Comparison of Demosaicking Methods for Color Information Extraction

Flore Faille

1. Introduction

Most digital color cameras are based on a single CCD or CMOS sensor combined with a color filter array (CFA): each pixel measures only one of the RGB colors. The most popular CFA is the Bayer CFA (Bayer, 1976) shown in fig. 1. Demosaicking algorithms interpolate the sparsely sampled color information to obtain a full resolution image, i.e. three color values per pixel. Many demosaicking algorithms were designed. However, even recent methods are prone to interpolation errors or artifacts, especially near edges. The most common artifacts are shown in fig. 2: "zipper" effects, wrong colors and wrong saturation of colored details. These artifacts are influenced by the sampled channel (R, G or B) and by the image content, like e.g. the orientation of the nearby edges. As a consequence, the same scene point may get a very different color value after a camera movement. This raises the two questions whether images acquired with a single chip camera can be used to extract reliable color information for further computer or robot vision tasks, and which demosaicking method suits best. Hence, this paper provides an overview and a detailed comparison of several state of the art and several recent demosaicking algorithms to enable the reader to choose the most appropriate demosaicking algorithm for his application. The previous comparisons between demosaicking methods, like the ones in (Lu & Tan, 2003; Ramanath et al., 2002), aimed at visually pleasing images. As a consequence, their evaluation criteria were, in addition to Mean Square Error (MSE) in RGB space, visual inspection and measures based on human perception like ΔE^*_{ab} (Lu & Tan, 2003; Ramanath et al., 2002). In computer vision tasks, separation between intensity and chrominance is widely used to increase robustness to illumination changes (Funt et al., 1998; Gevers & Smeulders, 1999). However, none of the previously used criteria can evaluate chrominance quality. For that reason, a detailed analysis using MSE in typical color spaces (HSI, Irb and YUV) is provided here. In addition, performance differences in colored, textured and homogeneous areas are emphasized. After an overview of the existing demosaicking algorithms, the methods chosen for comparison are introduced in section 2. The comparison framework and the results are presented in section 3. A conclusion is given in section 4.

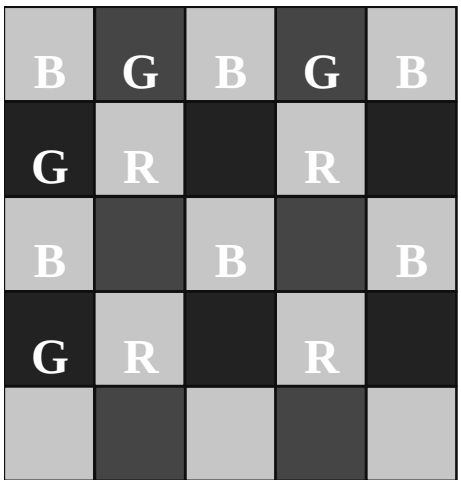


Figure 1. Bayer color filter array

Source: Cutting Edge Robotics, ISBN 3-86611-038-3, pp. 784, ARS/plV, Germany, July 2005 Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

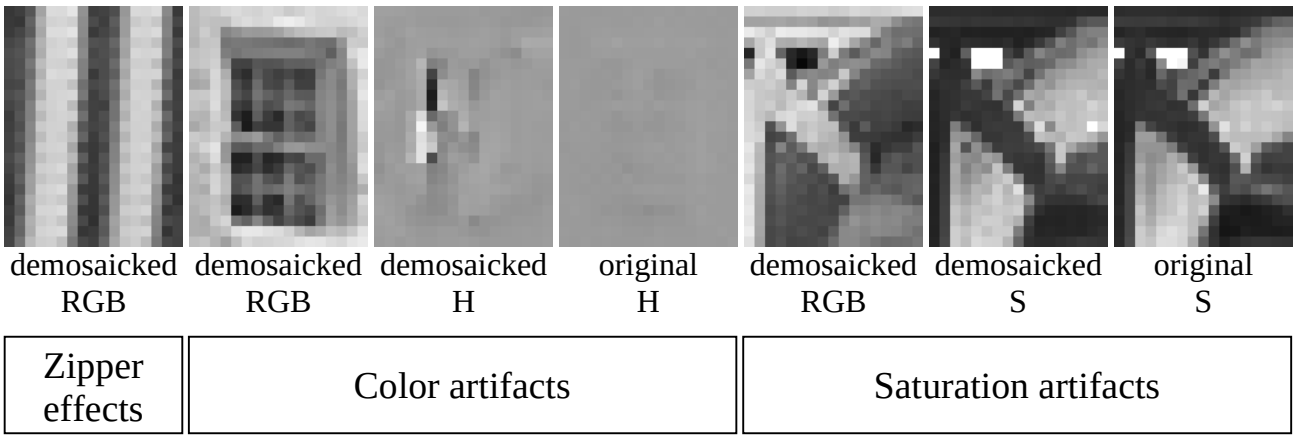


Figure 2. Overview of typical demosaicking artifacts on simulated CFA images. For a better visualization of the color and saturation artifacts, the hue (H) and the saturation (S) components are given for the demosaicked image and for the original three channel image. H and S components are scaled between 0 and 255. The images are available online in color (Faille, 2005)

