Neurocomputing 83 (2012) 222–228

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Classification-based de-mosaicing for digital cameras

Amin Ur Rehman, Ling Shao *

Department of Electronic & Electrical Engineering, The University of Sheffield, UK

ARTICLE INFO

Article history:
Received 18 July 2011
Received in revised form
19 December 2011
Accepted 26 December 2011
Communicated by Y. Yuan
Available online 13 January 2012

Keywords:
Bayer pattern
Demosaic
CFA array
RGB
Classification
Trained filter

ABSTRACT

In this paper, we propose a content adaptive demosaicing algorithm, utilising content analysis and correlation between the red, green and blue planes of a particular image. These two aspects are used for the classification of the technique in the generated trained filters. The proposed method aims to reconstruct a high quality demosaiced image from a CFA Bayer pattern. The strategy highlighted in this paper is very effective, as many of the image details are maintained during reconstruction. Since image content analysis and filter coefficient optimisation are performed during training and the training process is offline, the online de-mosaicing filter is very efficient.

1. Introduction

Digital cameras employ the usage of a CFA (Colour Filter Array) Bayer pattern. In order to fully utilise the chip design in digital cameras, just one array is used to capture pixels of the red plane, the green plane and the blue plane. Due to the higher sensitivity of the human vision system towards the green colour, the green pixels constitute 50 percent of the entire array, whereas the red and blue pixels constitute 25 percent each of the CFA. To fully reconstruct a coloured image, there is a requirement to compute the full resolutions of the red, blue and green planes. With the increase in the demand of high definition images, it has now become imperative that the reconstruction of these images is in very high quality. It is now very common to have digital cameras providing options of taking pictures with a resolution much higher than full HD. There is therefore a need to obtain the full resolution red, blue and green planes. A strategy such as linear interpolation or bi-cubic interpolation would result in a very low quality reconstruction. Such a strategy would cause a great deal of blurring and aliasing in the detailed regions of an image. Therefore, it is necessary to use more advanced methods to perform a high quality reconstruction of these planes.

Content adaptive demosaicing is therefore the need of the hour and due to its high demand, there have been methods proposed in this field. The majority of the content adaptive techniques deal with an edge detection mechanism as shown in Hamilton's method [1], where interpolation is performed along the edges of an image. The technique in [2] relates the concept of bilateral filtering to demosaicing. Bilateral filtering is used to cause a similarity measure between pixels. It is explained by an example wherein a Gaussian noise corrupted image undergoes a block blur operation. The image then undergoes a similarity kernel which, when combined with the block blur kernel, results in a bilateral filter kernel. The idea is then extended for a demosaicing process. The missing pixels for the green plane are computed by means of linear interpolation. The interpolated green pixels are then used to evaluate a bilateral kernel, with blurring and similarity measures. The bilateral kernel can then be applied to the red and blue planes, respectively to compute the missing pixels. The results seem to be much better than linear interpolation, however formation of the similarity kernel is crucial and is highly dependent on the similarity standard deviation. Also, any error occurrence in the evaluation of the complete green plane would propagate to the other two planes.

Menon et al. [3] proposed a very unique technique to maintain details after the demosaicing process. It is evident from the Bayer pattern that at pixel points where a green pixel is not present, either a blue or a red pixel is present. In the proposed method, the green plane is completed by means of linear interpolation. The pixel points at which an interpolated green pixel is inserted, the high frequency components of the pixel are replaced by the high frequency components of the respective original red/blue pixel. By this way, the green plane is "fine tuned". The red and blue pixels are interpolated by means of the similar interpolation of the R–G or the B–G difference. This results in all the three planes

* Corresponding author.
E-mail address: ling.shao@sheffield.ac.uk (L. Shao).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.12.025

being very high quality, although again suffers from the problem of error propagation if there is an erroneous calculation of the green plane.

A very similar technique was developed in [4]. It follows a calculation of the Jacobian at every point, and the complete green plane is constructed by means of a "weighted" horizontal and vertical interpolation. The weights (referred to as the voting parameters) of the interpolation are dependent on the Jacobian, in other terms, the edge details of the image. A linearly interpolated R–G (or B–G) difference frame is evaluated. This difference frame, when subtracted with the interpolated green plane, would result in interpolated red values to be retrieved, alongside the original red pixel values, hence enabling demosaicing. The results seem to be free from any sort of aliasing errors and computationally, it claims, to be less complex than most adaptive demosaicing techniques.

