Demosaicing using Optimal Recovery

D. Darian Muresan, Thomas W. Parks

Abstract

Color images in single chip digital cameras are obtained by interpolating mosaiced color samples. These samples are encoded in a single chip CCD by sampling the light after it passes through a color filter array (CFA) that contains different color filters (i.e. red, green, and blue) placed in some pattern. The resulting sparsely sampled images of the three-color planes are interpolated to obtain the complete color image. Interpolation usually introduces color artifacts due to the phase-shifted, aliased signals introduced by the sparse sampling of the CFAs. This paper introduces a non-linear interpolation scheme based on edge information that produces high quality visual results. The new method is especially good at reconstructing the image around edges, a place where the visual human system is most sensitive.

Index Terms

image modeling, quadratic classes, interpolation, denoising, demosaicing

I. INTRODUCTION

With the advent and proliferation of digital cameras and camcorders there is a dire need for good image modeling techniques to be used in applications such as image enlargement, demosaicing, and denoising. Modern single-chip CCD cameras with color filter arrays that use different lattices for each color pose interesting demosaicing problems in producing high quality color images. The CCD sensors in a single chip, color digital cameras are not capable of simultaneously sampling the red, green, and blue colors at the same pixel location[1]. Instead the camera uses a color filter array (CFA), as shown in Fig. 1, to sample different colors at different locations. The resulting sparsely sampled images of the three-color planes are interpolated to obtain dense images of the three-color planes and, thus, the complete color image. Interpolation usually introduces color artifacts (color moiré patterns) due to the phase shifted, aliased signals introduced by the sparse sampling of the CFAs. This paper discusses the application of optimal recovery interpolation [2], [3] to a non-linear CFA demosaicing scheme. Through examples it is shown that the approach presented in this paper produces good visual results when compared

---

[1]At the time of this work, Foveon's X3 image sensor technology has just been released. Foveon's X3 image sensor captures all three primary colors eliminating the need for demosaicing. However, at this time X3 has not crossed every technical hurdle. Issues such as cross-talk between different colors are still a problem [1]. With good DSP technology demosaicing will continue to play a role in the future of digital cameras, especially in low end cameras, such as those found in today's cell phones.
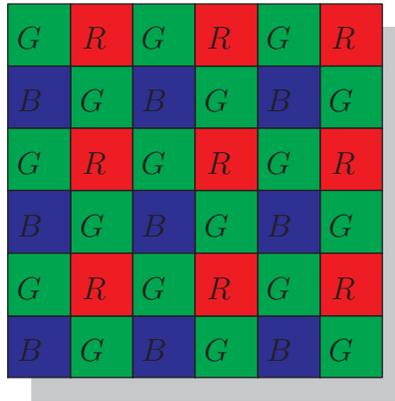
Fig. 1

DIGITAL CAMERA WITH SINGLE CCD CHIP USES A BAYER ARRAY PATTERN TO CAPTURE COLOR IMAGES.

against those obtained by linear interpolation and other recently published algorithms, such as the ones presented in [4], [5], [6], [7], [8].

The most basic demosaicing idea is to linearly and independently interpolate the R, G, and B planes. This type of interpolation, which is called linear interpolation, introduces serious aliasing artifacts. In recent years there has been a lot of interest in developing better demosaicing algorithms. In particular, the problem has been tackled from different angles including neural networks [9], B-splines [10], linear, minimized mean square estimators (LMMSE) [11], [12], frequency domain interpolators [5], gradient based methods [13], [14], [6], adaptive horizontal or vertical interpolation decisions [15], [16], and a wide range of edge directed algorithms. [8], [7], [17], [18], [19]. Recent reviews and discussions of some demosaicing solutions can be found in [20], [21], [4].

This paper presents an improved edge-directed demosaicing algorithm based on optimal recovery interpolation of gray scale images [2], [3]. The paper is organized as follows. Section II introduces the new demosaicing algorithm. Section III discusses the computational cost and Section IV presents the results.

II. Demosaicing

Linear demosaicing suffers from the decoupling of the R, G, and B planes. The red, green, and blue planes are very similar. Each plane depicts very similar images and each plane separately is a good representation of the gray scale image, especially if all the objects in the image have all three colors. Therefore, it is a valid assumption that object boundaries are the same in all three color planes, or that the high frequency components (edges depicting object boundaries) are similar in all three planes and equal with the high frequency component of the green plane. In frequency domain this means:

$$
\begin{aligned}
R &= R_L + G_H \\
G &= G_L + G_H \\
B &= B_L + G_H
\end{aligned}
$$
(1)

where underscore L and H stand for low-pass and high-pass components. From equation (1) it follows:

$$
\begin{aligned}
R_L - G_L &= R - G \\
B_L - G_L &= B - G
\end{aligned}
$$
(2)

Images $R_L - G_L$ and $B_L - G_L$ are low pass images and it follows from equation (2) that so are $R - G$ and $B - G$. Given that interpolation errors occur in high frequency regions it follows that it's better to interpolate the low-pass difference images between red and green and blue and green than to interpolate the red and blue planes separately. This observation has also been made by several other researchers [13], [14], [22], [5]. An alternate approach to the interpolation of the red and blue planes was proposed by [23], [8], [6] and is based on interpolating the ratios blue to green and red to green. The two approaches are tested on the color *lighthouse* image and the cropped results are depicted in Fig. 2. The ratio image maintains more of the high frequency in the fence, whereas the high frequency coefficients are much smaller in the difference image. In the example of Fig. 2 the image model of equation (1) is a better model.

A. Optimal Recovery

In [2], [3] we introduced a gray scale image interpolation algorithm based on optimal recovery estimation theory [25], [26], [27]. The algorithm works by estimating the missing sample of the
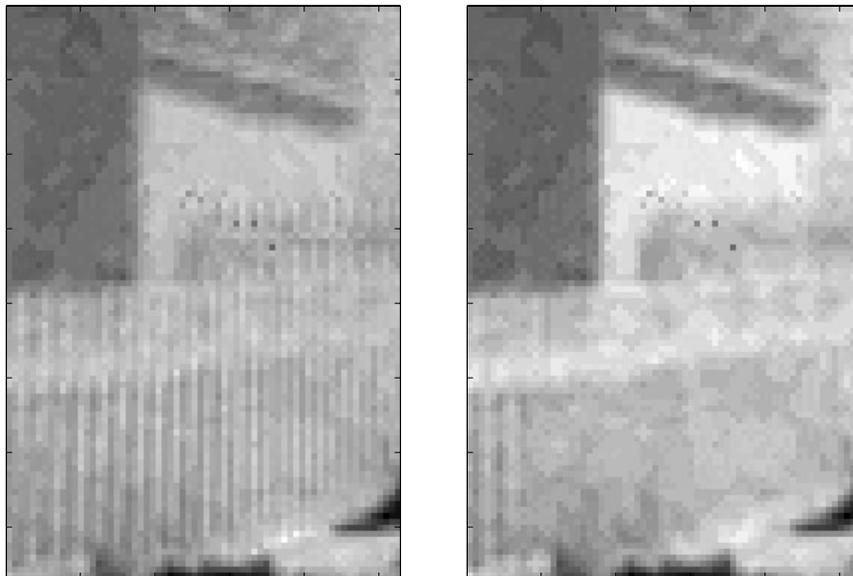
Fig. 2