2. Demosaicking Algorithms

2.1 Overview

The simplest demosaicking algorithm consists of a separate bilinear interpolation of the three channels. The obtained images are however very blurry and present many artifacts as shown in figs. 8(a) and 9(a). Demosaicking quality can be improved by enlarging the considered neighborhood. In addition, gradient information can be taken into account to adapt the used neighborhood to the image, as in (Hamilton & Adams, 1997). This allows interpolation to be performed along rather than across edges, hence reducing artifacts. Finally, many algorithms make use of the high inter-channel correlation to constrain the interpolation process: either the color differences R – G and B – G (Freeman, 1988; Hamilton & Adams, 1997) or the color ratios R/G and B/G (Cok, 1987) are assumed to remain constant in a small neighborhood. These principles improve demosaicking at a moderate increase of complexity. A good overview and comparison of many state of the art methods is given in (Ramanath et al., 2002). In this paper, the best two methods will be analyzed. The median-based postprocessing (MBP) enforces high inter-channel correlation to reduce artifacts after a first demosaicking step (Freeman, 1988). In addition to inter-channel correlation, the adaptive color plane interpolation (ACPI) takes gradients into account to select either horizontal or vertical interpolation (Hamilton & Adams, 1997). MBP and ACPI are described in more detail in sections 2.2 and 2.3.

The most interesting of the recently developed algorithms can be divided into three categories. The algorithms of the first category use principles similar to ACPI but provide more powerful and flexible ways to adapt the considered neighborhood. In (Lu & Tan, 2003) and in the first step of (Kimmel, 1999), each direction contributes to the interpolation with a weight which is proportional to the gradient inverse. In (Ramanath, 2003), the weights depend on the similarity to the center pixel after a first demosaicking step. Additionally, the framework to estimate R and B channels is improved in (Lu & Tan, 2003). This enhances the chrominance quality, so the method by (Lu & Tan, 2003) will be analyzed here. More details are given in section 2.4.

The second category of methods locally selects between horizontal and vertical interpolation as in ACPI, however based on more complex image measures than gradients. (Hirakawa & Parks, 2003) use image homogeneity in CIELAB color space.

(Omer & Werman, 2004) use the variation of R/G and B/G color ratios and the response to the Harris corner detector. As these complex measures cannot be estimated directly from the CFA sampled images, one horizontally interpolated image and one vertically interpolated image are generated first. The local direction selection is performed subsequently to generate a final image. This results in a very high computing time, even when direction selection is only performed in textured areas as in (Omer & Werman, 2004). For that reason, these methods will not be analyzed here.
The last category of methods works similarly to MBP. After a first demosaicking step, the result is postprocessed to enforce one or more constraints, hence reducing artifacts. In the second step of (Kimmel, 1999), the locally constant color ratio assumption is enforced, using thereby an adaptive neighborhood. In (Gunturk et al., 2002), a compromise between the following two constraints is reached: locally constant color differences and fidelity to sampled data. As these methods yield no significant enhancement over MBP, they will not be analyzed here.

2.2 Median-Based Postprocessing (MBP and EMBP)

MBP reduces demosaicking artifacts by enforcing high inter-channel correlation (Freeman, 1988). After a first demosaicking step, the difference images $\delta_R = R - G$ and $\delta_B = B - G$ are median-filtered. The image is then reconstructed using the filtered difference images δ_R and δ_B as well as the CFA sampled data. For example, at a sampled G pixel:

$$(R', G', B') = (\delta_R + G, G, \delta_B + G) \quad (1)$$

The algorithm works similarly at sampled R and B pixels. A larger filter kernel enhances the demosaicking results. As a consequence, a compromise between image quality and complexity is reached using the kernel proposed in (Freeman, 1988) and shown in Fig. 3: the value of the center pixel is the median of the nine pixel values indicated in grey. Demosaicking results are shown in figs. 8(b) and 9(b). Pixels with wrong intensity and wrong saturation appear near color edges with low inter-channel correlation such as for example red/white edges. To avoid such artifacts due to contradictions between the high inter-channel correlation model and the sampled data, an Enhanced Median-Based Postprocessing (EMBP) is used in (Hirakawa & Parks, 2003; Lu & Tan, 2003), in which sampled values are changed too. The reconstruction step becomes the same for all pixels:

