A Fast Demosaicking Method Directly Producing YCbCr 4:2:0 Output

Colin Doutre, *Student Member*, IEEE, Panos Nasiopoulos, *Member*, IEEE,
and Konstantinos N. Plataniotis, *Senior Member*, IEEE

Abstract — *Consumer digital cameras usually use a single light sensor together with a color filter array (CFA) for capturing color images. This results in a mosaic image being captured, where only one color sample (red, green or blue) is obtained at each pixel location. Demosaicking is the process of interpolating the two missing color samples at each location to generate a full color image. Virtually all current demosaicking methods produce an RGB output image. If the image is to be compressed afterwards, it will typically be converted from RGB to YCbCr 4:2:0 format. In this paper we propose a new demosaicking method that directly produces an YCbCr 4:2:0 image, so the output can be directly compressed. Simulation results show that the proposed method provides higher image quality than fast RGB based demosaicking methods and has lower computational complexity*[1].

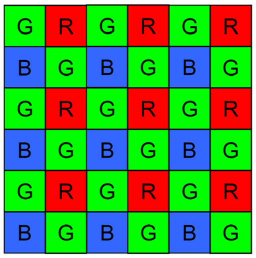
Index Terms — demosaicking, color filter array, YCbCr color space, digital cameras.

I. INTRODUCTION

Most consumer digital cameras capture color information with a single light sensor and a color filter array (CFA). Instead of capturing three color samples (typically red, green and blue) at each pixel location, these cameras capture a so called ‘mosaic’ image, where only one color is sampled at each location. The two missing colors must be interpolated from the surrounding samples in a process called demosaicking.

There are many different color filter array designs. The Bayer pattern [1] (Fig. 1) is the most commonly used. It captures samples in groups of four, each group containing two green samples and one red and blue sample. More green samples are captured because the human visual system is more sensitive to the green portion of the spectrum.

Virtually all demosaicking schemes reported in the literature produce RGB output images [2]. If the image is compressed afterwards it will typically be converted from RGB to YCbCr color space. The equations used for converting from RGB to YCbCr in the JPEG JFIF format [3] are:



G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Fig. 1: Bayer Pattern CFA

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ Cb &= -0.1687 \cdot R - 0.3313 \cdot G + 0.5 \cdot B \\ Cr &= 0.5 \cdot R - 0.4187 \cdot G - 0.0813 \cdot B \end{aligned} \tag{1}$$

The reverse equations for converting from YCbCr to RGB are:

$$\begin{aligned} R &= Y + 1.402 \cdot Cr \\ G &= Y - 0.34414 \cdot Cb - 0.71414 \cdot Cr \\ B &= Y + 1.772 \cdot Cb \end{aligned} \tag{2}$$

The YCbCr color space conversion strongly decorrelates the color channels, so they can be coded independently without loss of efficiency. Furthermore, since the human visual system is less sensitive to chrominance than luminance, the Cb and Cr channels can be down-sampled by a factor of two in both the horizontal and vertical directions without loss of perceived image quality. This downsampling produces YCbCr 4:2:0 sampling, where there are four Y samples for every Cb and Cr sample. In order to reduce aliasing in the downsampling process, the chrominance channels must first be low-pass filtered.

YCbCr 4:2:0 format is by far the most common color representation used in compressed images and video. It is frequently used in JPEG and JPEG2000 files, as well as the MPEG and H.26x families of video compression standards. It should be noted that the YCbCr color space is sometimes referred to as YUV, and there are minor variations of the equations used for converting between RGB and luminance and chrominance.

The green channel is the dominant component when calculating luminance and the red and blue channels are the main contributors to the Cr and Cb channels, respectively. Advanced demosaicking methods put a lot of computational effort into accurately reconstructing the high frequency

---

[1] This work was supported by the Natural Sciences and Engineering Research Council of Canada.

C. Doutre and P. Nasiopoulos are with the University of British Columbia, Vancouver, BC, Canada (e-mails: colind@ece.ubc.ca, panos@ece.ubc.ca).

K. N. Plataniotis is with The Edward S. Rogers Sr. Department of ECE, University of Toronto, Toronto, ON, Canada.

Contributed Paper
Manuscript received October 16, 2006

information in the red and blue channels. This task is challenging because red and blue are more sparsely sampled in the Bayer pattern than the green. It is also somewhat unnecessary because the high-frequency chrominance information will be lost in converting from RGB to YCbCr 4:2:0 format.

