CROSSTALK CORRECTION TECHNIQUE FOR SINGLE SENSOR CAMERA PROVIDED WITH BAYER COLOR FILTER ARRAY

Stefano Andriani and Harald Brendel

ARRI - Arnold and Richter Cine Technik GmbH
Image Science Group, R&D Department
Türkenstraße 89, 80799 Munich, Germany

ABSTRACT

In this paper we present a three-step crosstalk correction algorithm for single sensor still or video cameras provided with a Bayer color filter array. The first step is performed off-line, during the calibration or the development of the camera; it estimates the sensor response to different colors and analyzes the crosstalk. In the second step, the algorithm corrects “on-the-fly” the raw data of the sensor by using the crosstalk model estimated in the first step. The third, optional, step can be included to remove the residual crosstalk from the second step. It consists of a low-pass filter and it can be omitted if other image rescaling steps are included into the image processing chain. The resulting technique has proved to be effective in removing the crosstalk without introducing any other visual artifacts.

Index Terms— Image sensors, crosstalk, Bayer color filter array

1. INTRODUCTION

Most digital cameras produce color images by using a single sensor provided with a color filter array (CFA). In this way, adjacent pixels capture the light intensity value of different color bands, and the full color image is obtained by a color interpolation step (usually called “demosaicking”) which reconstructs the missing values. In recent years, many types of color filter arrays have been proposed in literature; however, the most popular one has been that presented by Bayer in [1]. It is composed by a repetition of $2 \times 2$ blocks where the green component has double the sample of the red and blue components, the green pixels are disposed in a quincunx grid and the red and blue ones are disposed in a rectangular grid. This choice is justified by the fact that the human visual system places more emphasis on the green rather then on the red and blue components.

The use of the color filter array is an effective way to obtain color images from single sensor cameras. The last years show that, for both still and video digital camera, the trend continues to be to increase the spatial resolution of the sensor without modifying the sensor size, i.e. the pixel pitch decreases and the pixels get smaller and closer to one another. This trend leads cameras to register more detail and to have a higher Nyquist frequency, reducing the aliasing problems, but also producing two main drawbacks: a higher noise level due to the reduced size of the pixel and a higher crosstalk due to the increased interaction among closer adjacent pixels.

Many factors contribute to the crosstalk problem, some optical and some electrical. Optically, if a ray of light arrives at the sensor plane with a certain angle it could be filtered by the color filter of one pixel and then strike the photosensor of the neighboring pixel; electrically, one factor is the electron leakage due to the minority carries and another is the read-out circuit which can allow influences between the signal read from one pixel and the signal read from another pixel. For this reason, the crosstalk problem is generally more evident along the read-out direction.

In the literature several solutions to this problem have been presented; discussing them here, the green pixels which belong to the green-red lines are called green-red pixels and denoted by $G_r$ and the green pixels which belong to the blue-green lines are called green-blue pixels and denoted by $G_b$.

M. Oizumi in [2] proposes to analyze the local behavior of the green pixels by taking into account a $5 \times 5$ windows centered on the green positions. If a flat region is detected, the algorithm first calculates a weighted average of the four closest green pixels and then linearly combines this average with the current green pixels. C.Y. P. Chao et al. in [3] present a two step approach: in the first step, they calculate an ad-hoc $3 \times 3$ two-dimensional filter on the basis of a crosstalk estimate and in the second step they apply this filter to the green pixels. J. Palum et al. in [4] propose to calculate the ratios between the green pixels belonging to the green-red lines and those belonging to the blue-green lines. Then they apply a smooth filter to these ratios and, finally, they use this smoothed version to correct the green-blue and the blue pixels. W. Li et al in [5] propose two algorithms: the first one performs the interpolation of the green-red pixels on the green-blue positions and then calculates the difference between the interpolated value and the green-blue pixel. The average of the differences, calculated in a $5 \times 3$ window centered on a green-red pixel, is then used to correct the central green-red pixel. The second algorithm calculates the average of the green-red pixels and the average of the green-blue pixels which belong to a $5 \times 5$ window and then uses the half of the average difference to correct both the green-red and the green-blue pixels. Hirakawa in [6] proposes a technique focused on color fidelity after or in conjuction with the demosaiucking. The main idea is to combine the crosstalk correction with the filters used during the demosaicking. To optimize this technique the author suggests precomputing this filter combination in an off-line step; however, this optimization can be easily performed only for simple demosaiicking algorithms.

The remainder of the paper is organized as follows. Section 2 presents the proposed approach to the crosstalk problem and de-
scribes the correction algorithm. Section 3 reports the experimental results and compares the proposed method with respect to the Oizumi [2] and Li [5] algorithms and in Section 4 the conclusions are reported.

2. PROPOSED ALGORITHM

To solve the crosstalk problem we start from another point of view and we consider this problem as a color dependent error instead of a spatial interaction which can be eliminated by filtering.

