PERFORMANCE ANALYSIS OF LOSSLESS COMPRESSION SCHEMES FOR BAYER PATTERN COLOR IMAGE VIDEO SEQUENCES

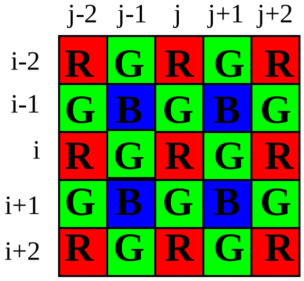
G. Mohanbabu[1] and Dr. P. Renuga[2]
[1]*Lecturer, Department of Electronics and Communication Engineering, PSNACET, Dindigul*
[1] kgbabu73@yahoo.com
[2]*Assistant Professor, Department of Electrical and Electronics Engineering, TCE, Madurai*
[2]preee@tce.edu

Abstract: Most consumer digital color cameras are equipped with a single chip. Such cameras capture only one color component per pixel (e.g., Bayer pattern) instead of an RGB triple. Conventionally, missing color components at each pixel are interpolated from its neighboring pixels, so that full color images are constructed. This process is typically referred to as demosaicing. After demosaicing, the full resolution RGB video frames are converted into YUV color space. U and V are then typically subsampled by a factor of four and the resulting video data in the 4:2:0 format become the input for the video encoder. In this letter, we look into the weakness of the conventional scheme and propose a novel solution for compressing Bayerpattern video data. The novelty of our work lies largely in the chroma subsampling. We properly choose the locations to calculate the chroma pixels U and V according to the positions of B and R pixels in the Bayer pattern and this leads to higher quality of the reconstructed images. In our experiments, we have observed an improvement in composite peak-signal-to-noise ratio performance of up to 1.5 dB at the same encoding rate. Based on this highly efficient approach, we propose also a low-complexity method which saves almost half of the computation at the expense of a small loss in coding efficiency.

Keywords :- Bayer-pattern video compression, chroma subsampling, color space conversion, H.264/AVC.

1. Introduction

In recent years, digital cameras for still images and movies became popular. There are many obvious advantages to digital images comparing to classical film based cameras, yet there are limitations as well. For example, the spatial resolution is limited due to the physical structure of the sensors. “Superresolution” beyond the sensors resolution can be achieved by considering a sequence of images. To reduce cost, most digital cameras use a single image sensor to capture color images. A Bayer color filter array (CFA) , as shown in Fig. 1, is usually coated over the sensor in these cameras to record only one of the three color components at each pixel location. The resultant image is referred to as a CFA image in this paper hereafter. In general, a CFA image is first interpolated via a demosaicing process to form a full color image before being compressed for storage. Fig. 2(a) shows the workflow of this imaging chain.



	j-2	j-1	j	j+1	j+2
i-2	R	G	R	G	R
i-1	G	B	G	B	G
i	R	G	R	G	R
i+1	G	B	G	B	G
i+2	R	G	R	G	R

Fig.1. Bayer pattern having a red sample as its centre

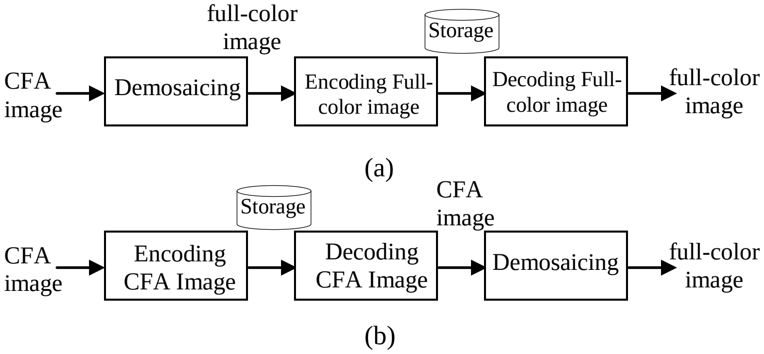


Fig.2. Single-sensor camera imaging chain:
(a) the demosaicing- first scheme; (b) the compression- first

There are two categories of CFA image compression schemes: lossy and lossless. Lossy schemes compress a CFA image by discarding its visually redundant information. These schemes usually yield a higher compression ratio as compared with the lossless schemes. There are Some different lossy compression techniques such as discrete cosine transform, vector quantization, subband coding with symmetric short kernel filters , transform followed by JPEG or JPEG 2000 , and low-pass filtering followed by JPEG-LS or JPEG 2000 (lossless mode) are used to reduce data redundancy.

