Nonlocal Mumford-Shah Regularizers for Color Image Restoration

Miyoun Jung, Xavier Bresson, Tony F. Chan, and Luminita A. Vese

Abstract—We propose here a class of restoration algorithms for color images, based upon the Mumford-Shah (MS) model and nonlocal image information. The Ambrosio-Tortorelli and Shah elliptic approximations are defined to work in a small local neighborhood, which are sufficient to denoise smooth regions with sharp boundaries. However, texture is nonlocal in nature and requires semilocal/non-local information for efficient image denoising and restoration. Inspired from recent works (nonlocal means of Buades, Coll, Morel, and nonlocal total variation of Gilboa, Osher), we extend the local Ambrosio-Tortorelli and Shah approximations to MS functional (MS) to novel nonlocal formulations, for better restoration of fine structures and texture. We present several applications of the proposed nonlocal MS regularizers in image processing such as color image denoising, color image deblurring in the presence of Gaussian or impulse noise, color image inpainting, color image super-resolution, and color filter array demosaicing. In all the applications, the proposed nonlocal regularizers produce superior results over the local ones, especially in image inpainting with large missing regions. We also prove several characterizations of minimizers based upon dual norm formulations.

Index Terms—Ambrosio-Tortorelli elliptic approximations, deblurring, demosaicing, denoising, impulse noise, inpainting, Mumford-Shah (MS) regularizer, nonlocal operators, super-resolution.

I. INTRODUCTION

We consider the degradation model of a color image

\[ f = Hu + n, \]

\[ (f^i = H^i u^i + n^i, i = R, G, B) \]  

(1)

where \( H \) is a linear operator accounting for some blurring, subsampling, or missing pixels (so that the observed data \( f \) loses some portion of the original image \( u \) we wish to recover), and \( n \) is additive noise. Problem (1) is highly ill-posed, thus, we formulate the restoration problem within the variational framework as:

\[ \inf_u \{ \Phi(f - Hu) + \Psi(\nabla u) \}, \]

where \( \Phi \) defines a data-fidelity term, and \( \Psi \) defines the regularization that enforces a smoothness constraint on \( u \), depending upon its gradient \( \nabla u \).

The regularization term \( \Psi \) alleviates the ill-posedness of the inverse problem by reflecting some a priori properties. Several edge-preserving regularization terms were suggested in the literature, including [2], [7], [14], [37], and [67]–[69]. These traditional regularization terms are based upon local image operators, which denoise and preserve edges and smooth regions very well, but may not deal well with fine structures like texture during the restoration process because texture is not local in nature.

In recent years, new image denoising models were proposed, based upon nonlocal image operators, to better deal with texture. Buades et al. [19] introduced the nonlocal means (NL-means) filter, producing excellent denoising and demosaicing results [21]. Kindermann et al. [52], and Gilboa-Osher [39], [40] formulated a variational framework of the NL-means filter by proposing nonlocal regularization functionals. Lou et al. [53] used the nonlocal total variation (NL-TV) of Gilboa-Osher in grey-scale image deblurring in the presence of Gaussian noise. Moreover, Peyré et al. [64] used the total variation on nonlocal graphs for grey-scale image inpainting, super-resolution of a single image, and compressive sensing. Zhang et al. [78] also applied NL/TV to compressive sensing, and Pross et al. [66] generalized the NL-means filter to super-resolution.

Prior work using nonlocal methods has been done for the Gaussian noise model, but no study has been developed for the impulse noise model using nonlocal information. However, the impulse noise model was studied in the local case. Bar et al. [7] used the Ambrosio-Tortorelli and Shah approximations to Mumford-Shah (MS) regularizing functional for color image deblurring in the presence of impulse noise, producing better restoration than by the total variation regularization; moreover, the edge set is detected concurrently in the restoration process. We propose in this paper nonlocal versions (NL/MS) of Ambrosio-Tortorelli [2] and Shah [69] approximations to the MS regularizer for the multichannel case. We also propose a) several applications of NL/MS to color image denoising, deblurring in the presence of Gaussian or impulse noise, inpainting with larger missing regions, super-resolution of a single image, color filter array demosaicing, and b) an efficient preprocessing step to compute the weights \( w \) in the deblurring-denosing model in the presence of impulse noise. In all these applications, we show that the proposed nonlocal regularizers produce superior results over the local ones. We mention that preliminary results of this work have been presented in conference proceedings [47], [49] and technical report [16]. The work in [16] proposes also non-local level set approximations of the Mumford and Shah functional for curve evolution and boundary detection.
II. BACKGROUND

1) Local Regularizers: We recall two MS regularizing functionals [3], [61], [69] and their elliptic approximations [2], [3], [69] that have been used in several segmentation and restoration algorithms. The MSH\textsuperscript{1} regularizer, depending upon the image \( u : \Omega \to \mathbb{R} \) and on its edge set \( K \subset \Omega \), giving preference to \( H^1 \)-convergence. The edge set \( E \) is the open image domain. The first term enforces smoothness of \( u \) everywhere except on the edge set \( K \), and the second one minimizes the total length of edges. It is difficult to minimize in practice the nonconvex MS functional.

There are several numerical approaches for minimizing the MS regularizing functional, one being the phase field approach using \( \Gamma \)-convergence [1], [2] (with applications to image deblurring and denoising [8], and to image inpainting [70]).

More specifically, Ambrosio and Tortorelli [2] approximated this functional by a sequence of regular functionals \( \psi_e \) using the \( \Gamma \)-convergence. The edge set \( K \) is represented by a smooth auxiliary function \( v \). Thus, we have an approximation to \( \Psi_{\text{MSH}}^{1} \) by [2]

\[
\Psi_{\text{MSH}}^{1}(u,v) = \beta \int_{\Omega} v^2 |\nabla u|^2 \, dx + \alpha \int_{\Omega} \mathcal{H}^1(\nabla u)
\]

where \( |\nabla u| = (1 + |\nabla u|^2)^{1/2} \) with \( x = (x_1, x_2) \). \( \mathcal{H}^1 \) is the 1-D Hausdorff measure, and \( \Omega \subset \mathbb{R}^2 \) is the open image domain. The first term enforces smoothness of \( u \) everywhere except on the edge set \( K \), and the second one minimizes the total length of edges. It is difficult to minimize in practice the nonconvex MS functional.

An alternative approach is the total variation regularization proposed in image restoration by Rudin, Osher, and Fatemi [67]: given a locally integrable function \( u \), define

\[
\Psi_{\text{TV}}(u) = \text{Supp} \left\{ \int_{\Omega} u \nabla \phi \, dx : \phi \in C^1_c(\Omega, \mathbb{R}), |\phi|_{L^\infty(\Omega)} \leq 1 \right\}
\]

which coincides with \( \int_{\Omega} |\nabla u| \, dx \) when \( u \in W^{1,1}(\Omega) \). Because of its benefit of preserving edges (which have high gradient levels) and convexity, the total variation has been widely used in image restoration.

Shah [69] suggested a modified version of the Ambrosio-Tortorelli approximation to the MS functional by replacing the quadratic term \( |\nabla u|^2 \) by \( |\nabla u| \) in the first term

\[
\Psi_{\text{MSH}}^{1}(u,v) = \beta \int_{\Omega} v^2 |\nabla u|^2 \, dx + \alpha \int_{\Omega} |\nabla u|^{\frac{3}{2}} \, dx.
\]

This functional \( \Gamma \)-converges to the \( \Psi_{\text{MSH}}^{1} \) functional [3]

\[
\Psi_{\text{MSH}}^{1}(u,v) = \beta \int_{\Omega \setminus K} |\nabla u| \, dx
\]

\[
+ \alpha \int_{K} \left| \frac{u^+ - u^-}{1 + |u^+ - u^-|} \right| \mathcal{H}^1 + |D_{\text{c}} u|(\Omega)
\]

where \( u^+ \) and \( u^- \) denote the image values on two sides of the jump set \( K = J_u \) of \( u \), and \( D_{\text{c}} u \) is the Cantor part of the measure-valued derivative \( D_u \). Note that the nonconvex term \( |u^+ - u^-|/1 + |u^+ - u^-| \) is similar with the prior regularization by Geman-Reynolds [37]. We observe that this regularizing functional is also similar with the total variation of \( u \in BV(\Omega) \) that can be written, for \( K = J_u \), as

\[
\Psi_{\text{TV}}(u) = \int_{\Omega \setminus K} |\nabla u| \, dx + \int_{K} |u^+ - u^-| \mathcal{H}^1 + |D_{\text{c}} u|(\Omega).
\]

By comparing the second terms in \( \Psi_{\text{TV}} \) and \( \Psi_{\text{MSH}}^{1} \), we see that the MSTV regularizer does not penalize the jump part as much as the TV regularizer does.

