AN EFFICIENT COLOR IMAGE ACQUISITION SYSTEM FOR WIRELESS HANDHELD DEVICES

Bei Tang and King F. Lee

Multimedia Architecture Lab
Motorola Labs
Schaumburg, Illinois 60196
USA

ABSTRACT

This paper presents a low-cost, low-power, and yet highly effective color image acquisition system for wireless handheld devices such as cellular camera phones and PDAs. Common color image acquisition system captures Bayer pattern image data from CMOS/CCD color image sensor and processes it to generate YCbCr color signal for the image compression engine. The proposed system uses a novel vertex-centered color interpolation algorithm for parallel interpolation of groups of pixels within a window and directly processes the image from Bayer pattern data to the YCbCr color space. The proposed system also uses pseudomedian filtering on the Cb and Cr color channels to enhance image quality with low hardware complexity.

1. INTRODUCTION

Camera functions are becoming increasingly popular in wireless handheld devices such as cellular phones and personal digital assistants (PDAs). As cellular camera phones turn into a new trend in the consumer electronics market, designers face significant technical challenges to develop a high quality, low-cost, and low-power camera module that is small enough to fit into a cellular phone. This work is motivated by the need to design a simple yet highly efficient image acquisition module [1] for cellular camera phones. We propose a novel color interpolation approach to simplify the image acquisition flow while still achieving image quality better than that of the more complex traditional image acquisition flow.

The remaining part of this paper is organized as follows. Section 2 presents the traditional image acquisition chain with pixel-centered interpolation. Section 3 describes the proposed image acquisition system with vertex-centered interpolation to obtain YCbCr image directly from raw Bayer pattern data and pseudomedian filtering to remove difficult color artifacts in the image. Section 4 compares the results from a traditional chain and the proposed flow. The article concludes with summary remarks in Section 5.

2. TRADITIONAL IMAGE ACQUISITION CHAIN

A typical color image acquisition chain and its data flow are shown in Figure 1 and Figure 2.

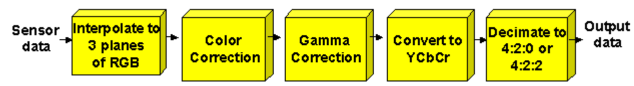

Figure 1. Traditional image acquisition chain

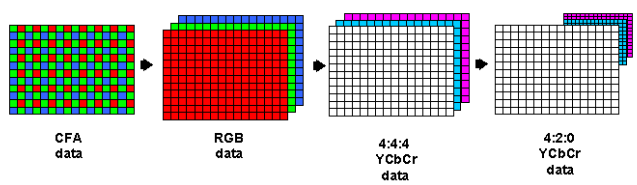

Figure 2. Data flow in a traditional acquisition chain

It consists of five major functional blocks: color interpolation, color correction, gamma correction, color space conversion, and color data decimation. Color interpolation is the most important and computational intensive step in a color image acquisition chain. Most color image sensors capture images with Bayer pattern color filter array (CFA) [2] where only one of the three color components is recorded at each pixel location. In order to obtain full color images, Bayer pattern data from the sensor is interpolated to generate the three planes of RGB data. Color correction is a processing step that "fine tunes" the image color so it replicates the actual scene color. The gamma correction step makes the input signal proportional to the light intensity for the display. The processed image data is then converted to the YCbCr space and down-sampled to the 4:2:0 or 4:2:2 format usually required by the image compression engine [3].

2.1. Pixel-Centered Interpolation

Bayer pattern data from color image sensors contain only one color component at each pixel. Therefore, the color interpolation step has to calculate the other missing color components from the neighboring pixels.

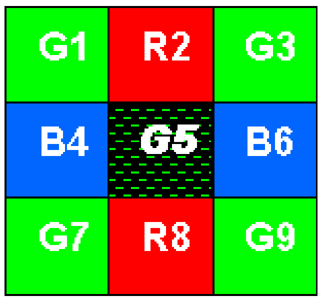

Figure 3. A typical RGB color filter array with pixel-centered interpolation

Many color interpolation algorithms have been developed [4]. Most of the previously proposed algorithms are based on pixel-centered interpolation that commonly uses a 3x3 window to interpolate the missing color components for the center pixel. Consider the center pixel 5 in Figure 3,

$$P5 = f\,(G1, R2, G3, B4, G5, B6, G7, R8, G9) \qquad (1)$$

where $f(.)$ is an interpolation algorithm that uses Bayer pattern data from the 9 pixels within the window to calculate the full RGB information for pixel 5. For handheld devices, the traditional bilinear interpolation is often chosen for its relative simplicity. For pixel 5, its bilinear interpolated RGB values are given by (2).

