On High-Order Denoising Models and Fast Algorithms for Vector-Valued Images

Carlos Brito-Loeza and Ke Chen

***Abstract*—Variational techniques for gray-scale image denoising have been deeply investigated for many years; however, little research has been done for the vector-valued denoising case and the very few existent works are all based on total-variation regularization. It is known that total-variation models for denoising gray-scaled images suffer from staircasing effect and there is no reason to suggest this effect is not transported into the vector-valued models. High-order models, on the contrary, do not present staircasing. In this paper, we introduce three high-order and curvature-based denoising models for vector-valued images. Their properties are analyzed and a fast multigrid algorithm for the numerical solution is provided. *AMS subject classifications*: 68U10, 65F10, 65K10.**

***Index Terms*—Fourth-order partial differential equations (PDEs), image denoising, multilevel methods, regularization, variational models.**

I. Introduction

DENOISING of gray-scale images has been extensively studied and investigated within the last decades. With the appearance of the total variation (TV) model of Rudin, Osher, and Fatemi [23], it became evident that variational approaches to the image denoising problem can yield often excellent results. However, for images with smooth features and nonpiecewise constant intensities, the TV model may produce images appearing blocky with the so called *staircase effect*. There exist several ways of improving the model by a different regularizer (see [20], [27] and [10] and references therein) and one effective approach is to use high-order models of partial differential equations (PDEs) [34], [7], [19], [18], [10], [35]. This paper will address such types of high-order models for vector valued images (including color images with three channels) and the associated fast algorithms.

In contrast to the extensive studies of the TV model to gray-scale images, extension models to color or vector-valued images have been less thoroughly investigated although several interesting works about this subject do exist [25], [2], [4], [28]. The most natural method of extension is the channel by channel (TV1-CbC) approach which implies using the TV model independently for each channel. It is not hard to find examples (where channels are quite different) when such a simple extension is inadequate. Thus taking into consideration the differences in channels, coupled models that we shall review are found to deliver better results than the simple TV1-CbC approach in a number of situations.

By combining the above mentioned two ideas: high-order regularization plus channel coupling, it would be ideal to have a high-order color denoising model with coupling among channels. Surprisingly and up to our knowledge, there is however no published work of such a kind of vector-valued high-order model. Of course, we expect such a model to deliver better results than its counterpart (the high-order CbC approach).

In this paper, we take the high-order and curvature-based denoising model of Zhu-Chan [35] for gray-scale images as our starting point and introduce two different ways to generalize it to vector-valued images obtaining respectively what we call *global* and *local* coupling among the channels. We will analyze the properties of these two new models and will show some examples suggesting that our *global high-order model* is not only better than its correspondent CbC high-order competitor but also better than TV based color models. Finally, we will show how to implement a fast multigrid algorithm for this recommended model.

This paper is organized as follows. First in Section II, we state the variational denoising problem for vector-valued images, presenting four TV-based models for vector-valued image denoising. Then in Section III, our two new high-order models (CU2-LCM and CU3-GCM) are introduced. In Section IV, the numerical discretization of these high-order models is presented followed by the numerical algorithms with emphasis on a multigrid algorithm for the recommended CU3-GCM model. Finally, numerical experiments Section V and conclusions Section VI will be presented.

II. Problem Formulation and Current Models

Define a vector-valued image as a function $\boldsymbol{u} = \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ i.e., $\boldsymbol{u} = (u_1, \ldots, u_m)$ with $u_\ell = u_\ell(x_1, \ldots, x_n) \quad \forall\, \ell = 1, \ldots, m$. A noisy image $\boldsymbol{u^0}$ is obtained by adding up Gaussian noise $\boldsymbol{\eta}$ to $\boldsymbol{u}$, i.e., $\boldsymbol{u^0} = \boldsymbol{u} + \boldsymbol{\eta}$. The variational approach to remove $\boldsymbol{\eta}$ from $\boldsymbol{u^0}$ is then

$$\min_{\boldsymbol{u}} \left\{ \alpha R(\boldsymbol{u}) + \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{u^0}\|_2^2 \right\} \tag{1}$$

where $\|\boldsymbol{u} - \boldsymbol{u^0}\|_2^2 = \sum_{\ell=1}^m \int_\Omega (u_\ell - u_\ell^0)^2 \; \boldsymbol{dx}, \boldsymbol{dx} = (dx_1, \ldots, dx_n)$ and $R(\boldsymbol{u})$ is a regularization term selecting the space of functions were $\boldsymbol{u}$ will belong.

In Fig. 1, we show an example of a synthetic color image and the way each one of its channels looks in an one-dimensional

Manuscript received August 17, 2009; revised December 21, 2009. First published February 17, 2010; current version published May 14, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Maya R. Gupta.

The authors are with the Centre for Mathematical Imaging Techniques (CMIT), Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, U.K. (e-mail: k.chen@liv.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2042655

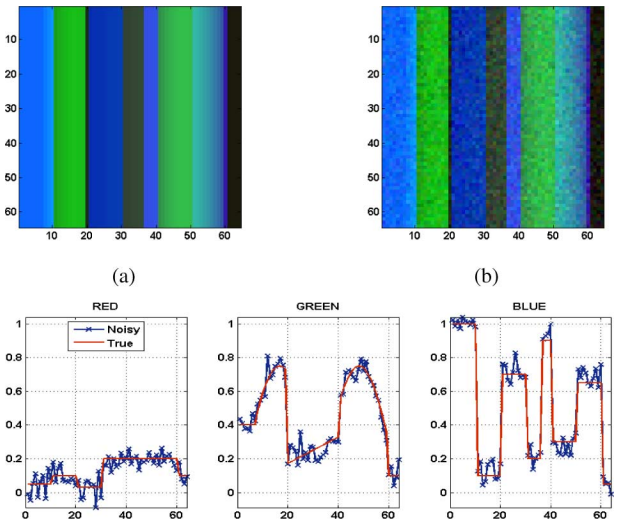


Fig. 1. Model problem, an image with a weak piecewise constant red channel, a piecewise smooth green channel, and a strong piecewise constant blue channel.

space. We will use this image to test the models we review here. We now proceed to present in historical order three different ways to generalize the total variation model of Rudin, Osher, and Fatemi [23] from gray-scale to vector-valued images. This will prove to be helpful at the moment of introducing our high-order color models.

In the widely used RGB image color model there are three channels (red, green, and blue) and therefore $n = 2$ and $m = 3$. Then we note that $\boldsymbol{u}(x_1, x_2) : \mathbb{R}^2 \to \mathbb{R}^3$ be an image with three channels $u_\ell(x_1, x_2) : \mathbb{R}^2 \to \mathbb{R}$, $\ell = 1, 2, 3$.

A. The Total Variation Model With Channel-by-Channel Coupling (TV1-CbC)

As commented, the simplest regularizer with channel-by-channel coupling for $R(\boldsymbol{u})$ is the following:

$$R(\boldsymbol{u}) = \sum_{\ell=1}^{m} \int_\Omega |\nabla u_\ell| \, dx_1 dx_2 \tag{2}$$

with which the resulting Euler–Lagrange equation of (1) is given by

$$-\alpha \nabla \cdot \frac{\nabla u_\ell}{|\nabla u_\ell|} + u_\ell - u_\ell^0 = 0 \quad \text{in } \Omega, \ \ell = 1, \ldots, m \tag{3}$$

where $|\nabla u_\ell| = \sqrt{((\partial u_\ell)/(\partial x_1))^2 + ((\partial u_\ell)/(\partial x_2))^2}$ and Neumann boundary conditions are applied to each channel.

Unless all u_ℓ are similar to each other, the TV1-CbC method is not robust. For instance, in Fig. 2 the regularization parameter α was selected to obtain the best restoration for the two strong channels and this caused the small step in the weak channel to be almost completely wiped out. Ideally one would like to use the regularizer of the type

$$R(\boldsymbol{u}) = \sum_{\ell=1}^{m} \alpha_\ell \int_\Omega |\nabla u_\ell| \, dx_1 dx_2 \tag{4}$$

with suitable α_ℓ adjusting to the channel variations. In fact the models reviewed next attempt to do this in an automatic way.

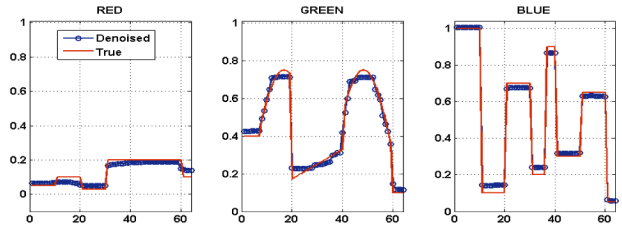


Fig. 2. Result from the TV1-CbC model.