In the case of color images, Blomgren and Chan [14] presented a color TV regularization by coupling the channels

\[
\sqrt{\int |\nabla u^R|^2 + \int |\nabla u^G|^2 + \int |\nabla u^B|^2}.
\]

The natural generalization of TV regularization to color images with coupled channels takes the form [5], [14], [75]

\[
\Psi_{\text{TV}} = \int_{\Omega} \|\nabla u\| \, dx
\]

\[
= \int_{\Omega} \sqrt{\|\nabla u^R\|^2 + \|\nabla u^G\|^2 + \|\nabla u^B\|^2} \, dx
\]

which was analyzed in [17]. Both channel coupling regularizers yield similar experimental results.

Bar et al. [7] used the color versions of MS regularizers for color image deblurring-denosing, by replacing the scalar-valued version of \( |\nabla u| \) by the vector-valued version \( \|\nabla u\| \) defined previously, such that

\[
\Psi_{\text{MSH}}^{1}(u,v) = \beta \int_{\Omega} v^2 |\nabla u|^2 \, dx + \alpha \int_{\Omega} \left( |\nabla u|^2 + \frac{(v - 1)^2}{4\varepsilon} \right) \, dx.
\]

\[
\Psi_{\text{MSH}}^{1}(u,v) = \beta \int_{\Omega} v^2 |\nabla u|^2 \, dx + \alpha \int_{\Omega} \left( |\nabla u|^2 + \frac{(v - 1)^2}{4\varepsilon} \right) \, dx.
\]

Note that, in both MS regularizers, the scalar-valued edge map \( v \) is common for the three channels and provides the necessary coupling between colors. We wish to refer to [18] for another very interesting work on color image restoration by regularization. We note that curve evolution level set approximations to MS functional have also been used for image deblurring in [50] and [48].
2) Nonlocal Methods: Nonlocal methods in image processing have been explored in many papers because they are well adapted to texture denoising while the standard denoising models working with local image information seem to consider texture as noise, which results in losing texture. Nonlocal methods are generalized from the neighborhood filters (e.g., Yaroslavsky filter [77], bilateral filter [74], Susan filter [71]) and patch based methods [31], [76]. The idea of a neighborhood filter is to restore a pixel by averaging the values of neighboring pixels with a similar grey level value.

Buades et al. [19] generalized this idea by applying the patch-based method, and proposed the nonlocal means (or NL-means) filter for denoising a given noisy image \( f \), given by

\[
NLf(x) = \frac{1}{C(x)} \int_{\Omega} e^{-d_a(f(x), f(y))/h^2} f(y)dy, \quad \text{with} \quad d_a(f(x), f(y)) = \int_{R^2} G_{a}(t) ||f(x+t) - f(y+t)||^2 dt
\]

where \( f(y) \) is the given color at pixel \( y \), \( d_a \) is the patch distance, \( G_{a} \) is the Gaussian kernel with standard deviation \( a \) determining the patch size, \( C(x) = \int_{\Omega} e^{-d_a(f(x), f(y))/h^2} dy \) is the normalization factor, and \( h \) is the filtering parameter which corresponds to the noise level; usually we set it to be the standard deviation of the noise. The NL-means not only compares the (color) value at a single point but the geometrical configuration in a whole neighborhood (patch).

3) Nonlocal Regularizers: Nonlocal filtering can be understood as a quadratic regularization based upon a nonlocal graph, as detailed for instance in the geometric diffusion framework of Coifman et al. [27], which was applied to nonlocal image denoising by Szlam et al. [72]. Denoising using quadratic penalization on image graphs was studied by Gilboa and Osher for image restoration and segmentation [39]. These quadratic regularizations were extended to nonsmooth energies such as the total variation on graphs that was defined over the continuous domain by Gilboa et al. [38] and over the discrete domain by Zhou and Schölkopf [79], [80]. Elmoataz et al. [33] considered a larger class of nonsmooth energies involving a p-laplacian for \( p < 2 \). Peyré replaced these nonlinear flows on graphs by a non-iterative thresholding in a nonlocal spectral basis [63].

These graph-based regularizations were used to solve general inverse problems such as image deblurring [20], [52], inpainting of thin holes and removal of texture irregularities [40]. Moreover, Peyré et al. [64] extended the total variation on nonlocal graphs to solve arbitrary inverse problems such as inpainting, super-resolution and compressive sampling. The graph-based regularizations are adaptive since the graph depends upon the image, in other words, the graph is directly estimated from the measurements. Recently, for grey-scale image deblurring with Gaussian noise, Lou et al. [53] used a preprocessed image obtained by Wiener filter, instead of \( f \), in order to construct the weights.

Let us review nonlocal differential operators over graphs and convex nonlocal functionals proposed by Gilboa and Osher [40]. Let \( u: \Omega \rightarrow \mathbb{R} \) be a function, and \( w: \Omega \times \Omega \rightarrow \mathbb{R} \) be a nonnegative and symmetric weight function. The nonlocal gradient vector \( \nabla_w u: \Omega \times \Omega \rightarrow \mathbb{R} \) is defined as

\[
|\nabla_w u(x)| := \sqrt{\int_{\Omega} (u(y) - u(x))^2 w(x, y)dy}
\]

The nonlocal divergence \( \text{div}_w \nabla : \Omega \rightarrow \mathbb{R} \) of the vector \( \nabla \) : \( \Omega \times \Omega \rightarrow \mathbb{R} \) is defined as the adjoint of the nonlocal gradient

\[
(\text{div}_w \nabla)(v) := \int_{\Omega} (w(x, y) \cdot v(y, x)) \sqrt{w(x, y)}dy.
\]

Based upon these nonlocal operators, Gilboa and Osher introduced nonlocal regularizing functionals of the general form

\[
\Psi(u) = \int_{\Omega} \phi(|\nabla_w u|^2)dx
\]

where \( s \mapsto \phi(s) \) is a positive increasing function, convex in \( \sqrt{s} \), and \( \phi(0) = 0 \). By taking \( \phi(s) = \sqrt{s} \), they proposed the NL/TV regularizer

\[
\Psi^{NL/TV}(u) = \int_{\Omega} |\nabla_w u|dx
\]

which corresponds in the local 2-D case to \( \Psi^{TV}(u) = \int_{\Omega} |\nabla u|dx \).

Inspired by the previously mentioned work, we propose in the next section nonlocal versions of Ambrosio-Tortorelli and Shah approximations to the MS regularizer for color image restoration, such as deblurring-denoising in the presence of Gaussian noise or impulse noise, inpainting with large missing regions, superresolution of a single image, and image demosaicing, by extending the scalar nonlocal operators to the vector-valued case. This is also continuation or nonlocal extension of the work by Bar et al. [7], [8], first to propose the use of local MS-like approximations to color image deblurring-denoising in the presence of noise.

III. PROPOSED NONLOCAL MS REGULARIZERS

We propose the following nonlocal MS regularizers (NL/MS) by applying the nonlocal operators to the multichannel approximations of the MS regularizer

\[
\Psi^{NL/MS}(u, v) = \beta \int_{\Omega} v^2 \phi(|\nabla_w u|^2)dx + \alpha \int_{\Omega} \left( \epsilon |\nabla u|^2 + \frac{(v - 1)^2}{4\epsilon} \right)dx
\]
where \( u : \Omega \rightarrow \mathbb{R}^3, v : \Omega \rightarrow [0,1] \), \( \phi(s) = s \) or \( \phi(s) = \sqrt{s} \) correspond to the nonlocal versions of \( \text{MSH}^1 \) and \( \text{MSTV} \) regularizers, called here \( \text{NL/MSH}^1 \) and \( \text{NL/MSTV} \), respectively:

\[
\begin{align*}
\Psi_{\text{NL/MSH}}^1(u,v) &= \beta \int_{\Omega} u^2 ||\nabla u||^2 dx \\
&+ \alpha \int_{\Omega} \left( \varepsilon ||\nabla v||^2 + \frac{(v - 1)^2}{4\varepsilon} \right) dx
\end{align*}
\]

\[
\Psi_{\text{NL/MSTV}}(u,v) = \beta \int_{\Omega} |\nabla u||dx \\
+ \alpha \int_{\Omega} \left( \varepsilon ||\nabla v||^2 + \frac{(v - 1)^2}{4\varepsilon} \right) dx
\]

where \( ||\nabla u|| : \Omega \rightarrow \mathbb{R} \) is defined as

\[
||\nabla u||_p(x) := \sqrt{\sum_{i=R,G,B} |\nabla u^i||_p(x)}
\]

We apply these nonlocal MS regularizers to color image denoising, color image deblurring in the presence of Gaussian or impulse noise, color image inpainting, color image super-resolution, and to color filter array demosaicing, by incorporating proper fidelity terms. Furthermore, for deblurring in the presence of impulse noise, we propose a preprocessing step to evaluate the weights \( w \) based upon the preprocessed image. In practice, we use the standard weight function \( w \) at \( (x, y) \in \Omega \times \Omega \) depending upon an image \( q : \Omega \rightarrow \mathbb{R} \)

\[
w(x, y) = \exp \left( -\frac{d_x(q(x), q(y))}{\sigma^2} \right)
\]

which gives the similarity of the color values as well as of image features between two pixels \( x \) and \( y \) in the image \( q \) that will be defined in each section. For a fixed pixel \( x \in \Omega \), we use a search window \( S(x) = \{ y \in \Omega : |x - y| \leq r \} \) instead of \( \Omega \), to compute \( w(x, y) \).

