GLOBAL GEOMETRIC DEMOSAICKING

By

Sira Ferradans

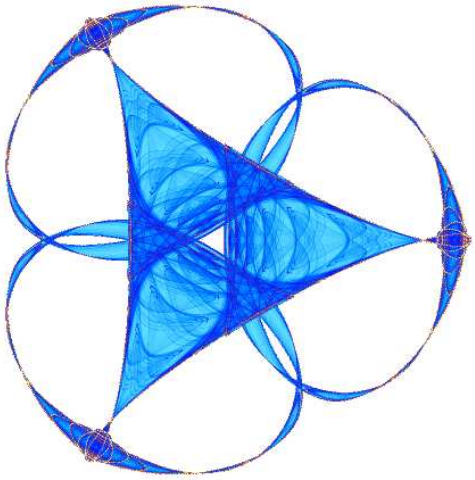
Marcelo Bertalmío

and

Vicent Caselles

IMA Preprint Series # 2192

(April 2008)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall

207 Church Street S.E.

Minneapolis, Minnesota 55455–0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: http://www.ima.umn.edu

Global Geometric Demosaicking

Sira Ferradans, Marcelo Bertalmío and Vicent Caselles

Abstract

Demosaicking is a particular case of interpolation problem where, from a scalar image in which each pixel has either the red, the green or the blue component, we want to interpolate the full-color image. State of the art demosaicking algorithms perform interpolation along edges, but these edges are estimated locally. We propose a global method to estimate image edges, inspired by the image inpainting literature. This method has a time complexity of $O(S)$, where S is the number of pixels in the image, and compares favorably with the state of the art algorithms both visually and in most relevant image quality measures.

Index Terms

Demosaicking, interpolation, inpainting, level-sets.

I. Introduction

When a digital camera captures a color image, for each pixel we have three values corresponding to three color channels: typically, three 8-bit numbers that measure the amount of red (R), green (G) and blue (B) present in the pixel. But for cost reasons most digital cameras (photo, digital video and mobile phone cameras) have only one sensor array, instead of three. This sensor is covered with a color filter array (CFA) which makes each pixel in the camera capture only one color channel. So in order to output a full-color image an interpolation process called *demosaicking* must be performed. Figure 1 (a) shows the Bayer pattern, the most common CFA.

A. The case for Demosaicking

As we mentioned, most digital imaging devices have one CCD or sensor array only. Digital photo and video cameras with three CCD's are way more expensive than their one-CCD counterparts, and mobile phone cameras have one CCD only and will apparently remain that way in the foreseeable future.

Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra, 08003 Barcelona, Spain (e-mail: {sira.ferradans, marcelo.bertalmio, vicent.caselles}@upf.edu).

Ramanath et al. point out in [1] that in order to counter-act the severe aliasing artifacts introduced by the CFA sampling, digital cameras use either birefringent materials or phase delay techniques to increase the correlation between pixels, helping the demosaicking process, but at the same time implying a reduction in the effective resolution of the camera. A good demosaicking process could therefore eliminate the need for this anti-aliasing stage.

The ever-increasing resolution of consumer cameras does not imply that "any demosaicking algorithm will do". In digital video the resolution is really not that high, while in digital photography a 10Mpixel image could probably get away with a very simple demosaicking procedure for display on a PC screen, but that would definitely not do for large-size printing or for zooming into details.

B. Previous work

There is abundant literature on demosaicking, and for works pre-dating 2005 the interested reader is directed to the excellent survey by Gunturk et al. [2]; as for more recent works, we will mention the following. In [3] the author proposes an iterative scheme where color differences are averaged to estimate the green plane, then the red and blue ones, and a spatially adaptive stopping criterion is defined that tries to minimize the creation of artifacts. In several works the Bayer image is interpolated first horizontally and then vertically, and both images are combined into a single image according to a certain criterion: by choosing the direction with less artifacts [4], or the direction of image contours [5], [6]. In [7] the direction of the interpolation for green plane is selected as the one (horizontal, vertical or diagonal) where the color difference has minimum variance. In [8] demosaicking is performed by averaging pixels with similar neighborhoods. In [9] the authors propose an interpolation method where an image dictionary of image patches is learned that leads to the sparsest representation of patches in the demosaicked image.