Out of the many prevalent techniques in demosaicing, the majority of them rely on the edge detection mechanism. Besides Hamilton's method [1], the algorithms in [5–9] all deal with this mechanism, but how the edges are detected varies. In [1], horizontal and vertical indicators for edge detection are proposed. Interpolation is then performed along the directions hence pointed to by the edge indicators. The method in [5] uses a weighted edge scheme, where weights of interpolation are dependent on the edges. The authors of [6] proposed 12 directional feature vectors and the green plane is interpolated and forms the basis of the red and blue planes.

Peiandi and Tam [10] proposed a colour difference model, Kr (difference between green and red planes) and Kb (difference between green and blue planes), and this new plane configuration is interpolated. A green plane interpolation technique in which the interpolation weights are based on the values of red and blue plane was presented in [11]. Correspondingly, a complete green plane assists in the interpolation of the red and blue planes, as the interpolated values in the green plane adjust the weights for the interpolation of the red and blue planes. Of most of the contemporary techniques in adaptive demosaicing, [12] is a variant strategy in which the power spectral density of the Bayer pattern is evaluated. Upon low pass filtering, the Y (luminous plane) is obtained and band pass filtering produces the U and V (chrominance) planes. It is noteworthy though that most of the prevalent techniques focus on a very precise construction of the green plane which forms the basis for the interpolation of the red and blue planes.

Filter coefficients of the existing content adaptive techniques are dependent on the quality of edge detection and are heuristically designed according to the edge orientation and strength. In this paper, we present a demosaicing method using optimised filters based on a training process and well-defined content classification. The classification-based trained filters have been successfully applied in many restoration and enhancement applications, such as resolution up-conversion [13,14], compression artifacts removal [13,15], and repairing low quality enhancement units [16].

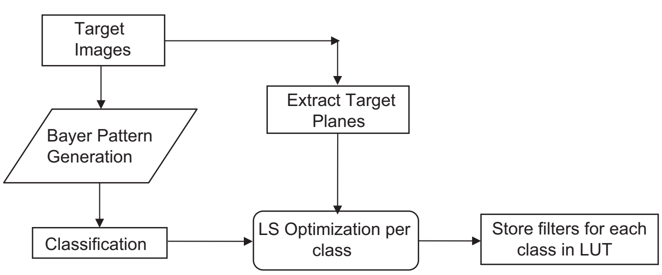


Fig. 1. Offline training algorithm.

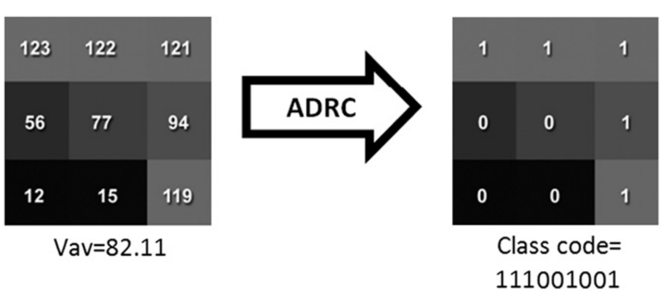


Fig. 2. Adaptive dynamic range coding.

2. Proposed technique

The proposed technique consists of an offline (training) part of the algorithm and an online part. The offline part has to undergo training on an extensive set of images to compose a comprehensive lookup table (LUT) which would cater for different contents in a particular image.

The emphasis in the paper is put on reconstructing a full resolution high quality green plane. This is because a very high quality green plane would contribute more to the overall visual acceptability of the demosaiced image as compared to the red and blue planes. Due to the high sensitivity of the human eye towards the green plane, the focus is to reconstruct the green plane as accurately as possible.

2.1. *Offline training*

Fig. 1 shows the training part of the proposed method. As a first step, the offline algorithm "degrades" (sub-sample) each image in to a GRBG Bayer pattern.