Note that the nonlocal and nonconvex continuous models proposed in the following sections have not been analyzed theoretically; however, these formulations become well-defined in the discrete, finite differences case, but we prefer to present them in the continuous setting for simplicity.

IV. IMAGE RESTORATION WITH NL/MS REGULARIZERS

A. Color Image Deblurring and Denoising

Image blur and noise are the most common problems in photography, which can be especially significant in light limited situations, resulting in a ruined photograph. Image deblurring (or deconvolution) is the process of recovering a sharp image from an input image corrupted by blurring and noise, where the blurring is due to convolution with a known or unknown kernel; see [7], [9], [17], [26], [44]. Recently, new image denoising models [19], [20], [39], [40], [55], [63], [72], based upon nonlocal image information, have been developed to better restore texture. The standard linear degradation model for color image deblurring-denoising (or denoising) is

\[
f = k * u + n \quad (f^i = k * u^i + n^i, i = R, G, B)
\]

where \( k \) is a (known) space-invariant blurring kernel, and \( n \) is additive Gaussian noise, additive Laplace noise, or impulse noise (salt-and-pepper noise or random-valued impulse noise; in this case, the relation between \( k * u \) and \( n \) is no longer of the above form).

First, in the case of Gaussian noise model, the \( L^2 \)-fidelity term led by the maximum likelihood estimation is commonly used

\[
\Phi(f - k * u) = \int_{\Omega} ||f^i - k * u^i||^2 dx.
\]

However, the quadratic data fidelity term considers the impulse noise, which might be caused by bit errors in transmissions or wrong pixels, as an outlier. So, for the impulse noise model (or the additive Laplace noise model), the \( L^1 \)-fidelity term is more appropriate, due to its robustness of removing outlier effects [4], [62]; moreover, we consider the case of independent channels noise [7]

\[
\Phi(f - k * u) = \int_{\Omega} ||f^i - k * u^i|| dx.
\]

Thus, we propose two types of total energies for color image deblurring-denoising (to be minimized with respect to \( u \) and \( v \), depending upon the type of noise, as follows (Gau: Gaussian noise, Im: impulse noise))

\[
\begin{align*}
E_{\text{Gau}}^i(u,v) &= \frac{1}{2} \int_{\Omega} \sum_{i} ||f^i - k * u^i||^2 dx + \Psi_{\text{NL/MS}}(u,v) \\
E_{\text{Im}}^i(u,v) &= \int_{\Omega} \sum_{i} ||f^i - k * u^i|| dx + \Psi_{\text{NL/MS}}(u,v)
\end{align*}
\]

1) Preprocessing Step for the Impulse Noise Model: To extend the nonlocal methods to the impulse noise case, we need a preprocessing step for the weight function \( w \) since we cannot directly use the data \( f \) to compute \( w \). In other words, in the presence of impulse noise, the noisy pixels tend to have larger weights than the other neighboring points, so it is likely to keep the noise value at such pixel. Thus, we propose a simple algorithm to obtain a preprocessed image \( \tilde{g} \), which removes the impulse noise (outliers) as well as preserves the texture as much as we can. Basically, we use the median filter, well-known for removing impulse noise. However, for the deblurring-denoising model, if we apply one-step of the median filter, then the output may be too smoothed out. In order to preserve fine structures as well as to remove noise properly, we define a preprocessing method for the deblurring-denoising model, inspired by the idea of Bregman iteration [15]. Thus, we propose the following algorithm to obtain a preprocessed image \( \tilde{g} \) that will be used only in the computation of the weight function \( w \)

Initialize: \( r_0 = 0, \tilde{g}_0 = 0, i = R, G, B \)

\[
do \text{(iterate } m = 0, 1, 2, \ldots) \text{ }
\begin{align*}
g_{m+1} &= \text{median} \left( f + r_m \right) \\
r_{m+1} &= r_m + f - k * g_{m+1}
\end{align*}
\]

while \( \sum_i ||f^i - k * g_{m+1}^i||_1 > \sum_i ||f^i - k * g_{m+1}||_1 \)
above a certain threshold (usually related to the noise variance). This results in very weak connections (low weight values) for singular regions, thus such regions are essentially isolated from the rest of the image. This way rare features which also have a very large patch distance between them and any other patch in the image can be regularized as well. The image \( \tilde{g} \) will be used only in the computation of the weights \( \psi \), while keeping \( f \) in the data fidelity term, thus, artifacts are not introduced by the median filter. We note that if we use the recent work [22] as our preprocessing step, then we might obtain better construction of weights. However, our proposed algorithm is simple, fast, and satisfactorily to construct the weights.

2) Characterization of Minimizers: In this section, we characterize the minimizers of the functional formulated with the nonlocal regularizers, using [59], [73]. The details of the proofs are given in the appendices. Assuming that a functional \( \| \cdot \| \) on a subspace of \( (L^2(\Omega))^3 \) is a seminorm, we can define the dual norm (where \( \langle \cdot, \cdot \rangle \) denotes the \( (L^2(\Omega))^3 \) inner product) of \( f \in (L^2(\Omega))^3 \subset (L^1(\Omega))^3 \) as \( \| f \|_* \doteq \sup_{\| \varphi \| = 1} \langle f, \varphi \rangle / \| \varphi \| \leq +\infty \), so that the usual duality \( \langle f, \varphi \rangle \leq \| f \| \| \varphi \| \) holds for \( \| \varphi \| \neq 0 \). We define the functionals (here \( K_u \doteq k \ast u \))

\[
F(u) = \lambda \int_{\Omega} \sum_{i} |f^i - Ku|^2 dx + |u|_{NL/TV}
\]

\[
E(u, v) = \int_{\Omega} \sum_{i} \sqrt{|f^i - Ku|^2 + \eta^2} dx + \beta \| u \|_{NL/MS} + \alpha \int_{\Omega} \left( \eta |\nabla v|^2 + \frac{(v - 1)^2}{4\epsilon} \right) dx
\]

\[
E'(u, v) = \sum_{i} \int_{\Omega} \sqrt{|f^i - Ku|^2 + \eta^2} dx + \beta \| u \|_{NL/MS} + \alpha \int_{\Omega} \left( \eta |\nabla v|^2 + \frac{(v - 1)^2}{4\epsilon} \right) dx
\]

where \( \lambda > 0 \), and \( \| u \|_{NL/MS} \in \{ \| u \|_{NL/MS^{1,\epsilon}}, \| u \|_{NL/MS^{TV,\epsilon}} \} \) with

\[
|u|_{NL/TV} = \int_{\Omega} \| \nabla u \| |(x)| dx
\]

\[
|u|_{NL/MS^{1,\epsilon}} = \int_{\Omega} v^2(x) \| \nabla u \|^2(x) dx
\]

\[
|u|_{NL/MS^{TV,\epsilon}} = \int_{\Omega} v^2(x) \| \nabla u \| |(x)| dx.
\]

Note that the regularizers \( |u|_{NL/TV} \), \( |u|_{NL/MS^{1,\epsilon}} \), and \( |u|_{NL/MS^{TV,\epsilon}} \) are seminorms. In addition, we modified the regularizing functional \( |u|_{NL/MS^{1,\epsilon}} \), the square-root term replaces the original term

\[
\int_{\Omega} v^2(x) \| \nabla u \| |(x)| dx
\]

of our model. It is introduced here to enable the characterization of minimizers below, but the numerical calculations utilize the original formulations, which solve the same equivalent problems. The following characterizations of minimizers allow us to give conditions on the existence of minimizers (including the

---

**Fig. 1.** Preprocessed images \( \tilde{g} \) in the presence of Random-valued impulse noise. First–fourth columns: data \( f \), \( g_1 \), \( g_2 \), \( \| f - k \ast g_m \|_1 \) vs \( m \). Data \( f \): (top) motion blur kernel with length of 8 and orientation 0, noise density \( d = 0.1 \), (bottom) motion blur kernel with length of 4 and orientation 0, noise density \( d = 0.2 \). Preprocessed images: \( \tilde{g} = g_2 \) (top), \( \tilde{g} = g_1 \) (bottom) with 3 × 3 median filter.