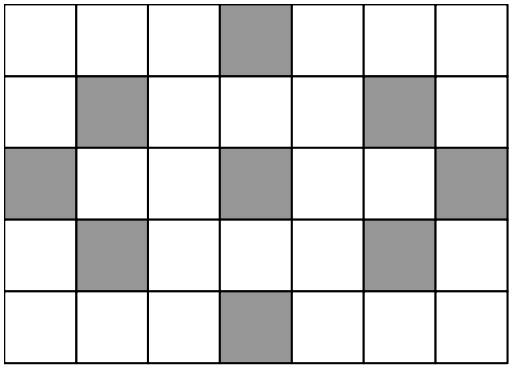


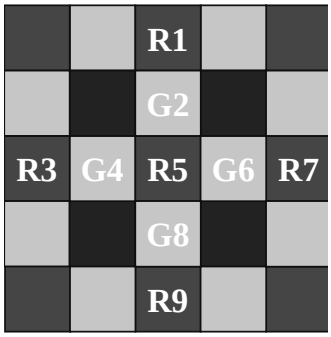
Figure 3. Kernel used for median filtering

$$(R', G', B') = (\delta_R + G', (R - \delta_R + B - \delta_B)/2, \delta_B + G') \quad (2)$$

The implementation proposed in (Lu & Tan, 2003) was chosen here. Already processed pixels are used to filter following pixels for a faster diffusion of the estimation. In addition, only textured areas (areas where the laplacian of the G channel is above a given threshold) are postprocessed, as homogeneous areas are less prone to artifacts. Results are presented in figs. 8(c) and 9(c). For the comparison in section 3, MBP and EMBP will be applied after the method by Lu and Tan (Lu & Tan, 2003) to obtain the best possible results.

2.3 Adaptive Color Plane Interpolation (ACPI)

ACPI is a state of the art method using gradient information and inter-channel correlation (Hamilton & Adams, 1997). As the Bayer CFA contains twice as many G pixels as R or B pixels, the G channel is interpolated first and is used to estimate R and B channels in a second step. Fig. 4 presents the G channel estimation. The interpolation is performed in the direction of the minimum gradient. To take the high inter-channel correlation into account, gradients and interpolated G values depend on the laplacian of the R (or B) channel. Similarly, estimated R and B values depend on the laplacian of the interpolated G channel. Gradient based neighborhood adaptation is only performed to estimate R (or B) values at sampled B (or R) pixels, for which four R (or B) neighbors exist. The demosaicking results are illustrated in figs. 8(d) and 9(d).



Compute the horizontal and the vertical gradients:

H = |G4 – G6| + |R5 – R3 + R5 – R7|

V = |G2 – G8| + |R5 – R1 + R5 – R9|

If H > V, interpolate along the vertical direction:

G5 = (G2 + G8) / 2 + (R5 – R1 + R5 – R9) / 4

else if V > H, interpolate along the horizontal direction:

G5 = (G4 + G6) / 2 + (R5 – R3 + R5 – R7) / 4

else use both directions:

G5 = (G2 + G8 + G4 + G6) / 4 + (4 R5 – R1 – R9 – R3 – R7) / 8

Figure 4. Interpolation of the G value at a sampled R pixel with ACPI (Hamilton & Adams, 1997). Sampled B pixels are processed similarly

2.4 Weighted Adaptive Color Plane Interpolation (WACPI)

WACPI (Lu & Tan, 2003) is based on the same principles as ACPI. G values are estimated first. Gradients and interpolated values depend on the laplacian of the other color channels, like in ACPI. Yet WACPI provides a more flexible framework to adapt the interpolation neighborhood to the image. Four directions (instead of two) are considered as shown in fig. 5, which enhances the performance near slanted edges and corners (see figs. 8(e) and 9(e)). The contributions of every direction to the interpolation $\tilde{G}_i$ are weighted by the gradient inverses α_i before they are summed up and normalized to build the estimate:

$$G = \frac{\alpha_{LEFT}\tilde{G}_{LEFT} + \alpha_{RIGHT}\tilde{G}_{RIGHT} + \alpha_{UP}\tilde{G}_{UP} + \alpha_{DOWN}\tilde{G}_{DOWN}}{\alpha_{LEFT} + \alpha_{RIGHT} + \alpha_{UP} + \alpha_{DOWN}} \quad (3)$$

The formulas for α_{LEFT} and $\tilde{G}_{LEFT}$ are given in fig. 6 as an example for all α_i and $\tilde{G}_i$. As can be seen, the contributions to the interpolation $\tilde{G}_i$ are the same as for ACPI. In contrast, the gradients are estimated on a larger neighborhood to compute the weights α_i. The constant additive term in the denominator of α_i (see fig. 6) avoids division by zero in homogeneous areas. The estimation of the R and B channels is also improved in comparison to ACPI, as the gradient based neighborhood adaptation is performed for the interpolation of all R and B values. To achieve this, R (or B) values at sampled B (or R) pixels are interpolated first. After this step, sampled G pixels have two sampled R (or B) neighbors and two estimated R (or B) neighbors. As a consequence, the neighborhood

adaptation framework can then be applied without any restriction. The reader should refer to (Lu & Tan, 2003) for the complete algorithm description.