So in effect, a mosaiced image is available to the algorithm as well as the original green plane. The next step is to determine the class index for each pixel location in the green plane according to a classification method. All the pixels belonging to a particular class and their corresponding pixels in the original images are accumulated and the least-squares optimisation is conducted to produce filter coefficients for that class. The optimised filter coefficients for different classes are stored in a LUT for use during the demosaicing process.

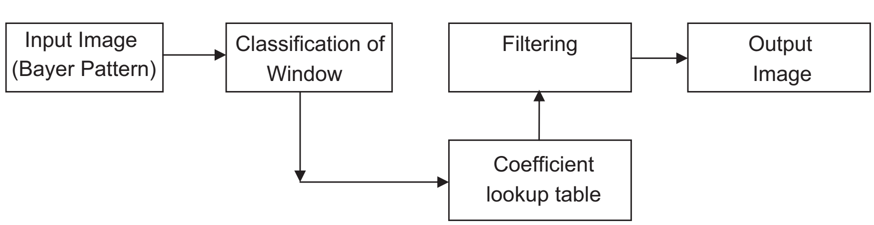


Fig. 3. Online de-mosaicing process.

Pixels and their neighbourhoods in a mosaiced image can be classified. By using their corresponding target (original) pixels, the optimal coefficients can be obtained by means of a Least Squares minimisation. The classification would attempt to treat similar image textures and contents as similar classes, as indicated in [13].

Suppose that

$F_{D,C}(i,j)$: are the apertures of the source images (degraded, in our case a mosaiced image) and $F_{F,C}(j)$: are the corresponding pixels in the target images for a particular class. $F_{F,C}(j)$, which are the filtered pixels, can be obtained by the desired optimal coefficients:

$$F_{F,c}(j) = \sum_{i=1}^{n} \omega_c(i) F_{D,c}(i,j) \tag{1}$$

where n: is the number of pixels in the aperture, $w_c(i)$: are the desired filter coefficients, j: indicates the particular aperture belonging to class c.

The sum of the squared error between the filtered pixels and the target pixels is:

$$\begin{aligned} e^2 &= \sum_{j=1}^{N_c} (F_{R,c}(j) - F_{F,c}(j))^2 \\ &= \sum_{j=1}^{N_c} \left[F_{R,c}(j) - \sum_{i=1}^{n} \omega_c(i) F_{D,c}(i,j) \right]^2 \end{aligned} \tag{2}$$

where N_c represents the number of training samples belonging to class c. To minimise e^2, the first derivative of e^2 to $w_c(k)$, $k \in [1 \ldots n]$ should be equal to zero.

$$\frac{\partial e^2}{\partial \omega_c(k)} = \sum_{j=1}^{N_c} 2F_{D,c}(k,j) \times \left[F_{R,c}(j) - \sum_{i=1}^{n} \omega_c(i) F_{D,c}(i,j) \right] = 0 \tag{3}$$

The equation can be solved using Gaussian elimination and the optimal coefficients for the filter can be obtained as shown in the following matrix equation. These coefficients can then be stored in a look-up table with its corresponding class.

$$\begin{bmatrix} \omega_c(1) \\ \omega_c(2) \\ \ldots \\ \omega_c(n) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N_c} F_{D,c}(1,j)F_{D,c}(1,j) & \ldots & \sum_{j=1}^{N_c} F_{D,c}(1,j)F_{D,c}(n,j) \\ \sum_{j=1}^{N_c} F_{D,c}(2,j)F_{D,c}(1,j) & \ldots & \sum_{j=1}^{N_c} F_{D,c}(2,j)F_{D,c}(n,j) \\ \ldots & \ldots & \ldots \\ F_{D,c}(n,j)F_{D,c}(1,j) & \ldots & \sum_{j=1}^{N_c} F_{D,c}(n,j)F_{D,c}(n,j) \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{j=1}^{N_c} F_{D,c}(1,j)F_{R,c}(j) \\ \sum_{j=1}^{N_c} F_{D,c}(2,j)F_{R,c}(j) \\ \ldots \\ \sum_{j=1}^{N_c} F_{D,c}(n,j)F_{R,c}(j) \end{bmatrix} \tag{4}$$

Fig. 4. The lighthouse image.

