INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Reduction Of Colour Shifts And Artefacts Using A Spectral Interpolation Model Based Demosaicing Algorithm

K. Hamsini
M. Tech Student, Srinivasa Institute Of Tehnology And Management Science, Chittoor, A. P., India
V. Vijayakishore
Profeesor In Sitams College, Chittoor, A.P., India

Abstract:
A new demosaicking algorithm for single sensor imaging devices operating on an RGB color filter array (CFA) is introduced and analyzed. This algorithm enhances the universal demosaicing algorithm of Lukac et al by defining a new spectral interpolation model that exploits not only the information on the color of pixels but also the relative distance between neighboring pixels within an image. Moreover, we include an edge-detection model that makes this algorithm adaptive and reduces the presence of color shifts and artifacts. Experimental results indicate that the proposed algorithm exhibits excellent performance in terms of the commonly used objective criteria and at the same time it produces demosaiced images with impressive visual quality.

Key words: *Color filter array (CFA), color image resampling, demosaicing, edge detection*

1.Introduction

Digital images are comprised of data samples arranged in a two dimensional grid. These data samples are usually referred to as picture elements or pixels. The number of pixels in an image determines its resolution. The higher number of pixels an image has, the more information it could contain and the better it could represent the original data. In other words, all other things being equal, a high resolution image has better quality than a low resolution one. Changing the resolution of an image is called image resampling. One may need to resample an image for a variety of reasons. For instance, if a display device has lower resolution than an image to be displayed, then the image needs to be down sampled so that it could fit to the display screen. Similarly, if an image takes up too much data storage space or takes too long to transmit, a possible solution (other than applying compression) is to downsample the image. On the other hand, a low resolution image can be upsampled to improve its visual quality. From a digital signal processing point of view, image downsampling is arguably simpler than upsampling because in the downsampling case all the information is already available and the only challenge is to represent it with a smaller number of pixels. However, for the upsamplingcase, one needs to create new information by interpolating the available input pixels. From this point on, we will refer to image upsampling when we talk about image interpolation or image resampling.

The overall thesis of this work as Section II gives the illustration about the performance evaluation of the proposed approach. It also gives the comparative results between the proposed and previous approaches by evaluating the mean square error occurred between the input image and retrieved the image. Finally, section III gives the concluding details

2.Design Approach

The algorithm proposed in this work globally follows the same steps as the algorithm of universal algorithm: First start by estimating the green components and then the red and blue components. A post processing step is further performed in order to improve image quality.

2.1.Spectral Interpolation

Spectral Interpolation: As in the case of the algorithm ofLukac et al., we employ a simple weighted sum-based interpolation to start the interpolation of the missing green components. The interpolation is made within a neighbourhood of fixed size so as to use only the known green pixels close to almost pixel. As our algorithm aims to be used for any CFApattern, it may happen that there are not

enough green pixels within a neighbourhood to interpolate a missed pixel. Thus, to prevent the demosaicing solution from operating in a neighbourhood where we lack sufficient data, the following control mechanism is introduced

$$\sum_{(i,j)\in\xi}(d(i,j))_k = 1) \geq \lambda \quad \text{---- (2.1)}$$

Where $\lambda$is a design parameter denoting the minimum number of input values needed to be present when processing the kthcolor channel (k=2 in this case) in a local neighbourhood (is generally used in the algorithm of [46]., and we will also use that value in all our experiments). Using a 3×3 sliding window$\Psi(p,q) = \{(i,j): (i,j) \in \xi(p,q) = \{(p-1,q-1),(p-1,q),\ldots.(p+1,q+1)\}\}$, which placesvector undthe algorithmtion at the center of $\Psi(p,q)$ thealgorithm updates$I(p,q)$ when both$d(p,q)_k = 0$ and (1) are satisfied for $k = 2$, as follows:

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi \\ d(i,j)_k=1}} \omega'_{\Psi(p,q),k,\xi}(i,j)I(i,j)_k \quad \text{---- (2.2)}$$

Note that refers to in this work. Terms are given by

$$\omega'_{\Psi(p,q),k,\xi}(i,j) = \omega_{\Psi(p,q),k,\xi}(i,j)/\sum_{(g,h)\in\xi}\omega_{\Psi(p,q),k,\xi}(g,h) \quad \text{---- (2.3)}$$