B. Total Variation Model of Bresson and Chan (TV2-BRC)

The Bresson and Chan model [4] formally generalizes the TV semi-norm to the case of vector-valued images before using it for $R(\boldsymbol{u})$ in (1). For a given vector-valued function $\boldsymbol{u} : \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ the vectorial total variation norm is denoted the finite positive measure

$$\int_\Omega |D\boldsymbol{u}| := \sup_{\boldsymbol{p} \in P} \left\{ \int_\Omega \langle \boldsymbol{u}, \nabla \cdot \boldsymbol{p} \rangle \, \boldsymbol{d}x \right\} \tag{5}$$

where $\boldsymbol{p} := (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m) : \Omega \to \mathbb{R}^{m \times n}$, $\boldsymbol{p}_\ell := (p_\ell^{x_1}, \ldots, p_\ell^{x_n}) : \Omega \to \mathbb{R}^n$, $\forall \ell \in [1, m]$, $\nabla \cdot$ is the divergence operator such that $\nabla \cdot \boldsymbol{q} := (\nabla \cdot \boldsymbol{q}_1, \ldots, \nabla \cdot \boldsymbol{q}_m) : \Omega \to \mathbb{R}^m$, $\forall \boldsymbol{q} : \Omega \to \mathbb{R}^{m \times n}$, $\nabla \cdot \boldsymbol{q}_\ell := \sum_{j=1}^{n} \partial_{x_j} q_\ell^{x_j} : \Omega \to \mathbb{R}$, $\forall \ell \in [1, m]$, the product $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product defined as $\langle \boldsymbol{v}, \boldsymbol{w} \rangle := \sum_{\ell=1}^{m} \langle v_\ell, w_\ell \rangle$, $\forall (\boldsymbol{v}, \boldsymbol{w}) \in (R^m)^2$, which implies that $\langle \boldsymbol{u}, \nabla \cdot \boldsymbol{p} \rangle = \sum_{\ell=1}^{m} \langle u_\ell, \nabla \cdot \boldsymbol{p}_\ell \rangle$ and the L^2 Euclidean norm $|\cdot|$ is naturally defined by $|\boldsymbol{v}| := \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle} = \sqrt{\sum_{\ell=1}^{s} v_\ell^2}$, $\forall \boldsymbol{v} \in \mathbb{R}^s$.

Depending on the set P of functions of the dual variable $\boldsymbol{p}$, the vectorial total variation semi-norm (VTV) can be defined in different ways; Bresson and Chan [4] considered two cases: $P_1 = \{\boldsymbol{p} \in C_c^1(\Omega; R^{m \times n}) : |\boldsymbol{p}|_\infty \leq 1\}$ and $P_2 = \{\boldsymbol{p} \in C_c^1(\Omega; R^{m \times n}) : |\boldsymbol{p}|_2 \leq 1\}$ where $|\boldsymbol{p}|_\infty = \max_{\ell=1,\ldots,m} |\boldsymbol{p}_\ell|$ and $|\boldsymbol{p}|_2 = \sqrt{\sum_{\ell=1}^{m} \langle \boldsymbol{p}_\ell, \boldsymbol{p}_\ell \rangle} = \sqrt{\sum_{\ell=1}^{m} \sum_{j=1}^{n} (p_\ell^{x_j})^2}$. Further setting $P = P_1$ [4] leads to the vectorial TV semi-norm

$$\int_\Omega |D\boldsymbol{u}| = \sum_{\ell=1}^{m} \int_\Omega |\nabla u_\ell| \, dx_1 dx_2 = \sum_{\ell=1}^{m} \mathrm{TV}(u_\ell) \tag{6}$$

and using it as $R(\boldsymbol{u})$ in (1) yields the total variation TV1-CbC color denoising equation of (3). Likewise selecting $P = P_2$ [4] leads to the vectorial TV semi-norm

$$\begin{aligned} \int_\Omega |D\boldsymbol{u}| &= \int_\Omega \frac{\sum_{\ell=1}^{m} \langle \nabla u_\ell, \nabla u_\ell \rangle}{|\nabla \boldsymbol{u}|} \, dx_1 dx_2 \\ &= \int_\Omega \sqrt{\sum_{\ell=1}^{m} |\nabla u_\ell|^2} \, dx_1 dx_2. \end{aligned} \tag{7}$$

and the Euler–Lagrange equations defined by

$$-\alpha \nabla \cdot \frac{\nabla u_\ell}{||\nabla \boldsymbol{u}||} + u_\ell - u_\ell^0 = 0 \quad \text{in } \Omega \quad \ell = 1, \ldots, m \tag{8}$$

where $||\nabla \boldsymbol{u}|| = \sqrt{|\nabla u_1|^2 + \ldots + |\nabla u_m|^2}$.

Observe that, in the system of PDEs (8), the coupling among the channels is through the diffusion coefficient $D(\boldsymbol{u}) = ||\nabla \boldsymbol{u}||^{-1}$. Since $D(\boldsymbol{u})$ takes different values across the

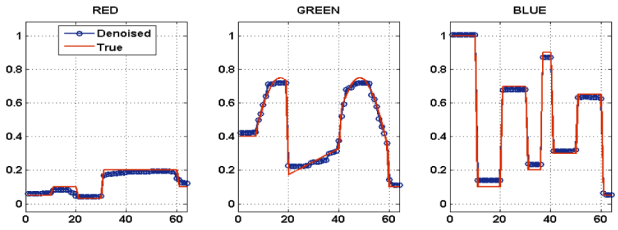


Fig. 3. Result from the TV2-BRC model.

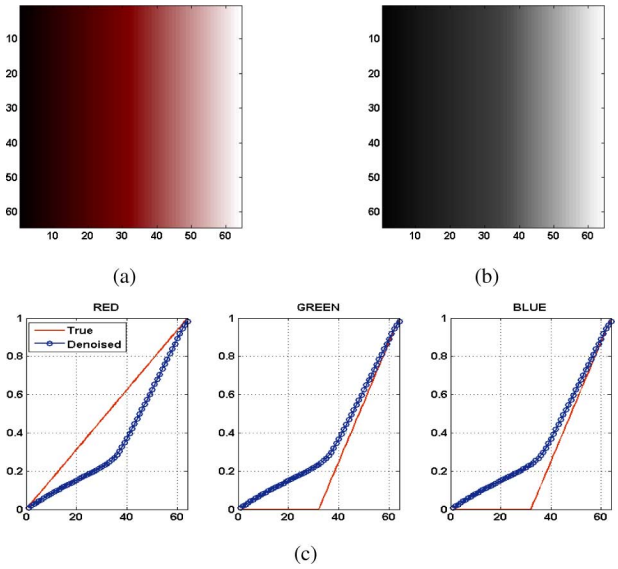


Fig. 4. Results from the TV2-BRC model applied to a simple 1-D color image with a kink in two of its channels. Dirichlet boundary conditions were used here. (a) Clean color image. (b) TV2-BRC denoising result. (c) 1-D plot of each channel. This problem was actually suggested in [2].

image, the level of coupling varies from one region to another *locally* adjusting the level of regularization. The result from applying the TV2-BRC model to our model problem of Fig. 1 is presented in Fig. 3. As can be seen for this example, the coupling between channels in the TV2-BRC model is so strong that it tends to align all channels causing color smearing in the image. This is further illustrated in Fig. 4 where a very large α in (8) was selected to make the phenomenon more evident.

C. Vectorial TV Model of Blomgren and Chan (TV3-BLC)

The Blomgren and Chan [2] model is based on an alternative generalization of the TV semi-norm (related to but different from (7)). Define $\mathrm{TV}(u_\ell) = \int_\Omega |\nabla u_\ell|\, dx_1 dx_2$ for channel ℓ. The following total variation semi-norm was proposed

$$\mathrm{TV}_m(\boldsymbol{u}) = \sqrt{\sum_{\ell=1}^{m} [\mathrm{TV}(u_\ell)]^2} \tag{9}$$

which, when used as $R(\boldsymbol{u})$ in (1), leads to the Euler–Lagrange equations (to be named as TV3-BLC) for $\ell = 1, \ldots, m$

$$-\alpha \frac{\mathrm{TV}(u_\ell)}{\mathrm{TV}_m(\boldsymbol{u})} \nabla \cdot \frac{\nabla u_\ell}{|\nabla u_\ell|} + u_\ell - u_\ell^0 = 0 \quad \text{in } \Omega \tag{10}$$

and $\nabla u_\ell \cdot \vec{\nu}_\ell = 0$ on the boundary $\partial\Omega$, where $\vec{\nu}_\ell$ is the normal unit vector on the boundary of the ℓth channel.