The red, green, and blue pixels in the Bayer pattern are separated into three arrays and then an MPEG-2 like video coder is used for compression. This method should have a limited coding efficiency for P-frames because severe aliasing is generally contained in Bayer pattern. To alleviate aliasing, it is proposed in [3], [4] that Bayerpattern videos are compressed using an H.264 video coder and the motion compensation is adapted to the Bayer pattern. However, these two schemes are both confined to the RGB domain and, due partly to this, outperform the conventional method only at relatively high-bit rates.

In this letter, we propose a novel scheme using adjusted chroma subsampling for compressing Bayer-pattern video sequences. We first transform the Bayer-pattern images from the RGB domain to the YUV domain . In our approach, however, the chroma pixels U and V are calculated at different positions according to the B and R components of the Bayer pattern. Then we forward the image data in the YUV domain to an H.264 video coder for compression. After we have the YUV data decoded, we transform them back to the RGB domain so that the original Bayer-pattern image is reconstructed. Our proposed scheme proves significantly more efficient than the conventional one over the entire bit rate range. Moreover, this method can be extended to a low-complexity version. The computational complexity is reduced by almost 50% at the expense of a small drop in rate-distortion performance.In Section II, we describe the conventional approach and look into its drawback. In Section III, we present our proposed method in detail. The experimental results, including rate-distortion curves and encoding time, are presented in Section IV. Finally, Section V concludes the letter.

2. Conventional Approach

The Conventional Bayer-pattern video compression begins with demosaicing or color interpolation, as illustrated in Fig. 2(a). Then full color RGB images are converted to the YUV domain. After this, chroma subsampling is applied to the components U and V by a factor of two both horizontally and vertically, as shown in Fig. 1, so that the YUV data are available in the standard format 4:2:0. Now the YUV data are compressed using an H.264 video coder. The decoder reconstructs the YUV data in the format 4:2:0 and then interpolates the components U and V to their full size. Finally, the YUV images are converted back to RGB images.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.813 & 0.391 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2)$$

Simplicity is the most significant advantage of the conventional approach. There are existing or standard methods for all the main steps, including demosaicing, color space transform, chroma subsampling, and H.264/AVC video coding. However, such a simple combination of different techniques limits the coding efficiency. particular drawback lies in the chroma sample positions for chroma subsampling.

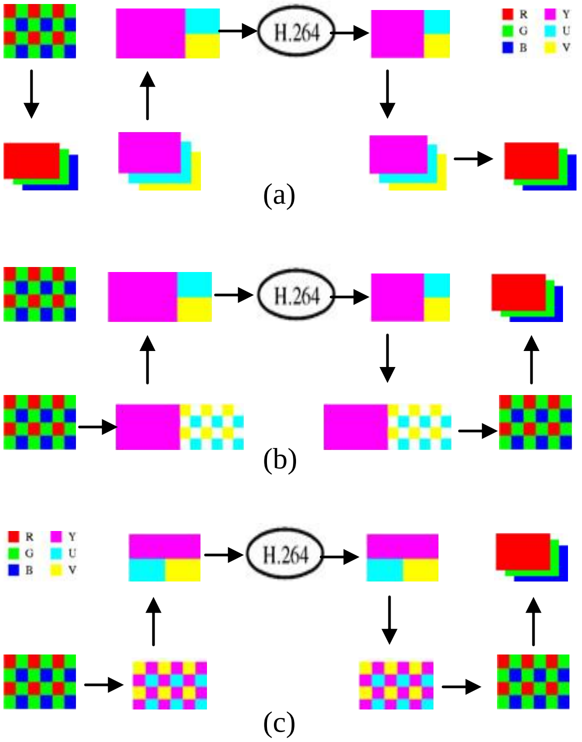
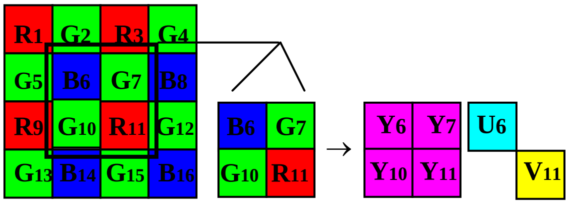


