

University of Ottawa  
Faculty of Engineering · Co-operative Education

DEMOSAICKING THE HEXAGONAL ROTATED  
BAYER STRUCTURE USING THE ADAPTIVE  
LEAST-SQUARES LUMA-CHROMA  
DE-MULTIPLEXING ALGORITHM

NAVIRE INTERNAL REPORT NAV-TR-2010-01

by  
**Maria Urlea**  
VIVA Research Laboratory  
School of Information Technology and Engineering  
University of Ottawa

January 12, 2011

# Abstract

Image capture and image demosaicking are two important fields in image processing. Different sampling structures and their associated demosaicking algorithms produce images that are more or less pleasant to the eye.

The pattern that is analyzed here, the hexagonal rotated Bayer structure, sits on a hexagonal *lattice*. This is different than what is seen on most devices that have rectangularly sampled displays.

This report describes the implementation of the Least-Squares Luma-Chroma de-multiplexing (LSLCD) algorithm on the hexagonal rotated Bayer structure. It strives to offer a detailed overview of the whole process. It starts by describing the hexagonal rotated Bayer structure in a mathematical framework. Then, it expands upon the spatial demosaicking strategies that can be used to recreate the image from the color filter array signal. Next, it details the frequency domain demosaicking and makes note of the fact that, in this particular case, an adaptive scheme can be used to improve the results of the algorithm. Finally, the demosaicking quality versus algorithm complexity issue is discussed.

## Acknowledgment

I would like to thank Dr. Eric Dubois for the support given throughout the semester, as well as for the helpful comments on the earlier drafts of this report. This work was supported by the NSERC USRA program and by the NSERC Discovery Grants program.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgment</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Formation of the Test Images</b>	<b>2</b>
<b>3 Simple Interpolation Algorithms</b>	<b>6</b>
3.1 Bilinear Demosaicking . . . . .	6
3.2 Bicubic Interpolation . . . . .	7
<b>4 Foundations of Frequency Domain Demosaicking</b>	<b>8</b>
<b>5 Adaptive Frequency-Domain Demosaicking</b>	<b>13</b>
5.1 Selection of the Weighting Coefficients . . . . .	14
5.2 The Algorithm . . . . .	16
<b>6 Filter Design for <math>h_1</math>, <math>h_2</math>, <math>h_{3a}</math> and <math>h_{3b}</math></b>	<b>18</b>
6.1 Design of the $h_2$ Filter . . . . .	19
6.2 Design of the $h_{3a}$ and $h_{3b}$ filters . . . . .	20
<b>7 Optimization of the Algorithm Performance</b>	<b>21</b>
7.1 Determination of Constants . . . . .	22
<b>8 Objective Analysis of the Image Quality versus the Filter Complexity</b>	<b>25</b>
8.1 Filter Complexity Reduction Techniques . . . . .	28
<b>9 Analysis of the Subjective Performance of the LSLCD Algorithm on the Rotated Bayer Structure</b>	<b>29</b>
<b>10 Recommendation</b>	<b>31</b>
<b>11 Conclusion</b>	<b>32</b>

<b>A</b>	<b>Anti-Aliasing Filters and Down-Sampled Image Quality</b>	<b>33</b>
A.1	Effect of the Transition Band Width on the Down-Sampled Image . . . . .	34
A.2	Effect of the Transition Band Position on the Down-Sampled Images . . . . .	34
A.3	Effect of the Filter Size on the Down-Sampled Image . . . . .	35
A.4	Effect of the Diamond-Shape Corners on the Down-Sampled Image . . . . .	36
<b>B</b>	<b>Numerical Results</b>	<b>37</b>
<b>C</b>	<b>Greedy Algorithm Results</b>	<b>39</b>
	<b>Glossary</b>	<b>42</b>
	<b>References</b>	<b>43</b>

# List of Figures

1	Memory storage on a rectangular matrix . . . . .	2
2	Voronoi cell of the reciprocal lattice of the down-sampled image . . . . .	3
3	Parameters of the anti-aliasing filter . . . . .	4
4	CFA samples of the down-sampled image . . . . .	5
5	CFA structure on an (under-laying) rectangular lattice . . . . .	6
6	Hexagonal rotated Bayer structure . . . . .	9
7	Reciprocal lattices of the hexagonal rotated Bayer structure. . . . .	10
8	Luma-chroma components of an image . . . . .	12
9	Power spectrum of a CFA signal on the hexagonal rotated Bayer structure. . . . .	12
10	Frequency-domain demosaicking of a sample image taking into consideration different chroma components . . . . .	14
11	Local neighborhood of pixels in frequency-domain . . . . .	14
12	Frequency-domain demosaicking of a sample image weighting the chroma com- ponents . . . . .	16
13	Block diagram of the adaptive demosaicking algorithm for the hexagonal ro- tated Bayer CFA structure. . . . .	17
14	Frequency response of the $h_2$ filter . . . . .	20
15	Frequency response of the $h_{3a}$ and $h_{3b}$ filters . . . . .	21
16	Algorithm performance as function of position of center frequency of the mod- ulated energy extraction filters. . . . .	23
17	LSLCD algorithm performance as function of Gaussian filter size . . . . .	24
18	LSLCD algorithm performance as function of filter size of $h_2$ , $h_{3a}$ and $h_{3b}$ . . . . .	25
19	Demosaicking quality versus algorithm complexity for the $\frac{1024 \times 1536}{2}$ px images . . . . .	27
20	Demosaicking quality versus algorithm complexity for the $\frac{683 \times 1024}{2}$ px images . . . . .	27
21	Chroma interference in the high luma frequencies. . . . .	29
22	Chroma 1 - luma interference. . . . .	30
23	Chroma 2 - luma interference. . . . .	30
24	Imperfect reconstruction of edges, color leak. . . . .	31
25	Anti-aliasing filter parameters. . . . .	33
26	Anti-aliasing filter transition band width . . . . .	34
27	Anti-aliasing filter transition band position . . . . .	35
28	Anti-aliasing filter size . . . . .	35
29	Anti-aliasing filter frequency response . . . . .	36

## List of Tables

1	Demosaicking results for the rotated Bayer structure - $\frac{1024 \times 1536}{2}$ images . . .	37
2	Demosaicking results for the rotated Bayer structure - $\frac{683 \times 1024}{2}$ images . . . .	38
3	Greedy iterations for the rotated Bayer structure - $\frac{1024 \times 1536}{2}$ px image . . . .	39
4	Greedy iterations for the rotated Bayer structure - $\frac{683 \times 1024}{2}$ px image . . . .	40

# 1 Introduction

Most digital color cameras use a charged-coupled device (CCD) sensor and a color filter array (*CFA*<sup>1</sup>) to capture color images. CFAs come in a variety of colors and patterns. Each sensor element of the CFA measures only one *tristimulus value* value. To reconstruct a full color image, one needs to estimate the values of the other two tristimulus value values at each point.

The process of reconstruction the image from incomplete samples is called *demosaicking*. Numerous demosaicking algorithms taking into consideration the spatial domain were proposed [1]. However, even when these algorithms are complex, the final image quality is not entirely satisfactory.

In their work, Alleysson et al. [2] pointed out that, for certain patterns, the luma and chroma components are reasonably well isolated in the frequency domain. This observation led to the development of numerous new demosaicking algorithms using the frequency domain analysis [3, 4, 5]. For the well-known *Bayer structure*, these algorithms produce high quality results with a lower complexity than that of the best spatial domain algorithms [6].

The hexagonal rotated Bayer structure that is discussed in this report is nothing more than a variation of the Bayer pattern. It consists of a combination of red, green and blue samples having the green sample density twice as large as the density of the other color samples and its underlying lattice is hexagonal.

The rotated Bayer structure is more commonly known under the name SuperCCD and sensors using this pattern have been used by Fujifilm [7] as early as 1999. SuperCCD is actually a family of sensors with octagonal-shaped photodiodes placed on lines angled at 45 degrees. Fuji uses this type of sensor in the FinePix S6500fd and the FinePix F-series cameras, which are credited with exceptional signal to noise ratios [8, 9].

A very important observation is that the rotated Bayer structure lies on a hexagonal lattice. This influences the way that the various demosaicking algorithms are implemented and the way their final output is displayed.

The analysis of the rotated Bayer structure was sketched by Dubois in [10]. This report completes the analysis, provides optimization and extensive results. A web site was created to make these results and the code used to generate them available [11].

This document offers a detailed overview of the demosaicking process for the rotated Bayer structure (RBS). The first part of the report focuses on the particularities of the RBS and

---

<sup>1</sup>The definition of the terms in italic is given at the end of the document in Appendix: Glossary of terms

on the construction of test images. The next section discusses the foundations of frequency domain demosaicking. The following section touches upon the adaptive frequency domain demosaicking. Then, the Least-Squares Luma-Chroma De-multiplexing (*LSLCD*) algorithm adapted to the rotated Bayer structure is described. The last sections analyze how the computational complexity of the *LSLCD* algorithm can be reduced.

## 2 Formation of the Test Images

The rotated Bayer structure lies on a hexagonal lattice. However, most image displays in use take rectangular images as inputs. The algorithms developed for the RBS demosaicking have a hexagonally sampled CFA signal as an input and output a rectangularly sampled image.

A set of images widely used in the image processing literature are the 24 full color Kodak images<sup>2</sup>. These are  $2048 \times 3072$  px images sampled on a rectangular lattice. To be able to compare the results obtained with the demosaicking results of other researchers or with the results obtained on other sampling structures, the test images used are down-sized versions of these Kodak images. For ease of manipulation, the down-sampled images are stored in a checker-board pattern on a rectangular matrix, that is double its size. Note that, from now on, the size of the image referred to is specified as the size of the underlying matrix size times  $\frac{1}{2}$ . i.e. the size of the matrix that supports the image. For example the size of the image below, containing 25 samples, is denoted  $\frac{7 \times 7}{2}$ .

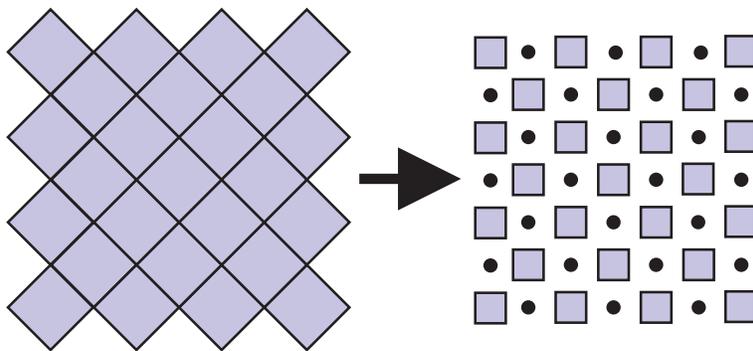


Figure 1: Hexagonally sampled image (left); Memory storage on a rectangular matrix (right).

The final demosaicked image lies on a rectangular matrix that is double the size of its corresponding CFA signal matrix. When the size of the down-sampled images was decided

<sup>2</sup>There are 24 original Kodak images. However, only 22 of these images are available to us. The files we have for the Kodak image 9 and the Kodak image 22 are corrupted.

upon, two sizes were considered:  $\frac{1024 \times 1536}{2}$  pixels and  $\frac{683 \times 1024}{2}$  pixels. The difference between these sizes is in the amount of detail that can be preserved. For the first option, the CFA signal has a number of pixels equal to the CFA signal pixel number used for testing the Bayer structure demosaicking algorithms [6]. This option would make it easier to compare the results obtained on the hexagonal rotated Bayer structure with those obtained on other structures such as the Bayer or the diagonal stripe patterns. The second option would allow for comparison of the algorithm output with demosaicked images having the same size. In the case of the smaller CFA, the final output image size is closer to the images sizes presented in [6].

Down-sampling an image by simply discarding some of its samples produces aliasing. To avoid this, one needs to pre-filter the image with an anti-aliasing filter. The characteristics of this filter are largely determined by the *Voronoi cell* of the down-sampled image. Figure 2 compares the unit cell of the reciprocal lattice of the original Kodak image with the unit cell of the reciprocal lattice of the down-sampled images. In this figure the units of  $u$  and  $v$  are in  $c/px$ . The unit cell of the reciprocal lattice of the original Kodak image would cover the interval  $[-0.5, 0.5] c/px \times [-0.5, 0.5] c/px$ .