**Fig. 2.** Preprocessed image using iterative median filter. Top: original image, image blurred with Gaussian blur kernel with \( \sigma_b = 2 \), blurry-noisy data \( f \) contaminated by Salt-and-Pepper noise with noise density \( d = 0.4 \). Bottom: (first) recovered image using one-step median filter of size 5 × 5, recovered image using iterative median filter of size (second) 5 × 5, (third and fourth columns) 7 × 7 and its corresponding plots of energies \( \| f - k \ast g_m \|_1 \) and \( \| g_m - g_{m+1} \|_1 \) vs \( m \).

where \( f \) is the given noisy-blurry data, and \( \text{median}(f_{[i,s,s]}^i) \) is the median filter of size \( s \times s \) with input \( f \). The residual energy \( \sum_i \| f^i - k \ast g_m^i \|_1 \) has a minimum value at the \( t \)th iteration, thus, we obtain a preprocessed image \( \tilde{g} = g_t \). We show in Figs. 1 and 2 the residual norms \( \sum_i \| f^i - k \ast g_m^i \|_1 \) versus steps \( m \). We also show in Fig. 2 the norm \( \| g_m + 1 - g_{m+1} \|_1 \), from which we can deduce that the sequence \( g_m \) does not converge to a limit. The preprocessed image \( \tilde{g} \) is a deburred and denoised version of \( f \), but it still includes some irregularities (or remaining impulse noise) that will be handled by constructing binary weights (values of 0 or 1) proposed for detecting and removing irregularities from texture in [40]; in the case where the weights are computed with a Gaussian-like formula, the weights decay fast for distances
Proposition 1: Let $K : (L^2(\Omega))^3 \rightarrow (L^2(\Omega))^2$ be a linear and continuous blurring operator with adjoint $K^*$ and let $F$ be the associated functional. Then:

1) $\|K^*f\|_* \leq 1/2\lambda$ if and only if $u = 0$ is a minimizer of $F$.

2) Assume that $1/2\lambda < \|K^*f\|_* < \infty$. Then $u$ is a minimizer of $F$ if and only if $\|K^*(f - Ku)\|_* = 1/2\lambda$ and $\langle u, K^*(f - Ku) \rangle = 1/2\lambda \|d_u \|_{NLTV}$;

where $\| \cdot \|_*$ is the corresponding dual norm of $\| \cdot \|_{NLTV}$.

We omit the proof of Proposition 1, since it is similar with proofs given in [59], [73].

Proposition 2: Let $K : (L^2(\Omega))^3 \rightarrow (L^2(\Omega))^2$ be a linear and continuous blurring operator with adjoint $K^*$ and let $E$ be the associated functional. If $(u, v)$ is a minimizer of $E$ with $v \in [0, 1]$, then

\[
\left\| K^* \frac{f - Ku}{\sqrt{\sum_i (f_i - Ku_i)^2 + \eta^2}} \right\|_* = \beta, \quad \text{and} \quad \left\langle K^* \frac{f - Ku}{\sqrt{\sum_i (f_i - Ku_i)^2 + \eta^2}}, u \right\rangle = \beta \|u\|_{NLTV}
\]

where $\| \cdot \|_*$ is the corresponding dual norm of $\| \cdot \|_{NLTV}$.

Note that, if we replace $E$ by $E'$ in Proposition 2, we obtain a similar result; the second formula in the conclusion must be replaced by

\[
\sum_i \left\langle K^* \frac{f_i - Ku_i}{\sqrt{(f_i - Ku_i)^2 + \eta^2}}, u_i \right\rangle = \beta \|u\|_{NLTV}
\]

where $\langle \cdot, \cdot \rangle$ is the $L^2(\Omega)$ inner product.

B. Color Image Inpainting

Image inpainting, also known as image interpolation, is the process of reconstructing lost or corrupted parts of an image. This is an interesting and important inverse problem with many applications such as removal of scratches in old photographs, removal of overlaid text or graphics, and filling-in missing blocks in unreliably transmitted images. Non texture image inpainting has received considerable interest since the pioneering paper by Masnou and Morel [57], [58], who proposed variational principles for image disocclusion. A recent wave of interest in inpainting has also started from Bertalmio et al. [11], where applications to the movie industry, video and art restoration were unified. These authors proposed nonlinear partial differential equations for non texture inpainting. Moreover, many contributed works have been proposed for the solution of this interpolation task based upon (a) diffusion and transport PDE/variational principle [6], [13], [23]–[25], [34], [64], [70], (b) exemplar region fill-in [12], [28], [30], [31], [65], [76], (c) compressive sensing [32]. Inpainting corresponds to the operation $H$ in (1) of losing pixels from an image, i.e., the observed data $f$ is given by

\[
f = u \quad \text{on} \quad \Omega - D
\]

where $D = D_0$ is the region where the input data $u$ has been damaged. Thus, inspired from [25], we propose the total energy functional for color image inpainting as

\[
E_{\text{inp}}(u, v) = \frac{\lambda}{2} \int_{\Omega} \chi_{\Omega - D}(x) \sum_i |f^i - u^i|^2 dx + \Psi_{NLTV}(u, v)
\]

where $\chi_{\Omega - D}$ is a characteristic function on $\Omega$ (i.e., $\chi_{\Omega - D}(x) = 1$ if $x \in \Omega - D$, 0 otherwise), and $\lambda > 0$ is a parameter. In addition, we update the weights $w$ only in the damaged region $D$ at every $m$th iteration for $u$ using the patch distance

\[
d_u^m(u(x), u(y)) = \int_{\mathbb{R}^2} \chi_{\Omega - R}(x + t)G_0(t)||u(x + t) - u(y + t)||^2 dt
\]

where $\chi_{\Omega - R}$ is a characteristic function on $\Omega$ defined previously, and $R \subset D$ is an un-recovered region (still missing region). Therefore, the missing region $D = D_0$ is recovered by the following iterative algorithm, producing the un-recovered regions $D^i, \ i = 0, 1, 2, \ldots, \ D_0 \supset D_1 \supset D_2 \supset \cdots$:

1) Compute weights $w$ for $x \in \Omega$ s.t. $P(x) \cap (\Omega - D_0) \neq \emptyset$ using $d_u^m(u_0(x), u_0(y))$ with $u_0 = f$ in $\Omega - D_0$ and in $D_0$, a patch $P(x)$ centered at $x$, and $y \in S(x) \cap (\Omega - D_0)$.

2) Iterate $n = 1, 2, \ldots$ to obtain a minimizer $(u, v)$ starting with $u = v_0$:

a) For fixed $u$, update $v$ in $\Omega$ to obtain $v^n$.

b) For fixed $v$, update $u$ in $\Omega$ to obtain $u^n$ with a recovered region $\Omega - D^n \supset \Omega - D^{n-1}$: at every $m$th iteration, update weights $w$ only in $D \cap D_0$ s.t. $P(x) \cap (\Omega - D_0)\neq \emptyset$ with

\[
d_u^{n,m}(u(x), u(y)) = \int_{\mathbb{R}^2} \chi_{\Omega - R}(x + t)G_0(t)||u(x + t) - u(y + t)||^2 dt
\]

where $y \in S(x) \cap (\Omega - D^{n,m})$. $D^{n,m}$ is an un-recovered region in $D_0$ until $m$th iteration with $D^{n,2m} \supset \cdots \supset D^{n,m} = D^n$.

C. Color Image Super-Resolution

Super-resolution corresponds to the recovery of a high resolution image from a filtered and down-sampled image. It is usually applied to a sequence of images in video; see [29], [35], [36], [56], [66]. We consider here a simpler problem of increasing the resolution of a single still image, and the observed data $f$ is given by

\[
f^i = D_k(h \ast u^i), \quad i \in R_q, G, B
\]

where $h$ is a low-pass filter, $D_k : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{p \times p}$ (with $p = \lceil n/k \rceil$ where $[q]$ is the integer part of $q$) is the down-sampling operator by factor $k$ along each axis. We want to recover a high resolution image $u \in (\mathbb{R}^{p \times p})^3$ by minimizing

\[
E_{\text{sup}}(u, v) = \frac{\lambda}{2} \int_{\Omega} \sum_i |f^i - D_k(h \ast u^i)|^2 dx + \Psi_{NLTV}(u, v)
\]

In addition, we use a super-resolved image $\tilde{f} \in (\mathbb{R}^{q \times q})^3$ obtained by bicubic interpolation of $f \in (\mathbb{R}^{p \times p})^3$ only for the computation of the weights $w$. 

