COLOR DEMOSAICING WITH CONTOUR STENCILS*

Pascal Getreuer
CMLA, ENS Cachan
61, av. du Président Wilson
94235 Cachan, France
Email: getreuer@gmail.com

***Abstract*—Demosaicing is the problem where, given a color image that has been subsampled by a color filter array, to interpolate the complete color information at each pixel. Many demosaicing methods adapt the interpolation according to estimated edge orientations. However, while the accuracy of these orientations is important for success, they are difficult to estimate from the mosaiced image. This paper extends contour stencils, a method for estimating image contours based on total variation along curves, to estimating contour orientations directly on mosaiced images. A method for demosaicing using these contour orientation estimates is proposed and compared with existing methods. The proposed demosaicing is performed as an energy minimization, using a graph regularization adapted according to the orientation estimates.**

***Index Terms*—demosaicing, total variation, orientation estimation**

I. Introduction

A mosaiced image is an image where the color channels have been subsampled such that only one color component is known at each pixel location. Most digital cameras capture mosaiced images, where incoming light is filtered with a Bayer color filter array (CFA) [1] to allow only red, green, or blue to pass through to each photosensor (see Fig. 1). To obtain full color information, the other color components must be interpolated from the neighboring pixels. This interpolation is called *demosaicing* (or demosaicking), which is also known as CFA interpolation or color reconstruction.

As proposed by Cok [2], many methods for demosaicing focus first on interpolating the green channel and then use the result to guide the interpolation of the red and blue channels. Alternatively, there are iterative methods which start from an initial interpolation and iteratively adjust all three channels [3]–[5].

A key challenge in demosaicing is that as a consequence of the mosaicing the channels become aliased. In order to overcome this, most methods impose (implicitly or explicitly) correlation between the color channels, for example, by ensuring that the differences $(r - g)$ and $(b - g)$ are smooth. Most methods additionally have some form of edge adaptivity to guide the interpolation. A survey of demosaicing methods is given in [6].

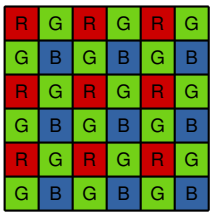

Fig. 1. The Bayer color filter array [1].

In this paper, we investigate particularly methods for inferring the local image geometry from the mosaiced data.

II. Orientation Estimates in Demosaicing

This section discusses existing methods for edge orientation estimation used in demosaicing. The red, green, and blue color intensities at pixel location (i, j) are denoted by $r_{i,j}$, $g_{i,j}$, $b_{i,j}$.

A. Laroche–Prescott

Laroche and Prescott [7] were among the first to use an edge-adaptive interpolation strategy. At a red pixel location (and similarly for blue), second-order central differences in the horizontal vertical directions:

$$\begin{aligned} D_h &= |r_{i-2,j} - 2r_{i,j} + r_{i+2,j}|\,, \\ D_v &= |r_{i,j-2} - 2r_{i,j} + r_{i,j+2}|\,. \end{aligned} \tag{1}$$

Depending on which magnitude is smaller, the green value $g_{i,j}$ is then interpolated either horizontally as $\frac{1}{2}(g_{i-1,j} + g_{i+1,j})$ or vertically as $\frac{1}{2}(g_{i,j-1} + g_{i,j+1})$.

B. Hibbard

Hibbard [8] proposed to use instead first-order central differences in the green channel,

$$\begin{aligned} D_h &= |g_{i-1,j} - g_{i+1,j}|\,, \\ D_v &= |g_{i,j-1} - g_{i,j+1}|\,, \end{aligned} \tag{2}$$

and $g_{i,j}$ is then interpolated as in Laroche–Prescott. Hibbard's method is better localized than Laroche–Prescott and is usually more accurate (see section IV-C).

*This material is based upon work supported by the National Science Foundation under Award No. DMS-1004694.
Source code for the presented method is available at http://www.ipol.im/pub/algo/g_demosaicking_with_contour_stencils.

C. Hamilton–Adams

Hamilton and Adams [9] added cross-channel corrections to Hibbard's method. At a red pixel location, the method computes the sum of (1) and (2)

$$\begin{aligned} D_h &= |g_{i-1,j} - g_{i+1,j}| + |r_{i-2,j} - 2r_{i,j} + r_{i+2,j}|\,, \\ D_v &= |g_{i,j-1} - g_{i,j+1}| + |r_{i,j-2} - 2r_{i,j} + r_{i,j+2}|\,. \end{aligned} \tag{3}$$

Interpolation is then performed using either the horizontal or vertical neighbors depending on which of these two magnitudes is smaller. If $D_h < D_v$, then $g_{i,j}$ is interpolated as

$$g_{i,j} = \tfrac{1}{2}(g_{i-1,j} + g_{i+1,j}) - \tfrac{1}{4}(r_{i-2,j} - 2r_{i,j} + r_{i+2,j}). \tag{4}$$

Hamilton–Adams is visually a significant improvement over Hibbard's method yet maintains low computational cost. For these reasons, many methods use Hamilton–Adams as an initialization and then apply additional steps to refine the result [3]–[5], [10], [11].