IMAGE RATIO OF RED TO GREEN (LEFT) AND IMAGE DIFFERENCE OF RED TO GREEN (RIGHT) FOR A PORTION OF THE

FENCE IN *lighthouse* IMAGE. (FIGURE ALSO AVAILABLE ON LINE [24].)

local image patch **x** from the known samples of **x** and the assumption that the local patch belongs to a known quadratic signal class K, where K is defined as:

$$K = \left\{ \mathbf{x} \in R^n : \mathbf{x}^T \mathbf{Q} \mathbf{x} \le \epsilon \right\} \tag{3}$$

The optimal recovery estimate for the missing sample minimizes the maximum error over all possible vectors in K that have the same known samples as **x**. The inverse of matrix **Q** is the covariance matrix of the local image patches. (Local image patches are also called training vectors.) Formally, if local image patches are arranged as columns in matrix S then the quadratic signal class is:

$$\begin{aligned} K &= \left\{ \mathbf{x} \in R^n : \mathbf{x}^T \mathbf{Q} \mathbf{x} \le \epsilon \right\} \\ &= \left\{ \mathbf{x} \in R^n : \mathbf{x}^T \left(S S^T \right)^{-1} \mathbf{x} \le \epsilon \right\} \end{aligned}$$

The basic problem of image interpolation is that of approximating an unknown function **x** at pixel x_0 in terms of its known values at pixels $x_1, \ldots, x_k$, with the additional assumption that **x**

$$\mathcal{F}^h = \left\{ \mathbf{x} \in \mathcal{R}^n : F_i(\mathbf{x}) = f_i, \forall i \right\}$$
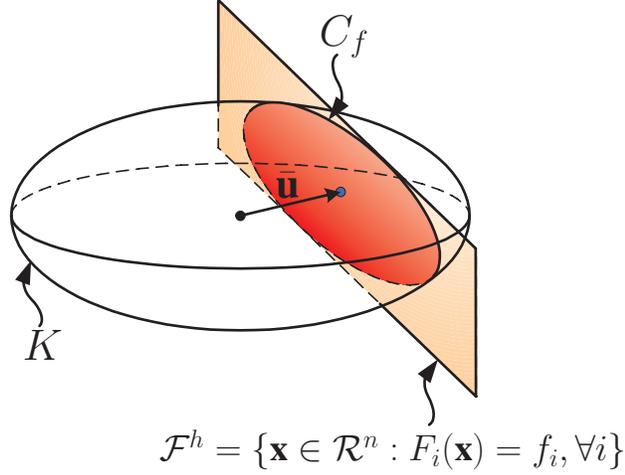
Fig. 3

INTERSECTION OF K WITH HYPER-PLANE $\mathcal{F}^h$.

is an element of K. More generally, the problem is to approximate a linear functional $F(\mathbf{x})$ in terms of other known linear functionals $F_1(\mathbf{x}), \ldots, F_k(\mathbf{x})$. (A linear functional $F(\mathbf{x})$ can be any linear function of $\mathbf{x}$, such as: samples, derivatives, integrals, etc.) Further, there is the assumption that F_i are linearly independent.

Noting the values of the functionals F_i by f_i, the unknown function $\mathbf{x}$ lies in the set C_f:

$$C_f = \left\{ \mathbf{x} \in \mathcal{R}^n : \mathbf{x}^T \mathbf{Q} \mathbf{x} \le \epsilon, F_i(\mathbf{x}) = f_i \text{ for } i = 1, \ldots, k \right\} \tag{4}$$

That is $\mathbf{x}$ lies in C_f, the hyper-circle defined by the intersection of the hyper-plane $\mathcal{F}^h = \left\{ \mathbf{x} \in \mathcal{R}^n : F_i(\mathbf{x}) = f_i, \forall i \right\}$ with the ellipsoid signal class K, as shown in Fig. 3. With $\mathbf{Q}$-norm of $\mathbf{x}$ defined as $\|\mathbf{x}\|_{\mathbf{Q}} = \mathbf{x}^T \mathbf{Q} \mathbf{x}$, let $\bar{\mathbf{u}}$ be the minimum $\mathbf{Q}$-norm signal in C_f:

$$\|\bar{\mathbf{u}}\|_{\mathbf{Q}} = \inf_{F_i(\mathbf{X})=f_i} \|\mathbf{x}\|_{\mathbf{Q}} \tag{5}$$

Then $F(\bar{\mathbf{u}})$ is the best approximation to the value of $F(\mathbf{x})$. That is $F(\bar{\mathbf{u}})$ is the Chebyshev center [28] of $F(\mathbf{x})$ on C_f:

$$\sup_{\mathbf{X} \in C_f} |F(\bar{\mathbf{u}}) - F(\mathbf{x})| = \inf_{\mathbf{u} \in C_f} \sup_{\mathbf{X} \in C_f} |F(\mathbf{u}) - F(\mathbf{x})| \tag{6}$$

Calculation of $\bar{\mathbf{u}}$ and $F(\bar{\mathbf{u}})$ is done using representors. By the Riesz representation theorem [29] there are elements $\phi, \phi_1, \ldots, \phi_k$ in $\mathcal{R}^n$ such that

$$F(\mathbf{x}) = (\phi, \mathbf{x})_{\mathbf{Q}}, \ \ F_i(\mathbf{x}) = (\phi_i, \mathbf{x})_{\mathbf{Q}}, \forall i \tag{7}$$

for all $\mathbf{x} \in K$. Vectors $\phi, \phi_1, \ldots, \phi_k$ are linearly independent since $F, F_1, \ldots, F_k$ are assumed to be linearly independent. Functionals $F_i(\mathbf{x}) = (\phi_i, \mathbf{x})_{\mathbf{Q}}$ remain constant for all $\mathbf{x} \in C_f$. That means subspace $\mathcal{F}$ is the set of all vectors in $\mathcal{R}^n$ orthogonal to the representors $\phi_i, \forall i$. Equivalently, $\phi_1, \ldots, \phi_k$ is a basis for $\mathcal{F}^\perp$. With $\bar{\mathbf{u}} \in \mathcal{F}^\perp$ it follows that $\bar{\mathbf{u}}$ is a linear combination of the representors ϕ_i:

$$\bar{\mathbf{u}} = \sum_i c_i \phi_i \tag{8}$$

Constants c_i are found by forcing $\bar{\mathbf{u}}$ to satisfy the given functionals:

$$
\begin{aligned}
F_i(\bar{\mathbf{u}}) &= (\phi_i, \bar{\mathbf{u}})_{\mathbf{Q}} & (9) \\
&= \left(\phi_i, \sum_j c_j \phi_j \right)_{\mathbf{Q}} & (10)
\end{aligned}
$$

With the adaptive optimal recovery in mind this paper proposes the following initial demosaicing algorithm:

1) Use optimal recovery interpolation to interpolate the green plane.
2) Use optimal recovery interpolation to interpolate the differences red to green and blue to green.