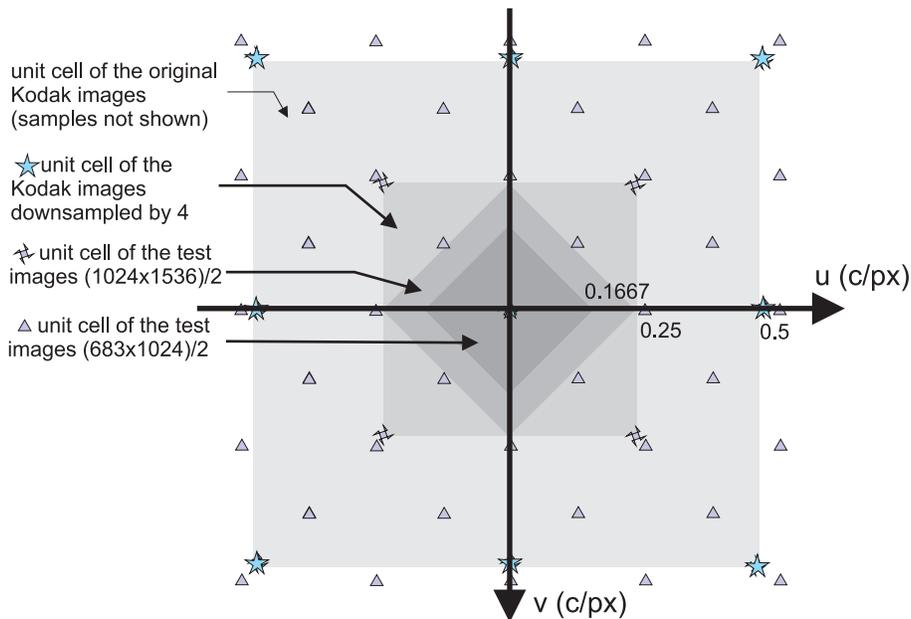


Figure 2: Voronoi cell of the reciprocal lattice of the original Kodak images compared to the unit cell of the reciprocal lattice of the down-sampled images.

Figure 2 gives a lot of insight about the pass-band of the filter that needs to be designed. To avoid aliasing, only the frequencies that are in the Voronoi cell of the smaller image should be kept.

To down-sample the original images to a suitable set of test images lying on a hexagonal lattice, the following steps need to be taken:

1. Pre-filter the images with a diamond shaped filter corresponding to the Voronoi cell of the desired down-sampled dimension;
2. Discard the samples that are not on the hexagonal lattice;
3. Create the corresponding rotated Bayer CFA image.

It is obvious that the ideal filter in the frequency domain has a pass-band region in the Voronoi cell of the lattice that is down-sampled to. Nonetheless, a very sharp cut-off of the filter bands in the frequency domain would cause rippling in space (Gibbs phenomenon). Having a smooth transition from the pass-band to the stop band minimizes this effect. However, if the image is pre-filtered too much, then the space domain representation obtained is blurry. If the image is pre-filtered too little, then aliasing will be noticeable in the space domain. The width and the relative position of the transition band are the two parameters that influence the filter design most. Other parameters that are taken into consideration when designing the filter are the filter size and the filter ‘corners’. The only limitation imposed on the filter is that the transition band be linear. The figure below explains what each of the filter parameters means. The filters are designed using the *windowing* design method.

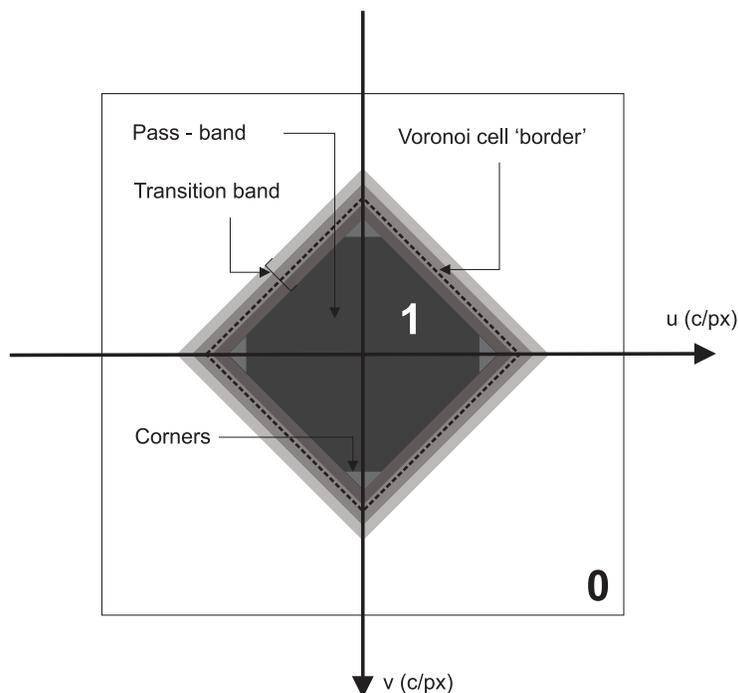


Figure 3: Parameters taken into consideration when designing the anti-aliasing filter.

The empirical filter construction algorithm is done following the steps shown below:

1. Observe the effect that the transition band width has in the space domain (look for rippling);
2. Observe the effect that the transition band position has in the space domain (look for blurring/aliasing);
3. Observe how the quality of the image is affected by the filter size (smallest filter size that can be used for a high subjective quality);
4. Analyze whether or not the high frequencies present in the image have a bad influence on the image subjective quality and decide whether the high horizontal or vertical frequencies must be filtered out more.

The image pre-filtering and down-sampling was carried with the help of MATLAB<sup>®</sup>. The decisions taken when designing the anti-aliasing filters are explained at the end of this document in Appendix A. For the  $\frac{1024 \times 1536}{2}$  down-sampled image the anti-aliasing filter has the transition band width of 0.10 c/px, centered at the Voronoi cell border. The filter size that gives the best quality vs. complexity performance is a  $25 \times 25$  filter. For the smaller image, the anti-aliasing filter is a diamond shaped filter having a transition band of 0.08 c/px, centered at the Voronoi cell border. In this case, the filter size that gives the best results is  $21 \times 21$ .

Once the antialiasing filtering is done, the down-sampling is done by discarding the samples as shown in figure 4. The samples that are kept are represented by the blue squares.

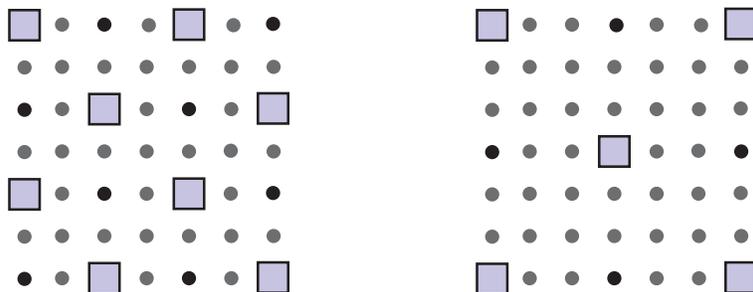


Figure 4: Samples kept when down-sampling to the  $\frac{1024 \times 1536}{2}$  image (right); and to the  $\frac{683 \times 1024}{2}$  image (left).

The final step when building the test images is to create the CFA signals corresponding to each of them. The CFA signals follow the rotated Bayer structure and are defined on a hexagonal lattice and stored in a rectangular matrix that is double its size.

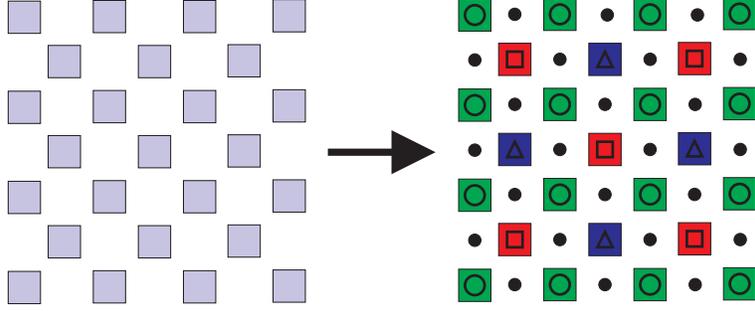


Figure 5: CFA sampling structure of the image and the way it is saved on a larger, [2:1] rectangular lattice.

### 3 Simple Interpolation Algorithms

This section discusses briefly the performance of the spatial de-multiplexing algorithms when used on the hexagonal rotated Bayer CFA. The RBS CFA is regular and periodic. It is comprised of three different samples: red, green and blue, and the number of green samples is double than that of any other color. The samples are arranged as shown in figure 5.

#### 3.1 Bilinear Demosaicking

With the exception of the nearest neighbor, bilinear interpolation is the simplest way to reconstruct an image. Bilinear interpolation of the rotated Bayer CFA can be done in a single step, using the following filters:

$$h_R = \frac{1}{4} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 1 & 0 & 4 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad h_G = \frac{1}{4} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad h_B = \frac{1}{4} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 1 & 0 & 4 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

The filters above are doing two things simultaneously: they bilinearly interpolate the CFA to full color in the hexagonal lattice and up-scale the image on the underlying rectangular lattice.

Objective results for this type of interpolation can be found at the end of the document in Appendix B: Numerical results.

## 3.2 Bicubic Interpolation

Bicubic interpolation is another spatial demosaicking algorithm. The CFAs are interpolated to the larger lattice in two steps. First, the CFA is bilinearly interpolated to the full color hexagonal image. Then, bicubic interpolation is used to up-sample to the rectangular matrix.

Constructing the matrix for the bicubic interpolation is a bit more complex. This process is explained here because this type of interpolation will also be used later on.

The rectangular  $[2 : 1]$  interpolator is assumed to be a  $\frac{5 \times 5}{2}$  finite impulse response (FIR) filter with quadrantal and diagonal symmetry. The general form of this type of filter is:

$$h = \begin{bmatrix} 0 & b & 0 & b & 0 \\ b & 0 & a & 0 & b \\ 0 & a & 1 & a & 0 \\ b & 0 & a & 0 & b \\ 0 & b & 0 & b & 0 \end{bmatrix}.$$

This filter has a frequency response of

$$\begin{aligned} H(u, v) &= 1 + a(\exp(-j2\pi u) + \exp(j2\pi u) + \exp(-j2\pi v) + \exp(j2\pi v)) + \\ &\quad b(\exp(-j2\pi(2u + v)) + \exp(j2\pi(2u + v)) + \exp(-j2\pi(2u - v)) + \\ &\quad \exp(j2\pi(2u - v)) + \exp(-j2\pi(u + 2v)) + \exp(j2\pi(u + 2v)) + \\ &\quad \exp(-j2\pi(u - 2v)) + \exp(j2\pi(u - 2v))) \\ &= 1 + 2a(\cos(2\pi u) + \cos(2\pi v)) + \\ &\quad 2b(\cos(2\pi(u + 2v)) + \cos(2\pi(u - 2v)) + \\ &\quad \cos(2\pi(2u + v)) + \cos(2\pi(2u - v))). \end{aligned}$$

The DC gain of this filter is required to be

$$H(0, 0) = 1 + 4a + 8b = 2.$$

This implies that

$$a + 2b = 0.25.$$

Note that if  $b = 0$  and  $a = 0.25$ , we obtain the bilinear interpolator.

The diagonal response of this filter is

$$H(u, u) = 1 + 4a \cos(2\pi u) + 4b(\cos(2\pi(3u)) + \cos(2\pi(2u))) + 4(a + b) \cos(2\pi u) + 4b \cos(6\pi u).$$

Given that the frequency response of the one-dimensional cubic interpolator is

$$1 + \frac{9}{8} \cos(2\pi u) - \frac{1}{8} \cos(6\pi u),$$

if we set  $a = \frac{5}{16}$  and  $b$  to  $b = -\frac{1}{32}$ , the bicubic interpolator matrix obtained becomes

$$h = \frac{1}{32} \begin{bmatrix} 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 10 & 0 & -1 \\ 0 & 10 & 32 & 10 & 0 \\ -1 & 0 & 10 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 \end{bmatrix}. \quad (1)$$

The RBS CFA is bicubically interpolated to a full color image and Appendix B offers a full list of results.

Spatial domain reconstruction of the images is very easy to understand and to implement. However, experience on other sampling structures such as the Bayer or the diagonal stripe [11] suggests that this approach does not give the best quality results.

## 4 Foundations of Frequency Domain Demosaicking

Figure 6 shows a pattern of hexagonal rotated Bayer CFA signal on a sampling lattice  $\Lambda$ . Similar to the well-known Bayer structure, the rotated Bayer structure is a pattern of red, green and blue samples multiplexed over the disjoint shifted sub-lattices  $\Lambda_R$ ,  $\Lambda_G$  and  $\Lambda_B$ . The density of the green samples is twice that of the red or blue samples and only one tristimulus value value is associated to each point of the lattice. One period of this pattern is comprised within the bold lines of the figure.  $X$  is used as unit of length,  $X = 1\text{px}$ .

In his book chapter, Dubois [10] gives a mathematically rigorous description of sampling structures for color filter arrays. Throughout this section, his notation is used to describe the hexagonal rotated Bayer structure. A serious user of this report is assumed to have assimilated the content of [10].

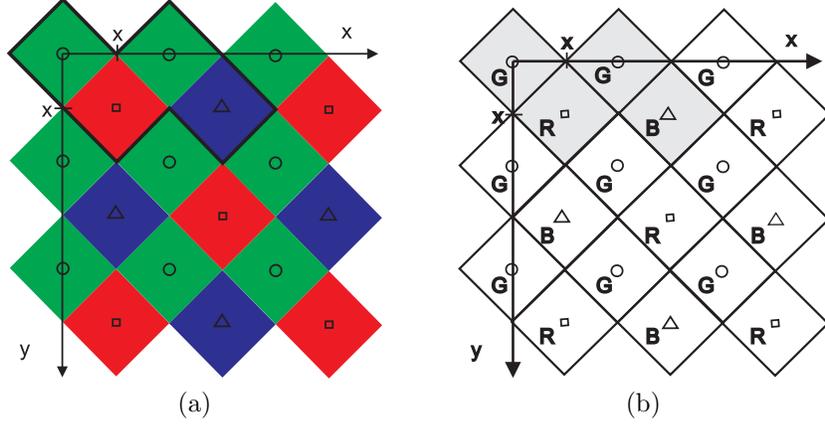


Figure 6: (a)Hexagonal rotated Bayer color structure; (b) Corresponding color structure lattices:  $\Lambda_R(\square)$ ,  $\Lambda_G(\circ)$  and  $\Lambda_B(\triangle)$ .