3. Comparison of the Algorithms

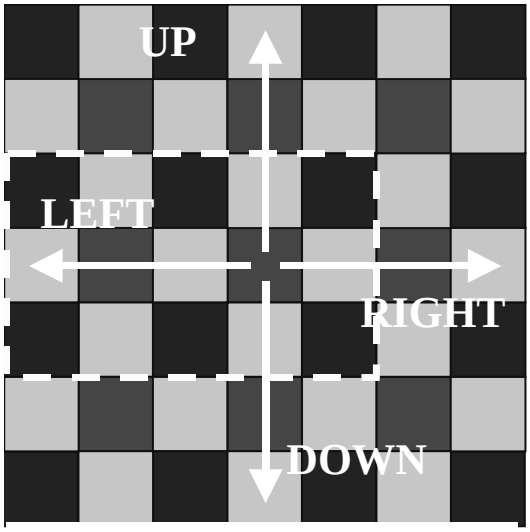
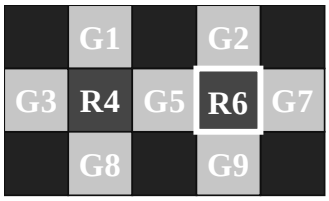


Figure 5. Considered directions and neighbourhood shown in fig. 6

The algorithms are evaluated on simulated CFA sampled images by comparison with the original three channel images. 24 images of the Kodak color image database are used. These images represent scenes of various content, like for example landscapes, persons, natural and man-made objects, as can be seen in Fig. 7.

Before the quantitative evaluation, the demosaicking results are illustrated in figs. 8 and 9 to show the different artifacts. The figures show one colorful area and one textured area of the Small Lighthouse image, a downsampled version of one of the Kodak database images used in (Kimmel, 1999; Lu & Tan, 2003).

The chosen areas illustrate best the results. To enable better visual analysis and understanding of the results, hue (H), saturation (S) and intensity (I) components are also presented, as HSI is an intuitive color space. For better visualization, the theoretical ranges of hue $[-\pi, \pi]$ and saturation [0, 1] are mapped to [0, 255]. Bilinear interpolation produces blurred images and artifacts known as "zipper" effects near edges (see figs. 8 and 9). For ACPI and WACPI, color artifacts may appear especially near slanted edges or corners as shown in fig. 9. MBP and EMBP correct these artifacts (see fig. 9) but introduce new ones near color edges (see fig. 8): MBP generates pixels with wrong intensity and wrong saturation, whereas EMBP generates areas with wrong saturation. In addition, MBP reconstructs hue more accurately than EMBP in colorful areas (see fig. 9).



$$\tilde{G}_{LEFT} = G5 + (R6 - R4)/2$$

$$\alpha_{LEFT} = \frac{1}{1 + |G7 - G5| + |G5 - G3| + |R6 - R4| + \frac{|G2 - G1| + |G9 - G8|}{2}}$$

Figure 6. Weight and contribution of the left direction (cf. fig. 5) to the interpolation of the G value at pixel position R6

