Adaptive Color Demosaicing and False Colors Removal

Mirko Guarnera, Giuseppe Messina, and Valeria Tomaselli*
Advanced System Technology Group
STMicroelectronics
Stradale Primosole 50,
95121 Catania, Italy

Abstract

Color interpolation solutions drastically influence the quality of the whole image generation pipeline, so they must guarantee the rendering of high quality pictures avoiding typical artifacts such as blurring, zipper effects and false colors. Moreover, demosaicing should avoid emphasizing typical artifacts of real sensors data, such as noise and green imbalance effect, which would be further accentuated by the subsequent steps of the processing pipeline. In this paper we propose a new adaptive algorithm, which decides the interpolation technique to apply to each pixel, according to its neighborhood analysis. Edges are effectively interpolated through a directional filtering approach which interpolates the missing colors selecting the suitable filter depending on the edge orientation. Regions close to edges are interpolated through a simpler demosaicing approach. Thus flat regions are identified and heavily low pass filtered in order to eliminate some residual noise and to minimize the annoying green imbalance effect. Finally an effective false colors removal algorithm is used as a post-processing step, in order to eliminate residual color errors. The experimental results show how sharp edges are preserved whereas undesired zipper effects are reduced, improving the edge resolution itself and obtaining superior image quality.

Keywords: Color Interpolation, False Color Removal.

*Electronic address: (mirko.guarnera,giuseppe.messina,valeria.tomaselli)@st.com

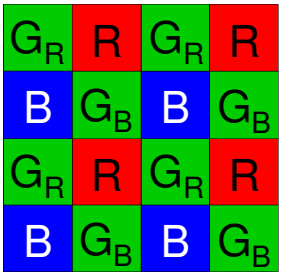


FIG. 1: Bayer Pattern

I. INTRODUCTION

Imaging consumer, prosumer and professional devices, such as digital still and video cameras, mobile phones, personal digital assistants, visual sensors for surveillance and automotive applications, usually capture the scene content by means of a single sensor (CCD or CMOS), covering its surface with a Color Filter Array (CFA), thus significantly reducing costs, sizes and registration errors. The most common arrangement of spectrally selective filters is known as Bayer pattern [1]. Since each image sensing element can only detect one color of illumination, the sensor provides a grayscale image, which then undergoes a color interpolation process to reconstruct the full resolution image.

Traditional color interpolation methods usually result in color edge artifacts in the image, due to the non-ideal sampling performed by the CFA. Given an input image *I(x,y)*, let *I(m,n)* be its CFA image. From the sampling theory it is well known that the Fourier transform of the sampled image *I(m,n)* contains scaled, periodic replications of the Fourier transform of the original image *I(x,y)* [2]. If the sampling is ideal, the repetitions do not overlap each other, thus the image *I(x,y)* can be faithfully reconstructed, otherwise the aliasing phenomenon occurs. The non-overlapping constraint implies a limited band, thus a not limited space, which cannot be performed on sensors . For this reason, in real cases, aliasing effects appear. The aliasing arises with false patterns or colors, and happens when the image frequencies exceed the sampling one. The two main types of demosaicing artifacts are usually named false colors and zipper effect. False colors are evident color errors which arise near the object boundaries, whereas zipper effect artifacts manifest as "on-off" patterns and are caused by an erroneous interpolation across edges. The green channel is less affected by aliasing than the red and blue channels, because it is sampled at a higher rate.

Color interpolation techniques should be implemented by considering the artifacts introduced by the sensor and the interactions with the other modules composing the image processing pipeline, as it has been well analyzed in [3]. This means that demosaicing approaches have to guarantee the rendering of high quality pictures avoiding typical artifacts, which could be emphasized by the sharpening module, thus drastically deteriorating the final image quality. In the meantime, demosaicing should avoid introducing false edge structures due to residual noise (not completely removed by the noise reduction block) or green imbalance effects. Green imbalance is a mismatch arising at G_R and G_B locations. This effect is mainly due to crosstalk [4].

A wide variety of papers and patents has been produced about color interpolation in the last years, exploiting a lot of different approaches. A dissertation of some of them can be found in [5]. As it is disclosed in [6], a well performing demosaicing technique has to

exploit two types of correlation: spatial correlation and spectral correlation. According to the first principle, within a homogeneous image region, neighboring pixels have similar color values, so a missing value can be retrieved by averaging the pixels which belong to the same object. Spatial correlation is well exploited by those techniques which interpolate missing information along edges, and not across them [7, 8]. Spectral correlation states that there is a high correlation between the three color channels (R, G, B) in a local image neighborhood. For real world images the color difference planes (Δ_{GR}=*G*-*R* and Δ_{GB}=*G*-*B*) are rather flat over small regions, and this property is widely exploited in demosaicing and antialiasing techniques.

The algorithms exploiting spatial correlation are less affected by zipper effect, which usually appears when directional information is disregarded. Among the most recent and interesting techniques belonging to this class, we mention the techniques disclosed in [9–11]. Wu and Zhang in [9] interpolate a single image multiple times, under different hypotheses on edge or texture directions; then the final estimate is chosen in the full resolution color image, according to the reconstruction which better preserves the highest correlation of the color gradients. The technique in [11], similarly to the previous one, reconstructs the original CFA image by both a horizontal and a vertical interpolation process; at the end, the decision between the two estimates is performed according to the local homogeneity of the image. Menon et al. in [10] propose a technique where the decision between two differently interpolated green estimates is achieved on a window basis, thus reducing the complexity of the algorithm.

The paper [12] proposes a color interpolation method, which expands the gradient based schemes by adopting Gaussian low-pass filter. It starts from the assumption that high frequency components of the green plane in an image are almost the same as those of the red or blue planes in the same image. After reconstructing the green plane, the red and blue planes are interpolated and then refinement is performed by median processing which considers edge direction.

To reduce aliasing, Gunturk et al. in [13] use the property according to which high frequency components of the three color planes are highly correlated; they demonstrated this characteristic by calculating the correlation values between the detail wavelet coefficients. In addition to this, Lian et al. in [14] have reported that the detail wavelet coefficients are also very similar to each other. This suggests that if there is a strong edge in the R channel, there is usually a strong edge at the same location in the G and B channels.

The algorithm proposed in [15] removes image blurs caused by an optical low-pass filter by using the Total Variation (TV) regularization approach. This method interpolates color signals while preserving their sharp color edges and, in addition to the TV regularization of each primary color signal, the TV regularization of color differences is performed, achieving more accurate color interpolation near sharp color edges, reducing the zipper and aliasing artifacts.

Although demosaicing solutions aim to eliminate false colors, some artifacts still remain. Thus imaging pipelines often include a post-processing module, with the aim of removing residual artifacts [16]. Post-processing techniques are usually more powerful in achieving false colors removal and sharpness enhancement, because their inputs are fully restored color images. Moreover, they can be applied more than once, in order to meet some quality criteria. For obtaining better performances, the antialiasing step often follows the color interpolation process, as a postprocessing step. Freeman's method [17] uses median filtered inter-channel differences to force pixels with distinct colors to be more similar to their

neighbors, avoiding false colors. The CFA color value at each pixel is not altered, and hence sharp edges are preserved. On the other side, it does not remove all the artifacts introduced by conventional color interpolation techniques, and moreover, it often produces an annoying zipper effect along horizontal and vertical directions. On the contrary, Lu and Tan's approach [16], taking into account the spectral correlation between color planes, removes more false colors and artifacts than Freeman's method, but it considerably blurs images, because it adjusts all three colors of each pixel without maintaining the CFA color samples unchanged.

The color correction method proposed in [18], based on Lu and Tan's technique, updates the R,G,B values adaptively; in particular two green estimates are calculated, starting from the two color difference domains (G-R and G-B), and then the final estimate is produced weighted-averaging these two values depending on their variances. Moreover, the paper [18] explicitly addresses the problem of noise in real sensors data, proposing to achieve noise reduction and color interpolation in a single step. The same idea is expressed by Hirakawa and Parks in [19].

We present a new adaptive demosaicing algorithm, with the aim of not magnifying the effect of noise, which is particularly visible in flat regions. The proposed technique decides among three different interpolation approaches, depending on the statistical characteristics of the neighborhood of the pixel to be interpolated. Edges are effectively interpolated through a directional filtering approach, which interpolates the missing colors selecting the suitable filter depending on the edge orientation. Regions close to edges are filtered through a simpler filter. Flat regions are identified and heavily low pass filtered in order to eliminate some residual noise and to minimize the green imbalance issue. A new powerful false colors removal algorithm has been also developed and used as a post-processing step, in order to eliminate residual color artifacts. The experimental results show how sharp edges are preserved, false colors and zipper effects are drastically reduced without accentuating noise.

The remainder of this paper is organized as follows. In Section II, the proposed color interpolation approach, based on directional and texture analysis, is presented. In Section III, a novel false color removal approach based on variances of channel differences is presented. Experimental results are reported in Section IV. Finally, some conclusions are addressed in Section V.

II. COLOR INTERPOLATION

As already outlined, the proposed color interpolation approach adaptively chooses the reconstruction methodology to recover the missing color information, upon the statistical characteristics of the surrounding pixels. As it is visible in Fig. 2 , the method is compound of three main steps:

1. The *direction estimation* stage computes the direction to be used in the interpolation step, if needed;

2. The *neighbor activity analysis* stage selects which interpolation algorithm shall be performed, according to the surrounding features (edge, texture, flat area, etc.);

3. The *interpolation* stage executes a particular demosaicing algorithm, depending on the choice performed by the neighbor activity analysis block; the demosaicing algorithms could use the direction information provided by the direction estimation block

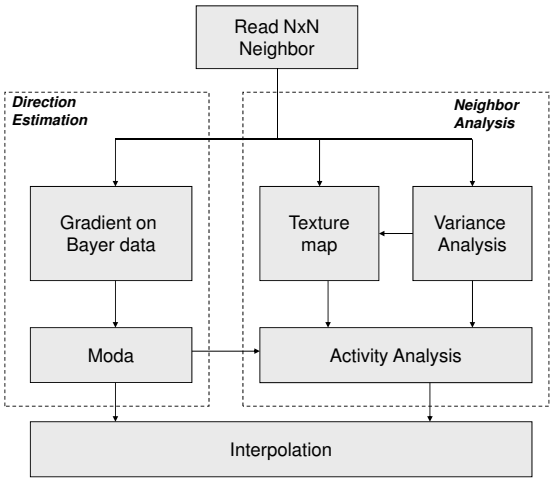


FIG. 2: Scheme of the color interpolation algorithm

for choosing the suitable filter to be applied. Moreover, the available demosaicing algorithms could have different kernel sizes.

Each phase of the algorithm will be extensively described in the following subsections.