III. Contour Stencils

Contour stencils are a method for estimating image contours based on total variation (TV) along curves [12]–[15]. Given an image u and a smooth simple curve C, define the *total variation along* C as

$$\|u\|_{\mathrm{TV}(C)} := \sup_{(t_i)} \sum_i \left|u\big(\gamma(t_{i+1})\big) - u\big(\gamma(t_i)\big)\right|, \tag{5}$$

where $\gamma : [0,T] \to C$ parameterizes C and the supremum is over all finite (t_i), $0 = t_0 < t_1 < \cdots < t_N = T$. If u is differentiable,

$$\|u\|_{\mathrm{TV}(C)} = \int_0^T \left|\tfrac{\partial}{\partial t} u\big(\gamma(t)\big)\right| \, dt. \tag{6}$$

If C happens to coincide with an image contour, then $\|u\|_{\mathrm{TV}(C)} = 0$. In the other direction, a small total variation along C suggests that C is approximately a contour. The strategy is to estimate contours by identifying curves with small TV.

A. Discretization

Total variation along curves can be discretized as a sum of absolute differences. A *contour stencil* is a function $\mathcal{S} : \mathbb{Z}^2 \times \mathbb{Z}^2 \to \mathbb{R}$ describing weighted edges approximating the curve C, and the stencil TV is computed as

$$\|u\|_{\mathrm{TV}(C)} \approx \sum_{m,n \in \mathbb{Z}^2} \mathcal{S}(m,n)\,|u(m) - u(n)|\,. \tag{7}$$

It is useful to consider the TV estimate along translations of the same curve, $\|u\|_{\mathrm{TV}(C+k)}$. The stencil TV at pixel k is defined as

$$(\mathcal{S} \star [u])(k) := \sum_{m,n \in \mathbb{Z}^2} \mathcal{S}(m,n)\,|u(k+m) - u(k+n)|\,. \tag{8}$$

To make a local estimate of the image contours about k, the stencil yielding the smallest TV is identified from among a set of candidate stencils,

$$\mathcal{S}^\star(k) = \arg\min_{\mathcal{S} \in \Sigma} (\mathcal{S} \star [u])(k). \tag{9}$$

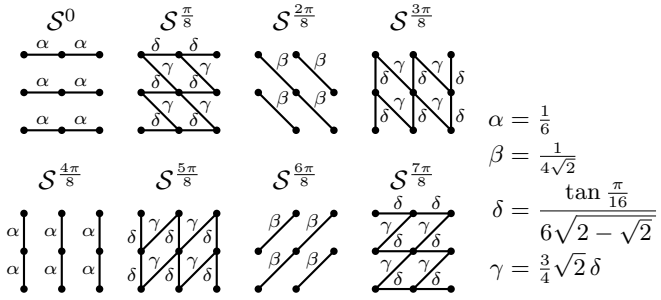

Fig. 2. An example stencil set of line-shaped stencils, where the edge weights $\mathcal{S}(m,n)$ are denoted by superscripts α, β, δ, γ. The stencils distinguish 8 orientations.

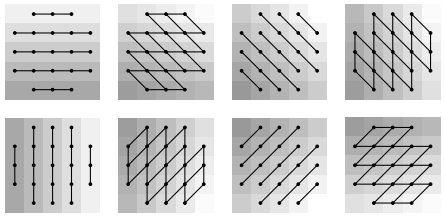

Fig. 3. A contour stencil set designed by minimizing (13). The features are shown as images with the corresponding stencil superimposed where an edge is drawn if $\mathcal{S}^j(m,n) = 1$.

Stencil $\mathcal{S}^\star(k)$ is called the *best-fitting stencil*, and it provides a model of the underlying contours. Fig. 2 shows an example stencil set Σ.

