Machine Vision and Applications (1999) 11: 197–202

Machine Vision and Applications

Restoring subsampled color images

Daniel Keren, Margarita Osadchy

Department of Computer Science, The University of Haifa, Haifa 31905, Israel; e-mail: dkeren@cs.haifa.ac.il

Abstract. In some capturing devices, such as digital cameras, there is only one color sensor at each pixel. Usually, 50% of the pixels have only a green sensor, 25% only a red sensor, and 25% only a blue sensor. The problem is then to restore the two missing colors at each pixel – this is called "demosaicing", because the original samples are usually arranged in a mosaic pattern. In this short paper, a few demosaicing algorithms are developed and compared. They all incorporate a notion of "smoothness in chroma space", by imposing conditions not only on the behavior of each color channel separately, but also on the correlation between the three channels.

Key words: Demosaicing – Color interpolation – Regularization

1 Introduction

With the advent of color technology, more and more of the images which are captured, copied, transmitted, and processed, are color images. Naturally, this results in a growing interest in extending various algorithms for gray-level images, to color images. Various researchers have discovered that applying standard deblurring or enhancement algorithms separately to every color channel is not optimal; better results can be obtained in view of the fact that, in natural images, there are strong correlations between the channels [6, 7, 9–12, 14–17, 20]. In this paper, the so-called "demosaicing" problem is addressed; this problem is relatively new, and is motivated by the fact that, in digital cameras and other capturing devices, there is often only one color sensor at each pixel. Thus, the missing colors have to be "filled in". The array of sensors is often called a "mosaic", hence the name "demosaicing" for the task of filling in the missing colors. The following algorithms are introduced and studied.

– *Regularization with a "color correlation term"*. This algorithm resembles ordinary regularization, however, an additional term is introduced to account for the between-channel correlations, using the estimated local averages and covariance matrices.
– *The local MAP algorithm*, which also uses the estimated local averages and covariance matrices of the color channels. Here, the missing colors are filled in simply by choosing, at each pixel, values which maximize the probability with regard to the Gaussian distribution induced by the local average and covariance matrices. The algorithm can be improved by segmenting each neighborhood before the local averages and covariance matrices are computed; this improves the results, particularly in areas of strong discontinuities between uniform regions.
– *Processing in the angle domain.* This is the simplest among the suggested algorithms, and consists of representing the color image in spherical coordinates, and then filtering it in the new representation; we tried both standard regularization and median filtering. However, only the angle components of the spherical representation are regularized. This allows preservation of luminance discontinuities in the image, while smoothing it in chroma space.
– *Regularization with a vector-product-based between-channel term*. Here, correlation between the channels is forced by minimizing not only the standard regularization smoothness term, but also a smoothness term which consists of the sum of the squared norms of vector products between neighboring pixels, where the color values are viewed as vectors in $\mathcal{R}^3$. The advantage of this algorithm is that it does not require any estimation of the between-channel correlation; also, it gives rather good results, and preserves image discontinuities. The vector-product-based term also proved useful for denoising color images [12].

1.1 Previous work

Most previous research on demosaicing took place in industry, due to the great importance of demosaicing in color-capturing devices. In [5], the missing color values are filled in by linear interpolation; then, the color differences are median filtered and the image restored according to the new

Some of the algorithms in this paper are included in a patent application to the US Patent Office.

Correspondence to: D. Keren

values, which reduces "color fringes" near edges. In [2], the missing green values are first filled in by a simple interpolation, and then the other missing values are filled, in a manner which forces the ratios between the different color channels to change slowly between neighboring pixels. In [3], it is suggested to first segment the image, based on the green (luminance) values, and fill in the missing colors in each region independently. In [19], the optimal approximation (in the mean square error sense) of the missing colors as linear combinations of the measured ones is derived, and used to interpolate the missing colors. In [1], the Bayesian restoration paradigm is extended to color images, by assuming a prior distribution over color images which favors positive correlation between the different channels. In [4], a nonlinear algorithm is suggested, which locates structures such as edges in the image, and tries to preserve discontinuities during the demosaicing process.