2.1.1. *Adaptive dynamic range coding*

Adaptive Dynamic Range Coding (ADRC) is a strategy to encode structure information in a particular image as used in [13]. It is simple and computationally low cost. Taking an example of a 3×3 block, the encoding of ADRC can be explained as follows:

(i) The average of the pixels in the block is taken and stored into a variable V_{av}. (ii) A class code of the same number of bits as the number of pixels of the block is allocated. (iii) The ith pixel is then compared to V_{av}. If the value of the respective pixel is greater than or equal to V_{av}, then the ith element in class index is set to 1. Otherwise, the ith element of the class index is set to 0. Fig. 2 shows an example of ADRC coding.

2.2. *Online de-mosaicing process*

The online de-mosaicing process performs the same classification upon the Bayer pattern as the offline process. The class index is used to retrieve the filter coefficients from the look-up table that was stored during the offline training process. The retrieved filter coefficients are applied to the input image kernel to produce a full resolution $W \times H$ green plane. Fig. 3 depicts the online de-mosaicing process.

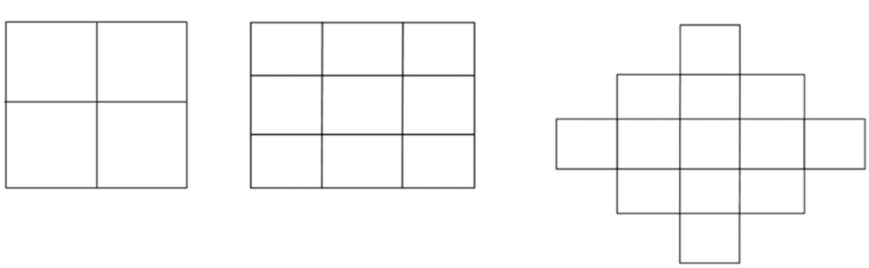


Fig. 5. Different classification windows for the green plane.

2.3. Classification

One of the crucial decisions to be made for classification is what shape and size of the window should be utilised for training. For de-mosaicing, the classification window is different to those in other enhancement methods, because the pixels are sub-sampled and distributed in a Bayer pattern.

2.3.1. Classification of the green plane

As a starting point it would be worthwhile to use only green pixels for classification. For this purpose, the "lighthouse" image, as shown in Fig. 4, was utilised as a test image to determine the best possible window option.

Fig. 5 shows the various windows which were used for classification of the green plane for the offline and online processes, namely a 2×2 window (needing 4 bits for classification), a 3×3 window (needing 9 bits for classification) and a diamond shaped window (needing 13 bits for classification). Here, only ADRC is adopted for classification. Note that a diamond shaped window is theoretically a good compromise, as it is bound to give results similar to a typical 5×5 window, however utilising a lower number of bits. The green pixels on all these three classification windows are selected from the sub-sampled Bayer pattern. Therefore, in the original Bayer pattern the classification windows are larger but sub-sampled.

Training was done using 2495 frames of varying resolutions and a Bayer pattern of the "lighthouse" image was the input to the online algorithm, for each of the three classification windows as shown in Fig. 5. The mean squared error (MSE) between the original image and the demosaiced image was obtained for the green plane, which can be seen in Fig. 6. Understandably, the MSE decreases as the classification window size increases, and the diamond shaped window performed the best amongst the tests.

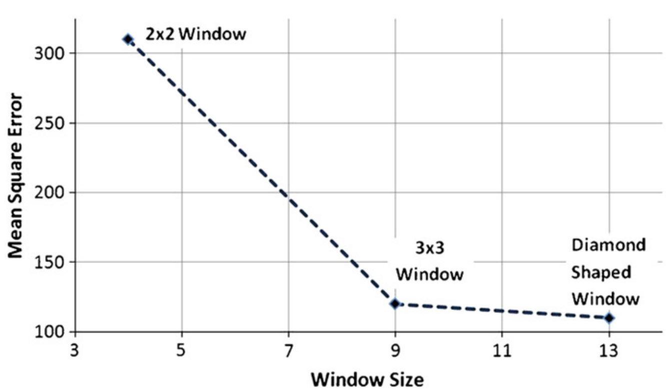


Fig. 6. MSE comparison of different classification windows.

Theoretically, it can be implied that the window can be made larger to further improve the results. It is, however, advantageous to utilise as few bits as possible for classification but still give the best possible MSE.