In [46], weights are computed as

$$\omega_{\Psi(p,q),k,\xi}(i,j) = \left[1 + \sum_{\substack{(g,h)\in\xi \\ d(g,h)_k=1}} I(i,j)_k - I(g,h)_k\right]^{-1} \quad \text{---- (2.4)}$$

The weights expressed in (2.4) are not realistic since they only depend on the colour of the pixels and do not include the relative distance to the centre of the neighbourhood. Thus, these weights suggest that pixel of the same colour band, whatever their location in the neighbourhood, have the same impact on the interpolated pixel, which is not realistic. To address this issue, we propose a spectral interpolation model inspired from the centre of mass model found in physics. The centre of the mass of a system of particles is the point at which the system's mass behaves as if it were concentrated. It is a function of the positions and masses (weights) of the particles of the system. In our interpolation model, the centre of mass is the missed pixel to interpolate ; the particles are the pixels in the neighbourhood, and the masses (weights) are the functions of the colour intensity of the pixels in the neighbourhood. To include the location of the pixels, we add to the weights in (2.4) penalizing factor ,which measures the distance between the position of the missedpixel$I(p,q)$, denoted by(p,q), and the position of the pixel$I(i,j)$, denoted by (i,j). The greater the distance is, the more the weight is penalized and contributes less to the value of the missed pixel$I(p,q)$. Thus, our spectral interpolation model suggests a realistic approach where the colour of a missed pixel is computed as a weighted sum, with weights depending on both the colour and the position of the pixels within the neighbourhood. In the case of the Euclidean distance), $dist((p,q),(i,j))$ is defined as follows:

$$dist\left((p,q),(i,j)\right) = \sqrt{(p-i)^2 + (q-j)^2} \quad \text{--- (2.5)}$$

Any other distance can be also used such as Manhattan or Tchebychev distance. Thus, our proposed weights for the spectral interpolation are defined by

$$\omega_{\Psi(p,q),k,\xi}(i,j) = \frac{1}{dist((p,q),(i,j))} \times \left[1 + \sum_{\substack{(g,h)\in\xi \\ d(g,h)_k=1}} I(i,j)_k - I(g,h)_k\right]^{-1} \quad \text{--- (2.6)}$$

Introduction of an Edge-Detection model: Human visual systems are sensitive to edges present in images, and nonadaptivecolor interpolation algorithms often fail around edges since they are not able to detect them. The spectral interpolation described so far does not include any edge-detection step. Consequently, at edge points, interpolation may not be done along the edges but across edges, which tends to create colour artefacts. To address this issue, we propose to include an edge-sensing model for the green components in our demosaicing algorithm. The idea is to detect, at a missed pixel, if there is a potential horizontal, vertical, or diagonal edge. If we detect a horizontal (respectivelyvertical or diagonal) edge, then we interpolate the missed pixel by using only the known pixels along the horizontal (respectivelyvertical or diagonal) direction. To detect the edge points, we draw our inspiration from the methods used in some demosaicingalgorithms based on Bayer CFA [22], [36], and we generalize our approach to any CFA pattern.

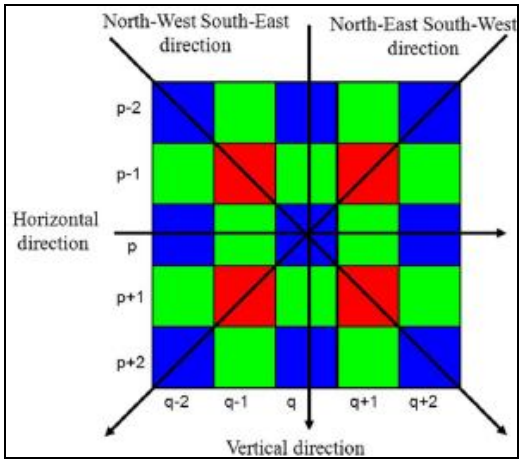

Figure1: Example Of CFA Configuration With The Directions Of Gradient Computation

The idea is to approximate, at a missed pixel, the horizontal gradient as the sum of two-by-two differences between successively known pixels of the same color in the horizontal line containing the missed pixel. The vertical gradient is similarly computed. The diagonal gradient is computed in two directions: the North–West South–East direction and the North–East South–West direction (see Fig. 1).