The details of our demosaicing algorithm are presented next.

B. The Green Plane

Optimal recovery interpolation for the green plane is a modification of the gray scale interpolation algorithm [2], [3]. The difference lies in the selection and shape of the training vectors used for learning the local quadratic signal class. Training set $\mathcal{S}$ (i.e. the collection of local image patches) is obtained by selecting the nearest vectors (both in space and value) to the estimated vector (vector $\mathbf{x}$ using the notation of previous section). Training set $\mathcal{S}$ is formed as follows:

1) **Closeness in space.** Decide on the size of the local window that is centered around vector $\mathbf{x}$. Vectors inside this window are close in space to $\mathbf{x}$ and they determine the initial training set $\mathcal{S}$. If the window size is too large local features of the image are missed and smaller
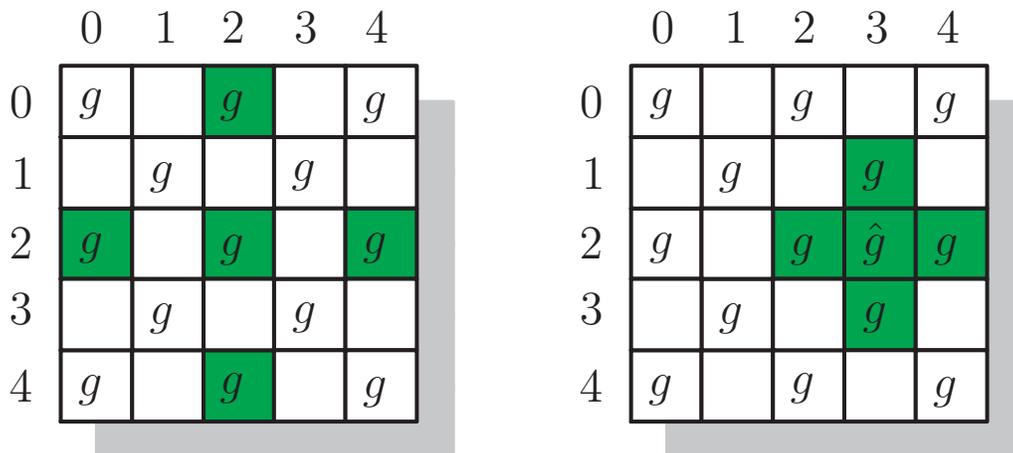
Fig. 4

window sizes seem to perform better than larger ones. Experimentally, a window of 15×15 pixels seems to perform well for a 256×256 image.

2) **Closeness in value.** If image patch $\mathbf{x}$ is close to a boundary region of two distinct textures (for example $\mathbf{x}$ is a fence patch at the boundary between the grass and fence in the *lighthouse* image) then half the vectors in $\mathcal{S}$ are more like $\mathbf{x}$ (i.e. more like the fence texture) and half are distinctly different (i.e. more like grass patches). Using this training set for learning the local quadratic class results in a class that is strongly influenced by the wrong training vectors, negatively affecting the interpolation algorithm. In the *lighthouse* image the vertical fence pattern is disrupted by the horizontal boundary line between the fence and grass and this introduces horizontal lines in the fence as shown on the left of Fig. 5. This problem can be alleviated by pruning $\mathcal{S}$ to contain only half of the vectors that are closest to $\mathbf{x}$ (i.e. only fence patches). (In the interpolation case $\mathbf{x}$ contains at least one missing sample. The distance between $\mathbf{x}$ and the vectors in $\mathcal{S}$ is calculated based on the known samples of $\mathbf{x}$.) Under most conditions boundary regions are split in half by the local window and taking half of the closest vectors is enough to guarantee good boundary interpolation, as shown on the right of Fig. 5. Notice the difference in the left and right
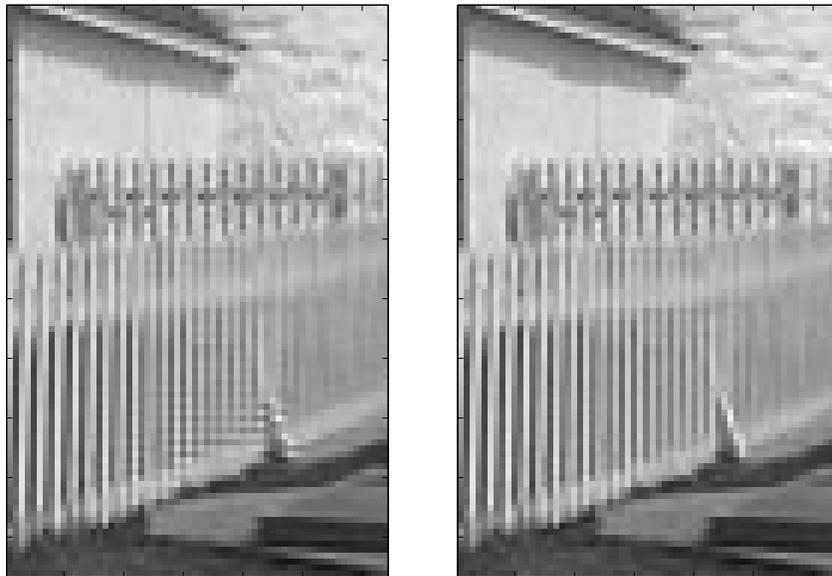
Fig. 5

INTERPOLATION OF THE GREEN CHANNEL IN THE *lighthouse* IMAGE USING THE ENTIRE TRAINING SET $\mathcal{S}$ (LEFT) AND THE

PRUNED SET $\mathcal{S}$ AS DISCUSSED IN THE TEXT (RIGHT). (FIGURE ALSO AVAILABLE ON LINE [24].)

images of Fig. 5, especially around boundary regions: the boundary between the fence and the grass, and the boundary between the shadow of the roof and the house siding.

3) **Shape of training vectors.** The quincunx sampling of the green plane forces a certain structure on the shape of the training vectors. Samples of one coarse scale vector are the shaded pixels on the left of Fig. 4 and samples of one fine scale vector are the shaded pixel on the right of Fig. 4.

Interpolation of the green plane proceeds as follows:

1) **Determine the local quadratic class.** For each local region use coarse scale vectors (Fig. 4-left) to generate set $\mathcal{S}$ as discussed above.