$$\begin{aligned} R5 &= (R2 + R8)/2 \\ G5 &= G5 \;\; \text{or} \;\; G5 = ((G1 + G3 + G7 + G9)/4 + G5)/2 \\ B5 &= (B4 + B6)/2 \end{aligned} \qquad (2)$$

Pixel-centered interpolation only provides the missing color components for the pixel at the center of the window. When the window center moves to a new pixel, the color pattern within the window is changed and the corresponding interpolation algorithm has to be changed accordingly. Figure 4 shows four interpolated pixels from four different windows with four different color patterns. Depending upon which of the four pixels the window is centered, four different interpolation algorithms are consequently required.

3. PROPOSED IMAGE ACQUISITION CHAIN

Pixel-centered interpolation algorithm flow requires four different algorithms for the different Bayer pattern orders.

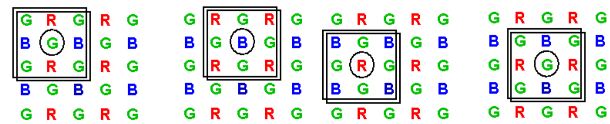

Figure 4. Four interpolated pixels from 4 different windows with 4 different patterns

These different algorithms increase the amount of hardware required to carry out the instructions and are major drain on the computational resources. Moreover, the interpolation is performed in a serial fashion, which presents a significant limitation to the processing rate. The traditional image acquisition chain also generates excessive amount of intermediate data that is essentially discarded in the final YCbCr decimation stage. All these extra processing steps and data are time consuming and consume significant resources. The proposed chain simplifies the process and uses a so-called vertex-centered interpolation algorithm to allow 4 pixels to be interpolated in parallel in a 3x3 window. The proposed chain conveniently combines the interpolation and color conversion steps together to directly generate YCbCr 4:2:0 or YCbCr 4:2:2 data to conserve memory in the data flow path. The proposed chain also uses a pseudomedian filtering technique in the Cb and Cr channels to remove color artifacts and performs color correction in YCbCr space to improve image quality. Figure 5 and Figure 6 show the proposed image acquisition chain and its data flow.

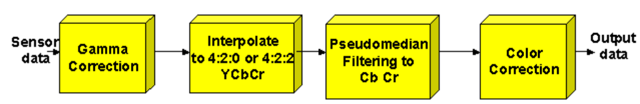

Figure 5. Proposed image acquisition chain

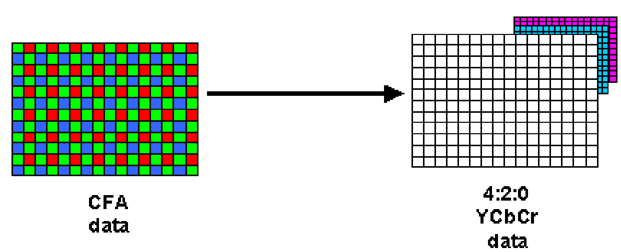

Figure 6. Data flow of the proposed image acquisition chain

3.1. Vertex-Centered Interpolation

Instead of interpolating one center pixel at a time for a 3x3 window, the proposed algorithm interpolates the vertices of the center pixel. This algorithm would generate in parallel four interpolated values for the 3x3 window. Figure 7 shows the four interpolated vertices P5a∼P5d in the 3x3 window.

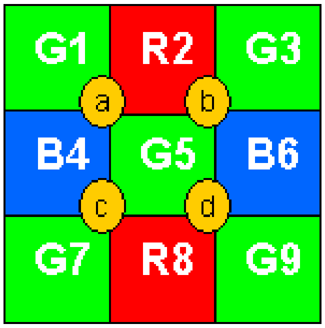

Figure 7. A typical RGB color filter array with vertex-centered interpolation

$$\begin{aligned} P5a &= f\,(R2, G1, G5, B4) \\ P5b &= f\,(R2, G3, G5, B6) \\ P5c &= f\,(R8, G7, G5, B4) \\ P5d &= f\,(R8, G9, G5, B6) \end{aligned} \qquad (3)$$

Consider P5a as an example, its interpolated RGB values are given by (4).

$$R5a = R2; \quad G5a = (G1 + G5)/2; \quad B5a = B4 \qquad (4)$$

Vertex-centered interpolation is more efficient than the pixel-centered approach because: 1) multiple pixel color data are generated in parallel, and 2) it allows the same color pattern to be maintained as the window is moved across the image frame and thus accommodates the use of a single algorithm throughout the entire image.