In nature, every color can be represented by a \([R', G', B']\) triplet as the tristimulus reduction of its spectral power distribution (SPD) filtered by the CFA materials [7]. When this color is captured by a single sensor camera provided with a Bayer CFA it is mapped into two representations: \([R, G_R, B]\) and \([R, G_B, B]\). These representations have the same red and blue values due to the Bayer pattern. In fact, as shown in Figure 1, the red and the blue pixels are always surrounded by four green pixels in the horizontal and vertical direction and by four blue or red pixels (respectively) in the diagonal directions. As a result, even if \(R\) is different from \(R'\) and \(B\) is different from \(B'\) their values are spatially constant just “only” slightly brighter or darker than the crosstalk-free one but they do not introduce any visually artifacts. On the other hand, the value of the green component depends on the green pixel position: the green-red pixels have two neighbor red pixels in the horizontal direction and two neighbor blue pixels in the vertical direction, and the green-blue pixels have two neighbor blue pixels in the horizontal direction and two neighbor red pixels in the vertical direction. As noted above, crosstalk is generally stronger along the read-out direction, thus the \(G_R\) and the \(G_B\) pixels are influenced by their neighbors in two different ways. This difference can be visually annoying and can create problems in post-acquisition processing of the image.

In our approach the crosstalk error is the difference between the two representations \([R, G_R, B]\) and \([R, G_B, B]\) and the aim of the proposed algorithm is to minimize it. When this minimization has been performed the difference between the crosstalk corrected \([R, G, B]\) and the real values \([R', G', B']\) is “just” a color matching problem.

To perform the difference minimization between \([R, G_R, B]\) and \([R, G_B, B]\) we propose a three step approach:

1. the crosstalk response of the camera sensor is analyzed and modeled by two functions having as input \([R, G_R, B]\) and \([R, G_B, B]\);

2. the models estimated in the first step are used on-the-fly to correct the green pixels;

3. (optional) the residual crosstalk error, due to the approxima-
tion of the models, is removed by a low-pass filtering.

The third step is optional because, at least in the digital cinema world, the spatial resolution of the camera is often different from the final production’s distribution formats, and an image rescale block will therefore be present in the image processing chain. In this situation, the low-pass filter of the rescaling block can also perform the crosstalk residual error removal and the third step can be skipped at the acquisition time without reducing the final visual quality at exhibition.

2.1. First step: Construction of the correction model

To analyze the crosstalk we used a programmable calibration lamp (see Figure 2). The lamp has three types of LEDs that allow specific but not completely selective exposure of the three color channels. The lamp was programmed to expose a three-dimensional lattice of \(n^3\) exponentially spaced RGB combinations. After some experimentation we found that \(n = 32\) was the best compromise between the useful dimension of the lattice and the simulation time needed to perform the test (with \(n = 32\) the simulation time is circa 4 hours).

When the crosstalk is plotted as the relative difference of the two green channels compared to the average of the two channels, the resulting graph resembles the shape of a fish. Testing several cameras revealed that the sensors differ in details of the crosstalk response (see Figure 3). In this figure, the crosstalk is expressed as the ratio between the difference of the two green channels and their average:

\[
\Delta G = \frac{2(G_R - G_B)}{G_R + G_B} \tag{1}
\]

The color of the dots indicates the amount of exposure in the red, green, and blue channel. A high exposure in blue, for example, results in in \(G_R > G_B\). The differences vanish when the exposure in red and blue are similar (magenta colors).

Fig. 2. ARRI Alexa camera with calibration lamp mounted.

Fig. 3. Crosstalk “Fish” error of two different cameras, named “A” above and “B” below.
An individual calibration of each sensor is not practical in factory camera production. Therefore, we decided to establish a model for the pooled data of several cameras. A direct measurement of the values that would populate a corrective 3D-LUT is not possible anyway, since the resulting signal levels are not on a regular lattice anymore. Two models need to be built, one that predicts the difference of the green channels based on $[R, GR, B]$ and one that predicts based on $[R, GB, B]$. To perform the regression of the data we chose thin plate regression splines [8] because they are a general solution to the problem of smoothing a function of multiple predictor variables from noisy observations of the function. To choose the dimension of the basis, we performed a cross validation. We fitted the data of five cameras using models with basis sizes from $3^3$ to $7^3$ and studied the performance of predicting the cross talk in four other cameras. It showed that no improvement is achieved by increasing the dimension beyond $6^3$. Later, we added measurements of four more cameras and fitted the model for the pooled data of nine cameras. Figure 4 shows the residual differences between the green channels with the prediction based on the pooled data; comparing this figure with Figure 3 we can see that the maximum residual is decreased from 0.25 to 0.08 which could be further reduced by a factor of 2 or 3 if a camera-specific model was used instead of the general one.