1588 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 20, NO. 6, JUNE 2011
D. Color Filter Array Demosaicing

In a demosaicing algorithm we have to reconstruct a full color image from the incomplete color samples output from an image sensor overlaid with a color filter array (CFA). A color filter array is a mosaic of color filters in front of the image sensor, and we use here the Bayer filter [10] that has alternating green ($G$) and red ($R$) filters for odd rows and alternating blue ($B$) and green ($G$) filters for even rows. Since each pixel of the sensor is behind a color filter, the output is an array of pixel values, each indicating a raw intensity of one of the three color filters. Thus, an algorithm is needed to estimate for each pixel the color levels for all color components, rather than a single component; see [21], [41]–[43], [45], [51], [54]. In this variational framework, we consider the observed data $f_i$ as

$$f_i = H^i \cdot u^i, \quad i \in \{R, G, B\}$$

where $\cdot$ is the pointwise product, and $H^i$ is the down-sampling operator; $H^R$ has alternating 0 and 1 values for odd rows and alternating 0 and 1 values for even rows, $H^G$ has alternating 0 and 1 values for odd rows and only 0 values for even rows, $H^B$ has only 0 values for odd rows and alternating 1 and 0 values for even rows. We propose the following minimization problem to recover a full color image $u$

$$E^{\text{Demo}}(u, v) = \frac{1}{2} \int_\Omega \sum_i |f_i - H^i \cdot u|^2 dx + \Psi^{\text{NL/MS}}(u, v).$$

(6)

Moreover, we use the interpolated image obtained by applying Hamilton-Adams algorithm [43] for the green channel and bilinear interpolation for $R \rightarrow G$ and $B \rightarrow G$, to compute the initial weight function $\hat{w}$. In the Hamilton-Adams method, the evaluation of the gradient at the missing green pixel is corrected by the second-order derivatives of the red or blue channels. In addition, as in [21], in order to gradually correct the erroneous structures and artifacts of the initial color image $u_0$, we also proceed by an iterative strategy, refining at each step the similarity search by reducing the value of parameter $h$ in the weights as:

- Initialize $u = u_0$ with an interpolated image with Hamilton-Adams algorithm.
- Iterate for $h$ (e.g., $h \in \{16, 8, 4\}$):
  a) Construct the weight function $w = w(u)$ using the image $u$.
  b) Compute a minimizer $(u, v)$ by minimizing the functional $E^{\text{Demo}}(u, v)$.

We refer to [21] using NL-means for prior relevant work that inspired us in this application.

V. NUMERICAL DISCRETIZATIONS

Minimization of the proposed functionals (2)–(6): $E^{\text{Gau}}$, $E^{\text{Im}}$, $E^{\text{Imp}}$, $E^{\text{Sup}}$, $E^{\text{Demo}}$ in $u$ and $v$ is carried out using the Euler–Lagrange equations

$$\frac{\partial E^{\text{Gau}, \text{Im}, \text{Imp}, \text{Sup}, \text{Demo}}}{\partial u} = \frac{\partial}{\partial u} \left( 2\beta \psi(\|\nabla u\|^2) - 2\kappa \Delta u + \alpha \left( \frac{u - 1}{2\kappa} \right) \right) = 0$$

$$\frac{\partial E^{\text{Gau}, \text{Im}, \text{Imp}, \text{Sup}, \text{Demo}}}{\partial v} = \frac{\partial}{\partial v} \left( 2\beta \psi(\|\nabla w\|^2) - 2\kappa \Delta v + \alpha \left( \frac{v - 1}{2\kappa} \right) \right) = 0$$

where $\hat{h}(x) = h(-x)$, $\hat{h}(x) = h(-x)$, $D_k^T : (\mathbb{R}^{\mathbb{A} \times n})^3 \rightarrow (\mathbb{R}^{\mathbb{A} \times n})^3$ is the transpose of $D_k$ i.e., the up-sampling operator, and

$$L^{\text{NL/MS}}(u) = -2 \int_\Omega \left\{ (w(y) - w(x))w(x, y) \cdot \left[ (\psi'(y) + \psi'(\|\nabla u\|^2)) + (\psi'(y)) + \psi'(\|\nabla u\|^2)) \right] \right\} dy.$$ 

To solve two Euler–Lagrange equations simultaneously, the alternate minimization approach is applied. Note that since the energy functionals are not convex in the joint variable $(u, v)$, we may compute only a local minimizer. In our algorithm, we define the initial guess $u^0$ to be the data $f$ (except for the super-resolution problem). As suggested by a referee, we have also tested other initial conditions ($u^0$ a constant, a random image, or the data $f$ perturbed by a random component; $u^0$ constant equal to 0 or 1); we have observed that the final steady state result does not change, only the number of iterations needed to reach the same result changes; we noticed that, if the initial image $u^0$ is the input data $f$, then fewer iterations are needed to reach the steady state. Due to its simplicity, we use Gauss-Seidel scheme for $v$, and an explicit scheme for $u$ by gradient descent method, leading to the following iterative algorithm:

1. Initialization: $u^0 \equiv f$, $v^0 = 1$.
2. Iterate $n = 0, 1, 2, \ldots$, until $(\|u^{n+1} - u^n\|_2 < \eta \|u^n\|_2$).

1) Solve the equation for $v^{n+1}$ using Gauss-Seidel scheme:

$$\left( 2\beta \psi(\|\nabla w\|^2) + \alpha \frac{\kappa}{2\kappa} - 2\kappa \Delta \right) \psi^{n+1} = \frac{\alpha}{2\kappa}.$$

2) Set $u^{n+1} = u^n$ and solve for $u^{n+1}$ by iterating on $l$:

$$u^{n+1, l+1} = u^{n+1, l} - d_l \frac{\partial E}{\partial u}(u^{n+1, l}, v^{n+1}).$$

Here $\eta$ is a small positive constant. The basic discretizations are explained next. Let $h_k$ denote the value of a pixel $k$ in the image $(1 \leq k \leq N)$ with channel $i$ (i.e., the discretized version of $w(x)$ defined on $\Omega$, and let $p_{k,l}$ be the discretized version of $p_{k,l}(x, y)$ with $x, y \in \Omega$. Also, $u_{k,l}$ is the sparsely discrete version of $w = w(x, y) : \Omega \times \Omega \rightarrow \mathbb{R}$. We use the neighbors set $l \in N_k$ defined by $l \in N_k := \{ : u_{k,l} > 0\}$. Then we have $\nabla_{ud}$ and $\nabla_{vid}$, the discretizations of $\nabla_{u}$ and $\nabla_{v}$, respectively given by [40]

$$\nabla_{ud}(u_k) := (u_l - u_k) / w_k, \quad l \in N_k$$

$$\nabla_{vid}(p_{k,l}) := \sum_{l \in N_k} (p_{k,l} - p_{k,l}) / \sqrt{w_k l}.$$
TABLE I
PSNR VALUES OF RECOVERED LENA IMAGES IN FIG. 5
WITH SALT-AND-PEPPER NOISE

| noisy density | d = 0.3 | d = 0.4 | d = 0.5 |
|---------------|---------|---------|---------|
| data f        | 10.3172 | 9.1047  | 8.1583  |
| preprocessed g| 26.1987 | 25.7309 | 24.6567 |
| TV            | 28.4720 | 28.0494 | 27.2286 |
| MSH¹          | 28.4402 | 28.1533 | 27.1543 |
| MSTV          | 28.8540 | 28.4862 | 27.5305 |
| NL/TV         | 28.7825 | 28.5808 | 27.7406 |
| NL/MSH¹       | 28.7334 | 28.2866 | 27.4064 |
| NL/MSTV       | 29.4641 | 28.7977 | 27.9640 |

Moreover, the magnitude of \( p_{k,l} \) at \( k \) is \(|p_k|_k = \sqrt{\sum_i(p_{k,i})^2}\), thus, the discretization of \( ||\nabla u_{tar}||^2 \) is done as

\[
||\nabla u_{tar}||^2_k = \sum_{i \in \{R(x,B)\}} |\nabla u_{tar}^i|^2_k = \sum_i \sum_{l \in N_k} (u^i_l - u^{\tilde{l}}_l)^2 w_{k,l}.\]

Basically, we construct the weight function \( w_{k,l} \), following the algorithm in [39]: for each pixel \( k \), (1) take a patch \( P_k \) around a pixel \( k \), compute the distances \((d_k)_{k,l}\) (a discretization of \( d_a \)) to all the patches \( P_l \) in the search window \( l \in S(k) \), and construct the neighbors set \( N_k \) by taking the \( m \) most similar and the four nearest neighbors of the pixel \( k \), (2) compute the weights \( w_{k,l} \), \( l \in N_k \) and set to zero for all the other connections \((w_{k,l} = 0, l \notin N_k)\), (3) set \( u_{k,l} = u_k, l \in N_k \). For deblurring in the presence of impulse noise, we used \( m = 5 \), so a maximum of up to \( 2m + 4 \) neighbors for each pixel is allowed, and \( 5 \times 5 \) pixel patches with \( a = 10 \), a search window of size \( 11 \times 11 \). The complexity of computing the weights using this algorithm is \( N \times \text{WindowSize} \times (\text{Patchsize} \times \text{Channelsize} + \log m) \). Thus, in this case, we need \( 121 \times (25 \times 3 + 2.5) \approx 9619 \) operations per pixel. Note that, when we use a preprocessed image \( g \) to compute \( u \) in the impulse noise model, we construct the weight function \( w \) with the binary values of 0 or 1 [40]; in step (2) above, for \( l \in N_k \), we assign the value 1 to \( w_{k,l} \) and \( w_{l,k} \).

