GRAPH-BASED REGULARIZATION FOR COLOR IMAGE DEMOSAICKING

Chenhui Hu[1]*, Lin Cheng*[2]*, Yue M. Lu*[1]

[1]School of Engineering and Applied Sciences
Harvard University
{hu4,yuelu}@seas.harvard.edu

[2]Department of Engineering
Trinity College
lin.cheng@trincoll.edu

ABSTRACT

We present a novel regularization framework for demosaicking by viewing images as smooth signals defined on weighted graphs. The restoration problem is formulated as a minimization of variation of these graph-domain signals. As an initial step, we build a weight matrix which measures the similarity between every pair of pixels, from an estimate of the full color image. Subsequently, a two-stage optimization is carried out: first, we assume that the graph Laplacian is signal dependent and solve a non-quadratic problem by gradient descent; then, we pose a variational problem on graphs with a fixed Laplacian, subject to the constraint of consistency given by available samples in each color channel. Performance evaluation shows that our approach can improve existing demosaicking methods both quantitively and visually, by reducing color artifacts.

***Index Terms*—** Demosaicking, weighted graph, Laplacian, regularization method

1. INTRODUCTION

Color image demosaicking represents the process of interpolating a full color image from a subsampled version, which is a result of using color filter array (CFA) in digital cameras. Existing works devoted to perform the task are quite extensive and [1] serves a nice overview. While most of the algorithms rely on local structures of the images (e.g. edges, corners, curvatures), non-local methods [3] [8] are of particular interest since they can exploit the global structure. This paper sets up a new non-local demosaicking approach by viewing image as smooth signal on a weighted graph and perform a graph-based regularization. First, we build a weight matrix to measure the similarity between every pair of pixels. This weight matrix corresponds to a weighted graph by viewing the pixels as nodes in the graph. Then, we solve an optimization problem to minimize the variation of the image on this graph under the constraint of observations. This framework is featured with several advantages: one is the superiority of the non-local nature when local structures can not be inferred from the neighboring pixels; the other is the independence with the CFA pattern of the image, making it applicable in a much wider range. As related works, Menon et al. proposed a regularization method in [2] by exploiting smoothness of each color component and correlation between the different channels; In [8], Elmoataz et al. introduced a non-local discrete regularization on weighted graphs for image and manifold processing. Our work is distinct from them in that we are dealing with an interpolation problem and we add a preprocess to initialize the weight matrix. Specifically, we first work out a variational problem assuming weight matrix is a function of the image. Afterwards, we proceed the graph-based regularization with a constant weight matrix. Experiment results show that our approach improves the initial estimate considerably, through effectively reducing color bar and zipper in complex area.

The paper is organized as follows: Section 2 formulates the problem and the graph-based regularization method. Section 3 verifies the concept by numerical simulations followed with discussions. We then conclude in Section 4.

2. GRAPH-BASED REGULARIZATION

Suppose that we are given a $N_1 \times N_2$ image which is a subsampled version of a full color image according to a certain CFA pattern. For convenience, we present with a Bayer pattern [9]. Let f_r^0, f_g^0, f_b^0 be the stacking vectors of the true values of the three color components, and S_r, S_g, S_b be the sampling matrices in each channel (they are diagonal binary matrices with non-zero entries corresponding to the sampling locations), respectively. The goal is to restore the original image from those samples in red, green and blue channels, i.e. $S_c f_c^0, c = r, g, b$. In this section, we introduce a graph-based regularization framework to fulfill the task.

2.1. Regularization on a Weighted Graph

Graph model is a generic representation of data, including both images and data bases [7]. A *weighted graph* $G(V, E, w)$ consists of a set of vertices $V = \{v_1, \cdots, v_N\}$, a set of edges $E \subset V \times V$, and a weight function w defined on the edges. For an edge (u, v) connecting vertices u and v, the weight of it $w(u, v)$ quantifies the similarity between them. We use $u \sim v$ to denote two adjacent

vertices. Generally, the weight function satisfies the following properties: (1) $w(u,v) = w(v,u)$ for any $u \sim v$; (2) $w(u,v) \geq 0$ and $w(u,v) = 0$ if u is not adjacent to v in G. In addition, we define the degree $d(v)$ of a vertex v as $d(v) = \sum_u w(u,v)$. Then, the *graph Laplacian* matrix L is given by $L(u,v) = d(v)$ if $u = v$; $L(u,v) = -w(u,v)$ if $u \sim v$. Let $\mathcal{H}(V)$ be a Hilbert space defined on the vertices of G. A function $f : V \to \mathbb{R}$ of $\mathcal{H}(V)$ maps a real value $f(v)$ to each vertex $v \in V$. Note that f can also be viewed as a $|V| \times 1$ vector. Thus, from the spectral graph theory, we have $f^T L f = \sum_{u \sim v} \left(f(u) - f(v) \right)^2 \cdot w(u,v)$ for any function $f \in \mathcal{H}(V)$. This quantity reflects the variation of f on the graph.