2) **Use optimal recovery to estimate the missing sample.** The fine scale vector $\mathbf{x}$ (formed by the shaded pixels of Fig. 4-right) contains four known samples at locations $(2,2), (1,3), (2,4)$ and $(3,3)$. The unknown sample is at location $(2,3)$. The known linear functionals are $F_1(\mathbf{x}) = \mathbf{x}_{(2,2)}, F_2(\mathbf{x}) = \mathbf{x}_{(1,3)}, F_3(\mathbf{x}) = \mathbf{x}_{(2,4)}$ and $F_4(\mathbf{x}) = \mathbf{x}_{(3,3)}$. The representors of the known functionals (i.e. $\{\phi_1, \ldots, \phi_4\}$) are products between $\mathbf{Q}^{-1}$ and vectors with 1 in the
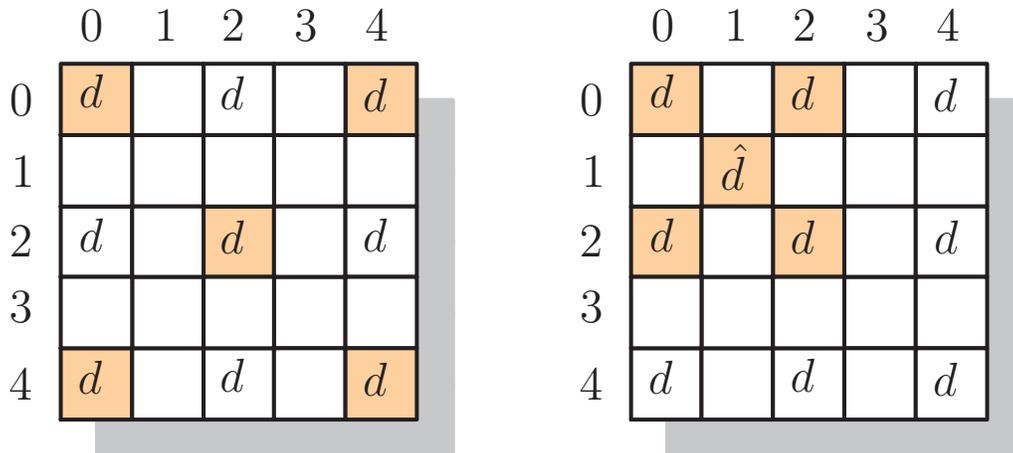
Fig. 6

SHADED PIXELS REPRESENT THE COARSE SCALE (LEFT) AND FINE SCALE (RIGHT) SAMPLES OF SOME LOCAL VECTORS. PIXELS LABELED WITH d ARE THE BAYER ARRAY RED-GREEN DIFFERENCE PIXELS AND PIXEL $\hat{d}$ IS THE INTERPOLATED DIFFERENCE.

location of the known values and zero everywhere else (i.e. for F_1 the vector would have 1 at location $(2, 2)$). The optimal recovery solution is vector $\bar{\mathbf{u}}$ which is a linear combination of the representors. The estimate of the missing pixel is the sample of $\bar{\mathbf{u}}$ at location $(2, 3)$.

3) **Second pass.** Once the missing green pixels are estimated, re-run optimal recovery interpolation. This time form the training set $\mathcal{S}$ using fine scale vectors (Fig. 4-right). Experimentally, this step is necessary only for high frequency patterns. It tends to improve the results, but it may not always be necessary.

C. The Reds and Blues

After the green plane is interpolated optimal recovery interpolation is applied to the decimated differences of red to green and blue to green. To obtain the full image this requires a $2\times$ interpolation factor. Optimal recovery interpolation for the difference image is the same as our $2\times$ interpolation algorithm presented in [2] and is similar to the algorithm for green pixels. To interpolate from the decimated image to quincunx sampling use coarse scale training vectors (Fig. 6-left) to learn K. Then estimate the missing pixel at location $(1, 1)$ (Fig. 6-right) using

known functionals at locations $(0,0), (0,2), (2,0)$, and $(2,2)$. To go from quincunx sampling to the full image use the same interpolation as for the green plane.

The difference image is low-passed and it does not contain sharp edges, as shown in Fig. 2. Hence, the interpolation in this second step can be replaced with a simpler and faster interpolation algorithm in order to increase speed. For blue and red interpolation we obtained better interpolation results using the weighted gradient algorithm of [6] applied to differences instead of ratios.

III. Computational Complexity

With S a matrix of n rows and m columns, the computational cost of estimating one pixel using optimal recovery is dominated by the computation of matrix $\mathbf{Q}^{-1} = SS^T$. It requires $n^2(2m-1)$ multiplications and additions. Taking into account the symmetry of $\mathbf{Q}$, the number of multiplications and additions can be reduced to $(n^2/2 + n/2)(2m-1) = n^2m + O(nm)$ operations. In our demosaicing examples $n = 5$ and with a window size of radius 15, $m = 98$ (after pruning). This means that the computational cost for one pixel is on the order of 2500 multiplications, additions, or comparisons. For an image of the size $M \times N$ there are $2MN$ pixels that need to be interpolated. Not considering second passes (as in step 3 of the green channel interpolation) the complexity of our demosaicing algorithm is on the order of $5000MN$, which is roughly an order of magnitude slower than the demosaicing algorithm of [4]. If optimal recovery is applied only to green pixels computational complexity drops to $1200MN$ plus the additional cost of interpolating the red and blue pixels. For this reason it is strongly recommended that the demosaicing algorithm for the red and blue channel is a faster algorithm. The blue and red interpolation algorithm of [6] applied to color differences, instead of ratios, performs well. In particular, assume the red colors are interpolated at pixel P_5 of Fig. 7. First interpolate the missing reds at the blue locations

$$R_5 = G_5 + \frac{E_1\left(R_1 - G_1\right) + E_3\left(R_3 - G_3\right) + E_7\left(R_7 - G_7\right) + E_9\left(R_9 - G_9\right)}{E_1 + E_3 + E_7 + E_9} \tag{11}$$

and second, interpolate the missing reds at the green locations.

$$R_5 = G_5 + \frac{E_2\left(R_2 - G_2\right) + E_4\left(R_4 - G_4\right) + E_6\left(R_6 - G_6\right) + E_8\left(R_8 - G_8\right)}{E_2 + E_4 + E_6 + E_8} \tag{12}$$

where the weights E_i are determined as in [6]. The blue pixels are interpolated in the same manner as the red ones.
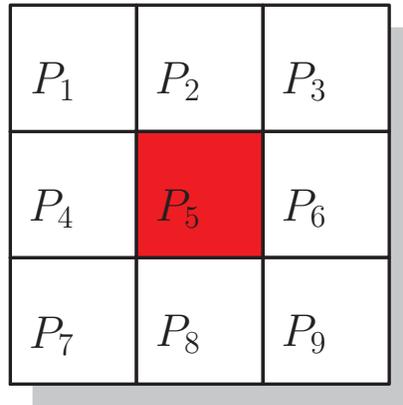
Fig. 7

INTERPOLATING PIXEL P_5.