B. Stencil Design

The candidate stencils Σ can be chosen from the point of view of distinguishing different features. Let $f^1, \ldots, f^J$, $f^j : \mathbb{R}^2 \to \mathbb{R}$, be a set of features, for example linear functions with different orientations,

$$f^j(x) = x_1 \sin \tfrac{1}{J}\pi j - x_2 \cos \tfrac{1}{J}\pi j. \tag{10}$$

The goal is to design stencils $\Sigma = \{\mathcal{S}^1, \ldots, \mathcal{S}^J\}$ such that $\mathcal{S}^j$ is the best-fitting stencil on feature f^j. That is, the stencils should satisfy

$$(\mathcal{S}^j \star [f^j])(0) < (\mathcal{S}^i \star [f^j])(0) \quad \forall i \neq j \tag{11}$$

or equivalently

$$\big((\mathcal{S}^j - \mathcal{S}^i) \star [f^j]\big)(0) < 0 \quad \forall i \neq j. \tag{12}$$

A simple method to approximate condition (12) is to design the stencils as

$$\begin{cases} \displaystyle\arg\min_{\mathcal{S}^1,\ldots,\mathcal{S}^J} \sum_{i=1}^J \sum_{j=1}^J \big((\mathcal{S}^j - \mathcal{S}^i) \star [f^j]\big)(0) \\ \text{subject to } \; 0 \le \mathcal{S}^j(m,n) \le 1 \quad (m,n) \in \mathcal{E}, \\ \qquad\qquad \mathcal{S}^j(m,n) = 0 \qquad \text{otherwise}, \end{cases} \tag{13}$$

where $\mathcal{E}$ is an allowed set of edges,

$$\begin{aligned} \mathcal{E} = \big\{ &(m,n) \in \mathbb{Z}^2 \times \mathbb{Z}^2 : \\ &\|m\| \le R, \|n\| \le R, m \in \mathcal{N}(n) \big\} \end{aligned} \tag{14}$$

and $\mathcal{N}(n)$ are the eight adjacent neighbors of n. These constraints localize the stencils to a disk of radius R and only allow edges between neighboring pixels. The minimization has a closed-form solution

$$\mathcal{S}^j(m,n) = \begin{cases} 1 & \text{if } (m,n) \in \mathcal{E} \text{ and } |f^j(m) - f^j(n)| \\ & \quad < \frac{1}{J}\sum_{i=1}^{J} |f^i(m) - f^i(n)|, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Since the minimization does not guarantee that $\big((\mathcal{S}^j - \mathcal{S}^i) \star [f^j]\big)(0) < 0$, the solution should be verified. Fig. 3 shows an example using the line shaped features (10). The features need not be linear, for example (13) can also design corner-shaped contour stencils.

The stencil design (13) is effective for selecting the support of the stencil edges, but the binary-valued edge weights it produces are too crude to use directly. The weights must be normalized so that they are compared fairly in (9). For the stencil set in Fig. 2, the weights were selected to optimize the approximation of rotational invariance [14].

IV. Mosaiced Contour Stencils

For demosaicing, we would like to apply contour stencils directly on the mosaiced image data. This is possible by observing a key difference: absolute difference is meaningful only between samples of the same color channel.

Let f be the observed mosaiced data represented as a flat 2D array,

$$f_{i,j} = \begin{cases} r_{i,j} & \text{if } i \text{ and } j \text{ are both even,} \\ b_{i,j} & \text{if } i \text{ and } j \text{ are both odd,} \\ g_{i,j} & \text{otherwise.} \end{cases} \tag{16}$$

We consider a variation estimate of the form

$$(\mathcal{S} \star [f]) := \sum_{m,n \in \mathbb{Z}^2} \mathcal{S}(m,n)\,|f_m - f_n|\,. \tag{17}$$

A. Stencil Design

Stencils should have nonzero edges $\mathcal{S}(m,n) \neq 0$ only where m and n are nearby locations in the same channel. We impose this requirement by adjusting the notion of neighbors $\mathcal{N}(n)$ in (14). For the Bayer CFA, the neighbors depend on whether n is a green pixel location:

$$\begin{aligned} \tilde{\mathcal{N}}(n) &= \{m : \|m-n\| = \sqrt{2} \text{ or } 2\} \quad \text{if } n \in \textit{green}, \\ \tilde{\mathcal{N}}(n) &= \{m : \|m-n\| = 2 \text{ or } 2\sqrt{2}\} \quad \text{if } n \notin \textit{green}. \end{aligned} \tag{18}$$

Fig. 4 shows stencils designed by (13) for eight oriented linear features (10) on the Bayer CFA. The design is restricted to a disk-shaped support of radius $\sqrt{5}$.

Another peculiarity is that mosaicing is not shift-invariant, but for periodic CFAs it is invariant under shifts by multiples of the period. There are two distinct cases for contour stencils on the Bayer CFA: one where the center pixel is green and another where the center pixel is red or blue. The top row of Fig. 4 shows stencils centered on green pixels in a blue row; for green pixels in red rows, the roles of red and blue are exchanged.