To summarize, these steps are carried out in the conventional approach of first demosaicking, and then converting to YCbCr 4:2:0 format for compression:

1) Perform demosaicking, generating full red, green and blue color channels
2) Conversion from RGB to YCbCr color space with equation (1)
3) Low-pass filtering and downsampling the Cb and Cr channels to give 4:2:0 sampling

All of these steps can be computationally expensive. Steps 2 and 3 often involve many floating point calculations, which are expensive to implement on the low cost embedded processors found in digital cameras. In this paper we propose a demosaicking method that directly produces YCbCr 4:2:0 output, thus reducing the computational and memory requirements of the system performing demosaicking.

There is one previously published work that addresses the issue of demosaicking directly to YCbCr 4:2:0 [4]. Four related methods are proposed in [4], here we described only the most competitive method, YUV through green interpolation with median filtering post-processing (YUVGM). That method starts by filling the green channel with the method used for calculating the green samples in a previous demosaicking algorithm [5]. Referring to Fig. 2, which shows the unit cell of the Bayer pattern, the YUVGM method produces four luminance samples (Y1-Y4) and one sample of each chrominance channel (Cb1 and Cr1) for each cell. Let *Y(R,G,B)*, *Cb(R,G,B)* and *Cr(R,G,B)* denote functions for converting from RGB to YCbCr space given in (1). Then these equations are used for generating the four luminance samples and each chrominance sample for the 2x2 cell:

$$
\begin{aligned}
\mathrm{Y1} &= Y(\mathrm{R2},\mathrm{G1},\mathrm{B3})\\
\mathrm{Y2} &= Y(\mathrm{R2},\mathrm{G2},\mathrm{B3})\\
\mathrm{Y3} &= Y(\mathrm{R2},\mathrm{G3},\mathrm{B3})\\
\mathrm{Y4} &= Y(\mathrm{R2},\mathrm{G4},\mathrm{B3})\\
\mathrm{G}_{\mathrm{avg}} &= (\mathrm{G1}+\mathrm{G2}+\mathrm{G3}+\mathrm{G4})/4\\
\mathrm{Cb1} &= Cb(\mathrm{R2},\mathrm{G}_{\mathrm{avg}},\mathrm{B3})\\
\mathrm{Cr1} &= Cr(\mathrm{R2},\mathrm{G}_{\mathrm{avg}},\mathrm{B3})
\end{aligned}
\tag{3}
$$

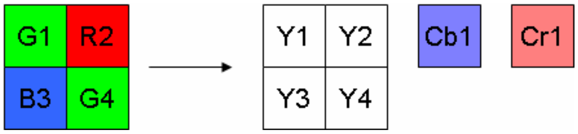


Fig. 2: Conversion from Bayer unit cell to YCbCr 4:2:0 samples

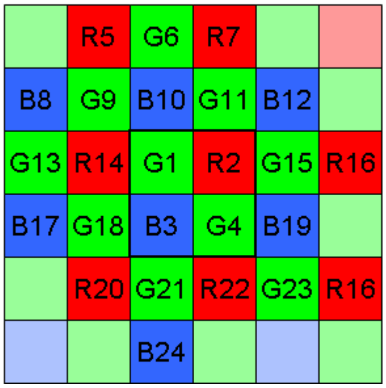


Fig. 3: Neighborhood of pixels used for generating the luminance and chrominance samples in the cell containing positions 1-4

After the samples have been calculated with (3), median filtering is performed on the Cb and Cr channels, to remove some color artifacts. Essentially, the YUVGM method in [4] is using zero-order hold interpolation on the red and blue channels. This produces poor results around edges and results in false colors appearing in the demosaicked image. The median filtering operation is an ad-hoc and computationally expensive method of removing some of the color artifacts.

Our demosaicking method generates a full green channel, and low-pass filtered, down-sampled red and blue channels. The green channel contains the high-frequency detail needed to construct an accurate luminance channel, and the low-pass red and blue samples allow us to directly compute down-sampled chrominance components.

The rest of this paper is organized as follows. The proposed method is described in Section II. Simulation results comparing the proposed method against other fast demosaicking algorithms are presented in Section III. A complexity analysis of our method is given in Section IV. Conclusions are made in Section IV.

II. PROPOSED METHOD

The Bayer pattern consists of cells of size 2x2 pixels, each cell containing two green samples and one red and blue sample. To produce YCbCr 4:2:0 output, four luminance samples and one Cb and Cr samples must be generated for each cell. Figure 3 shows a 2x2 cell (locations 1-4) and the surrounding Bayer pattern samples that will be used to calculate the luminance and chrominance samples in the cell. In the following section we describe how our algorithm generates chrominance samples at location 1 (denoted as Cb1, Cr1), and luminance samples at locations 1-4 (denoted Y1, Y2, Y3, Y4). The steps described are repeated on every 2x2 cell to generate the entire YCbCr 4:2:0 image. The location numbering of Figure 3 will be used throughout the rest of the paper.