2.3.2. Classification of three colours

An alternative strategy is to also include the red and blue planes during classification. This would utilise the correlation between the three planes and should theoretically improve results in terms of MSE significantly. In this strategy, the ADRC is applied separately for the green pixels, the red pixels and the blue pixels. This means that three average values are calculated:

(i) V_{av_red}: average of red pixels, (ii) V_{av_green}: average of green pixels, and (iii) V_{av_blue}: average of blue pixels.

Each pixel in the kernel is compared with its respective V_{av}, which is average of the pixels of the plane that the pixel belongs to. Various sizes and shapes of the classification window were implemented and tested on the "lighthouse" image and the results are being discussed in this section.

Fig. 7 shows the various windows that were used for classification. Note that all of these classifications have two scenarios while parsing through a Bayer pattern: (a) when the central pixel is a red pixel and (b) when the central pixel is a blue pixel. The respective ADRC codes of three different colours are concatenated to constitute a class index for the look-up table. Two bits are appended to the combined ADRC class code to further refine the classification. These 2 bits represent the dynamic range of the green pixels, i.e., the difference between the maximum and the minimum of the green pixel values. The reason of using these 2 bits is to differentiate regions of high dynamic range from flat regions. One more bit is appended to the class index, which is a flag to indicate which "scenario" is being encountered in the classification. Notice that such a classification of the Bayer pattern, while utilising the features in the green plane, also utilises the correlation of the green pixels with the pixels in the other two planes. This is a key factor which is bound to improve demosaicing results and is the basis for such a classification strategy.

The Bayer pattern of the "lighthouse" image (Fig. 4) is used as a test image for all the different classification windows. The MSE is calculated between the original green plane and the reconstructed (demosaiced) green plane and the results are compared in Fig. 8. It can be observed that a 5×3 window produces the best results. Compared with Fig. 6, we can see that the utilisation of all three colours for classification substantially improves the results. Fig. 9 shows the reconstructed green plane of the "lighthouse" image. It can be seen that the reconstruction is very accurate with very minimal aliasing at the fence areas in the "lighthouse" image.

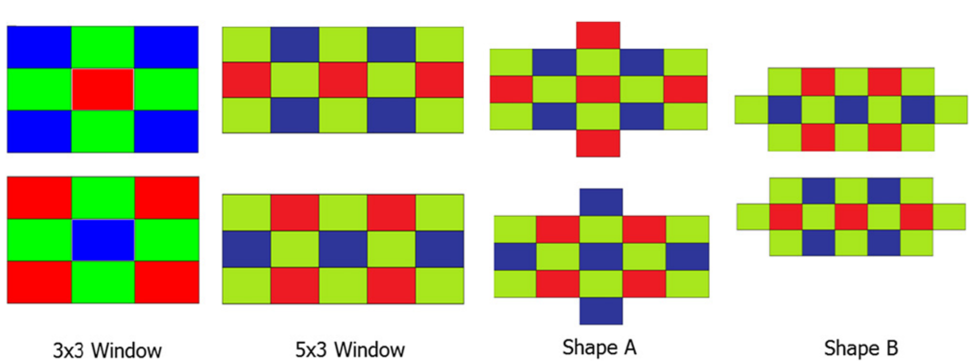


Fig. 7. Different classification windows using three colours.

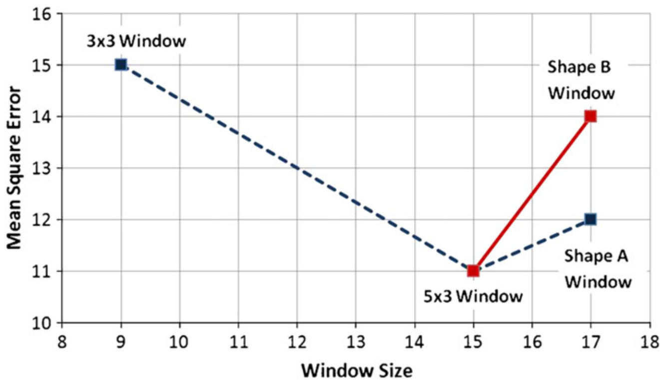


Fig. 8. MSE comparison of different classification windows.