Next, we consider the demosaicking problem. We view each pixel as a vertex and assign a weight to the edge linking pixels u, v as follows

$$w(u,v) = \exp\left(- \sum_{c=r,g,b} \alpha_c (f_c^0(u) - f_c^0(v))^2 / \varepsilon_1^2 \right) \times \exp\left(- ((u_x - v_x)^2 + (u_y - v_y)^2)/\varepsilon_2^2 \right), \quad (1)$$

where u_x, v_x, u_y, v_y are the horizontal and vertical coordinates of the pixels, $\alpha_r, \alpha_g, \alpha_b$ are the emphases on each channel and $\varepsilon_1, \varepsilon_2$ are two scale factors. This formula accounts both the difference of pixel intensities and locations, which was proposed as *bilateral filtering* in [6]. Since the components of natural images are highly correlated [4], the first term in (1) combines information from all the channels. Usually, the emphases are chosen according to the number of samples in the channels. For the Bayer pattern, we set α_g to be the largest, since there are more samples in the green channel. Ideally, $w(u,v)$ should be built from the true image. In reality, we interpolate a full color image with two approaches: Alternating Projections (AP) in [4] and Directional LMMSE (DL) in [5]. AP defines constraint sets on red and blue channels to force their high-frequency components to be close to that of the green channel; DL estimates the missing green samples adaptively in horizontal and vertical directions by the linear minimum mean square-error estimation (LMMSE). Then, we compute the weights based on the estimate. After that, the Laplacian matrix follows immediately.

Let N_c be the null spaces of S_c for $c = r, g, b$, respectively. By definition, every column of a null space is a member of a standard basis with the non-zero element corresponding to an unknown pixel. For the Bayer pattern, the size of N_g is $N \times \frac{N}{2}$; the size of N_r, N_b are both $N \times \frac{3N}{4}$, where $N = N_1 N_2$ is the number of pixels in the image. Thus, we could express each component of the block

$$f_c = S_c f_c^0 + N_c \beta_c, \quad (2)$$

where $\beta_c, c = r, g, b$ are column vectors formed by unknown R,G,B values accordingly. Since natural images are usually smooth, the regularization problem that we pose to perform demosaicking is

$$f_c = \min_{f_c} f_c^T L f_c, \quad (3)$$

subject to the constraint in (2). Replacing f_c with (2), we obtain an unconstrained quadratic optimization problem

$$\begin{aligned} \beta_c &= \min_{\beta_c} (S_c f_c^0 + N_c \beta_c)^T L (S_c f_c^0 + N_c \beta_c) \\ &= \min_{\beta_c} \beta_c^T N_c^T L N_c \beta_c + 2\beta_c^T N_c^T L S_c f_c^0 + C_0, \end{aligned} \quad (4)$$

where $C_0 = (f_c^0)^T S_c^T L S_c f_c^0$ is a constant that does not depend on β_c. Setting the derivative with respect to (w.r.t.) β_c to zero, we obtain

$$\beta_c = -(N_c^T L N_c)^{-1} N_c^T L S_c f_c^0. \quad (5)$$

The demosaicked version of each color component can be obtained after plugging (5) into (2).

2.2. Refinements of the Algorithm

There are possible refinements to the above framework. One is to regularize the differences $f_r - f_g$ and $f_b - f_g$ instead of implementing it on f_r and f_b directly. The reason is that these two quantities are smoother signals on the weighted graph. For the green channel, we perform the graph-based regularization as introduced in Section 2.1, which gives an output f_g. Then, for the red and blue channel, we consider the following minimization problem

$$f_c = \min_{f_c} (f_c - f_g)^T L (f_c - f_g), \quad (6)$$

subject to $f_c = S_c f_c^0 + N_c \beta_c$ for $c = r, b$. Using a similar approach, we find that the optimal coefficient β_c is

$$\beta_c = -(N_c^T L N_c)^{-1} N_c^T L (S_c f_c^0 - f_g), \quad (7)$$

and the solution to (6) follows immediately. The other improvement is to adapt the weights for different channels. The weight formula in (1) allows us to change $\alpha_r, \alpha_g, \alpha_b$ when we demosaick a specific color component. Usually, we let α_g be large and fix all the parameters. But in many cases, this adaptation ensures that we can enhance every color component. In addition, the matrix $N_c^T L N_c$ we want to inverse is sometimes ill-conditioned. To avoid unreliable solution, we skip the regularization when the condition number of it is greater than a large threshold.