A. Genrating a Full Green Channel

Since a full luminance channel is needed, and green is the dominant component is determining luminance, our method begins by filling the green channel. A complete green channel allows us to estimate a high-quality luminance channel.

Many existing demosaicking methods start by generating a complete green channel. We base our method for calculating the missing green samples on the method by Hamilton and Adams [5], which is a popular low complexity demosaicking algorithm.

The idea behind the method for filling the green channel is to calculate horizontal and vertical gradients at the current location, and interpolate along the direction that contains the lower gradient. This results in interpolation being performed along edges rather than across edges. If the gradients are similar in magnitude, interpolation is done using samples in both directions. There are two cases that need to be considered when calculating the missing green samples; generating a green at a red location (R2) or blue location (B3).

At red location (R2), the gradients are calculated as:

$$\begin{aligned} \mathrm{DH} &= \left|\mathrm{R14}+\mathrm{R16}-2\cdot\mathrm{R2}\right|+\left|\mathrm{G1}-\mathrm{G15}\right|. \\ \mathrm{DV} &= \left|\mathrm{R7}+\mathrm{R22}-2\cdot\mathrm{R2}\right|+\left|\mathrm{G11}-\mathrm{G4}\right| \end{aligned} \tag{4}$$

The missing green sample is calculated as follows:

$$\begin{aligned} &\text{if } (\mathrm{DH}+\mathrm{T}<\mathrm{DV}) \\ &\quad \mathrm{G2}=\frac{\mathrm{G1}+\mathrm{G15}}{2}+\frac{2\cdot\mathrm{R2}-\mathrm{R14}-\mathrm{R16}}{4} \\ &\text{else if } (\mathrm{DV}+\mathrm{T}<\mathrm{DH}) \\ &\quad \mathrm{G2}=\frac{\mathrm{G11}+\mathrm{G4}}{2}+\frac{2\cdot\mathrm{R2}-\mathrm{R7}-\mathrm{R22}}{4} \\ &\text{else} \\ &\quad \mathrm{G2}=\frac{\mathrm{G1}+\mathrm{G11}+\mathrm{G15}+\mathrm{G4}}{4}+\frac{4\cdot\mathrm{R2}-\mathrm{R7}-\mathrm{R14}-\mathrm{R16}-\mathrm{R22}}{8} \end{aligned} \tag{5}$$

The second term in each sum in (5) is a second order correction term that enhances the directional bilinear interpolation as described in [5].

The value T is a threshold that ensures the gradients are sufficiently different for interpolation to happen in one direction. If the difference between the gradients is less then T (the last case in equation (5)) the interpolation uses samples in both directions. A threshold is not used in the method by Hamilton and Adams [5]. If the threshold is set to zero, the method for filling the green channel would be identical to their method. Experimentally, we found a threshold of 35 to provide good results for a wide range of images.

At the blue location (B3), the gradients and missing green sample are calculated analogously to the red location using the following equations.

$$\begin{aligned} \mathrm{DH} &= \left|\mathrm{B17}+\mathrm{B19}-2\cdot\mathrm{B3}\right|+\left|\mathrm{G18}-\mathrm{G4}\right| \\ \mathrm{DV} &= \left|\mathrm{B10}+\mathrm{B24}-2\cdot\mathrm{B3}\right|+\left|\mathrm{G1}-\mathrm{G21}\right| \end{aligned} \tag{6}$$

$$\begin{aligned} &\text{if } (\mathrm{DH}+\mathrm{T}<\mathrm{DV}) \\ &\quad \mathrm{G3}=\frac{\mathrm{G18}+\mathrm{G4}}{2}+\frac{2\cdot\mathrm{B3}-\mathrm{B17}-\mathrm{B19}}{4} \\ &\text{else if } (\mathrm{DV}+\mathrm{T}<\mathrm{DH}) \\ &\quad \mathrm{G3}=\frac{\mathrm{G1}+\mathrm{G21}}{2}+\frac{2\cdot\mathrm{B3}-\mathrm{B10}-\mathrm{B24}}{4} \\ &\text{else} \\ &\quad \mathrm{G3}=\frac{\mathrm{G1}+\mathrm{G4}+\mathrm{G18}+\mathrm{G21}}{4}+\frac{4\cdot\mathrm{B3}-\mathrm{B17}-\mathrm{B19}-\mathrm{B10}-\mathrm{B24}}{8} \end{aligned} \tag{7}$$

B. Calculating Low-pass Red and Blue Samples

Since the conversion from RGB to YCbCr is a linear process, we can equivalently perform loss-pass filtering in the RGB domain rather than the YCbCr domain. We generate low-pass R and B samples located at the location of G1 in Fig. 3. Instead of performing interpolation on the red and blue channels themselves, we perform interpolation on the difference between green and red or blue. The R-G and B-G images are generally much smoother than the R and B channels, so they are more suitable for interpolation [6].