C. Our contribution

State of the art results for demosaicking [5], [6], [7] are obtained with *edge direction* approaches: the missing color information is interpolated along the direction of the edges or contours of the image. But to the best of our knowledge in all the demosaicking literature the procedure of estimating the direction of edges is always *local*: in order to estimate the direction of the edge at pixel p, these algorithms look at the neighborhood of p and make a decision which is independent of the decision for any other pixel q.

In this article we propose a demosaicking algorithm inspired by works in the inpainting literature, such as the disocclusion approach of Masnou [10] and the variational fill-in of Ballester et al. [11]. We

define an energy functional whose minimization, a *global procedure*, gives the edge direction at every pixel. Although we use a variational, level-set approach in the formulation, in the implementation we do not use Partial Differential Equations (PDE's) but a very simple dynamic programming scheme of linear computational time complexity: $O(S)$ where S is the number of pixels in the image.

Experimentally we have found that our algorithm compared favorably with state of the art methods in all these quality measures: CIELAB distance, Peak Signal-to-Noise Ratio (PSNR), and Zipper Effect measure [12].

In this paper we will refer to pixels as integer coordinates in the image, this is, the position of pixel p will be defined as (i, j), and its intensity as $I(p)$. The red, green and blue planes will be referred to as I_R, I_G and I_B respectively. Red pixels with original value in the Bayer pattern will be noted as $I_R(p)$; and the interpolated values will be noted as $\hat{I}_R(p)$. The same notation is used for the green and blue channels.

II. The proposed algorithm: The Global Geometric Demosaicking

A. Hypothesis: edges coincide in the three color channels

The main hypothesis in our formulation is that the three color channels are strongly correlated and share the same geometric information, i.e. the same family of level lines. This assumption can be considered as a reasonable one for natural images and it was experimentally checked in [13]. It is also consistent with the experiments on correlation among color channels reported by Gunturk et al. [14]. Indeed, it is actually at the basis of many demosaicking algorithms, e.g. [15], [2], [5], [7]. The algorithms using *hue-based interpolation* first interpolate I_G and then assume perfect correlation among channels to interpolate I_R and I_B, see [2]. The same hypothesis is used by state of the art methods in other color interpolation problems such as colorization of grayscale images [16], [17], [18], deinterlacing [19] or movie denoising [20].

B. Demosaicking, level-sets and inpainting

In [21] Caselles et al. introduced the concept of topographic maps for natural images. Given a grayscale image I, the upper level-set $X_\lambda(I)$ of level λ of I is the set of pixels which take a gray value greater than λ. The boundary of the level-set $X_\lambda(I)$ is the level-line of level λ: a closed curve which is the contour of level λ of I. The family of all level lines of I is the topographic map of I. If we have the topographic map of I then we can reconstruct I, up to a change in contrast, i.e., all the geometric information of an image is contained in its family of level lines. Therefore, if we apply a change of contrast to the image

its topographic map does not change since its geometry, the shape of its level-lines or contours, remains the same. Another property of topographic maps is that level-lines do not cross each other.

In section II-A we introduced our only hypothesis, which can now be expressed in the following way: I_R, I_G and I_B have the same topographic map. Therefore, demosaicking amounts to finding the level-lines in the empty regions of I_G and then interpolating I_R and I_B along those lines. We choose I_G because it is the most densely sampled.

With the Bayer CFA, let us say that the empty regions of I_G are the odd 45° diagonals (see fig. 1 (b)). In order to fill-in the odd diagonal of number n, the green values we create must respect the level-set structure of the adjacent diagonals $n-1$ and $n+1$: one way to state this is by requiring that, if the λ level-line intersects diagonal number $n-1$ at pixel p and diagonal number $n+1$ at pixel q, then we must fill-in pixel $\frac{1}{2}(p+q)$, which lies on diagonal number n, with the green value λ.

Now we can see the connection with these works in the inpainting literature: the disocclusion approach of Masnou [10] and the variational fill-in of Ballester et al. [11]. Both approaches propose to fill-in information in an empty region by creating the level-lines inside it, and these level-lines are chosen so as to minimize a global energy functional related to the Elastica functional [22]: the idea is to favor level-lines which are short and smooth. In our case, we can decompose the demosaicking problem into many inpainting problems, one for each missing diagonal; in each of these problems, the inpainting region or *image gap* will be the missing diagonal n and the boundary conditions are given by the green values at diagonals $n-1$ and $n+1$.