2 Demosaicing algorithms studied in this paper

First, let us formulate the problem in more exact terms. As noted, many capturing devices have only one color sensor at each pixel; a standard "mosaic" of sensors is the following (Bayer pattern); here, and in the sequel, RGB will stand for red, green, and blue, respectively.

R	G	R	G	R
G	B	G	B	G
R	G	R	G	R
G	B	G	B	G
R	G	R	G	R

One obvious way to restore the image is to perform standard single-channel restoration on each channel, and combine the restored channels into a color image. Such an approach, however, does not work well in general. Usually, the resulting image is contaminated with false colors (also referred to as "color artifacts"); that is, certain areas contain streaks of colors which do not exist in the original image. This problem is more acute in highly textured areas, or image discontinuities; see Figs. 1, 2, and 3.

2.1 Regularization with "color correlation term"

Regularization has proven quite successful in image restoration and other computer vision problems. Some general references are [13, 18]. Regularization usually tries to minimize a "cost functional" of the restored image, which is composed of a "data fidelity term", measuring the compatibility of the image with the measured data, and a "quality term", which reflects some prior assumption about the image. The latter is usually defined as a "spatial smoothness term", which is small for smooth images, but attains large values on "rough" and noisy images. A typical example of how regularization is applied is the following: if an image was measured at the locations $\{(x_i, y_i)\}$ with the measurement values being $\{z_i\}$, then the restored image is the one minimizing the cost functional

Fig. 1. Real image of vending machine. Results are presented for Kodak algorithm for the DCS 200 Camera (*upper*), simple restoration (*lower left*), and the algorithm suggested in Sect. 2.4 (*lower right*). The real-world image is gray level. Note the very strong color artifacts in areas with discontinuities, which are greatly reduced by using the algorithm of Sect. 2.4

$$\sum_{i=1}^{n} \frac{[F(x_i, y_i) - z_i]^2}{2\sigma^2} + \lambda \iint (F_{xx}^2 + 2F_{xy}^2 + F_{yy}^2) dx dy, \quad (1)$$

where σ is the noise variance, and λ a positive constant, which can be chosen by various methods [10, 20]. Here, we assume that there is very little noise in the capturing device, so in the sequel we will omit the first summand in Eq. 2; however, all the algorithms suggested in this short paper can be directly extended to deal with measurement noise [12].

Regularization has also been extended to restoring color images [9, 10, 12, 17, 20].

One possibility of extending the cost functional to color images is to estimate, for each pixel, the probability for its combination of colors. Formally, if $C_{x,y}$ is the covariance matrix of the RGB values at a pixel (x, y), then, assuming a Gaussian distribution, the probability for the combination of colors $(R(x, y), G(x, y), B(x, y))$ is proportional to

$$\exp\Big(-\frac{1}{2}(R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B}) \times C_{x,y}^{-1}(R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B})^t, \quad (2)$$

where $(\bar{R}, \bar{G}, \bar{B})$ are the average colors in the pixel's vicinity. This suggests adding the following expression ("color correlation term")

$$\iint (R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B}) \times C_{x,y}^{-1}(R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B})^t dx dy, \quad (3)$$

to the functional of Eq. 1. This term assigns lower values to combinations of colors which are consistent with the local color distribution. Thus, the combined expression to minimize is