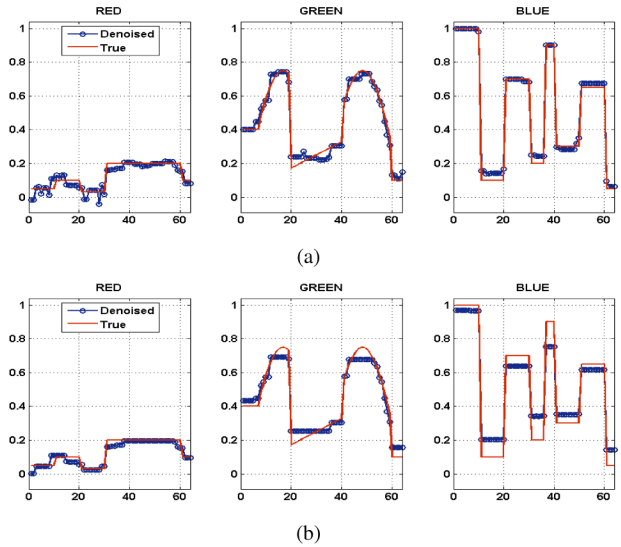


Fig. 5. Results from the TV3-BLC model for two different values of α, (a) $\alpha = 1/13$, and (b) $\alpha = 1/5$.

As seen from (10), the global quantity $\mathcal{A}(u_\ell) = \alpha(\mathrm{TV}(u_\ell))/(\mathrm{TV}_m(\boldsymbol{u}))$ plays the role of α_ℓ in (4) so the coupling in this model between channels is *global*.

This adjusting has the effect of preventing the wiping out of weak channels—a problem that the total variation TV1-CbC approach does have. This phenomenon is illustrated with the help of the Fig. 5 where we can compare with the Fig. 2 for TV1-CbC.

D. Vectorial Anisotropic Diffusion Model of Sapiro and Ringach (TV4-SaR)

The Sapiro and Ringach approach [25] constructs a new edge detector for vector valued images for a Perona–Malik model. Let $P = (x_1^0, x_2^0)$ and $Q = (x_1^1, x_2^1)$ be two points in the 3-D image space; then the difference of image values at these two points is defined as $D\boldsymbol{u} = \boldsymbol{u}(P) - \boldsymbol{u}(Q)$. When the Euclidean distance between two points tends to zero, the difference becomes the arc element $d\boldsymbol{u} = \sum_{i=1}^{2} (\partial \boldsymbol{u})/(\partial x_i) dx_i$ and its squared norm also called the *first fundamental form* [12], [21] is defined by $\sum_{i=1}^{2}\sum_{j=1}^{2} (\partial \boldsymbol{u})/(\partial x_i)(\partial \boldsymbol{u})/(\partial x_j) dx_i dx_j$, i.e.,

$$d\boldsymbol{u}^2 = \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}^T \begin{bmatrix} g_{1,1} & g_{1,2} \\ g_{2,1} & g_{2,2} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} \tag{11}$$

where $g_{i,j} = \partial \boldsymbol{u}/\partial x_i \cdot \partial \boldsymbol{u}/\partial x_j$. If $v = [dx_1\ dx_2]^T$ denotes a unit direction, then $d\boldsymbol{u}^2$ measures the rate of change in the direction v. To find the extrema of such changes, we are led to consider the directions of the eigenvectors $(\cos\theta_\pm, \sin\theta_\pm)$, and the corresponding eigenvalues $\lambda_\pm$ of the matrix $[g_{i,j}]$, given by

$$\theta_+ = \frac{1}{2}\tan^{-1}\left(\frac{2g_{1,2}}{g_{1,1} - g_{2,2}}\right) \quad \text{and} \quad \theta_- = \theta_+ + \frac{\pi}{2}$$

$$\lambda_\pm = \frac{g_{1,1} + g_{2,2} \pm \sqrt{(g_{1,1} - g_{2,2})^2 + 4g_{1,2}^2}}{2}.$$

Here $(\cos\theta_+, \sin\theta_+)$ is the direction of maximal change and λ_+ the maximal rate of change; similarly, $(\cos\theta_-, \sin\theta_-)$ points to the minimal change and λ_- the minimal rate.

It is proposed [25] to detect image discontinuities by defining a function $f = f(\lambda_+, \lambda_-)$ that measures the dissimilarity between λ_+ and λ_-. In [25], it was proposed to use any decreasing function $g = g(\lambda_+ - \lambda_-)$ and the evolution equation

$$\frac{\partial \boldsymbol{u}}{\partial t} = g(\lambda_+ - \lambda_-)\frac{\partial^2 \boldsymbol{u}}{\partial \theta^2} \tag{12}$$

to denoise color images. When $m = 1$ for the gray-scale images, $\lambda_+ = |\nabla u|^2$, $\lambda_- = 0$ so we have

$$\int_\Omega \sqrt{\lambda_+ - \lambda_-}\, dx_1 dx_2 = \int_\Omega |\nabla u|\, dx_1 dx_2 \tag{13}$$

which means that the above model (left) based on eigenvalues can be reduced to the TV model for single-valued images.

However, as remarked by Blomgren and Chan, for $m > 1$, the above choice of regularization using $R(\boldsymbol{u}) = \int_\Omega \sqrt{\lambda_+ - \lambda_-}\, dx_1 dx_2$ leads to extremely inconvenient minimization and the equally-justifiable alternative is to use $R(\boldsymbol{u}) = \int_\Omega \sqrt{\lambda_+ + \lambda_-}\, dx_1 dx_2 = \int_\Omega \sqrt{g_{1,1} + g_{2,2}}\, dx_1 dx_2$ which happens to be precisely the TV1-CbC.

In summary, and as remarked in [2] and [4], the TV3-BLC model seems to be superior in quality of restoration to both the TV1-CbC approach and the TV2-BRC model although the latter is equipped with a faster numerical solver [4].

III. New and High-Order Vector-Valued Models

We are ready to state our proposed models. As in the vectorial TV case we need to have a starting point and from there start moving on. In other words, we have to select a working high-order model for gray-scale images and upgrade it to vector-valued images. The idea is also to have coupling among channels as in the vectorial TV models since this has proven to improve the quality of reconstruction.

To this end we decided to use the curvature-based model [35] as our starting point. Our motivations are three fold: 1) this model has nice properties like no staircase effect and contrast and corners preservation [35], 2) a fast multigrid solver can be implemented for this model, and 3) curvature is an intrinsic geometric feature so generalization is easier.

First we review very briefly what it would be the CU1-CbC approach of this curvature-based model for color image denoising.

A. Channel by Channel Curvature-Based Model—(CU1-CbC)

Denote as usual $\boldsymbol{u} = (u_1, \ldots, u_m)$ the true vector-valued image, $\boldsymbol{u}^0 = (u_1^0, \ldots, u_m^0)$ the noisy image, and $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_m)$ the curvature vector with $\kappa_\ell = \nabla \cdot (\nabla u_\ell)/(|\nabla u_\ell|)$ the curvature of the ℓth channel of $\boldsymbol{u}$. Then solve

$$\min_{u_1,\ldots,u_m} \left\{ \alpha \int_\Omega \sum_{\ell=1}^{m} \Phi_\ell(\kappa_\ell)\, dx_1 dx_2 + \frac{1}{2}||\boldsymbol{u} - \boldsymbol{u}^0||_2^2 \right\} \tag{14}$$

which may be done by solving the system of Euler–Lagrange equations in Ω for $\ell = 1, \ldots, m$:

$$\alpha \nabla \cdot \left(\frac{\nabla \Phi_\ell'}{|\nabla u_\ell|} - \frac{\nabla u_\ell \cdot \nabla \Phi_\ell'}{|\nabla u_\ell|^3} \nabla u_\ell \right) + u_\ell - u_\ell^0 = 0 \tag{15}$$

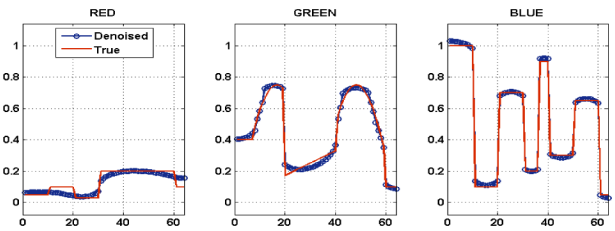


Fig. 6. Result from the curvature-based CU1-CbC approach. The small step in the weak red channel has been almost wiped out.

with boundary conditions $(\nabla \Phi_\ell')/(|\nabla u_\ell|) \cdot \vec{\nu_\ell} = 0$ for each channel. The objective of Φ_ℓ is to efficiently measuring the random high frequency oscillations representing noise in u_ℓ^0. Many different high-order options including $\Phi_\ell(k_\ell) = \kappa_\ell^2$ or $|\kappa_\ell|$ have been studied; some of those can be found in [34], [18], [19], and [7] and references therein. In particular, we use here and throughout the rest of this work $\Phi_\ell(k_\ell) = \kappa_\ell^2$ due to its nice properties already summarized above and detailed in [35]. With such a selection $\Phi_\ell' = 2\kappa_\ell$.

Here as in the TV1-CbC approach for the vectorial TV model, α is the same for all channels so we expect to have difficulties when denoising a color image having weak channels. This is, by selecting the best α for one channel we may over-smooth the weak channel. We illustrate this effect in the Fig. 6.