Fig. 9. Reconstructed green plane of the “lighthouse” image.

2.4. Reconstruction of the red and blue planes

As mentioned, for a full coloured image, the red and blue planes also need to be interpolated. Both red and blue planes constitute 25% of pixels each in the Bayer pattern.

Bilinear interpolation is first applied on the red and blue planes, which are refined according to the high quality demosaiced green plane using the proposed method in the previous sections. The high frequency content of the green plane is super-imposed onto the red and blue planes. For this purpose, once the demosaiced green plane is obtained for an image, the high frequency details of this plane are extracted. This is achieved by passing the green plane through a blurring mechanism, and the result is stored in a separate buffer.

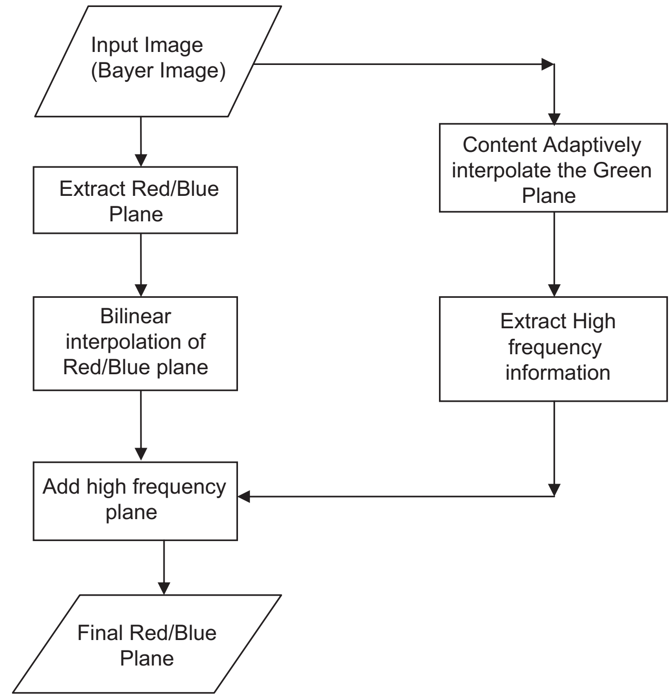


Fig. 10. Reconstruction of the Red and Blue planes.

When the difference of the blurred version is taken from the demosaiced green plane, the high frequency content of the green plane is obtained. This is then added pixel-wise onto the red and blue planes with some thresholds. Fig. 10 illustrates the steps in reconstructing the red and blue planes. Mathematically, for the red plane:

$$R(i,j) = Rl(i,j) + Gh(i,j)$$

where $R(i,j)$: Resultant R pixel at point (i,j), $Rl(i,j)$: Low Frequency R component at point (i,j), $Gh(i,j)$: High Frequency G component at point (i,j).

3. Results

The proposed method was tested on 24 images from the Kodak Image Suite [17]. Fig. 11 shows the result of the proposed method for the “lighthouse” image, which is a very popular image for demosaicing. It can be seen that, in areas near the fence which is highly detailed, very little aliasing is observed, while most demosaicing techniques would produce strong aliasing effects on those regions. Most of the aforementioned prevalent demosaicing techniques involve an edge detection mechanism. This is followed by an interpolation along the edges that are detected in the technique. A great level of computational power is therefore used in edge detection, which would get more pronounced in high definition images. The proposed technique provides an alternative where in a look up table is utilised and as Table 1 points out, it provides comparable and at times better PSNR values. It does not need to utilise an edge detection mechanism, and wholly relies on

Fig. 11. Full RGB reconstruction of the "lighthouse" image.

Table 1
PSNR comparison of results.

