frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 14 June 2019
doi: 10.3389/fams.2019.00027

Modeling Variational Inpainting Methods With Splines

Florian Boßmann[1], *Tomas Sauer*[2] and *Nada Sissouno*[3]*

[1] Department of Mathematics, Harbin Institute of Technology, Harbin, China, [2] Chair of Digital Image Processing, University of Passau, Passau, Germany, [3] Department of Mathematics, Technical University of Munich, Garching, Germany

Mathematical methods of image inpainting often involve the discretization of a given continuous model. Typically, this is done by a pointwise discretization. We present a method that avoids this by modeling known variational approaches using a finite dimensional spline space. The way we build our algorithm we are able to prove that the basis of the spline space is stable. Further, due to the compact supports and structure of the basis the model involves a sparse system matrix allowing a fast and memory efficient implementation. Besides the analysis of the resulting model, we present a numerical implementation based on the alternating method of multipliers. We compare the results numerically with classical TV inpainting and give examples of applications, such as text removal, restoration, and noise removal.

Keywords: spline interpolation, inpainting, discrete approximation, optimization, variational methods

OPEN ACCESS

Edited by:
Volker Michel,
University of Siegen, Germany

Reviewed by:
Gerlind Plonka,
University of Göttingen, Germany
Matthias Augustin,
Saarland University, Germany

****Correspondence:***
Nada Sissouno
sissouno@ma.tum.de

Specialty section:
This article was submitted to Mathematics of Computation and Data Science,
a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *22 December 2018*
Accepted: *17 May 2019*
Published: *14 June 2019*

Citation:
Boßmann F, Sauer T and Sissouno N (2019) Modeling Variational Inpainting Methods With Splines.
Front. Appl. Math. Stat. 5:27.
doi: 10.3389/fams.2019.00027

1. INTRODUCTION

In this paper we study the continuous inpainting problem for digital images. That is, our goal is to restore parts Ω of a digital image Ω', $\Omega \subset \Omega' \subset \mathbb{R}^d$, where the information has been removed, damaged, or is missing, from the remaining and well-maintained part. Naturally, this is only possible under suitable modeling assumptions on the image. For continuous images a common and practical model is that of a function of bounded variation as first studied in the well-known work of Rudin and Osher [1], and Rudin et al. [2]. For a numerical solution and its implementation, these models need to be discretized. For digital images, given by a set of pixels on a uniform grid, those functions and their derivatives are typically discretized using difference schemes for the pixel values. In this paper we rather use a finite dimensional function space, the space of tensor product spline functions based on B-splines, see e.g., [3, 4]. The advantage of this approach is that we avoid a pointwise discretization of the involved functions and derivatives. In addition, the compact support of the basis of B-splines contributes to an efficient and stable implementation.

In general, the inpainting problem involves recovering missing structure and missing texture. In this paper we focus on structural inpainting; for texture reconstruction other methods need to be used or added, see for example [5] and [6]. Also for structural inpainting there exists a variety of approaches. Examples include PDE-based methods (e.g., [5, 7–10]) and variational approaches as in [11–15]. Many of these methods can be combined with different models described by various finite dimensional function spaces [13, 15–18] which is also the approach we will follow in this paper. We especially want to point out the thesis of Hong [13] and the article of Hong et al. [15] which study a variational model similar to ours but using spline functions over triangulations. In our numerical section we also include some examples to compare the performance of our method with examples given in [13] and [15]. For a short survey of splines in image processing in general, see [19].

Naturally, this short summary cannot give a complete account of all related works. For example the problem of image completion (see [20] and references therein) is closely related. For a comprehensive overview to image processing from a mathematical perspective that also discusses inpainting, the authors recommend the books of Chan and Shen [21] as well as Aubert and Kornprobst [22].

Outline and Contribution

We will follow a variational approach by, roughly speaking, minimizing some functional over an extended inpainting area $U(\Omega) \supset \Omega$ subject to constraints which ensure the reproduction of the well-maintained part of the image. Typically, the functional involves the image function u and some of its derivatives. The constraint, on the other hand, is chosen over some neighborhood $B \subseteq \Omega' \setminus \Omega$ of the inpainting area. We will model the class of functions for the variational method with tensor product B-splines together with a focus on *total variation (TV) inpainting*, [23], that is

$$\int_{U(\Omega)} |\nabla u(\boldsymbol{x})|\, d\boldsymbol{x}, \tag{1}$$

the effective, often used ROF-functional, see [2], which results in level curves of minimal length. Note that the application to this model is exemplary and the presented approach can be applied to other functionals as well.

We begin by giving some basic properties and notation for the tensor product splines used in this paper in section 2. Afterwards, in section 3 we show how to model the inpainting problem with those splines and analyze its properties. Numerical results are given in section 4. We present an algorithm and its implementation using a first-order primal-dual algorithm [24], compare the results of our method to standard TV inpainting and show some examples of applications.

2. PRELIMINARIES AND NOTATION

In the sequel, we will use the following notations and basic facts about B-splines and their derivatives. Even if images are usually considered in 2D, we will present the theory in an arbitrary number of variables; applications in higher dimensions would, for example include medical applications such as inpainting in voxel data provided by computerized tomography. The tensor product B-splines for inpainting which we will use in section 3 are defined over a rectangle $R = \otimes_{j=1}^{d}[a_j, b_j]$, $d \in \mathbb{N}$. They are based on a *tensor product knot grid* which is given as

$$T := \otimes_{j=1}^{d}\{\tau_{j,1}, \dots, \tau_{j,m_j+2n_j}\} \tag{2}$$

with $\tau_{j,i} < \tau_{j,i+n_j}$ for $n_j \leq i \leq m_j + n_j$. At the boundary, we request multiple knots

$$\begin{aligned} \tau_{j,1} &= \cdots = \tau_{j,n_j} = a_j, \\ \tau_{j,m_j+n_j+1} &= \cdots = \tau_{j,m_j+2n_j} = b_j, \end{aligned} \tag{3}$$

for $n_j, m_j \in \mathbb{N}$, $1 \leq j \leq d$. We set $\boldsymbol{n} := (n_1, \dots, n_d)$ and $\boldsymbol{m} := (m_1, \dots, m_d)$ for the dimensions and degrees in the individual coordinates, respectively. The grid width $\boldsymbol{h} := (h_1, \dots, h_d)$, defined as $h_j := \max_k |\tau_{j,k} - \tau_{j,k+1}|$, is known to influence the approximation properties of the spline space. For $\boldsymbol{k} \in \mathrm{K} := \otimes_{j=1}^{d}\{1, \dots, m_j + 2n_j - 1\}$ a *grid cell* is denoted by $Z_{\boldsymbol{k}} := \otimes_{j=1}^{d}[\tau_{j,k_j}, \tau_{j,k_j+1}]$.