The rotated Bayer structure lies on a hexagonal lattice  $\Lambda$  that can be written as

$$\Lambda = LAT \left( \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \right).$$

Alternatively,

$$\Lambda = \{(m_1, m_2) \in \mathbb{Z}^2 \mid m_1 + m_2 \text{ is even}\}.$$

Figure 6 shows the origin of this lattice placed on the upper-left corner, on a green sample. The  $y$  axis is oriented downward. The hexagonal rotated Bayer structure pattern contains three sensor classes (red, green and blue) and four sensor elements (forming the basic pattern) each lying on a disjoint shifted sub-lattice of  $\Lambda$ .

The period of this pattern is comprised of four sensor elements that can be expressed by

$$\mathbf{B} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

The sensor colors can be expressed by

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The first two rows of this matrix represent the green color. The reciprocal lattices corre-

sponding to the structure and its basic pattern cell are found to be

$$\Lambda^* = LAT \left( \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} \right)$$

and

$$\Gamma^* = LAT \left( \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{4} \end{bmatrix} \right).$$

Figure 7 illustrates the reciprocal lattices corresponding to the rotated Bayer structure and the associated Voronoi cell.

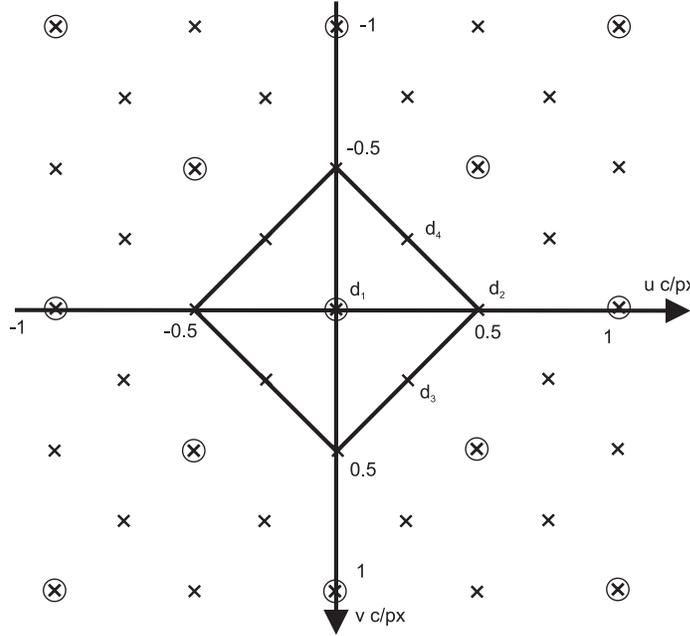


Figure 7: Reciprocal lattices of the hexagonal rotated Bayer structure.

The reciprocal coset representative matrix  $D$  is:

$$\mathbf{D} = \begin{bmatrix} 0 & 0.5 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & -0.25 \end{bmatrix}$$

In their paper, Alleysson et al. [2] showed that the CFA signal can be manipulated such that the luma-chroma representation of the signal becomes obvious in the frequency domain. Dubois [10] showed that the resulting luma and chroma signals can be obtained as a function of the CFA RGB samples by calculating the matrix  $M = \frac{1}{K}[\exp(-j2\pi\mathbf{D}^T\mathbf{B})]\mathbf{J}$ .

This gives:

$$\mathbf{M} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ -1 & 2 & -1 \\ -1 & 0 & 1 \\ 1 & 0 & -1 \end{bmatrix}$$

and the corresponding signals:

- Luma:

$$q_1[\mathbf{x}] = \frac{1}{4}f_1[\mathbf{x}] + \frac{1}{2}f_2[\mathbf{x}] + \frac{1}{4}f_3[\mathbf{x}]$$

- Chroma 1:

$$q_2[\mathbf{x}] = -\frac{1}{4}f_1[\mathbf{x}] + \frac{1}{2}f_2[\mathbf{x}] - \frac{1}{4}f_3[\mathbf{x}]$$

- Chroma 2a

$$q_{3a}[\mathbf{x}] = -\frac{1}{4}f_1[\mathbf{x}] + \frac{1}{4}f_3[\mathbf{x}]$$

- Chroma 2b

$$q_{3b}[\mathbf{x}] = \frac{1}{4}f_1[\mathbf{x}] - \frac{1}{4}f_3[\mathbf{x}]$$

Notice that  $q_{3a} = -q_{3b}$ . The CFA signal representing the image in the frequency domain can be written as:

$$F_{\text{CFA}}(u, v) = Q_1(u, v) + Q_2(u - 0.5, v) + Q_3(u - 0.25, v - 0.25) - Q_3(u - 0.25, v + 0.25)$$

The CFA signal is a combination of a baseband component (luma) and two modulated color difference signals (chroma) [10]. Figure 9 shows where the different components can be identified on the CFA signal power spectrum. This spectrum is obtained by using the method of averaging modified periodograms on image 8 of the modified Kodak set, the Alfred image, and the different components are easily identified on this figure.

In spatial domain, the chroma 1 appears as green-magenta and the chroma 2 component as blue-orange. Figure 8 shows the different frequency-domain components equivalent in spatial domain.

A closer look at the power spectrum helps identify potential problems with frequency domain demosaicking algorithms. High frequency luma patterns intrude into the chrominance bands, possibly causing false colors. High frequency chrominance information overlaps with the luma band resulting in false patterns. This will be discussed in more detail later in section 5.



Figure 8: Luma-chroma components of an image: (a) Original image; (b) Luma component; (c) Chroma 1 component; (d) Chroma 2 component.

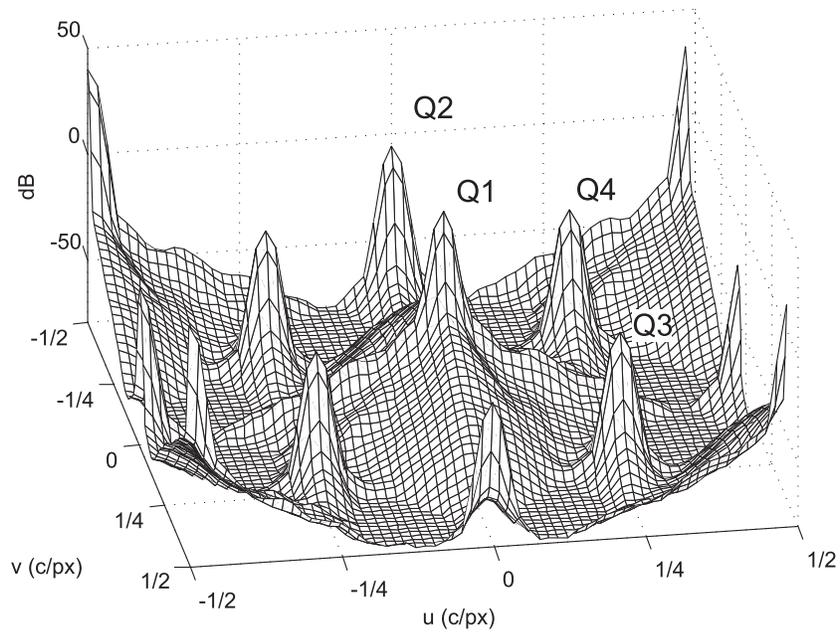


Figure 9: Power spectrum of a CFA signal on the hexagonal rotated Bayer structure.

Filters can be designed to extract the different modulated components from this signal [10]. Then, the approximated color channels can be obtained by de-modulation and simple matrix

transformations:

$$\begin{aligned} R[\mathbf{x}] &= q_1[\mathbf{x}] - q_2[\mathbf{x}] - 2q_3[\mathbf{x}] \\ G[\mathbf{x}] &= q_1[\mathbf{x}] + q_2[\mathbf{x}] \\ B[\mathbf{x}] &= q_1[\mathbf{x}] - q_2[\mathbf{x}] + 2q_3[\mathbf{x}] \end{aligned}$$

Depending on the filters chosen, this approach can significantly improve the demosaicking algorithm results. Simply designing Gaussian filters and modulating them at the appropriate chroma frequencies gives results that are more satisfactory than the spatial domain algorithms results [11]. However, taking into account the sampling structure particularities, these results may be improved.

## 5 Adaptive Frequency-Domain Demosaicking

On the power spectrum of the CFA signal in figure 9, there are two important things one needs to notice. The first is that  $Q_3$  appears both at  $(0.25, 0.25)$  and  $(0.25, -0.25)$ . The second is that high frequency luma components overlap with high frequency chroma components: high frequency luma patterns intrude into the chrominance bands, resulting in false colors and high-frequency chrominance information intrudes into the luma band, resulting in false luma patterns, often having a zipper-like appearance.

Either one of the two copies of  $Q_3$  can be used to reconstruct the signal. However, due to local properties of the image, both reconstructions suffer from cross-talk, but in different ways.

Each of the two copies privileges one of the diagonal directions. For example, in the second image of figure 10, the details on the -45 degree diagonal are the ones to suffer the most.

The local neighborhood of each pixel can be viewed in frequency domain as shown in figure 11.

For the first scenario,  $q_{3a}$  is the better estimate. For the second one, it is  $q_{3b}$  that would give better results. When the pixels belong to one of the 45 degree diagonal pattern, only one of the two copies will suffer from cross-talk due to the luma interference. The other one can be used for a very good reconstruction of the image. The following section details the steps taken to develop an algorithm that is able to analyze the neighborhood of each of the picture pixels and detect which of the two estimates is the local better choice.

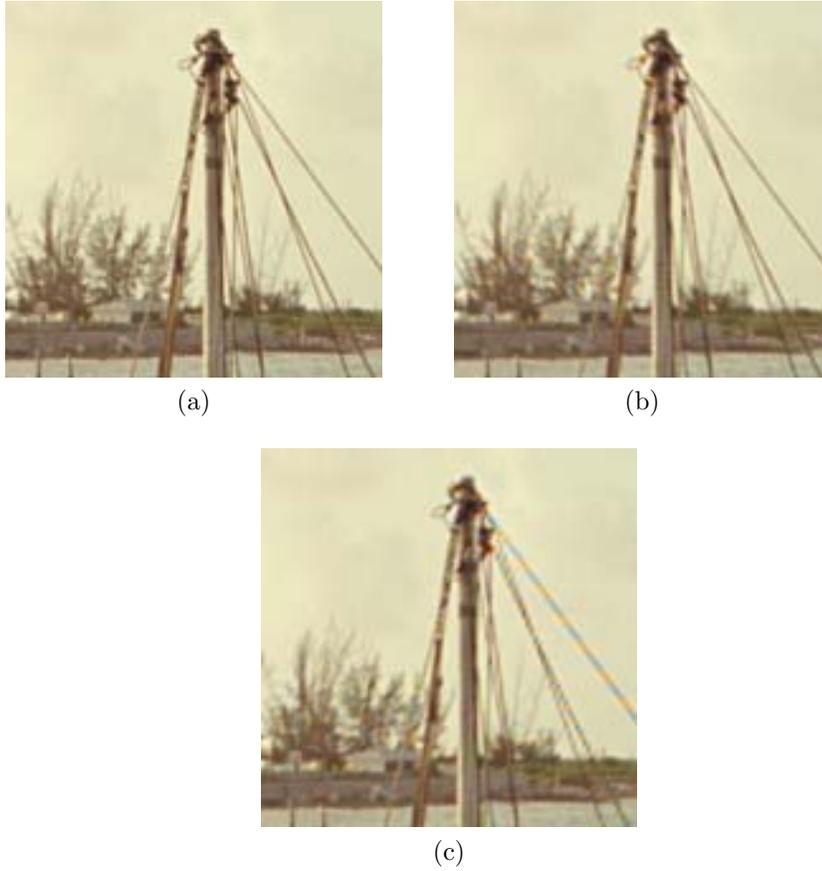


Figure 10: Frequency-domain demosaicking of a sample image taking into consideration different chroma components: (a) Original image; (b) Only  $q_{3a}$  is used; (c) Only  $q_{3b}$  is used.

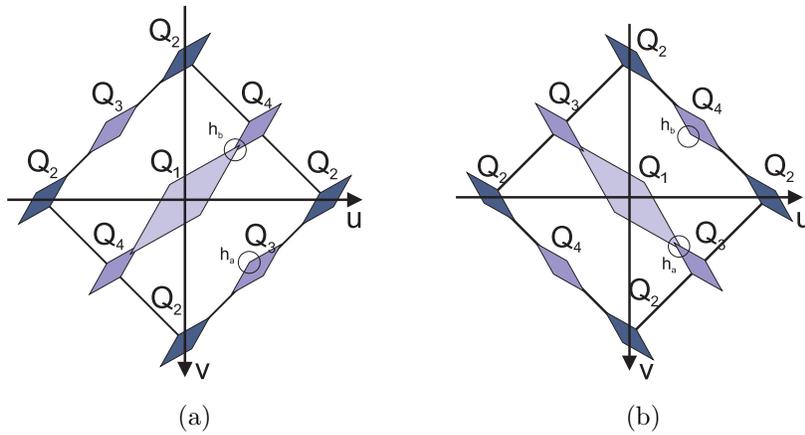


Figure 11: Local neighborhood of pixels in frequency-domain: (a)  $q_{3a}$  is the better estimate; (b)  $q_{3b}$  is the better estimate.