The following 2D filter is used on the R-G channel:

$$h_{lp}=\begin{bmatrix} 1/8 & 0 & 1/8 \\ 0 & 0 & 0 \\ 1/4 & 0 & 1/4 \\ 0 & 0 & 0 \\ 1/8 & 0 & 1/8 \end{bmatrix} \tag{8}$$

The filter provides low-pass filtering in both the horizontal and vertical directions. The resulting equation for calculating the red sample is

$$\begin{aligned} \mathrm{R1}_{lp} = \mathrm{G1} &+ \frac{\mathrm{R14}-\mathrm{G14}+\mathrm{R2}-\mathrm{G2}}{4} \\ &+ \frac{\mathrm{R5}-\mathrm{G5}+\mathrm{R7}-\mathrm{G7}+\mathrm{R20}-\mathrm{G20}+\mathrm{R22}-\mathrm{G22}}{8} \end{aligned} \tag{9}$$

The last two terms in the sum in (9) are the result of applying the low-pass filter of (8) to the R-G channel. Adding the G1 sample gives an estimate for R1.

The low-pass blue sample at location G1 is calculated equivalently as:

$$\begin{aligned} \mathrm{B1}_{lp} = \mathrm{G1} &+ \frac{\mathrm{B10}-\mathrm{G10}+\mathrm{B3}-\mathrm{G3}}{4} \\ &+ \frac{\mathrm{B8}-\mathrm{G8}+\mathrm{B12}-\mathrm{G12}+\mathrm{B17}-\mathrm{G17}+\mathrm{B19}-\mathrm{G19}}{8} \end{aligned} \tag{10}$$

C. Calculating Down-sampled Chrominance Channels

With low-pass, down-sampled red and blue channels, the chrominance channels can easily be calculated with the standard equations:.

$$\begin{aligned} \mathrm{Cb1} &= -0.1687\cdot\mathrm{R1}_{lp}-0.3313\cdot\mathrm{G1}+0.5\cdot\mathrm{B1}_{lp} \\ \mathrm{Cr1} &= 0.5\cdot\mathrm{R1}_{lp}-0.4187\cdot\mathrm{G1}-0.0813\cdot\mathrm{B1}_{lp} \end{aligned} \tag{11}$$

Ideally, the green channel should also be low-pass filtered for calculating the chrominance samples. However low-pass filtering the green channel would add additional complexity and we found that omitting the filtering has minimal impact on the quality of the chrominance channels.

D. Calculating the Full Luminance Channels

Once the down-sampled chrominance values have been calculated, the only task left is to generate the full luminance channel. We considered the task of generating luminance samples at locations 1 through 4 separately, because different samples are available at each location.

At the location of G1, we already have green, red and blue samples available, so the luminance sample can be directly calculated by:

$$Y1 = 0.299 \cdot R1_{lp} + 0.587 \cdot G1 + 0.114 \cdot B1_{lp} \tag{12}$$

Note that we are using low-pass red and blue samples, when ideally unfiltered samples should be used. However, since the green sample has not been filtered and green is the dominant component in calculating luminance, the value calculated with equation (12) is still a good estimate.

At the location of R2, we have red and green samples available. An assumption often made in demosaicking methods is that chrominance varies smoothly in natural images, so bilinear interpolation provides a good estimate for missing chrominance samples. Using this assumption, an estimate for the blue chrominance at R2 is found as:

$$Cb2 = \frac{Cb1 + Cb15}{2} \tag{13}$$

The Cb2 sample in (13) does not need to be calculated and stored, but the equation will be used for deriving an expression for Y2. We would like to calculate the luminance value at location 2 using R2, G2 and Cb2. This can be done by substituting the equation for B in (2) into the Y definition in (1).

$$Y2 = 0.299 \cdot R2 + 0.587 \cdot G2 + 0.114 \cdot (Y2 + 1.772 \cdot Cb2) \tag{14}$$

Further substituting the estimate for Cb2 given by (13) and solving for Y2 yields:

$$Y2 = 0.3375 \cdot R2 + 0.6625 \cdot G2 + 0.114 \cdot (Cb1 + Cb15) \tag{15}$$

At location 3, green and blue samples are available. Using an analogous method as described for Y2, only now with bilinear interpolation performed on Cr samples, Y3 is calculated as:

$$Y3 = 0.299 \cdot (Cr1 + Cr21) + 0.8374 \cdot G2 + 0.1626 \cdot B3 \tag{16}$$

At location 4, only a green sample is available. So here we use bilinear interpolation on both the Cb and Cr channels to calculate Y4. Substituting the R and B equations from (2) into the definition of luminance gives:

$$\begin{aligned} Y4 = {} & 0.299 \cdot (Y4 + 1.402 \cdot Cr4) + 0.587 \cdot G \\ & + 0.114 \cdot (Y4 + 1.772 \cdot Cr4) \end{aligned} \tag{17}$$