The *tensor product B-splines* of order $\boldsymbol{n}$ with respect to the grid T are denoted by

$$B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x}) = \prod_{j=1}^{d} B_{\alpha_j}^{n_j}(x_j), \qquad \boldsymbol{\alpha} \leq \boldsymbol{m} + \boldsymbol{n}, \tag{4}$$

i.e., $\boldsymbol{\alpha} \in \mathrm{I}_{\boldsymbol{m},\boldsymbol{n}} := \otimes_{j=1}^{d}\{1, \dots, m_j + n_j\}$. They have the *support*

$$S_{\boldsymbol{\alpha}}^{\boldsymbol{n}} := \otimes_{j=1}^{d} S_{\alpha_j}^{n_j}, \qquad S_{\alpha_j}^{n_j} := [\tau_{j,\alpha_j}, \tau_{j,\alpha_j+n_j}], \tag{5}$$

respectively. The *spline space* $\mathscr{S}^{\boldsymbol{n}}(T, \Omega)$ of order $\boldsymbol{n}$ restricted to a domain $\Omega \subset \mathbb{R}^d$ is spanned by all B-splines $B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}$ with $S_{\boldsymbol{\alpha}}^{\boldsymbol{n}} \cap \Omega \neq \emptyset$. The index set of all B-splines relevant for Ω is defined as $\mathrm{I}_\Omega := \{\boldsymbol{\alpha} \in \mathrm{I}_{\boldsymbol{m},\boldsymbol{n}} | S_{\boldsymbol{\alpha}}^{\boldsymbol{n}} \cap \Omega \neq \emptyset\}$ and their number as $\#(\mathrm{I}_\Omega)$. If $\Omega = R$, then $\mathrm{I}_R = \mathrm{I}_{\boldsymbol{m},\boldsymbol{n}}$ and $\#(\mathrm{I}_R) = \prod_{j=1}^{d} m_j + n_j$.
A tensor product spline of order $\boldsymbol{n}$ with respect to T over a domain Ω is given by

$$s(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathrm{I}_\Omega} g_{\boldsymbol{\alpha}} B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \tag{6}$$

with coefficients $g_{\boldsymbol{\alpha}} \in \mathbb{R}$. Let $\mathbf{g} = \left(g_{\boldsymbol{\alpha}}\right)_{\boldsymbol{\alpha} \in \mathrm{I}_\Omega}$ and $\mathbf{B}(\boldsymbol{x})^T := \left(B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x})\right)_{\boldsymbol{\alpha} \in \mathrm{I}_\Omega}$ denote the vector of the coefficients and B-splines, respectively, then the spline s is given in matrix notation by $s(\boldsymbol{x}) = \mathbf{B}(\boldsymbol{x})\mathbf{g}$.

With respect to inpainting methods an important property of the spline functions is that their derivatives can be expressed in terms of the coefficients of the spline function itself, that is

$$\partial_j s(\boldsymbol{x}) := \frac{\partial}{\partial x_j} s(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathrm{I}_\Omega} g_{\boldsymbol{\alpha}}\, \partial_j B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x}) \tag{7}$$

where

$$\partial_j B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x}) := \frac{\partial}{\partial x_j} B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x}) = (n_j - 1)\left(\frac{B_{\boldsymbol{\alpha}}^{\boldsymbol{n}-\varepsilon_j}(\boldsymbol{x})}{|S_{\alpha_j}^{n_j-1}|} - \frac{B_{\boldsymbol{\alpha}+\varepsilon_j}^{\boldsymbol{n}-\varepsilon_j}(\boldsymbol{x})}{|S_{\alpha_j+1}^{n_j-1}|}\right) \tag{8}$$

denotes the derivative of the B-splines. Here, ε_j is the j-th unit vector.

The ℓ^p-norm of vectors will be denoted by $\| \cdot \|_p$ and the L^p-norm of functions by $\| \cdot \|_{p,\Omega}$.

3. MODELING OF INPAINTING PROBLEM

Assume we want to reproduce a rectangular image $\Omega' \subset \mathbb{R}^d$ by a tensor product spline. This can easily be done by choosing a tensor product grid with multiple knots on the boundary $\partial\Omega'$ such that Ω' is the domain of definition. This method guarantees that the B-spline basis is stable.

Theorem 1. *For any rectangular domain $R = \otimes_{j=1}^{d}[a_j, b_j]$, $d \in \mathbb{N}$, and any spline order $\boldsymbol{n} = (n_1, \ldots, n_d) \in \mathbb{N}^d$, the B-spline basis $\{B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{x})\}_{\boldsymbol{\alpha}\in I_{\boldsymbol{m},\boldsymbol{n}}}$ with respect to knots defined as in (2) and (3) is stable, that is, there exist constants $c, C > 0$ such that*

$$c\|\mathbf{g}\|_p \leq \|\mathbf{Bg}\|_{p,\Omega} \leq C\|\mathbf{g}\|_p, \qquad \mathbf{g} \in \mathbb{R}^{I_\Omega}, \tag{9}$$

The constants are only depending on $\boldsymbol{n}$ and d.

Due to the tensor product structure and the multiple knots on the boundary, this result can be proven analogously to the classical univariate results for intervals, see, e.g., [25]. For further information about stability of tensor product B-spline bases see for example [26]. In addition to stability, multiple knots on the boundary of R help us to avoid artifacts at the boundary since they result in an interpolation at the boundary and, thus, satisfying a non-homogeneous Dirichlet boundary condition.

We introduce a general version of our interpolation problem before stating the concrete model for which we give proofs and examples. It seems worthwhile, however, to mention that it also works with other models. For arbitrary $\Omega' \subset \mathbb{R}^d$, the inpainting problem considered here can be described in the following way: Given two bounded domains Ω and Ω' such that

1. $\Omega \subset \Omega' \subset \mathbb{R}^d$,
2. there exists $R = \otimes_{j=1}^{d}[a_j, b_j]$ such that $\Omega \subset R \subseteq \Omega'$ and $\partial\Omega \cap \partial R = \emptyset$.

The goal is to reconstruct a function or picture f on Ω that is known only on $\Omega^* := \Omega' \setminus \Omega$ by using tensor product splines defined over R. The situation is illustrated in **Figure 1**. Note that in case of multiple disconnected unknown inpainting areas, as for example illustrated in the left image of **Figure 3**, the method can be applied for each area separately. For easier comparability we choose $R = \Omega'$ for all of our examples considered in section 4.

We require that the spline fulfills a side condition on $R^* := R \setminus \Omega \subset \Omega^*$ and minimizes a functional F over some neighborhood $U(\Omega)$ including Ω. The general variational inpainting model using splines can now be formulated as follows.

Spline Inpainting Model 1. *Let $U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega)$ be some neighborhood of Ω only depending on $\boldsymbol{h}$ and $\boldsymbol{n}$ with $\Omega \subseteq U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega) \subseteq R$. The spline inpainting model is given by: Determine $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$ by*

$$\text{minimize} \quad F(s, \nabla s, \ldots)(\boldsymbol{x}) \;\; \text{for } \boldsymbol{x} \in U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega) \tag{10}$$

$$\text{subject to} \quad s(\boldsymbol{x}) = f(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in R^*. \tag{11}$$

Remark 1. *Since hard constraints can be problematic, especially in the presence of noise, the minimization problem is often relaxed to*

$$\text{minimize} \quad F(s, \nabla s, \ldots)|_{U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega)} + \frac{\varepsilon}{2}\|s - f\|_{2,R^*}^2 \tag{12}$$

for some $\varepsilon > 0$. This relaxed formulation will be discussed briefly in section 4.

For the explicit modeling or the discretization of the Spline Inpainting model 1, the concrete functional in the minimization is, of course, crucial. As already mentioned in section 1, we will follow one of the first important approaches and focus on TV inpainting [23] and, therefore, on the ROF-functional [2].