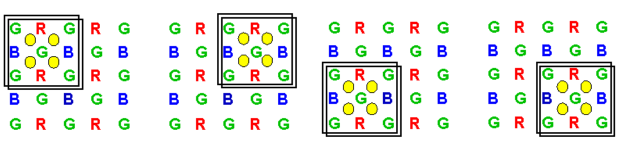

Figure 8. Four interpolated pixels from windows with the same pattern

Image compression engine in a digital camera system usually requires YCbCr 4:2:0 or YCbCr 4:2:2 data. The vertex-centered interpolation method is well suited for interpolation directly from the raw Bayer pattern data to the decimated YCbCr data.

The formula for conversion from the RGB space to the YCbCr space is given by [5,6]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ -.169 & -.331 & .5 \\ .5 & -.419 & -.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \qquad (5)$$

Implementation of the vertex-centered interpolation algorithm directly to either YCbCr 4:2:0 or 4:2:2 space is shown in (6) and Figure 9.

$$\begin{aligned} Y5a &= 0.299R2 + 0.587(G1 + G5)/2 + 0.114B4 \\ Y5b &= 0.299R2 + 0.587(G3 + G5)/2 + 0.114B6 \\ Y5c &= 0.299R8 + 0.587(G7 + G5)/2 + 0.114B4 \\ Y5d &= 0.299R8 + 0.587(G9 + G5)/2 + 0.114B6 \end{aligned}$$

Cb and Cr values in 4:2:0 YCbCr

$$\begin{aligned} Cb &= -0.169(R2 + R8)/2 + 0.5(B4 + B6)/2 - 0.1655G5 \\ &\quad -0.1655(G1 + G3 + G7 + G9)/4 + 128 \\ Cr &= 0.5(R2 + R8)/2 - 0.081(B4 + B6)/2 - 0.21G5 \\ &\quad -0.21(G1 + G3 + G7 + G9)/4 + 128 \end{aligned}$$

Cb and Cr values in 4:2:2 YCbCr

$$\begin{aligned} Cb1 &= -0.169R2 + 0.5(B4 + B6)/2 \\ &\quad -0.331(G1 + G3 + G5)/3 + 128 \\ Cr1 &= 0.5R2 - 0.081(B4 + B6)/2 \\ &\quad -0.419(G1 + G3 + G5)/3 + 128 \\ Cb2 &= -0.169R8 + 0.5(B4 + B6)/2 \\ &\quad -0.331(G7 + G9 + G5)/3 + 128 \\ Cr2 &= 0.5R8 - 0.081(B4 + B6)/2 \\ &\quad -0.419(G7 + G9 + G5)/3 + 128 \end{aligned} \qquad (6)$$

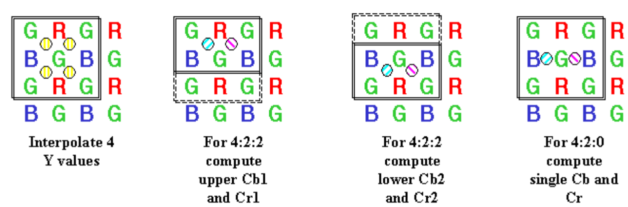

Figure 9. Direct YCbCr 4:2:0 and 4:2:2 output signals from vertex-centered interpolation

3.2. Pseudomedian filtering

Median filtering is an effective means for image noise removal. It performs especially well at reducing color noise. In the proposed image acquisition system, a median-like operation is implemented for the Cb and Cr color signals to remove difficult color artifacts in the interpolated images. For true median operation, a 3x3 window is typically used to determine the median value of the 9 pixels and the median value is then used to replace the value of the center pixel. Unfortunately, this type of operation is computational expensive. Instead, pseudomedian filter [7] is used in the proposed system to simplify the noise removal process. It filters the Cb and Cr signals using following operation.