A. Direction Estimation Block

The quality of demosaiced images is clearly dependent on the extracted gradient information from the input CFA image, but the information on mosaic image is not so accurate, as each pixel in the mosaic image only has one color channel, as stated in [20]. The direction estimation block is based on the one proposed in the patent [21], where the derivatives along both horizontal and vertical directions are computed through the classical 3x3 $Sobel_x$ and $Sobel_y$ filters, which are shown in equation (1).

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{1}$$

Although these filters are widely used on full color images, their employment on Bayer images is not a commonly defined procedure. We introduced this technique after fixing the following assumptions. Let Q be the generic 3x3 neighborhood from the Bayer pattern as it is defined in (2):

$$Q = \begin{bmatrix} G_{11} & J_{12} & G_{13} \\ H_{21} & G_{22} & H_{23} \\ G_{31} & J_{32} & G_{33} \end{bmatrix} \tag{2}$$

where G_i are the green components and H_i and J_i are the generic red and/or blue components. In order to compute the derivatives only on the green channel, we exploit the spectral correlation property. In particular, as already mentioned, Gunturk et al. in [13] demonstrated that high frequency components of the three color planes are highly correlated.

For instance, if it is assumed a red central pixel, the green component can be determined as:

$$G(i,j) = G_{LPF}(i,j) + R_{HPF}(i,j) \tag{3}$$

Where R_{HPF} is the high frequency content of the R channel, and G_{LPF} and R_{LPF} are the low frequency components of the G and R channels, respectively. Since the high frequency content of the red channel can be calculated as:

$$R_{HPF}(i,j) = R(i,j) - R_{LPF}(i,j) \tag{4}$$

The equation (3) can be rewritten as follows:

$$\begin{array}{rcl} G(i,j) & = & R(i,j) + G_{LPF}(i,j) - R_{LPF}(i,j) \\ & = & R(i,j) + \Delta_{GR}(i,j) \end{array} \tag{5}$$

This implies that a Q' neighborhood, which is a matrix containing only G samples could be written as:

$$Q' = \begin{bmatrix} G_{11} & J_{12} + \Delta_{12} & G_{13} \\ H_{21} + \Delta_{21} & G_{22} & H_{23} + \Delta_{23} \\ G_{31} & J_{32} + \Delta_{32} & G_{33} \end{bmatrix} \tag{6}$$

Therefore the mathematical convolution between Q' and, for example, the derivative filter $Sobel_x$, becomes (7) :

$$\frac{\partial Q'}{\partial x} = Q' \otimes Sobel_x = \begin{bmatrix} G_{11} & J_{12} + \Delta_{12} & G_{13} \\ H_{21} + \Delta_{21} & G_{22} & H_{23} + \Delta_{23} \\ G_{31} & J_{32} + \Delta_{32} & G_{33} \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} =$$

$$= G_{13} + 2(H_{23} + \Delta_{23}) + G_{33} - G_{11} - 2(H_{21} + \Delta_{21}) - G_{31} \tag{7}$$

where Δ_{23} and Δ_{21} are unknown values. Since for real world images the color difference planes (Δ_{GR}=G-R and Δ_{GB}=G-B) are rather flat over small regions, the difference between Δ_{23} and Δ_{21} could be assumed irrelevant, and so the equation (7) becomes:

$$\frac{\partial Q'}{\partial x} = \frac{\partial Q}{\partial x} = G_{13} + 2H_{23} + G_{33} - G_{11} - 2H_{21} - G_{31} \tag{8}$$

A similar condition obviously holds for the computation of $Sobel_y$. This means that Sobel filters can be applied directly on Bayer images.

Unfortunately, if the color planes are not correlated, e.g. red to blue transitions (as shown in Fig.3), the spectral correlation property is not valid [22], and the just disclosed procedure for computing horizontal and vertical gradients could fail. To solve this issue, we developed a method to establish if the color channels are not correlated, directly acting on the Bayer pattern. Let M be the generic 3x3 window, which is shown in (9):

$$M = \begin{bmatrix} M_{NW} & M_N & M_{NE} \\ M_W & M_C & M_E \\ M_{SW} & M_S & M_{SE} \end{bmatrix} \tag{9}$$

For each direction (horizontal and vertical) we define two different gradients, external (ext) and central (cnt), each of which involving a single bayer channel, as it is apparent from equation (10):

$$\begin{array}{ll} \frac{\partial M}{\partial x}ext = M_{NE} + M_{SE} - M_{NW} - M_{SW}; & \frac{\partial M}{\partial x}cnt = M_E - M_W \\ \frac{\partial M}{\partial y}ext = M_{NE} + M_{NW} - M_{SE} - M_{SW}; & \frac{\partial M}{\partial y}cnt = M_N - M_S \end{array} \tag{10}$$

The lack of correlation between color channels is evaluated in two different ways according to the central pixel channel (G or not G). If the central pixel is green, the equation (11) is applied :

$$NoCorrelation\,(M) = \begin{cases} 1 \;\; if \;\; \left(\begin{array}{c} SIGN\left(\frac{\partial M}{\partial x}ext\right) \neq SIGN\left(\frac{\partial M}{\partial x}cnt\right) \; OR \\ SIGN\left(\frac{\partial M}{\partial y}ext\right) \neq SIGN\left(\frac{\partial M}{\partial y}cnt\right) \end{array} \right) \\ 0 \qquad\qquad otherwise \end{cases} \tag{11}$$

Otherwise, if the central pixel is red or blue the equation (12) is applied :

$$NoCorrelation\,(M) = \begin{cases} 1 \;\; if \;\; \left(\begin{array}{l} \left(\begin{array}{c} SIGN\left(\frac{\partial M}{\partial x}ext\right) \neq SIGN\left(\frac{\partial M}{\partial x}cnt\right) \; AND \\ SIGN\left(\frac{\partial M}{\partial y}ext\right) \neq SIGN\left(\frac{\partial M}{\partial y}cnt\right) \end{array} \right) \\ OR \\ \left(\begin{array}{c} \left|\frac{\partial M}{\partial x}ext\right| > 2\left|\frac{\partial M}{\partial x}cnt\right| \; AND \\ \left|\frac{\partial M}{\partial y}ext\right| > 2\left|\frac{\partial M}{\partial y}cnt\right| \end{array} \right) \end{array} \right) \\ 0 \qquad\qquad otherwise \end{cases} \tag{12}$$

where

$$SIGN\,(x) = \begin{cases} 1 & if \; x \geq 0 \\ -1 & otherwise \end{cases} \tag{13}$$

The difference between (11) and (12) is due to the different arrangement of color channels between G and non-G central pixel cases. In fact, if the central pixel is green the horizontal central derivative has the information about the blue (or red) channel, whereas the vertical central derivative has the information about the red (or blue) channel, so the evaluation of cross-correlation could use only simple sign criteria. In particular, equation (11) has the aim of evaluating if at least either green-red or green-blue are opposite correlated. On the contrary, if the central pixel is not green, both horizontal and vertical external derivatives have the information of the same color channel, and hence more complex criteria can be used to properly evaluate cross-correlation. More specifically, in equation (12) the first term of the OR expression is used to evaluate if there is an opposite correlation between green and the color channel which has its samples at the corners of the 3x3 mask; the second term is used to evaluate the lack of correlation between green and the other color channel even if both the external and the central gradients have the same sign. This could happen if the green channel is quite flat with respect to the other color channel. In this case the classic Sobel operators could fail in providing a good gradient estimation.

Let $SimpleGrad_x$ and $SimpleGrad_y$ be two simpler gradient filters, defined in equation (14):

$$SimpleGrad_x = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}; \qquad SimpleGrad_y = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix} \tag{14}$$

These gradient operators could be used every time a lack of correlation between the three color channels is suspected, because they involve a single color channel, and hence they are not negatively influenced by the different variations of the R, G and B channels. Thus, the gradient for the central pixel of the Q neighborhood could be calculated according to the following equation :

$$\nabla Q = \left(\frac{\partial Q}{\partial x}, \frac{\partial Q}{\partial y}\right) \begin{cases} (Q \otimes Sobel_x, Q \otimes Sobel_y) & if\ \ NoCorrelation\,(Q) = 0 \\ (Q \otimes SimpleGrad_x, Q \otimes SimpleGrad_y) & if\ \ NoCorrelation\,(Q) = 1 \end{cases} \tag{15}$$

It is worthwhile to underline that the equations (11) and (12) have especially the aim to identify the cases where the bayer channels are not correlated, to avoid the application of Sobel operators, which involve different color channels. On the contrary, even if the simpler gradient operators were applied on a region where the bayer channels are correlated this would not be a problem, because they provide a quite good gradient estimation, even if Sobel operators are better. Moreover, the final gradient estimation for the central pixel depends on the gradients within its 3x3 neighborhood, as it will be further explained. Once the gradient is computed, its direction and magnitude are calculated by the (16) and (17), respectively.

$$\psi\,(\nabla Q) = \begin{cases} arctg\left(\frac{\partial Q/\partial y}{\partial Q/\partial x}\right) & if\ \partial Q/\partial x > 0 \\ \frac{\pi}{2} & otherwise \end{cases} \tag{16}$$

$$|\nabla Q| = \sqrt{\left(\frac{\partial Q}{\partial x}\right)^2 + \left(\frac{\partial Q}{\partial y}\right)^2} \tag{17}$$

It is important to specify that $\psi\,(\nabla Q)$ and $|\nabla Q|$ represent the direction and magnitude of the gradient of the central pixel of Q. If we set *(x,y)* as the coordinates of the central pixel of Q, the direction and magnitude associated to its gradient could be indicated as:

$$\begin{aligned} or\,(x,y) &= \psi\,(\nabla Q) \\ mag\,(x,y) &= |\nabla Q| \end{aligned} \tag{18}$$

The gradient orientation *or(x,y)* could be quantized into k predefined directions:

$$direction_i = \frac{i \cdot \pi}{k} \qquad i \in [0, k-1]\,, \quad k \in \aleph \tag{19}$$

and, according to the equation (19), the gradient orientation $\overline{or}\,(x,y)$ is set as follows:

$$\overline{or}(x,y) = \{direction_i | direction_i \leq or(x,y) < direction_{i+1}, i \in [0, k-1]\} \tag{20}$$

To avoid the influence of noise on the gradient estimation, a "weighted-mode" (WM) operator is applied on each pixel (x,y) computing the prominent direction in its 3x3 neighborhood. For estimating this direction, the magnitudes of the pixels in the neighborhood are firstly accumulated according to their associated directions:

$$Acc\,(x,y,i) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} mag(x+u, y+v) \cdot t(x+u, y+v, i) \tag{21}$$

where

$$t\,(x,y,i) = \begin{cases} 1 & if\ \overline{or}\,(x,y) = direction_i \\ 0 & otherwise \end{cases} \tag{22}$$

for $i \in [0,\ k\text{-}1],\ k \in \aleph$

Therefore the WM operator selects the direction associated to the maximum sum of magnitudes in the neighborhood of the considered pixel:

$$WM(x,y) = j \ \ such\ \ that\ Acc\,(x,y,j) = \max_{i=0..k-1} \left(Acc\,(x,y,i)\right),\ j \in [0, k-1] \tag{23}$$

Finally, after the gradient direction is retrieved, through the process already described, the edge direction is obtained by taking the orthogonal direction to the gradient one. The edge direction is provided to the neighbor analysis block for the texture analysis, and to the interpolation block when the directional color interpolation is selected.