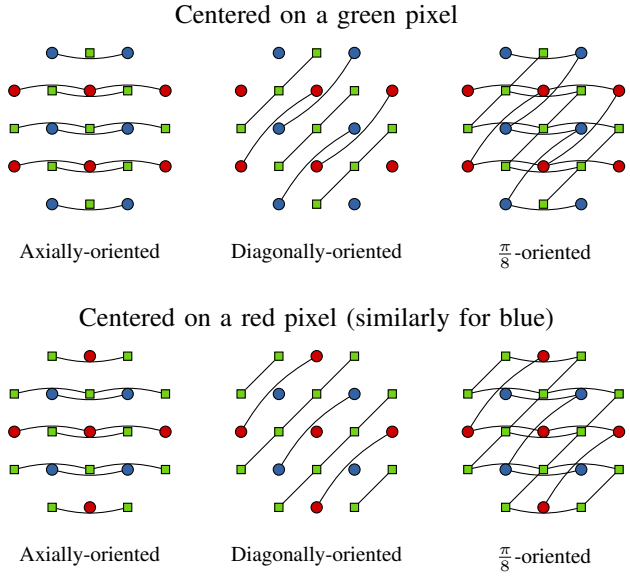

Fig. 4. Contour stencils designed by (13) for Bayer mosaiced data. Similarly by rotations of these stencils, 8 different orientations are distinguished.

B. Stencil Weighting

The stencils need to be reweighted so that they are compared fairly in the stencil selection (9). We denote an unnormalized stencil by tilde $\tilde{\mathcal{S}}$. We define $f(x) = x_1 \sin\theta - x_2 \cos\theta$ and normalize such that

$$\big|\theta - \tfrac{\pi}{8}j\big| < \tfrac{\pi}{16} \quad \Longrightarrow \quad \mathcal{S}^{\frac{\pi}{8}j} = \operatorname*{arg\,min}_{\mathcal{S} \in \Sigma} (\mathcal{S} \star [f]). \tag{19}$$

For the unnormalized axial and diagonal stencils,

$$(\tilde{\mathcal{S}}^0 \star [f]) = 22\,|\sin\theta|\,, \tag{20a}$$
$$(\tilde{\mathcal{S}}^{\frac{\pi}{4}} \star [f]) = 14\sqrt{2}\,\big|\sin(\theta - \tfrac{\pi}{4})\big|\,, \tag{20b}$$
$$(\tilde{\mathcal{S}}^{\frac{\pi}{2}} \star [f]) = 22\,\big|\sin(\theta - \tfrac{\pi}{2})\big|\,, \tag{20c}$$
$$(\tilde{\mathcal{S}}^{\frac{3\pi}{4}} \star [f]) = 14\sqrt{2}\,\big|\sin(\theta - \tfrac{3\pi}{4})\big|\,. \tag{20d}$$

For these stencils, $(\tilde{\mathcal{S}}^\alpha \star [f])$ is symmetric in θ about the point $\theta = \alpha$. However, for the stencils $\tilde{\mathcal{S}}^{\frac{\pi}{8}j}$ for odd j, the unnormalized stencils are not symmetric about $\frac{\pi}{8}j$. To obtain symmetry, we set $\tilde{\mathcal{S}}^{\frac{\pi}{8}} = \tilde{\mathcal{S}}^0 + \mu\tilde{\mathcal{S}}^{\frac{\pi}{4}}$ with

$$\mu = \frac{22}{14(1 + \tan\frac{\pi}{8})} \tag{21}$$

such that $(\tilde{\mathcal{S}}^{\frac{\pi}{8}} \star [f])$ is proportional to $\cos(\theta - \frac{\pi}{8})$,

The axial and diagonal stencils are normalized by their weighted arc length,

$$\begin{aligned} \mathcal{S}^\alpha &= \tilde{\mathcal{S}}^\alpha / |\tilde{\mathcal{S}}^\alpha|, \quad \alpha = 0, \tfrac{\pi}{4}, \tfrac{\pi}{2}, \tfrac{3\pi}{4}, \\ |\tilde{\mathcal{S}}| &:= \sum_{m,n \in \mathbb{Z}^2} \tilde{\mathcal{S}}(m,n)\,\|m-n\|\,. \end{aligned} \tag{22}$$