Image no	Bi-linear	[4]	[5]	[6]	[7]	Proposed method
1	26.22	33.60	37.91	38.01	38.66	34.91
2	37.12	38.30	39.10	39.62	39.87	41.14
3	34.50	40.51	41.53	41.74	42.82	43.36
4	33.77	38.58	39.96	40.63	40.60	42.11
5	26.67	34.80	37.45	37.88	37.94	35.57
6	27.86	34.82	38.78	39.96	40.02	37.72
7	33.50	40.77	41.59	42.07	42.58	42.11
8	23.61	32.14	35.64	35.27	36.67	35.12
9	32.52	40.23	41.88	42.16	43.06	41.14
10	32.51	39.55	41.85	41.29	42.40	41.14
11	29.22	35.98	39.38	39.81	39.77	38.59
12	33.48	40.48	42.61	43.05	43.59	42.11
13	23.92	29.80	34.55	35.13	35.07	32.95
14	29.28	35.34	35.74	36.01	36.68	37.72
15	32.35	37.30	38.95	39.39	39.42	40.35
16	31.37	38.30	42.22	43.70	43.34	42.11
17	32.16	38.46	41.53	41.69	41.43	39.68
18	28.03	33.82	37.13	37.30	37.34	36.67
19	28.12	36.90	40.14	40.41	41.29	39.10
20	31.64	38.47	40.69	40.40	41.38	40.35
21	28.55	35.27	38.95	38.34	39.59	37.72
22	30.47	36.46	37.77	38.21	38.48	38.59
23	35.21	41.53	41.86	42.20	42.87	43.36
24	26.71	31.93	34.67	35.35	34.80	35.34

ADRC, which in itself is a "pixel averaging" methodology. Despite being computationally low cost method, there is indeed a trade-off as the proposed technique requires a bit more memory for look up table storage, which is perhaps not a stringent requirement of the prevalent techniques.

Table 1 shows the PSNR results of the proposed algorithm and a few other prevalent techniques on images from the Kodak Image Suite [17]. The training of the look-up table is done using 2495 frames of varying resolutions and with different image contents. The classification window is a 5×3 window as shown in Fig. 7. From Table 1, we can see that the proposed de-mosaicing method is comparative to state-of-the-art techniques. The results can be improved even further, if more training images are used. The online algorithm was tested on a Pentium(R) Dual-Core CPU @2.00 GHz and 3 GB RAM, and it takes 0.35 s to de-mosaic an image of resolution 512×640.

4. Conclusion

A content adaptive de-mosaicing algorithm based on the trained filter is proposed in this paper. The offline training process optimises the de-mosaicing filter coefficients using pairs of original full resolution images and mosaiced Bayer patterns. Classification is done by employing the Adaptive Dynamic Range Coding (ADRC) and the focus is on accurately reconstructing the green plane, as the human eye is most sensitive to this plane. The red and blue planes are initially bi-linearly interpolated and then refined by adding high frequencies from the reconstructed green plane. Extensive experimentation shows the effectiveness of the proposed solution. The trained filter can be similarly applied on the red and blue planes to improve the overall de-mosaicing performance, but more memory will be needed to store the extra look-up tables.

References

[1] J.E. Adams, J.F. Hamilton Jr., Adaptive color plane interpolation in single color electronic camera, US Patent 5 506 619, April 1996.

[2] R. Ramanath, W.E. Snyder, Adaptive demosaicking, J. Electron. Imaging 12 (4) (2003) 633–642.

[3] D. Menon, S. Andriani, G. Calvagno, A novel technique for reducing demosaicing artifacts, In: Proc. European Signal Processing Conference (EUSIPCO 2006), Florence Italy – Int. Conf. Proceedings, Sept. 4 2006.

[4] R. Kakarala, Z. Baharav, Adaptive demosaicing with the principal vector method, IEEE Trans. Consum. Electron. 48 (4) (2002) 932–937.

[5] L. Chang, Y.P. Tan, Effective use of spatial and spectral correlations for color filter array demosaicking, IEEE Trans. Consum. Electron. 50 (2004) 355–365.

[6] J. Lee, T. Jeong, C. Lee, Edge-adaptive demosaicking for artifact suppression along line edges, IEEE Trans. Consum. Electron. 53 (3) (2007) 1076–1083.

[7] R. Lukac, K.N. Plataniotis, D. Hatzinakos, M. Aleksic, A new CFA interpolation framework, Signal Process. 86 (7) (2006) 1559–1579.

[8] K.H. Chung, Y.H. Chan, Color demosaicing using variance of color differences, IEEE Trans. Image Process. 15 (10) (2006) 2944–2955.

[9] K. Chung, W. Yang, W. Yan, C. Wang, Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection, IEEE Trans. Image Process. 17 (12) (2008) 2356–2367.

