosed architecture.

DETAILED DESCRIPTION

[0018] In the bilinear reconstruction of full RGB patterns, each pixel has three color values related to red (R), green (G) and blue (B). In image/video frame reconstruction, the target pixel is reconstructed from real pixels of the source image/frame. Denoting T as the transform matrix and (X_r, Y_r) the target pixel coordinates, the relevant pixel in the source image/frame is at the position (x, y) . Since x and y are not likely to be integers, the source pixel value can be calculated by interpolating from four adjacent real pixels around the new position in source image/frame.

[0019] The following Equation Set 1 is a bilinear reconstruction method for full RGB patterns.

$$[x, y, 1]' = T \cdot [X_r, Y_r, 1]'$$

$$R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} R(X, Y) \\ R(X + 1, Y) \\ R(X, Y + 1) \\ R(X + 1, Y + 1) \end{bmatrix}$$

$$G(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} G(X, Y) \\ G(X + 1, Y) \\ G(X, Y + 1) \\ G(X + 1, Y + 1) \end{bmatrix}$$

$$B(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} B(X, Y) \\ B(X + 1, Y) \\ B(X, Y + 1) \\ B(X + 1, Y + 1) \end{bmatrix}$$

where $X=[x]$, $Y=[y]$; $[x]$ means the largest integer that is smaller than x ; $\Delta x=x-X$ and $\Delta y=y-Y$.

[0020] Many imagers (e.g., CMOS and CCD image/video sensors) use a Bayer pattern as the color filter array. Because of bandwidth or memory size limitation, it is sometimes desired to have the reconstruction outputs in the Bayer pattern the same as the Bayer pattern of the source image/frame. The relevant pixels in the source image/frame also use the four adjacent real pixels of the same color to interpolate the target

pixel value. The interpolation method is similar to that described above for full RGB reconstruction.

[0021] Demosaicing (also referred to as mosaic decoding) interpolates a mosaic image in a Bayer pattern to an image in full RGB pattern. Bilinear interpolation can also be employed to do this. In bilinear interpolation, real values of real pixels are left untouched, and the interpolated color value is obtained from adjacent real pixel values of the same color. (The following set of equations (here, applied to the red (R) pixels) will be referred to generally herein as Equation Set 3.) For example, where a red pixel R8 in the target image maps to a center position of four surrounding real red pixels, the target red pixel R8 is interpolated by the simple average of the color values of the surrounding real pixels in the source image such that interpolated pixel $R8 = (R2 + R4 + R12 + R14)/4$. Similarly, for a green pattern target pixel that maps back to a position centered in four surrounding real green pixels, the target green pixel G8 is interpolated by the simple average of the values of the surrounding real green pixels in the source image such that interpolated green pixel $G8 = (G3 + G7 + G9 + G13)/4$.

[0022] Where the target pixel maps directly to the source pixel, interpolation is such that the value of the target pixel is equivalent to the value of the real source pixel; thus, for example, an interpolated blue pixel B8 in the target image is the same as the real pixel value such that $B8 = B8$. Where the target pixel maps to a position between two adjacent real pixels, the target pixel R13 in the red pattern, for example, is interpolated as the simple average of the adjacent real red pixel values, such that $R13 = (R12 + R14)/2$.

[0023] The disclosed architecture facilitates reconstruction directly from the Bayer pattern, which improves the performance greatly while keeping the simplicity of implementation in hardware. Unlike reconstruction in full RGB patterns, reconstruction directly from the Bayer pattern facilitates a reduction in memory size and communication bandwidth.

[0024] Reference is now made to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding thereof. It may be evident, however, that the novel embodiments can be practiced without these specific details. In other instances, well known structures and devices are shown in block diagram form in order to facilitate a description thereof. The intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the claimed subject matter.

[0025] FIG. 1 illustrates a computer-implemented image processing system 100 in accordance with the disclosed architecture. The system 100 includes a decoding component 102 for decoding a source image mosaic 104 of pixels of a source image into source base color patterns, and a reconstruction component 106 for reconstructing target pixels in a target mosaic 108 of a target image directly from the source base color patterns.