2.3. Non-Static Laplacian

Previously, we assumed that the Laplacian matrix is static in the regularization formula. Namely, it behaves like a constant once determined from the estimate of the image. However, this estimated Laplacian is not exactly the same as that built from the true image (although they are close). Thus, to obtain

a better Laplacian matrix, we add the following optimization problem as a pre-process

$$f_c = \min_{f_c} f_c^T L(f_c) f_c, \;\; c = r, g, b \tag{8}$$

where $L(f_c)$ is a function of f_c corresponding to a same weight with (1) except that f_c^0 is replaced by f_c. The problem in (8) is non-quadratic. Hence, no close form solution is available. For simplicity, we derive the results only when $c = g$. Denote $Q = f_g^T L(f_g) f_g$, then we can write

$$Q = \sum_{u,v} C_1 e^{-\alpha_g (f_g(u) - f_g(v))^2 / \varepsilon_1^2} (f_g(u) - f_g(v))^2, \tag{9}$$

where C_1 is given by

$$\begin{aligned} C_1 = \exp \; & - \sum_{c=r,b} \alpha_c (f_c(u) - f_c(v))^2 / \varepsilon_1^2 \;\; \times \\ & \exp \; - ((u_x - v_x)^2 + (u_y - v_y)^2) / \varepsilon_2^2 \;\; . \end{aligned} \tag{10}$$

Since C_1 does not contain the green component, taking derivative of Q w.r.t. a particular *unknown* entry, we get

$$\begin{aligned} \frac{\partial Q}{\partial f_g(u)} = 2 \sum_v \; & C_1 e^{-(f_g(u) - f_g(v))^2 \frac{\alpha_g}{\varepsilon_1^2}} (f_g(u) - f_g(v)) \\ & \times \; 1 - \frac{\alpha_g}{\varepsilon_1^2} (f_g(u) - f_g(v))^2 \;\; . \end{aligned} \tag{11}$$

Similarly, we could write the derivative w.r.t. any unknown entry of the other two channels. Therefore, the solution to (8) is achieved iteratively by the gradient descent method

$$f_c^{update}(u) = f_c(u) - \gamma \cdot \frac{\partial Q}{\partial f_c(u)}, \;\; c = r, g, b \tag{12}$$

where γ is the searching step. After solving this non-quadratic problem, we calculate the Laplacian matrix which will initialize the graph-based regularization in Section 2.1.

3. EXPERIMENTAL RESULTS

We conduct the experiments on 24 color images of size 512×768 in the Kodak dataset [10], which are widely adopted in the literature [2] [4] [5]. For a particular image, first we sample it according to the Bayer pattern to mimic a mosaic one. Then we cut it into *blocks* by equally dividing its height and width to reduce dimensionality, and perform graph-based demosaicking upon each block. The outputs in each channel are stacked into a final estimate $f_c, c = r, g, b$ for the full image. Due to space limits, we present the results of four example images (indicated by indices in the dataset). The resulting image is evaluated in terms of the *color peak signal-to-noise ratio* (CPSNR) defined as

$$\text{CPSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{3} \sum_{c=r,g,b} \text{MSE}_c}, \tag{13}$$

with the mean square error MSE_c for each channel given by

$$\text{MSE}_c = \frac{1}{N_1 N_2} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \|f_c - f_c^0\|^2. \tag{14}$$

For a certain channel, we specify its PSNR via substituting the average MSE in (13) with the MSE of the particular channel.

As a first step, the initial estimate of the true image is obtained from either AP or DL. In both cases, we can easily adapt our framework to different CFA patterns just by generating different sampling matrices and the corresponding null spaces. Note that once they are generated, they can be used for all the image blocks.

3.1. Regularization after AP or DL

AP is a simple but effective demosaicking scheme, although it can hardly compete with the state-of-the-arts algorithms, such as DL used here. We demonstrate the strength of our framework based on these two initial estimates. At the first stage, we select the step γ in the gradient descent as 2×10^{-4} and the number of iterations as 3. At the second stage, the default parameters for building weight matrix are $\varepsilon_1 = \varepsilon_2 = 8$ and $\alpha_g = 4, \alpha_r = \alpha_b = 1$. We also apply the refinements introduced in Section 2.2. The experimental results of regularization after AP and DL are sorted in Table 3, where 'GR-AP' and 'GR-DL' stands for graph-based regularization after AP and DL, respectively. Normally, the block size is 16×24, while '*' indicates a larger size of 32×48. We list both the CPSNR of the full image and the PSNR of the channel which receives the largest improvement, as indicated in the rows named 'Channel'.