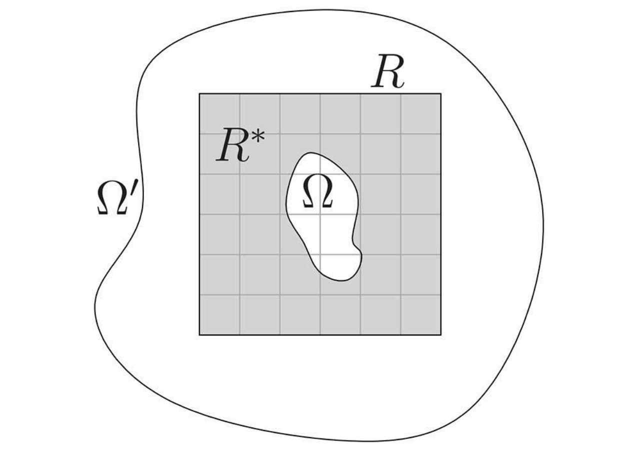

FIGURE 1 | Illustration of domains.

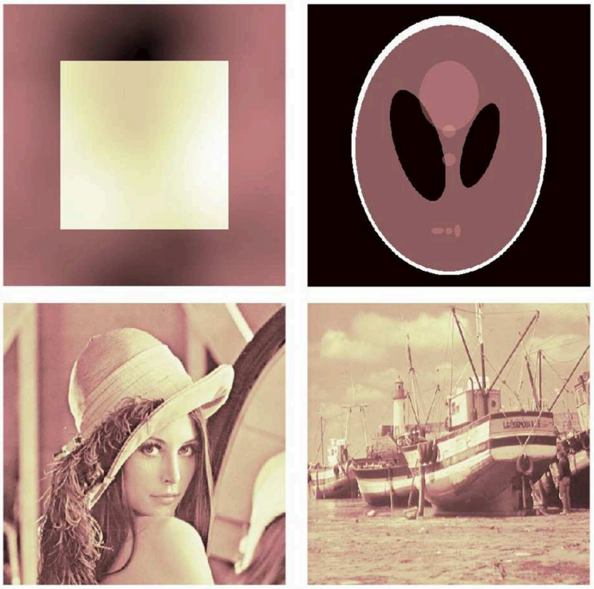

FIGURE 2 | Exemplary test data: cartoon-like images and natural images.

Spline Inpainting Model 2. *Let $U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega)$ be some neighborhood of Ω only depending on $\boldsymbol{h}$ and $\boldsymbol{n}$ with $\Omega \subseteq U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega) \subseteq R$. The TV spline inpainting model is given by: Determine $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$ by*

$$\text{minimize} \quad \int_{U_{\boldsymbol{h}}^{\boldsymbol{n}}(\Omega)} |\nabla s(\boldsymbol{x})| \, d\boldsymbol{x} \tag{13}$$

$$\text{subject to} \quad s(\boldsymbol{x}) = f(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in R^*, \tag{14}$$

where $|\nabla s(\boldsymbol{x})| := \sqrt{\sum_{i=1}^{d} \left(\partial_i s(\boldsymbol{x})\right)^2}$.

Considering discrete problems (e.g., digital images) and, thus, a discrete set of points, the constraint coincides with an interpolation of a finite set of points. To apply the interpolation, usually, a discretization of all involved functions and derivatives

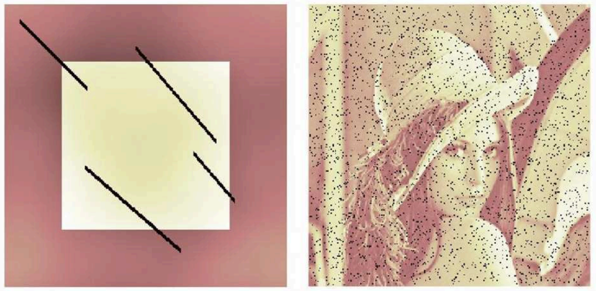

FIGURE 3 | Considered inpainting area types: Scratches **(Left)** and randomly missing pixels **(Right)**.

is necessary. In the next two subsections we show that for the continuous Spline Inpainting Models 2 only the discretization of the integral over grid cells $Z_{\boldsymbol{k}}$ in a neighborhood of the inpainting area Ω is needed. Note that Theorem 2 concerning the constraint also holds true in case of the general model Spline Inpainting Model 1. Theorem 3 treats the special functional considered in (13). Nevertheless, the decomposition in (32) as well as the representation of the involved spline functions and derivatives remain the same. Thus, depending on concrete given functional Theorem 3 and the Discrete Spline Inpainting Model 1 would have to be adapted. As a simple approach one could use the squared gradient norm in (13), i.e., $|\nabla s(\boldsymbol{x})|^2$. This puts a stronger penalty term on large gradients what can be useful for large inpainting areas. We will also test this approach in the numerical section of this work. In case of the often considered Laplace and biharmonic equations (e.g., [17] and references therein) our spline approach would reduce to a linear system with respect to the coefficients of the spline function.

Clearly, there are different possible interpretations of a discrete image. In our situation, where we evaluate functions at points in R, those interpretations influence the value of the image that we assume to be at that points. Since it does not change the method or modeling, we interpret, for the sake of simplicity, the discrete values as piecewise constant functions with constant values over rectangles. This means, for an image f considered over $R = \otimes_{j=1}^{d}[a_j, b_j]$ consisting of $\mu_1 \times \cdots \times \mu_d$ pixels, the single pixels are defined as uniform rectangles $P_{\boldsymbol{\beta}} := \otimes_{j=1}^{d}[a_j + (\beta_j - 1)p_j, a_j + \beta_j p_j)$ for $p_j := (b_j - a_j)/\mu_j$ and $1 \le \beta_j \le \mu_j$. The value of the image over $P_{\boldsymbol{\beta}}$ is chosen corresponding to the value at the center $c(P_{\boldsymbol{\beta}}) := \left(a_j + (\beta_j - \frac{1}{2})p_j\right)_{j=1}^{d}$ of this pixel; the j-th coordinate will be denoted by $c(P_{\boldsymbol{\beta}})_j$.

3.1. Constraint

In the reproduction of digital images on R^*, the constraint (14) corresponds to the interpolation of the pixel values in R^*. We use the centers of the pixels $c(P_{\boldsymbol{\beta}})$ as those interpolation points. In the space of tensor product splines an interpolation problem is solvable if the number of interpolation points is not greater than the degrees of freedom $\#(\mathrm{I}_R)$ and if there is a relationship between the interpolation sites and the knots, known as the Schoenberg–Whitney condition. Since we also need some degrees of freedom for the minimization problem over $U(\Omega)$, this forces us to choose a spline space of sufficiently large dimension.