[0026] As will be described herein, the mosaic of the source image and the mosaic of the target image can be Bayer patterns. The reconstruction component 106 selects an interpolation algorithm that calculates a color value for a target pixel based on a source pixel distribution that includes real and interpolated source pixels. The pixel distribution includes three interpolated source pixels and a real pixel, and applies to red and blue base color patterns. The pixel distribution can

also include two interpolated source pixels and six real pixels, which applies to a green base color pattern.

[0027] In other words, the reconstruction occurs directly from the source mosaic to the target mosaic as a concurrent operation. The mosaics can be Bayer patterns such that the two processes of demosaicing and bilinear reconstruction are combined using equations that facilitate the direct reconstruction to provide better performance with nearly equal simplicity.

[0028] FIG. 2 illustrates a more detailed image processing system 200 that employs interpolation algorithms 202 for specific interpolation based on the position of the real pixels in the source mosaic 104. The system 200 includes the decoding component 102 for decoding a source Bayer pattern of pixels of a source image into source base color patterns, and the reconstruction component 106 for reconstructing target pixels in a target mosaic of a target image directly from the source base color patterns to generate target pixels in a target Bayer pattern of a target image. The reconstruction component 106 selects one of the interpolation algorithms 202 that calculates an interpolated color value for a target pixel based on a real and interpolated pixel distribution in the base color patterns.

[0029] As is described herein below, the pixel distribution can include three interpolated source pixels and a real pixel for red and blue base color patterns. The pixel distribution can include two interpolated source pixels and six real pixels for a green base color pattern.

[0030] The description for the interpolation algorithms 202 is based on red, blue and green pixels; however, it is to be understood that other color spaces can be employed, such as for luma (Y) and chroma (Cr and Cb) components, for example.

[0031] The equations below are derived from the Equations Set 1 and Equation Set 3 above. The derivation for the concurrent computation of decoding and reconstruction begins with an input image in full RGB pattern, and according to the Equation Set 1, a target pixel value can be obtained in the output (or target) image using:

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} R(X, Y) \\ R(X + 1, Y) \\ R(X, Y + 1) \\ R(X + 1, Y + 1) \end{bmatrix}$$

[0032] However, in the disclosed implementation, the source image is a mosaic, as well as the target image. Since the images are mosaics (e.g., Bayer) patterns, this means that the source image is not a full RGB pattern. Demosaicing is used to obtain an interpolated (or “virtual”) full RGB pattern. Thus, for the above equation, there can be four scenarios, described in general form as the following.

[0033] Scenario S1 is where $R(X, Y)$ is a real pixel, while $R(X+1, Y)$, $R(X, Y+1)$, and $R(X+1, Y+1)$ are not real pixels in the Bayer pattern of the source image. These non-real pixel values are then calculated from other real pixels, such that,

$$R(X + 1, Y) = \frac{R(X, Y) + R(X + 2, Y)}{2}$$

-continued

$$R(X, Y + 1) = \frac{R(X, Y) + R(X, Y + 2)}{2}$$

$$R(X + 1, Y + 1) = \frac{R(X, Y) + R(X + 2, Y) + R(X, Y + 2) + R(X + 2, Y + 2)}{4}$$

[0034] Scenario S2 is where $R(X+1, Y)$ is a real pixel, while $R(X, Y)$, $R(X, Y+1)$, and $R(X+1, Y+1)$ are non-real pixels in the Bayer pattern, and are calculated from other real pixels.

$$R(X, Y) = \frac{R(X - 1, Y) + R(X + 1, Y)}{2}$$

$$R(X, Y + 1) = \frac{R(X - 1, Y + 2) + R(X + 1, Y + 2)}{4}$$

$$R(X + 1, Y + 1) = \frac{R(X + 1, Y) + R(X + 1, Y + 2)}{2}$$

[0035] Scenario S3 is where $R(X, Y+1)$ is a real pixel, while $R(X, Y)$, $R(X+1, Y)$, and $R(X+1, Y+1)$ are non-real pixels in the Bayer pattern, and are calculated from other real pixels.