### 2.2. Second step: on-the-fly crosstalk correction

Once the models for $GR$ and $GB$ pixels are estimated, they are stored into two 3D-LUTs having dimension $16 \times 16 \times 16$. In the correction step, the algorithm calculates the red and blue values on the green position by averaging the two closest red and blue pixels, respectively, and then obtains the crosstalk correction values via trilinear interpolation of the data contained into the 3D-LUTs. These LUTs are internally log spaced in order to have a better quantization of the dark areas, so the linear data of the image must be converted into the 3D-LUT log domain by using this function:

$$f_{\text{log}}(C_{\text{lin}}) = \frac{n - 1}{2} \left(1 + 99 \frac{C_{\text{lin}}}{C_{\text{lin, max}}}ight)$$  

where $C_{\text{lin, max}}$ is the maximum value of the input image and $n$ the dimension of the LUT, in this case $n = 16$. In our implementation, this conversion has been pre-calculated and written into the LOGMAP LUT and its values are used as index for the 3D-LUTs.

The algorithm works in parallel and corrects the $GR$ and $GB$ pixels which belong to the same $2 \times 2$ Bayer block at the same time (refer to Figure 5). Taking $GR$ as the $GR$ pixel at position $[y][x]$, $G_b$ as the $GB$ pixel at the position $[y+1][x+1]$ and $BP[I][J]$ as the matrix which contains the Bayer pattern data from the sensor, the pseudo-code of the second step is as follows:

$$\begin{align*}
Gr &= \text{LOGMAP}[BP[I][J][x]]; \\
G_b &= \text{LOGMAP}[BP[I][J][x+1]]; \\
R_{Gr} &= \text{LOGMAP}([BP[I][J][x-1] + BP[I][J][x+1])/2; \\
B_{Gr} &= \text{LOGMAP}([BP[I][J][x-1] + BP[I][J][x+1])/2; \\
R_{Gb} &= \text{LOGMAP}([BP[I][J][x+1] + BP[I][J][x+2]/2]; \\
B_{Gb} &= \text{LOGMAP}([BP[I][J][x+1] + BP[I][J][x+2])/2];
\end{align*}$$

$$\begin{align*}
x_{tc, Gr} &= \text{trilinear}(3DLUT_{GR}, R_{Gr}, Gr, B_{Gr}); \\
x_{tc, Gb} &= \text{trilinear}(3DLUT_{GB}, R_{Gb}, Gb, B_{Gb});
\end{align*}$$

where $\text{trilinear}$ is the function which performs the trilinear interpolation of the 3D-LUT data, $3DLUT_{GR}$ is the 3D-LUT which contains the crosstalk correction data for the $GR$ pixels and $3DLUT_{GB}$ is the 3D-LUT for the $GB$ pixels.

Considering that the interpolation of the red and blue pixels on the green positions is performed by averaging the two closest pixels (i.e. it requires two sums and two bit-shifts for each green pixel), the complexity of this step can be easily approximated by the complexity of the implementation of the trilinear interpolation.

### 2.3. Third optional step: Low-pass filtering

In this paper, the residual crosstalk error has been removed by using a low-pass filter having the following characteristics:

- cut-off frequency: $\approx 0.7f_N$;
- bandpass ripple: $\leq 0.04$ dB;
- attenuation in stopband: $-60$ dB;
- order: 14

where $f_N$ is the Nyquist frequency of the sensor.
3. RESULTS

In the color processing chain of a digital camera, the crosstalk correction is the first algorithm after the read-out of the sensor, just before the white balancing and the color demosaicking. For the comparison we used real images captured by the ARRI Alexa camera and stored in the uncompressed raw data format. In this way, we were sure to work on the sensor data material without any other processing; however, as a drawback, we do not have any full color reference images or crosstalk-free reference images, so the only available comparisons among different algorithms is the visual one. For this test, we chose an image set which includes images with bright saturated colors and also detailed images such as a resolution chart. In this way, we can verify the effectiveness of the algorithms in removing the crosstalk without ruining the sharpness and the edge details of the images.

The results are shown in Figure 6. The method proposed by Oizumi in [2] works very well on the bright saturated color areas, removing all the structures introduced by the crosstalk; however, it slightly reduces the sharpness of the image, creates some greenish borders around the edges and where the frequencies are really high it introduces a strong color aliasing. The algorithm 2 proposed by Li et al. in [5] introduces a color shift noticeable in almost all the tested images and a red/blue border around all the edges. On the other hand, the crosstalk removal efficiency of the proposed method without the low-pass filter depends on the accuracy of the fitting model. In the figure, we can see that the crosstalk in the saturated red has been correctly removed; however, the crosstalk in the bright blue area still remains, i.e. the correction model correctly fits the reddish colors but has some problem for the bluish ones. In any case, it shows no problem with edges and, in general, with the high frequencies. If we consider the proposed method with the third low-pass filtering step we can see that also the crosstalk error in the blue colors has been completely removed without reducing the sharpness of the image, or at least, not in any visible way.

4. CONCLUSIONS

In this paper, we have presented an effective multi-step algorithm for the crosstalk problem correction. The main idea is to consider the crosstalk as a color problem and not a local behavior of the image which can be removed by filtering. The proposed algorithm has been shown to be effective in removing the crosstalk and to have a low complexity which make it a suitable candidate for a hardware implementation as an in-camera crosstalk reduction step.

5. REFERENCES