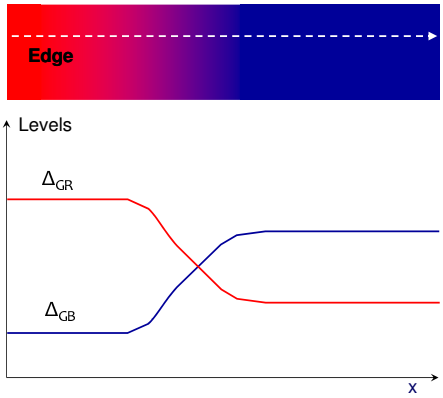


FIG. 3: Uncorrelated planes: example of red to blue transitions.

B. Neighbor Analysis

The aim of this part of the proposed method (whose detailed functional scheme is shown in Fig. 4) is to evaluate the presence of texture, edges or flat zones in the pixel neighbor. This analysis is performed by estimating the variances very close to the pixel and on a larger surrounding. The comparison of these variances provides useful information about neighbor activity. In case of textured zones, an additional analysis is necessary, in order to validate the texture direction estimated by the *Direction Estimation Block*. According to the results of this analysis, the appropriate interpolation scheme is then used. For our purposes we used two kernel sizes: 5x5 for the larger surrounding analysis and 3x3 for the analysis close to the pixel. The variance is computed, as a measure of the activity within a neighborhood. In particular, let P and Q be the generic 5x5 and 3x3 windows from the Bayer pattern, respectively.

$$P = \begin{bmatrix} G_{11} & H_{12} & G_{13} & H_{14} & G_{15} \\ J_{21} & G_{22} & J_{23} & G_{24} & J_{25} \\ G_{31} & H_{32} & G_{33} & H_{34} & G_{35} \\ J_{41} & G_{42} & J_{43} & G_{44} & J_{45} \\ G_{51} & H_{52} & G_{53} & H_{54} & G_{55} \end{bmatrix} \qquad Q = \begin{bmatrix} G_{22} & J_{23} & G_{24} \\ H_{32} & G_{33} & H_{34} \\ G_{42} & J_{43} & G_{44} \end{bmatrix} \tag{24}$$

where G_i are the green components and H_i and J_i are the generic red and/or blue components. The mean value and the variance, for a generic neighborhood S of the central pixel, are computed according to the following equations:

$$\begin{aligned} \mu_X(S) &= \frac{\sum_{(i,j)\in S}\Big(S(i,j) * Mask_X(i,j)\Big)}{\sum_{(i,j)\in S} Mask_X(i,j)} \\ \sigma_X^2(S) &= \frac{\sum_{(i,j)\in S}\Big(S(i,j) * Mask_X(i,j) - \mu_X(S)\Big)^2}{\sum_{(i,j)\in S} Mask_X(i,j)} \\ \sigma^2(S) &= (\sigma_G^2(S) + \sigma_H^2(S) + \sigma_J^2(S)) \\ S &\in \{P, Q\} \quad X \in \{G, H, J\} \end{aligned} \tag{25}$$

where $Mask_X$ identifies the involved pixels components present in the pattern of the processed color component. Let Λ_G, Λ_H and Λ_J be the set of pixel locations, *(x,y)*, that have the samples of green, red (blue) and blue (red) channels, respectively. The binary mask *MaskX* is defined as:

$$\mathrm{Mask_X}(x,y) = \begin{cases} 1 & if \quad (\mathrm{x,y}) \in \Lambda_\mathrm{X} \\ 0 & otherwise \end{cases} \tag{26}$$

where $X = G, H \; or \; J$.

By using the equations in (25) on the neighborhoods P and Q, defined by the equation (24), two values are calculated, $\sigma^2(P)$ and $\sigma^2(Q)$, which represent a measure of the activity in the relative neighborhoods.

Two thresholds are set for distinguishing different conditions. With reference to Fig. 4 , if both the variance values are under the lower threshold, the region is considered flat, and hence a low pass filter based interpolation could be used, in order to remove residual noise.

When the variances have values not too high or low, the evaluation is not straightforward, because it could be due either to a zone nor flat nor textured, or to a zone with a texture without strong edges. In the first case a directional interpolation could introduce artifacts, while in the second one the directional interpolation could highlight the texture.

The mean values and the standard deviations of each color channel, within the P neighborhood, are used to build a texture Mask (we call it TLM, three level mask), which is a matrix having the same size as P and having each element calculated according to the equation (27).

It is straightforward to note that each value within the current P window is associated, in the TLM, with one of three possible levels: Low (value 0), Medium (value 1) or High (value 2), according to its position with respect to its mean value and its standard deviation. The generation of the TLM is illustrated in Fig. 5. In the top left of the figure, a 5x5 crop of a Bayer image is represented, then its numeric representation is shown. The mean values and the standard deviations of each color channel are reported in a box.

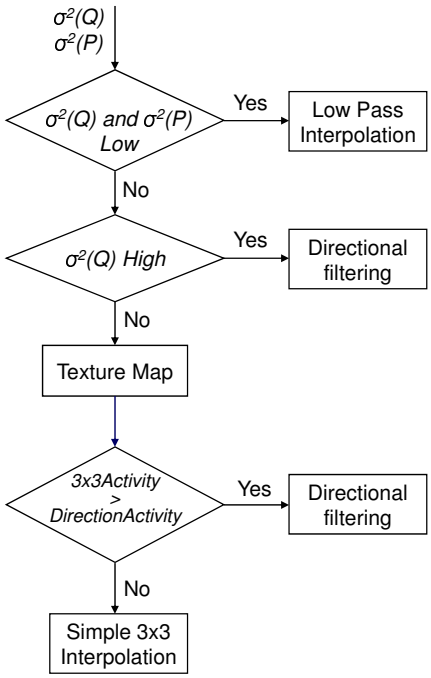


FIG. 4: Neighbor analysis block scheme

$$TLM\left(i,j\right)=\begin{cases}0 & if\ P\left(i,j\right)<\left(\mu_X\left(P\right)-\sigma_X\left(P\right)\right) & with\ \left(i,j\right)\in\Lambda_X\\ 1\ if\ \left(\mu_X\left(P\right)-\sigma_X\left(P\right)\right)<P\left(i,j\right)<\left(\mu_X\left(P\right)+\sigma_X\left(P\right)\right) & with\ \left(i,j\right)\in\Lambda_X & X=\{G,R,B\}\\ 2 & if\ P\left(i,j\right)>\left(\mu_X\left(P\right)+\sigma_X\left(P\right)\right) & with\ \left(i,j\right)\in\Lambda_X\end{cases} \tag{27}$$

$$\text{where } \sigma_X\left(P\right)=\sqrt{\sigma_X^2\left(P\right)}.$$

Let us define *"Activity"* the sum of absolute differences of contiguous elements of the TLM within a certain working mask.

Once the TLM is generated, two working masks are taken into account and hence two activity values are calculated accordingly. The first working mask is the same for each pixel: it is the squared 3x3 mask centered in the pixel under consideration. The second one depends on the direction provided by the direction estimation block for the current pixel: it includes the pixels along the identified direction. It is obvious that a greater activity within a certain working mask represents a lower uniformity of the image content in it. Fig. 6 shows an example of activity computation. Fig. 6(a) illustrates the TLM already presented in fig. 5; in fig. 6(b) the squared working mask is highlighted in pink. Finally, in this example the edge direction is estimated to be 135°, and the correspondent directional working mask is highlighted in Fig. 6(c).

Let T_i represent the generic element of the TLM, the activities within the 3x3 squared mask and within the 5x5 directional mask are calculated through the equations (28) and (29).

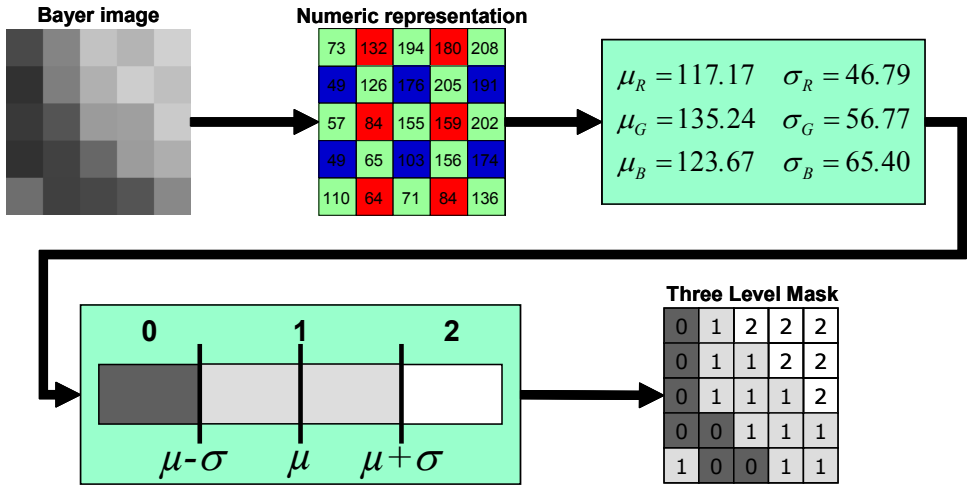
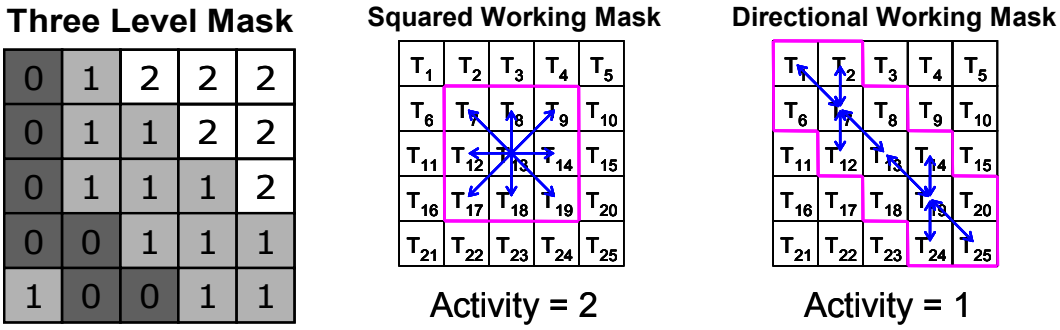