$$R(X, Y) = \frac{R(X, Y - 1) + R(X, Y + 1)}{2}$$

$$R(X + 1, Y) = \frac{R(X, Y - 1) + R(X + 2, Y - 1) + R(X, Y + 1) + R(X + 2, Y + 1)}{4}$$

$$R(X + 1, Y + 1) = \frac{R(X, Y + 1) + R(X + 2, Y + 1)}{2}$$

[0036] Scenario S4 is where, $R(X+1, Y+1)$ is a real pixel, while $R(X, Y)$, $R(X+1, Y)$, and $R(X, Y+1)$ are non-real pixels in the Bayer pattern, and are calculated from other real pixels.

$$R(X, Y) = \frac{R(X - 1, Y - 1) + R(X + 1, Y - 1) + R(X - 1, Y + 1) + R(X + 1, Y + 1)}{4}$$

$$R(X + 1, Y) = \frac{R(X + 1, Y - 1) + R(X + 1, Y + 1)}{2}$$

$$R(X, Y + 1) = \frac{R(X - 1, Y + 1) + R(X + 1, Y + 1)}{2}$$

[0037] Making the substitutions in the Equation Set 1 for the red (R) pixels, the scenarios are the following. In scenario S1,

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} R(X, Y) \\ R(X + 1, Y) \\ R(X, Y + 1) \\ R(X + 1, Y + 1) \end{bmatrix} =$$

-continued

$$\begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} \frac{R(X, Y)R(X, Y) + R(X + 2, Y)}{2} \\ \frac{R(X, Y) + R(X, Y + 2)}{2} \\ \frac{R(X, Y) + R(X + 2, Y) + R(X, Y + 2) + R(X + 2, Y + 2)}{4} \end{bmatrix} =$$

$$\begin{bmatrix} 1 - \frac{\Delta x}{2} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x \Delta y}{4} \end{bmatrix}^T \cdot \begin{bmatrix} R(X, Y) \\ R(X + 2, Y) \\ R(X, Y + 2) \\ R(X + 2, Y + 2) \end{bmatrix}$$

[0038] In scenario S2,

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} R(X, Y) \\ R(X + 1, Y) \\ R(X, Y + 1) \\ R(X + 1, Y + 1) \end{bmatrix} =$$

$$\begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} \frac{R(X - 1, Y) + R(X + 1, Y)}{2} \\ \frac{R(X + 1, Y)R(X - 1, Y) + R(X + 1, Y) + R(X - 1, Y + 2) + R(X + 1, Y + 2)}{4} \\ \frac{R(X + 1, Y) + R(X + 1, Y + 2)}{2} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{2} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} + \frac{\Delta x}{2} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix}^T \cdot \begin{bmatrix} R(X - 1, Y) \\ R(X + 1, Y) \\ R(X - 1, Y + 2) \\ R(X + 1, Y + 2) \end{bmatrix}$$

[0039] In scenario S3,

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}^T \cdot \begin{bmatrix} R(X, Y) \\ R(X + 1, Y) \\ R(X, Y + 1) \\ R(X + 1, Y + 1) \end{bmatrix} =$$

-continued

$$\begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} \frac{R(X, Y-1) + R(X, Y+1)}{2} \\ \frac{R(X, Y-1) + R(X+2, Y-1) + R(X, Y+1) + R(X+2, Y+1)}{4} \\ \frac{R(X, Y+1)R(X, Y+1) + R(X+2, Y+1)}{2} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{4} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} - \frac{\Delta x}{4} + \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix}' \cdot \begin{bmatrix} R(X, Y-1) \\ R(X+2, Y-1) \\ R(X, Y+1) \\ R(X+2, Y+1) \end{bmatrix}$$

[0040] In scenario S4,

$$R(X_t, Y_t) = R(x, y) = \begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} R(X, Y) \\ R(X+1, Y) \\ R(X, Y+1) \\ R(X+1, Y+1) \end{bmatrix} =$$

$$\begin{bmatrix} 1 - \Delta x - \Delta y + \Delta x \Delta y \\ \Delta x - \Delta x \Delta y \\ \Delta y - \Delta x \Delta y \\ \Delta x \Delta y \end{bmatrix}' \cdot \begin{bmatrix} \frac{R(X-1, Y-1) + R(X+1, Y-1) + R(X-1, Y+1) + R(X+1, Y+1)}{4} \\ \frac{R(X+1, Y-1) + R(X+1, Y+1)}{2} \\ \frac{R(X-1, Y+1) + R(X+1, Y+1)}{2} \\ R(X+1, Y+1) \end{bmatrix} =$$

$$\begin{bmatrix} \frac{1}{4} - \frac{\Delta x}{4} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} - \frac{\Delta x}{4} + \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} + \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix}' \cdot \begin{bmatrix} R(X-1, Y-1) \\ R(X+1, Y-1) \\ R(X-1, Y+1) \\ R(X+1, Y+1) \end{bmatrix}$$