Figure 7. Four images of the Kodak color image database

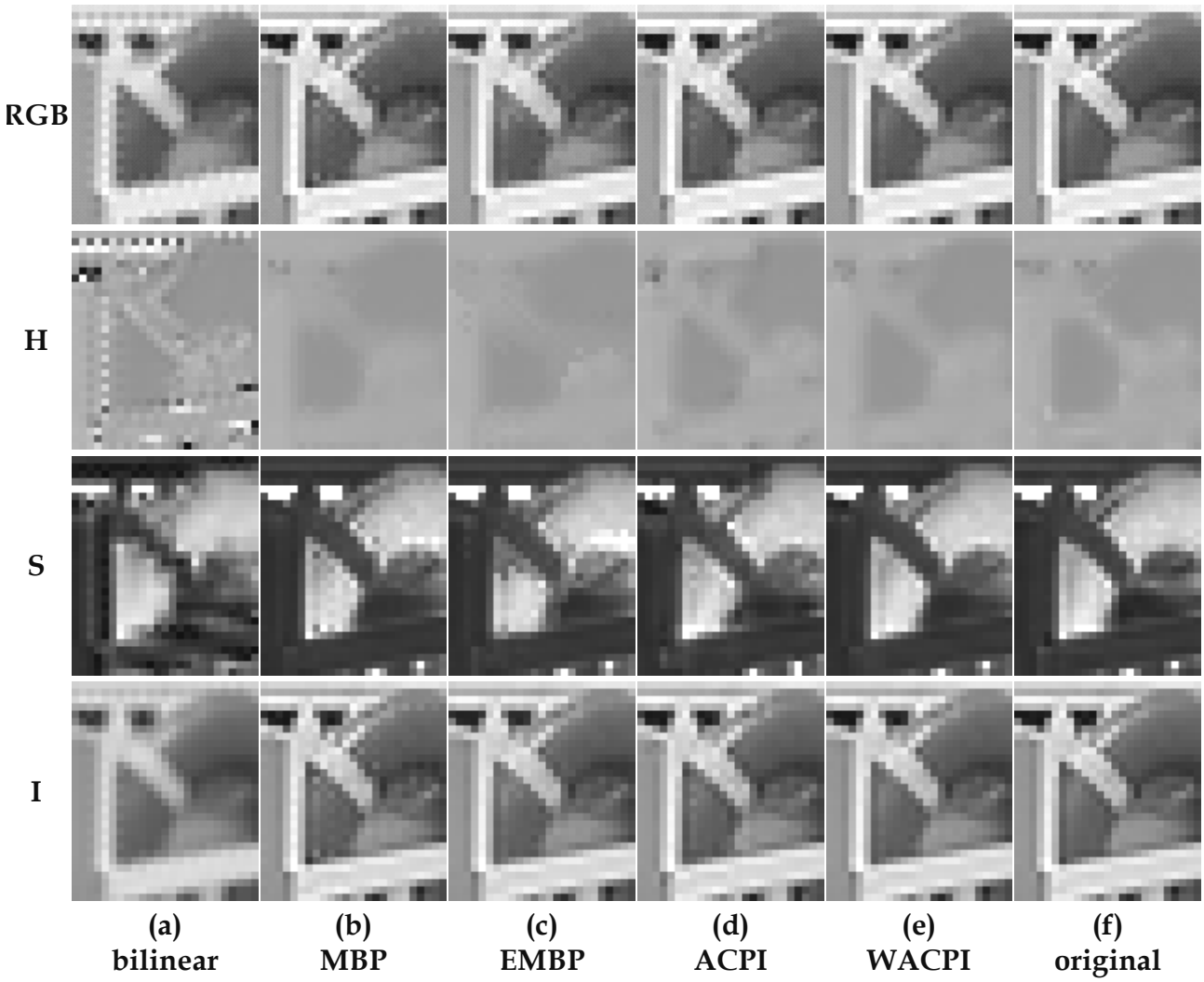


Figure 8. Enlarged colorful detail of the *Small Lighthouse* image showing a buoy. From left to right: (a) bilinear interpolation, (b) MBP, (c) EMBP, (d) ACPI, (e) WACPI, (f) original three channel image. From top to bottom: RGB image, hue, saturation, intensity. This figure is available online in color (Faille, 2005)

As mentioned before, popular color spaces for computer vision separate intensity and chrominance information to reduce sensitivity to light direction and intensity (Funt et al., 1998; Gevers & Smeulders, 1999).
To achieve this, chrominance components are based on ratios between channels. In addition, robustness to specularities can be achieved using channel differences (Gevers & Smeulders, 1999). HSI and Irb are two well-known color spaces based on these principles:

$$\begin{pmatrix} H \\ S \\ I \end{pmatrix} = \begin{pmatrix} \arctan \dfrac{\sqrt{3}(G-B)}{2R-G-B} \\ 1-\min(R,G,B)/I \\ (R+G+B)/3 \end{pmatrix} \text{ and } \begin{pmatrix} I \\ r \\ b \end{pmatrix} = \begin{pmatrix} (R+G+B)/3 \\ R/I \\ B/I \end{pmatrix} \tag{4}$$

YUV is another popular color space, as it is linear and often directly delivered by color cameras. Y gives the intensity information. Color information is represented by channel differences U and V instead of channel ratios like in HSI and Irb spaces:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.3 & 0.59 & 0.11 \\ -0.168 & -0.33 & 0.498 \\ 0.499 & -0.421 & -0.078 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \quad (5)$$