## 5.1 Selection of the Weighting Coefficients

Averaging the coefficients corresponding to the chroma 2 components improves the result that is obtained when using only one of these two copies, however this output is not good

enough. One would like a function that is able to detect which of the two copies is less corrupted.

In order to automatically choose the  $q_3$  copy that is locally less influenced by cross-talk, one needs to implement a function, (say  $w$ ) that is able to make this selection.

The requirements for  $w$  are similar to those already used for the Bayer structure [12] and therefore the scheme has been adopted:  $\{w, (1 - w)\} \in [0, 1]$ . Let  $T(x, y)$  be the spatial-domain function that describes the way  $w$  is chosen.  $T$  will be a function of how much the chroma 2 copies interfere with the luma component.  $T : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  has to respect the following conditions:

1.  $T = \frac{1}{2}$  , if the two chroma 2 elements are equally corrupted
2.  $0 \leq T < \frac{1}{2}$  if  $q_{3a}$  is locally more corrupted than  $q_{3b}$  i.e. locally,  $e_a < e_b$  and  $\lim_{e_b \rightarrow 0} T(x, y) = 0$
3.  $\frac{1}{2} < T \leq 1$  if  $q_{3b}$  is locally more corrupted than  $q_{3a}$  and  $\lim_{e_a \rightarrow 0} T(x, y) = 1$

One such function is:

$$T(x, y) = \frac{e_b}{e_a + e_b}$$

where  $e_a$  and  $e_b$  are the local luma-chroma energies corresponding to each pixel and are defined in equation 2.

The  $q_3$  value that will be used when demosaicking is calculated by the weighted average:

$$\hat{q}_3[m_1, m_2] = w[m_1, m_2] \cdot \hat{q}_{3a}[m_1, m_2] + (1 - w[m_1, m_2]) \cdot \hat{q}_{3b}[m_1, m_2], \quad (m_1, m_2) \in \Lambda$$

Using this estimator when demosaicking eliminates a lot of the cross-talk and outputs a better quality colored image.

By using one or the other  $q_3$  component, the cross-talk in specific regions of the picture can be eliminated. Locally, the image is more or less diagonal and deciding upon the weighting coefficients helps eliminate the cross-talk in the reconstruction of regions.

An important element when deciding which  $q_3$  copy to use is the energy of the pixel neighborhood. In the frequency domain, the coefficients of different components are additive. The modulated high frequencies of  $q_3$  overlap on the high luma frequencies. To obtain the local pixel energy for each of the two chroma 2 copies, we need to sample the energies of the frequencies that are in the overlapping region and demodulate them to baseband.



Figure 12: Frequency-domain demosaicking of a sample image weighting the chroma components: (a) Original image; (b) Demosaicked image with  $q_{3a}$  and  $q_{3b}$  components weighted.

The cross-talk estimates can be written as:

$$e_k = (f_{\text{CFA}} * h_k)^2 * h_m \quad (2)$$

where  $h_k$  filters out the overlapping region, the power two transforms the quantity into an energy and  $h_m$  is a moving average filter used for local smoothing.

Empirical results suggest that filtering (overlapping) regions centered at  $(0.0950, 0.0950)c/\text{px}$  and  $(0.0950, -0.0950)c/\text{px}$  and using their energies in the demosaicking algorithm adaptive scheme gives the best results<sup>3</sup>.

## 5.2 The Algorithm

Once the luma and chroma components are filtered efficiently and a representative interval of the cross-talk energies is extracted, the reconstructed image should be relatively free of color artifacts.

The algorithm block diagram is drawn in figure 13. In this figure, the points considered are in  $\Lambda = \{(m_1, m_2) \in \mathbb{Z} \mid m_1 + m_2 \text{ is even}\}$ .

The frequency-domain adaptive demosaicking algorithm becomes:

1. filter  $f_{\text{CFA}}$  by  $h_2$  to get  $\hat{f}_{C1m} = f_{\text{CFA}} * h_2$  modulated at  $[0.0, 0.5]$ . Demodulate to baseband to get  $\hat{f}_{C1}[m_1, m_2] = \hat{f}_{C1m}[m_1, m_2](-1)^{m_1}$ .

---

<sup>3</sup>The method used to find these locations is discussed in a later section.

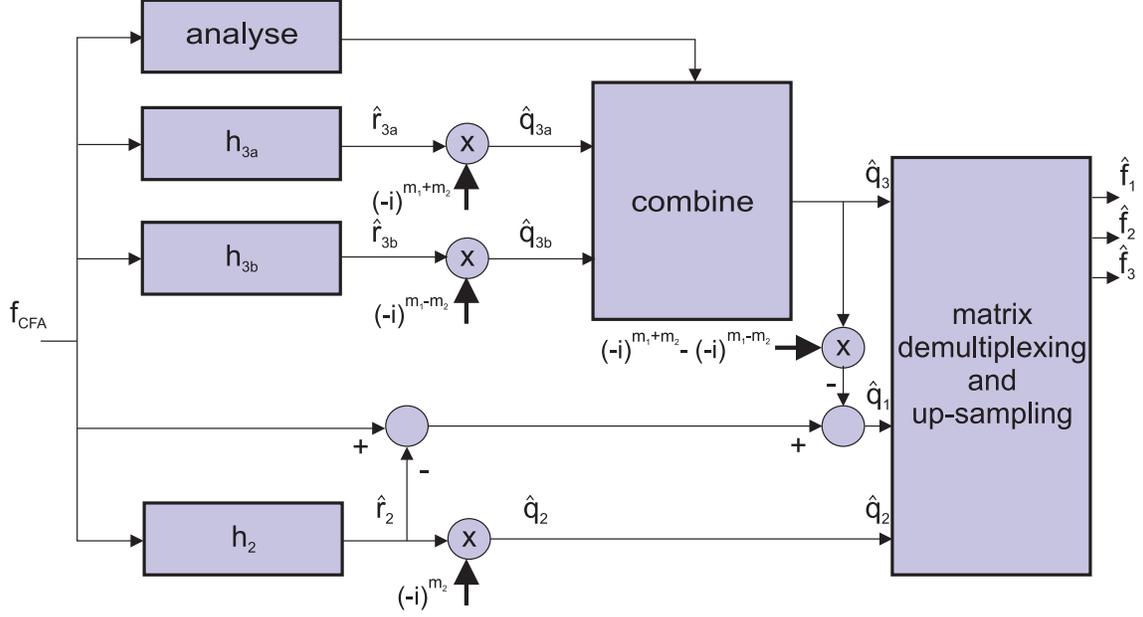


Figure 13: Block diagram of the adaptive demosaicking algorithm for the hexagonal rotated Bayer CFA structure.

2. filter  $f_{CFA}$  by  $h_{3a}$  to get  $\hat{f}_{C2ma} = f_{CFA} * h_{3a}$  modulated at  $[0.25, 0.25]$  and by  $h_{3b}$  to get  $\hat{f}_{C2mb} = f_{CFA} * h_{3b}$  modulated at  $[0.25, -0.25]$ . Demodulate  $\hat{f}_{C2ma}$  and  $\hat{f}_{C2mb}$  to get  $\hat{f}_{C2a}[m_1, m_2] = \hat{f}_{C2ma}[m_1, m_2](-i)^{m_1+m_2} \cdot \text{mod}(m_1 + m_2 + 1, 2)$  and  $\hat{f}_{C2b}[m_1, m_2] = -\hat{f}_{C2mb}[m_1, m_2](-i)^{m_1-m_2} \cdot \text{mod}(m_1 + m_2 + 1, 2)$ .
3. Use  $h_a$  and  $h_b$  in equation 2 to filter the cross-talk regions and compute the weighting coefficients  $w_a$  and  $w_b$ . Obtain the  $C_2$  estimate  $\hat{f}_{C2}[m_1, m_2] = w[m_1, m_2] \cdot \hat{f}_{C2ma}[m_1, m_2] + (1 - w) \cdot \hat{f}_{C2mb}[m_1, m_2]$ .
4. Compute the baseband estimate of luma:  $\hat{f}_L[m_1, m_2] = f_{CFA}[m_1, m_2] - \hat{f}_{C1m}[m_1, m_2] - \hat{f}_{C2}[m_1, m_2]((-i)^{m_1+m_2} - (-i)^{m_1-m_2})$ .
5. Apply the linear transform described by the matrix  $M$  to obtain the RGB components of the image.
6. Up-sample to an image that is twice the size and lies on a rectangular lattice.

The luma component is calculated as a difference between the CFA and the previously computed modulated chromas. This way, the adaptive algorithm is taken into consideration (simply filtering out the luma would not deal with the interference caused by chroma 2).

The up-sampling is done using bicubic interpolation and this has proved to give satisfactory results. Other spatial interpolation methods, such as those presented in [1], may further optimize the demosaicking of the rotated Bayer structure, but since the focus is on the

extraction of the frequency-domain components, they were not considered in the analysis.

As it will be seen in the following sections, using well-designed filters for the luma and chroma components and also taking into consideration the adaptive nature of the algorithm presented above gives excellent demosaicking results on the considered training set.

## 6 Filter Design for $h_1$ , $h_2$ , $h_{3a}$ and $h_{3b}$

The adaptive scheme gives excellent frequency-domain demosaicking results. A good choice of the chroma filters should improve them.

The CFA signal that is being used lies on a hexagonal lattice. For ease of manipulation, this lattice is placed on a larger, rectangular lattice for which half of the points are 0 or undefined/NULL. When designing the filters for the chroma components of the image, this aspect must be taken into consideration. The filters designed have to be defined in the same space: on the hexagonal lattice.

Designing the chroma filters can be accomplished using many methods. The simplest ones that come to mind are Gaussian filters and filters designed with the window design method. Although these approaches offer a certain flexibility for the shape and modulation position of the filters, they do not necessarily output the optimum results. To improve the results, the filter design can be done following the least squares approach. This method is already used by Dubois in [10]. It uses a representative image training set to help model the desired filters such that the squared difference between the original and the reconstructed images is minimum over the training set.

Each image of the training set constructed in section 2 is considered the original. The luma and chroma filters are constructed such that the total signal difference between the demosaicked image and the original is minimum.

For each of the four signals,  $Q_{1-3b}$ , the quantity that needs to be minimized is the total squared difference between the original signal and its demodulated approximation

$$h_y = \operatorname{argmin}_h \sum_{i=1}^P \sum_{(m_1, m_2) \in W^{(l)}} (r_Y^{(l)}[m_1, m_2] - \sum_{(k_1, k_2) \in S} h[k_1, k_2] f_{\text{CFA}}^{(l)}[m_1 - k_1, m_2 - k_2])^2.$$

This expression is a least squares equation of the form

$$h_Y = \operatorname{argmin}_h \sum_{i=1}^P \|Z^{(i)}h - r_Y^{(i)}\|^2$$

that is known to have the solution:

$$\mathbf{h}_Y = \left[ \sum_{i=1}^P \mathbf{Z}^{(i)H} \mathbf{Z}^{(i)} \right]^{-1} \left[ \sum_{i=1}^P \mathbf{Z}^{(i)H} \mathbf{r}_Y^{(i)} \right]. \quad (3)$$

To apply the results in equation 3, the matrices describing the images need to be re-arranged in vector form. Once this step is done, MATLAB<sup>®</sup> is used to determine the least squares filters.

## 6.1 Design of the $h_2$ Filter

The  $h_2$  filter is used to extract the chroma 1 component. This is found on the horizontal and vertical axis, at the corner of the Voronoi cell at  $[\pm 0.5, 0]$  c/px and  $[0, \pm 0.5]$  c/px.

The idea is to compare the value of the chroma 1 component of the original image:

$$q_2[\mathbf{x}] = -\frac{1}{4}f_1[\mathbf{x}] + \frac{1}{2}f_2[\mathbf{x}] - \frac{1}{4}f_3[\mathbf{x}]$$

to that extracted from the CFA signal.

This is done on all the training set images in steps. The first step is to divide the image into sub-images. Then, let  $N_B = |S|$  be the number of filter coefficients that need to be determined and  $N_W$  be the number of samples in the sub-images. A  $N_B \times 1$  vector is formed by scanning the coefficients of  $h$  column by column, from left to right. A  $N_W$  vector is formed by scanning the elements of the sub-image in the same order. Finally, a  $N_W \times N_B = \mathbf{Z}^{(l)}$  matrix is formed from the elements of  $f_{\text{CFA}}$  as follows: each column of  $\mathbf{Z}^{(l)}$  corresponds to an element  $(k_1, k_2) \in S$  scanned in the same order. Each column of the  $\mathbf{Z}$  matrix corresponds to the CFA elements neighboring a fixed point, that would be modified by the chroma filter at convolution. After applying equation 3 and obtaining  $h$  in vectorial form all that is left to do is to re-arrange the coefficients of  $h_2$ .

For the chroma 1 filter we obtain the results similar to the example shown in figure 14.