For $\tilde{\mathcal{S}}^{\frac{\pi}{8}}$, we normalize as $\mathcal{S}^{\frac{\pi}{8}} = \tilde{\mathcal{S}}^{\frac{\pi}{8}}/w$ with

$$w = 22\big(1 + \tfrac{1}{\sqrt{2}}(\cot\tfrac{\pi}{16} - 1)\big) \tag{23}$$

such that $(\mathcal{S}^0 \star [f]) = (\mathcal{S}^{\frac{\pi}{8}} \star [f])$ when $\theta = \frac{\pi}{16}$. With these normalized stencil weights, the stencil set satisfies the stencil

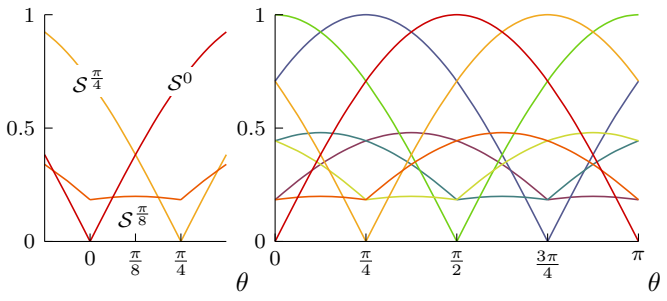

Fig. 5. Normalized stencil total variations ($\mathcal{S}^{\frac{\pi}{8}j} \star [f]$) vs. θ. *Left:* The total variations of $\mathcal{S}^{0}$, $\mathcal{S}^{\frac{\pi}{8}}$, and $\mathcal{S}^{\frac{\pi}{4}}$. *Right:* All eight stencils.

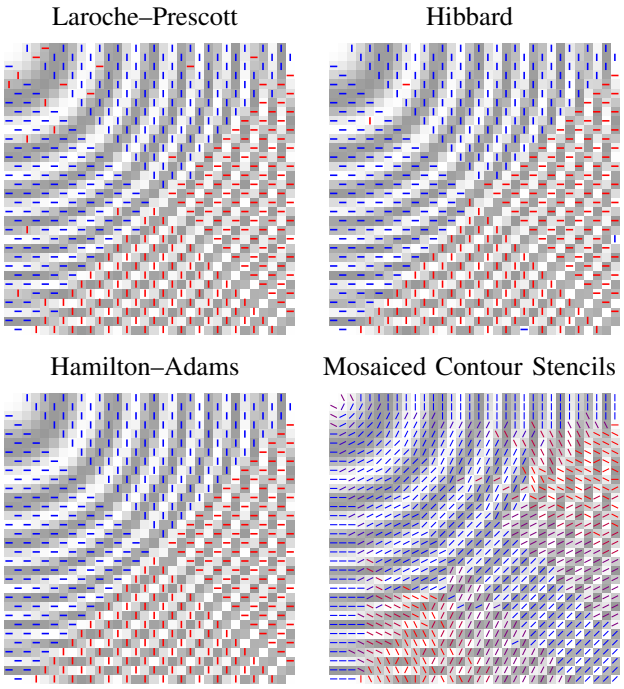

Fig. 6. Comparison of orientations estimated by Laroche–Prescott, Hibbard, Hamilton–Adams, and mosaiced contour stencils.

TABLE I
ORIENTATION ESTIMATE ACCURACIES FOR FIG. 6

Method	% Accurate	Average Error
Laroche–Prescott	53.33%	0.2090π
Hibbard	55.83%	0.1995π
Hamilton–Adams	56.25%	0.1990π
Mosaiced Contour Stencils	88.27%	0.1167π

design condition (11) for linear features at eight orientations and approximates rotational invariance (19). Fig. 5 shows the normalized stencil total variations ($\mathcal{S}^{\frac{\pi}{8}j} \star [f]$) vs. θ.

C. Comparison

We compare the proposed mosaiced contour stencils with the orientation estimation methods of Laroche and Prescott [7], Hibbard [8], and Hamilton and Adams [9].

A grayscale synthetic image is mosaiced with the Bayer CFA as in Fig. 1. Each method is applied to the mosaiced image to compute estimated orientations $\theta_{i,j}$. Fig. 6 shows the estimated orientations superimposed on the full unmosaiced image. To avoid boundary effects, a margin of 4 pixels is removed. The lines are colored to visualize accuracy, with blue indicating correct detections and red indicating errors. For the contour stencils, shades of purple depict the error magnitude.