FIG. 5: Example of the Three Level Mask (TLM) generation



(a)First (b)Second (c)Third

FIG. 6: Example of the Activity evaluation assuming that the estimated direction is 135°.

$$\begin{aligned} 3 \times 3Activity &= |T_7 - T_{13}| + |T_8 - T_{13}| + |T_9 - T_{13}| + |T_{14} - T_{13}| \\ &+ |T_{19} - T_{13}| + |T_{18} - T_{13}| + |T_{17} - T_{13}| + |T_{12} - T_{13}| \end{aligned} \tag{28}$$

$$\begin{aligned} Direction\ Activity(135°) &= |T_1 - T_7| + |T_2 - T_7| + |T_7 - T_{12}| + |T_7 - T_{13}| \\ &+ |T_{13} - T_{19}| + |T_{14} - T_{19}| + |T_{19} - T_{24}| + |T_{19} - T_{25}| \end{aligned} \tag{29}$$

It is straightforward that the 5x5 directional working mask and its related activity computation formula depend on the edge direction, which is provided by the direction estimation block. Finally the activity values computed in equations (28) and (29) are compared to evaluate if the activity within the 3x3 squared mask is less than the activity along the direction provided by the direction estimation block. If this is the case, a simple 3x3 interpolation could be used. This control allows avoiding the directional 5x5 interpolation in case of a wrong estimated direction, or in case of a region close to an edge, thus reducing the introduction of unpleasant artifacts. In the example of fig. 5, the directional interpolation is chosen for the central pixel, because the 3x3 squared mask activity (Fig. 6(b)) is greater than the 5x5 directional one (Fig. 6(c)).

For the sake of completeness, the activity computation formulas applied in case of edge direction equal to 0° and 22.5° are shown in equations (30) and (31), respectively. All the other activity computation formulas can be easily derived from the three examples already presented.

$$\begin{aligned} \textit{Direction Activity}(0°) &= |T_6 - T_8| + |T_8 - T_{10}| + |T_{11} - T_{12}| + |T_{12} - T_{13}| \\ &+ |T_{13} - T_{14}| + |T_{14} - T_{15}| + |T_{16} - T_{18}| + |T_{18} - T_{20}| \end{aligned} \tag{30}$$

$$\begin{aligned} \textit{Direction Activity}(22.5°) &= |T_9 - T_{10}| + |T_{11} - T_{12}| + |T_{12} - T_{13}| + |T_{13} - T_{14}| \\ &+ |T_{14} - T_{15}| + |T_{16} - T_{17}| + |T_{13} - T_{10}| + |T_{13} - T_{16}| \end{aligned} \tag{31}$$

C. Interpolation

As already pointed out, according to the analysis of the "Direction Estimation" and "Neighbor Analysis", the appropriate color interpolation scheme is used. in particular we use:

- 5x5 directional filter;
- 3x3 simple interpolation;
- 5x5 omnidirectional low pass filter.

In the following paragraphs each one of the aforementioned algorithms will be described.

1. Directional filtering color interpolation

If a strong edge or a texture is detected, the directional filtering is used. This interpolation is carried out through elliptical shaped Gaussian filters, given by:

$$f(u, v, \alpha) = he^{-\frac{\tilde{u}^2}{2\sigma_u^2} - \frac{\tilde{v}^2}{2\sigma_v^2}} \tag{32}$$

where

$$\begin{aligned} \tilde{u} &= u\cos(\alpha) - v\sin(\alpha), \\ \tilde{v} &= u\sin(\alpha) + v\sin(\alpha), \end{aligned} \tag{33}$$

and σ_u^2, σ_v^2 are the variances along the two dimensions, h is a normalization constant and α is the orientation angle. Through heuristic experiments $\sigma_u = 8$ and $\sigma_v = 0.38$ have been fixed.

These interpolation kernels can be computed only once, after the number of admissible directions, k, has been set. More specifically, fixing the coordinates *(u,v)* in a given range (i.e., 5x5), the kernel filters DF_i can be generated, for $i \in [0, k-1]$. The generic element of these matrices, $DF_i(u,v)$, is defined as follows:

$$DF_i\,(u, v) = f\,(u, v, i \cdot \pi/k) \tag{34}$$

During the directional filtering color interpolation, once the direction j for the central pixel is derived by the direction estimation block, the kernel DF_j is applied to the central

pixel to obtain the low pass filter (LPF) color components R_{LPF_DF}, G_{LPF_DF} and B_{LPF_DF}, preserving the image from zigzag effect and partially from false colors.

Let P be the generic 5x5 neighborhood from the Bayer pattern as it is defined in equation (24).

The LPF components, for the central pixel (x,y), are computed by assuming the following rules:

$$\begin{aligned} H_{LPF\_DF} &= P \otimes Mask_H \otimes DF_j \\ G_{LPF\_DF} &= P \otimes Mask_G \otimes DF_j \\ J_{LPF\_DF} &= P \otimes Mask_J \otimes DF_j \end{aligned} \tag{35}$$

where $Mask_X$, which has been already defined in equation (26), identifies the pixels belonging to the processed color component.

As already aforementioned, the high frequency components of the three color channels are highly correlated, and hence any color component can be exploited to reconstruct the high frequencies of the remaining color components. For this reason, the directional LPF component of the central pixel is likewise computed in order to calculate its high pass filter (HPF) component, which will be added to the other two color channels. Assuming a green central pixel, its high pass component can be calculated according to the equation (36).

$$G_{HPF} = G - G_{LPF\_DF} \tag{36}$$

The resulting G_{HPF} value is added to the unknown values, in this case R and B (e.g. H), to increase their high frequency content:

$$H_{DF} = H_{LPF\_DF} + G_{HPF} \tag{37}$$

The central pixel value will be maintained to its original value:

$$G_{DF} = G \tag{38}$$

2. Simple 3x3 interpolation

The 5x5 directional filtering color interpolation could produce some artifacts near edges, especially in text images. This problem could arise when a wrong direction is estimated or when a large interpolation window is used for interpolating a small region, enclosed by edges. To overcome this problem a 3x3 interpolation could be performed in regions which are close to edges.

A wide variety of 3x3 interpolation algorithms exists: bilinear interpolation, edge sensing, etc. In order to achieve good quality results near corners and edges, it is possible to use an algorithm which analyzes the 5x5 neighborhood of the central pixel, and then applies a 3x3 interpolation, avoiding the interpolation across edges.

Taking idea from the paper in [7], the proposed 3x3 interpolation algorithm uses a threshold-based variable number of gradients. According to the central pixel channel, four (G case) or eight (R or B case) gradients are computed in a 5x5 neighborhood. Each gradient is defined as the sum of absolute differences of the like-colored pixels in this neighborhood. For each color channel to be interpolated, a proper subset of these gradients is selected

to determine a threshold of acceptable gradients. Each missing color channel is then obtained by averaging the color values of the same channel in the 3x3 neighborhood, which are associated with gradients lower than the calculated threshold.

3. Omnidirectional low pass filter

This interpolation is performed on flat regions. A simple 5x5 omnidirectional low pass filter is applied for smoothing the image, removing some residual noise from it. In this case the original sampled values are also modified. The kernels used to achieve the interpolation are shown in equation (39). In particular, the A kernel is applied in case of G central pixel, whereas the B kernel is used in case of R/B central pixel.

$$A = \frac{\begin{bmatrix} 0 & 8 & 4 & 8 & 0 \\ 8 & 8 & 16 & 8 & 8 \\ 4 & 16 & 16 & 16 & 4 \\ 8 & 8 & 16 & 8 & 8 \\ 0 & 8 & 4 & 8 & 0 \end{bmatrix}}{64} \qquad B = \frac{\begin{bmatrix} 0 & 3 & 9 & 3 & 0 \\ 3 & 16 & 10 & 16 & 3 \\ 9 & 10 & 28 & 10 & 9 \\ 3 & 16 & 10 & 16 & 3 \\ 0 & 3 & 9 & 3 & 0 \end{bmatrix}}{64} \tag{39}$$

III. FALSE COLORS REMOVAL

Since the color interpolation step quite well exploits spatial correlation, zipper effect does not often arises. On the contrary, residual false colors can be introduced by the directional filters. For this reason, a postprocessing technique has been developed, which eliminates the residual false colors, thus considerably improving the final image quality.

Many methods have been developed in the past to reduce false colors. Among them, the most interesting techniques median filter the inter-channel differences, thus removing spikes and valley from them, which usually correspond to false colors. Freeman's approach [17] and Lu and Tan's technique [16] have been already presented in the introduction section. The post-processing approach described in [23] not only is based on the color difference model, but also uses an edge-sensing mechanism. Another interesting technique, which is proposed in [18], updates the R, G, B values adaptively, modifying also the original pixel value which could be corrupted, due to the effect of noise. Two updated values for the green channel are calculated using each color difference domain:

$$\begin{aligned} G^R(i,j) &= R(i,j) + v_{GR}(i,j) \\ G^B(i,j) &= B(i,j) + v_{GB}(i,j) \end{aligned} \tag{40}$$

where

$$\begin{aligned} v_{GR}(i,j) &= median\{G(k,l) - R(k,l) \,|(k,l) \in A\} \\ v_{GB}(i,j) &= median\{G(k,l) - B(k,l) \,|(k,l) \in A\} \end{aligned} \tag{41}$$

And A denotes the support of the 5x5 local window centered in (i,j).

The updated G value is determined by the weighted sum of two updated G^R and G^B values of each color difference domain and original G value. Subsequently, R and B values are updated using the updated G value. This process is expressed as:

$$\begin{array}{l} G'(i,j) = \frac{1}{2}G(i,j) + \frac{1}{2}\left\{(1-a(i,j))\,G^R(i,j) + a(i,j)\,G^B(i,j)\right\} \\ R'(i,j) = \frac{1}{2}R(i,j) + \frac{1}{2}\left\{G'(i,j) - v_{GR}(i,j)\right\} \\ B'(i,j) = \frac{1}{2}B(i,j) + \frac{1}{2}\left\{G'(i,j) - v_{GB}(i,j)\right\} \end{array} \tag{42}$$

where $a(i,j)$ is a weight, expressed as:

$$a(i,j) = \frac{\sigma^2_{(G-R)}(i,j)}{\sigma^2_{(G-R)}(i,j) + \sigma^2_{(G-B)}(i,j)}, 0 < a(i,j) < 1 \tag{43}$$

$\sigma^2_{(G-R)}$ and $\sigma^2_{(G-B)}$ represent the variances of interchannel differences.