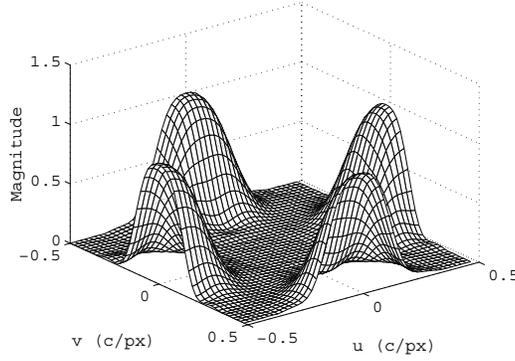


Figure 14: Frequency response of the  $h_2$  filter as designed with the least squares method.

## 6.2 Design of the $h_{3a}$ and $h_{3b}$ filters

There are two copies of the chroma 2 component in the frequency domain. To benefit from the adaptive component of the algorithm, the two filters extracting the  $Q_3$  component need to be designed simultaneously. This time  $h_{3a}$  and  $h_{3b}$  need to be optimized such that the error between  $q_3$  and  $\hat{q}_3$  is minimized. The chroma 2 component of the de-mosaicked image is written as:

$$\begin{aligned} \hat{q}_3[m_1, m_2] &= w[m_1, m_2](i)^{m_1+m_2} \times \sum_{(k_1, k_2) \in S} h_{3a}[k_1, k_2] f_{\text{CFA}}[m_1 - k_1, m_2 - k_2] \\ &\quad - (1 - w[m_1, m_2])(i)^{m_1-m_2} \times \sum_{(k_1, k_2) \in S} h_{3b}[k_1, k_2] f_{\text{CFA}}[m_1 - k_1, m_2 - k_2] \end{aligned}$$

The variables are plugged into MATLAB<sup>®</sup> as follows: First, two vectors of length  $N_B$  are formed by scanning every column of  $h_{3a}$  and  $h_{3b}$  from top to bottom and left to right and stacking them on top of each other. Then, a  $2N_B \times 1$  vector holding the coefficients of these two filters is formed. The column vector of  $q_3$  is obtained by scanning the elements of  $q_3$  in the same order:

$$\mathbf{h}_3 = \begin{bmatrix} \mathbf{h}_{3a} \\ \mathbf{h}_{3b} \end{bmatrix}.$$

Finally, a  $N_W \times 2N_B$  matrix  $W^{(l)}$  is formed as follows: the first  $N_B$  rows are formed by reshaping  $w^{(l)}[m_1, m_2](i)^{m_1+m_2} f_{\text{CFA}}^{(l)}[m_1 - k_1, m_2 - k_2]$  for each  $(k_1, k_2) \in S$  while the next  $N_B$  rows are formed by scanning  $-(1 - w^{(l)}[m_1, m_2])(i)^{m_1-m_2} f_{\text{CFA}}^{(l)}[m_1 - k_1, m_2 - k_2]$  in a similar manner.

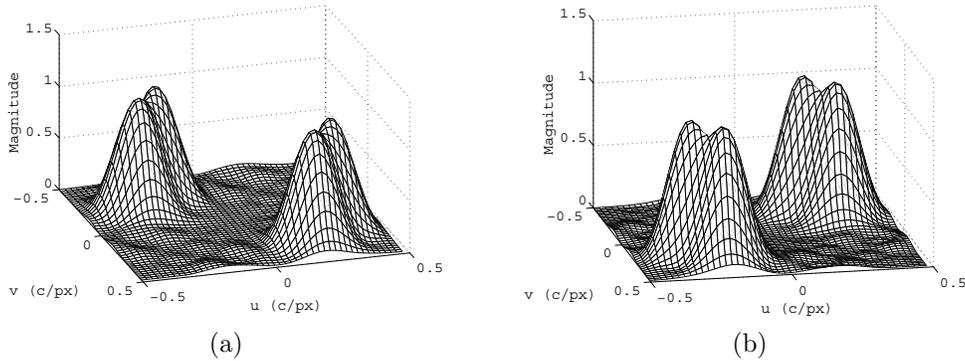


Figure 15: Frequency response of the  $h_{3a}$  and  $h_{3b}$  filters as designed with the least squares method.

This least squares problem can be written as:

$$\mathbf{h}_{3ab} = \operatorname{argmin}_h \sum_{l=1}^P \|\mathbf{W}^{(l)} \mathbf{h} - \mathbf{q}_3^{(l)}\|^2$$

with the solution:

$$\mathbf{h}_{3ab} = \left[ \sum_{l=1}^P \mathbf{W}^{(l)H} \mathbf{W}^{(l)} \right]^{-1} \left[ \sum_{l=1}^P \mathbf{W}^{(l)H} \mathbf{q}_3^{(l)} \right]$$

Reshaping  $h_{3ab}$  gives the two chroma 2 filters  $h_{3a}$  and  $h_{3b}$ .

## 7 Optimization of the Algorithm Performance

The LSLCD algorithm is relatively complex. Among the parameters that determine its performance are the filter sizes, the filter shape, the images chosen in the training set, etc.

The 24 Kodak images are considered representative for a set of natural images. Because the images considered for demosaicking are images that would normally be taken with a photo camera, the possibility of changing the filter training set type (to a set of artificially constructed images) will not be explored. The demosaicking filters are trained on all the training set images.

Optimizing all the algorithm parameters simultaneously would be a very long process. Moreover, because the training set is finite (and quite small) this kind of effort would be irrelevant. The parameters considered are optimized one by one while keeping everything else fixed. Depending on the observations gathered, this process may be performed several times.

The system parameters to be optimized are:

1. filter  $h_2$  of order  $M_1 \times M_2$ : the filter size is be optimized to give a maximum image quality while having the lowest possible complexity;
2. filters  $h_{3a}$  and  $h_{3b}$  of order  $N_1 \times N_2$ : the only parameter optimized is the filter size;
3. filters  $h_a$  and  $h_b$  used for the luma-chroma inference extraction:
  - (a) the middle modulation frequency,  $f_m$ ;
  - (b) the standard deviations;
  - (c) the filter size,  $P_1 \times P_2$ .

In general, larger filter sizes tend to improve visual quality. This is because the impulse response of a larger filter can be shaped more precisely to extract the luma and chroma components. However, using larger filters may increase the filter design and demosaicking time.

The size of the moving average filter  $h_m$  in equation 2, used for constructing the weighting coefficients will not be optimized. In the experiments, a size of  $\frac{11 \times 11}{2}$  was considered for this filter.

The CPSNR (Color Peak Signal-to-Noise Ratio) is the objective measure chosen to evaluate the image quality. The CPSNR is measured in decibels (dB) and is equal to 10 times the logarithm of the *Color Mean Squared Error* (color mean squared-error) between the original and the demosaicked image. In general, a larger CPSNR indicates less overall error and corresponds to a better visual quality. However, it is important to verify all the conclusions by visual inspection since a higher CPSNR does not necessarily imply a better subjective quality of the image.

## 7.1 Determination of Constants

With the exception of the Gaussian filters (used for the luma-chroma interference energy extraction), all the filters that are used in the LSLCD algorithm are modeled by training on an image set. Up until now, the Gaussian filter parameters and modulation frequency were considered constant. One would like to know if the choices made are good or if there is a need to change these parameters in order to obtain optimal results.

The Gaussian filters that are used are first designed with respect to the  $XY$  axes, then rotated by 45 degrees and modulated. During the experimental trials, it was observed

that diagonally oriented filters perform better. However, it is much easier to perform the optimization of parameters such as the filter size before the rotation.

The first thing that is optimized is the modulation frequency of these filters for the  $\frac{1024 \times 1536}{2}$  images. The  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  filter sizes are fixed at  $\frac{11 \times 11}{2}$ . Then, assuming that the best middle modulation frequency would be found on the  $\pm 45$  degrees diagonals, filters modulated at all the possible  $(a, a)$ ,  $(b, -b)$  combinations within the 0.01 to 0.17c/px interval with a step size of 0.01c/px are designed. The observations made in this interval suggest that the search be refined in the  $[0.0945, 0.0955]$ c/px interval with a step size of 0.0001c/px. The modulation frequency  $f_m$  is finally fixed at  $(0.0950, 0.0950)$  and  $(0.0950, -0.0950)$  respectively. The plot corresponding to this search can be seen in figure 16.

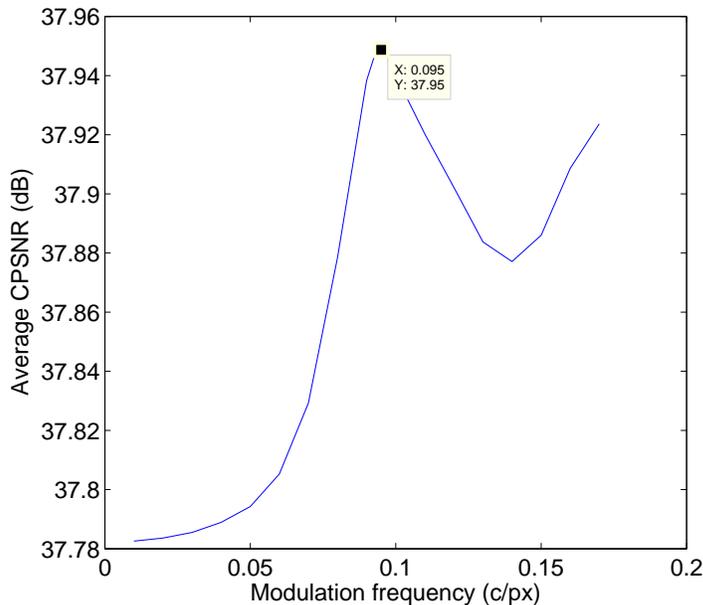


Figure 16: Algorithm performance as function of position of center frequency of the modulated energy extraction filters.

The second parameter that is optimized for the Gaussian filters is their standard deviations. The testing conditions are left the same and the modulation frequency is set to  $(0.0950, \pm 0.0950)$  as previously established. The standard deviation is fixed after trying all the  $(a, b)$   $(c, d)$  combinations with  $a$ ,  $b$ ,  $c$  and  $d$  going from 1.0 to 15.0 in steps of 1.0. The test conditions are similar: LSLCD designed luma and chroma filters having a size of  $\frac{11 \times 11}{2}$  are used. The results obtained by this brute-force optimization are very similar to what was obtained for the Bayer structure in [13]:  $r_{G1} = 3.0$  and  $r_{G2} = 1.0$ .

Finally, the Gaussian filter size is determined by trying all the  $\frac{a \times b}{2}$ ,  $\frac{a \times b}{2}$  combinations with  $a$  and  $b$  going from 4 to 20 in steps of two. During these experiments, the size for the  $h_2$ ,  $h_{3a}$

and  $h_{3b}$  filters is fixed at  $\frac{11 \times 11}{2}$ . The other Gaussian parameters are as determined above. For the larger image, the optimal Gaussian filter size is found to be  $12 \times 10$  before the filter rotation. Note that the horizontal size mentioned is equivalent to the actual filter size when these filters are rotated. The filter size before the rotation takes place is used because this ‘rectangle’ is much easier to visualize. Figure 17 shows how the algorithm performance varies with the Gaussian filter size.

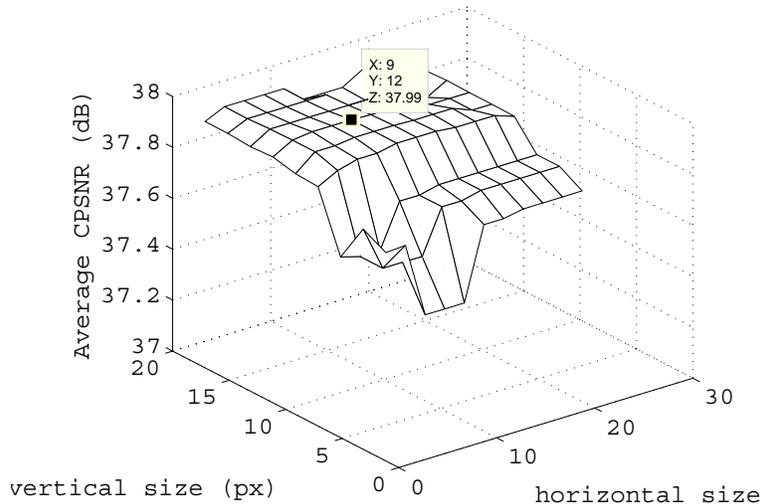


Figure 17: LSLCD algorithm performance as function of Gaussian filter size when  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  filter sizes are fixed.

The way the luma-chroma interference is filtered has a big impact on the final objective quality of the image. Although the Gaussian filters considered are diagonally oriented, their initial design is done with respect to the  $XY$  axes. This is easier from a computational point of view and has the additional advantage that, this way, the Gaussian filters can be considered separable in a naturally rectangular space such as MATLAB<sup>®</sup> and, thus, reduce the algorithm complexity.

Having fixed the Gaussian filter size, the next step is to find out which size do  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  need to have for the algorithm to be optimized. All the filter size combinations from  $\frac{5 \times 5}{2}$  to  $\frac{23 \times 23}{2}$  in steps of two are tried while the following constraints are imposed:

1. The filters are square. It was shown that for the Bayer structure demosaicking this type of filters give the best type of performance [13]. Having square filters has the advantage that they are symmetric in the two directions ( $XY$  axes or diagonals). The non-square filters can be seen as a particular case of the square filters in the sense that if the optimum filter size were rectangular the algorithm would produce it and would pad the rest of the area with zeros up until the filters became square.