The idea of summing the differences between pixels of the same color allows us to detect changes in any of the red, green, and blue channels along a direction. Thus, by considering edgepoints as sharp changes in the red, green, and blue channels along a given direction, we are able to detect them. Additionally, instead of looking in the direction of the greatest contrast or change among all possible directions, we reduce the possible directions to the horizontal, vertical, and diagonal directions. This simplifies the complexity of the model and reduces the computing time. Knowing the horizontal, vertical, and diagonal gradients, we are able to interpolate missing pixels using neighboring known pixels that are located in the direction of the smallest contrast in order to preserve edges as much as possible without altering smooth regions. Thus, let us suppose that we want to compute the horizontal, vertical, and diagonal gradients at position (p,q) surrounded by windows (neighborhood) positions $\Psi(p,q)$ centered on (p,q). Let us define ζ the set of all position pixels forming $\Psi(p,q)$. Let us also define by ξ_h (respectively ξ_v) The set of horizontal (respectively vertical) positions of the pixels in the same horizontal (respectively vertical) line than pixel $K_{p,q}$ in $\Psi(p,q)$. In the same way, let us define by ξ_{WE} (respectively ξ_{SW}) The set of positions of the pixels in the WE direction (respectively SW direction) of $(p,q) \in \Psi(p,q)$. If $g_h(p,q)$ denotes the horizontal gradient, $g_v(p,q)$ the verticalgradient, $g_{WE}(p,q)$ and $g_{SW}(p,q)$ and the diagonal gradientsat location (p,q) , then we define the following:

$$g_h(p,q) = \sum_{(p,j_1),(p,j_2)} \sum_{\substack{(p,j_1),(p,j_2)\in\xi_h \\ j_1<j_2 \\ d(p,j_1)_h=d(p,j_2)_h=1 \\ \forall (p,j)\in\xi_h, j_1<j<j_2 \Rightarrow d(p,j)_h=1}} \left| K(p,j)_h - K(p,j_2)_h \right| \quad \text{--- (2.7)}$$

Condition $(p,j_1)(p,j_2) \in \xi_h$, means that the gradient is computed in pixels in the same horizontal line than a pixel (p,q). Condition $j_1 < j_2$ Creates a direction for computing the differences (in this case, from the pixels at the left in the horizontal neighborhood to the pixels at the right). Condition $d(p,j_1)_h = d(p,j_2)_h = 1$ means that the differences are computed only for known pixels from the CFA patterns. Condition $\forall (p,j_2) \in \xi_h, j_1 < j_2 < j_2 \Rightarrow d(p,j)_h = 1$, indicates and imposes that the differences are

computed between consecutiveknown pixels in the direction given by condition$i_1 < i_2$. The other gradients are computed similarly to $g_v(p,q)$as in(2.8)–(2.10), shown below.

$$g_v(p,q) = \sum_{k=1,2,3} \sum_{\substack{(i_1,q),(i_2,q)\in\xi_p \\ i_1<i_2 \\ \delta(i_1,q)_k=\delta(i_2,q)_k=1 \\ \forall(i_3,q)\in\xi_p,\ i_1<i_3<i_2 \Rightarrow \delta(i_3,q)_k=0}} |I(i_1,q)_k - I(i_2,q)_k| \quad \text{--- (2.8)}$$

$$g_{NE}(p,q) = \sum_{k=1,2,3} \sum_{\substack{(i_1,j_1),(i_2,j_2)\in\xi_p \\ i_1<i_2,\ j_1>j_2 \\ \delta(i_1,j_1)_k=\delta(i_2,j_2)_k=1 \\ \forall(i_3,j_3)\in\xi_p,\ i_1<i_3<i_2,\ j_1>j_3>j_2 \Rightarrow \delta(i_3,j_3)_k=0}} |I(i_1,j_1)_k - I(i_2,j_2)_k| \quad \text{---(2.9)}$$

$$g_{SW}(p,q) = \sum_{k=1,2,3} \sum_{\substack{(i_1,j_1),(i_2,j_2)\in\xi_p \\ i_1<i_2,\ j_1<j_2 \\ \delta(i_1,j_1)_k=\delta(i_2,j_2)_k=1 \\ \forall(i_3,j_3)\in\xi_p,\ i_1<i_3<i_2,\ j_1<j_3<j_2 \Rightarrow \delta(i_3,j_3)_k=0}} |I(i_1,j_1)_k - I(i_2,j_2)_k| \quad \text{– (2.10)}$$