Fig. 2. Comparison of the conventional method, the proposed methods B-4:2:0 and B-4:2:2. (a) Conventional method 4:2:0. (b) Proposed method B-4:2:0. (c) Proposed method B-4:2:2.

3. Proposed Approach

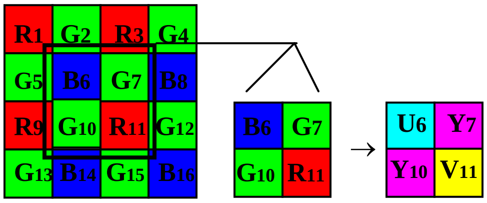
A. Proposed Method B-4:2:0

The color space transform of our proposed method B-4:2:0 is illustrated in Fig. 3(a). We calculate Y pixels at all the locations in the Bayer-pattern image, no matter if it is an R pixel, a G pixel or a B pixel. When it comes to chroma pixels U and V, the positions to calculate them are carefully chosen. V values are calculated at the positions of R pixels, and U values at B pixels. We can justify this if we look into the equations in (2). During the inverse color space transform, only Y and V are needed to reconstruct R pixels. This means, V is more important than U at positions of R pixels, that's why we calculate V values. For the same reason, U values are calculated where B pixels exist. The different positions selected for U and V pixels in our proposed scheme are more reasonable for the reconstruction of R and B pixels than the standardized chroma sample positions in the conventional method. The standard chroma subsampling takes U and V always at the same location without taking into account the different positions of R and B pixels in the Bayer pattern, thus it cannot be optimum for Bayer-pattern image and video compression. This is the fundamental reason, why our proposed scheme B-4:2:0 can have a better rate-distortion performance than the conventional one. Color space transform requires all three color components R, G, and B for every pixel, but Bayer-pattern images have only one, either R or G or B, at each position, so missing components for every pixel need to be interpolated from adjacent pixels. This process is just the so-called demosaicing or color interpolation. Now we base our discussion on demosaicing using bilinear interpolation and the equations are listed in Fig. 3(a). More advanced interpolation techniques can also be introduced into our system. In the section for experimental results, not only the results for bilinear interpolation are shown but also those for the edge-directed interpolation with secondorder gradients as correction terms, also known as Laplacian interpolation.The luma and chroma pixels are then written into a standard YUV file and forwarded to the H.264 video coder for compression. As shown in Fig. 3(a), for every 2 × 2 image block, all the four Y pixels are calculated and only one U pixel and one V pixel are calculated. Although the chroma value location differs from the nominal locations shown in Fig. 1, the resulting YUV images can be compressed using the 4:2:0 mode of H.264. At the decoder, YUV image data are first reconstructed. Of course, we also need to interpolate the missing U or V pixels in order to transform the data back to RGB Bayer-pattern images. Finally, we apply demosaicing by bilinear interpolation on the Bayer-pattern images to obtain RGB full color images.



for Y7 : G = G7 B = (B6+B8)/2 R = (R3+R11) / 2
for Y10 : G = G10 B = (B6+B14) / 2 R = (R9+R11) / 2
for U6 and Y6: G = (G2+G5+G7+G10) / 5 B=B6 R= (R1+R3+R(+R11) / 4
for V11 and Y11 G = (G7+G10+G12+G15) / 4 B = (B6+B8+B14+B16)/4 R = R11

(a)



for Y7 : G = G7 B = (B6+B8) / 2 R = (R3+R11) / 2
for Y10 : G = G10 B = (B6+B14) / 2 R = (R9+R11) / 2
for U6 : G = (G2+G5+G7+G10) / 4 B=B6 R= (R1+R3+R(+R11) / 4
for V11 : G = (G7+G10+G12+G15)/4 B = (B6+B8+B14+B16) / 4 R = R11

(b)

Fig. 3. Color space transform from the RGB domain to the YUV domain.
(a) Proposed method B-4:2:0. (b) Proposed method B-4:2:2.