C. Inpainting one diagonal

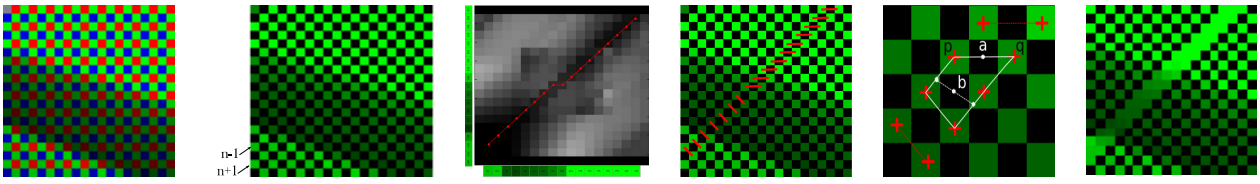
When the image gap is only one-pixel wide, as it is in our case, it makes sense to impose that the level-lines are straight-line segments, and therefore finding the orientation of every level line means to compute the optimal matching function between two pixel sequences: diagonals $(n-1)$ and $(n+1)$. Because level lines never cross, the global matching function will be non-decreasing (see fig. 1 (c)) and the optimization problem can be solved with dynamic programming as Cox et al. propose in the context of image stereo in [23].

We define the cost of a segment pq (where p is in diagonal $n-1$ and q is in diagonal $n+1$) as a function that measures the difference between the neighborhoods of p and q (higher difference implies higher cost) and the length of the segment (higher length also implies higher cost). The cost function between pixel p and pixel q is defined as:

$$cost(p,q) = \alpha + \beta|p-q|D(p,q) \tag{1}$$

where α and β are constants and $D(p,q)$ measures the difference between the neighborhoods of p and q, as it will be explained in detail in section III.

We define our inpainting solution, that is, the optimal match between the diagonals, as the one which has lowest energy, i.e. the set of correspondences that minimize the global accumulated cost. See [19] for an application of this same approach to de-interlacing. The use of dynamic programming for inpainting was proposed by Masnou in [10].



(a) (b) (c) (d) (e) (f)

Fig. 1. **(a)** Image mosaicked with the Bayer filter. **(b)** Green plane, note the 45° diagonals $n-1$, $n+1$. **(c)** Cost matrix for diagonals $n-1$ and $n+1$ **(d)** Dense set of correspondences (points in the optimal path) obtained between diagonals $n-1$ and $n+1$ with optimal path (minimum accumulated cost) superimposed in red. **(e)** Interpolation process, note the different cases for pixel a and pixel b **(f)** Complete diagonal obtained by interpolation.

Once we have the optimum matching, we have a dense set of correspondences between pixels in diagonals $n-1$ and $n+1$. So for each matching pair (p,q) we fill pixel $\frac{1}{2}(p+q)$ of the empty diagonal n with the average value of pixels p and q (see figure 1 (e) pixel a). For those pixels which do not correspond to the middle point of any matching pair (p,q), bilinear interpolation is computed within the trapezoid defined by the closest matchings, see figure 1 (e) pixel b.

Second derivatives are also introduced in the interpolation as a small correction term in the spirit of [7]. For instance, if pixel a has a red value in the Bayer pattern and a is in the middle point of the matching pair (p,q), we compute the green value at a as:

$$\hat{I}_G(a) = \frac{I_G(p)+I_G(q)}{2} + \begin{cases} \frac{(I_R(a)-I_R(a-(0,2))+(I_R(a)-I_R(a+(0,2)))}{4} & \text{if } \bar{pq} \text{ is horizontal} \\ \frac{(I_R(a)-I_R(a-(2,0))+(I_R(a)-I_R(a+(2,0))}{4} & \text{if } \bar{pq} \text{ is vertical} \\ \frac{(4I_R(a)-I_R(a-(0,2))-I_R(a+(0,2))-I_R(a-(2,0))-I_R(a+(2,0)))}{8} & \text{otherwise} \end{cases}$$

Notice that this identity can be written as $D_v^2 I_G(a) = D_v^2 I_R(a)$ where D_v^2 denotes the second derivative in the direction of the level line represented by v and this is consistent with our main assumption described in Section II-A.