Definition 1. *Given a tensor product grid T over R and a set of discrete points $\Xi := \{\boldsymbol{\xi}_{\boldsymbol{\beta}} \in R | \boldsymbol{\beta} \in \mathbb{Z}^d; \#(\boldsymbol{\beta}) \le \#(\mathrm{I}_R)\}$, let $\lambda_{\boldsymbol{\beta}}$ be the point evaluation functionals $\lambda_{\boldsymbol{\beta}}(g) := g(\boldsymbol{\xi}_{\boldsymbol{\beta}})$, $g : \Xi \to \mathbb{R}$.* A spline $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$ *is the* spline interpolant of a function g at Ξ *if*

$$\lambda_{\boldsymbol{\beta}}(s) = \sum_{\boldsymbol{\alpha} \in \mathrm{I}_R} g_{\boldsymbol{\alpha}} B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\boldsymbol{\xi}_{\boldsymbol{\beta}}) = \lambda_{\boldsymbol{\beta}}(g), \qquad \boldsymbol{\xi}_{\boldsymbol{\beta}} \in \Xi. \tag{15}$$

For functions defined over R and grids T satisfying (2) and (3) this interpolation problem is uniquely solvable if the Schoenberg-Whitney condition (see, e.g., [27]) is satisfied. Since R is a rectangle, this can be guaranteed by choosing the tensor product *Greville abscissae* as interpolation sites, that is, by setting

$$\Xi_{Gr} := \left\{ \boldsymbol{\xi}_{\boldsymbol{\gamma}} \in \mathbb{R}^d | \xi_{\boldsymbol{\gamma},j} := \frac{1}{n_j - 1} \sum_{i=1}^{n_j - 1} \tau_{j,\gamma_j + i}, \boldsymbol{\gamma} \in \mathrm{I}_R \right\}. \tag{16}$$

It should be noted that the position of the *Greville abscissae* depends on the position of the knots of T, hence, the interpolation points depend on T. Since the centers of the pixels in R^* need to be interpolated, we have to choose a knot grid such that those are contained in the set of Greville abscissae. In case of digital images, where each rectangle $P_{\boldsymbol{\beta}}$ represents a pixel of the image, we have the following result for the reproduction of all known image pixel values.

Theorem 2. *Given an image f of size $\otimes_{j=1}^{d}\mu_j$, an area $R^* := \cup_{\boldsymbol{\beta} \in \mathrm{J}_{R^*}} P_{\boldsymbol{\beta}}$, $\mathrm{J}_{R^*} := \{\boldsymbol{\beta} \in \mathbb{Z}^d : 1 \le \beta_j \le \mu_j,\ 1 \le j \le d \text{ and } c(P_{\boldsymbol{\beta}}) \in R \setminus \Omega\}$ of known pixel values and an order $\boldsymbol{n}$ of a spline space, there exist a knot grid T and a set of interpolation points Ξ^* such that*

$$\{c(P_{\boldsymbol{\beta}})\}_{\boldsymbol{\beta} \in \mathrm{J}_{R^*}} \subset \Xi^*, \qquad \lambda_{\boldsymbol{\gamma}}(s) = \lambda_{\boldsymbol{\gamma}}(f), \qquad \boldsymbol{\xi}_{\boldsymbol{\gamma}} \in \Xi^*, \tag{17}$$

for $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$.

Proof: Due to the tensor product structure, it suffices to consider a single coordinate direction j for the construction of the grid and set of interpolation points.

In doing so, we have to distinguish between even and odd values of n_j. We set $m_j = \mu_j - 1$ for n_j odd and $m_j = \mu_j$ otherwise, and choose

$$\begin{cases} \tau_{j,n_j+k_j} := a_j + k_j \cdot p_j & n_j \text{ odd}, \\ \tau_{j,n_j+k_j} := a_j + (k_j - \frac{1}{2}) \cdot p_j & n_j \text{ even} \end{cases} \tag{18}$$

for $1 \le k_j \le m_j$ and $p_j := \frac{b_j - a_j}{\mu_j}$. The boundary knots are determined with proper multiplicity according to (3). Recalling

(16), the j-th coordinates of the associated Greville abscissae for odd n_j are given by

$$\xi_{\boldsymbol{\gamma},j} = \begin{cases} a_j & \text{for } \gamma_j = 1, \\ a_j + \frac{p_j}{n_j-1}\left(\sum\limits_{\ell=1}^{\gamma_j-1} \ell\right) & \text{for } \begin{matrix}\gamma_j \geq 2 \\ \gamma_j \leq n_j - 1,\end{matrix} \\ a_j + \frac{p_j}{n_j-1}\left(\sum\limits_{\ell=\gamma_j+1-n_j}^{\gamma_j-1} \ell\right) & \text{for } \begin{matrix}\gamma_j \geq n_j \\ \gamma_j \leq m_j + 1,\end{matrix} \\ b_j - \frac{p_j}{n_j-1}\left(\sum\limits_{\ell=1}^{m_j+n_j-\gamma_j} \ell\right) & \text{for } \begin{matrix}\gamma_j \geq m_j + 2 \\ \gamma_j \leq m_j + n_j - 1,\end{matrix} \\ b_j & \text{for } \gamma_j = m_j + n_j. \end{cases} \tag{19}$$

Replacing the terms ℓ of the sums by $\ell - 1/2$, results in the Greville abscissae for even n_j, respectively. Therefore, we have

$$\xi_{\boldsymbol{\gamma},j} = \begin{cases} a_j + \frac{(\gamma_j-1)\gamma_j}{2(n_j-1)}\, p_j & \text{for } \begin{matrix}\gamma_j \geq 1 \\ \gamma_j \leq n_j - 1,\end{matrix} \\ a_j + \left(\gamma_j - \frac{n_j}{2}\right) p_j & \text{for } \begin{matrix}\gamma_j \geq n_j \\ \gamma_j \leq m_j + 1,\end{matrix} \\ b_j - \frac{(m_j+n_j-\gamma_j+1)(m_j+n_j-\gamma_j)}{2(n_j-1)}\, p_j & \text{for } \begin{matrix}\gamma_j \geq m_j + 2 \\ \gamma_j \leq m_j + n_j.\end{matrix} \end{cases} \tag{20}$$

for odd n_j and

$$\xi_{\boldsymbol{\gamma},j} = \begin{cases} a_j & \text{for } \gamma_j = 1, \\ a_j + \frac{(\gamma_j-1)^2}{2(n_j-1)}\, p_j & \text{for } \begin{matrix}\gamma_j \geq 2 \\ \gamma_j \leq n_j - 1,\end{matrix} \\ a_j + \left(\gamma_j - \frac{n_j+1}{2}\right) p_j & \text{for } \begin{matrix}\gamma_j \geq n_j \\ \gamma_j \leq m_j + 1,\end{matrix} \\ b_j - \frac{(m_j+n_j-\gamma_j)^2}{2(n_j-1)}\, p_j & \text{for } \begin{matrix}\gamma_j \geq m_j + 2 \\ \gamma_j \leq m_j + n_j - 1,\end{matrix} \\ b_j & \text{for } \gamma_j = m_j + n_j. \end{cases} \tag{21}$$

for even n_j, respectively. Now $\xi_{\boldsymbol{\gamma},j}$ coincides, for $n_j \leq \gamma_j \leq m_j+1$, with $c(P_{\boldsymbol{\beta}})_j$ for some $1 \leq \beta_j \leq \mu_j$; more precisely, with centers $c(P_{\boldsymbol{\beta}})_j$ such that

$$\beta_j \in \mathring{\mathrm{I}} := \begin{cases} \{\frac{n_j+1}{2}, \ldots, \mu_j - \frac{n_j-1}{2}\} & \text{for odd } n_j \\ \{\frac{n_j}{2}, \ldots, \mu_j + 1 - \frac{n_j}{2}\} & \text{for even } n_j \end{cases}. \tag{22}$$

The Greville abscissae of *boundary B-splines*, that is, of B-splines with some knots at a_j or b_j, are not necessarily placed at a center, but they can be replaced by the center that is closest to them without loss of the Schoenberg-Whitney condition $B_{\gamma_j}^{n_j}(\xi_j) > 0$, due to property (5).