B. Local Curvature-Based Color Model—(CU2-LCM)

Here we introduce our first high-order denoising model for vector-valued images. We construct this model based on ideas from the Bresson and Chan model. This model is to solve

$$\min_{u_1,\ldots,u_m} \left\{ \alpha \int_\Omega \sqrt{\sum_{\ell=1}^{m} \Phi_\ell(\kappa_\ell)}\, dx_1 dx_2 + \frac{1}{2}||\boldsymbol{u} - \boldsymbol{u}^0||_2^2 \right\} \tag{16}$$

which leads to solve the Euler–Lagrange equations

$$\alpha \nabla \cdot \left(\frac{\nabla \Psi_\ell}{|\nabla u_\ell|} - \frac{\nabla u_\ell \cdot \nabla \Psi_\ell}{|\nabla u_\ell|^3} \nabla u_\ell \right) + u_\ell - u_\ell^0 = 0 \tag{17}$$

in Ω for $\ell = 1, \ldots, m$, with boundary conditions $(\nabla \Psi_\ell)/(|\nabla u_\ell|) \cdot \vec{\nu_\ell} = 0$ and Ψ_ℓ defined as

$$\Psi_\ell = \frac{\Phi_\ell'(\kappa_\ell)}{\sqrt{\sum_{\ell=1}^{m} \Phi_\ell(\kappa_\ell)}}. \tag{18}$$

As can be appreciated, the amount of diffusion in this model, is mainly affected by $\nabla \Psi_\ell$, a vector that *locally* varies across the image. Due to this, we name this model *the local curvature-based model* (CU2-LCM).

By analyzing the above equation, we observe that here we do not have the same problem as in the TV2-BRC model (its equivalent *local* TV model). In the TV2-BRC model the coefficient $||\nabla \boldsymbol{u}||^{-1}$ never stops diffusing across edges unless all are aligned. As a consequence, color is smeared. In the CU2-LCM model the two coefficients $|\nabla u_\ell|^{-1}$ and $|\nabla u_\ell|^{-3}$ only depend on u_ℓ so diffusion is properly stopped in every channel.

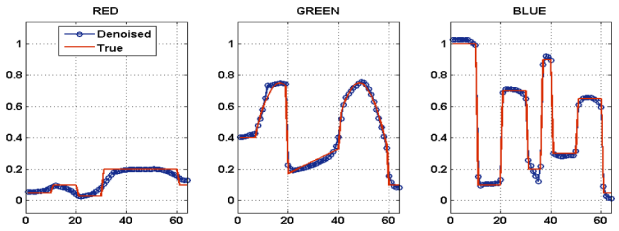


Fig. 7. Result from the CU2-LCM model. The small step in the weak red channel is better recovered but a kink appeared in the blue channel.

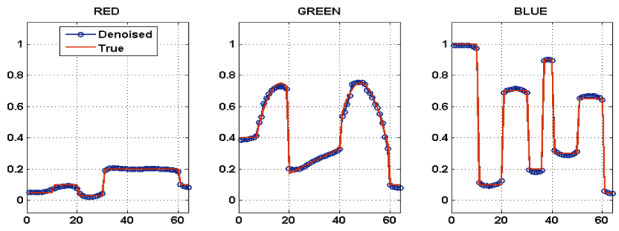


Fig. 8. Result from the CU3-GCM model. All channels, including the weak one, are very well reconstructed.

Unfortunately, our experiments revealed that this model has a tendency to develop kinks (see the blue channel in Fig. 7) in the image not only decimating the quality of restoration but making more difficult to construct stable numerical solvers as well.

C. Global Curvature-Based Color Model—(CU3-GCM)

Our second new high-order model is inspired by the Blomgren and Chan TV-based model. Here we propose to solve

$$\min_{u_1,\ldots,u_m}\left\{\alpha\sqrt{\int_\Omega \sum_{\ell=1}^{m}\Phi_\ell(\kappa_\ell)\,dx_1dx_2}+\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{u}^0\|_2^2\right\} \tag{19}$$

which leads to the Euler–Lagrange equations in Ω

$$\alpha\frac{Q_\ell(u_\ell)}{Q(\boldsymbol{u})}\nabla\cdot\left(\frac{\nabla\Phi'_\ell}{|\nabla u_\ell|}-\frac{\nabla u_\ell\cdot\nabla\Phi'_\ell}{|\nabla u_\ell|^3}\nabla u_\ell\right)+u_\ell-u_\ell^0=0 \tag{20}$$

with boundary conditions $(\nabla\Phi'_\ell)/(|\nabla u_\ell|)\cdot\vec{\nu_\ell}=0$ for $\ell=1,\ldots,m$ where $Q(\boldsymbol{u})$ and $Q_\ell(u_\ell)$ are defined by

$$\begin{aligned}Q(\boldsymbol{u})&=\sqrt{\sum_{\ell=1}^{m}Q_\ell(u_\ell)}\\ Q_\ell(u_\ell)&=\int_\Omega\Phi_\ell(\kappa_\ell)dx_1dx_2.\end{aligned} \tag{21}$$

Note that in the total variation TV3-BLC model, the amount of regularization or diffusion is determined by the quantity $\mathcal{A}(u_\ell)=\alpha\ \mathrm{TV}(u_\ell)/\mathrm{TV}_m(\boldsymbol{u})$ while here with the curvature model CU3-GCM we have $\mathcal{A}(u_\ell)=\alpha Q_\ell(u_\ell)/Q(\boldsymbol{u})$ which acts as a regularization weight for channel ℓ. For this reason, we call this model *the global curvature-based model*.

Thus, we expect this model to reduce regularization for weak channels, i.e., avoiding smearing them. This is a very similar idea to the one used in the TV3-BLC model; however, our experiments show that applied together with the curvature-based model, it delivers much better results. A clear example is given in Fig. 8 where the reconstruction of the small step in the weak left channel is very good and the quality in the others channels is preserved.

An extra feature of this model is that a fast nonlinear multigrid algorithm can be constructed as we will show in Section IV-A.

IV. Numerical Solution of the High-Order Models

Here, we shall first discuss the finite differences discretization of the curvature models CU1-CbC, CU2-LCM, and CU3-GCM, and then present numerical solution techniques. As the above tests have demonstrated, CU3-GCM is the best model of the three so we shall mainly address the solution of this model although the discussion applies to all three methods.

Assume the continuous domain $\Omega=[0,p]\times[0,q]$ and let (h_x,h_y) represent a vector of finite mesh sizes. We also define the infinite grid G_h as $G_h=\{(x,y):x=x_i=ih_x,\ y=y_j=jh_y;\ i,j\in\mathbb{Z}\}$ and for simplicity assume $p=q$ and $h=h_x=h_y$. Then the discrete grid is defined as $\Omega_h=\Omega\cap G_h$ and a discrete function on the grid Ω_h as $u_h=u_h(x,y)=u_h(x_i,y_j)=u_h(ih_x,jh_y)$. These discrete functions take on scaled values in the interval $[0,1]$. We also denote the derivative with respect to any variable ψ as $(\cdot)_\psi$.

Thus, for any vector $V=(V^1,V^2)$ its divergence is approximated using central differences, i.e., $\nabla\cdot V_{i,j}=(V^1_{i+1/2,j}-V^1_{i-1/2,j})/h+(V^2_{i,j+1/2}-V^2_{i,j-1/2})/h$.

We now show how to approximate all the other involved quantities at the half-points. It shall be enough to show it only for $V^1_{i+1/2,j}$ and $V^1_{i-1/2,j}$ since the others can be done similarly.

Curvature by

$$\kappa_{i,j}=\frac{(u_x)_{i+\frac{1}{2},j}}{|\nabla u|_{i+\frac{1}{2},j}}-\frac{(u_x)_{i-\frac{1}{2},j}}{|\nabla u|_{i-\frac{1}{2},j}}+\frac{(u_y)_{i,j+\frac{1}{2}}}{|\nabla u|_{i,j+\frac{1}{2}}}-\frac{(u_y)_{i,j-\frac{1}{2}}}{|\nabla u|_{i,j-\frac{1}{2}}}.$$