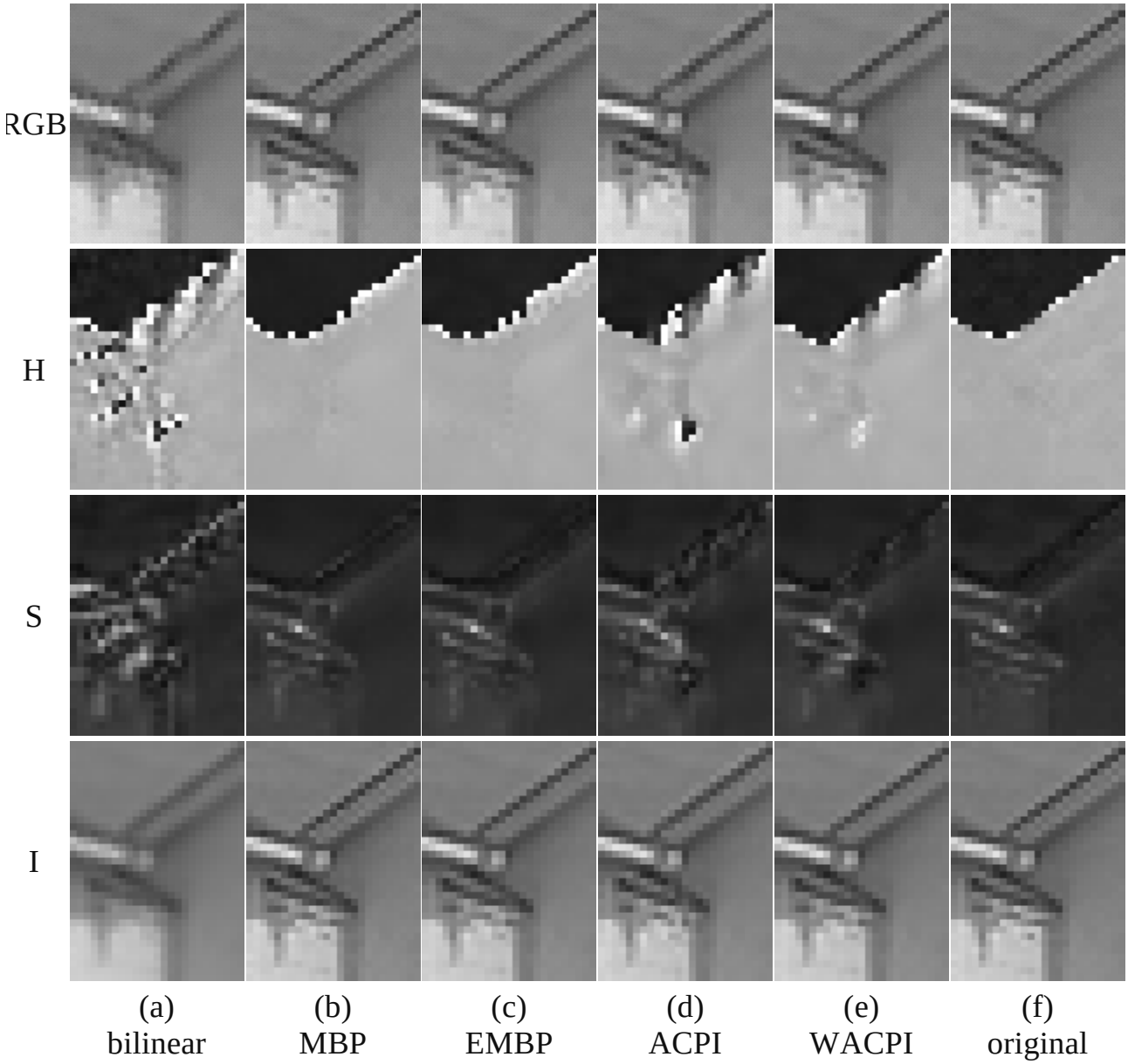


Figure 9. Enlarged textured detail of the *Small Lighthouse* image showing roofs. From left to right: (a) bilinear interpolation, (b) MBP, (c) EMBP, (d) ACPI, (e) WACPI, (f) original three channel image. From top to bottom: RGB image, hue, saturation, intensity. This figure is available online in color (Faille, 2005)

To analyze demosaicking quality for further computer or robot vision applications, performances will be evaluated using the Mean Square Error (MSE) between original and demosaicked images in HSI, Irb and YUV spaces. The MSE in RGB space is added for comparison with previous evaluations like (Lu & Tan, 2003; Ramanath et al., 2002). H, S, r and b are scaled by 100, so that their order of magnitude is similar to the other components. To avoid numerical instabilities, MSE for S, r and b was only estimated in sufficiently bright areas ($I > 0.9$). Similarly, MSE for H was only estimated in sufficiently

colored areas (S > 0.1). In addition, a three pixel wide border was left out to suppress any influence by border effects. As the results depend on image content, the average MSE over all 24 images is presented in table 1.
The performance discrepancy in textured and in homogeneous areas is emphasized (texture was detected in the original images with a Laplacian filter). As far as complexity is concerned, ACPI, WACPI, MBP (+ WACPI) and EMBP (+ WACPI) require on average over all images 2, 8.5, 20 and 14 times as much computation time as bilinear interpolation.