The final proposed algorithm is:

1) Apply optimal recovery interpolation to the green channel.

2) Interpolate the blue and red channel using the modified algorithm of [6].

and we refer to it as *optimal recovery demosaicing*.

IV. RESULTS

The new method is compared against three other algorithms: linear, weighted gradient of [6] and the alternating projections algorithm of [4]. The input images used for this test are Kodak Photo CD images of 384×256 pixels. The full color images are down-sampled as in Fig. 1 (with the same top-left pattern) to obtain the Bayer array mosaiced images. The mosaiced images are then interpolated back to full color images. Results are reported using sample color images, PSNR[2] values and in some cases the error norm is reported in the S-CIELab metric [30], [31]. The demosaiced results presented in this paper can be viewed and downloaded on line from [24].

First, we test the four demosaicing algorithms on the *lighthouse*, *sails*, *window* and *statue* images. The color image results of [6] were obtained directly from the author's web page in a

[2]PSNR is calculated as $10 \times \log_{10} \left(255^2 / var(orig - demosaic) \right)$

TABLE I

PSNR VALUES (DB). THE DIFFERENT METHODS ARE: LINEAR, WEIGHTED GRADIENT OF [6], ALTERNATE PROJECTIONS OF
[4], OPTIMAL RECOVERY WITH TWO PASSES, OPTIMAL RECOVERY WITH ONE PASS, AND OPTIMAL RECOVERY APPLIED TO
THE RED AND BLUE CHANNELS.

Method	Lighthouse	Sails	Window	Statue
Linear	24.70	27.22	28.34	27.32
Weight. Gr.	32.94	36.16	35.71	33.19
Alt. Proj.	35.49	37.83	**38.91**	**36.89**
Opt. Rec.	**36.12**	38.41	38.05	36.74
Opt. Rec. One	35.57	**38.52**	38.42	36.88
Opt. Rec. Diff.	31.83	35.75	33.26	31.43

lossless PPM format and the algorithm of [4] was obtained directly from the author. The cropped results are displayed in Fig. 8-11. In order to better discriminate the color artifacts the cropped images are enlarged 4×4 times using pixel replication. In Fig. 8 the bluish color artifacts in the fence are removed best using the optimal recovery algorithm. The cropped version of the *sails* image is shown in Fig. 9. The color artifacts in linear and weighted gradient are very obvious. In all four images there is a noticeable zippering color artifact (i.e. noticeable color differences in neighboring pixels). The artifact is most pronounced around the numbers and at the boundary between the yellow and blue colors. In the four images the zippering effect is least pronounced in the optimal recovery result. Fig. 10 depicts a cropped version of the *window* image. The color differences are harder to notice, but looking at the four results the optimal recovery image again seems to have the smallest zippering artifact. Finally, Fig. 11 depicts the cropped version of the statue image. While the artifacts between the first two and the last two images are easier to notice, the difference between the alternating projection algorithm and optimal recovery are much more difficult to observe.

Our next evaluation is based on comparisons of PSNRs. Table I lists the PSNR values for the four demosaicing algorithms. In all cases optimal recovery outperforms linear and weighted gradient demosaicing. For the *lighthouse* and *sails* image optimal recovery outperforms the alternating projection algorithm, while for the *window* and *statue* image the alternating projection algorithm performs better. The fifth row of Table I shows the PSNR values when using only

TABLE II

PSNR FOR *birds* DATA SET.

Method	Num 1	Num 2	Num 3	Num 4	Num 5	Num 6	Num 7
Alt. Proj.	**41.53**	**38.02**	**34.13**	40.21	44.84	37.88	44.16
Opt. Rec.	41.52	37.69	32.95	**41.13**	**45.25**	**38.77**	**44.43**

TABLE III

PSNR FOR *flowers* DATA SET.

Method	Num 1	Num 2	Num 3	Num 4	Num 5	Num 6	Num 7
Alt. Proj.	36.05	32.15	**32.55**	**35.66**	28.41	32.91	**30.42**
Opt. Rec.	**36.59**	**32.46**	32.35	35.54	**29.69**	**33.76**	30.27

TABLE IV

PSNR FOR *structures* DATA SET.

Method	Num 1	Num 2	Num 3	Num 4	Num 5	Num 6	Num 7
Alt. Proj.	34.01	32.18	**26.40**	**31.71**	**33.69**	36.21	35.98
Opt. Rec.	**34.15**	**32.19**	25.89	31.30	33.16	**37.12**	**36.08**

one pass in interpolating the green channel. Notice that in most cases using one pass gives better PSNR values, but using two passes improves the *lighthouse* results by 0.55 dB. The last row of Table I shows the results when optimal recovery is applied to the differences of red to green and blue to green. The results are slightly worse than the weighted gradient algorithm. This seems surprising at first. However, interpolating the red and blue channels uses training vectors that are coarser (i.e. the location of the samples in the training vectors are more distant from each other) and therefore there is more error in estimating the proper local quadratic class, which is particularly bad for textured regions. On the positive side, using optimal recovery on the difference image is not recommended since the computational time almost triples.

To further test the difference between the two algorithms optimal recovery is compared against alternating projection on a much larger data set containing image classes of: birds, faces, flowers, nature, patterns, structures, and textures. Due to space limitations the TIF results from these tests and original images are available only on line [24]. Looking strictly at PSNR values, the image database seems to be very evenly divided into images for which optimal recovery gives better

values and images for which alternating projections gives better values. A subset of these PSNR values are listed in Tables II-IV. In this case PSNR values do not provide a convincing argument for the superiority of one algorithm over the other. Here a subjective assessment seems to be more valuable. Looking at the images presented in Fig. 8-11 and on line [24] the results suggest that in textured regions the alternating projections algorithm performs best, while in regions of well defined edges optimal recovery produces better results. The intuition behind these results is that optimal recovery's strength is reconstructing well defined (i.e. longer) edges. In regions of long edges the training data is more representative of the local edge and the algorithm determines a quadratic signal class that better represents the edge behavior. In textured regions, where edges tend to be shorter and in different directions, the local training set is not as representative of the local data and the algorithm introduces more errors. The *lighthouse* and *sails* images contain such long edges: the fence and the mast. On the other hand, images *window* and *statue* have large texture regions with poorly defined edges.

In evaluating image quality we also compared the different methods using S-CIELab metrics[3] [30], [31]. The results are shown in Table V. In this case the alternating projection algorithm consistently gives a smaller (better) value for the metric when evaluated over the entire image. To demonstrate the visual impression that optimal recovery is better in regions with edges, we evaluated the S-CIELab metric on selected regions of the image. For each image there are two types of patches being considered: a textured patch and a strong edge patch, as in Fig. 12. The S-CIELab results are shown in Table VI. It is now clear that in regions with strong edges, where the human eye is most sensitive, optimal recovery performs well, while in textured regions, where vision is not as sensitive, its performance deteriorates.