We now claim: *For every* $\beta_j \in \{1, \ldots, \mu_j\} \setminus \mathring{\mathrm{I}}$, *there exists* γ_j *such that*

$$-\frac{1}{2}p_j < \xi_{\boldsymbol{\gamma},j} - c(P_{\boldsymbol{\beta}})_j \leq \frac{1}{2}p_j. \tag{23}$$

There are four cases to be considered: small and large β_j (or γ_j) and even or odd order. We prove the claim for small β_j or γ_j and odd order, the other cases can be verified in the same way. For $\gamma_j \in \{2, \ldots, n_j - 1\}$ and odd n_j we have

$$\xi_{\boldsymbol{\gamma},j} - c(P_{\boldsymbol{\beta}})_j = \left(\frac{(\gamma_j - 1)\gamma_j}{2(n_j - 1)} - \beta_j + \frac{1}{2}\right) \cdot p_j. \tag{24}$$

Therefore, we need to show that for every $\beta_j \in \{1, \ldots, (n_j - 1)/2\}$ there exists a $\gamma_j \in \{2, \ldots, n_j - 1\}$ such that

$$\beta_j - 1 < \frac{(\gamma_j - 1)\gamma_j}{2(n_j - 1)} \leq \beta_j. \tag{25}$$

Set $h(\gamma_j) := \frac{(\gamma_j-1)\gamma_j}{2(n_j-1)}$. Then we have $0 < h(2) \leq 1$ and $\frac{n_j-1}{2} - 1 < h(n_j - 1) \leq \frac{n_j-1}{2}$. As a consequence, $0 < h(\gamma_j) \leq \frac{n_j-1}{2}$ for all $\gamma_j \in \{2, \ldots, n_j - 1\}$. Clearly, $h(\gamma_j)$ is monotonically increasing and $\#\gamma_j = n_j - 2$ distributed over $(n_j - 1)/2$ intervals. Further, $h(\gamma_j) - h(\gamma_j - 1) = \frac{\gamma_j-1}{n_j-1} \leq 1$ and, thus, the claim is valid.

Having this property at hand, we replace those $\xi_{\boldsymbol{\gamma},j}$ by $c(P_{\boldsymbol{\beta}})_j$ that are closest to the center, to guarantee that all centers are contained in the set of interpolation points. For $\beta \in \{1, \ldots, \mu_j\} \setminus \mathring{\mathrm{I}}$ set

$$\tilde{\mathrm{I}} := \left\{\gamma_j \in \{1, \ldots, m_j + n_j\} : \xi_{\boldsymbol{\gamma},j} = \min_{\tilde{\gamma}_j}\{\operatorname{argmin}_{\xi_{\tilde{\boldsymbol{\gamma}},j}} |\xi_{\tilde{\boldsymbol{\gamma}},j} - c(P_{\boldsymbol{\beta}})_j|\}\right\} \tag{26}$$

and

$$\tilde{\xi}_{\boldsymbol{\gamma},j} := \operatorname{argmin}_{c(P_{\boldsymbol{\beta}})_j} |\xi_{\tilde{\boldsymbol{\gamma}},j} - c(P_{\boldsymbol{\beta}})_j| \quad \text{for } \gamma_j \in \tilde{\mathrm{I}}. \tag{27}$$

The set

$$\Xi_j := \left\{\xi_{\boldsymbol{\gamma},j} : \gamma_j \in \{1, \ldots, m_j + n_j\} \setminus \tilde{\mathrm{I}}\right\} \cup \left\{\tilde{\xi}_{\boldsymbol{\gamma},j} : \gamma_j \in \tilde{\mathrm{I}}\right\} \tag{28}$$

gives us the j-th coordinate of Ξ, the set of possible interpolation points. Finally, the set Ξ^* of interpolation points is determined by restriction of Ξ to R^*, that is $\boldsymbol{\xi_\gamma} \in \Xi^* := \Xi \cap R^*$. By construction, $c(P_{\boldsymbol{\beta}}) \in \Xi^*$ for $\boldsymbol{\beta} \in \mathrm{J}_{R^*}$ and there exists a spline interpolant $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$ for f at Ξ^*. □

In the sequel, Ξ and Ξ^* always will refer to the sets of interpolation sites determined in the way described in the proof of Theorem 2. For applications we also want to describe the constraints in matrix notation. For the coefficients $\mathbf{g} = \left(g_{\boldsymbol{\alpha}}\right)_{\boldsymbol{\alpha} \in \mathrm{I}_R} \in \mathbb{R}^{\#(\mathrm{I}_R)}$, we get

$$\mathbf{s}_{\Xi^*} = \mathbf{B}_{\Xi^*}\mathbf{g} = \mathbf{f}_{\Xi^*} \tag{29}$$

with

$$\mathbf{f}_{\Xi^*} = \left(f(\xi_{\boldsymbol{\beta}})\right)_{\boldsymbol{\beta} \in \Xi^*} \in \mathbb{R}^{\#(\Xi^*)}, \tag{30}$$

$$\mathbf{B}_{\Xi^*} = \left[B_{\boldsymbol{\alpha}}^{\boldsymbol{n}}(\xi_{\boldsymbol{\beta}}) : \begin{matrix}\xi_{\boldsymbol{\beta}} \in \Xi^* \\ \boldsymbol{\alpha} \in \mathrm{I}_R\end{matrix}\right] \in \mathbb{R}^{\#(\Xi^*) \times \#(\mathrm{I}_R)}. \tag{31}$$

3.2. Minimization

Depending on Ω, there are still some degrees of freedom left for minimization; their number is given by $\#(\Xi \setminus \Xi^*)$. We define $U_T^{\boldsymbol{n}}(\Omega)$ as the union of all supports of B-splines corresponding to $\xi_{\boldsymbol{\alpha}} \in \Xi \setminus \Xi^*$, that is

$$U_T^{\boldsymbol{n}}(\Omega) := \bigcup_{\boldsymbol{\alpha} \in \mathrm{I}_R \setminus \mathrm{I}_{\Xi^*}} S_{\boldsymbol{\alpha}}^{\boldsymbol{n}}. \tag{32}$$

The index set of all cells in $U_T^{\boldsymbol{n}}(\Omega)$ is denoted by $\mathrm{K}_\Omega := \{\boldsymbol{k} \in \mathrm{K} |\, Z_{\boldsymbol{k}} \subset U_T^{\boldsymbol{n}}(\Omega)\}$.

Proposition 3. *Given T and $\Omega \subset R$. For $U_T^{\boldsymbol{n}}(\Omega)$ defined as in* (32)*, the minimization problem* (13) *reduces to*

$$\min_{\mathbf{g}} \sum_{\boldsymbol{k}\in \mathrm{K}_\Omega} \int_{Z_{\boldsymbol{k}}} |\big(\mathbf{B}^j(\boldsymbol{x})\mathbf{g}\big)_{j=1}^{d}|\, d\boldsymbol{x}, \tag{33}$$

where $\mathbf{g} = \big(g_{\boldsymbol{\alpha}}\big)_{\boldsymbol{\alpha}\in \mathrm{I}_\Omega}$, $g_{\boldsymbol{\alpha}} \in \mathbb{R}$, *and*

$$[\mathbf{B}^j(\boldsymbol{x})]^T := \big(\partial_j B^{\boldsymbol{n}}_{\boldsymbol{\alpha}}(\boldsymbol{x})\big)_{\boldsymbol{\alpha}\in \mathrm{I}_R} \in \mathbb{R}^{\#(\mathrm{I}_R)}. \tag{34}$$