3.2. Discussions

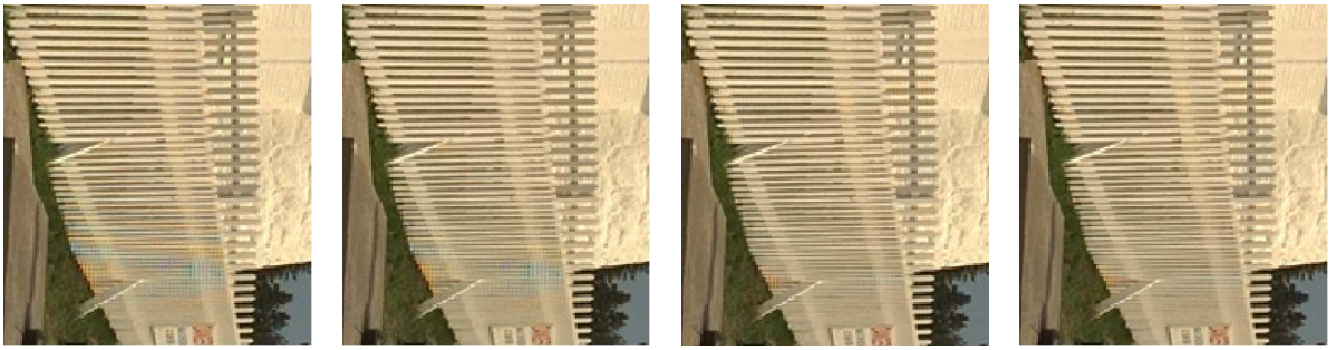
From the experiments, we can see that the graph-based regularization improves CPSNR upon both AP and DL. For different images or under different initial estimation algorithms, the most remarkable enhanced channels may be different. A larger block size helps to reach higher CPSNR values, while it increases the computational cost. The partition with 32×48 blocks is a good balance between these two aspects. As shown in Figure 1, our method effectively reduces the color bar in the lighthouse image (Image 19). We could also observe reduction of zipper effect from the hat image (Image 03), although we do not have space to present the results here.

4. CONCLUSION

We proposed a new graph-based regularization method for color image demosaicking. By viewing natural image as smooth signal on a weighted graph which measures the similarity between each pair of pixels, we aimed to search a full

Image No.	01		03		08		19	
Channel	Blue	CPSNR	Green	CPSNR	Blue	CPSNR	Red	CPSNR
AP	35.1623	36.2459	43.2929	38.9652	32.6050	34.4082	38.4929	39.3855
GR-AP	35.8081	36.5171	43.8211	39.1046	33.1002	34.6902	38.8590	39.5316
GR-AP*	36.0989	36.8047	43.9118	39.1788	33.8726	35.2119	38.8790	39.6256
Channel	Green	CPSNR	Green	CPSNR	Green	CPSNR	Green	CPSNR
DL	40.4101	38.6621	46.0436	42.9251	38.7402	36.2348	43.0665	41.0549
GR-DL	41.1608	38.8194	46.5178	43.0012	38.9973	36.2821	43.9918	41.2343
GR-DL*	41.9392	38.9615	47.2250	43.1030	39.3124	36.3372	45.2334	41.4309

Table 1. The summation of performance for graph-based regularization based on AP and DL.



(a) AP (b) GR-AP (c) DL (d) GR-DL

Fig. 1. Comparison between the initial estimates and the results of our scheme on a part of the *lighthouse* image. (a) Initial estimate of AP; (b) Regularization result based on AP; (c) Initial estimate of DL; (d) Regularization result based on DL;

color image that has a small variation and is consistent with the samples. This framework effectively improved our initial estimate, which was obtained from either AP or DL. We may achieve a better performance by choosing a larger block size, at the cost of computation. In addition, the mapping between image and signal on graph provides a general method for image processing. In the future, we are going to consider the task of joint demosaicking and denoising, observing that in reality the samples are always corrupted by noise. This would be a ready extension as long as we add another penalty term to the object function.

5. REFERENCES

[1] D. Menon, G. Calvagno, "Color image demosaicking: an overview," *Signal Processing: Image Commun.*, vol. 26, no. 8-9, pp. 518-533, Oct. 2011.

[2] _____, "Regularization approaches to demosaicking," *IEEE Trans. Image Proc.*, vol. 18, no. 10, pp. 2209-2220, Oct. 2009.

[3] A. Buades, B. Coll, J. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. Image Proc.*, vol. 18, no. 6, pp. 1192-1202, Jun. 2009.

[4] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plan interpolation using alternating projections," *IEEE Trans. Image Proc.*, vol. 11, no. 9, pp. 997-1013, Sept. 2002.

[5] L. Zhang, X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Proc.*, vol. 14, no. 12, pp. 2167 - 2178, Dec. 2005.

[6] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proc. of the 6th Int. Conf. on Computer Vision*, pp. 839, 1998.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373 - 1396, 2003.

[8] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Trans. Image Proc.*, vol. 17, no. 7, pp. 1047 - 1060, Jul. 2008.

[9] B. E. Bayer, "Color imaging array," U. S. Patent No. 3971065, Jul. 1976.

[10] URL: <`http://r0k.us/graphics/kodak/`>.