As it is apparent from the equation (42), the color correction algorithm proposed in [18], thanks to the variance information, weights more the flatter color difference domain than the other. Moreover, the initial interpolated value is not totally exchanged by the updated value, but it is equally weighted for correction. Subsequently, the color values of the central pixel are replaced by R', G' and B' so that they will be involved in filtering the updating pixels.

The local statistics are effectively estimated from a running square window, through the formulas in (44):

$$\begin{array}{l} E[A(i,j)] = \frac{\sum_{k,l\in A} e(k,l)\cdot A(k,l)}{\sum_{k,l\in A} e(k,l)} \\ \sigma^2_A(i,j) = \frac{\sum_{k,l\in A} e(k,l)\cdot (A(k,l) - E[A(i,j)])^2}{\sum_{k,l\in A} e(k,l)} \\ e(k,l) = 1 - (A(i,j) - A(k,l)) \end{array} \tag{44}$$

The just described technique has the disadvantage of weighting the unfiltered values together with the filtered ones, so false colors are reduced, without being completely removed. If the unfiltered values were not weighted in the color correction process, false colors would be removed much better but true colors could be removed as well. In fact, it is well known [24] that a window width $2k+1$ median filter can only preserve details lasting more than $k+1$ points. To preserve smaller details in signals, a smaller window width median filter must be used. Unfortunately, the smaller the filter window is, the poorer its false color reduction capability becomes.

To overcome this problem, we have developed a new solution which takes into account both a 3x3 window median filter and a 5x5 window median filter. In particular, local variances of interchannel differences are evaluated through the equation (44) in both 3x3 and 5x5 square windows, thus obtaining $\sigma^2_{(G-R)3\times3}, \sigma^2_{(G-B)3\times3,}\sigma^2_{(G-R)5\times5}, \sigma^2_{(G-B)5\times5}$. Similarly, four median values $(v_{GR3x3}, v_{GB3x3}, v_{GR5x5}, v_{GB5x5})$ are computed.

Due to the Bayer arrangement and to a quite good interpolation achieved by the previous demosaicing step, the green plane is less affected by false colors than the R and B planes, and hence it is left unmodified. The color correction on R and B planes is performed according to the following rules:

$$\begin{array}{l} R'(i,j) = G(i,j) - [(1 - kr_{ratio}(i,j))\cdot v_{GR3\times3}(i,j) + kr_{ratio}(i,j)\cdot v_{GR5\times5}(i,j)] \\ B'(i,j) = G(i,j) - [(1 - kb_{ratio}(i,j))\cdot v_{GB3\times3}(i,j) + kb_{ratio}(i,j)\cdot v_{GB5\times5}(i,j)] \end{array} \tag{45}$$

Where

$$\begin{array}{l} kr_{ratio}(i,j) = \frac{\sigma^2_{(G-R)3\times3}(i,j)}{\sigma^2_{(G-R)3\times3}(i,j) + \sigma^2_{(G-R)5\times5}(i,j)} \\ kb_{ratio}(i,j) = \frac{\sigma^2_{(G-B)3\times3}(i,j)}{\sigma^2_{(G-B)3\times3}(i,j) + \sigma^2_{(G-B)5\times5}(i,j)} \end{array} \tag{46}$$

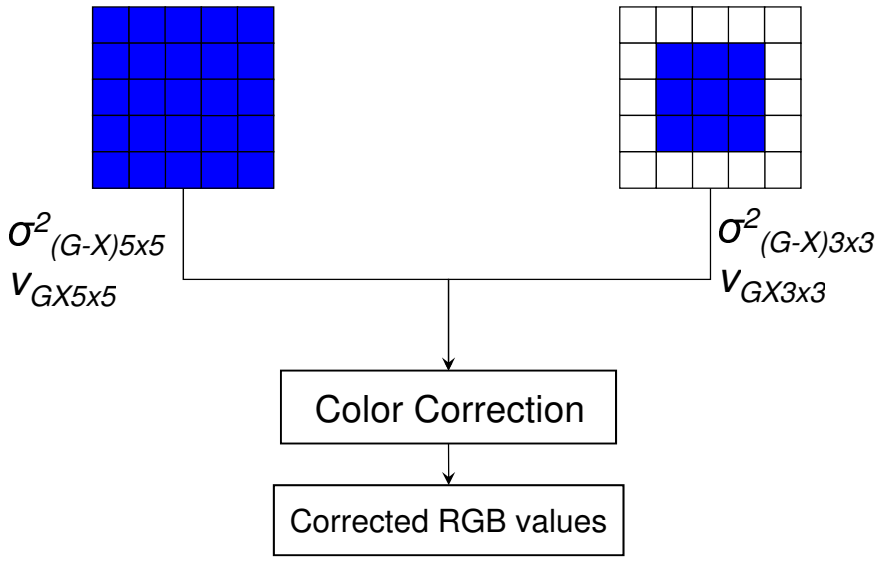


FIG. 7: Schematic representation of the proposed approach

By this way, the greatest weight is assigned to the median value associated to the neighborhood having the lowest variance. So, if the pixel being processed belongs to an edge, the 3x3 variances of interchannel differences are likely to be greater than the correspondent 5x5 variances, and hence the filtering action will be stronger. Vice versa, if the 3x3 neighborhood is flatter than the 5x5 neighborhood, the 3x3 median is weighted more than the 5x5 median, and hence details lasting 2 pixels at least are preserved.

Fig.7 shows a schematic representation of the proposed approach. Since variances could be used as measures of the flatness of a region, it is possible to perform the color correction process only where it is needed, thus reducing the power consumption. In the color difference domain, a flat color difference neighborhood is characterized by: expectation values close to the central pixel values, low variance values. Defined $Diff_{GR}(i,j) = G(i,j) - R(i,j)$ and $Diff_{GB}(i,j) = G(i,j) - B(i,j)$, these two conditions can be expressed with the following formulas:

$$\begin{aligned} |E\left[Diff_{GR}(i,j)\right] - Diff_{GR}(i,j)| &< Mean_{Threshold} \\ |E\left[Diff_{GB}(i,j)\right] - Diff_{GB}(i,j)| &< Mean_{Threshold} \\ \sigma^2_{(G-R)3\times3}(i,j) &< Variance_{Threshold} \\ \sigma^2_{(G-B)3\times3}(i,j) &< Variance_{Threshold} \end{aligned} \tag{47}$$

where *MeanThreshold* and *VarianceThreshold* are two thresholds, which can be set to 16 for 8 bits images.

If the conditions stated above are satisfied, the region is considered flat, thus the central pixel can be left unchanged.

According to the expectation value and to the variance, a map of homogeneous (black) vs. inhomogeneous (white) regions can be achieved, as it is apparent from Fig.8.

(a)Processed image

(b)Map of homogeneous vs. inhomogeneous regions

FIG. 8: Map of homogeneous vs. inhomogeneous regions

Homogeneous regions (black) can be left unchanged or can be low pass filtered. Inhomogeneous regions (white) are processed by the color correction algorithm.

IV. EXPERIMENTAL RESULTS

In this section, we will show the experimental results obtained using the proposed algorithm compared with other demosaicing techniques. The results are highligthed both visually and numerically. For coherence with the most of papers on demosaicing, which use the Kodak image test set to make comparisons with other techniques, we downloaded this database from [25] and then we processed these images through the proposed approach. These images contains a lot of edges and textures, thus they are useful for highlight how the various algorithms handle the high frequency content. Since the Kodak image test set is full color, CFA images are simulated by subsampling the color channels according to the Bayer Pattern. This means that the algorithms are not applied on genuine Bayer data (derived from a real sensor), but on images that have already had matrix and gamma applied, so their gamut is far wider than that of data coming from a sensor. Moreover they are immune from noise and other impairments related to the sensor technology. For these reasons we also applied the proposed algorithm to Bayer data from real sensors.

In our experiments, we have used the PSNR (Peak signal to Noise Ratio) for evaluating the quality of the images (only for Kodak database). Given two $N \times M$ images A and B, the PSNR is expressed as:

$$PSNR = 20 \times \log_{10} \left(\frac{255}{\sqrt{\frac{1}{N \times M} \sum_{i=1}^{N \times M} (A_i - B_i)^2}} \right) \tag{48}$$

Higher values (expressed in decibel) of the PSNR generally imply better quality.

For demonstrating the effectiveness of the proposed algorithm, we will compare it with other five demosaicing algorithms: bilinear interpolation (the simplest), edge sensing interpolation, Gunturk's method, and the directional interpolations proposed by Hirakawa et al. in [11] and by Menon et al. in [10], respectively. The matlab codes of both bilinear and

edge sensing interpolation were obtained from [26], while the source codes of Gunturk's, Hirakawa's and Menon's approaches were downloaded from [13], [19] and [10], respectively. The PSNRs have been computed separately for each color channel. The Table I shows the collected PSNR measures for each image of the test set. The proposed method, Gunturk's and Menon's approaches give the highest PSNR values. However, as it is well known, the PSNR is not always representative of the visual quality of an image. Two images having the same PSNR could appear very different from the visual standpoint. Moreover, the PSNR approach is a full-reference metric which cannot be applied on real sensor data. For these reasons, a subjective image quality analysis has been also achieved.

Figures 10 and 11 illustrate the ROIs (region of interest) relative to the "hats" test image (kodim03) and to the "mountain" test image (kodim13), respectively. They were obtained cropping a detail from the originals and interpolated images. In these figures, (a) represents the ROI of the original image, (b) is the result of the bilinear interpolation, (c) is the result of the edge sensing algorithm, (d) is obtained applying the interpolation by Gunturk, (e) is the result of the Hirakawa's approach, (f) is the output of the technique proposed by Menon *et al.* and, finally, (g) is obtained using the proposed algorithm. The ROIs show how our algorithm drastically reduces both the zipper and false color defects. Fig. 10.b presents a great amount of false colors and blur. Edge sensing algorithm quite well interpolates along edges but introduces many false colors (see fig. 10.c). Figs. 10.d and 10.f show that both Gunturk's and Menon's approach still maintain few zipper artifacts near edges, although for this image the PSNR of the Menon's technique is comparable to ours and higher than the Hirakawa's approach, which outputs an image free from zipper effects and false colors (see fig. 10.e) like our demosaicing technique. Looking at fig. 11, it is possible to see that the proposed technique (fig. 11.g) outperforms all the others in removing false colors, providing an image where also edges are effectively interpolated and sharpness is maintained. Both bilinear (fig. 11.b) and edge sensing (11.c) techniques produce images heavily affected by false colors. Gunturk's (fig. 11.d) and Menon's (11.f) approaches also introduce some visible false colors. Finally, Hirakawa's technique still maintains a minimum amount of false colors in comparison with our approach.