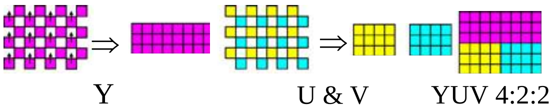


Fig. 4. Structure conversion for YUV data in the proposed method B-4:2:2

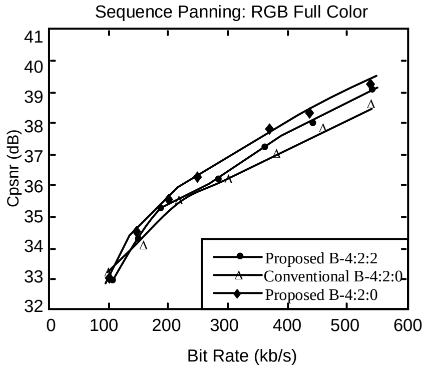
B. Proposed Method B-4:2:2

Based on the proposed method B-4:2:0, we propose the scheme B-4:2:2 mainly to further reduce the computational complexity. Instead of calculating all the luma pixels and compressing them, we keep only the Y pixels at the positions of G pixels and get rid of those at R and B pixels. Now that the Y pixels to compress are halved, the encoding time can be reduced almost by 50%. The color space transform of this method based on demosaicing using bilinear interpolation is illustrated in Fig. 3(b). The Y pixels we compress form a quincunx pattern as G pixels in the Bayer pattern. Therefore, it is necessary to add in a step of structure conversion [5], which converts this quincunx pattern to a rectangular one, in order that the H.264 video coder works correctly. Such structure conversion is shown in Fig. 4. We move Y pixels in the even rows one unit upward and then push the resulting complete rows of Y pixels together to a rectangular array. The chroma pixels are also pressed together. After this structure conversion, the YUV data match the standard format 4:2:2 and are ready to be compressed by an H.264 video coder using the 4:2:2 mode. The decoder needs to convert the rectangular pattern of the reconstructed Y pixels back to the quincunx pattern. Then missing Y pixels are interpolated before RGB values in the

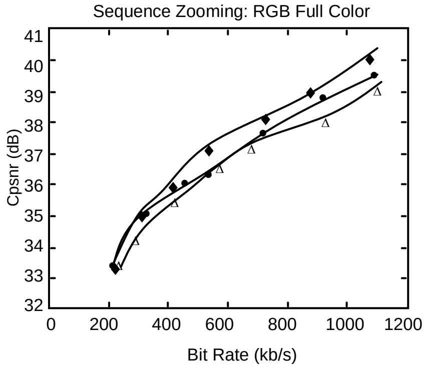
Bayer pattern can be calculated. Finally, full color RGB images are generated by demosaicing the Bayer-pattern images. One thing worth mentioning is that the interpolation scheme we use for Y pixels is of the same order as the one for G pixels in the demosaicing of RGB images.

4. Experimental Results

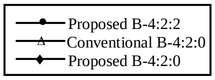
For the simulation, we capture in our laboratory three Bayerpattern video sequences which represent three different motion modes, *Panning*, *Zooming*, and *Moving Object*, over a static background. The video frames are captured in common intermediate format (CIF) 352 × 288 and at a rate of 30 frames/s. Screenshots of selected frames of the three test videos are shown in Fig. 5. The H.264/AVC reference software JM 12.2 is used for video compression in our simulation. The group of picture (GOP) size is 40 and the GOP structure is set to I-P-P-P. The YUV format is set to 4:2:0 or 4:2:2, according to different simulations. As for other parameters in the configuration file of the JM reference software, we keep their default values. Rate-distortion curves for different methods and different test sequences are plotted in Fig. 6 for the case of bilinear interpolation. The original Bayer-pattern images and the reconstructed ones are interpolated to full color RGB images using bilinear interpolation and the composite peak-signal-to noise ratio (CPSNR) is calculated between them. Finally, the CPSNR for all the images in a sequence is averaged. Equations (3) and (4) are used to calculate CPSNR of an image.