Consider a 3x3 neighborhood of Cb1∼Cb9,

Cb1	Cb2	Cb3
Cb4	Cb5	Cb6
Cb7	Cb8	Cb9

the median of Cb1, Cb2 and Cb3 is first computed. Likewise, the median of Cb4, Cb5 and Cb6, and the median of Cb7, Cb8 and Cb9 are calculated separately. Finally the median of these three median values replaces the value Cb5 in the output signal. The same procedure is applied to Cr signal as well. Pseudomedian filtering provides an efficient form of median-like filtering with negligible loss of image quality at much-reduced hardware requirements.

4. EXPERIMENTAL RESULTS

Figure 10 shows a color image processed through the proposed image acquisition chain. Figure 11 shows the same image through a traditional image acquisition chain. The figures clearly illustrate the performance of the proposed system and the effectiveness of the pseudomedian filter in removing color artifacts especially in the highlighted water spray area. Despite the fact that the proposed image acquisition system has lower complexity, it achieves a better image quality than the traditional approach.

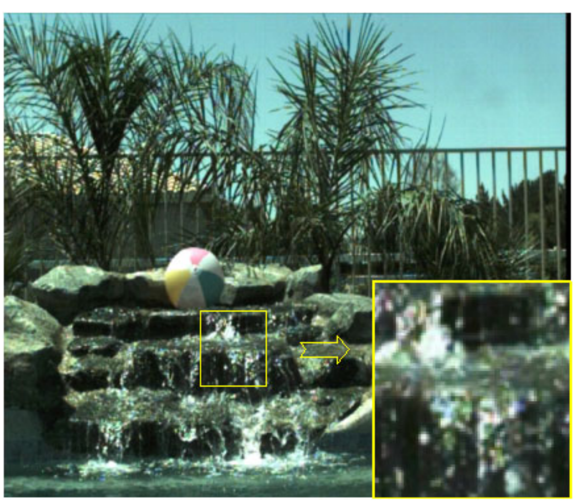

Figure 10. Interpolated image with the proposed image acquisition chain

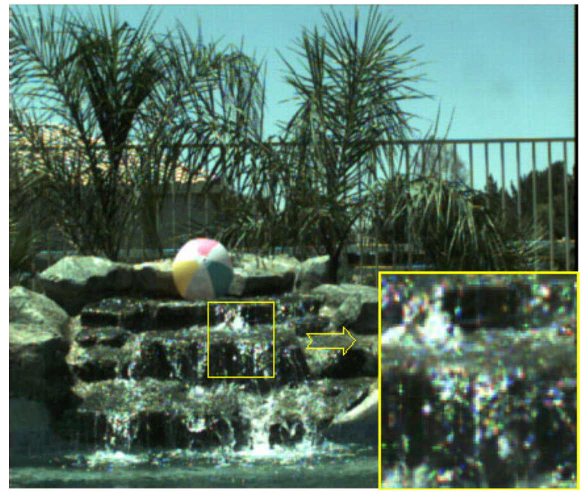

Figure 11. Interpolated image with a traditional image acquisition chain

5. SUMMARY

This paper presented an efficient color image acquisition system that uses vertex-centered interpolation directly to the YCbCr space with all subsequent processing steps performed in the YCbCr space. The proposed approach allows simpler hardware implementation with lower data memory requirement, lower gate count, and lower power and therefore is well suited for wireless handheld devices.

6. REFERENCES

[1] Y. H. Jung et al., “Design of real-time image enhancement preprocessor for CMOS image sensor,” *IEEE Trans. on Consumer Electronics*, vol. 46, no. 1, pp. 68–75, 2000.

[2] B. E. Bayer, *Color imaging array*, US patent 3,971,065, Patent and Trademark Office, Washington, D.C., 1976.

[3] G. Sharma, M. J. Vrhel, and H. J. Trussel, “Color imaging for multimedia,” *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1088–1108, 1998.

[4] T. Sakamoto et al., “Software pixel interpolation for digital still cameras suitable for a 32- bit MCU,” *IEEE Trans. on Consumer Electronics*, vol. 44, no. 4, pp.1342–1352, 1998.

[5] C. A. Poynton, “Component video color coding,” *A Technical Introduction to Digital Video*, John Wiley & Sons, Inc., pp. 171–177, 1996.

[6] C. A. Poynton, “Merging computing with studio video: Converting between R’G’B’ and 4:2:2,” *http://www.poynton.com/Poynton-articles.html*.

[7] W. K. Pratt, T. J. Cooper, and I. Kabir, “Pseudomedian filter,” *Architectures and Algorithms for Digital Image Processing II, Proc SPIE*, vol. 534, F. J. Corbett, Ed., pp. 34–43, 1985.