Partial derivatives in x by the central differencing of two adjacent *whole* pixels

$$\begin{aligned}(u_x)_{i+\frac{1}{2},j}&=\frac{u_{i+1,j}-u_{i,j}}{h}\\ (u_x)_{i-\frac{1}{2},j}&=\frac{u_{i,j}-u_{i-1,j}}{h}\\ (\Phi'_x)_{i+\frac{1}{2},j}&=\frac{\Phi'_{i+1,j}-\Phi'_{i,j}}{h}\\ (\Phi'_x)_{i-\frac{1}{2},j}&=\frac{\Phi'_{i,j}-\Phi'_{i-1,j}}{h}\\ |\nabla u|_{i+\frac{1}{2},j}&=\sqrt{\left((u_x)_{i+\frac{1}{2},j}\right)^2+\left((u_y)_{i,j+\frac{1}{2}}\right)^2+\beta}\end{aligned}$$

where $\beta > 0$ is a small regularization parameter. Partial derivatives in y by the *min-mod* [23] of $(\cdot)_y$'s at two adjacent *whole* points

$$
\begin{aligned}
(u_y)_{i+\frac{1}{2},j} &= \min-\mathrm{mod} \\
&\quad \times \left(\frac{u_{i+1,j+1} - u_{i+1,j-1}}{2h}, \frac{u_{i,j+1} - u_{i,j-1}}{2h} \right) \\
(u_y)_{i-\frac{1}{2},j} &= \min-\mathrm{mod} \\
&\quad \times \frac{u_{i,j+1} - u_{i,j-1}}{2h}, \frac{u_{i-1,j+1} - u_{i-1,j-1}}{2h} \\
(\Phi'_y)_{i+\frac{1}{2},j} &= \min-\mathrm{mod}(\zeta, \vartheta) \text{ with} \\
\zeta &= \frac{\Phi'_{i+1,j+1} - \Phi'_{i+1,j-1}}{2h} \text{ and} \\
\vartheta &= \frac{\Phi'_{i,j+1} - \Phi'_{i,j-1}}{2h}. \\
(\Phi'_y)_{i-\frac{1}{2},j} &= \min-\mathrm{mod}(\zeta, \vartheta) \text{ with} \\
\zeta &= \frac{\Phi'_{i,j+1} - \Phi'_{i,j-1}}{2h}, \vartheta = \frac{\Phi'_{i-1,j+1} - \Phi'_{i-1,j-1}}{2h}. \\
|\nabla u|_{i-\frac{1}{2},j} &= \sqrt{\left((u_x)_{i-\frac{1}{2},j}\right)^2 + \left((u_y)_{i,j-\frac{1}{2}}\right)^2 + \beta}.
\end{aligned}
$$

For all models we use Neumann's boundary condition on $\partial\Omega$ since this class automatically satisfies the particular boundary conditions of each model. This is done as follows: $u_{i,0} = u_{i,1},\ u_{i,q+1} = u_{i,q},\ u_{0,j} = u_{1,j},\ u_{p+1,j} = u_{p,j}$.

Additionally, for the CU2-LCM model, Ψ_ℓ is computed by

$$
(\Psi_\ell)_{i+\frac{1}{2},j} = \frac{(\Phi_\ell)'_{i+\frac{1}{2},j}(\kappa_\ell)}{\sqrt{\sum_{\ell=1}^{m}(\Phi_\ell)_{i+\frac{1}{2},j}(\kappa_\ell)}}. \tag{22}
$$

Also for the CU3-GCM model, $Q_\ell(u_\ell)$ and $Q(\boldsymbol{u})$ are computed in the following way:

$$
Q_\ell(u_\ell) = \sum_{i=1}^{p}\sum_{i=1}^{q}\Phi_\ell(k_\ell)_{i,j},\ Q(\boldsymbol{u}) = \sqrt{\sum_{\ell=1}^{m} Q_\ell(u_\ell)}. \tag{23}
$$

A. Stable Fixed Point Method

Here we present a general method to implement a fixed point algorithm for any of the high-order models presented so far. Later on this section, we will show how to modify slightly this method to obtain an optimal performance for the CU3-GCM model. Then we will introduce a nonlinear multigrid method for it.

The Euler–Lagrange equations of all of the high-order models can be written in the general form

$$
\mathcal{A}(u_\ell)\nabla\cdot(\mathcal{B}(u_\ell) - \mathcal{C}(u_\ell)\nabla u_\ell) + u_\ell - u_\ell^0 = 0 \tag{24}
$$

where for $\ell = 1, \ldots, m$ and, respectively, for the CU1-CbC model, the CU2-LCM model, and the CU3-GCM model

$$
\begin{aligned}
\mathcal{A}(u_\ell) &= \alpha, \quad \mathcal{B}(u_\ell) = \frac{\nabla\Phi'_\ell}{|\nabla u_\ell|} \\
\mathcal{C}(u_\ell) &= \frac{\nabla u_\ell \cdot \nabla\Phi'_\ell}{(|\nabla u_\ell|)^3}; \\
\mathcal{A}(u_\ell) &= \alpha\frac{Q(u_\ell)}{Q(\boldsymbol{u})}, \quad \mathcal{B}(u_\ell) = \frac{\nabla\Phi'_\ell}{|\nabla u_\ell|} \\
\mathcal{C}(u_\ell) &= \frac{\nabla u_\ell \cdot \nabla\Phi'_\ell}{(|\nabla u_\ell|)^3}. \\
\mathcal{A}(u_\ell) &= \alpha, \quad \mathcal{B}(u_\ell) = \frac{\nabla\Psi_\ell}{|\nabla u_\ell|} \\
\mathcal{C}(u_\ell) &= \frac{\nabla u_\ell \cdot \nabla\Psi_\ell}{(|\nabla u_\ell|)^3};
\end{aligned}
$$

To obtain a fast solution of the above PDE a fixed point method similar to the ones described in [33], [32], [8], and [26] for the TV model would be desirable. Straightforward implementation of such a FP method does not work for PDEs like (24) so we use a method where a stabilizing term $\mathcal{N}(u_\ell)$ is included and the following fast fixed point scheme is used:

$$
\begin{aligned}
&-\gamma\mathcal{N}\left(u_\ell^{k+1}\right) - \mathcal{A}\left(u_\ell^{k+1}\right)\nabla\cdot\left(\mathcal{C}\left(u_\ell^{k+1}\right)\nabla u_\ell^{k+1}\right) + u_\ell^{k+1} \\
&\qquad = -\gamma\mathcal{N}\left(u_\ell^{k}\right) - \mathcal{A}\left(u_\ell^{k}\right)\nabla\cdot\left(\mathcal{B}\left(u_\ell^{k}\right)\right) + u_\ell^{0}.
\end{aligned} \tag{25}
$$

For the CU1-CbC and CU3-GCM models $\mathcal{N}(u_\ell) = \nabla\cdot(\nabla u_\ell)/(|\nabla u_\ell|)$ is selected. For the CU2-LCM model however $\mathcal{N}(u_\ell) = \nabla\cdot((\nabla u_\ell)/(|\nabla u_\ell|\sqrt{\sum_{\ell=1}^{m}(\Phi_\ell)_{i,j}(\kappa_\ell)}))$ provides a better performance of the algorithm.

To solve (25), we linearize it by lagging the nonlinear terms $|\nabla u_\ell|^{-1}$ in $\mathcal{N}(u_\ell^{k+1})$, $\mathcal{A}(u_\ell)$ and $\mathcal{C}(u_\ell)$ in its left-hand side, i.e., evaluating them at kth step. Hence, we are led to solve a linear system of equations of the form

$$
\begin{aligned}
&(u_\ell)_{i,j}^{k+1}\mathbb{S}_{i,j}^{k} - (u_\ell)_{i+1,j}^{k+1}C_{i+\frac{1}{2},j}^{k} - (u_\ell)_{i-1,j}^{k+1}C_{i-\frac{1}{2},j}^{k} \\
&\quad - (u_\ell)_{i,j+1}^{k+1}C_{i,j+\frac{1}{2}}^{k} - (u_\ell)_{i,j-1}^{k+1}C_{i,j-\frac{1}{2}}^{k} \\
&\qquad = f\left(u_\ell^k, u_\ell^0, \alpha, \gamma\right)_{i,j}
\end{aligned} \tag{26}
$$

where for example for the CU1-CbC and CU3-GCM models $C_{i+1/2,j}^{k} = (\gamma)/(|\nabla u_\ell^k|_{i+1/2,j}) + \alpha((\nabla u_\ell^k\cdot\nabla\Phi'_\ell(\kappa_\ell))/((|\nabla u_\ell^k|)^3))_{i+1/2,j}$ and so on,

$$
\mathbb{S}_{i,j}^{k} = 1 + C_{i+\frac{1}{2},j}^{k} + C_{i-\frac{1}{2},j}^{k} + C_{i,j+\frac{1}{2}}^{k} + C_{i,j-\frac{1}{2}}^{k} \tag{27}
$$

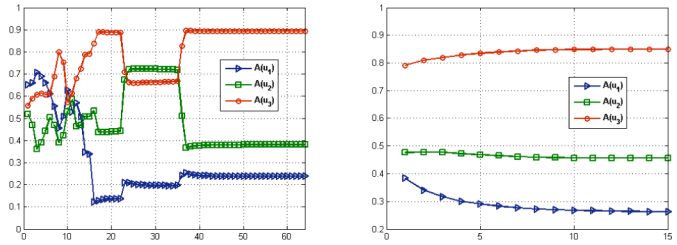
and f is defined as the right-hand side of (25). This linear system can be arranged in matrix form $A(u_\ell^k)u_\ell^{k+1} = f(u_\ell^k, u_\ell^0, \alpha, \gamma)$ with A a sparse, symmetric and positive definite matrix. To solve this system we use a simple lexicographic Gauss–Seidel method. The procedure is the following.