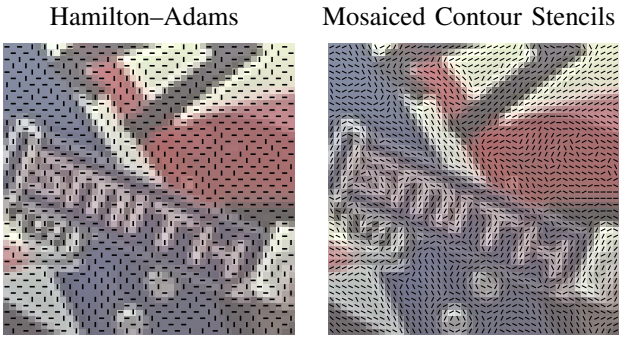

Fig. 7. Orientations estimated by Hamilton–Adams and mosaiced contour stencils on a crop from image 5 of the Kodak Image Suite.

The Laroche–Prescott, Hibbard, and Hamilton–Adams methods only define orientations at red and blue pixel locations. Additionally, the orientation is considered undefined if the detection is ambiguous (e.g., $D_h = D_v$).

Denoting by $\mathcal{D}$ the set of pixels where orientation estimates are defined, Table I compares the results quantitatively with

$$\%\ \text{Accurate} := 100\frac{\#\left\{(i,j) \in \mathcal{D} : e_{i,j} < \frac{\pi}{4}\right\}}{\#(\mathcal{D})}, \tag{24}$$

$$\text{Average Error} := \frac{1}{\#(\mathcal{D})} \sum_{(i,j)\in\mathcal{D}} e_{i,j}, \tag{25}$$

where $\#(\cdot)$ denotes cardinality and $e_{i,j} = |\theta_{i,j} - \theta_{i,j}^{\text{exact}}|$ is the angular distance from the the exact orientation at (i,j).

Fig. 7 compares Hamilton–Adams and mosaiced contour stencils on natural images. The methods are applied to a mosaiced crop of image 5 from the Kodak Image Suite [16]. The orientations estimated by the mosaiced contour stencils follow to the underlying image structure much more accurately.

These comparisons show that the mosaiced contour stencils estimates have finer angular precision, allowing more accurate approximation of curved features and edges with non-axial orientation. Additionally, the contour stencils estimates are defined at every pixel location, providing an orientation map with finer spatial resolution.

V. DEMOSAICING

A demosaicing method can be designed from two components: a method for detecting image structure and a method for interpolation according to the detected structure. These components are modular as any detection method can be used to guide any interpolation method that understands the structure information. To show the potential of mosaiced contour stencils for the detection, this section develops a simple interpolation method that is guided by the stencils.

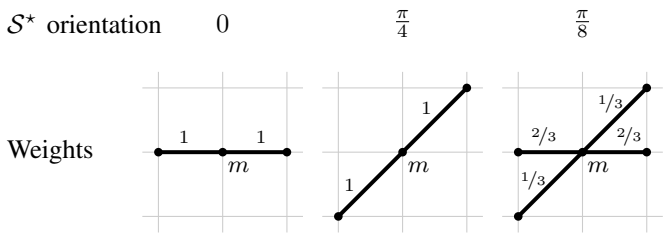

Fig. 8. The weights $w^{\text{init}}_{m,n}$ are defined according to the best-fitting mosaiced contour stencil $\mathcal{S}^\star(m)$.

A. Energy Minimization Model

Let f be the input mosaiced image and let Ω^k denote the pixel locations where the kth channel is known, $k \in \{R, G, B\}$. Similar to the interpolation method proposed in [13], we consider demosaicing using a graph regularization [17],

$$\begin{cases} \arg\min\limits_{u} \sum\limits_{m} \Big(\sum\limits_{n} \big(w_{m,n} \|u_m - u_n\|_L \big)^2 \Big)^{1/2} \\ \qquad + \alpha \sum\limits_{m} \Big(\sum\limits_{n} \big(w_{m,n} \|u_m - u_n\|_C \big)^2 \Big)^{1/2} \\ \text{subject to } u^k_n = f_n,\ n \in \Omega^k, k \in \{R, G, B\}, \end{cases} \tag{26}$$

where $\|\cdot\|_Y$ and $\|\cdot\|_C$ are seminorms in the color space suggested by Condat [18],

$$\|x\|_L := \tfrac{1}{\sqrt{3}} |x^R + x^G + x^B|, \tag{27}$$

$$\|x\|_C := \sqrt{\tfrac{1}{2}(x^R - x^B)^2 + \tfrac{1}{6}(x^R - 2x^G + x^B)^2}. \tag{28}$$

The first term of (26) regularizes the luminance while the second term regularizes chrominance. The parameter α on the second term balances between luminance and chrominance regularization. A larger value of α reduces color artifacts but increases zipper artifacts.