VI. EXPERIMENTAL RESULTS AND COMPARISONS

The nonlocal MS regularizers proposed here, NL/MSH¹ and NL/MSTV, are tested on several color images corrupted by different blur kernels or different types of noise, on color images with missing regions, on subsampled color images, as well as on incomplete color samples outputs. We mostly compare them with their local versions [7]. For deblurring in the presence of impulse noise, we also present results with TV [17] and NL/TV [53] models because our work is the first trial to extend nonlocal methods to impulse noise model. For denoising or deblurring in the presence of Gaussian noise and super-resolution, we only present the results of MSTV and NL/MSTV because NL/MSH¹ produces smoother images leading to similar PSNR values with MSTV. For more extensive results and comparisons obtained by other competitive methods, we would like to refer the reader to the webpage and work of Peyman Milanfar et al. [60] on image restoration.

First, in Figs. 1 and 3, we recover blurred images contaminated by random-valued impulse noise with noise density \( d = 0.1 \) or \( d = 0.2 \). Fig. 1 presents the preprocessed images \( g \) obtained by iterative median filter of size \( 3 \times 3 \), and the corresponding residual energies \( ||f - k \ast g_m||_1 \) vs \( m \). As \( m \) increases, the image \( g_m \) gets deblurred to some extent where the residual energy has a minimum. Thus, the image having the minimum energy value is chosen as a preprocessed image \( g \), but this still contains some impulse noise or artifacts. However, the recovered images using nonlocal regularizers in Fig. 3 show that the artifacts on the preprocessed images are well handled when constructing weights, thus, these do not influence the final recovered images. Hence, by computing the weight function \( u \) based upon the preprocessed images \( g \), all nonlocal regularizers recover texture better and reduce the artifacts by blur kernel (especially on the face and hand), providing cleaner images as well as higher PSNR values (as seen in Table II). Fig. 4 provides the edge set \( v \) of local or nonlocal MS regularizers, concurrently obtained during the restoration process in Fig. 3.

In Figs. 2 and 5, we recover blurred images contaminated by salt-and-pepper noise with various noise densities \( d = 0.3, 0.4, 0.5 \). In Fig. 2, we present the preprocessed images \( g \) obtained by iterative median filter with different size, and the image obtained by one-step median filter for comparison. First, we observe that the images using iterative median filters are deblurred and denoised versions of noisy-blurry data. For the case of \( d = 0.4 \), we choose the preprocessed image with iterative median filter of size \( 7 \times 7 \); the one with smaller median filter (\( 5 \times 5 \))
produces more severe artifacts (that can influence the weights) tending to remain in the final recovered images. We show the recovered images using local and nonlocal regularizers in Fig. 5 and the corresponding PSNR values in Table I. Visually and according to PSNR values, the nonlocal regularizers recover the degraded images better than the local ones. Specifically, NL/MSH\(^1\) reduces ringing artifacts appeared in the recovered images with MSH\(^1\); NL/MSTV and NL/TV reduce the staircase effect appeared in the results obtained by the corresponding local models, and suppress less the texture. Furthermore, we observe that the (local or nonlocal) MSTV regularizers are more robust than the (local or nonlocal) TV regularizers, resulting in clearer restoration, which might be caused by the fact that these do not penalize the edge part as much as TV regularizers do, and additionally yielding image edge sets \(\psi\). Comparing with the (local or nonlocal) MSH\(^1\) regularizers, the (local or nonlocal) MSTV regularizers produce cleaner images, despite the cartoon-like restored images especially in the case of high noise density.

Moreover, in Fig. 6, we use the “Girl” image corrupted by random-valued impulse noise; either high blur and noise with \(d = 0.3\) (second, third rows) or low blur and higher noise with \(d = 0.4\) (third, fourth rows). In the first case, NL/MSH\(^1\) reduces very much the ringing effect (especially appeared on the cloth part with MSH\(^1\)), providing cleaner image, and NL/TV gives much better restored image than by TV, leading to much

**TABLE II**

| Image | Fig. 3 | Fig. 6 |
|-------|--------|--------|
| Data  | \(l = 8\), \(d = 0.1\) | \(l = 4\), \(d = 0.2\) | \(r = 5\), \(d = 0.3\) | \(r = 3\), \(d = 0.4\) |
| TV    | 20     | 25     | 0.4    | 0.09   | 0.065  | 0.04   |
| MSH\(^1\) | 20     | 20     | 0.17   | 0.07   | 0.04   | 0.02   |
| MSTV  | 13     | 23     | 0.2    | 0.08   | 0.07   | 0.04   |
| NL/TV | 5      | 12     | 0.8    | 0.2    | 0.08   | 0.08   |
| NL/MSH\(^1\) | 14     | 33     | 0.3    | 0.12   | 0.05   | 0.03   |
| NL/MSTV | 2.5    | 5      | 5      | 2      | 0.4    | 0.18   |

**TABLE III**

| Parameters Selections (\(\lambda, \beta\)) for Figs. 3, 5, and 6 |
|------------------|------------------|------------------|------------------|
| TV               | NLTV             | MSH\(^1\)        | NL/MSH\(^1\)     | MSTV            | NL/MSTV         |
| Barbara          | 17.5             | 25               | 0.4              | 0.09             | 0.065           | 0.04             |
| Lena             | 20               | 30               | 0.17             | 0.07             | 0.04             | 0.02             |
| Girl             | 13               | 23               | 0.2              | 0.08             | 0.07             | 0.04             |
| 11×11 (fourth and fifth rows) | 5      | 12     | 0.8    | 0.2    | 0.08   | 0.08   |
| (third, fourth rows) | 14     | 33     | 0.3    | 0.12   | 0.05   | 0.03   |
| (second and third rows) | 2.5    | 5      | 5      | 2      | 0.4    | 0.18   |

Fig. 5. DEBLURRING IN THE PRESENCE OF SALT-AND-PEPPER NOISE. Data \(f\): Gaussian blur kernel with \(\sigma_b = 2\), noise density \(d = 0.3\) (Top two rows), \(d = 0.4\) (Middle two rows), \(d = 0.5\) (Bottom two rows). First column: (top) data \(f\), (bottom) preprocessed image \(g\). Second–fourth columns: recovered images using (top) local regularizers (MSH\(^1\), MSTV, TV) and (bottom) nonlocal regularizers (NL/MSH\(^1\), NL/MSTV, NL/TV).

Fig. 6. DEBLURRING IN THE PRESENCE OF RANDOM-VALUED IMPULSE NOISE. (First row-a): original image. Data \(f\): (second and third rows) out of focus blur kernel with radius \(r = 3\) (first row-b) and noise density \(d = 0.3\), (fourth and fifth rows) out of focus blur kernel with radius \(r = 5\) (first row-c) and noise density \(d = 0.4\). First column: (top) data \(f\), (bottom) preprocessed image \(g\) with 9×9 (second and third rows) and 11×11 (fourth and fifth rows) median filters. Second–fourth columns: recovered images using (top) local regularizers (MSH\(^1\), MSTV, TV) and (bottom) nonlocal regularizers (NL/MSH\(^1\), NL/MSTV, NL/TV).
higher PSNR (as seen in Table II). Even though MSTV gives desired recovered image already, NL/MSTV additionally reduces the staircase effect (seen on the image obtained with MSTV), resulting in more realistic image as well as higher PSNR. Both NL/MSTV and NLTV give very well recovered images visually and according to PSNR values. In the second case with less blur but more noise, NL/MSTV and NLTV give sharper images than by corresponding local ones and the MSH\(^1\) regularizers.

In Fig. 8, we only test MSTV and NL/MSTV models for the Gaussian noise model. For the noisy Lena image corrupted by Gaussian noise with variance \(\sigma_n = 0.02\), NL/MSTV recovers texture much better (hat part) and provides cleaner edges, while MSTV smoothes out many details and has noisy edges. In addition, for the noisy-blurry castle image, NL/MSTV gives cleaner and sharper restored images, leading to higher PSNR values.