[0041] FIG. 3 illustrates the resulting equations of the above scenarios 300 for red pixels. Although derived for red pixels, the above scenarios also apply to the blue (B) pixels. Equation 302 (also referred to as scenario S1) calculates the color values for interpolated pixels when the real pixel is in a top-left position of a 4-pixel group. Equation 304 (also referred to as scenario S2) calculates the color values for

interpolated pixels when the real pixel is in a top-right position of a 4-pixel group. Equation 306 (also referred to as scenario S3) calculates the color values for interpolated pixels when the real pixel is in a bottom-left position of a 4-pixel group. Equation 308 (also referred to as scenario S4) calculates the color values for interpolated pixels when the real pixel is in a bottom-right position of a 4-pixel group.

[0042] FIG. 4 illustrates the scenarios of FIG. 3 on pixel sets 400. On a first pixel set 402, the first scenario S1 selects equation 302, since the real pixel P11 is in the top-left corner of three adjacent interpolated pixels P12, P21, and P22. On a second pixel set 404, the second scenario S2 selects equation 304, since the real pixel P13 is in the top-right corner of three adjacent interpolated pixels P12, P22, and P23. On a third pixel set 406, the third scenario S3 selects equation 306, since the real pixel P31 is in the bottom-left corner of three adjacent interpolated pixels P21, P22, and P32. On a fourth pixel set 408, the fourth scenario S4 selects equation 308, since the real pixel P33 is in the bottom-right corner of three adjacent interpolated pixels P22, P23, and P32.

[0043] From above, it is known that in full RGB pattern reconstruction, the bilinear method uses the four adjacent real pixels of the same color to interpolate the target pixel value. In FIG. 4, it is also desirable to real source pixels are used to interpolate the target pixel value; however, three of the four adjacent pixels are not real pixels captured from sensor, but virtual pixels interpolated from other real pixels. Thus, there can be four different pixel distributions for performing interpolation for reconstruction.

[0044] FIG. 5 illustrates the scenarios 500 for the green pixel reconstruction in a Bayer pattern. The scenarios 500 include a first scenario 502 and a second scenario 504. The first scenario S1 involves real pixels in the top-left and bottom-right positions, and the second scenario S2 involves real pixels in the top-right and bottom-left positions.

[0045] Although not derived here, the derivation method for the green pixel reconstruction is similar to the derivation applied for the red and blue pixels. The number of green pixels in the Bayer pattern is twice the number of the red or blue pixels. For green pixels, the relevant pixel in the source image may locate in the evaluation box of scenario S1 or scenario S2. Combining the equations for bilinear reconstruction and demosaicing, the target green pixel values can be calculated as in Equation 3 below.

[0046] For pixels located in the evaluation box of scenario S1:

$$G(x, y) = \begin{bmatrix} \frac{\Delta x}{4} - \frac{\Delta x \Delta y}{4} \\ 1 - \frac{3\Delta x}{4} - \frac{3\Delta y}{4} + \frac{\Delta x \Delta y}{2} \\ \frac{\Delta x}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{4} + \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{2} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \end{bmatrix}' \cdot \begin{bmatrix} P_{13} \\ P_{22} \\ P_{24} \\ P_{31} \\ P_{33} \\ P_{42} \end{bmatrix}$$

[0047] For pixels located in the evaluation box of scenario S2:

$$G(x, y) = \begin{bmatrix} \frac{1}{4} - \frac{\Delta x}{4} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} - \frac{\Delta x}{4} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{3\Delta x}{4} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{2} \\ \frac{1}{4} - \frac{\Delta x}{4} + \frac{3\Delta y}{4} - \frac{\Delta x \Delta y}{2} \\ \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} P_{13} \\ P_{22} \\ P_{24} \\ P_{33} \\ P_{35} \\ P_{44} \end{bmatrix}$$

where $[x, y, 1]^T = T^*[X_r, Y_r, 1]^T$; $X=[x]$ and $Y=[y]$; $[x]$ means the largest integer that is smaller than x , and $\Delta x = x - X$ and $\Delta y = y - Y$.