Algorithm 1 CFPGS

$\boldsymbol{u} \leftarrow \mathrm{CFPGS}(\boldsymbol{u}, \boldsymbol{u}^0, gsiter, h, \alpha, \gamma, \nu)$

Require: On a grid with mesh size h, choose an initial guess $\boldsymbol{u} = (u_1, \ldots, u_m)$ for (25)

1: **for** $\ell = 1$ to m **do**
2: **for** $k = 1$ to ν **do**
3: Apply $gsiter$ Gauss-Seidel iterations to the linear system $A(u_\ell^k)u_\ell^{k+1} = f(u_\ell^k)$
4: **end for**
5: **end for**



(a) (b)

Fig. 9. Evolution of the values of $\mathcal{A}(u_\ell)$ in the CU3-GCM and TV3-BLC models when solving the problem of Fig. 1. In the CU3-GCM model and within the first 40 iterations the values continuously and strongly change until reaching a stable state.

From the evidence shown in Section III, there is clear indication that the CU3-GCM method is the best among the high-order models. Due to this, we now concentrate on optimizing the Algorithm 1 for this model.

A number of experiments revealed that the correct selection of the value of the stabilizing constant γ is very important for the good performance of the numerical algorithm (25) and the value of γ among others strongly depends on the value of the regularization parameter α. In the CU3-GCM model the term $\mathcal{A}(u_\ell)$ is a constant that is varying as the iterative algorithm evolves. That is, we can view the CU3-GCM model as the CU1-CbC model with varying regularization parameter α.

We show in Fig. 9 an example of the evolution of $\mathcal{A}(u_\ell)$ for the CU3-GCM and TV3-BLC models where for the latter variations are much more moderate. Clearly, in our high-order CU3-GCM model $\mathcal{A}(u_\ell)$ will affect the performance of the algorithm if a fixed γ is selected. This was confirmed in our initial experiments. In view of this, we use $\gamma_\ell = \gamma/\mathcal{A}(u_\ell)$ and solve instead the equation

$$-\gamma_\ell \mathcal{N}\left(u_\ell^{k+1}\right) - \mathcal{A}\left(u_\ell^{k+1}\right)\nabla\cdot\left(\mathcal{C}\left(u_\ell^{k+1}\right)\nabla u_\ell^{k+1}\right) + u_\ell^{k+1} = -\gamma_\ell \mathcal{N}\left(u_\ell^{k}\right) - \mathcal{A}\left(u_\ell^{k}\right)\nabla\cdot\left(\mathcal{B}\left(u_\ell^{k}\right)\right) + u_\ell^{0}. \quad (28)$$

This method automatically increases the value of γ_ℓ when $\mathcal{A}(u_\ell)$ is small and decreases it when $\mathcal{A}(u_\ell)$ is large. We now proceed to introduce a multigrid algorithm for the CU3-GCM model.

B. Nonlinear Multigrid Algorithm

We can go one step further and use the above fixed-point method (28) as the foundation for a nonlinear FAS MG algorithm [31], [9]. This algorithm has been successfully developed and tested in a number of imaging problems, for instance: [15] on image registration, [1], [22] on image segmentation, [26], [27], [6], [13], [11] on image denoising-deblurring, and [5] on image inpainting.

Multigrid schemes considerably speed up numerical processes achieving fast results by constructing a hierarchy of discretizations where at each level the error equation is partially solved and the new approximation transported to next coarser level. This process is recursively applied until reaching the coarsest level where and exact but computationally cheap solution is obtained. Then the process move backwards on the hierarchical structure transporting the more accurate error and updating the approximate solution at each level until reaching the finest level again. Usually standard coarsening is used to construct the hierarchical structure halving the number of variables on each dimension at each level.

To apply this scheme to our problem we define the nonlinear discrete equations

$$(N_\ell u_\ell)_{i,j} = \left(u_\ell^0\right)_{i,j}, \quad \ell = 1,\ldots,m \quad (29)$$

where $(N_\ell u_\ell)_{i,j} = \mathcal{A}_{i,j}(u_\ell)\nabla\cdot(\mathcal{B}_{i,j}(u_\ell) - \mathcal{C}_{i,j}(u_\ell)(\nabla u_\ell)_{i,j}) + (u_\ell)_{i,j}$, and define $\boldsymbol{N}_h = [N_1,\ldots,N_m]$ as the vector of nonlinear operators on the grid of mesh size h such that $\boldsymbol{N}_h\boldsymbol{u}_h = [N_1u_1,\ldots,N_mu_m]$. Define the residual equations as $r_\ell = N_\ell u_\ell - u_\ell^0$ and the correspondent vector $\boldsymbol{r} = [r_1,\ldots,r_m]$. Then the nonlinear MG scheme for the vector-valued problem is stated in Algorithms 2 and 3.

Algorithm 2 Nonlinear Multigrid Method

Require: Select an initial guess $\boldsymbol{u}$ on the finest grid h

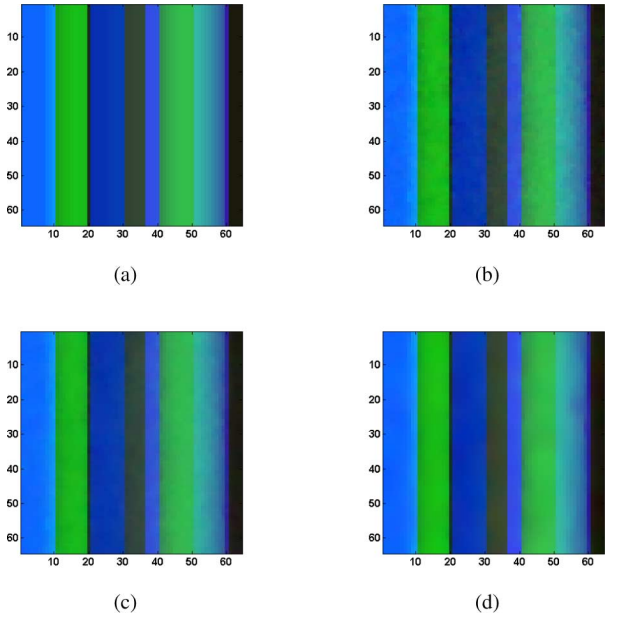
1: $k \leftarrow 0, err \leftarrow tol + 1$
2: **while** $err < tol$ **do**
3: $\boldsymbol{u}_h^{k+1} \leftarrow FAS(\boldsymbol{u}_h^k, \boldsymbol{N}_h^k, \boldsymbol{u}_h^0, \nu_0, \nu_1, \nu_2, gsiter, \alpha, \gamma)$
4: $err = \max\{\|(r_1)_h^{k+1}\|_2, \ldots, \|(r_m)_h^{k+1}\|_2\}$
5: $k \leftarrow k + 1$
6: **end while**

Algorithm 3 FAS Cycle

$u_h \leftarrow FAS(\boldsymbol{u}_h, \boldsymbol{N}_h, \boldsymbol{u}_h^0, \nu_0, \nu_1, \nu_2, gsiter, \alpha, \gamma)$

1: **if** Ω_h = coarsest grid **then**
2: solve $\boldsymbol{N}_h\boldsymbol{u}_h = \boldsymbol{u}_h^0$ accurately (i.e., ν_0 iterations by CFPGS) and return.
3: **else**
4: continue with step 6.
5: **end if**
6: Pre-smoothing: Do ν_1 steps of $\boldsymbol{u}_h \leftarrow CFPGS(\boldsymbol{u}_h, \boldsymbol{u}_h^0, gsiter, \alpha, \gamma, \nu_1)$
7: Restrict to the coarse grid, $\boldsymbol{u}_{2h} \leftarrow R_h^{2h}\boldsymbol{u}_h$
8: Set the initial solution for the next level, $\bar{\boldsymbol{u}}_{2h} \leftarrow \boldsymbol{u}_{2h}$
9: Compute the new right hand side $\boldsymbol{u}_{2h}^0 \leftarrow R_h^{2h}(\boldsymbol{u}_h^0 - \boldsymbol{N}_h\boldsymbol{u}_h) + \boldsymbol{N}_{2h}\boldsymbol{u}_{2h}$
10: Implement $\boldsymbol{u}_{2h} \leftarrow FAS_{2h}(\boldsymbol{u}_{2h}, \boldsymbol{N}_{2h}, \boldsymbol{u}_{2h}^0, \nu_0, \nu_1, \nu_2, gsiter, \alpha, \gamma)$
11: Add the residual correction, $\boldsymbol{u}_h \leftarrow \boldsymbol{u}_h + I_{2h}^h(\boldsymbol{u}_{2h} - \bar{\boldsymbol{u}}_{2h})$
12: Post-smoothing: Do ν_2 steps of $\boldsymbol{u}_h \leftarrow CFPGS(\boldsymbol{u}_h, \boldsymbol{u}_h^0, gsiter, \alpha, \gamma, \nu_2)$

As a smoother we use our fixed-point CFPGS algorithm; however, to achieve the best possible performance of the MG algorithm, we found that applying over-relaxation into the GS sweeps (SOR method [24]) considerably speeds convergence, i.e., we update the unknown using $u^{p+1} = \omega u^{GS} + (1-\omega)u^p$ with $\omega = 1.5$. We also apply a small number of extra local relaxation steps around the difficult points (edges of the image) as suggested in [3] which is where the most in-homogeneous



(a) (b) (c) (d)

Fig. 10. (a) Clean image. (b) Denoised image using the TV3-BLC model with $\alpha = 1/13$. (c) Denoised image using the TV3-BLC model with $\alpha = 1/5$. (d) Denoised image using the CU3-GCM model.

parts are located. This guarantees the residual to be smooth enough before being transported to the next coarser grid and is computationally very cheap since these regions represent normally less than 1/7 of the whole image domain so the overall cost of the MG algorithm is incremented in a very small percentage.