From our subjective results the most objectionable artifacts seem to be artifacts in regions of strong edges and not artifacts in textured regions. This seems to indicate that optimal recovery is the preferred algorithm from a visually subjective point of view. As suggested by one of the reviewers, an alternative approach would be to consider a hybrid algorithm where optimal recovery demosaicing is applied around edges, while a different demosaicing algorithm is applied in textured regions. This can be an area of future research.

---

[3]For the S-CIELab metric we used the default viewing conditions that came with the Matlab sample code.

TABLE V

DISTANCE IN S-CIELAB METRICS FOR THE ENTIRE IMAGE.

Method	Lighthouse	Sails	Window	Statue
Linear	1318.22	917.27	1126.13	908.97
Weight. Gr.	1056.43	597.54	1076.00	681.63
Alt. Proj.	**558.97**	**419.25**	**448.21**	**388.67**
Opt. Rec.	573.59	433.59	510.0	493.19

TABLE VI

DISTANCE IN S-CIELAB METRICS FOR DIFFERENT IMAGE PATCHES. THE COORDINATES OF EACH PATCH ARE IN MATLAB

NOTATION. IMAGE PATCHES ARE SHOWN IN FIG. 12.

Method	Lighthouse		Sails	
	Texture	Edge	Texture	Edge
	[292:363,105:202]	[166:262,87:151]	[264:360,18:82]	[149:245,113:177]
Linear	215.24	552.89	325.63	294.16
Weight. Gr.	138.69	410.80	191.33	195.73
Alt. Proj.	**113.56**	197.12	**152.69**	142.15
Opt. Rec.	139.00	**130.47**	159.16	**125.82**

V. CONCLUSION

This paper developed a novel demosaicing algorithm using optimal recovery interpolation [2]. Through examples and PSNR values it demonstrated that the new method performs especially well in regions of strong edges, where most other algorithms fail and where the human visual system seems most sensitive to errors.

VI. ACKNOWLEDGMENT

We are grateful to Ron Kimmel for the helpful feedback on his algorithm, to Bahadir K. Gunturk for providing his demosaicing code, and to our reviewers for their valuable suggestion.

REFERENCES

[1] P. Gwynne, "Tricolor sensors create a sharper image," in *IEEE Spectrum*, May 2002, pp. 23–24.

[2] D. D. Muresan and T. W. Parks, "Optimal recovery approach to image interpolation," in *IEEE Proc. ICIP.*, vol. 3, 2001, pp. 7–10.

[3] ——, "Adaptively quadratic (AQua) image interpolation," *To appear in IEEE Trans. Image Processing*, 2004.

[4] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, pp. 997–1013, 2002.

[5] J. Glotzbach, R. Schafer, and K. Illgner, "A method of color filter array interpolation with alias cancellation properties," in *IEEE Proc. ICIP.*, 2001, pp. 141–144.

[6] R. Kimmel, "Demosaicing: Image reconstruction from color ccd samples," *IEEE Trans. Image Processing*, vol. 8, pp. 1221–1228, 1999.

[7] J. J.E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," in *Proceedings of SPIE*, vol. 3028, 1997, pp. 117–125.

[8] D. R. Cok, "Reconstruction of ccd images using template matching," in *Proc. of IS&T's Annual Conference/ICPS*, 1994, pp. 380–385.

[9] J. Go and C. Lee, "Interpolation using neural network for digital still cameras," in *2000 Digest of Tech. Papers. Int. Conf. Cons. Elec.*, 2000, pp. 176–177.

[10] B.Lee, J. Kim, and C. Lee, "High quality image interpolation for color filter arrays," in *Proc. Int. Conf. Systems, Man, and Cybernetics*, vol. 2, 2000, pp. 1547–1550.

[11] D. Taubman, "Generalized wiener reconstruction of images from color sensor data using a scale invariant prior," in *Proc. Int. Conf. Image Proc.*, 2000, pp. 801–804.

[12] H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicing," *IEEE Transactions on Image Processing*, vol. 11, pp. 485–492, 2002.

[13] C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," U.S. Patent 5,373,322, 1994.

[14] J. E. Adams and J. F. Hamilton, "Adaptive color plan interpolation in single sensor color electric camera," U.S. Patent 5,506,619, 1996.

[15] J. F. Hamilton and J. E. Adams, "Adaptive color plan interpolation in single sensor color electric camera," U.S. Patent 5,629,734, 1997.

[16] J. E. Adams and J. F. Hamilton, "Adaptive color plan interpolation in single sensor color electronic camera," U.S. Patent 5,652,621, 1996.

[17] T. Kuno, H. Sugiura, and N. Matoba, "New interpolation method using discriminated color correlation for digital still cameras," *IEEE Transaction Consumer Electronics*, vol. 45, no. 1, pp. 259–267, 1999.

[18] E. Chang, C. Shiufun, and D. Pan, "Color filter array recovery using a threshold-based variable number of gradients," in *Proceedings of SPIE*, vol. 3650, 1999, pp. 36–43.

[19] X. Li, "Edge directed statistical inference with applications to image processing," Ph.D. dissertation, Princeton University, Princeton, NJ, 2000.

[20] J. Mukherjee, R. Parthasarathi, and S. Goyal, "Markov random field processing for color demosaicing," *Pattern Recognition Letter*, vol. 22, pp. 339–351, 2001.

[21] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 123–132, 2002.

[22] W. T. Freeman and Polaroid Corporation, "Method and apparatus for reconstructing missing color samples," U.S. Patent 4,774,565, 1998.

[23] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent 4,642,678, 1987.

[24] (2002) The DSP website. [Online]. Available: http://dsplab.ece.cornell.edu/papers/results/ordemosaic/

[25] M. Golomb and H. F. Weinberger, *On Numerical Approximation*. R. E. Langer Ed. The University of Wisconsin Press, 1959.

[26] C. A. Michelli and T. J. Rivlin, *Optimal Estimation in Approximation Theory*. Eds. New York: Plenum 1976, 1976.

[27] D. D. Muresan, "Review of optimal recovery," Cornell University, Tech. Rep., 2002. [Online]. Available: http://dsplab.ece.cornell.edu/papers

[28] R. Shenoy and T. W. Parks, "An optimal recovery approach to interpolation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1987–1996, 1992.

[29] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, Inc., 1976.

[30] (2003) S-cielab metric. [Online]. Available: http://white.stanford.edu/~brian/scielab/scielab.html

[31] X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, "Color image quality metric s-cielab and its application on halftone texture visibility," in *IEEE Compcon Proceedings*, 1997, pp. 44–48.

D. Darian Muresan received his B.S. degree in Mathematics and Electrical Engineering from University of Washington, Seattle, WA., and his M.Eng. and Ph.D. degrees in Electrical and Computer Engineering from Cornell University, Ithaca, NY. His interests include image and signal processing, and hardware design. He is the co-inventor of an Analog-to-Digital converter and has several other patents pending. He interned with HP as a Hardware Design Engineer and is the founder of Digital Multi-Media Design (http://www.dmmd.net).