[0048] In other words, scenario S1 calculates the interpolated pixels P23 and P32 based on real pixels P13, P22, P24, P31, P33, and P42. The second scenario S2 calculates the interpolated pixels P23 and P34 based on real pixels P13, P22, P24, P33, P35, and P44.

[0049] FIG. 6 illustrates a demosaicing process 600 that decodes a mosaic 602 and the equations for generating interpolated pixels from the real pixels. The mosaic 602 includes red pixels (denoted as octagons), blue pixels (denoted as squares) and green pixels (denoted as diamonds). The mosaic 602 is decoded into the three base (or constituent) color patterns: a green (G) pixel pattern 604, a blue (B) pixel pattern 606, and a red (R) pixel pattern 608. In the green color pattern 604, the real pixels are, for example, G1, G3, G5, etc. The Equation Set 3 describes the general methods for computing the interpolated pixels G2, G4, G6, etc. For example, the interpolated pixel color value for G2 is calculated as $(G1 + G3)/2$; the interpolated pixel color value for G8 is calculated as $(G3 + G7 + G9 + G13)/4$, and so on, for all interpolated pixels. The same Equation Set 3 applies for calculating the interpolated color values for the blue pixel pattern 606 and the red pixel pattern 608.

[0050] The following example applies one of the equation scenarios above for red pixel reconstruction for Bayer pattern. The pixels represented by circles with cross-hatching are the virtual target pixels interpolated from adjacent real source pixels by the reconstruction method. The pixels represented as octagons are the real (original) red pixels of the source image captured by the image sensor. In other words, P11 is the real pixel that captured from sensor, and P12, P21, P22 are the virtual target pixels that are interpolated from other real source pixels. For example, $P11 = P11$, $P12 = (P11 + P13)/2$; $P21 = (P11 + P31)/2$; and, $P22 = (P11 + P13 + P31 + P33)/4$.

[0051] The transform matrix T can be the following, for example.

$$T = \begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

Accordingly, if $X_r = 1$ and $Y_r = 1$, then $x = 1.5$ and $y = 1.5$ (using Equation Set 1) such that the source real pixel becomes P11.

Thus, the first scenario S1 applies where the real pixel is in the top-left corner. In another example, if $X_r = 2$ and $Y_r = 2$, then $x = 2.5$ and $y = 2.5$ (using Equation Set 1), and real pixel P22 is in bottom-right corner such that the equation of scenario S4 applies. This process is the same for the blue pixels.

[0052] For pixels located in the evaluation box of scenario S1:

$$R(x, y) = \begin{bmatrix} 1 - \frac{\Delta x}{2} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{13} \\ P_{31} \\ P_{33} \end{bmatrix}$$

[0053] For pixels located in the evaluation box of scenario S2:

$$R(x, y) = \begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{2} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} + \frac{\Delta x}{2} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{13} \\ P_{31} \\ P_{33} \end{bmatrix}$$

[0054] For pixels located in the evaluation box of scenario S3:

$$R(x, y) = \begin{bmatrix} \frac{1}{2} - \frac{\Delta x}{4} - \frac{\Delta y}{2} + \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{2} - \frac{\Delta x}{4} + \frac{\Delta y}{2} - \frac{\Delta x \Delta y}{4} \\ \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{13} \\ P_{31} \\ P_{33} \end{bmatrix}$$

[0055] For pixels located in the evaluation box of scenario S4:

$$R(x, y) = \begin{bmatrix} \frac{1}{4} - \frac{\Delta x}{4} - \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} - \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} - \frac{\Delta x}{4} + \frac{\Delta y}{4} - \frac{\Delta x \Delta y}{4} \\ \frac{1}{4} + \frac{\Delta x}{4} + \frac{\Delta y}{4} + \frac{\Delta x \Delta y}{4} \end{bmatrix} \cdot \begin{bmatrix} P_{11} \\ P_{13} \\ P_{31} \\ P_{33} \end{bmatrix}$$

where $[x, y, 1]^T = T^*[X_r, Y_r, 1]^T$; $X=[x]$, $Y=[y]$; $[x]$ means the largest integer that is smaller than x , and $\Delta x = x - X$, $\Delta y = y - Y$.