For each pixel m, graph weights $w^{\text{init}}_{m,n}$ between m and its adjacent neighbors n are first defined according to the best-fitting mosaiced contour stencil as shown in Fig. 8 and zero for all other n. To ensure that the graph is connected, a small value ϵ is added to all $w^{\text{init}}_{m,n}$ where m and n are axial or diagonal neighbors,

$$w^{\text{reg}}_{m,n} := w^{\text{init}}_{m,n} + \epsilon \mathbb{1}_{\{m \in \mathcal{N}(n)\}}. \tag{29}$$

Finally, the weights are symmetrized and spatially filtered,

$$w_{m,n} := \sum_{m'} (w^{\text{reg}}_{m',n} + w^{\text{reg}}_{n,m'}) \exp\big(-\tfrac{\|m-m'\|^2}{2\sigma^2}\big). \tag{30}$$

B. Results

This section shows demosaicing results with the energy minimization (26) using the parameters $\alpha = 1.7$, $\epsilon = 0.2$, $\sigma = 0.5$. The minimization is solved by the split Bregman method [19].

All results were produced by the following procedure. We simulate a mosaic by subsampling a color image with a red location in the upper-left corner as in Fig. 1. To avoid boundary effects, the image borders of the mosaic are padded with 16 pixels using whole-sample symmetric extension.

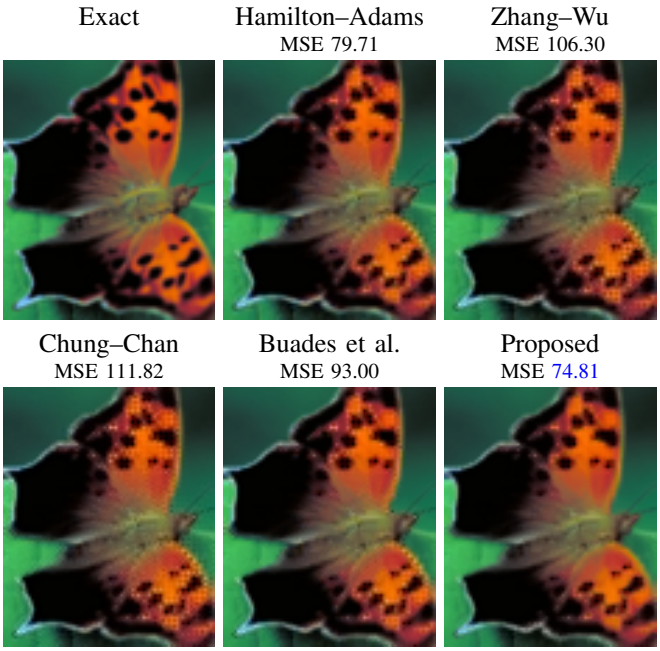

Fig. 9. A test with strong color discontinuities. *Photograph by Thomas G. Barnes, U.S. Fish and Wildlife Service.*

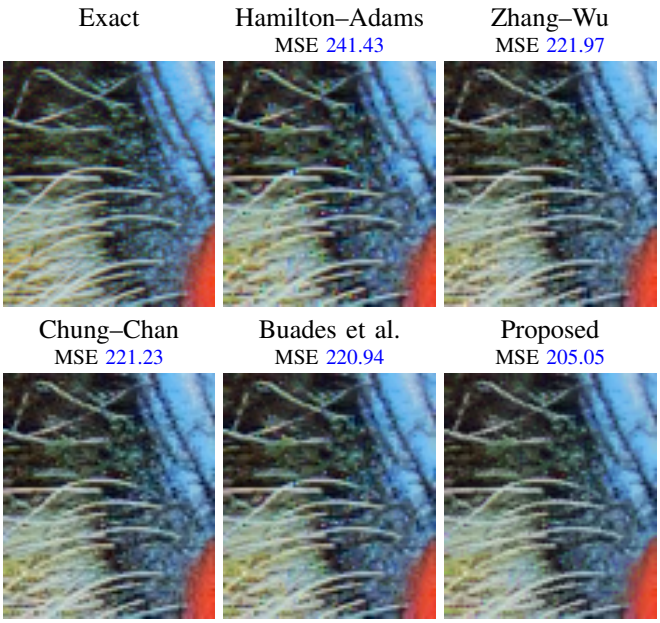

Fig. 10. A test with texture: comparison on a crop of the mandrill image.

We compare with bilinear, Hamilton and Adams [9], Gunturk et al. [3], Zhang and Wu [11], Chung and Chan [20], Lian et al. [21], and Buades et al. [5] using the default parameters suggested in the papers. After removing the 16-pixel padding, the mean squared error (MSE) is computed in RGB coordinates relative to the intensity range $[0, 255]$.