I. if$g_v \le \tau$(we may be along an edge)

if $g_h(p,q) = g_v$, interpolate along the horizontal direction as follows

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi_p \\ \delta(i,j)_k=1}} \omega'_{p,q,k,\xi}(i,j)\,I(i,j)_k \quad \text{----- (2.12)}$$

Where

$$\omega'_{p,q,k,\xi}(i,j) = \omega_{p,q,k,\xi}(i,j) \Big/ \sum_{(g,h)\in\xi} \omega_{p,q,k,\xi}(g,h) \quad \text{----- (2.13)}$$

For all ξ and pixel (p,q), $\omega'_{p,q,k,\xi}(i,j)$ is computed by replacing ξ by ξ_p in (4.13), where as $\omega_{p,q,k,\xi}(i,j)$ is defined in (2.6).

II. if$g_h(p,q) = g_v$, interpolate along the vertical directions as follows

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi_q \\ \delta(i,j)_k=1}} \omega'_{p,q,k,\xi}(i,j)\,I(i,j)_k \quad \text{---- (2.14)}$$

Where $\omega'_{p,q,k,\xi}$ is computed by replacing the ζ by ξ_q in (2.13).

III. if $g_{NE}(p,q) = g_v$, interpolate along the NE directions as follows:

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi_{NE} \\ \delta(i,j)_k=1}} \omega_{p,q,k,\xi_{NE}}(i,j)\,I(i,j)_k \quad \text{---- (2.15)}$$

IV. if $g_{SW}(p,q) = g_v$, interpolate along the NE directions as follows:

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi_{SW} \\ \delta(i,j)_k=1}} \omega'_{p,q,k,\xi_{SW}}(i,j)\,I(i,j)_k \quad \text{----- (2.16)}$$

VI. *if* $g_v > \tau$, we are not on an edge, and we interpolateusing the complete neighborhood ξcontaining themissing pixel as follows:

$$I(p,q)_k = \sum_{\substack{(i,j)\in\xi \\ \delta(i,j)_k=1}} \omega'_{p,q,k,\xi}(i,j)\,I(i,j)_k$$

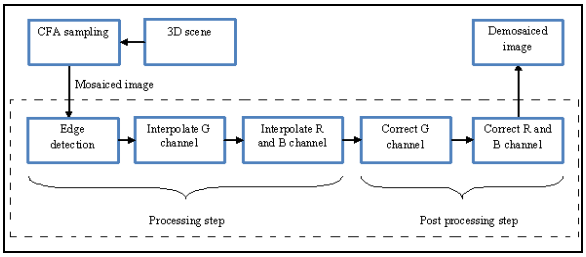

Figure 2: Block Diagram Of Proposed Demosaicing Algorithm

2.2.Spectral Interpolation Of The Red And Blue Components

They are computed from the G components by using a constant colour-difference model [34] –[37], [40], [41], [43], [46], which assumes that the red and blue pixels are correlated to the green pixels up to a simple offset over the extent of a neighbourhood. In fact, in natural images, the colour channels are highly mutually correlated [34]. In addition, the contrasts of the differences G–B And G–R are quite flat over small regions, and this property is suitable for interpolation [40]. Furthermore, the luminance information, which is important and essential for the perceived sharpness of a digital image, is mostly composed of green colour. Thus, by considering the colour differences within a neighbourhood, we can write the following:

$$I(p,q)_k = I(p,q)_2 + \sum_{\substack{(i,j)\in\xi \\ d(i,j)_k=1}} \omega'_{pq,ij\xi}\,(i,j)\,[I(i,j)_k - I(i,j)_2] \quad \text{----- (4.18)}$$

Where $k=1$ for the red pixels, $k=3$ for the blue pixels, and normalized coefficients ω' are calculated like in (2.13) with $\omega_{pq,ij\xi}$ defined this time as

$$\omega_{pq,ij\xi}(i,j) = \frac{1}{d_{pq}+(p,q)/(i,j)} \times \left[1 + \sum_{\substack{(g,h)\in\xi \\ d(g,h)_k=1}} \left\| [I(i,j)_k - I(i,j)_2] - [I(g,h)_k - I(g,h)_2] \right\| \right]^{-1} \quad \text{--(2.19)}$$

The procedure described here successively updates all flags $d(\cdot,\cdot)_k$ in a similar way with the G components. Note that no edge detection step is used for computing the R and B components since we have noticed that it is a good estimation of the G components that really improves the R and B components, and thus, we only focus on using edge detection for the G components.