Using bilinear interpolation on the four surrounding samples to estimate Cb4 and Cr4, and solving for Y4 yields:

$$\begin{aligned} Y4 = {} & 0.1785 \cdot (Cr1 + Cr15 + Cr21 + Cr23) + G4 \\ & + 0.086 \cdot (Cb1 + Cb15 + Cb21 + Cb23) \end{aligned} \tag{18}$$

E. Summary of Complete Algorithm

Our complete demosaicking algorithm for producing YCbCr 4:2:0 output consists of the following steps. Each step must be carried out on every 2x2 cell before proceeding to the next step.

1) Fill the missing green samples with equations (4), (5), (6) and (7)
2) Using (9) and (10) find low-pass red and blue samples
3) Using (11) calculate the final Cb and Cr samples
4) Fill the luminance channel with (12), (15), (16) and (18)

III. EXPERIMENTAL RESULTS

The 24 RGB images from the Kodac set where used in our experiments. These images have been extensively used in demosaicking research. Thumbnails of the images are provided in Fig. 4. CFA images were obtained by sampling the RGB images with the Bayer pattern.

We compared our proposed method against the YUV method in [4], and some fast demosaicking methods that operate in RGB space. The RGB methods tested were bilinear interpolation, and the methods in [5] and [6].

Here objective quality is measured in the YCbCr 4:2:0 domain using the peak signal-to-noise ratio (PSNR). The reference images were obtained by converting from RGB to YCbCr space with (1), filtering the Cb and Cr channels with a 9-tap FIR low-pass filter and downsampling. The 9-tap low-pass filter closely approximates an ideal filter with cut-off 0.5 cycles/sample so the reference images contain very little aliasing in the down-sampled chrominance channels.

For the demosaicking methods that operate in RGB space, the following low-complexity filter was used for filtering the chrominance channels in the downsampling process:

$$h = [1/4 \quad 1/2 \quad 1/4] \tag{19}$$

Fig. 4: Test Images. Numbered 1-24, from top left to bottom right.

This filter provides a good tradeoff between complexity and limiting aliasing.

Table I shows the PSNR (in dB) obtained in the Y, Cb and Cr channels with each demosaicking method. In almost all cases, the proposed method gives higher PSNR than the other tested methods. On average the proposed method gives about 1 dB higher PSNR in each channel than the RGB based methods in [5],[6]. For all images, the proposed method gives far better performance (over 5 dB higher PSNR in luminance) than the only other YUV 4:2:0 based demosaicking method presented in [4].

Since PSNR is not always an accurate measure of perceived image quality we also provide a subjective quality comparison. Fig. 5 shows a cropped portion of image 6 and the result of applying each demosaicking method to the image. Close visual inspection of Fig. 5 shows that the proposed method produces fewer color artifacts and results in less blurring of edges than the other methods. Note how despite providing competitive PSNR, the method in [6] produces unpleasing zipper artifacts along some edges (Fig. 5d).

IV. COMPLEXITY ANALYSIS

A key advantage of the proposed method is the computational complexity saved by directly producing YCbCr 4:2:0 output rather than performing demosaicking in RGB space and then converting to YCbCr 4:2:0. Table II shows a summary of the number of operations per pixel in the CFA image needed for the proposed demosaicking method. Note there are some fractional values in Table II because many equations are not evaluated out at every pixel location. The number of operations performed when evaluating equation (5) is variable depending on the result of the comparisons; only the worst case complexity is shown in Table II.

For comparison, we present a complexity analysis of the method in [6], which is one of lowest complexity RGB based demosaicking algorithms reported in the literature. Table III

TABLE I: PSNR COMPARISON OF DIFFERENT DEMOSAICKING METHODS (DB)