D. Full-color image

After we have performed the inpainting of all empty $+45^\circ$ diagonals we have the whole G channel interpolated. We complete the R and B channels using the classical hue-based interpolation technique described in [2]. For instance, if pixel a is in a green position (we have $I_G(a)$) and we want to interpolate the red value $\hat{I}_R(a)$, we define a window Ψ_R centered at a and compute:

$$\hat{I}_R(a) = I_G(a) + \sum_{w \in \Psi_R} \frac{I_R(a+w) - \hat{I}_G(a+w)}{N} \tag{2}$$

where in Ψ_R we consider only the pixels located at the red channel positions, their number being N. The same process is performed for the blue plane and at the end we obtain a full color image I_+.

We repeat the same process but now using as inpainting gaps the empty -45° diagonals, obtaining another full color demosaicked image I_-. We merge both images I_+ and I_- using a self-similarity measure: for each pixel p we choose the value in the image ($I_+(p)$ or $I_-(p)$) which has a higher self-similarity value measured as the minimum color difference between this pixel (p) and the pixels in its neighborhood. Finally, we perform a refinement step as in [24] which is similar in spirit to anisotropic diffusion along the level lines of the image.

III. The algorithm in detail

The process described in section II which we call our core algorithm is iterated three times using in each step a different neighborhood distance measure D in the cost function (1).

First of all, we use as a difference measure in (1) the comparison of neighborhoods in I_G; with this particular choice of cost function our core algorithm produces a demosaicked output which we call I_1. Secondly, in order to capture high frequencies, we use in (1) an interchannel-interpixel measure in the spirit of [5], and obtain I_2. These two images are merged with the self-similarity measure described above, obtaining I_m. Then we apply the core algorithm to I_m, using in (1) the comparison of neighborhoods in the three channels (since I_m is a full color image) to obtain I_{m3}. Finally, we apply the self-similarity merging algorithm to I_m and I_{m3} obtaining the final output I_o of our algorithm. Even though the resulting image I_o has accurate edges and details, a final refinement step as in [24] can be applied.

As we stated in the last section, the definition of the cost depends directly on the difference function D used in (1). The difference measures D_i, $i = 1, 2, 3$ that we use in the three iterations of our core algorithm are defined as:

1) Green channel comparison:

$$D_1(p,q) = \sqrt{\sum_{w\in\Psi_G}\left(\frac{I_G(p+w) - \bar{I}_{G,p} - I_G(q+w) + \bar{I}_{G,q}}{N}\right)^2} \tag{3}$$

where Ψ_G is a $5x5$ neighborhood centered in $(0,0)$ which only has effective pixels at the G channel locations, $\bar{I}$ is the average value within Ψ_G, and N is the number of effective pixels in Ψ_G (therefore N = 13).

2) Interchannel and interpixel comparison. We define S as an interchannel derivative; let p be at a red position in the Bayer pattern, the horizontal derivative is defined as:

$$S_h(p) = I_R(p) - I_G(p+(0,1)) \tag{4}$$

This measure can be applied to every combination of contiguous pixels. Because we assume that level lines coincide, S is expected to be constant within a homogeneous area thus its derivative will be zero; for further analysis refer to [5]. Therefore we obtain a measure of the homogeneity of a neighborhood by comparing its S measures:

$$Deriv_h(p) = S_h(p) - S_h(p+(0,2)) \tag{5}$$

$$IID_h(p,q) = \sqrt{\sum_{w\in\Psi}\frac{|Deriv_h(p+w) - Deriv_h(q+w)|}{N}} \tag{6}$$

The size of the window was also set to $5x5$ therefore N=10. We compute these derivatives along the horizontal and vertical orientations, and average them:

$$D_2(p,q) = \frac{IID_h(p,q) + IID_v(p,q)}{2} \tag{7}$$

3) Three channel comparison:

$$D_3(p,q) = \frac{1}{N}\sum_{w\in\Psi}\sum_{c\in R,G,B}[I_c(p+w) - \bar{I}_{c,p} - I_c(q+w) + \bar{I}_{c,q}]^2 \tag{8}$$