Algorithm	Edges				Homogeneous areas				Entire images			
	RGB	YUV	HSI	rb	RGB	YUV	HSI	rb	RGB	YUV	HSI	rb
Bilinear	215	84.2	21.1	1.40	14.1	5.32	.900	.0545	104	40.4	10.7	.669
	87.1	49.3	.966	1.57	5.64	3.42	.0549	.0587	42.0	24.0	.460	.748
	222	45.8	95.4		15.1	3.14	6.14		108	22.2	45.9	
ACPI	38.1	18.1	3.10	.394	4.85	1.97	.370	.0245	20.1	9.60	1.79	.202
	24.9	9.59	.268	.319	2.60	1.27	.0223	.0232	13.2	5.20	.141	.168
	33.6	10.1	15.8		4.52	1.26	1.89		18.4	5.30	8.46	
WACPI	**23.6**	10.4	1.63	**.232**	**3.44**	**1.34**	.273	**.0178**	**12.4**	5.57	.936	**.118**
	13.3	5.99	**.187**	**.210**	**1.71**	**.956**	**.0173**	**.0180**	7.22	3.25	**.0959**	.108
	21.6	**5.99**	9.44		**3.52**	**.884**	**1.35**		11.7	3.15	5.08	
MBP	25.2	**8.51**	**1.62**	.249	4.01	1.52	**.269**	.0206	12.6	**4.39**	**.814**	.120
	9.93	**5.57**	.195	.212	1.91	1.09	.0204	.0212	**5.18**	**2.95**	.0976	**.106**
	18.4	6.08	**8.13**		3.67	.962	1.49		**9.72**	**3.02**	**4.20**	
EMBP	29.7	11.0	2.30	.543	3.51	1.36	.281	.0201	14.4	5.63	1.09	.245
	18.1	6.38	.264	.283	1.79	.980	.0182	.0188	8.57	3.18	.124	.134
	24.0	11.0	9.64		3.58	.938	1.36		12.1	4.86	5.07	

Table 1. Demosaicking performance near edges, in homogeneous areas and on entire images. The average MSE over 24 images of the Kodak image database is given for RGB, YUV, HSI and Irb spaces. Each row shows a channel (top row for R, Y, H and r, etc.). As I is the same in HSI and Irb, the MSE for I is only given for HSI. The best performance in each category is indicated in boldface

Bilinear interpolation, which does not use gradients and inter-channel correlation, yields by far the worst results.
Despite its low complexity, ACPI allows significant enhancement, proving the strength of the used principles. WACPI and MBP perform best. The quality of color ratios, of saturation and of color differences are on average comparable for both (see MSE for r, b, S, U and V). MBP improves texture estimation (see MSE for R, G, B, Y and I) and reduces color artifacts (see MSE for H). But its performance in homogeneous areas is worse. This is mostly due to its higher sensitivity to the inaccuracy of the high inter-channel correlation model (constant color differences in a neighborhood) in colored areas. Even very low contrast edges in colored areas may result in pixels with wrong intensity and saturation. As told in the overview in section 2.1, the recent methods similar to MBP (Kimmel, 1999; Gunturk et al., 2002) do not significantly reduce this sensitivity to model inaccuracy. The overall performance of EMBP is moderate. It better estimates homogeneous areas than MBP, but achieves poor chrominance quality compared to WACPI and MBP.
To emphasize the model inaccuracy in colored areas, table 2 gives the average MSE in areas with high saturation (S ≥ 0.3). As in table 1, WACPI and MBP perform best. Yet MBP shows the maximal performance drop, especially for the estimation of texture (see MSE for

R, G, B, Y and I) and color differences (U and V). In this case, WACPI achieves the best results. MBP's higher sensitivity to the model inaccuracy is also shown by a higher number of negative (hence invalid) interpolated values: if these are not corrected to 0, the average MSE for all 24 entire images e.g. for r becomes 0.441 for WACPI and 0.965 for MBP. To summarize, MBP better estimates texture and reduces wrong color artifacts, but it is outperformed by WACPI in homogeneous and in colored areas. This explains our observation that MBP is better on images with fine textures (e.g. landscapes or natural objects), while WACPI is better on images showing man-made objects and on close-ups. Another consequence is that MBP should be applied after white balancing, so that the number of edges in colored areas is reduced to a minimum.

ACPI				WACPI				MBP				EMBP			
RGB	YUV	HSI	rb	RGB	YUV	HSI	rb	RGB	YUV	HSI	rb	RGB	YUV	HSI	rb
30.8	14.1	2.15	.747	**19.2**	**8.05**	1.12	**.435**	28.2	9.43	**.817**	.452	27.2	9.25	1.06	.806
19.6	7.13	.756	.598	**10.7**	**4.67**	.584	.409	11.3	5.54	**.524**	**.407**	15.8	6.02	.759	.484
27.6	8.78	12.8		**18.8**	**5.49**	7.75		19.2	7.18	8.99		21.0	10.3	7.95	

Table 2. Demosaicking performances in colored areas (with $S \geq 0.3$). As in table 1, the average MSE over all 24 images is given in RGB, YUV, HSI and Irb color spaces. The results for the bilinear interpolation are omitted, as it yields by far the worst results in table 1