Fig. 9 tests the methods on an image with strong color discontinuities and curving boundaries. Most methods produce

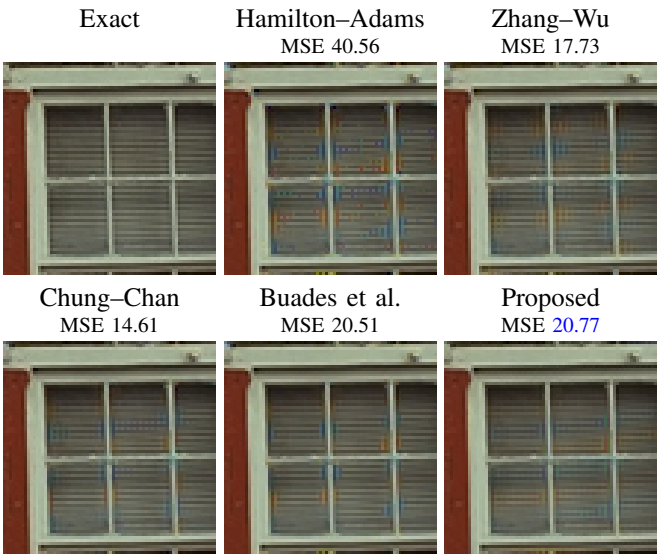

Fig. 11. A test with aliasing: comparison on a crop of image 1 from the Kodak Image Suite.

TABLE II
AVERAGE MSE ON THE KODAK AND IMAX IMAGES.

Method	Kodak	IMAX
Bilinear	83.70	45.71
Hamilton–Adams [9]	16.23	35.37
Gunturk et al. [3]	9.42	60.85
Zhang–Wu [11]	7.66	38.88
Chung–Chan [20]	7.75	41.90
Lian et al. [21]	8.70	36.85
Buades et al. [5]	12.00	36.00
Proposed	28.36	30.09

significant zipper artifacts.

Fig. 10 compares how the methods handle texture with a crop from the mandrill image. The texture and intricate geometry of the whiskers make this test very challenging.

Fig. 11 tests how the methods handle aliasing. The test image contains a pattern approaching the Nyquist limit so that mosaicing severely aliases the red and blue channels. The proposed method works reasonably well for such images with limited color and zipper artifacts.

Table II shows the MSE averages over the Kodak Image Suite [16] and the IMAX images introduced in [6]. The Kodak and IMAX images are quite different: the IMAX images are statistically sharper and more colorful [5]. The table shows that while the proposed method is relatively weak on the Kodak images, it has the best performance on the IMAX images.

VI. CONCLUSION

We have developed a new method for estimating edge orientations on mosaiced image data using contour stencils. The contour stencil method is significantly more reliable and provides greater angular resolution than existing methods. We applied the contour stencil orientations in designing an effective edge-adaptive demosaicing method.

ACKNOWLEDGMENT

I would like to thank Prof. J.-M. Morel for useful conversations on this work.

REFERENCES

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3,971,065, 1976.
[2] 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.
[3] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.
[4] X. Li, "Demosaicing by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
[5] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self similarity driven demosaicking," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1192–1202, 2009.
[6] X. Li, B. K. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," in *Proc. IS&T/SPIE Conf. on Visual Comm. and Image Proc.*, vol. 6822, 2008.
[7] 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.
[8] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," U.S. Patent 5,382,976, 1995.
[9] J. F. Hamilton, Jr. and J. E. Adams, Jr., "Adaptive color plan interpolation in single sensor color electronic camera," U.S. Patent 5,629,734, 1997.
[10] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.
[11] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, 2005.
[12] P. Getreuer, "Contour stencils for edge-adaptive image interpolation," in *Proc. SPIE*, vol. 7257, 2009.
[13] ——, "Image zooming with contour stencils," in *Proc. SPIE*, vol. 7246, 2009.
[14] ——, "Image interpolation with contour stencils," *Image Processing On Line*, 2011.
[15] ——, "Contour stencils: Total variation along curves for adapted image interpolation," 2011.
[16] "Kodak true color image collection," http://r0k.us/graphics/kodak/.
[17] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.
[18] L. Condat, "A new color lter array with optimal sensing properties," in *Proc. IEEE ICIP*, 2009.
[19] T. Goldstein and S. J. Osher, "The split bregman method for l1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
[20] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *Image Processing, IEEE Transactions on*, vol. 15, no. 10, pp. 2944–2955, 2006.
[21] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *Image Processing, IEEE Transactions on*, vol. 16, no. 10, pp. 2515–2525, 2007.