2. The size of the  $h_{3a}$  and  $h_{3b}$  is the same. This makes the filter design process much easier and the optimization process shorter.

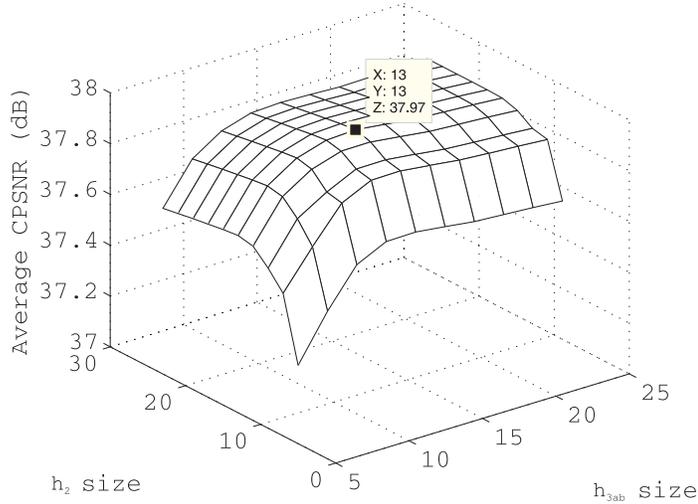


Figure 18: LSLCD algorithm performance as function of filter size of  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  when the energy extracting filters are fixed.

The algorithm performance does not change very much once the Gaussian filter parameters are determined. In fact, for all the  $(h_2, h_{3a}, h_{3b})$  combinations, the average CPSNR variations occur at the order of the 3<sup>rd</sup> or 4<sup>th</sup> decimal. This type of variation can be included in the algorithm error. Changing the selected training set would probably give a similar difference in the average CPSNR.

## 8 Objective Analysis of the Image Quality versus the Filter Complexity

The computational complexity of a demosaicking algorithm ultimately translates into demosaicking time and cost of hardware. One would like to have an algorithm that outputs a high quality image and uses very little resources. The LSLCD algorithm is comprised mostly of spatial filtering realized by convolution operations. The amount of computational work required depends on the signal that needs to be reconstructed and on the sizes of the filters used when performing this operation. Out of these two parameters, the only ones that can be controlled are the filter specifications.

This section, does not focus on optimizing the algorithm performance or the filter design but tries to find out what are the minimum filter sizes that can be used to obtain a good quality image.

Without loss of generality, the number of computations it takes to demosaic one pixel is used as a measure of algorithm complexity. The convolutions used in the LSLCD algorithm are in general 2-dimensional convolutions by an  $\frac{K_1 \times K_2}{2}$  filter. These take  $\frac{K_1 \cdot K_2}{2}$  multiplications per pixel. The divisions have a similar complexity as the multiplications. Additions are not counted in the number of computations because they take significantly less time to complete than multiplications.

By tracing the demosaicking steps in figure 13, the number of multiplications  $S$  required to demosaic each pixel is:

$$S = \frac{1}{2}(2(P_1 + P_2) + M_1 M_2 + 2N_1 N_2) + 5$$

where  $P_1 + P_2$  and  $P_2 + P_1$  are the orders of the (separable) Gaussian filters,  $h_2$  has the order  $M_1 \times M_2$  and the filters  $h_{3a}$  and  $h_{3b}$  have the order of  $N_1 \times N_2$  and  $N_2 \times N_1$  respectively. The other counted multiplications are due to the modulation/de-modulation and interpolation operations including the weighting of the chroma 2 coefficients.

Basing the algorithm on an observation that was done for the Bayer structure [12, 13], and because this is computationally easier on a system that is used to working with rectangular matrices, only square sizes are considered for the  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  filters in the filter kernel optimization. A greedy algorithm is used to explore the relationship between the objective demosaicking quality and computational complexity.

Let  $[M \ N \ P_1 \ P_2]$  represent the filter orders used at each iteration. The size of the  $h_2$  filter is  $M \times M$ , the one of the  $h_{3a}$  and  $h_{3b}$  filters is  $N \times N$  and the size of the two Gaussian filters used to extract the luma-chroma interference energies is  $P_1 \times P_2$ . The iteration begins at  $[21 \ 21 \ 21 \ 21]$  and ends with the final configuration  $[1 \ 1 \ 1 \ 1]$ . At each iteration, the algorithm creates four temporary configurations by decrementing each entry in the current configuration by two at a time, e.g.  $cf^{(1)} = [M - \Delta M \ N \ P_1 \ P_2]$ ,  $cf^{(2)} = [M \ N - \Delta N \ P_1 \ P_2]$ , etc. For each temporary configuration, the algorithm prepares four implementations of the demosaicking algorithm with the appropriate least squares and Gaussian filters that match it. Then, it uses this set of filters to perform demosaicking over a set of images. Finally, the greedy algorithm keeps the best of the four temporary configurations as the ‘current configuration’ for its next iteration.

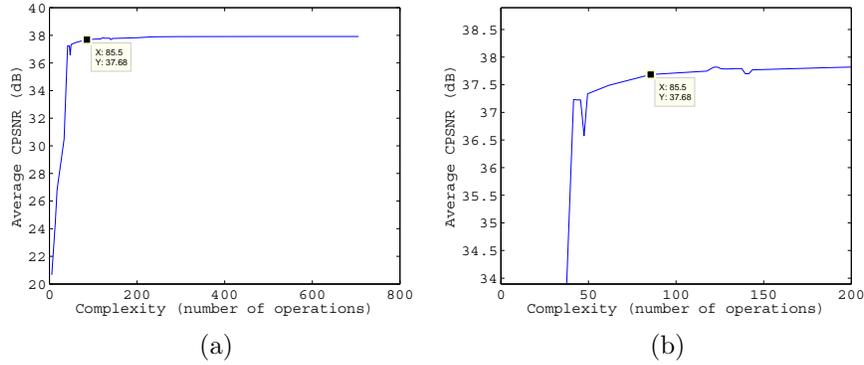


Figure 19: Demosaicking quality versus algorithm complexity (greedy) for the  $\frac{1024 \times 1536}{2}$  px images (left); Zoomed-in version of the same plot (right).

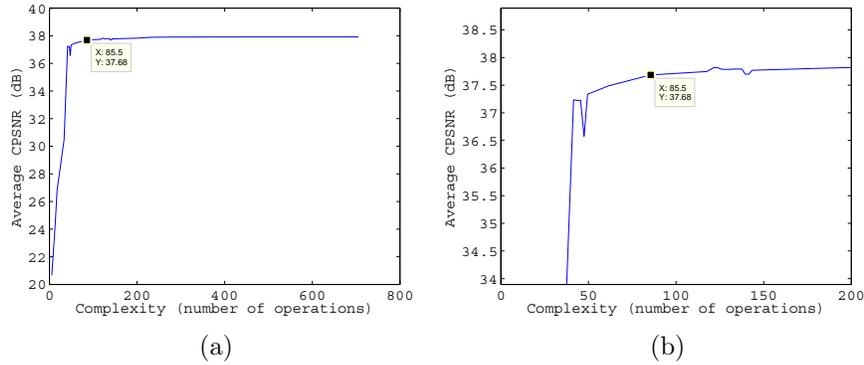


Figure 20: Demosaicking quality versus algorithm complexity (greedy) for the  $\frac{683 \times 1024}{2}$  px images (left); Zoomed-in version of the same plot (right).

The objective quality of the temporary configurations is determined by the CPSNR metric on the RGB color space. Specifically, averaging the PSNR values obtained for the red, green and blue channels gives the training set CPSNR. A higher CPSNR indicates a better objective quality. The data gathered when running the greedy algorithm is presented in Appendix C.

Figure 19 indicates that, for the larger images, a total filter complexity of around 85 may be optimum in what the algorithm speed is concerned. For all the higher complexity configurations, the improvement in demosaicking quality is too small considering the supplementary effort required. The table in Appendix C suggests that starting with the [7 7 9 1] configuration satisfactory results are obtained. For the smaller images, the minimum filter complexity should be 61. In the greedy algorithm output, this corresponds to the [7 9 13 3] configuration.

This sort of iteration allows testing the filter performance for a full range of filter complexities. However, it does not necessarily choose the optimal performance for each complexity level. Trying all the  $11^4$  configurations would take an immense amount of time and would also be impractical because the optimization would be done only on a small set of training images that are not necessarily the best representatives of natural images. Nonetheless, performing this type of algorithm gives a general idea about how long it would take to obtain a certain image quality.

## 8.1 Filter Complexity Reduction Techniques

Once the optimal filter size is decided upon, one can question if the number of computations can be further reduced. A complexity reduction technique that is worth considering is to create quadrantly symmetric filters. This technique can be applied on the pre-designed luma, chroma and Gaussian filters.

The quadrantly symmetric approximation of the least squares filters is created by averaging every set of four quadrantly related filter coefficients and then redistributing this average value to those four locations. Quadrantal symmetry generally reduces the number of required multiplications in the convolution by a factor of close to four. The number of multiplications required to demosaic each pixel is then:

$$S = \frac{1}{2}(2(P_1 + P_2) + \frac{M_1M_2}{4} + \frac{N_1N_2}{2}) + 5$$

The image quality that was obtained after demosaicking with quadrantly symmetric filters is not much different from what was previously obtained. The average CPSNR when demosaicking the 22 images was: 37.1670 dB compared to 37.9865 dB (for the larger images) when the LSLCD filters were used unmodified. For the smaller images, the average CPSNR was 35.9315 dB compared to 35.9558 dB when the original filters were used. The numerical results are listed at the end of the report in Appendix B.

The quadrantal symmetry is applied to the filters after their design is completed. Not only is this very easy to do at post-design time, but it does not unnecessarily complicate the filter design process. The filter performance of these newly designed filters is only slightly worse than that of the non-reduced ones.

Whether or not a complexity reduction technique was used, a trade-off between the objective image quality and the algorithm complexity was noticed.

## 9 Analysis of the Subjective Performance of the LSLCD Algorithm on the Rotated Bayer Structure

The rotated Bayer structure contains half the red and blue samples than it contains green samples. The green samples weight a lot in the luma component of the CFA signal. This way, the sampling structure approaches the way that the human visual system is constructed [2]. Moreover, the RBS structure privileges the high horizontal and vertical frequencies. This may prove to be an advantage when judging the subjective quality of the images because humans are more sensitive to horizontal and vertical lines than they are to diagonals [14]. Because of the aforementioned properties, it is possible that the rotated Bayer structure gives a better (subjective) image quality than other sampling structures when the objective demosaicking quality is the same.

In the experiments, the most bothersome demosaicking flaw noticed was the false colors. The reasons behind this phenomenon are various and they will be exemplified and explained in what follows. Because of the more restrained frequency interval, these effects are more visible when smaller size images are demosaicked.

False colors (blue-orange) were observed in relatively non-uniform picture areas where the luminance varies a lot such as the walls or the water. These are due to the chroma 1 - luma interference in the high luma frequencies and can be seen in figure 21.

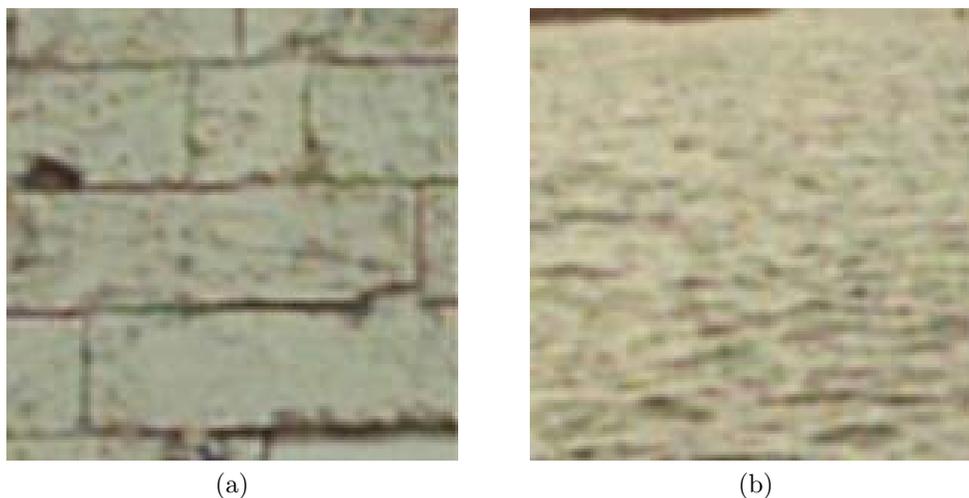


Figure 21: Chroma interference in the high luma frequencies.

On the diagonals, the chroma 1 and luma components are very hard to separate. As seen in figure 22, a blue-orange pattern appears on lines and patterns that are close to the  $\pm 45$  diagonals.



Figure 22: Chroma 1 - luma interference.

Probably the most bothersome false color effect is due to the luma - chroma 2 interference. A turquoise-magenta pattern appears on lines and patterns that are (almost) horizontal or vertical. For the smaller images, where the frequency interval is not very large, this effect greatly diminishes the image quality (figure 23). There may be a green-turquoise interference on diagonal lines as well — where the luma value varies a lot.