Remark 2. *It is worthwhile to point out that the derivatives $\partial_j B^{\boldsymbol{n}}_{\boldsymbol{\alpha}}(\boldsymbol{x})$ are just weighted differences of B-splines of lower order, the explicit formula being given in section 2. This fact is of crucial importance for the efficient implementation of the method as it leads to sparse difference matrices.*

Proof of Proposition 3: The result follows by applying (7) and (32). The first one directly gives us $\nabla s(x) = \big(\mathbf{B}^j(\boldsymbol{x})\mathbf{g}\big)_{j=1}^{d}$ with unknown coefficients $\mathbf{g}$. The area of integration $U_T^{\boldsymbol{n}}(\Omega)$ consists of a union of grid cells and, thus, the integral can be split up into those grid cells $Z_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathrm{K}_\Omega$. □

Typically, for digital images the integration and the derivatives need to be discretized, as already mentioned in section 1. In our case, we only need to discretize the integral and the way the problem is modeled helps with that, too: the discretization of the integrals can be done simply and efficiently by using a Gaussian quadrature formula for the grid cells $Z_{\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathrm{K}_\Omega$. The tensor product structure allows us to use the univariate Gauss-Legendre quadrature formula in each coordinate direction. Let $\Theta \subset U_T^{\boldsymbol{n}}(\Omega)$ be the set of the nodes of the Gauss-Legendre quadrature and $w_{\boldsymbol{\theta}}$ be the corresponding weights. We get

$$\sum_{\boldsymbol{k}\in \mathrm{K}_\Omega} \int_{Z_{\boldsymbol{k}}} |\big(\mathbf{B}^j(\boldsymbol{x})\mathbf{g}\big)_{j=1}^{d}|\, d\boldsymbol{x} \approx \sum_{\boldsymbol{\theta}} w_{\boldsymbol{\theta}}\, \big\|\big(\mathbf{B}^j(\boldsymbol{\theta})\mathbf{g}\big)_{j=1}^{d}\big\|_2, \tag{35}$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

Discrete Spline Inpainting Model 1. *Using Proposition 3 and 2 [or* (29)*] the discrete form of TV Spline Inpainting Model 2 is given by: Determine $s \in \mathscr{S}^{\boldsymbol{n}}(T, R)$ by*

$$\begin{aligned} &\textit{minimize}_{g_{\boldsymbol{\alpha}}} && \textstyle\sum_{\boldsymbol{\theta}\in\Theta} w_{\boldsymbol{\theta}}\, \|\mathbf{B}_{\boldsymbol{\theta}}\mathbf{g}\|_2 \\ &\textit{subject to} && \mathbf{B}_{\Xi^*}\mathbf{g} = \mathbf{f}_{\Xi^*} \end{aligned}$$

where $\mathbf{B}_{\boldsymbol{\theta}} := \big(\mathbf{B}^j(\boldsymbol{\theta})\big)_{j=1}^{d} \in \mathbb{R}^{d\times\#(\mathrm{I}_R)}$.

This approach is also suitable for other types of Spline Inpainting Models 1, as long as the functionals depend on the function and its derivatives and it can be applied as soon as the explicit functional is given.

4. NUMERICAL IMPLEMENTATION AND EXPERIMENTS

Define the convex operators $F : \mathbb{R}^{d\times\#(\Theta)} \to \mathbb{R}$ and $G : \mathbb{R}^{\#(\mathrm{I}_\Omega)} \to \mathbb{R}$ with

$$F(x_1, \ldots, x_{\#(\Theta)}) = \sum_{k=1}^{\#(\Theta)} \|x_k\|_2, \tag{36}$$

$$G(\mathbf{g}) = \begin{cases} 0 & \mathbf{B}_{\Xi^*}\mathbf{g} = \mathbf{f}_{\Xi^*}, \\ \infty & \text{otherwise}, \end{cases} \tag{37}$$

and the linear operator $K : \mathbb{R}^{\#(\mathrm{I}_\Omega)} \to \mathbb{R}^{2\times\#(\Theta)}$ with $K(\mathbf{g}) = (w_{\boldsymbol{\theta}}\mathbf{B}_{\boldsymbol{\theta}}\mathbf{g})_{\boldsymbol{\theta}\in\Theta}$. Using this, the Discrete Spline Inpainting Model 1 can be reformulated as an optimization problem in the following way:

Optimization Problem 1. *Our Spline Inpainting Model 1 is equivalent to the unconstrained problem*

$$\textit{minimize} \qquad F(K\mathbf{g}) + G(\mathbf{g}).$$

This formulation is suitable for the first-order primal-dual algorithm presented in [24]. Therefore, the λ-proximity operators $prox_{\lambda F^*}$, $prox_{\lambda G}$ need to be calculated. Note that the convex conjugate function $F^* : \mathbb{R}^{d\times\#(\Theta)} \to \mathbb{R}$ is given by

$$F^*(y_1, \ldots, y_{\#(\Theta)}) = \begin{cases} 0 & \|y_k\|_2 \le 1,\ \forall k \le \#(\Theta) \\ \infty & \text{otherwise} \end{cases}. \tag{38}$$

Both, F^* and G are indicator functions and, thus, the proximity operators are equivalent to the projections:

$$\underset{\lambda F^*}{prox}(y_1, \ldots, y_{\#(\Theta)}) = \left(\frac{y_1}{\max(1, \|y_1\|_2)}, \ldots, \frac{y_{\#(\Theta)}}{\max(1, \|y_{\#\Theta}\|_2)}\right), \tag{39}$$

$$\underset{\lambda G}{prox}(\mathbf{g}) = \mathbf{g} - \mathbf{B}^{+}_{\Xi^*}(\mathbf{B}_{\Xi^*}\mathbf{g} - \mathbf{f}_{\Xi^*}), \tag{40}$$

where $\mathbf{B}^{+}_{\Xi^*}$ is the pseudoinverse of the rectangular matrix $\mathbf{B}_{\Xi^*}$. We remark that both operators are independent of λ. In our experiments, we use the MATLAB implementation of the primal-dual solver due to Gabriel Peyre [28] and apply the tests to several cartoon-like images and natural images in different sizes; some are shown in **Figure 2**.

We consider two different scenarios for the inpainting region Ω. In one scenario, the image is damaged by one or several "scratches" of variable width; in the second case we consider randomly missing pixels similar to "salt-and-pepper" noise. **Figure 3** gives an example for both types (in comparison to **Figure 2** the contrast is changed to emphasize the inpainting area).

In the next subsection we discuss the optimal spline order for the inpainting problem as well as reasonable strategies to guess a starting value for the primal-dual algorithm. We compare our results with the standard TV inpainting using, again, an implementation by Gabriel Peyre [28]. Afterwards, we demonstrate the benefits of our method by two examples: Text removal and salt-and-pepper denoising.

4.1. Numerical Evaluation

4.1.1. Starting Guess

Both implementations, spline inpainting and standard TV inpainting, are iterative methods. Thus, they rely on a suitable initial value to return a good solution after a reasonable number of iterations. We tried several strategies among which two stood out and will be detailed in what follows. In a first strategy we choose random uniformly distributed values in $[0, 255]$ (for standard grayscale images). Note that standard TV inpainting directly iterates on the pixel values of the inpainting area while our spline approach works on the spline coefficients as its variables. Thus, although both algorithms use uniformly distributed values, the starting guess strategies differ slightly. For our second strategy, we calculate the mean value ω_{mean} of the image outside the inpainting area. In contrast to that, standard TV sets the pixels inside the inpainting area to ω_{mean}. Our spline approach uses the starting values $\mathbf{g} = prox_{\lambda G}(\omega_{\text{mean}}\mathbf{1})$, where $\mathbf{1}$ is a vector of ones. Since splines form a partition of unity, the coefficient vector $\omega_{\text{mean}}\mathbf{1}$ generates a constant image with value ω_{mean}. Using $prox_{\lambda G}$, this vector is projected onto the interpolation space.