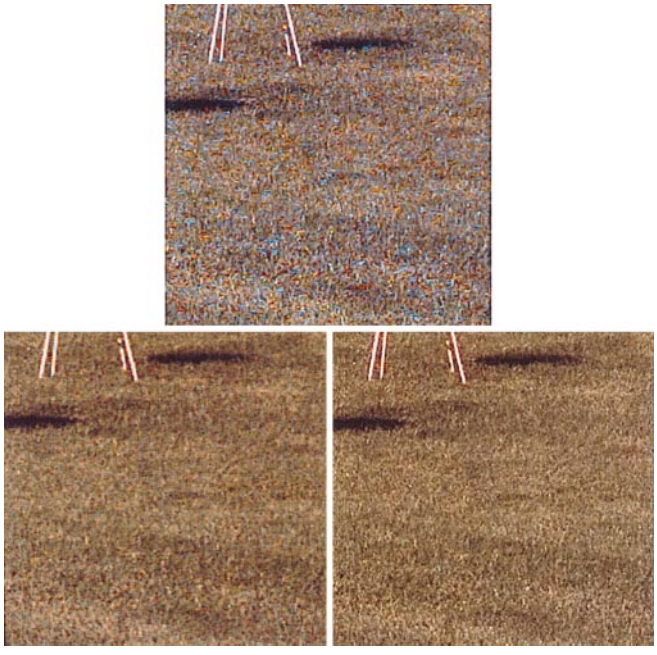

Fig. 2. Grass scene. Results are presented for Kodak algorithm for the DCS 200 Camera (*upper*), simple restoration (*lower left*), and the algorithm suggested in Sect. 2.4 (*lower right*). We used the same color correction as the Kodak algorithm used. Again, it can be seen that the algorithm of Sect. 2.4 greatly reduces the color artifacts

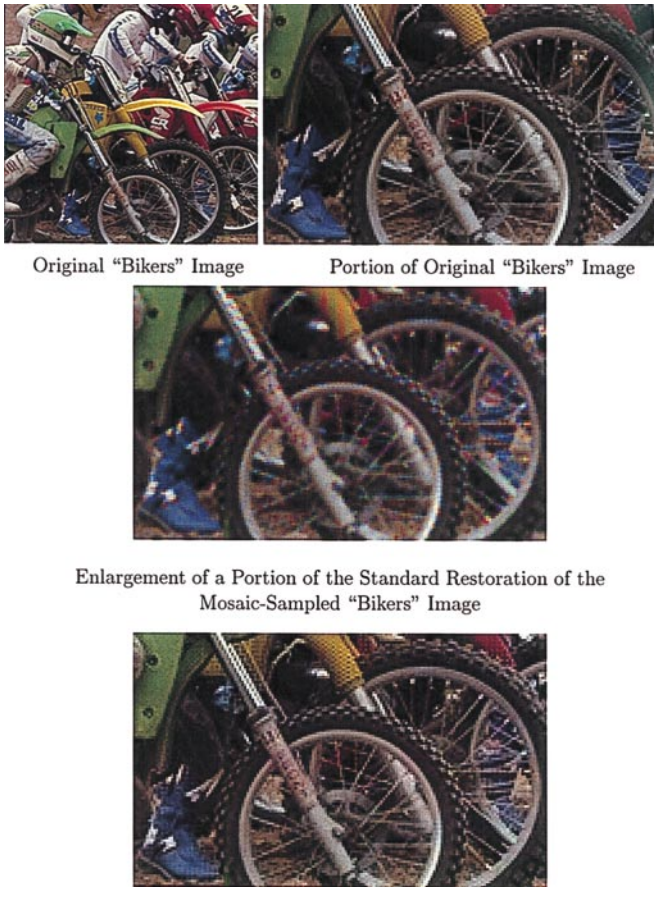

Fig. 3. Synthetic image, one of those used to estimate the MSE restoration error of the suggested algorithms. Note very strong color artifacts in some of metallic parts in the simple restoration, and how the algorithm of Sect. 2.4 succeeds to restore them

$$\iint (R_{xx}^2 + 2R_{xy}^2 + R_{yy}^2 + G_{xx}^2 + 2G_{xy}^2 + G_{yy}^2 + B_{xx}^2 + 2B_{xy}^2 + B_{yy}^2)dxdy + \lambda \iint (R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B}) \times C_{x,y}^{-1}(R(x,y)-\bar{R}, G(x,y)-\bar{G}, B(x,y)-\bar{B})^t dxdy, \quad (4)$$

subject, of course, to the measurements obtained from the digital camera. We used a simple finite-element method to minimize this cost functional. It results in an iterative scheme, which is applied only to the colors that have to be filled in. For instance, if we know the red value at a pixel (because it coincided with a camera pixel that has a red sensor), we update only its green and blue values at each iteration.