[10] S.C. Peiandi, K. Tam, Effective color interpolation in CCD color filter arrays using signal correlation, IEEE Trans. Circuits Syst. Video Technol. 13 (6) (2003) 503–513.

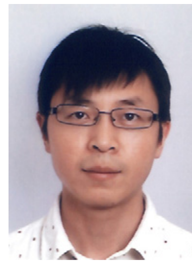
[11] W. Lu, Y.P. Tan, Color filter array demosaicking: new method and performance measures, IEEE Trans. Image Process. 12 (10) (2003) 1194–1210.

[12] E. Dubois, Frequency-domain methods for demosaicking of Bayer-sampled color images, IEEE Signal Process Lett. 12 (12) (2005) 847–850.

[13] L. Shao, H. Zhang, G. de Haan, An overview and performance evaluation of classification-based least squares trained filters, IEEE Trans. Image Process. 17 (10) (2008) 1772–1782.
[14] L. Shao, H. Hu, G. de Haan, Coding artifacts robust resolution up-conversion, IEEE Int. Conf. Image Process. 5 (2007) V-409–V-412.
[15] L. Shao, J. Wang, I. Kirenko, G. de Haan, Quality adaptive least squares filters for compression artifacts removal using a no-reference block visibility metric, J. Visual Commun. Image Represent. 22 (2011) 23–32(1) 22 (2011) 23–32.
[16] L. Shao, H. Zhang, L. Wang, L. Wang, Repairing imperfect video enhancement algorithms using classification-based trained filters, Signal, Image Video Process. 5 (3) (2011) 307–313.
[17] "Kodak Lossless True Color Image Suite", Jan. 15 2004 [Online] Available: ⟨http://r0k.us/graphics/kodak/⟩ [Accessed: Aug. 03, 2010].

Amin Ur Rehman received an MSc in Data Communications from the University of Sheffield, UK in 2010 and a B.E. in Electrical Engineering from NUST, Rawalpindi, Pakistan in 2007. From 2007 to 2009, he served as a software developer in the Video Encoding group in Streaming Networks Ltd, Pakistan. He was one of the front developers as he worked on products such as iRecord, iRecord Pro, iRecord Covert and Video Communicator. He also worked extensively on the H.264 Encoder during his time at Streaming Networks Pvt. Ltd and the emphasis of most of the development was on Phillips Trimedia PNX 1700 and PNX 1500. From 2010 to 2011, he served as a developer in Slowink Software Arts as a chief developer in "Music Information Retrieval". He was responsible for developing an "emotional features extraction" program wherein subjective content such as degree of "Melody", "Energy", "Beat presence", "Vocal content", etc. could be extracted from a song for the purpose of providing a signature to it.

Currently, he is serving as a software developer at Vision Base Ltd, Reading, UK, where he is currently working on developing and maintaining state of the art CCTV and Imaging Systems, which are installed in many key government locations worldwide. He has contributed actively to the development of Vision Base's unique video processing products.



Ling Shao received the B.Eng. degree in Electronic Engineering from the University of Science and Technology of China (USTC), the MSc degree in Medical Image Analysis and the PhD (D.Phil.) degree in Computer Vision at the Robotics Research Group from the University of Oxford.

Dr. Ling Shao is currently a Senior Lecturer (Associate Professor) in the Department of Electronic and Electrical Engineering at the University of Sheffield. Before joining Sheffield University, he worked for 4 years as a Senior Scientist in Philips Research, The Netherlands. Prior to that, he worked shortly as a Senior Research Engineer at the Institute of Electronics, Communications and Information Technology, Queen's University of Belfast. His research interests include Computer Vision, Pattern Recognition and Video Processing. He has published over 70 academic papers in refereed journals and conference proceedings and has filed over 10 patent applications. Ling Shao is an associate editor of the International Journal of Image and Graphics, the EURASIP Journal on Advances in Signal Processing, and Neurocomputing, and has edited several special issues for journals of IEEE, Elsevier and Springer. He has organized several workshops with top conferences, such as ICCV, ACM Multimedia and ACCV. He has been serving as Program Committee member for many international conferences, including CVPR, ICIP, ICASSP, ICME, ICMR, ACM MM, CIVR, BMVC, etc. He is a senior member of the IEEE.