[0056] Included herein is a set of flow charts representative of exemplary methodologies for performing novel aspects of the disclosed architecture. While, for purposes of simplicity of explanation, the one or more methodologies shown herein, for example, in the form of a flow chart or flow diagram, are

shown and described as a series of acts, it is to be understood and appreciated that the methodologies are not limited by the order of acts, as some acts may, in accordance therewith, occur in a different order and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of inter-related states or events, such as in a state diagram. Moreover, not all acts illustrated in a methodology may be required for a novel implementation.

[0057] FIG. 7 illustrates a method of processing an image. At 700, a source mosaic of pixels of a source image is decoded into base color patterns. At 702, the base color patterns are reconstructed concurrently with the decoding to generate target pixels in a target mosaic of a target image. In other words, the target mosaic is reconstructed directly from the input mosaic.

[0058] FIG. 8 illustrates a method of reconstructing a mosaic pattern. At 800, decoded base color patterns are received. At 802, a full color pattern is created for each of the base color patterns using bilinear interpolation. At 804, the location of the source pixel is calculated based on the corresponding target pixel. At 806, a pixel distribution algorithm is applied based on the location of the source (real) pixel.

[0059] FIG. 9 illustrates a method of applying pixel distribution algorithms. At 900, a pixel distribution algorithm is selected. At 902, the color value for the red and blue target pixels is calculated based on the location of the source pixel using three adjacent virtual pixels. At 904, the color value for the green target pixels is calculated based on the location of the six source pixels and using two adjacent virtual pixels.

[0060] As used in this application, the terms “component” and “system” are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component can be, but is not limited to being, a process running on a processor, a processor, a hard disk drive, multiple storage drives (of optical and/or magnetic storage medium), an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and/or thread of execution, and a component can be localized on one computer and/or distributed between two or more computers. The word “exemplary” may be used herein to mean serving as an example, instance, or illustration. Any aspect or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects or designs.

[0061] Referring now to FIG. 10, there is illustrated a block diagram of a computing system 1000 operable to execute mosaic reconstruction in accordance with the disclosed architecture. In order to provide additional context for various aspects thereof, FIG. 10 and the following discussion are intended to provide a brief, general description of the suitable computing system 1000 in which the various aspects can be implemented. While the description above is in the general context of computer-executable instructions that can run on one or more computers, those skilled in the art will recognize that a novel embodiment also can be implemented in combination with other program modules and/or as a combination of hardware and software.

[0062] The computing system 1000 for implementing various aspects includes the computer 1002 having processing

unit(s) 1004, a system memory 1006, and a system bus 1008. The processing unit(s) 1004 can be any of various commercially available processors such as single-processor, multi-processor, single-core units and multi-core units. Moreover, those skilled in the art will appreciate that the novel methods can be practiced with other computer system configurations, including minicomputers, mainframe computers, as well as personal computers (e.g., desktop, laptop, etc.), hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like, each of which can be operatively coupled to one or more associated devices.

[0063] The system memory 1006 can include volatile (VOL.) memory 1010 (e.g., random access memory (RAM)) and non-volatile memory (NON-VOL.) 1012 (e.g., ROM, EPROM, EEPROM, etc.). A basic input/output system (BIOS) can be stored in the non-volatile memory 1012, and includes the basic routines that facilitate the communication of data and signals between components within the computer 1002, such as during startup. The volatile memory 1010 can also include a high-speed RAM such as static RAM for caching data.

[0064] The system bus 1008 provides an interface for system components including, but not limited to, the memory subsystem 1006 to the processing unit(s) 1004. The system bus 1008 can be any of several types of bus structure that can further interconnect to a memory bus (with or without a memory controller), and a peripheral bus (e.g., PCI, PCIE, AGP, LPC, etc.), using any of a variety of commercially available bus architectures.