In order to run this iterative scheme, we have to know the local averages and color correlation matrices. These cannot be estimated directly from the original image, since it only has one color per pixel. To overcome this, a simple bootstrapping technique is used: each channel is restored separately, thus resulting in three color channels at each pixel. These are then used to compute $C_{x,y}$ and $(\bar{R}, \bar{G}, \bar{B})$.

Usually, ten or so iterations are enough, and the image does not change any more. Using the estimated local averages and covariance matrices from the new images, and running the iterative scheme again, did not improve the result.

2.2 Local MAP algorithm

This algorithm, like the one described in Sect. 2.1, first estimates the local averages and covariance matrices from a standard restoration carried out separately in each color channel. Then, it fills in the missing colors at every pixel P with the values that maximize the probability of P, given the estimated local averages and covariance matrix, and the (single) color channel at P whose value is known from the measurements. It may be therefore viewed as a "local maximum a priori probability", or "local minimal square error", approach.

Suppose, for instance, that the red value is known, and the green and blue values have to be filled in. Also given are the estimates for the local averages $(\bar{R}, \bar{G}, \bar{B})$ and covariance matrix $C_{x,y}$. Assuming, as before, a Gaussian distribution, the probability of the color vector $(R(x,y), G(x,y), B(x,y))$ is given by Eq. 2, and it is easy to see that the values of $G(x,y)$ and $B(x,y)$, which maximize this probability (for a given $R(x,y)$), are

$$G(x,y) = \frac{c_{11}\bar{G} + c_{12}(R(x,y) - \bar{R})}{c_{11}},$$

$$B(x,y) = \frac{c_{11}\bar{B} + c_{13}(R(x,y) - \bar{R})}{c_{11}}.$$

2.2.1 An improvement to the local MAP approach: vector quantization

The success of the local MAP algorithm depends to a large extent on whether the estimates for the local averages and

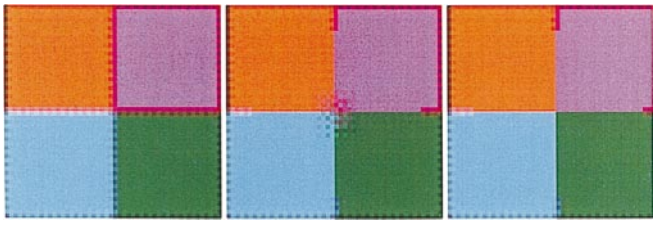

Fig. 4. Synthetic image with uniform regions separated by discontinuities. Simple restoration (*left*), local MAP algorithm (*middle*), and local MAP with vector quantization (*right*)

covariance matrix at a pixel P are accurate. If there are strong discontinuities near P, these estimates may be influenced by pixels whose colors are very different from P's, which is undesirable.

To solve this problem, the neighborhood of each pixel P is segmented into two regions (one may perform this segmentation only if a certain test for the neighborhood's homogeneity fails). Then, the local average and covariance matrix at P are estimated using only the pixels in the largest region. Since the pixels' natural representation is as vectors in 3D color space, vector quantization [8] was used to segment them. The improvement when using the vector quantization is less noticeable in natural images, and more in images which have relatively uniform regions separated by strong discontinuities (see Fig. 4).

Our tests indicated that the local MAP algorithm gave results of the same quality as the algorithm presented in Sect. 2.1, but in shorter running time. The somewhat surprising result is that the lack of a spatial regularization term, which was used in Sect. 2.1, hardly affects the results. This is probably due to the fact that spatial smoothness is implicitly forced via the covariance matrices, which are close for neighboring pixels, since they are computed on intersecting neighborhoods.