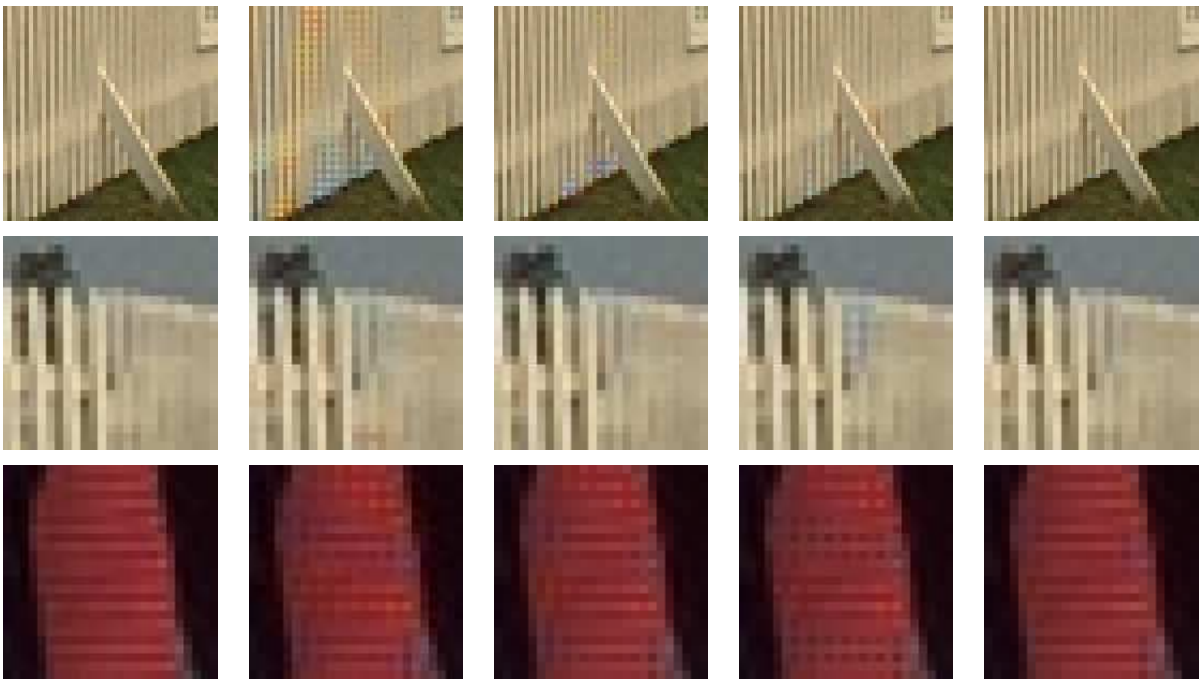
where $\bar{I}$ is the average value within the window Ψ whose size is set to $9x9$, and therefore N=243.

Let us comment on the choice of parameters of our algorithm. The neighborhood size in the merging step is 11×11. In (1): $\alpha = 0.9$ and $\beta = 0.1$. Also, a gap penalty was introduced in the dynamic programming method: whenever the computed cost between two pixels is higher that C_{max}, the pixels will not be matched. The proposed difference functions D_i can be understood as a pixel by pixel comparison of the standard deviation between neighborhoods. Therefore, value C_{max} is defined as the maximum standard deviation allowed between the neighborhoods of two pixels when they belong to the same level

line. This value was set to one grey level per pixel in the comparing window, therefore C_{max} equals N in the distance formulas (3), (6), and (8).

IV. Experiments

The algorithm was tested over the Kodak database (see http://r0k.us/graphics/kodak for the images' index) consisting of 24 images of size 512×768. These full color RGB images were artificially mosaicked with a Bayer CFA, and demosaicked with our algorithm as well as several state of the art methods, for comparison: Gunturk et al. [2], Menon et al. [6], and Chung et al. [7]. We performed both quantitative and qualitative evaluations of the results.