Thomas W. Parks B.E.E., M.S., Ph.D.(Cornell) - From 1967-86 he was on the Electrical Engineering faculty at Rice University, Houston, TX. In 1986 he joined Cornell as a Professor of Electrical Engineering. He is a Fellow of the IEEE and a recipient of the IEEE Third Millennium Medal. He received the Humboldt Foundation Senior Scientist Award, and has been a Senior Fulbright Fellow. He has co-authored a number of books on digital signal processing. His research interests are signal theory and digital signal processing.

Fig. 8

LIGHTHOUSE FENCE (LEFT TO RIGHT, TOP TO BOTTOM): LINEAR, WEIGHTED GRADIENT, ALTERNATING PROJECTIONS, AND OPTIMAL RECOVERY. (AVAILABLE ON LINE [24].)

Fig. 9

SAILS NUMBER (LEFT TO RIGHT, TOP TO BOTTOM): LINEAR, WEIGHTED GRADIENT, ALTERNATING PROJECTIONS, AND

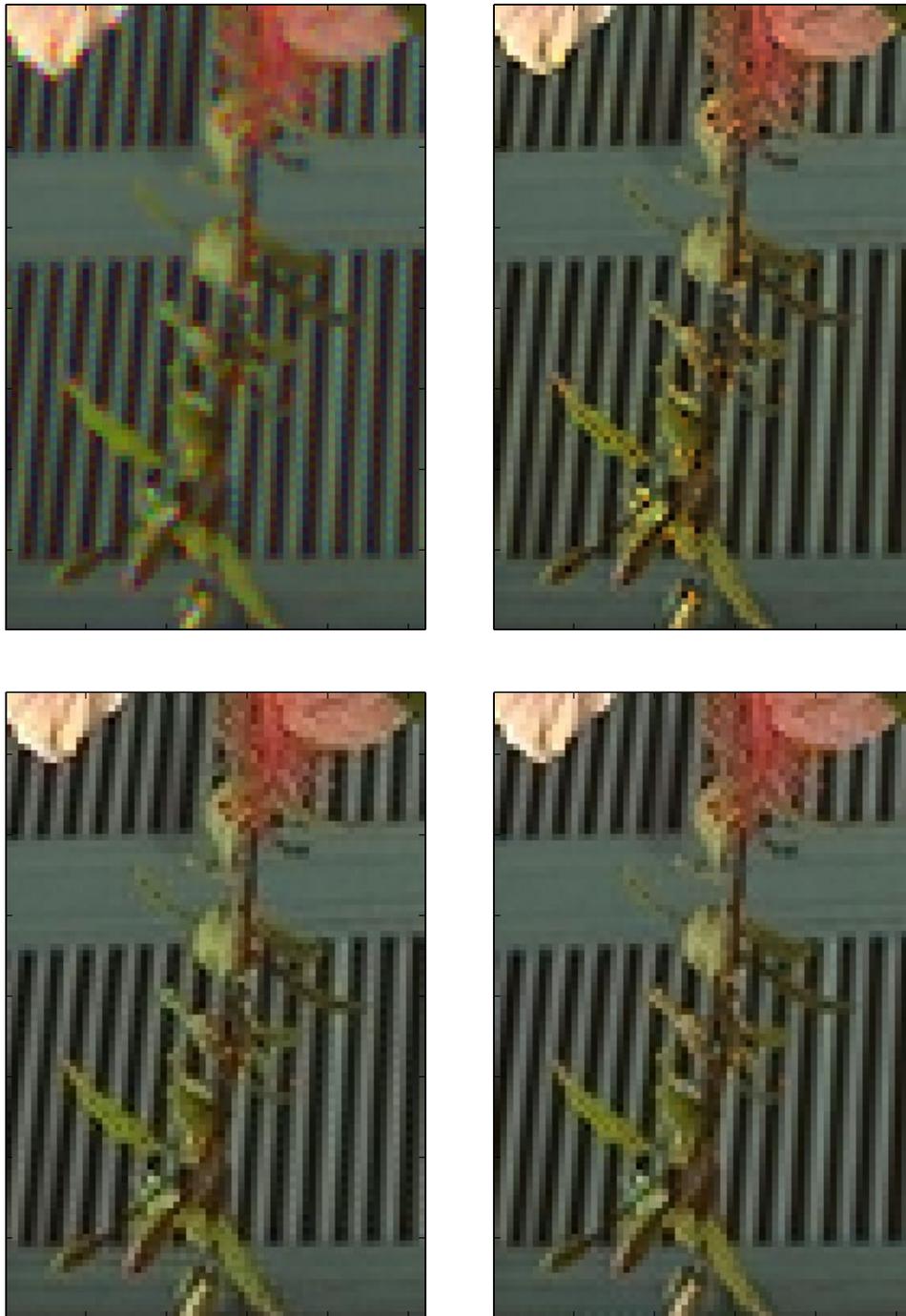OPTIMAL RECOVERY. (AVAILABLE ON LINE [24].)