2.3 Processing in the angle domain

This is conceptually the simplest algorithm suggested in this short paper. It attempts to remove color artifacts by forcing adjacent pixels to have similar directions in 3D Euclidean space $\mathcal{R}^3$. The intuition is straightforward – color artifacts are usually caused by adjacent pixels with different chromas, so it is natural to try and prevent this from happening. It is important to note that spatial smoothness alone is not enough to detect different chromas; the difference between spatial smoothness and chroma smoothness can be demonstrated by a simple example [12]. Suppose P_1 and P_2 are adjacent pixels. In case (a), let the color values of P_1 be (r, g, b), and those of P_2 $(r(1+\delta), g(1+\delta), b(1+\delta))$, for some small constant δ. In case (b), the values are (r, g, b) and $(r(1+\delta), g(1-\delta), b(1+\delta))$, respectively. In both cases, the spatial smoothness, as measured separately in the three color channels, is the same; however, in case (b), the clique of P_1 and P_2 is less smooth in chroma space. This is because, in case (a), the orientations of P_1 and P_2 in RGB space are identical, but are different in case (b).

The following simple algorithm was used to force smoothness in chroma space. First, standard regularization is applied separately to each color channel. Then, each pixel is represented in spherical coordinates $R = \rho\cos(\theta)\sin(\phi), G = \rho\cos(\theta)\cos(\phi), B = \rho\sin(\theta)$, and then some filtering – regularization, or median filtering – is applied to the two angles ϕ and θ only. This allows the preservation of luminance discontinuities between neighboring pixels, while forcing their chromas to be similar; this is because ρ is closely related to the luminance component of the image. After each iteration of the filtering process, the image is transformed back to RGB space, and the colors measured by the camera are set back to their original values.

2.4 Regularization with a vector-product-based between-channel term

This algorithm combines standard regularization with a novel method for forcing a correlation between the color channels. The missing colors are restored in a manner which forces neighboring pixels to have similar directions in color space; this is the same idea as in Sect. 2.3, however, here, chroma smoothness and spatial smoothness are merged into a single cost functional (see Sect. 2.1), and treated in a uniform manner.

The algorithm presented in this section results both in sharp images and in practically eliminating the "color artifacts" mentioned in Sect. 2.1. The solution is given as a closed-form expression, which is iteratively applied to the image. The technical details follow; they are elaborated upon in some detail, as the functional to be optimized is not standard, due to the appearance of fourth-order powers in it. We also chose to explain the iterative scheme in some detail, as it gave the best results among all the algorithms suggested and tested in this short paper.

Let P be a pixel in the image, and let us look at a 5×5 square around P. Denote this neighborhood by N_{22}, and index its pixels with indexes running from $(0,0)$ to $(4,4)$, where the index of P is $(2,2)$:

	0	1	2	3	4
0	r_{00},g_{00},b_{00}	r_{01},g_{01},b_{01}	r_{02},g_{02},b_{02}	r_{03},g_{03},b_{03}	r_{04},g_{04},b_{04}
1	r_{10},g_{10},b_{10}	r_{11},g_{11},b_{11}	r_{12},g_{12},b_{12}	r_{13},g_{13},b_{13}	r_{14},g_{14},b_{14}
2	r_{20},g_{20},b_{20}	r_{21},g_{21},b_{21}	$P = (r_{22},g_{22},b_{22})$	r_{23},g_{23},b_{23}	r_{24},g_{24},b_{24}
3	r_{30},g_{30},b_{30}	r_{31},g_{31},b_{31}	r_{32},g_{32},b_{32}	r_{33},g_{33},b_{33}	r_{34},g_{34},b_{34}
4	r_{40},g_{40},b_{40}	r_{41},g_{41},b_{41}	r_{42},g_{42},b_{42}	r_{43},g_{43},b_{43}	r_{44},g_{44},b_{44}