2.3.Post-Processing Step

In the previous processing steps, the missed green components were computed using known green neighbours components (intrachannelmodel), whereas the red (respectively, blue) components were computed using the red and green (respectively, blue and green) components (interchannel). The post-processing step aims to correct the green components by using the red and blue components and then to use the newly computed green components to refine in turn the blue and red components. Note that the interchannel model could not have been used in the processing step for computing the G components since it would have caused a kind of chicken-or-egg dilemma: The R and B components are required for computing the G components, and at the same time, the G components are required for computing the R and B components. Many algorithms can be used for the post-processing step [43], [58] but we draw our inspiration from the method proposed in [46] since it is adapted for various CFA patterns. To guide the post-processing step, initial flag values $d(p,q)_k$, which were modified in the previous steps, are restored using original stored values $\sigma(p,q)_k$, for $p=1,2,\ldots,M, q=1,2,\ldots,N$ and $k=1,2,3$. Then, post processing of the G colour plane, in all locations (p,q) with both the constraints $d(p,q)_2=0$ and (4.1) enforced for $k=2$, can be realized using R or B components and the colour-difference model as follows:

$$I(p,q)_2 = I(p,q)_k + \sum_{\substack{(i,j)\in\xi \\ d(i,j)_2=1}} \omega'_{pq,ij\xi}\,(i,j)\,[I(i,j)_2 - I(i,j)_k] \quad \text{--- (2.20)}$$

Where weights ω' are those computed in (4.13) by using (4.19). Post-processing is applied only at spatial locations (p,q) where the components have been obtained using (4.11)–(4.17), i.e., or original spatialocations corresponding to missed green components. If (p,q) corresponds to the red CFA location $d(p,q)_1=1$, then the parameter $k=1$ is used in (4.20); otherwise, $k=3$ is used for the blue CFA location. After the G plane is enhanced, the post-processing step is completed by enhancing the R and B planes using the same equation defined in (4.18) for $k=1$ and $k=3$, respectively. This step is performed for R (respectively) components only in locations where they were originally missing in the CFA pattern, that is, $d(p,q)_1=0$ (respectively, $d(p,q)_3=0$).

3.Experimental Results

This chapter gives the details about the performance evaluation of the proposed approach. Here, we make a series of tests on various images. The images are extracted from the Kodak database.We also use several CFA patterns, for comparing our demosaicing algorithm to the algorithm of Lukac *et al.* The experimental protocol is as follows: An original image is sampled using a CFA, which gives a mosaicked image as output. A demosaicing algorithm (ours and that of Lucas *et al.*) is then applied to the masked image, and we obtain a final image. The final image is further compared with the original image in order to assess the quality of reconstruction. An initial subjective comparison is made by inspecting some visual artefacts such as colour aliasing on the images obtained through the demosaicing algorithms.

3.1.Visual And Numerical Analysis

In this work, we use two objective quality measures to assess how a demosaiced image is close to the original image: 1) the mean squared error (MSE) and 2) the CIEDE2000 [24],[25]. The MSE for a color image is computed as the mean valueof all the pixel-by-pixel squared differences for all colour bands.A small value of the MSE indicates small reconstruction errors,whereas a high value of the MSE indicates non-negligible reconstruction errors. The CIEDE2000 formula was published by the CIE in 2001. It works in the lab colour space and provides an improved procedure for the computation of perceptual colour differences. The CIEDE2000 variable in range [0, 1] and indicates thepercentage of color distortion between two images. A value thatis close to 0 indicates that the two colour images to be compared are quite identical, whereas a value that is close to 1 indicates high differences between the images.

Fig. 3 we present a part of an original image and the images obtained through the demosaicing algorithm of Lucas *et al.* And through our demosaicing algorithm, for the Bayer CFA. As we can notice, the algorithm of Lukac *et al.* Produces a lot of coloraliasing artefacts, which is not the case for our algorithm. Coloraliasing results from a poor application of the colour-difference model, which is shown near the edges. In fact, the introduction of an edge-detection step in our demosaicing algorithm enables to judiciously interpolate at the edge points. This explains why our algorithm performs better than the algorithm of Lucas *et al.*around the edges.