As it has been aforementioned, some tests were performed on real sensor images. Some visual results will be shown in comparison with both Hirakawa's and Menon's algorithms, which are both based on directional approaches, like the proposed technique. The image shown in fig. 12.a has a huge high frequency content, especially in the windows, which have details at the Nyquist's frequency of the sensor. A ROI from the output images of the Hirakawa's, Menon's and our techniques are shown in figs. 12.b, 12.c and 12.d, respectively. It is readily apparent, in this example, that the proposed technique outperforms the other two in interpolating edges and in removing false colors.

Another test which is usually performed to compare different demosaicing algorithm is the interpolation of a colored resolution chart, which allows to see how each algorithm resolves color details. For this purpose, the image shown in fig. 13.a was taken into consideration. Looking at fig. 13.d it is quite evident that the proposed technique resolves color details much better than the other two approaches (see figs. 13.b and 13.c). This is due to the proposed direction estimation block which evaluates the correlation among different Bayer colors, to determine the direction to be used in the interpolation phase.

As it has been already discussed, images coming from real sensors are often affected by artifacts due to noise and green imbalance, so it is important to design an interpolation algorithm able to not exalt and even to reduce these kinds of issues, which would be further

(a)Original image

(b)Map of interpolation

FIG. 9: Map of interpolation method

enhanced by the successive sharpening step. As the proposed technique low pass filters flat regions, green imbalance and noise are heavily reduced, thus producing high quality images. Looking at fig. 14.c it is possible to notice that residual noise is quite well reduced by the proposed technique, whereas edges are still well interpolated. Both figs. 14.a and 14.b are still affected by noise. In order to have a numeric measure of the noise immunity of the presented demosaicing technique, we added three different amounts of gaussian noise (having standard deviations equal to 8, 12 an 25, respectively) to a subset of the original Kodak images, and then interpolated them with some state of the art color interpolation approaches and the proposed technique. Afterwards, we calculated the PSNR with respect to the original kodak images. The PSNR values are reported in TableXXX, where the proposed approach is compared with the algorithms proposed in [10], [27] and [9]. From this table it is evident how the proposed demosaicing algorithm has the greates PSNR values for almost all the processed images. Moreover, the improvement is geater especially in images having large flat areas, where the human visual system is more sensitive to noise. The noise immunity of the algorithm depends on the threshold used to identify flat regions. In the aforementioned experiment it was set to 300, even if it should be made dependent on the noise level of the input image. Fig. 9 is a map showing which interpolation is chosen for each pixel of the 'lighthouse' image, where gaussian noise having sigma equal to 12 was added. In particular, black regions are interpolated through the 5x5 omnidirectional low pass filter, red regions are interpolated by the directional approach and green regions are interpolated through the simple 3x3 technique. It is quite evident that strong edges are interpolated by the directional technique, whereas regions near edges are reconstructed by the 3x3 approach. Noise in flat regions is reduced by the omnidirectional low pass filter. A ROI of the 'lightouse' image is shown in fig. 15 where the original kodak image is compared with the noisy image (with gaussian noise, sigma=12)interpolated by the Menon's technique [10], the Li's approach [27], the Wu and Zhang's algorithm [9] and the proposed technique.

Finally, one more visual comparison is presented to validate the effectiveness of the proposed post-processing false colors removal algorithm with respect to other state of the art

approaches. In particular, the images from the Kodak data set were interpolated through the demosaicing step proposed in Section II, then they were processed by the Freeman's technique [17], the Lu's algorithm [16], the post-processing approach proposed by Lukac *et al.* in [23], the color correction technique presented in [18] and the proposed algorithm, already disclosed in Section III. Fig. 16 is a ROI from the "hotel" test image (kodim08). In particular, (a) represents the ROI of the original image, (b) is the result of the color interpolation disclosed in Section II, (c) is obtained processing (b) with the Freeman's approach, (d) is the result of the Lu's technique, (e) is the output of the Lukac's post-processing technique, (f) is the result of [18] and, finally, (g) is obtained processing (b) with the algorithm presented in Section III, so it corresponds to the output of the method proposed in this paper. It is quite evident that the proposed color interpolation technique (fig. 16.b) very well exploits spatial correlation, but lots of false colors are introduced, and hence the application of a post-processing aliasing removal is indispensable. Freeman's technique (fig. 16.c) not only does not remove many false colors but also introduces annoying zipper artifacts. Lu's approach does not introduce zipper effects, but considerably blurs the image. Lukac's algorithm is neither able to remove all the color artifacts. Kim's approach, weighting the initially interpolated values to produce the corrected ones, is not able to effectively reduce the false colors. The proposed technique (fig. 16.g) quite well reduces false colors without introducing zipper effects.

V. CONCLUSION

In this paper, we have presented a color interpolation method based on edge and texture analysis. According to this analysis we proposed the usage of different interpolation approaches, where the novel proposed directional filtering exploits spatial-spectral correlation. A powerful post-processing algorithm effectively removes residual impairments introduced by the color interpolation step. This paper compares demosaicing performance of our improved method with the state-of-the-art demosaicing methods. The cascading structure of the proposed method has a merit of the pipelined real-time processing for the hardware implementation, while some other methods require iterative computations. The proposed method still opens a possibility of further improvement, for example, applying stripe detection to improve the performances in near Nyquist frequencies.

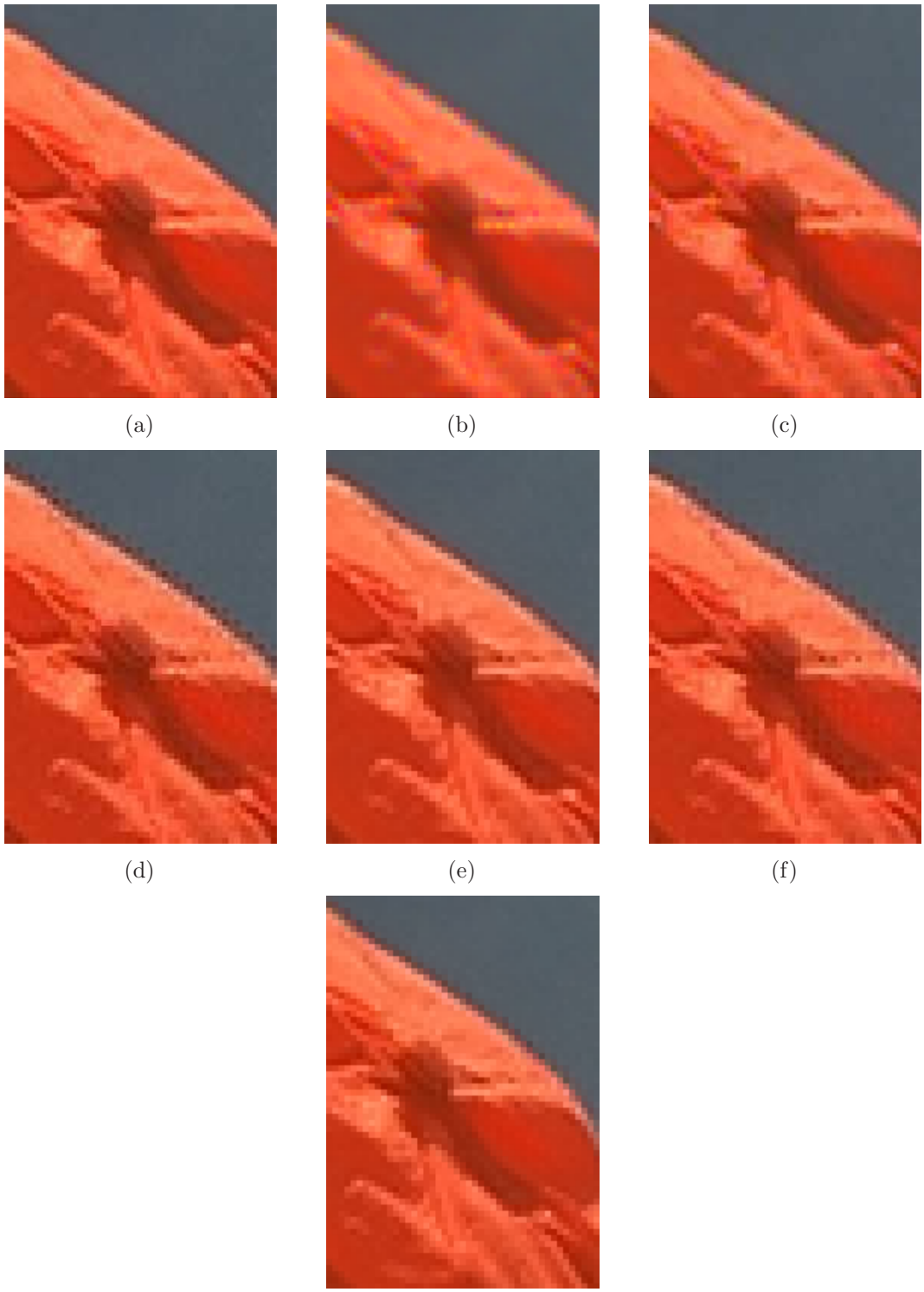
---

[1] B. E. Bayer, *Color imaging array*, US Patent 3,971,065 (1976).

[2] N. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, *Adaptive filtering for color filter array demosaicking*, IEEE Transactions on Image Processing pp. 2515–2525 (2007).

[3] J. E. Adams, *Interactions between color plane interpolation and other image processing functions in electronic photography*, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, edited by C. N. Anagnostopoulos and M. P. Lesser (1995), vol. 2416 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 144–151.

[4] K. Hirakawa, *Cross-talk explained*, in *IEEE Int. Conference on Image processing, (ICIP 2008)* (2008), pp. 677–680.