Next, define two scalar functions of the 25 pixels in N_{22}. The first, denoted by $\mathcal{ROUGH}$, measures the amount of roughness in N_{22}; the larger $\mathcal{ROUGH}$ is, the less smooth the image. $\mathcal{ROUGH}$ is defined as

$$\iint (R_{xx}^2 + 2R_{xy}^2 + R_{yy}^2 + G_{xx}^2 + 2G_{xy}^2 + G_{yy}^2 + B_{xx}^2 + 2B_{xy}^2 + B_{yy}^2)dxdy.$$

The discrete analog of the integral is obtained in a standard manner, by representing the derivatives as finite differences (see, for instance, [18]). It is desired that $\mathcal{ROUGH}$ be small, for this results in an image which is nicely behaved in the spatial domain.

Next, define another scalar function of the pixels in N_{22}, which is called the $\mathcal{CHROMA}-\mathcal{TERM}$. This function

measures what may be described as smoothness in chroma space. It will be smaller when adjacent pixels have similar directions in (R, G, B) space. Formally, the $\mathcal{CHROMA}$–$\mathcal{TERM}$ is defined as follows. For each pixel (viewed as a vector in $\mathcal{R}^3$), compute its vector products with all its neighbors; this gives eight vectors. Then, add the squared norms of all these vectors, and this sum (when, of course, added over all the pixels in N_{22}) constitutes the $\mathcal{CHROMA}-\mathcal{TERM}$. Thus, a typical summand is $\|(r_{22}, g_{22}, b_{22}) \times (r_{21}, g_{21}, b_{21})\|^2$. The intuition behind this definition is simple. If adjacent pixels have similar directions in color space, and hence similar chromas, the vector product's norm will be small, as it behaves like $\sin(\alpha)$, where α is the angle between the vectors. Conversely, chroma smoothness can be forced by minimizing this norm. In [12], a more elaborate explanation as to why the vector product's norm is a good quality measure is provided (it is shown how it is related to the surface area of the color image, when embedded in 5D Euclidean space; see [14–16].

Next, define the cost functional $\mathcal{COST}$, as a weighed sum of the two terms defined above. It is desired that it will be small, since this will mean that the image is nicely behaved both in the spatial domain and in chroma space:

$$\mathcal{COST} = \mathcal{ROUGH} + \lambda(\mathcal{CHROMA}-\mathcal{TERM}),$$

where λ is a positive constant. $\mathcal{COST}$ is minimized by a standard finite-element method, using Gauss-Seidel iterations on the set of equations $\{\frac{\partial(\mathcal{COST})}{\partial\rfloor_{ij}} = 0\}$, where c_{ij} stands, in turn, for all the colors in every pixel. For instance, the iterative step for r_{22} is defined as follows (assuming it was not measured by the camera): if $r_{ij}^{(n)}, b_{ij}^{(n)}, g_{ij}^{(n)}$ denote the current values of the red, green and blue values in the (i, j)-th pixel in the neighborhood N_{22}, the value assigned in the next iteration to P's red channel is