[0065] The computer 1002 further includes storage subsystem(s) 1014 and storage interface(s) 1016 for interfacing the storage subsystem(s) 1014 to the system bus 1008 and other desired computer components. The storage subsystem(s) 1014 can include one or more of a hard disk drive (HDD), a magnetic floppy disk drive (FDD), and/or optical disk storage drive (e.g., a CD-ROM drive DVD drive), for example. The storage interface(s) 1016 can include interface technologies such as EIDE, ATA, SATA, and IEEE 1394, for example.

[0066] One or more programs and data can be stored in the memory subsystem 1006, a removable memory subsystem 1018 (e.g., flash drive form factor technology), and/or the storage subsystem(s) 1014, including an operating system 1020, one or more application programs 1022, other program modules 1024, and program data 1026. The one or more application programs 1022, other program modules 1024, and program data 1026 can include the disclosed decoding and reconstruction architecture for direct mosaic-to-mosaic reconstruction. Accordingly, the one or more application programs 1022, other program modules 1024, and program data 1026 can include the system and elements of FIG. 1, FIG. 2, equations 300, pixel interpolation distributions of FIG. 4 and FIG. 5, and methods of FIGS. 7-9. Alternatively, or in combination therewith, all or portions of the architecture can be embodied in hardware such as a FPGA, for example.

[0067] Generally, programs include routines, methods, data structures, other software components, etc., that perform particular tasks or implement particular abstract data types. All or portions of the operating system 1020, applications 1022, modules 1024, and/or data 1026 can also be cached in memory such as the volatile memory 1010, for example. It is to be appreciated that the disclosed architecture can be implemented with various commercially available operating systems or combinations of operating systems (e.g., as virtual machines).

[0068] The storage subsystem(s) **1014** and memory subsystems (**1006** and **1018**) serve as computer readable media for volatile and non-volatile storage of data, data structures, computer-executable instructions, and so forth. Computer readable media can be any available media that can be accessed by the computer **1002** and includes volatile and non-volatile media, removable and non-removable media. For the computer **1002**, the media accommodate the storage of data in any suitable digital format. It should be appreciated by those skilled in the art that other types of computer readable media can be employed such as zip drives, magnetic tape, flash memory cards, cartridges, and the like, for storing computer executable instructions for performing the novel methods of the disclosed architecture.

[0069] A user can interact with the computer **1002**, programs, and data using external user input devices **1028** such as a keyboard and a mouse. Other external user input devices **1028** can include a microphone, an IR (infrared) remote control, a joystick, a game pad, camera recognition systems, a stylus pen, touch screen, gesture systems (e.g., eye movement, head movement, etc.), and/or the like. The user can interact with the computer **1002**, programs, and data using onboard user input devices **1030** such as a touchpad, microphone, keyboard, etc., where the computer **1002** is a portable computer, for example. These and other input devices are connected to the processing unit(s) **1004** through input/output (I/O) device interface(s) **1032** via the system bus **1008**, but can be connected by other interfaces such as a parallel port, IEEE 1394 serial port, a game port, a USB port, an IR interface, etc. The I/O device interface(s) **1032** also facilitate the use of output peripherals **1034** such as printers, audio devices, camera devices, and so on, such as a sound card and/or onboard audio processing capability.

[0070] One or more graphics interface(s) **1036** (also commonly referred to as a graphics processing unit (GPU)) provide graphics and video signals between the computer **1002** and external display(s) **1038** (e.g., LCD, plasma) and/or onboard displays **1040** (e.g., for portable computer). The graphics interface(s) **1036** can also be manufactured as part of the computer system board.

[0071] The computer **1002** can operate in a networked environment (e.g., IP) using logical connections via a wire/wireless communications subsystem **1042** to one or more networks and/or other computers. The other computers can include workstations, servers, routers, personal computers, microprocessor-based entertainment appliance, a peer device or other common network node, and typically include many or all of the elements described relative to the computer **1002**. The logical connections can include wire/wireless connectivity to a local area network (LAN), a wide area network (WAN), hotspot, and so on. LAN and WAN networking environments are commonplace in offices and companies and facilitate enterprise-wide computer networks, such as intranets, all of which may connect to a global communications network such as the Internet.