In Figs. 9 and 10, we use the NL/MS regularizers to recover images with texture and large missing regions. In Fig. 9, we present the process of inpainting, and final recovered images using NL/MS regularizers. We can easily see that both nonlocal regularizers recover the missing regions very well, and moreover NL/MSTV gives slightly better result than NL/MSH\(^1\) according to PSNR even though these visually seem to produce very similar results. However, in Fig. 10 with a real image, NL/MSH\(^1\) gives slightly higher PSNR values, especially recovering better the part damaged by the rectangle in the bottom. Both NL/MS regularizers gradually recover the missing regions, as seen in Fig. 10 bottom row, while local MS regularizers fail to recover them.

In Figs. 11 and 12, we recover an image filtered with a low-pass filter and then subsampled, using MSTV and NL/MSTV.
It is common in practice to work with a 9-patch for deblurring-denoising, and a 21-patch for the deblurring-denoising model. The PSNR values were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$.

Finally, we note that the parameters $\alpha$, $\beta$ and $\epsilon$ were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$.

Finally, we note that the parameters $\alpha$, $\beta$ and $\epsilon$ were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$.

Finally, we note that the parameters $\alpha$, $\beta$ and $\epsilon$ were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$.

Finally, we note that the parameters $\alpha$, $\beta$ and $\epsilon$ were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$.

Finally, we note that the parameters $\alpha$, $\beta$ and $\epsilon$ were selected manually to provide the best PSNR results. The smoothness parameter $\beta$ increases with the noise level, while the other parameters $\alpha$, $\epsilon$ are approximately fixed, $\alpha = 0.001$, $\epsilon = 0.00000001$ for deblurring-denoising, inpainting, super-resolution, and $\alpha = 0.1$, $\epsilon = 0.001$ for denoising (although in theory $\epsilon \to 0$, it is common in practice to work with a small fixed $\epsilon$). For the weights $\psi$, we use $11 \times 11$ search window with a $5 \times 5$ patch for the deblurring-denoising model (or denoising), $21 \times 21$ search window with a $9 \times 9$ patch for super-resolution, and $15 \times 15$ search window with a $3 \times 3$ patch for denoising, while larger search windows are needed for inpainting. For the computational time, it takes about 5 minutes for constructing the weight function of a 256 x 256 image with the initial $u_0$, but in the case where the artifacts in $u_0$ are severe such as the Fence and the Roof images, these do not look different from the initial $u_0$. 
the 11 × 11 search window and 5 × 5 patch in MATLAB on a
dual core laptop with 2 GHz processor and 2 GB memory. The
minimization for the (local or nonlocal) MS regularizers in the
deblurring-denoising model takes about 150 s for the computa-
tions of both u using an explicit scheme based upon the gradient
descent method and v using a semiimplicit scheme with the total
iterations 5 × (2 + 100) (without including the computation of the
weight function w(x, y)). For the inpainting model with
150 × 150 size image, it takes about 20 minutes with total itera-
tion numbers 5 × (2 + 100) since we update the weight function
at every 50th iteration for u. For both super-resolution and
demosaicing, 10 × (2 + 200) iteration numbers are needed for all
regularizers.

VII. SUMMARY AND CONCLUSIONS

The proposed nonlocal MS regularizers, NL/MSH\textsuperscript{1} and
NL/MSTV, outperform the local ones on all the applications.
For deblurring in the presence of impulse noise, the nonlocal
regularizers (including NL/TV) incorporating an efficient pre-
processing step perform very well and provide better recovered
images than by the local ones; NL/MSH\textsuperscript{1} reduces ringing
artifacts appeared in MSH\textsuperscript{1}, and both NL/MSTV and NL/TV
reduce the staircase effect appeared in images obtained by
local models, resulting in more realistic images, and better
recovery of details. For denoising or deblurring in the presence
of Gaussian noise, NL/MSTV recovers fine scales better, and
gives cleaner and shaper images than by local regularizers or by
NL/MSH\textsuperscript{1}. Even for super-resolution, NL/MSTV provides the
cleanest and sharpest results, and for color filter array demos-
aicing, NL/MSTV reconstructs images best. For inpainting,
both NL/MSH\textsuperscript{1} and NL/MSTV provide superior results to those by local models by better recovering texture and large
missing regions. To sum up, in all the experiments (except
inpainting), NL/MSTV produces superior results to local regu-
larizers and to NL/MSH\textsuperscript{1}. Moreover, NL/MSTV provides edge
sets concurrently obtained in the restoration process, which is
another stronger point over the NL/TV model, while NL/TV
model is computationally faster.

APPENDIX A

We show that the following regularizing functionals are semi-
norms, a necessary step in the proofs of Propositions 1 and 2. We let

\[ |u|_{NLTV} = \int_\Omega \sqrt{\sum_i \int_\Omega (\hat{u}_i(y) - \hat{u}_i(x))^2 w(x,y)dydx} \]

\[ |u|_{NL/MSTV^\nu} = \int_\Omega \sqrt{\sum_i \int_\Omega (\hat{u}(y) - \hat{u}(x))^2 w(x,y)dydx} \]

\[ |u|_{NL/MSH^\nu} = \int_\Omega \sqrt{\sum_i \int_\Omega (\hat{u}_i(y) - \hat{u}_i(x))^2 w(x,y)dydx} \]

with \( u : \Omega \rightarrow \mathbb{R}^3 \), \( v : \Omega \rightarrow \mathbb{R} \) and \( w : \Omega \times \Omega \rightarrow \mathbb{R} \) is nonnegative
and symmetric. We only need to show that these functionals
satisfy the triangle inequality.

Define

\[ |u| = \int_\Omega g(x) \left( \sqrt{\sum_i \int_\Omega (\hat{u}_i(y) - \hat{u}_i(x))^2 w(x,y)dydx} \right) \]

for any nonnegative function \( g : \Omega \rightarrow \mathbb{R} \), and show that \( |u+v| \leq |u| + |v| \) for \( u, v : \Omega \rightarrow \mathbb{R}^3 \). First, we have the equality

\[ \sum_i \int_\Omega (\hat{u}_i^+ + \hat{v}_i)(y) - (\hat{u}_i^+ + \hat{v}_i^+)(x))^2 w(x,y)dydx \]

\[ = \sum_i \int_\Omega (\hat{u}_i^+(y) - \hat{u}_i^+(x))^2 w(x,y)dydx + \sum_i \int_\Omega (\hat{v}_i^+(y) - \hat{v}_i^+(x))^2 w(x,y)dydx + 2 \sum_i \int_\Omega (\hat{v}_i^+(y) - \hat{u}_i^+(x)) \cdot (\hat{v}_i^+(y) - \hat{v}_i^+(x)) w(x,y)dydx. \]

Using Cauchy-Schwarz inequality, we have

\[ \int_\Omega (\hat{u}_i^+(y) - \hat{u}_i^+(x))(\hat{v}_i^+(y) - \hat{v}_i^+(x)) w(x,y)dydx \]

\[ \leq \left( \int_\Omega (\hat{u}_i^+(y) - \hat{u}_i^+(x))^2 w(x,y)dydx \right)^{1/2} \cdot \left( \int_\Omega (\hat{v}_i^+(y) - \hat{v}_i^+(x))^2 w(x,y)dydx \right)^{1/2}. \]

Denote

\[ a^i = \int_\Omega (\hat{u}_i^+(y) - \hat{u}_i^+(x))^2 w(x,y)dydx \]

\[ b^i = \int_\Omega (\hat{v}_i^+(y) - \hat{v}_i^+(x))^2 w(x,y)dydx. \]

Using \( \sum_i \sqrt{a^i \cdot b^i} \leq \sqrt{\sum_i a^i} \cdot \sqrt{\sum_i b^i} \), we obtain

\[ \sum_i \int_\Omega (\hat{u}_i^+ + \hat{v}_i)(y) - (\hat{u}_i^+ + \hat{v}_i^+)(x))^2 w(x,y)dydx \]

\[ \leq \sum_i (a^i + b^i + 2\sqrt{a^i b^i}) \]

\[ = \sum_i a^i + \sum_i b^i + 2 \sum_i \sqrt{a^i} \cdot \sqrt{b^i} \]

\[ \leq \sum_i a^i + \sum_i b^i + 2 \left( \sum_i a^i \cdot \sum_i b^i \right) \]

\[ = \left( \sqrt{\sum_i a^i} + \sqrt{\sum_i b^i} \right)^2. \]
which finally leads to

\[
\sqrt{\sum_i \int_{\Omega} (u^i + \phi^i)(y) - (u^i + \phi^i)(x))^2 w(x, y) dy} \leq \sqrt{\sum_i \int_{\Omega} (u^i(y) - u^i(x))^2 w(x, y) dy} + \sqrt{\sum_i \int_{\Omega} (\phi^i(y) - \phi^i(x))^2 w(x, y) dy}.
\]

Multiplying by \(g(x)\) and integrating both sides w.r.t \(x\), we obtain

\[
\int_{\Omega} g(x) \sqrt{\sum_i \int_{\Omega} (u^i + \phi^i)(y) - (u^i + \phi^i)(x))^2 w(x, y) dy dx} \leq \int_{\Omega} g(x) \sqrt{\sum_i \int_{\Omega} (u^i(y) - u^i(x))^2 w(x, y) dy dx} + \int_{\Omega} g(x) \sqrt{\sum_i \int_{\Omega} (\phi^i(y) - \phi^i(x))^2 w(x, y) dy dx}.
\]

Thus, \([u]\) satisfies the triangle inequality, so we conclude that \([u]\) is a seminorm. Specifically, by taking \(g(x) = 1\) or \(g(x) = v^2(x)\), \([u]_{\text{NL/TV}}\) and \([u]_{\text{NL/MSTV,v}}\) are seminorms.