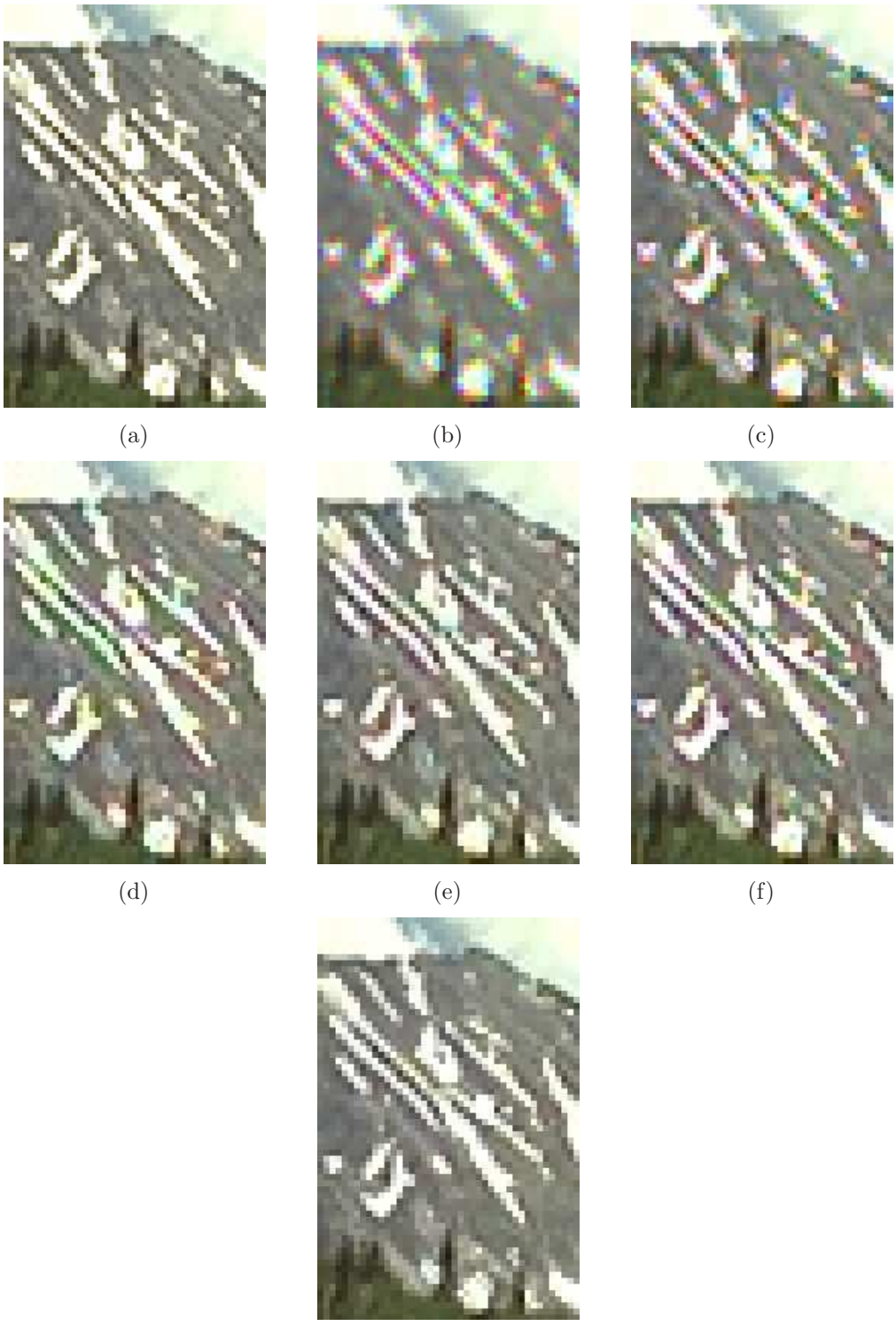


(a) (b) (c)

(d) (e) (f)

(g)

FIG. 10: Example of ROI visual comparison relative to the "hats" test image (kodim03). (a) Original image; (b) Bilinear Interpolation; (c) Edge sensing [26]; (d) Gunturk [13]; (e) Hirakawa [11]; (f) Menon [10]; (g) Proposed solution.



(a) (b) (c)

(d) (e) (f)

(g)

FIG. 11: Example of ROI visual comparison relative to the "mountains" test image (kodim13). (a) Original image; (b) Bilinear Interpolation; (c) Edge sensing [26]; (d) Gunturk [13]; (e) Hirakawa [11]; (f) Menon [10]; (g) Proposed solution.

TABLE I: Prova di tabella

	Bilinear			Edge Sensing			Gunturk			Hirakawa			Menon			Proposed		
Image Name	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
kodim01	25.39	29.45	24.91	30.84	31.47	30.89	34.99	40.12	35.08	34.61	36.26	34.77	36.24	38.30	36.46	35.68	39.78	36.55
kodim02	31.92	35.75	31.34	35.64	37.41	36.84	38.46	40.29	35.25	36.38	41.52	40.12	37.86	43.08	41.10	36.82	42.99	40.71
kodim03	33.36	36.49	32.48	38.09	38.54	37.64	40.80	43.31	36.15	40.58	43.69	40.10	41.67	44.95	40.94	41.63	44.64	41.05
kodim04	32.34	36.07	32.77	35.27	37.57	37.38	37.91	42.25	37.58	35.56	41.70	40.91	36.78	43.37	42.12	37.24	43.42	41.94
kodim05	25.83	29.25	25.66	31.28	31.08	31.05	37.05	39.50	34.10	35.13	37.44	34.42	37.05	39.63	36.30	36.11	39.31	35.80
kodim06	26.60	30.84	26.59	32.51	33.00	32.12	37.69	41.29	34.00	37.58	39.11	36.45	39.29	41.06	37.91	37.16	40.92	36.73
kodim07	32.55	36.04	31.89	38.25	38.40	37.76	41.29	43.34	36.93	40.06	42.66	39.01	41.39	44.09	40.22	40.64	43.77	40.19
kodim08	22.55	27.37	22.45	29.60	30.42	29.48	34.27	38.08	32.59	33.21	35.35	33.24	34.66	37.33	34.76	33.50	37.93	33.67
kodim09	31.33	35.43	31.36	37.47	38.16	37.81	40.98	44.10	38.74	40.31	43.01	40.64	41.25	44.39	42.94	40.87	43.97	40.39
kodim10	31.50	35.04	31.04	36.87	37.37	36.73	39.83	44.36	38.50	39.48	42.87	39.80	40.84	44.47	41.62	40.68	44.13	40.49
kodim11	28.25	32.15	28.25	33.38	33.81	33.47	38.42	41.03	36.83	36.63	39.01	37.46	38.33	40.93	39.02	37.05	41.43	38.38
kodim12	32.01	36.35	31.67	37.93	39.01	37.93	40.55	44.36	38.22	40.54	43.96	40.78	42.00	45.43	42.18	40.84	44.57	41.35
kodim13	23.12	26.51	22.99	27.18	27.28	26.80	33.80	36.86	32.15	31.53	32.40	30.61	33.38	34.49	32.38	33.78	36.58	32.75
kodim14	28.18	31.79	28.06	32.89	33.46	32.91	35.87	37.87	32.55	34.16	38.00	34.93	35.74	39.52	36.08	35.14	39.16	35.51
kodim15	30.80	34.34	29.67	34.52	36.31	35.67	36.80	40.17	34.87	35.49	40.66	38.56	36.77	42.15	39.76	36.57	42.01	40.04
kodim16	30.34	34.30	29.61	36.04	36.80	35.90	40.25	44.47	35.44	41.36	42.86	40.55	42.89	44.59	41.98	40.29	43.95	39.91
kodim17	31.43	34.48	30.67	35.97	35.69	35.47	40.38	43.75	37.80	39.18	40.65	38.61	40.45	42.33	40.35	40.06	42.70	39.69
kodim18	27.27	30.40	26.72	31.23	31.28	30.89	35.71	39.35	34.96	33.72	35.60	33.83	35.14	37.57	35.90	35.21	38.41	35.61
kodim19	26.85	31.75	27.01	34.48	35.04	34.60	38.14	42.62	38.44	37.61	39.50	37.86	38.78	41.23	39.98	37.86	41.69	38.39
kodim20	30.25	33.11	30.45	36.17	36.26	35.10	36.72	43.23	37.14	38.90	40.65	36.91	40.23	42.26	37.84	40.29	43.20	38.51
kodim21	27.68	31.43	27.46	32.74	33.02	32.30	38.01	41.58	35.90	36.62	37.93	35.57	37.79	39.62	36.83	37.63	40.91	36.56
kodim22	29.94	33.13	29.22	34.17	34.71	33.80	37.78	39.89	35.91	35.68	38.45	35.47	37.00	39.80	36.70	36.39	39.43	36.11
kodim23	34.53	37.17	33.83	39.41	39.84	38.23	42.20	43.48	38.64	38.96	43.27	39.57	39.97	45.08	39.92	41.16	44.79	41.76
kodim24	26.39	29.30	25.29	30.24	30.06	28.81	34.37	37.29	32.26	32.62	34.95	31.60	34.34	36.58	32.59	34.29	37.29	32.58
Average	29.18	32.83	28.81	34.26	34.83	34.15	38.01	41.36	35.83	36.91	39.65	37.16	38.33	41.34	38.58	37.79	41.54	38.11

[5] S. Battiato, M. Guarnera, G. Messina, and V. Tomaselli, *Recent patents on color demosaicing*, Recent Patents on Computer Science, Bentham Science Publishers Ltd pp. 194–207 (2008).

[6] L. Chang and Y.-P. Tan, *Effective use of spatial and spectral correlations for color filter array demosaicking*, Consumer Electronics, IEEE Transactions on pp. 355–365 (2004), ISSN 0098-3063.

[7] E. Chang, S. Cheung, and D. Y. Pan, *Color filter array recovery using a threshold-based variable number of gradients*, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, edited by N. Sampat and T. Yeh (1999), vol. 3650 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 36–43.

[8] T. Kuno, H. Sugiura, and N. Matoba, *Digital still cameras*, Consumer Electronics, IEEE Transactions on pp. 259–267 (1999), ISSN 0098-3063.

[9] X. Wu and N. Zhang, *Primary-consistent soft-decision color demosaicking for digital cameras (patent pending)*, IEEE Transactions on Image Processing pp. 1263–1274 (2004).

[10] D. Menon, S. Andriani, and G. Calvagno, *Demosaicing with directional filtering and a posteriori decision*, IEEE Transactions on Image Processing pp. 132–141 (2007), http://www.danielemenon.it/ pub/ dfapd/ dfapd.html.