(a) (b) (c) (d) (e)

Fig. 2. Details of results obtained with several algorithms on the images shown in fig. 4 (a) and 4 (c) : **(a)** Original image **(b)** Gunturk et al. **(c)** Menon et al.**(d)** Chung et al. **(e)** Global Geometric Demosaicking algorithm. Note that our results show neither color nor geometric artifacts.

a) Qualitative Evaluation: In figure 2 we show some details of the images shown in fig. 4 (a) and fig. 4 (c) processed using different algorithms. Note that our algorithm demosaicks correctly the areas of the gate in the Lighthouse image and the ribbon in the little girl's image, where the state of the art algorithms produce patterns and color artifacts. The Global Geometric Demosaicking algorithm also performs well when applied to videos, not creating any temporal artifacts (see fig. 4 (e), the whole

video is available at http://www.dtic.upf.edu/∼mbertalmio/demosaicking). However there are some cases in which all algorithms fail, as it is shown in figure 3.

(a) (b) (c) (d) (e)

Fig. 3. An example where all algorithms fail: **(a)** Detail of the original image **(b)** Gunturk et al. **(c)** Menon et al. **(d)** Chung et al. **(e)** Global Geometric Demosaicking method.

b) Quantitative Evaluation: Table I shows the numerical results of quantitative evaluation of the methods using different error measures. For reasons of space we show only the average results over a whole 24-image data set, and the results for just six of the images. All measures where computed ignoring a 12 pixel-wide band around the border of the image:

- Peak signal to noise ratio (PSNR). The results for this measure in table I were computed using $PSNR(I_o, I_d) = 10log_{10}\left(\frac{255 \cdot 255 \cdot N}{\sum_{i=1}^{N}(I_o(i)-I_d(i))^2}\right)$ where I_o is the original image, I_d the demosaicked image and N is the total number of pixels in the image. Note that the measure is inversely proportional to the error, therefore the higher the measure the better the image.
- CIELAB distance. Euclidean distance in the CIE space, and therefore perceptually more significant than PSNR. The measure increases with the error.
- Zipper Effect Measure. The zipper effect (ZE) is one of the most common artifacts in demosaicking and consists in an on-off pattern created in areas with saturated colors (see figure 3). Lu et al. in [12] propose a measure to count the pixels with ZE, and we compute it in table I. Since it is a percentage of pixels with ZE, the best measure would be zero and the worst 100%.

Note that the proposed method improves the others in PSNR and CIELAB measures, and with the Zipper Effect measure our results are comparable to the best method.

V. Conclusions and future work

A new edge based demosaicking method was proposed. Due to its global nature, our algorithm improves state of the art methods both from a qualitative and a quantitative point of view. The only assumption made was that all the color channels of the image have the same edges, i.e. that level lines coincide.

file	Gunturk et al.			Menon et al.			Chung et al.			Global Geometric algorithm		
	PSNR	CIELAB	ZE	PSNR	CIELAB	ZE	PSNR	CIELAB	ZE	PSNR	CIELAB	ZE
Kod15	39.4	2.02	7.94	39.2	2.09	10.36	39.8	2.28	6.7	39	1.89	6.47
Kod18	37.13	3.23	15.55	36.5	3.35	19.69	37.24	3.51	13.43	37.63	3.02	13.78
Kod19	39.3	2.38	8.36	39	2.27	11.07	41.02	2.66	5.23	41.01	2.05	5.6
Kod5	37.55	2.9	15.83	37.32	2.96	18.82	37.95	3.05	12.99	38.10	2.66	12.81
Kod7	41.78	1.77	6.59	41.73	1.71	6.6	42.24	2.08	4.99	42.48	1.6	4.76
Kod8	35.12	3.5	19.28	35.3	3.45	20.73	36.39	3.41	13.08	36.58	2.9	12.82
Average	39.02	2.37	9.54	39.06	2.39	12.36	39.86	2.61	**7.5**	**40**	**2.12**	**7.53**

TABLE I

RESULTS FOR SOME IMAGES AND THE AVERAGE ERROR OF THE COMPLETE KODAK DATA SET FOR SOME METHODS IN THE STATE OF THE ART. THE ERROR MEASURES ARE PSNR, CIELAB, AND ZIPPER EFFECT RESPECTIVELY BY COLUMN.

There is still room for improvement, as some zipper effect artifacts are alleviated but nonetheless present in our results. Also, the algorithm could be implemented on a GPU, increasing its performance in the spirit of [19].

ACKNOWLEDGMENTS

M. Bertalmo and V. Caselles acknowledge partial support by PNPGC project, reference MTM2006-14836. The authors are deeply grateful to King-Hong Chung, Daniele Menon and Bahadir K. Gunturk for providing us with the code of their algorithms. We would also like to thank our colleagues Fabien Girardin, Coloma Ballester and Luis Garrido for their very kindly help.