$$\begin{aligned}
r_{22}^{(n+1)} = (&r_{02}^{(n)} + r_{42}^{(n)} + r_{20}^{(n)} + r_{24}^{(n)} + \lambda\, r_{11}^{(n)}\, b_{22}^{(n)}\, b_{11} \\
&+\lambda\, r_{11}^{(n)}\, g_{22}^{(n)} 22\, g_{11}^{(n)} + \lambda\, r_{21}^{(n)}\, b_{22}\, b_{21}^{(n)} + \lambda\, r_{21}^{(n)}\, g_{22}^{(n)}\, g_{21}^{(n)} \\
&+\lambda\, r_{31}^{(n)}\, b_{22}^{(n)}\, b_{31}^{(n)} + \lambda\, r_{31}^{(n)}\, g_{22}^{(n)}\, g_{31}^{(n)} \\
&+\lambda\, r_{12}\, b_{22}^{(n)}\, b_{21}^{(n)} + \lambda\, r_{12}^{(n)}\, g_{22}^{(n)}\, g_{12}^{(n)} \\
&+\lambda\, r_{32}^{(n)}\, b_{22}^{(n)}\, b_{32}^{(n)} \\
&+\lambda\, r_{32}^{(n)}\, g_{22}^{(n)}\, g_{32} + \lambda\, r_{13}^{(n)}\, b_{22}^{(n)}\, b_{13}^{(n)} \\
&+\lambda\, r_{13}^{(n)}\, g_{22}^{(n)}\, g_{13}^{(n)} + \lambda\, r_{23}^{(n)}\, b_{22}^{(n)}\, b_{23}^{(n)} \\
&+\lambda\, r_{23}^{(n)}\, g_{22}^{(n)}\, g_{23}^{(n)} + \lambda\, r_{33}^{(n)}\, b_{22}^{(n)}\, b_{33}^{(n)} + \lambda\, r_{33}\, g_{22}^{(n)}\, g_{33}^{(n)})/ \\
&(\lambda\, b_{12}^{(n)^2} + \lambda\, g_{12}^{(n)^2} + \lambda\, b_{11}^{(n)^2} + \lambda\, g_{11}^{(n)^2} + \lambda\, b_{21}^{(n)^2} \\
&+\lambda\, g_{21}^{(n)^2} + \lambda\, b_{31}^{(n)^2} + \lambda\, g_{31}^{(n)^2} \\
&+\lambda\, b_{32}^{(n)^2} + \lambda\, g_{32}^{(n)^2} + \lambda\, b_{13}^{(n)^2} \\
&+\lambda\, g_{13}^{(n)^2} + \lambda\, b_{23}^{(n)^2} + \lambda\, g_{23}^{(n)^2} + \lambda\, b_{33}^{(n)^2} + \lambda\, g_{33}^{(n)^2} + 4).
\end{aligned}$$

The expressions for updating the green and blue values are derived similarly. These iterations are repeated over the entire image, with each pixel in its turn assigned the role of P in the iterative process, and N_{22} replaced by the neighborhood of the pixel which is now updated. The starting point for the iterations is the image consisting of the three independently restored channels.

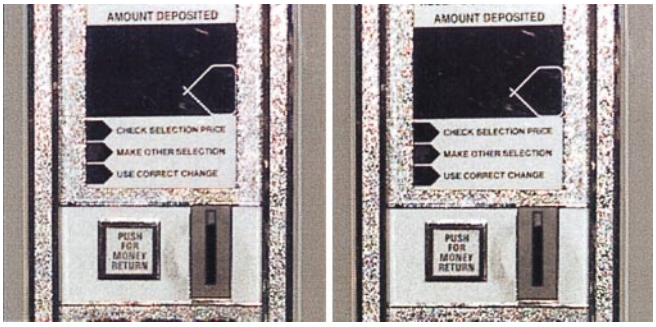

Fig. 5. Vending machine (same image as in Fig. 1) restored using local MAP algorithm (*left*) and processing in the angle domain (*right*)

Note that, although the process is described for each pixel separately, a global cost function is minimized (which consists of the sum over all pixel neighborhoods of the distinct pixels).

Due to the density of the measurements, the convergence is fast (4–5 iterations will do). It turns out that the value of λ hardly makes a difference, once it is large enough; a value of 1.0, for all images on which the algorithm was tested, gave satisfactory results. We are currently investigating other methods for choosing λ.