Fig. 10

WINDOW (LEFT TO RIGHT, TOP TO BOTTOM): LINEAR, WEIGHTED GRADIENT, ALTERNATING PROJECTIONS, AND OPTIMAL
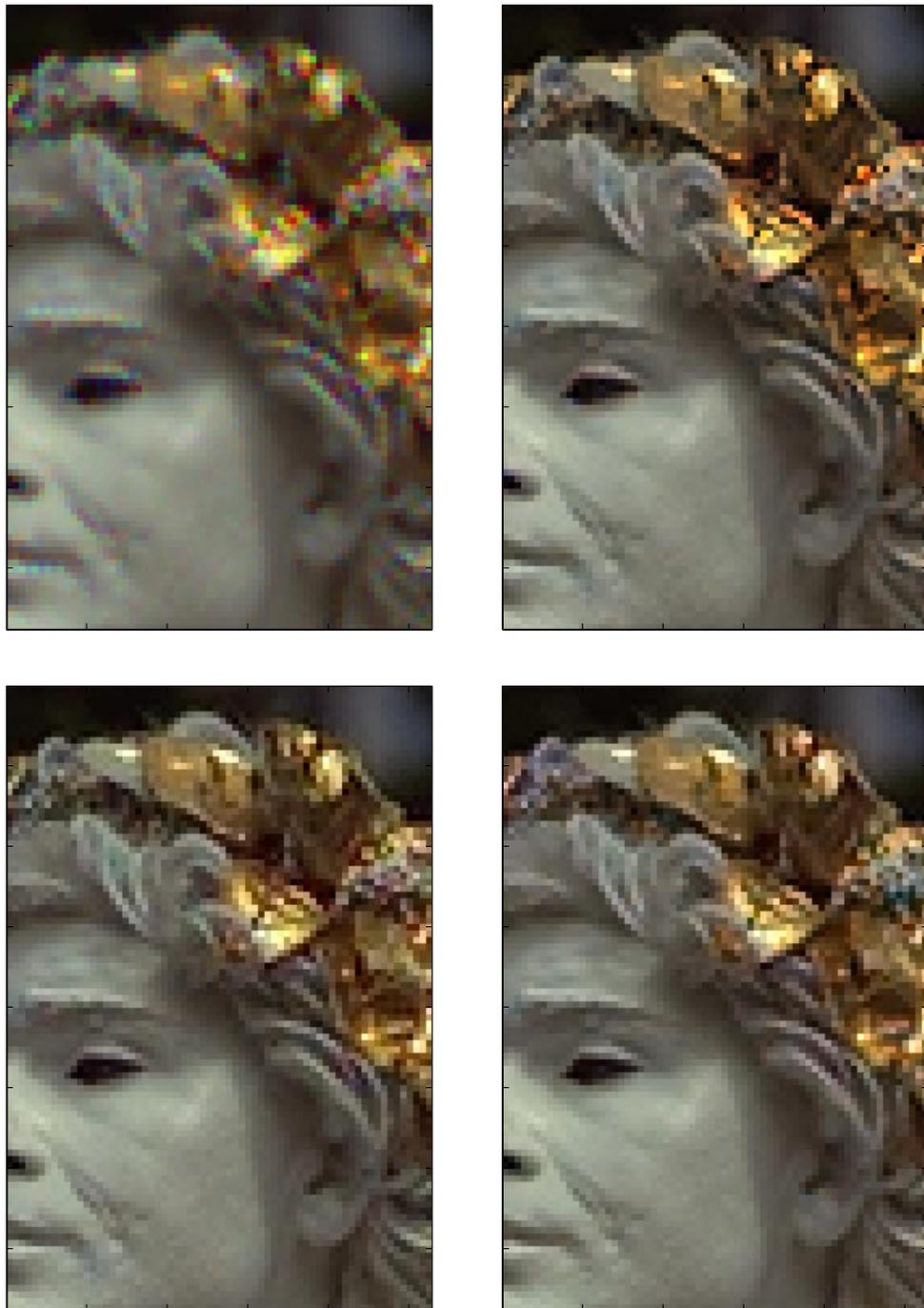
RECOVERY. (AVAILABLE ON LINE [24].)

Fig. 11

STATUE (LEFT TO RIGHT, TOP TO BOTTOM): LINEAR, WEIGHTED GRADIENT, ALTERNATING PROJECTIONS, AND OPTIMAL
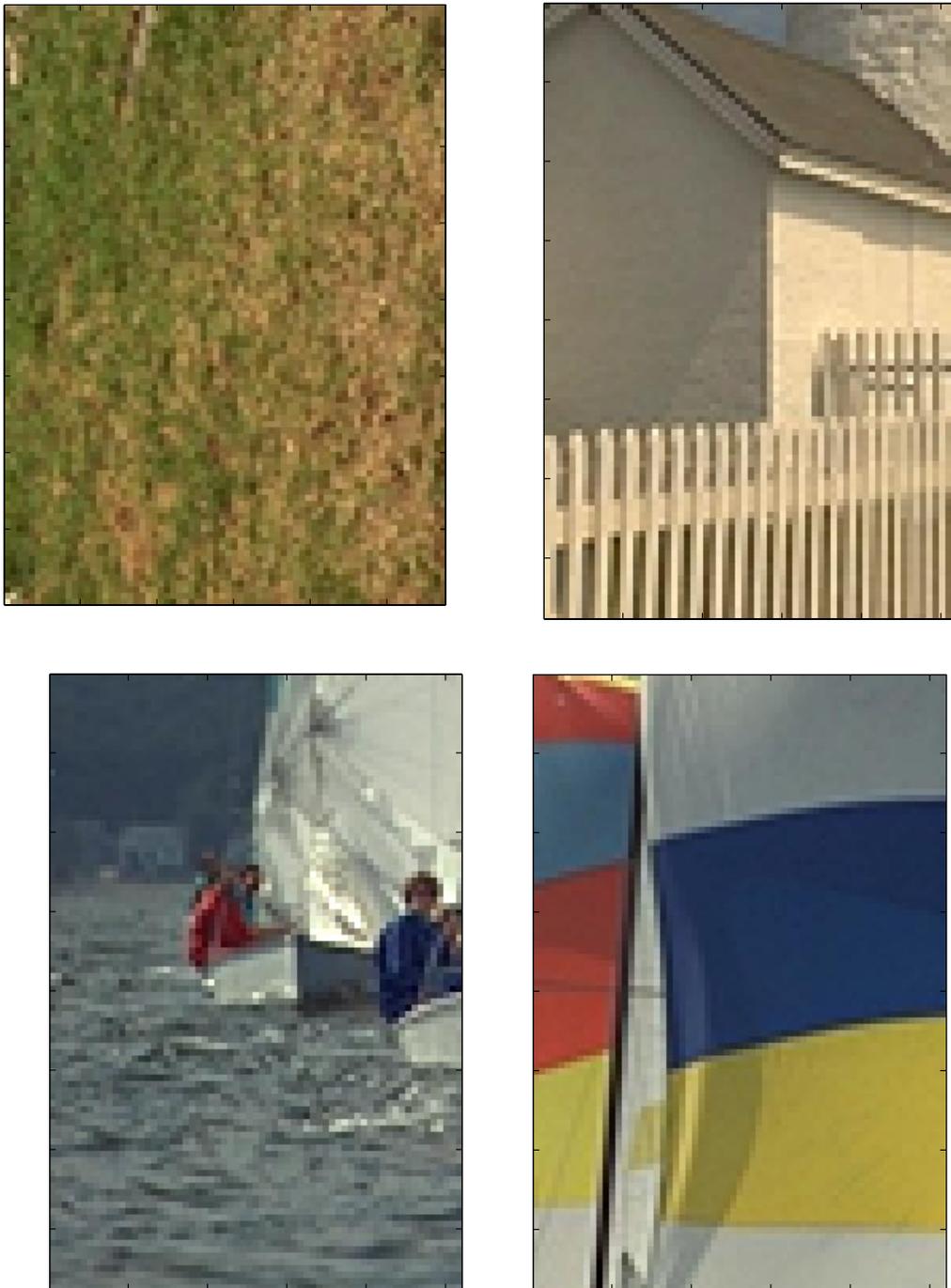
RECOVERY. (AVAILABLE ON LINE [24].)

Fig. 12

LIGHTHOUSE TEXTURE (UPPER-LEFT) AND EDGE (UPPER-RIGHT) REGIONS. SAILS TEXTURE (LOWER-LEFT) AND EDGE

(LOWER-RIGHT) REGIONS.)