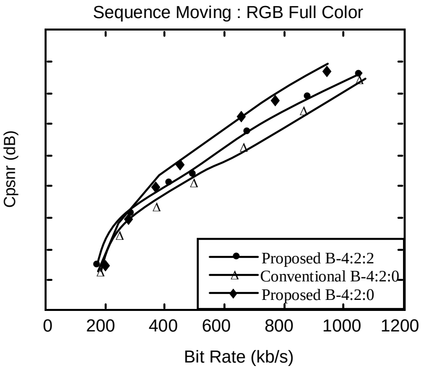


(a)



(b)





(c)

$$\text{CPSNR} = 10 \text{ Log}_{10} \frac{255^2}{\text{MSE}}$$

$$\text{MSE} = \frac{1}{3MN} \sum_{k=1}^{3} \sum_{i=1}^{N} \sum_{j=1}^{B} [\, I\,(I,j,k) - I'\,(i,j,k)]^2$$

5. Conclusion

The rate-distortion performance of our proposed method B-4:2:2, however, could be further improved. As discussed, the structure conversion in the proposed method B-4:2:2 destroys the regular arrangement of Y pixels, resulting in a lower accuracy of motion compensated prediction in video coding. If we modify the motion estimation and compensation of the H.264 video coder, the coding efficiency of the method B-4:2:2 can become higher.

References

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.

[2] F. Gastaldi, C. C. Koh, M. Carli, A. Neri, and S. K. Mitra, "Compression of videos captured via bayer patterned color filter arrays," in *Proc. 13th Eur. Signal Process. Conf.*, Antalya, Turkey, Sep. 2005.

[3] C. Doutre and P. Nasiopoulos, "An efficient compression scheme for colour filter array video sequences," in *Proc. IEEE 8th Workshop Multimedia Signal Process.*, Victoria, BC, Oct. 2006, pp. 166–169.

[4] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis, "H.264-based compression of bayer pattern video sequences," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 18, no. 6, pp. 725–734, Jun. 2008.

[5] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1448–1456, Nov. 2003.

[6] J. Mukherjee, M. K. Lang, and S. K. Mitra, "Color demosaicing in YUV color space," in *Proc. Int. Assoc. Sci. Tech. Dev. Conf.*, Malaga, Spain, Sep. 2002, pp. 96–101.

[7] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep.1999.

[8] S. Pei and I. K. Tam, "Effective color interpolation in CCD color filter array using signal correlation," in *Proc. IEEE Int. Conf. Image Process.*,vol. 3. Sep. 2000, pp. 10–13.

[9] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 4. May 2002, pp. 3333–3336.

[10] T. Kuno and H. Sugiura, "New interpolation method using discriminated color correlation for digital still cameras," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 259–267, Feb. 1999.

[11] C. Weerasinghe, I. Kharitonenko, and P. Ogunbona, "Method of color interpolation in a single sensor color camera using green channel separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 4. May 2002, pp. 3233–3236.

[12] S. Ferradans, M. Bertalmio, and V. Caselles, "Geometry-based demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 665–670, Mar. 2009.

[13] J. E. Adams and J. F. Hamilton, "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. Soc. Photo-Optical Instrum. Eng.*, vol. 3028. Feb. 1997, pp. 117–125.

[14] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.

[15] K. Jack, "Color spaces," in *Video Demystified*, 3rd ed. Eagle Rock, VA: LLH Technology Publishing, 2001, pp. 18–19.