Similarly, we can also show that \([u]_{\text{NL/MSH,v}}\) is a seminorm using Cauchy-Schwarz inequality and the inequality

\[
\sum_i \sqrt{a^i} \sqrt{b^i} \leq \sqrt{\sum_i a^i \sum_i b^i}.
\]

Using Cauchy-Schwarz inequality

\[
\int_{\Omega} \int_{\Omega} v^2(x)(u^i(y) - u^i(x))(\phi^i(y) - \phi^i(x)) w(x, y) dy dx 
\leq \sum_i \int_{\Omega} \int_{\Omega} v^2(x)(u^i(y) - u^i(x))^2 w(x, y) dy dx 
+ \sum_i \int_{\Omega} \int_{\Omega} v^2(x)(\phi^i(y) - \phi^i(x))^2 w(x, y) dy dx 
+ 2 \sum_i \int_{\Omega} \int_{\Omega} v^2(x)(u^i(y) - u^i(x))(\phi^i(y) - \phi^i(x)) w(x, y) dy dx.
\]

Let

\[
g(\epsilon) := \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}.
\]

Taylor’s expansion gives

\[
g(\epsilon) = \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2 - \epsilon(f - Ku) \cdot (K\phi) / \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} + \epsilon^2 / 2g(\epsilon) \text{ and, hence}
\]

\[
\int_{\Omega} \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2} dx 
\leq \int_{\Omega} \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2} dx 
- \epsilon \left( \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , K\phi \right) 
+ \frac{\epsilon^2}{2} \max_x |g'(x)|.
\]

Then, the first inequality implies that

\[
\epsilon \left( \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , K\phi \right) 
\leq \epsilon \beta \phi_{\text{NL/MS}} + \frac{\epsilon^2}{2} \max_x |g'(x)|.
\]

Dividing by \(\epsilon > 0\) and letting \(\epsilon \downarrow 0_+\) (while noticing that \(\lim_{\epsilon \to 0_+} \epsilon^2 / 2 \max_x |g'(x)| = 0\)) yields that for any \(\phi \in \text{NL/MS}(\Omega)\),

\[
\left( K^* \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , \phi \right) \leq \beta \phi_{\text{NL/MS}}.
\]

**APPENDIX B**

**Proof of Proposition 2:** Let \([u, v]\) be a minimizing pair. Considering the variation of \(F\) only with respect to \(u\), we find that for any \(\phi \in \text{NL/MS}(\Omega) = \{ u \in (L^2(\Omega))^3 : |u|_{\text{NL/MS}} < \infty \}\), we have

\[
\int_{\Omega} \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2} dx 
\leq \int_{\Omega} \sqrt{\sum_i (f^i - Ku^i + \epsilon \phi^i)^2 + \eta^2} dx 
+ \frac{\epsilon^2}{2} \max_x |g'(x)|.
\]

Taylor’s expansion gives

\[
g(\epsilon) = \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2 - \epsilon(f - Ku) \cdot (K\phi) / \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} + \epsilon^2 / 2g(\epsilon) \text{ and, hence}
\]

\[
\int_{\Omega} \sqrt{\sum_i (f^i - Ku^i + \epsilon \phi^i)^2 + \eta^2} dx 
\leq \int_{\Omega} \sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2} dx 
- \epsilon \left( \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , K\phi \right) 
+ \frac{\epsilon^2}{2} \max_x |g'(x)|.
\]

Then, the first inequality implies that

\[
\epsilon \left( \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , K\phi \right) 
\leq \epsilon \beta \phi_{\text{NL/MS}} + \frac{\epsilon^2}{2} \max_x |g'(x)|.
\]

Dividing by \(\epsilon > 0\) and letting \(\epsilon \downarrow 0_+\) (while noticing that \(\lim_{\epsilon \to 0_+} \epsilon^2 / 2 \max_x |g'(x)| = 0\)) yields that for any \(\phi \in \text{NL/MS}(\Omega)\),

\[
\left( K^* \frac{f - Ku}{\sqrt{\sum_i (f^i - Ku^i)^2 + \eta^2}} , \phi \right) \leq \beta \phi_{\text{NL/MS}}.
\]
\[ J'(u)h = G'(0) = 2 \sum_i \int_{\Omega} g(x)\phi'(||\nabla_w(u)||^2(x)) \left[ \int_{\Omega} (u(y) - u(x))(h(y) - h(x))w(x,y)dy \right] dx \]

\[ = 2 \int_{\Omega} \int g(x)\phi'(||\nabla_w(u)||^2(x))(u(y) - u(x))w(x,y)dyh(y)dy \]

\[ - 2 \int_{\Omega} g(x)\phi'(||\nabla_w(u)||^2(x)) \left[ \int_{\Omega} (u(y) - u(x))w(x,y)dy \right] h(x)dx \]

\[ = 2 \int_{\Omega} \int g(y)\phi'(||\nabla_w(u)||^2(y))(u(x) - u(y))w(y,x)dyh(x)dx \]

\[ - 2 \int_{\Omega} g(x)\phi'(||\nabla_w(u)||^2(x)) \left[ \int_{\Omega} (u(y) - u(x))w(x,y)dy \right] h(x)dx \]

\[ = -2 \int_{\Omega} \int g(y)\phi'(||\nabla_w(u)||^2(y))(u(y) - u(x))w(y,x)dyh(x)dx \]

\[ - 2 \int_{\Omega} g(x)\phi'(||\nabla_w(u)||^2(x)) \left[ \int_{\Omega} (u(y) - u(x))w(x,y)dy \right] h(x)dx \]

\[ \text{Taking } \epsilon = 0, \text{ we obtain the variation of } J \text{ with respect to } u \text{ i.e.,} \]

\[ \text{See equation at the top of the page. where } \phi'(s) \text{ is the derivative} \]

\[ \phi' \text{ of } \phi \text{ with respect to } s \text{ and } w(x,y) = u(y, x). \text{ Hence, we obtain} \]

\[ Lu = -2 \int_{\Omega} (u(y) - u(x))w(x,y) \]

\[ \times \left[ (g(y)\phi'(||\nabla_w(u)||^2(y)) + g(x)\phi'(||\nabla_w(u)||^2(x)) \right] dy \]

where the operator \( L \) is the gradient flow corresponding to the functional \( J \).

Specifically, by taking \( g(x) = v^2(x) \) and \( \phi(s) = s \) or \( \phi(s) = \sqrt{s} \), we obtain two functionals and the corresponding gradient flows

\[ J_{NL/MSF}^L(u) = \int_{\Omega} v^2||\nabla_w u||^2(x)dx : \]

\[ L_{NL/MSF}^L u = -2\nabla w \cdot (v^2(x)\nabla_w u(x)) \]

\[ = -2 \int_{\Omega} (u(y) - u(x))w(x,y) \]

\[ \times \left[ v^2(y) + v^2(x) \right] dy \]

\[ J_{NL/MS}^L(u) = \int_{\Omega} v^2||\nabla_w u||^2(x)dx : \]

\[ L_{NL/MS}^L u = -\nabla w \cdot \left( v^2(x)||\nabla_w u(x)||^2(x) \right) \]

\[ = - \int_{\Omega} (u(y) - u(x))w(x,y) \]

\[ \times \left[ v^2(y) + v^2(x) \right] \]

\[ \times \left[ ||\nabla_w u||^2(y) + ||\nabla_w u||^2(x) \right] \].

\textbf{REFERENCES}

[1] L. Ambrosio and V. M. Tortorelli, “Approximation of functionals depending on jumps by elliptic functionals via \( \Gamma \)-convergence,” Commun. Pure Appl. Math., vol. 43, pp. 999–1036, 1990.

[2] L. Ambrosio and V. M. Tortorelli, “On the approximation of free discontinuity problems,” Bollettino dell’Unione Matematica Italiana, vol. B(7), no. 6, pp. 105–123, 1992.