[0072] When used in a networking environment the computer **1002** connects to the network via a wire/wireless communication subsystem **1042** (e.g., a network interface adapter, onboard transceiver subsystem, etc.) to communicate with wire/wireless networks, wire/wireless printers, wire/wireless input devices **1044**, and so on. The computer **1002** can include a modem or has other means for establishing communications over the network. In a networked environment, programs and data relative to the computer **1002** can be

stored in the remote memory/storage device, as is associated with a distributed system. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

[0073] The computer **1002** is operable to communicate with wire/wireless devices or entities using the radio technologies such as the IEEE 802.xx family of standards, such as wireless devices operatively disposed in wireless communication (e.g., IEEE 802.11 over-the-air modulation techniques) with, for example, a printer, scanner, desktop and/or portable computer, personal digital assistant (PDA), communications satellite, any piece of equipment or location associated with a wirelessly detectable tag (e.g., a kiosk, news stand, restroom), and telephone. This includes at least Wi-Fi (or Wireless Fidelity) for hotspots, WiMax, and Bluetooth™ wireless technologies. Thus, the communications can be a predefined structure as with a conventional network or simply an ad hoc communication between at least two devices. Wi-Fi networks use radio technologies called IEEE 802.11x (a, b, g, etc.) to provide secure, reliable, fast wireless connectivity. A Wi-Fi network can be used to connect computers to each other, to the Internet, and to wire networks (which use IEEE 802.3-related media and functions).

[0074] What has been described above includes examples of the disclosed architecture. It is, of course, not possible to describe every conceivable combination of components and/or methodologies, but one of ordinary skill in the art may recognize that many further combinations and permutations are possible. Accordingly, the novel architecture is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

What is claimed is:

1. A computer-implemented image processing system, comprising:
 - a decoding component for decoding a source mosaic of pixels of a source image into source base color patterns; and
 - a reconstruction component for reconstructing target pixels in a target mosaic of a target image directly from the source base color patterns.
2. The system of claim 1, wherein the mosaic of the source image and the mosaic of the target image are Bayer patterns.
3. The system of claim 1, wherein the reconstruction component selects an interpolation algorithm that calculates a color value for a target pixel based on a source pixel distribution that includes real and interpolated source pixels.
4. The system of claim 3, wherein the pixel distribution includes three interpolated source pixels and a real pixel.
5. The system of claim 4, wherein the pixel distribution applies to red and blue base color patterns.
6. The system of claim 3, wherein the pixel distribution includes two interpolated source pixels and six real pixels.
7. The system of claim 6, wherein the pixel distribution applies to a green base color pattern.
8. A computer-implemented image processing system, comprising:

a decoding component for decoding a source Bayer pattern of pixels of a source image into source base color patterns; and

a reconstruction component for reconstructing target pixels in a target mosaic of a target image directly from the source base color patterns to generate target pixels in a target Bayer pattern of a target image, the reconstruction component selecting an interpolation algorithm that calculates an interpolated color value for a target pixel based on a real and interpolated pixel distribution in the base color patterns.

9. The system of claim 8, wherein the pixel distribution includes three interpolated source pixels and a real pixel.

10. The system of claim 9, wherein the pixel distribution applies to red and blue base color patterns.

11. The system of claim 8, wherein the pixel distribution includes two interpolated source pixels and six real pixels.

12. The system of claim 11, wherein the pixel distribution applies to a green base color pattern.

13. A computer-implemented method of processing an image, comprising:

decoding a source mosaic of pixels of a source image into source base color patterns; and

reconstructing corresponding target base color patterns directly from the source color patterns to generate a target mosaic of a target image.

14. The method of claim 13, wherein the source mosaic and the target mosaic are a same color pattern of pixels.

15. The method of claim 13, further comprising creating full color patterns for the source base color patterns using bilinear interpolation.

16. The method of claim 13, further comprising calculating location of a corresponding source pixel based on a target pixel and applying a pixel distribution algorithm based on location of the source pixel.

17. The method of claim 16, wherein the pixel distribution algorithm calculates a color value for the target pixel based on three virtual pixels adjacent to the source pixel.

18. The method of claim 17, wherein the target pixel is blue or red.

19. The method of claim 16, wherein the pixel distribution algorithm calculates a color value for the target pixel based on two virtual pixels adjacent to six real pixels.

20. The method of claim 13, wherein the target pixel is green.

* * * * *