Image Number	Bilinear			Method in [5]			Method in [6]			YUV Method in [4]			Proposed		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
1	29.58	35.03	35.97	35.99	40.03	41.36	35.97	39.37	39.33	30.90	37.85	37.83	37.43	42.68	42.36
2	36.31	41.73	39.88	41.68	45.01	42.04	41.28	44.25	40.62	37.04	42.46	38.55	42.21	45.48	41.53
3	37.45	42.76	44.34	43.44	45.46	46.36	43.40	45.36	44.24	38.17	41.85	42.16	44.32	45.94	46.01
4	36.82	43.45	41.18	41.71	45.69	42.36	42.33	46.07	41.40	37.65	43.31	39.13	42.52	47.56	41.90
5	29.60	37.22	36.74	37.49	40.87	42.02	37.09	40.73	39.06	30.85	36.61	37.24	38.36	41.59	42.18
6	31.04	36.17	38.93	37.47	41.30	42.77	37.38	40.63	40.77	32.53	38.29	39.24	38.87	43.35	43.76
7	36.59	42.88	42.51	43.60	45.60	46.23	42.42	44.67	43.65	36.88	39.87	40.51	43.92	45.40	45.89
8	27.08	32.05	30.25	34.76	38.44	39.29	33.11	35.73	35.19	28.55	33.49	34.03	35.85	40.35	40.03
9	35.67	41.19	40.88	42.75	45.22	46.34	41.55	44.06	43.66	36.57	40.84	42.05	43.81	45.38	46.65
10	35.57	41.53	42.03	42.63	45.61	46.30	42.36	45.21	44.23	36.78	41.64	42.24	43.58	46.17	46.45
11	32.33	37.86	38.02	38.57	42.53	42.62	38.44	41.84	40.58	33.58	40.11	39.29	39.65	44.36	42.82
12	36.82	41.97	42.70	43.32	46.35	46.10	42.47	45.15	44.27	38.12	43.01	42.51	43.93	46.60	46.03
13	26.90	32.78	34.78	32.18	36.18	38.57	33.51	37.08	38.19	28.18	35.50	37.01	33.57	38.90	40.28
14	32.23	37.75	37.92	38.50	40.60	40.29	38.16	40.03	38.41	33.07	36.09	36.37	39.09	40.35	39.57
15	35.98	42.01	39.77	40.32	43.61	41.03	41.43	44.93	40.69	36.64	41.58	38.46	41.26	45.81	41.05
16	34.62	39.10	43.35	40.94	44.32	45.92	40.42	43.15	43.92	36.04	41.03	42.13	42.24	45.68	46.68
17	35.17	41.78	41.48	41.04	44.44	46.09	41.50	44.68	44.55	36.31	41.40	43.10	42.11	45.27	46.87
18	31.06	37.09	37.58	36.51	40.29	41.40	37.48	40.76	40.24	32.15	37.83	38.57	37.71	41.75	42.05
19	31.49	36.78	36.36	39.75	43.00	44.24	37.38	40.37	39.80	32.94	37.16	38.05	40.94	43.85	44.48
20	34.78	40.64	40.53	41.23	43.91	45.86	41.02	43.45	43.03	35.83	40.73	42.60	42.40	44.38	46.41
21	31.69	37.12	38.86	37.80	41.34	43.15	38.00	41.12	41.22	32.90	38.69	39.92	39.18	43.26	44.19
22	33.67	39.12	38.27	39.24	42.17	42.23	39.33	41.80	41.00	34.88	39.15	39.01	40.35	42.57	42.36
23	38.21	44.66	43.91	44.58	46.79	45.97	43.75	45.87	43.56	38.69	41.62	40.98	44.86	45.46	45.28
24	29.90	35.23	37.12	35.03	38.04	40.32	36.36	38.95	39.72	31.06	36.15	37.58	35.88	39.53	40.71
Average	33.36	39.08	39.31	39.60	42.78	43.29	39.42	42.30	41.31	34.43	39.43	39.52	40.58	43.82	43.56