V. Further Numerical Experiments

We now proceed to show some results obtained using the CU3-GCM model and the multigrid algorithm outlined above, using three realistic color images.

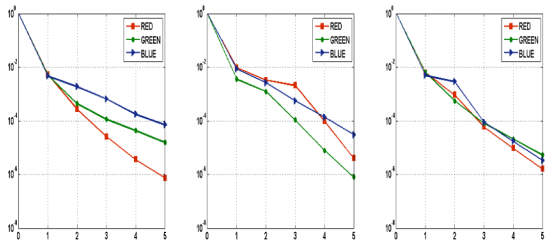
Of the three total variation models, the general belief [2], [4] is that the TV3-BLC model delivers the best result. Of the three high-order models (CU1-CbC, CU2-LCM, and CU3-GCM) we have presented here, our experiments suggest that CU3-GCM is better than the other two. Due to these facts, we only compare the quality of restoration yielded by the TV3-BLC and CU3-GCM models.

A. Quality of Restoration

The one-dimensional plots shown in Figs. 5 and 8 already suggest that our color global curvature-based model outperforms the TV3-BLC model. Maybe a more accurate comparison can be carried out by computing the peak-signal-to-noise-ratio (PSNR) values between the true image $\boldsymbol{u}^t$ and the denoised image $\boldsymbol{u}$ for each channel. The PSNR measure for one channel of size $p \times q$ is defined as

$$\mathrm{PSNR} = 20 \log_{10} \left(\frac{1}{\mathrm{RMSE}\,(u_\ell, u_\ell^t)} \right) \tag{30}$$

where $\mathrm{RMSE}(u_\ell, u_\ell^t) = \sqrt{(\sum_{i,j}((u_\ell)_{i,j} - (u_\ell^t)_{i,j})^2)/(pq)}$ and usually the larger the PSNR is, the better restoration of the image is obtained. It should be noted however that PNSR not always correlates with human perception. In a real life situation, such a measure is also not possible because $\boldsymbol{u}^t$ is not known.

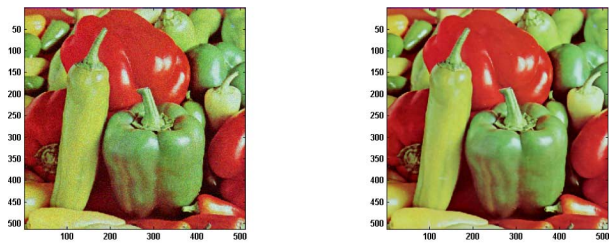


(a) (b) (c)

Fig. 11. Residuals versus the cycles of a nonlinear multigrid algorithm when solving the problems of (a) Fig. 12, (b) Fig. 13, and (c) Fig. 14 all with size 256×256 and SNR ≈ 20.

TABLE I
PSNR Values From the TV3-BLC and CU3-GCM Models

Model	PSNR		
	RED	GREEN	BLUE
TV3-BLC ($\alpha = 1/13$)	75.54	74.86	73.30
TV3-BLC ($\alpha = 1/5$)	79.89	74.52	63.36
CU3-GCM	82.00	80.94	77.93



(a) (b)

Fig. 12. Example 1: Denoising example using the CU3-GCM model.

From the obtained PSNR values for the model problem of Fig. 1 and presented in Table I, it becomes quite clear that the CU3-GCM model delivers a much better restoration than the TV3-BLC model. In this case, is visually quite evident from Fig. 10 that this time PSNR does correspond with human perception.

Even more, when carefully inspecting Fig. 10(b), we see that the denoised image coming from the TV3-BLC method with $\alpha = 1/13$ looks a bit dirty. This is a combined effect of staircase plus noise still present on the weak red channel, as also backed up by the results shown in Fig. 5(a). By increasing the regularization with $\alpha = 1/5$, now the denoised image in Fig. 10(c) looks visually much better because noise has been removed from the red channel which we can confirm in Fig. 5(b). However, by doing so, the blue channel is over smoothed, see again Fig. 5(b), and its correspondent PSNR value gets worsened as seen from Table I. Also notice that the staircase effect still can be appreciated in Fig. 10(c). After all seems to be that the TV3-BLC model cannot cope easily with unbalanced channels.

Finally, in Figs. 12–14, we present some qualitative results. Fig. 12 shows a noisy image and its nice reconstruction by the CU3-GCM model. The aim of Figs. 13 and 14 is to compare

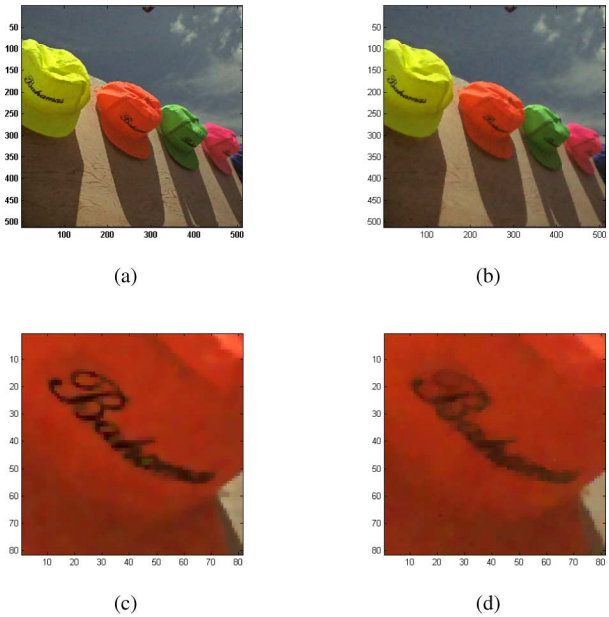


Fig. 13. Example 2: (a) Denoising example using the CU3-GCM model. (b) Denoising example using the TV3-BLC model. (c) A magnified region taken from (a). Correspondent magnified region taken from (b).

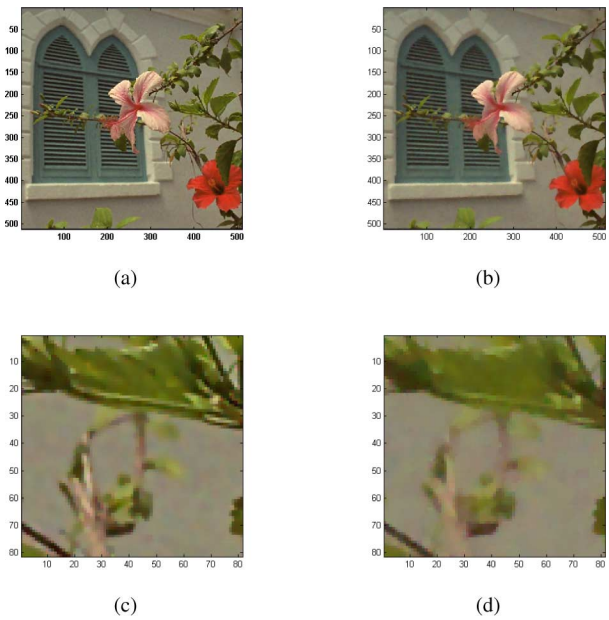


Fig. 14. Example 3: (a) Denoising example using the CU3-GCM model. (b) Denoising example using the TV3-BLC model. (c) A magnified region taken from (a). Correspondent magnified region taken from (b).

the quality of restoration between TV3-BLC and CU3-GCM. Although both restorations of Fig. 13(a) and (b) may look very similar, when details are magnified as in Fig. 13(c) and (d), the CU3-GCM model shows a superior quality of restoration. Similar behavior is observed in Fig. 14.

B. Multigrid Performance

Now we proceed to illustrate the fast performance of the nonlinear multigrid algorithm for the CU3-GCM model. In Fig. 11, we present the history iteration for solving each one of the problems from Figs. 12–14 all with SNR ≈ 20. For these problems we used the following parameters: $\gamma = 40$, $\beta = 10^{-2}, \alpha = 1/350$, $\nu_1 = \nu_2 = 10$, $gsiter = 2$. Clearly, the MG iteration is very good reaching very quickly very small residuals. A good stopping criteria for the MG algorithm is to stop when the relative residual is less than 10^{-4}.