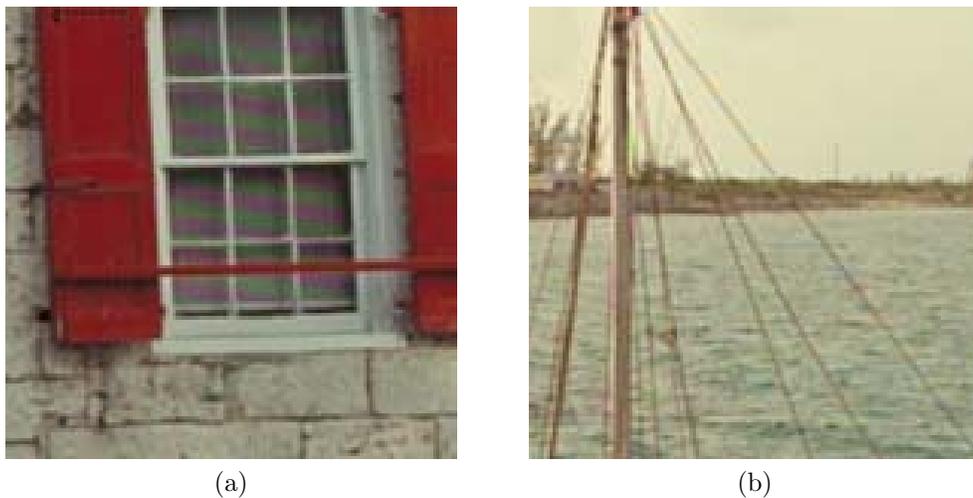


Figure 23: Chroma 2 - luma interference.

At the object edges or when there is a color change, the pattern pixelation becomes visible (one can tell which tristimulus value value was missing from a certain pixel). This is especially true for the bright red areas (figure 24) and it is due to the fact there are not enough red samples available to properly define the edges.



Figure 24: Imperfect reconstruction of edges, color leak.

Comparison with images that have been sampled differently and demosaicked using the LSLCD algorithm is rather difficult because of the extra interpolation step taken to finalize the demosaicking of the RBS structure. However, the images that were obtained seem to have a better quality than the images that were sampled on the diagonal stripe pattern. Whether or not this structure performs better than the Bayer structure is debatable and depends on the picture details (density of horizontal and/or diagonal frequencies).

When it comes to the way the filter sizes influence the image quality, quadrantly symmetric filters have reduced complexity and perform almost as well as the filters they were derived from. No difference is noticeable with the naked eye.

## 10 Recommendation

Designing the appropriate LSLCD filters for the rotated Bayer structure is a relatively complex process. However, once all the filter parameters are optimized, the algorithm gives very good results even when the filter complexity is not too high.

To obtain a good picture quality while keeping a relatively low complexity, the filters should be set as follows:

1. The  $h_2$  filter, filtering the chroma 1 component should be a  $\frac{13 \times 13}{2}$  filter;
2. The  $h_{3a}$  and  $h_{3b}$  filters, filtering the two chroma 2 components, should have the size  $\frac{13 \times 13}{2}$ ;

3. For the adaptive part of the algorithm one should use  $\frac{12 \times 10}{2}$  Gaussian filters rotated by 45 or  $-45$  degrees and modulated at  $(0.0950, 0.0950)$  c/px and  $(0.0950, -0.0950)$  c/px respectively. The Gaussian filter specifications are  $r_{G1} = 3$  and  $r_{G2} = 1$ .

If the algorithm complexity needs to be reduced, the first thing that needs to be considered is to profit from the quadrantal symmetry. The greedy algorithm analysis that was performed suggests that if there is need for further complexity reduction it is best to first reduce the filter order of the  $h_2$ ,  $h_{3a}$ , and  $h_{3b}$  filters before changing any of the Gaussian filter parameters.

## 11 Conclusion

This report addresses the problem of demosaicking the hexagonally sampled rotated Bayer structure. For better comparison with the literature, the test images used are down-sampled versions of the well-known full color Kodak images during the testing. The custom-built rotated Bayer CFA signals are derived from those images.

An important observation to be made is that frequency domain based demosaicking offers a better demosaicking quality on the rotated Bayer structure than the spatial demosaicking techniques. The adaptive frequency-domain algorithm using filters designed by the least squares method gives excellent demosaicking quality on the images considered.

The adaptive Least-Squares Luma-Chroma Demosaicking filters were optimized in order to obtain a high demosaicking quality. Moreover, the iterative optimization procedure confirmed that the demosaicking quality and complexity of algorithm form a trade-off relationship and pointed to the most computationally efficient demosaicking configuration that would also output a high quality demosaicked image.

Extensive results were obtained for the LSLCD of the rotated Bayer structure. These results and the software used to produce them are available in [11].

The rotated Bayer sensor model that was used is only a rough approximation of the actual sensor. Future work would involve taking into consideration a more realistic modeling of the octagonal shape of the sensor elements. The analysis of other members of the SuperCCD sensor family is also an open future direction of research.

# A Anti-Aliasing Filters and Down-Sampled Image Quality

When down-sampling a digital image, the frequency domain of the resulting output has a much smaller unit cell than the unit cell of the original image. To avoid aliasing effects due to this unit cell ‘contraction’, an anti-aliasing filter must be applied to the original image before its samples are thrown away.

This appendix is dedicated to the anti-aliasing filter needed to downsample a  $2048 \times 3072$  rectangularly sampled image to two smaller images lying on a hexagonal lattice: one of  $\frac{1024 \times 1536}{2}$  pixels, the other having a size of  $\frac{683 \times 1024}{2}$  pixels (they lie on a larger rectangular lattice of  $1024 \times 1536$  px and  $683 \times 1024$  px, respectively).

The pass-band of the anti-aliasing filter has to be in the Voronoi cell of the down-sampled image. The Voronoi cell of the  $\frac{1024 \times 1536}{2}$  pixel image is diamond shaped. The diamond has a diagonal of  $0.5$  c/px. The unit cell of the smaller down-sampled image is also diamond shaped, having a diagonal of  $0.3333$  c/px. The parameters that will be taken into consideration are the transition band width and its position, and whether or not the high horizontal and vertical frequencies affect the image quality.

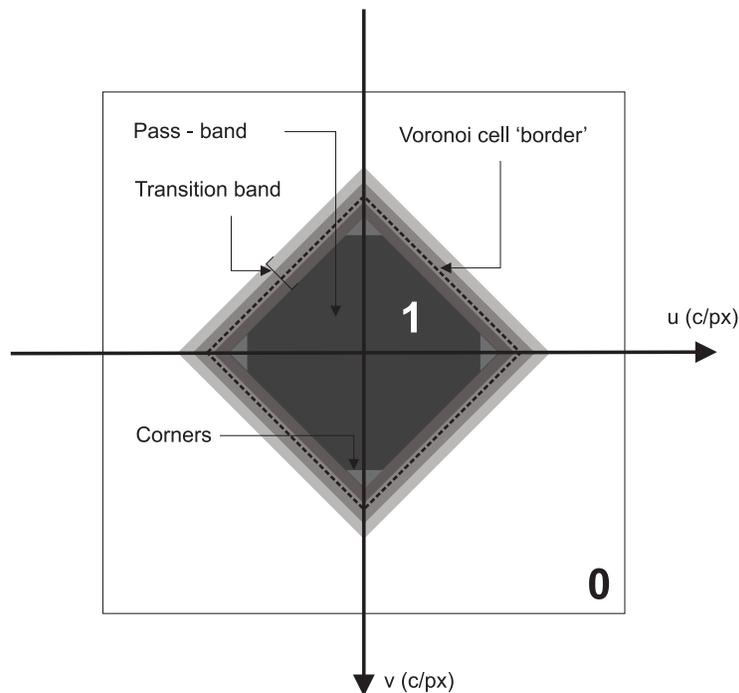


Figure 25: Anti-aliasing filter parameters.

For the empirical experiments, the Kodak image number 8, the Alfred image, was chosen. This image has a lot of different details and, because of this, was appropriate for the anti-aliasing observations to be made.

## A.1 Effect of the Transition Band Width on the Down-Sampled Image

A narrow transition band helps obtain the desired effect in frequency domain (a unit cell that has the size very close to the ideal one). However, this may cause rippling in space. A wide transition band may over-filter the small frequencies and thus cause the image to be blurry when looked at. The figure below shows details of the Alfred image when filtered with different transition band widths centered at the Voronoi cell border.

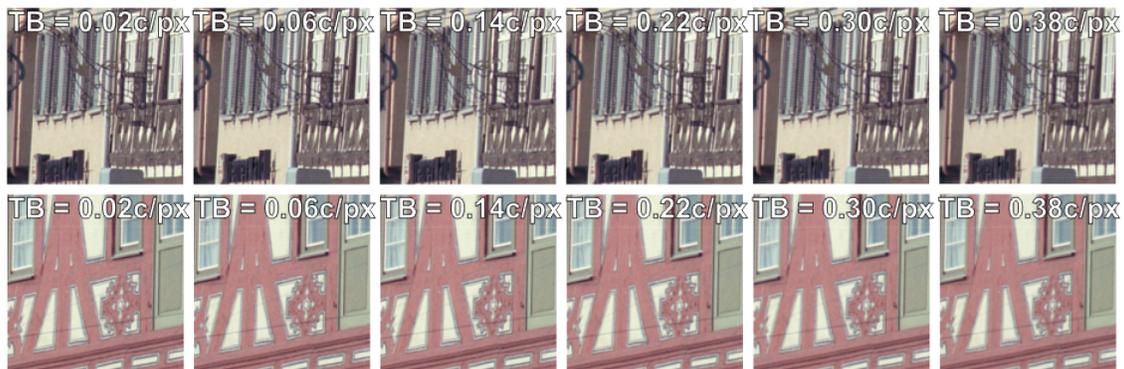


Figure 26: Alfred picture details for the transition band width of the anti-aliasing filter used for the  $\frac{1024 \times 1536}{2}$  image.

A transition band having the width of approximately  $0.10 c/px$  would be optimal for the larger down-sampled image. The same way, a transition band width of  $0.08 c/px$  would suit the smaller down-sampled image best.

## A.2 Effect of the Transition Band Position on the Down-Sampled Images

Deciding the transition band width should address the question of rippling in space. Where this transition band is located will influence the degree of blurriness (not enough frequencies) or aliasing (high frequency coefficients overlapping the low frequency ones).

Figure 27 illustrates the effect of the transition band position on the down-sampled images.



Figure 27: Alfred picture details for the transition band position of the anti-aliasing filter used for the  $1024 \times 1536$  image. The width is of  $10c/px$ .

The image having the transition band centered at  $0.15c/px$  seems to have the best subjective quality for the large down-sampled image (transition band is going from  $0.15$  to  $0.25c/px$ —the Voronoi cell border) and the anti-aliasing filter is fully comprised in the unit cell. In the case of the smaller down-sampled image, it seems wiser not to move the center of the transition band (have it going from  $0.12667$  to  $0.20667c/px$ ).

An aspect that should be taken into consideration is that these images are twice as big as the down-sampled images that are needed. The fine tuning done on this bigger images may not work as well on the small ones. However, because the image frequency range is the same, the observations should hold.

### A.3 Effect of the Filter Size on the Down-Sampled Image



Figure 28: Alfred picture details for the size of the anti-aliasing filter used for the  $1024 \times 1536$  image.

The larger the filter size is, the better the image is filtered (details are fine tuned). However, after a certain point, the difference in the subjective quality of the image obtained is below the human perception threshold. Going beyond that filter size would only increase the computational time needed to filter the images. The images in figure 28 show the effect of different filter sizes on the Alfred image.

Observation shows that a filter size of  $25 \times 25$  can be used for the larger image and a filter size of  $21 \times 21$  for the smaller one.

#### A.4 Effect of the Diamond-Shape Corners on the Down-Sampled Image

No improvement if any was observed when the high horizontal and vertical frequencies of the down-sampled images were pre-filtered more.

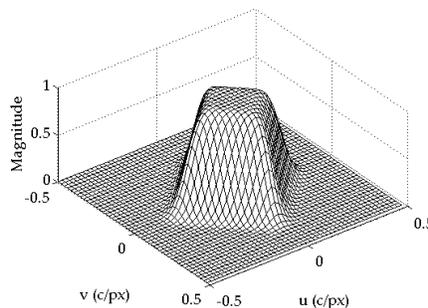


Figure 29: Frequency response of the anti-aliasing filter for the  $\frac{1024 \times 1536}{2}$  image.

For the  $\frac{1024 \times 1536}{2}$  down-sampled image, the filter that was chosen has the transition band width of  $0.10$  c/px, centered at the Voronoi cell border. The filter size that gave the best quality vs complexity performance was a  $25 \times 25$  filter. For the smaller image, the anti-aliasing filter chosen was a diamond shaped filter having a transition band of  $0.08$  c/px, centered at the Voronoi cell border. In this case, the filter size that gave the best results was  $21 \times 21$ .