3 Results

We present here some results on real and synthetic images. A few images are used, to highlight the performance of the different algorithms. The real images were captured with a Kodak DCS 200 digital camera (which allows extraction of the raw data, as well as the restored, uncompressed image), and then demosaiced both according to the Kodak off-line algorithm and some of the algorithms presented in this short paper. The Kodak algorithm usually gave good results, however, it sometimes gave unsatisfactory results on images with strong high frequencies (see Figs. 1 and 2). Synthetic images were used to compare the algorithms of Sects. 2.1–2.4 in terms of the MSE restoration error (this could not be done with the captured images, as, for those, we do not have the "ground truth"). The algorithm of Sect. 2.4 was the winner in this category, with the exception of the image in Fig. 4, for which the local MAP algorithm with vector quantization gave the best results. On the aggregate of images we have tested, the average MSE error was 26.4 for the "simple restoration" (that is, independent regularization in every channel), 21.3 for the local MAP algorithm, 20.3 for the local MAP algorithm with vector quantization, 17.1 for processing in the angle domain, and 14.9 for regularization with a vector product based term. The latter also gave results which were, in general, superior in appearance.

References

1. Brainard DH (1994) Bayesian method for reconstructing color images from trichromatic samples. In: Proceedings of the IS and T Annual Meeting, pp 375–379
2. Cok DR (1987) Signal processing method and apparatus for producing interpolated chrominance values in a sampled color images signal. US Patent 4,642,678

3. Cok DR (1991) Method of processing sampled signal values produced by a color imaging device. US Patent 5,040,064
4. Cok DR (1994) Reconstruction of CCD images using template matching. In: Proceedings of the IS and T Annual Meeting, pp 380–385
5. Freeman WT (1988) Method and apparatus for recontructing missing color samples. US Patent 4,774,565
6. Galatsanos NP, Chin RT (1991) Restoration of color images by multichannel Kalman filtering. IEEE Trans Signal Process (IP)-39.10:2237–2252
7. Galatsanos NP, Katsagellos AT, Chin RT, Hillery AD (1991) Least squares restoration of multichannel images. IEEE Transaction on Acoustic, Speech, Signal Processing 39.10:2222–2236
8. Gersho A, Gray RM (1992) Vector Quantization and Signal Compression. Kluwer Academic Publishers, Amsterdam
9. Kang MG, Katsaggelos AG (1995) General choice of the regularization functional in regularized image restoration. IEEE Trans Image Process 4:594–602
10. Kang MG, Katsaggelos AG (1997) Simultaneous multichannel image restoration and estimation of the regularization parameters. IEEE Trans Image Process 6.5:774–778
11. Katsaggelos AK, Lay KT, Galatsanos NP (1993) A general framework for frequency domain multi-channel signal processing. IEEE Trans Image Process 2.3:417–420
12. Keren D, Gotlib A (1998) Denoising color images using regularization and correlation terms. J Visual Commun Image Represent 9:352–365
13. Keren D, Werman M (1993) Probabilistic analysis of regularization. IEEE Trans Pattern Anal Mach Intell 15:982–995
14. Kimmel R (1998) A natural norm for color processing. In: 3rd Asian Conference on Computer Vision, Hong Kong, 8–11 January 1998
15. Kimmel R, Malladi R, Sochen N (1997) Images as embedding maps and minimal surfaces: Movies, color, and volumetric medical images. In: IEEE CVPR, Puerto Rico, June 1997
16. Sapiro G, Ringach DL (1996) Anisotropic diffusion of multivalued images with applications to color filtering. IEEE Trans Image Proces 5:1582–1586
17. Schultz RR, Stevenson RL (1995) Stochastic modeling and estimation of multispectral image data. IEEE Trans Image Process 4.8:1109–1119
18. Terzopoulos D (1984) Multi-level surface reconstruction. In: Rosenfeld A (ed) Multiresolution Image Processing and Analysis. Springer-Verlag, Berlin Heidelberg New York
19. Wober MA, Soini R (1995) Method and apparatus for recovering image data through the use of a color test pattern. US Patent 5,475,769
20. Zhu W, Galatsanos NP, Katsaggelos AK (1995) Regularized multichannel restoration using cross-validation. Graph Models Image Process 57.1:38–54