4. Conclusion

After an overview over state of the art and recent demosaicking methods, selected algorithms were compared using images with various content. To verify if demosaicked images are suitable for computer or robot vision tasks, the average MSE in typical color spaces (RGB, YUV, HSI and Irb) was measured. While the high inter-channel correlation model improves interpolation results significantly, it was also shown to be inaccurate in colored areas. WACPI and MBP (+WACPI) provide the best results. WACPI performs better in colored and in homogeneous areas. MBP better reconstructs texture and reduces wrong color artifacts. The choice between both algorithms should depend on the application, according to the most relevant information and to the image type (indoor/outdoor, natural/man-made objects). The MBP algorithm could be enhanced by processing only edge pixels like in EMBP: this would reduce the computation time and improve the performance in homogeneous areas. In addition, regions with saturated colors could be left unchanged. If emphasis lies on efficiency, ACPI and WACPI achieve the best compromise between speed and quality.

5. References

Bayer, B. E. (1976). Color imaging array. United States Patent No. 3,971,065

Cox, D. R. (1987). Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. United States Patent No. 4,724,395

Faille, F. (2005) http://www.rcs.ei.tum.de/~faille/ARSbook.html

Freeman, W. T. (1988). Method and apparatus for reconstructing missing color samples. United States Patent No. 4,774,565

Funt, B.; Barnard, K. & Martin, L. (1998). Is machine colour constancy good enough?, Proceedings of the European Conference on Computer Vision (ECCV98), Burkhardt,

H. & Neumann, B. (Eds.), Vol. 1, pp. 445-459, ISBN: 3-540-64569-1, Freiburg, Germany, June 1998, Springer

Gevers, T. & Smeulders, A. W. M. (1999). Color-based object recognition. Pattern Recognition, Vol. 32, No. 3, March 1999, pp. 453-464, ISSN: 0031-3203

Gunturk, B. K.; Altunbasak, Y. & Mersereau, R. M. (2002). Color plane interpolation using alternating projections. IEEE Transactions on Image Processing, Vol. 11, No. 9, September 2002, pp. 997-1013, ISSN: 1057-7149

Kimmel, R. (1999). Demosaicing: image reconstruction from color CCD samples. IEEE Transactions on Image Processing, Vol. 8, No. 9, September 1999, pp. 1221-1228, ISSN: 1057-7149

Hamilton, J. & Adams, J. (1997). Adaptive color plane interpolation in single sensor color electronic camera. United States Patent No. 5,629,734

Hirakawa, K. & Parks, T. W. (2003). Adaptive homogeneity-directed demosaicing algorithm, Proceedings of the IEEE International Conference on Image Processing (ICIP03), Vol. 2, pp. 669-672, ISBN: 0-7803-7750-8, Barcelona, Spain, September 2003, IEEE Press

Lu, W. & Tan, Y.-P. (2003) Color filter array demosaicking: New method and performance measures. IEEE Transactions on Image Processing, Vol. 12, No. 10, October 2003, pp. 1194-1210, ISSN: 1057-7149

Omer, I. & Werman, M. (2004). Using natural image properties as demosaicing hints, Proceedings of the IEEE International Conference on Image Processing (ICIP04), pp. 1665-1670, ISBN: 0-7803-8554-3, Singapore, October 2004, IEEE Press

Ramanath, R.; Snyder, W. E.; Bilbro, G. L. & Sander, W. A. (2002). Demosaicking methods for bayer color arrays. Journal of Electronic Imaging, Vol. 11, No. 3, July 2002, pp. 306-315, ISSN: 1017-9909

Ramanath, R. & Snyder W. E. (2003). Adaptive demosaicking. Journal of Electronic Imaging, Vol. 12, No. 4, October 2003, pp. 633-642, ISSN: 1017-9909

Cutting Edge Robotics

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-038-3

Hard cover, 784 pages

Publisher Pro Literatur Verlag, Germany

Published online 01, July, 2005

Published in print edition July, 2005

This book is the result of inspirations and contributions from many researchers worldwide. It presents a collection of wide range research results of robotics scientific community. Various aspects of current research in robotics area are explored and discussed. The book begins with researches in robot modelling & design, in which different approaches in kinematical, dynamical and other design issues of mobile robots are discussed. Second chapter deals with various sensor systems, but the major part of the chapter is devoted to robotic vision systems. Chapter III is devoted to robot navigation and presents different navigation architectures. The chapter IV is devoted to research on adaptive and learning systems in mobile robots area. The chapter V speaks about different application areas of multi-robot systems. Other emerging field is discussed in chapter VI - the human- robot interaction. Chapter VII gives a great tutorial on legged robot systems and one research overview on design of a humanoid robot.The different examples of service robots are showed in chapter VIII. Chapter IX is oriented to industrial robots, i.e. robot manipulators. Different mechatronic systems oriented on robotics are explored in the last chapter of the book.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821