Figure 4 illustrates the peak signal-to-noise ratio (PSNR) for both starting strategies using 300 experiments on several images (128 × 128 pixels) and inpainting areas after 100 iterations. We use two types of inpainting areas: on the one hand 3% of the pixels are randomly set to zero (left), on the other hand 3 scratches with a 4 pixel width are used (right). The mean PSNR is calculated for cartoon-like images (top) and natural images (bottom) separately.

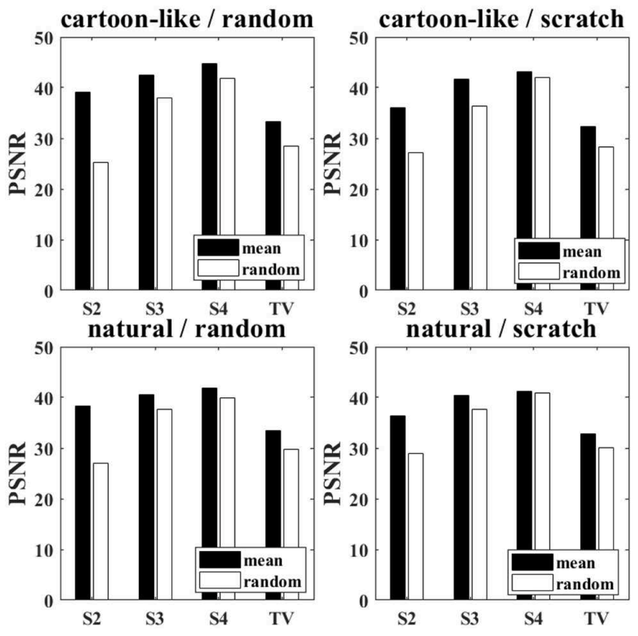

FIGURE 4 | Obtained PSNR values using random starting guess (white) or mean value (black). **(Left):** random inpainting area (3% of the pixels); **(Right):** scratches (3 scratches, 4 pixels width); top: cartoon-like images; bottom: natural images.

We see that the mean value starting guess performs better for all combinations of image and noise type. However, the effect seems to be stronger for a small spline order. Also, random missing pixel noise is more influenced by the starting guess strategy than scratch inpainting. In **Figure 5** a visual comparison of both reconstructions is given. The "Lena" image shown in **Figure 3** with randomly missing pixels is inpainted with splines of order 2; first with random starting guess and a second time with mean value. The obtained PSNR values are 26.21 and 31.80, respectively.

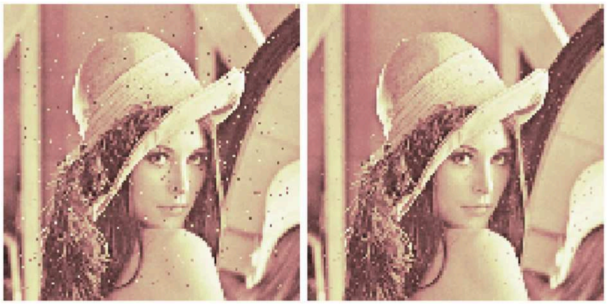

FIGURE 5 | Reconstruction of Lena (see **Figure 3**) using splines of order 2 with random starting guess **(left)** and mean value strategy **(right)**.

4.1.2. Spline Order

Figure 4 already makes clear that a higher spline order yields a better reconstruction. We analyze this in more details in the next experiment. Again, we calculate the mean PSNR over 300 runs using 128 × 128 pixel images. This time, we plot the PSNR against the percentage of unknown pixels (random inpainting area) or scratch width (scratch inpainting area) for both image types. **Figure 6** shows the obtained PSNR for different spline orders and standard TV as comparison. For both algorithms, we use the optimal starting guess just discussed.

We can clearly see that a higher spline order in all cases performs a better reconstruction. However, the difference is not as strong for small inpainting areas, such as the randomly missing pixels. Moreover, the effect dimishes when going from order 3 to 4 which yield nearly similar results. Note that all spline orders clearly outperform standard TV inpainting. Thus, we recommend to use at least splines of order 3.

Figure 7 gives an example how the reconstruction quality can increase if a higher order spline is used on natural images. The scratched image was reconstructed using splines of order 2 and 3 obtaining a PSNR of 33.83 and 38.93. As a comparison, the PSNR in case of standard TV is 30.67.

The runtime of our algorithm is shown in **Figure 8**. For all performed experiments the time in seconds is plotted against the size of the inpainting area in pixels. We ran the algorithm on Windows 10 with 2.81 Ghz and 8 GB Ram using Matlab R2018b. The graph shows a linear behavior for all spline orders

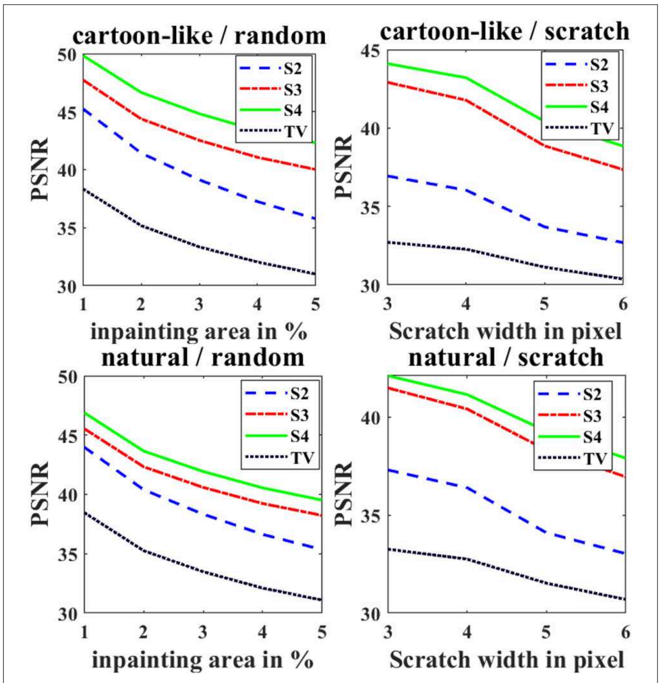

FIGURE 6 | Obtained PSNR values for different spline orders and standard TV. Left: random inpainting area; right: scratches; top: cartoon-like images; bottom: natural images.