REFERENCES

[1] R. Ramanath, W. Snyder, Y. Yoo, and M. Drew, "Color image processing pipeline," *Signal Processing Magazine, IEEE*, vol. 22, no. 1, pp. 34–43, 2005.

[2] B. Gunturk, J. Glotzbach, Y. Altunbasak, R. Schafer, and R. Mersereau, "Demosaicking: color filter array interpolation," *Signal Processing Magazine, IEEE*, vol. 22, no. 1, pp. 44–54, 2005.

[3] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.

[4] K. Hirakawa and T. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[5] C. Tsai and K. Song, "Heterogeneity-Projection Hard-Decision Color Interpolation Using Spectral-Spatial Correlation," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 78–91, 2007.

[6] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing With Directional Filtering and a posteriori Decision," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 132–141, 2007.

[7] K. Chung and Y. Chan, "Color demosaicing using variance of color differences." *IEEE Trans. Image Processing*, pp. 2944–55, 2006.

[8] A. Buades, B. Coll, J. Morel, and C. Sbert, "Non local demosaicing," CMLA Preprint 2007-15 (submitted to IEEE Trans. Image Processing), Tech. Rep., 2007.

[9] J. Mairal, M. Elad, and G. Sapiro, "Sparse Representation for Color Image Restoration," IMA Preprint Series no. 2139 (submitted to IEEE Trans. Image Processing), Tech. Rep., 2007.

[10] S. Masnou, "Disocclusion: a variational approach using level lines," *IEEE Trans. Image Processing*, vol. 11, pp. 68–76, 2002.

[11] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by Joint Interpolation of Vector Fields and Grey Levels," *IEEE Trans. Image Processing*, vol. 10, pp. 1200–1211, 2001.

[12] W. Lu and Y. Tan, "Color filter array demosaicking: new method and performance measures," *IEEE Trans. Image Processing*, vol. 12, no. 10, pp. 1194–1210, 2003.

[13] V. Caselles, B. Coll, and J. Morel, "Geometry and Color in Natural Images," *Journal of Mathematical Imaging and Vision*, vol. 16, no. 2, pp. 89–105, 2002.

[14] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[15] R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," *IEEE Trans. Image Processing*.

[16] M. Fornasier, "Nonlinear Projection Recovery in Digital Inpainting for Color Image Restoration," *Journal of Mathematical Imaging and Vision*, vol. 24, no. 3, pp. 359–373, 2006.

[17] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[18] S. Kang and R. March, "Variational Models for Image Colorization via Chromaticity and Brightness Decomposition," *IEEE Trans. Image Processing*, vol. 16, no. 9, p. 2251, 2007.

[19] C. Ballester, B. Bertalmío, V. Caselles, L. Garrido, A. Marques, and F. Ranchin, "An Inpainting-Based Deinterlacing Method," *IEEE Trans. Image Processing*, vol. 16, no. 10, pp. 2476–2491, 2007.

[20] M. Bertalmío, V. Caselles, and A. Pardo, "Movie Denoising by Average of Warped Lines," *IEEE Trans. Image Processing*, vol. 16, no. 9, p. 2333, 2007.

[21] V. Caselles, B. Coll, and J. Morel, "Topographic Maps and Local Contrast Changes in Natural Images," *International Journal of Computer Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[22] D. Mumford, "Elastica and Computer Vision," in *Algebraic geometry and its applications*, C. Bajaj, Ed. Springer-Verlag, 1994, pp. 491–506.

[23] I. J. Cox, S. L. Hingorani, and S. B. Rao, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 542–567, May 1996.

[24] L. Chang and Y. Tan, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Trans. Consumer Electronics,*, vol. 50, no. 1, pp. 355–365, 2004.

(a) (b)

(c)

(d)

(e)

Fig. 4. **(a)**, **(c)** Original images: Kod19, Kod15 respectively. **(b)**,**(d)** Images processed with the global geometric method. **(e)** Several frames of the video processed with the global geometric method. Note that the algorithm performs well despite the highly-textured nature of the images. The whole video is available at http://www.dtic.upf.edu/∼mbertalmio/demosaicking