## B Numerical Results

1. Bilinear: Bilinear interpolation from the CFA signal to a fully colored hexagonal lattice followed by bilinear up-sampling to a fully colored, rectangularly sampled image that is twice the size.
2. Bicubic: Bilinear interpolation from the CFA signal to a fully colored hexagonal lattice followed by bicubic up-sampling to a fully colored, rectangularly sampled image that is twice the size.
3. LSLCD: Adaptive LSLCD with optimized filters
4. Bayer: Demosaicking of 45 degrees rotated images using the LSLCD filters recommended for the Bayer structure [3] followed by up-sampling of to a rectangularly sampled image.

The picture frame is the set of pixels that are saved at the image edge and that are not part of the image. These are usually gray or have very intense colors. The picture frame is not considered in the filter design and is not taken into consideration when computing the algorithm objective quality.

The frame size considered when computing the CPSNR performance on each of the images is equal to the size of the largest filter used in the demosaicking algorithm i.e. the maximum size between the sizes of  $h_2$ ,  $h_{3a}$  and  $h_{3b}$  filters. For the bilinear column and the bicubic column the frame is considered to have 11 pixels.

Table 1: Demosaicking results for the rotated Bayer structure -  $\frac{1024 \times 1536}{2}$  images

Image	Bilinear	Bicubic	LSLCD	Quadrantal	Bayer
1	29.0257	29.3556	35.7755	35.5411	35.7641
2	33.9313	34.3244	34.6903	34.4164	34.2040
3	38.0995	38.4781	41.9793	40.4301	41.5521
4	35.7440	36.1647	38.2757	37.7515	37.5057
5	30.8792	31.3947	37.3353	35.1008	36.9209
6	31.3768	31.7119	37.7791	36.9774	37.8621
7	36.9954	37.4811	41.4458	39.9242	40.9190
8	27.8359	28.2007	33.1569	32.6447	33.2946
10	37.1578	37.5811	40.9391	40.3063	40.5966
11	31.8762	32.2029	36.9791	36.1633	36.9288

Demosaicking results for the rotated Bayer structure -  $\frac{1024 \times 1536}{2}$  images (cont.)

Image	Bilinear	Bicubic	LSLCD	Quadrantal	Bayer
12	38.4232	38.8085	43.5510	42.6721	43.1742
13	27.5777	27.8492	32.5701	32.0342	32.5497
14	32.4616	32.8552	36.2502	35.2011	35.9307
15	36.3346	36.6817	39.4469	38.7128	38.9883
16	32.9722	33.2401	38.5091	38.2211	38.7264
17	35.0808	35.3792	38.9319	37.8505	38.8320
18	30.8503	31.1578	34.1922	33.6369	34.0406
19	32.3208	32.7255	37.4232	36.8817	37.3686
20	35.0601	35.4085	39.5934	38.5268	39.4572
21	31.6535	31.9811	36.9288	36.3345	36.9909
23	39.2863	39.7862	42.2507	41.2840	41.6760
24	31.7668	32.2093	37.6705	37.0633	37.7960

Table 2: Demosaicking results for the rotated Bayer structure -  $\frac{683 \times 1024}{2}$  images

Image	Bilinear	Bicubic	LSLCD	Quadrantal	Bayer
1	27.0560	27.2997	33.2000	33.1645	33.5156
2	30.8898	31.0865	32.8074	32.7919	32.6371
3	33.5519	33.6404	39.2475	39.1625	39.5155
4	32.1346	32.3054	35.7667	35.7548	35.2627
5	27.3678	27.6793	33.0932	33.0442	33.5295
6	29.1014	29.2836	35.6480	35.6240	35.9800
7	32.3786	32.6243	38.3826	38.3822	38.3448
8	25.9593	26.2318	31.9278	31.8960	32.2928
10	33.0073	33.1434	39.6834	39.6034	39.7579
11	29.6975	29.8769	35.0613	35.0416	35.2532
12	33.6087	33.6644	41.0475	41.0029	41.0117
13	26.4125	26.5951	31.5262	31.5062	31.6288
14	29.4722	29.7297	33.1919	33.1620	33.2161
15	32.1030	32.2722	36.6649	36.6487	36.6114
16	30.8916	31.0122	37.4668	37.4396	37.7522
17	31.8138	31.9889	38.2850	38.2757	38.6477
18	29.0839	29.3217	33.3654	33.4036	33.4032
19	29.5370	29.7540	36.0014	35.9748	36.3073
20	31.2063	31.2969	38.1445	38.1352	
21	29.3701	29.5568	35.7396	35.7481	35.9688
23	34.4657	34.6428	39.7283	39.7079	39.7374
24	28.3494	28.5017	35.0480	35.0229	35.3020

## C Greedy Algorithm Results

Table 3: Greedy iterations for the rotated Bayer structure -  $\frac{1024 \times 1536}{2}$  px image

Configuration	Image quality CPSNR
[21 21 21 21]	37.9144
[19 21 21 21]	37.9143
[17 21 21 21]	37.9139
[15 21 21 21]	37.9139
[15 19 21 21]	37.9120
[15 17 21 21]	37.9102
[15 15 21 21]	37.9089
[15 13 21 21]	37.9070
[13 13 21 21]	37.9047
[13 11 21 21]	37.8943
[11 11 21 21]	37.8774
[11 11 21 19]	37.8593
[11 11 21 17]	37.8593
[11 11 21 15]	37.8593
[ 9 11 21 15]	37.8193
[ 7 11 21 15]	37.8062
[ 7 9 21 15]	37.7692
[ 7 9 21 13]	37.6995
[ 7 9 21 11]	37.6995
[ 7 9 21 9]	37.7913
[ 7 9 21 7]	37.7913
[ 7 9 21 5]	37.7912
[ 7 9 21 3]	37.7871
[ 7 9 19 3]	37.7859
[ 7 9 17 3]	37.7859
[ 7 9 15 3]	37.7913
[ 7 9 13 3]	37.8188
[ 7 9 11 3]	37.8178
[ 7 9 9 3]	37.7827
[ 7 9 9 1]	37.7464
[ 7 7 9 1]	37.6845
[ 7 5 9 1]	37.4881
[ 5 5 9 1]	37.3358
[ 5 5 7 1]	36.5761
[ 5 5 5 1]	37.2249
[ 5 5 3 1]	37.2249
[ 5 5 1 1]	37.2332

Greedy iterations for the rotated Bayer structure -  $\frac{1024 \times 1536}{2}$  px image (cont.)

Configuration	Image quality CPSNR
[ 3 5 1 1]	30.4808
[ 3 3 1 1]	26.8223
[ 1 3 1 1]	24.4291
[ 1 1 1 1]	20.6578

Table 4: Greedy iterations for the rotated Bayer structure -  $\frac{683 \times 1024}{2}$  px image

Configuration	Image quality CPSNR
[21 21 21 21]	36.0440
[19 21 21 21]	36.0439
[17 21 21 21]	36.0440
[15 21 21 21]	36.0442
[15 19 21 21]	36.0430
[15 17 21 21]	36.0419
[15 15 21 21]	36.0406
[13 15 21 21]	36.0389
[13 13 21 21]	36.0363
[13 13 21 19]	36.0323
[13 13 21 17]	36.0323
[13 13 21 15]	36.0323
[13 11 21 15]	36.0243
[11 11 21 15]	36.0154
[9 11 21 15]	35.9988
[7 11 21 15]	35.9987
[7 9 21 15]	35.9702
[7 9 21 13]	35.9257
[7 9 21 11]	35.9257
[7 9 21 9]	35.9818
[7 9 21 7]	35.9818
[7 9 21 5]	35.9816
[7 9 21 3]	35.9733
[7 9 19 3]	35.9720
[7 9 17 3]	35.9719
[7 9 15 3]	35.9890
[7 9 13 3]	36.0224
[7 9 11 3]	36.0222
[7 9 9 3]	36.0092
[7 7 9 3]	35.9622
[7 7 9 1]	35.9133
[7 5 9 1]	35.7408

Greedy iterations for the rotated Bayer structure -  $\frac{683 \times 1024}{2}$  px image (cont.)

Configuration	Image quality CPSNR
[5 5 9 1]	35.5427
[5 5 7 1]	34.8299
[5 5 5 1]	35.3900
[5 5 3 1]	35.3900
[5 5 1 1]	35.3985
[3 5 1 1]	29.8092
[1 5 1 1]	26.4200
[1 3 1 1]	24.2389
[1 1 1 1]	20.6083

## Appendix: Glossary of Terms

**Bayer structure** Sampling structure comprised of red, green and blue samples that lie on a rectangular lattice. The green samples are twice as dense as the red or blue ones and lie on a checker-board pattern. The red and blue samples alternate in such that the distance between two samples of the same color is always larger than 3 units. 1, 3, 8, 15, 23, 24, 26, 37

**CFA** Color Filter Array. Describes the way that different color filters are positioned on the photo-sensitive sensor of a digital camera. 1–8, 10, 11, 13, 17–19, 29, 32, 37

**Color Mean Squared Error** The average of the square of the differences between the original and the estimated values. 22

**demosaicking** Algorithm used to construct a full color image from the subsamples provided by a camera sensor. 1–3, 7, 11, 13, 15–18, 21, 22, 24–28, 32, 37

**lattice** In two-dimensions is the set of all linear combinations, with integer coefficients, of two linearly independent vectors in  $\mathbb{R}^2$ . ii, 1–9, 17, 18, 33, 37

**LSLCD** Least-Squares Luma-Chroma De-multiplexing. Name given to the filter design method using the least squares approach. 2, 21–23, 25, 26, 28, 31, 37

**tristimulus value** Value of the coefficient of the primary color when added to form the desired color. 1, 8, 30

**Voronoi cell** In frequency domain, the geometrical space of the points that are closer to the sample situated at the origin than they are to any other sample. 3–5, 10, 19, 33–36

**windowing** MATLAB<sup>®</sup> filter design method in which the user specifies the ideal impulse response of a signal filter in the frequency domain. The method calculates the inverse Fourier transform of the specifications and then truncates it to a finite number of coefficients using the ‘windowing’ function. 4

## References

- [1] B. Gunturk, J. Glotzbach, Y. Altunbasak, R. Schafer, and R. Mersereau, “Demosaicking: Color Filter Array Interpolation,” *IEEE Signal Process. Mag.*, vol. 22, pp. 44–54, January 2005.
- [2] D. Alleysson, S. Süsstrunk, and J. Héroult, “Linear demosaicking inspired by the human visual system,” *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, April 2005.
- [3] E. Dubois, “Frequency-domain methods for demosaicking of Bayer-sampled color image,” *IEEE Signal Process. Lett.*, vol. 12, pp. 847–850, December 2005.
- [4] ———, “Filter design for adaptive frequency-domain Bayer demosaicking,” in *2006 IEEE International Conference on Image Processing (ICIP)*, October 2006, pp. 2705–2708.
- [5] B. C. de Lavarene, D. Alleysson, and J. Héroult, “Practical implementation of LMMSE demosaicking using luminance and chrominance spaces,” *Computer Vision and Image Understanding*, vol. 107, pp. 3–13, July–August 2007.
- [6] E. Dubois. (2005) Frequency Domain Methods for Demosaicking of Bayer-Sampled Color Images: Additional Results. [Online]. Available: <http://www.site.uottawa.ca/~edubois/demosaicking/>
- [7] Digital Photography Review. New honeycomb ‘Super CCD’ from FujiFilm. [Online]. Available: <http://www.dpreview.com/news/9910/99102003superccd.asp>
- [8] C. Smith. Fujifilm FinePix F40fd Review. [Online]. Available: <http://www.trustedreviews.com/digital-cameras/review/2007/06/05/Fujifilm-FinePix-F40fd/p3>
- [9] S. Joinson. Compact Camera High ISO modes: Separating the facts from the hype. [Online]. Available: <http://www.dpreview.com/articles/compactcamerahighiso/>
- [10] E. Dubois, “Color filter array sampling of color images: Frequency-domain analysis and associated demosaicking algorithms,” in *Single Sensor Imaging: Methods and Applications for Digital Cameras*, R. Lucak, Ed. Boca Raton, FL: CRC Press, 2009, ch. 7, pp. 183–212.
- [11] E. Dubois and M. Urlea. (2010) Color-filter-array Sampling of Color Images: Frequency-domain Analysis and Associated Demosaicking Algorithms: Additional Results. [Online]. Available: <http://www.site.uottawa.ca/~edubois/SingleSensorImaging/>

- [12] B. Leung, “System Optimization and Improvements to Adaptive Frequency-Domain Bayer Demosaicking,” no. NAV-TR-2006-02, Navire Internal Report, School of Information Technology and Engineering, University of Ottawa, 2006.
- [13] B. Leung, E. Dubois, and G. Jeon, “Least-Squares Luma-Chroma Demultiplexing Algorithm for Bayer Demosaicking,” *IEEE Trans. Image Process.*, 2011, in pres.
- [14] A. Watanabe, T. Mori, S. Nagata, and K. Hiwatashi, “Spatial sine-wave responses of the human visual system,” *Vision Research*, vol. 8, pp. 1245–1263, September 1968.

## Statement of Approval by Supervisor

As supervisor of co-op student **Maria Urlea**, I, **Eric Dubois**, certify that, to the best of my knowledge, this report is entirely the student's work and is free of confidential information to the extent that it can be read by university faculty members.

Signature \_\_\_\_\_

Date \_\_\_\_\_