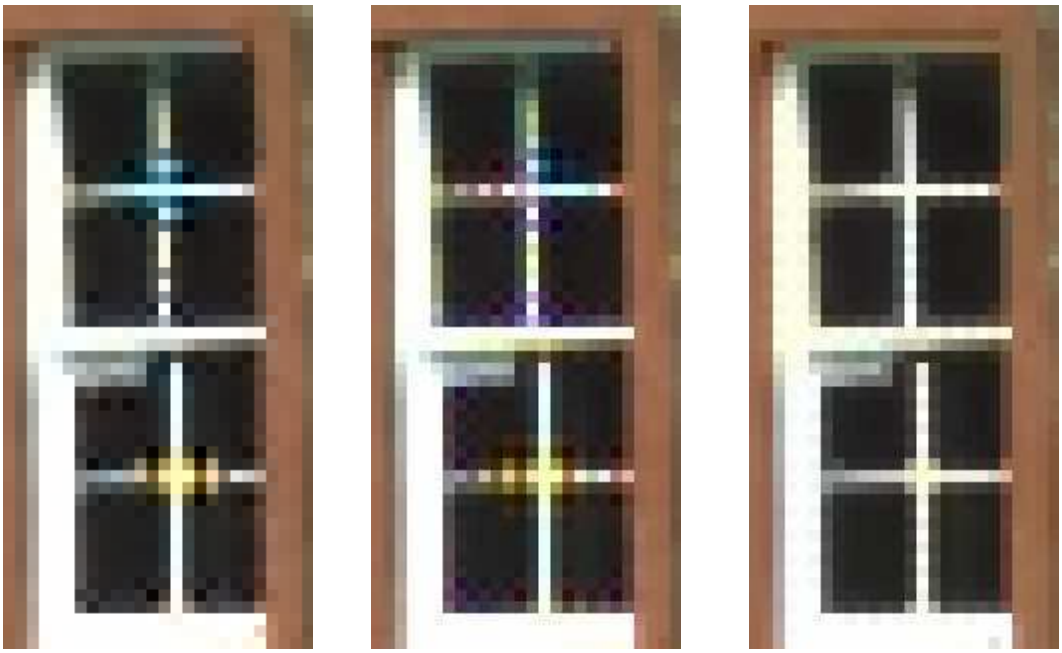
[11] K. Hirakawa and T. W. Parks, *Joint demosaicing and denoising*, IEEE Transactions on Image Processing pp. 2146–2157 (2006).

[12] E. Yun, J. Kim, and S. Kim, *Color interpolation by expanding a gradient method*, IEEE Transactions on Consumer Electronics pp. 1531–1539 (2008).

[13] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, *Color plane interpolation using alternating projections*, IEEE Transactions on Image Processing pp. 997–1013 (2002), http://www.ece.gatech.edu/ research/ labs/ MCCL/ research/ p_demosaick.html.

[14] N. Lian, V. Zagorodnov, and Y.-P. Tan, *Edge-preserving image denoising via optimal color space projection*, IEEE Transactions on Image Processing pp. 2575–2587 (2006).

(a)



(b) (c) (d)

FIG. 12: Example of visual comparison on real sensor image.(a) Real sensor image; (b) ROI by Hirakawa [11]; (c) ROI by Menon [10]; (d) ROI by proposed solution.

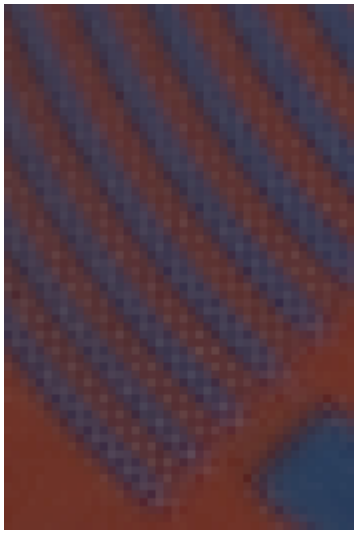
[15] T. Saito and T. Komatsu, *Demosaicing approach based on extended color total variation regularization*, in *IEEE Int. Conference on Image Processing, (ICIP 2008)* (2008), pp. 885–888.

[16] W. Lu and Y.-P. Tan, *Color filter array demosaicking: new method and performance measures*, IEEE Transactions on Image Processing pp. 1194–1210 (2003).

[17] T. W. Freeman, *Median filter for reconstructing missing color samples*, US Patent 4,724,395 (1988).

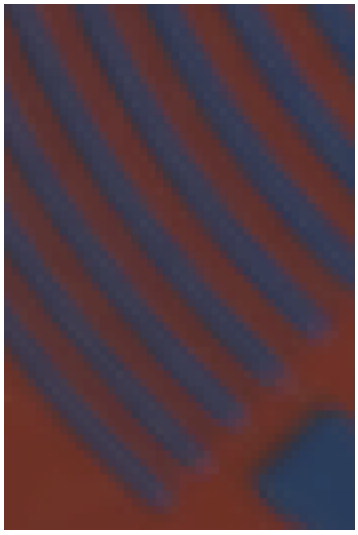
[18] C. W. Kim and M. G. Kang, *Noise insensitive high resolution demosaicing algorithm consid-*

(a)



(b)

(c)



(d)

FIG. 13: Example of visual comparison on colored resolution chart.(a) Real sensor image; (b) ROI by Hirakawa [11]; (c) ROI by Menon [10]; (d) ROI by proposed solution.

(a)

(b)

(c)

FIG. 14: Example of visual comparison on noise reduction effects.(a) ROI by Hirakawa [11]; (b) ROI by Menon [10]; (c) ROI by proposed solution.

(a) (b)

(c) (d)

(e)

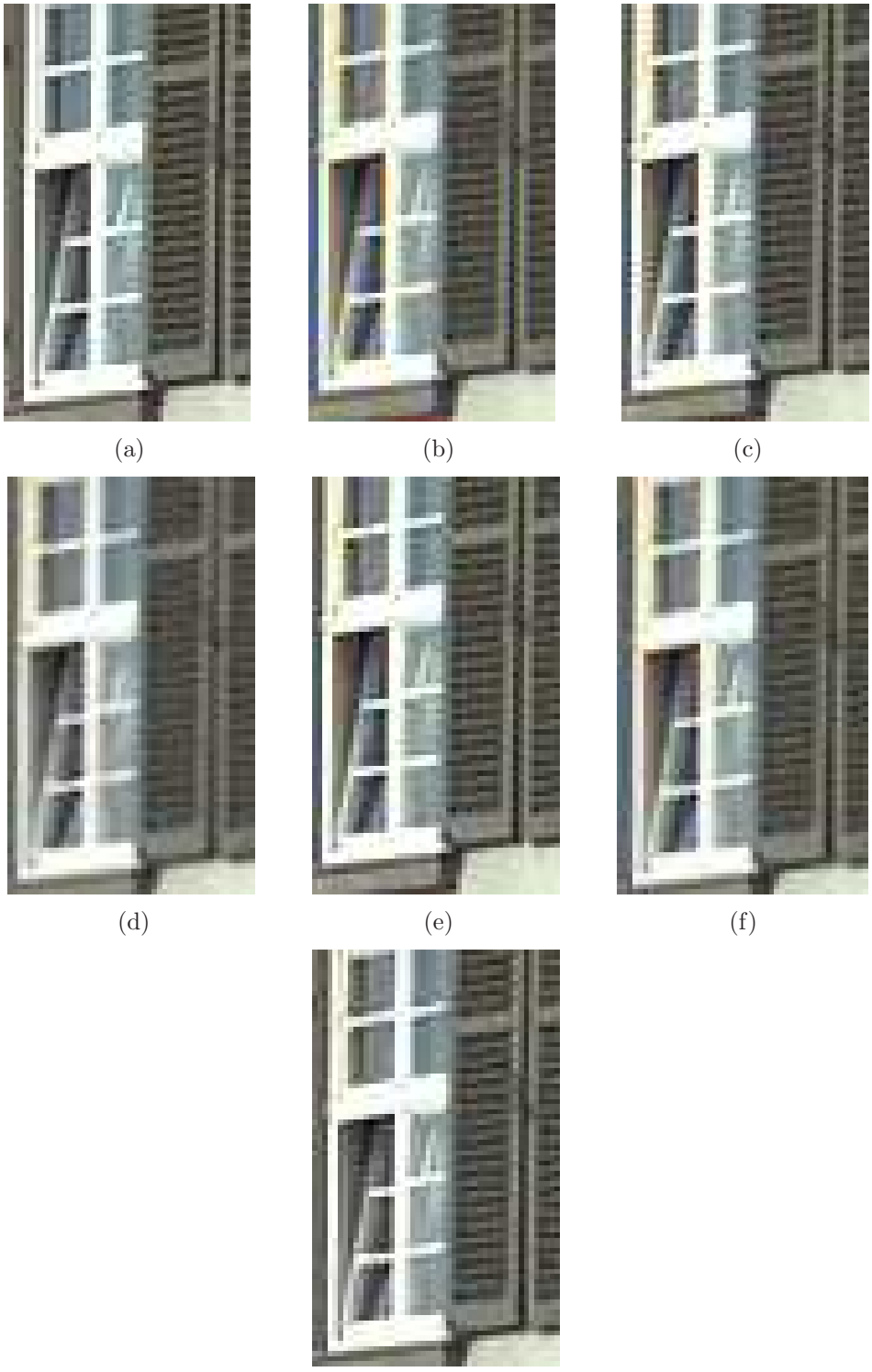
FIG. 15: Example of ROI visual comparison relative to the "lighthouse" test image (kodim19) corrupted by gaussian noise with sigma=12. (a) Original image; (b) Menon [10]; (c) Li [27]; (d) Wu and Zhang [9]; (e) Proposed solution.

ering cross-channel correlation, in *ICIP (3)* (2005), pp. 1100–1103.

[19] K. Hirakawa and T. W. Parks, *Adaptive homogeneity-directed demosaicing algorithm*, IEEE Transactions on Image Processing pp. 360–369 (2005), http://www.accidentalmark.com/ research/ packages/ MNdemosaic.zip.

[20] K. Chung, W. Yang, W. Yan, and C. Wang, *Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection*, IEEE Transactions on Image Processing pp. 2356–2367 (2008).

[21] M. Guarnera, V. Tomaselli, and G. Messina, *Method and relative device of color interpolation of an image acquired by a digital color sensor*, US Patent Application 2009/0010539 (2009).



(a) (b) (c)

(d) (e) (f)

(g)

FIG. 16: Example of ROI visual comparison relative to the "hotel" test image (kodim08). (a) Original image; (b) Interpolation through proposed approach (Sec.II); (c) image (b) filtered by Freeman [17]; (d) image (b) filtered by Lu [16]; (e) image (b) filtered by Lukac [23]; (f) image (b) filtered by Kim [18]; (g) Proposed solution.

[22] R. Kimmel, *Demosaicing: Image reconstruction from ccd samples*, IEEE Transactions on Image Processing pp. 1221–1228 (1999).

[23] R. Lukac and K. N. Plataniotis, *A robust, cost-effective postprocessor for enhancing demosaicked camera images*, Real-Time Imaging, Special Issue on Spectral Imaging II pp. 139–150 (2005), http://www.dsp.utoronto.ca/ ˜lukacr/ download.

[24] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, *Optimal weighted median filtering under structural constraints*, Signal Processing, IEEE Transactions on pp. 591–604 (1995), ISSN 1053-587X.

[25] R. Franzen, *Kodak lossless true color image suite*, Internet Link, http://r0k.us/ graphics/ kodak/.

[26] T. Chen, *A study of spatial color interpolation algorithms for single-detector digital cameras*, Internet Link, http://scien.stanford.edu/ class/ psych221/ projects/ 99/ tingchen/ main.htm.

[27] X. Li, *Demosaicing by successive approximation*, IEEE Transactions on Image Processing pp. 370–379 (2005).