Figure 3: (A) Original Lena Image. (B) Gray Scale Image.
(C) Demosaiced Image Using The Algorithm Of Lukac Et Al. With The Bayer CFA
(D) Correspondingdemosaiced Image Using Our Algorithm

The increase in visual quality of our demosaicing algorithm compared with the algorithm of Lukac *et al.* is numerically confirmedby using the MSE and the CIEDE2000 quality measures.In Fig. 4, we plot the mean value of the MSE for each CFA pattern in order to observe how the pixel values of the demosaiced images are close to the values of the origina limage.

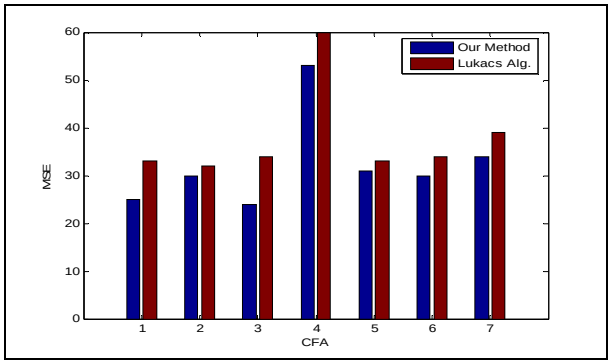

Figure 4: MSE Mean Values For The Various CFA Patterns

As shown in Fig.4, the mean value of the MSE for the differentCFA patterns is smaller for our new algorithm, compared withthe algorithm of Lukac *et al.*, meaning smaller reconstruction errors.

In this project we also noticed that the errors vary asfunction of the CFA pattern, meaning that the choice of the CFAis important for demosaicing algorithms. In Fig.5, we showthemean values of the CIEDE2000 quality measure for the differentCFA patterns.

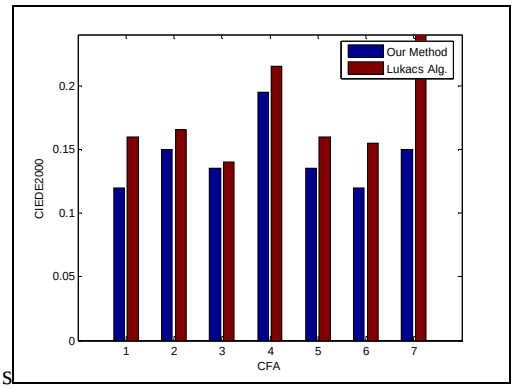

Figure 5: CIEDE2000 Mean Values For The Various CFA Patterns

This measure is considered to be better correlated with the perception of the human visual system than the MSE.As shown, the CIEDE2000 values for the algorithm of Lukac *etal.* are greater than those of our algorithm, meaning more color-distortion for the algorithm of Lukac *et al.*, compared with our algorithm.

4.Conclusion

A generic demosaicing algorithm has been proposed, which can process the CFA image captured from any RGB-CFA pattern. Our algorithm is based on a new spectral interpolation model inspired from the center of mass theory in physics. It also employs an efficient edge-sensing model that enables, at edge points, to direct the interpolation along the edge and not across
the edge. The experimental results have shown that our algorithm gives better images in terms of visual quality and objective quality measures, compared with the universal demosaicing algorithm of Lukac *et al.*

5.References

1. Adams Jr, J. and Hamilton Jr, J., \Adaptive color plane interpolation insingle sensor color electronic camera," Apr. 9 1996. US Patent 5,506,619
2. Atkins, C., Bouman, C., and Allebach, J.,\Optimal image scaling usingpixel classi_cation," in Image Processing, 2001.Proceedings. 2001 International
3. Conference on, vol. 3, pp. 864{867, IEEE, 2001.
4. Bayer, B., \Color imaging array," July 20 1976. US Patent 3,971,065.
5. Carey, W., Chuang, D., and Hemami, S., \Regularity-preserving imageinterpolation," Image Processing, IEEE Transactions on, vol. 8, no. 9, pp. 1293{1297,1999.
6. Chang, S., Cvetkovic, Z., and Vetterli, M., \Locally adaptive waveletbasedimage interpolation," Image Processing, IEEE Transactions on, vol. 15,no. 6, pp. 1471{1485, 2006.
7. Chen, Q. and Weinhous, M., \Sub pixel shift with Fourier transform to achieveefficient and high-quality image interpolation," in Proceedings of SPIE, vol. 3661,p. 728, 1999.