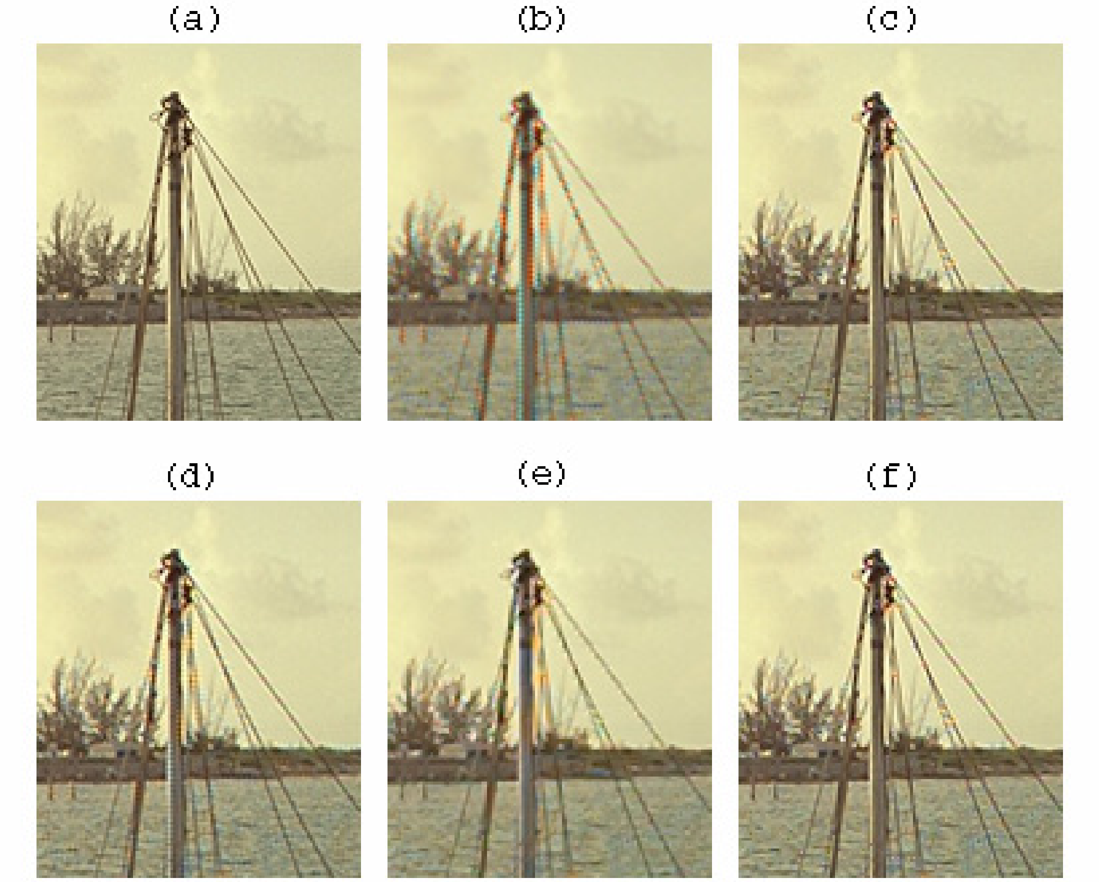


Fig. 5: Comparison of demosaicking methods on a cropped portion of image 6. (a) original image (b) bilinear (c) method in [5] (d) method in [6] (e) YUV method in [4] (f) proposed method

shows the number of operations required for demosaicking with the method in [6] and then converting to YCbCr 4:2:0 color space. In this analysis, the simple filter in equation (19) is used for limiting aliasing in the Cb and Cr channels and an efficient downsampling scheme is used (where the filtering operations are performed at the lower sampling rate [7]).

Comparison of Tables II and III shows that the proposed demosaicking method has lower complexity than the method in [6]. In the worst case, our method requires the same number of additions and fewer shifts operations. More importantly, the proposed method uses fewer multiplication operations, which are expensive to implement in the low cost DSP chips used in digital cameras. The multiplications are required for the RGB to YCbCr conversion, so our proposed method uses fewer multiplications than any RGB based method and subsequent conversion to YCbCr space.

The demosaicking method in [5] has considerably higher complexity than the method in [6], so our method has much lower complexity than that of [5]. We are not aware of any demosaicking methods with complexity lower or equal to [6] that provide comparable image quality.

V. CONCLUSIONS

In this paper we present a fast demosaicking algorithm that directly produces YCbCr 4:2:0 output. Our method saves considerable computational complexity by avoiding the need for performing demosaicking in RGB space and then converting from RGB to YCbCr 4:2:0. The proposed method achieves much higher PSNR than the only other demosaicking method that produces luminance and chrominance output. It also achieves better quality than fast RGB based demosaicking methods, with lower complexity than performing demosaicking in RGB space and then converting to YCbCr 4:2:0.

REFERENCES

[1] B.E. Bayer, “Color Imaging Array,” U.S. Patent 3,971,065.

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schaffer, and R. M. Murserau, “Demosaicking: color filter array interpolation,” IEEE Signal Processing Magazine, vol. 22, no. 1, pp. 44-54, January 2005.

[3] Hamilton, E., “JPEG File Interchange Format, Version 1.02,” Sept., 1992. available: http://www.w3.org/Graphics/JPEG/jfif.pdf.

[4] Mukherjee, J., Lang, M. K., and Mitra, S. K. 2005. Demosaicing of images obtained from single-chip imaging sensors in YUV color space. Pattern Recognition Letters. 26, 7, pp. 985-997, May 2005.

[5] J.F. Hamilton and J.E. Adams, “Adaptive color plane interpolation in single sensor color electronic camera,” U.S. Patent 5,629,734, 1997.

[6] S. C. Pei and I. K. Tam, “Effective color interpolation in CCD color filter arrays using signal correlation,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 6, pp. 503–513, Jun. 2003.

[7] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, Upper Saddle River, NJ: Prentice-Hall, 1996.