TABLE II
COMPARISON OF CPU-TIMES AND PSNR VALUES FROM THE MG AND CFPGS ALGORITHMS WHEN SOLVING THE PROBLEMS OF FIGS. 12–14. THE MG ALGORITHM IS ROUGHLY FROM 5 TO 6 TIMES FASTER AND BOTH DELIVER THE SAME QUALITY OF RECONSTRUCTION

Image	Size pixels^2	Multigrid			
		CPU	RED	GREEN	BLUE
Ex. 1 Hats	128^2	79	72.38	72.80	74.95
	256^2	332	74.67	74.63	75.37
	512^2	1,409	74.27	74.25	73.69
Ex. 2 Flowers	128^2	78	64.37	62.55	63.02
	256^2	325	68.72	68.26	68.79
	512^2	1,401	72.51	72.20	72.39 5
Ex. 3 Peppers	128^2	91	69.51	67.30	68.78 0
	256^2	383	73.03	71.98	72.74
	512^2	1,411	71.13	71.83	71.60
		Fixed Point			
		CPU	RED	GREEN	BLUE
Ex. 1 Hats	128^2	458	72.45	72.99	74.17
	256^2	1,790	74.56	74.56	75.30
	512^2	7,460	74.12	74.09	73.61
Ex. 2 Flowers	128^2	451	64.28	62.44	63.23
	256^2	1,715	68.37	68.19	68.62
	512^2	7,425	72.47	72.11	72.35
Ex. 3 Peppers	128^2	537	69.32	67.33	68.70
	256^2	2,336	72.98	72.10	72.87
	512^2	8,748	70.97	71.76	71.52

In Table II, we present the CPU-time consumed for both, MG and CFPGS algorithms, when solving the same problems of above now with noisier images i.e., SNR ≈ 7 and using the stopping criteria just described above. Although both methods are pretty fast, MG is roughly from 5 to 6 times faster than CFPGS. We also present the obtained PSNR values to illustrate the well-balanced restoration of all of the three channels of the color images. All simulations were carried out using Matlab® 2008a on a 2.8-GHz Intel-Xeon-based computer.

VI. CONCLUSION

In this paper, we have introduced three high-order models for color image denoising. These models were designed from the curvature-based denoising model for gray-scale images originally published in [35].

The new models are designed to have coupling among the channels of the color image which many researchers have identified as highly desirable. Of the presented three curvature models, the global CU3-GCM model delivers the best results. Further a fast numerical multigrid algorithm is constructed for this model.

A comparison between the new CU3-GCM model and the previously best TV3-BLC model was presented in denoising color images. The CU3-GCM model is found to cope better with unbalanced channels in the common situation where different levels of illumination are present when a color picture (with noise) is produced.

REFERENCES

[1] N. Badshah and K. Chen, "Multigrid method for the chan-vese model in variational segmentation," *Commun. Comput. Phys.*, vol. 4, no. 2, pp. 294–316, 2008.

[2] P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 304–309, Mar. 1998.

[3] A. Brandt, "Multigrid techniques: 1984 guide with applications to fluid dynamics," *Gesellschaft für Mathematik und Datenverarbeitung*, 1984, GMD-Studie Nr. 85.

[4] X. Bresson and T. F. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *Inverse Problems and Imaging*, vol. 2, no. 4, pp. 455–484, 2008.

[5] C. Brito-Loeza and K. Chen, "Multigrid method for a modified curvature driven diffusion model for image inpainting," *J. Comput. Math.*, vol. 26, pp. 856–875, 2008.

[6] T. F. Chan, K. Chen, and J. L. Carter, "Iterative methods for solving the dual formulation arising from image restoration," *Electron. Trans. Numer. Anal.*, vol. 26, pp. 299–311, 2007.

[7] T. F. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 503–516, 2000.

[8] T. F. Chan and J. Shen, *Image Processing and Analysis—Variational, PDE, Wavelet, and Stochastic Methods.*. Philadelphia, PA: SIAM, 2005.

[9] K. Chen, *Matrix Preconditioning Techniques and Applications.*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[10] Y. Chen, S. Levine, and M. Rao, "Variable exponent, linear growth functionals in image restoration," *SIAM J. Appl. Math.*, vol. 66, no. 4, pp. 1383–1406, 2006.

[11] M. Donatelli, "A multigrid for image deblurring with Tikhonov regularization," *Numer. Linear Algebra With Applicat.*, vol. 12, pp. 715–729, 2005.

[12] M. Do Carmo, *Differential Geometry of Curves and Surfaces.*. New York: Prentice-Hall, 1976.

[13] C. Frohn-Schauf, S. Henn, and K. Witsch, "Nonlinear multigrid methods for total variation image denoising," *Comput. Visualiz. in Sci.*, vol. 7, pp. 199–206, 2004.

[14] S. Henn and K. Witsch, "A multigrid approach for minimizing a nonlinear functional for digital image matching," *Computing*, vol. 64, pp. 339–348, 2000.

[15] L. Homke, "A multigrid method for anisotropic PDEs in elastic image registration," *Numer. Linear Algebra With Applicat.*, vol. 13, pp. 215–229, 2006.

[16] S. L. Keeling and G. Haase, "Geometric multigrid for high-order regularizations of early vision problems," *Appl. Math. Comput.*, vol. 184, pp. 536–556, 2007.

[17] H. Kostler, K. Ruhnau, and R. Wienands, "Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer," *Numer. Linear Algebra With Applicat.*, vol. 15, pp. 201–218, 2008.

[18] M. Lysacker, A. Lundervold, and X.-C. Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1579–1590, Dec. 2003.

[19] M. Lysacker, S. Osher, and X.-C. Tai, "Noise removal using smoothed normals and surface fitting," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1345–1357, Oct. 2004.

[20] A. Marquina and S. Osher, "Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 387–405, 2000.

[21] B. O'Neil, *Elementary Differential Geometry*. London, U.K.: Academic, 1997.

[22] G. Papandreou and P. Maragos, "Multigrid geometric active contour models," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 229–240, Jan. 2007.

[23] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.

[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA: SIAM, 2003.

[25] G. Sapiro and D. L. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Trans. Image Process.*, vol. 5, no. 11, pp. 1582–1586, Nov. 1996.

[26] J. Savage and K. Chen, "An improved and accelerated non-linear multigrid method for total-variation denoising," *Int. J. Comput. Math.*, vol. 82, no. 8, pp. 1001–1015, 2005.

[27] J. Savage and K. Chen, "On multigrids for solving a class of improved total variation based staircasing reduction models," in *Image Processing Based on Partial Differential Equations*, X.-C. Tai, K.-A. Lie, T. F. Chan, and S. Osher, Eds. New York: Springer-Verlag, 2006, pp. 69–94.

[28] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 310–318, Mar. 1998.

[29] R. S. Spitareli, R. March, and D. Arena, "A multigrid finite-difference method for the solution of Euler equations of the variational image segmentation," *Appl. Numer. Math.*, vol. 39, pp. 181–189, 2001.

[30] M. Sturmer, H. Kostler, and U. Rude, "A fast full multigrid solver for applications in image processing," *Numer. Linear Algebra With Applicat.*, vol. 15, pp. 187–200, 2008.

[31] U. Trottenberg, C. Oosterlee, and A. Schuller, *Multigrid*. San Diego, CA: Academic, 2001.

[32] C. R. Vogel, *Computational Methods for Inverse Problems*. Philadelphia, PA: SIAM, 2002.

[33] C. R. Vogel and M. E. Oman, "Iterative algorithms for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, pp. 227–238, 1996.

[34] Y.-L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1723–1730, Oct. 2000.

[35] W. Zhu and T. F. Chan, "Image denoising using mean curvature," [Online]. Available: http://www.math.nyu.edu/~wzhu/, Preprint

Carlos Brito-Loeza received the B.Sc. degree in electronics from the Technological Institute of Mérida, Yucatan, Mexico, and the M.Sc. degree in mathematics from the University of Yucatan. He is currently pursuing the Ph.D. degree in fast algorithms for high-order variational models with applications to image inpainting and denoising at the Department of Mathematical Sciences, University of Liverpool, Liverpool, U.K.

Ke Chen received the B.Sc., M.Sc., and Ph.D. degrees in mathematics, respectively, from the Dalian University of Technology, China, University of Manchester, U.K., and the University of Plymouth, U.K.

He is a computational mathematician specializing in developing novel and fast numerical algorithms for various scientific computing applications, as reflected in his recent research monograph *Matrix Preconditioning Techniques and Applications* (Cambridge Univ. Press, 2005). He is the Director of a multidisciplinary research Centre for Mathematical Imaging Techniques (CMIT) and a Professor of mathematical sciences of the University of Liverpool, Liverpool, U.K., where he heads the computational mathematics group. His research interests are currently centered on developing effective solvers for variational image processing models, including image restoration, image inpainting, image segmentation, and registration.