TABLE II: NUMBER OF OPERATIONS PER PIXEL REQUIRED FOR THE PROPOSED METHOD

Step	Addition	Shift	Multiplication	Absolute Value	Comparison
Green interpolation (worst case)	9	2.5	0	2	1
Low-pass red and blue	6	1	0	0	0
Generating chrominance	2	0	3	0	0
Calculating Luminance	4	0	2.75	0	0
Total (worst case)	21	3.5	5.75	2	1

TABLE III: NUMBER OF OPERATIONS PER PIXEL REQUIRED FOR THE METHOD IN [6] PLUS CONVERSION TO YCbCr 4:2:0 FORMAT

Step	Addition	Shift	Multiplication	Absolute Value	Comparison
Green Interpolation	4	1.5	0	0	0
Red Interpolation	4	0.75	0	0	0
Blue Interpolation	4	0.75	0	0	0
RGB to YCbCr Conversion	6	0	9	0	0
Filtering and downsampling rows, Cr	1	1	0	0	0
Filtering and downsampling columns, Cr	1	1	0	0	0
Filtering and downsampling rows, Cb	0.5	0.5	0	0	0
Filtering and downsampling columns , Cb	0.5	0.5	0	0	0
Total	21	6	9	0	0

Colin Doutre was born in Montreal, QC, Canada, in 1982. He received the B.Sc. degree in Electrical Engineering in 2005 from Queen’s University in Kingston, ON, Canada. He is currently working toward the M.A.Sc degree in Electrical and Computer Engineering at the University of British Columbia, Vancouver, BC, Canada. His research interests include video coding, image processing and digital cameras.

Mr. Doutre has received numerous awards and scholarships, including the Professional Engineers of Ontario Gold Medal and the Natural Sciences and Engineering Research Council of Canada (NSERC) Canada Graduate Scholarship (CGS).

Panos Nasiopoulos holds a Bachelor degree in Physics from the Aristotle University of Thessaloniki, Greece, and a Bachelor, Master and Ph.D. in Electrical and Computer Engineering from the University of British Columbia, Canada. He is an Associate Professor with the University of British Columbia (UBC) department of Electrical and Computer Engineering, the holder of the Professorship in Digital Multimedia, and the current Director of the Master of Software Systems Program at UBC. Before joining UBC, he was the President of Daikin Comtec US. Daikin had worked together with Toshiba to introduce the first complete DVD solution to the world in 1996 and developed all the software components of the DVD system. He was voted as one of the most influential DVD executives in the world. He is recognized as a leading authority on DVD and multimedia and has published numerous papers on the subjects of digital video compression and communications. He has organized and chaired numerous conferences and seminars and he is a featured speaker at multimedia/DVD conferences worldwide.

Dr. Nasiopoulos has been an active member of IEEE, the DVD Association and SPMTE as well as the ISO/ITU and In-Flight-Entertainment committees.

Konstantinos N. Plataniotis received his B. Eng. degree in Computer Engineering & Informatics from University of Patras, Greece in 1988 and his M.S and Ph.D degrees in Electrical Engineering from Florida Institute of Technology (Florida Tech) in Melbourne, Florida, in 1992 and 1994 respectively. He was with the Computer Technology Institute (C.T.I) in Patras, Greece from 1989 to 1991. He was a Postdoctoral Fellow at the Digital Signal & Image Processing Laboratory, Department of Electrical and Computer Engineering University of Toronto, Toronto, Canada from November 1994 to May 1997. From September 1997 to June 1999 he was an Assistant Professor with the School of Computer Science at Ryerson University, Toronto, Ontario. He is now an Associate Professor with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in Toronto, Ontario, Canada. His research interests include multimedia systems, image processing, signal processing, communications systems, biometrics, stochastic estimation, target tracking and pattern recognition.

Kostas Plataniotis is a registered professional engineer in the province of Ontario, and a member of the Technical Chamber of Greece. He is a Senior Member of IEEE, and he served as the IEEE Toronto Section 2004/05 Chair. Kostas is an Associate Editor for the IEEE Trans. on Neural Networks and the IEEE Signal Processing Letters . He is the image processing Area Editor for the IEEE Signal Processing Society's e-letter, the Technical Program Co-Chair for ICME 2006 and the Vice-Chair of ITSC 2006 .

Dr. Plataniotis is the 2005 recipient of IEEE Canada's Outstanding Engineering Educator Award "for contributions to engineering education and inspirational guidance of graduate students" and the co-recipient of the 2006 IEEE Trans. on Neural Networks Outstanding Paper Award for the published in 2003 paper " Face Recognition Using Kernel Direct Discriminate Analysis Alagorithms" (Juwei Lu, Konstantinos N. Plataniotis and Anastasios N. Venetsanopoulos, IEEE TNN, Vol. 14, No. 1, 2003).