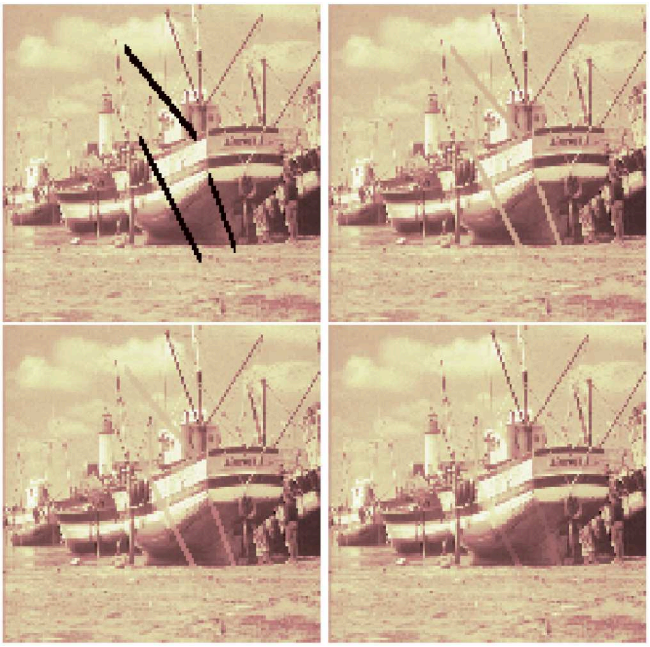

FIGURE 7 | Inpainting scratches: damaged image (top, left), reconstruction using standard TV inpainting (top, right), and reconstruction using spline inpainting of order 2 and 3 (bottom).

with growth depending on this order. However, for order 3 or higher the system matrix is created using the B-spline recursion formulae. This increases the runtime by a constant factor. Note that we implemented the algorithm in this way to be flexible in the spline order. When the order is fixed a direct creation of the system matrix can be used and, thus, the overall runtime can be reduced.

4.2. Numerical Examples: Inpainting Area

In this subsection we numerically demonstrate the reconstruction quality of our method for different images and varying inpainting areas. We follow the numerical setup of [13, Figure 3.1] for text removal and [15, Figures 5.5, 5.6] for hole filling and highly incomplete data. An exact comparison is not possible since the detailed setup in [13, 15] is not exactly specified, but the visual comparison shows similar reconstruction quality. However, our method does not need a triangulation of the image which does require a lot of extra memory [15]. We use images of size 256 × 256 pixels for all experiments in this subsection.

Our first example is the removal of unwanted text in images. In **Figure 9** we compare the performance of TV reconstruction and our spline approach of order 2 and 4 for an image covered by text. This inpainting example is especially hard to solve since the text covers many different regions of the image and crosses edges. TV inpainting is not designed to cover such situations. Using splines of order 2 we obtain an PSNR of 26.26. Increasing the spline order to 4 improves the PSNR to 32.79. The PSNR of standard TV inpainting is 26.69.

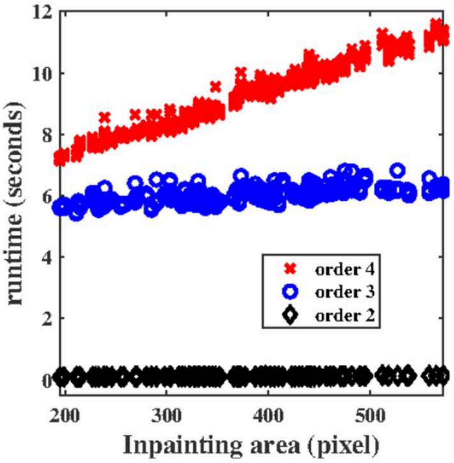

FIGURE 8 | Runtime of the algorithm for different spline orders plotted against the inpainting area in pixels.

Next, we consider the hole filling problem, i.e., a large inpainting area in the image. Naturally, this problem is considerably more difficult when the inpainting area crosses edges of the image as compared to the case when it does not. We give an example of both these scenarios. Moreover, we also combine our spline approach with a slightly different functional by using a squared TV penalization, i.e., (13) using squared absolute value. This functional should be more suitable for large inpainting areas. In **Figure 10** a monochromatic region

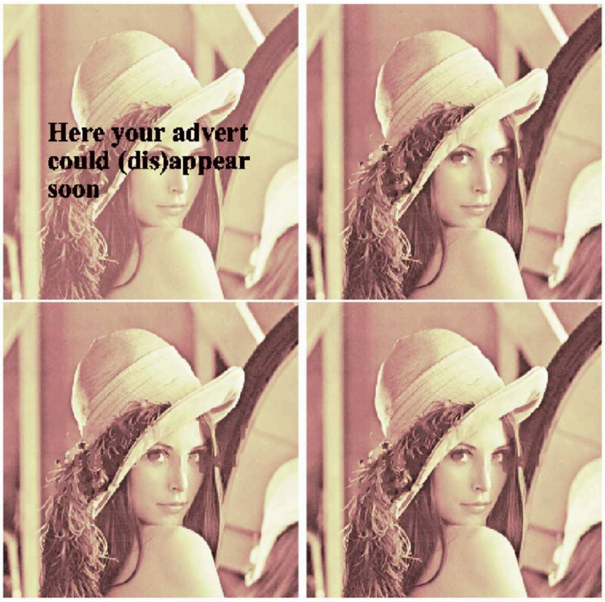

FIGURE 9 | Removing text from Lena: text image (top, left), TV inpainting (top, right), and spline inpainting using order 2 and 4 (bottom).

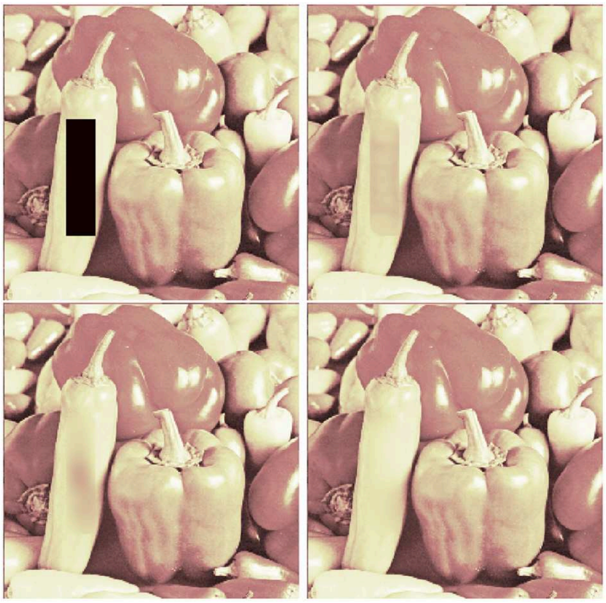

FIGURE 10 | Hole filling (smooth area): damaged image (top, left), reconstruction using standard TV inpainting (top, right) and reconstruction using spline inpainting of order 3 with TV norm (bottom, left), and squared TV norm (bottom, right).

in the peppers image is missing. We use splines of order 3 for inpainting. We obtain an PSNR of 27.97 using normal TV regularization and 41.93 using the squared penalty term. Standard TV inpainting achieves a PSNR of 26.40. An example of a more difficult scenario is given in **Figure 11** where parts of the phantom image have to be reconstructed. However, using our spline approach the reconstruction quality increases significantly. While standard TV inpainting only achieves a PSNR of 27, we obtain a PSNR of 32.62 with order 4 splines and even 36.29 using the squared functional. These two examples clearly demonstrate the advantages of our spline approach.

As a last experiment we consider the problem of highly incomplete data. We simulated the cases of 80% missing data (following [15, Figure 5.6]) and 95% missing data. The results are shown in **Figures 12**, **13**. We use splines of order 3 and again also test the squared TV functional. For 80% missing data we achieve PSNR values of 25.41 (spline inpainting) and 24.34 (spine inpainting with squared TV). Standard TV inpainting results in a PSNR of 24.89. However, the squared functional gets more stable when we increase the number of missing pixels. For 95% missing data we obtain PSNR values of 16.29 (standard TV), 17.81 (spline inpainting), and 18.76 (spline inpainting with squared TV). We assume that in all cases a spline order of 4 or higher may even obtain better results. However, as runtime and memory usage scale (linear) with the number of unknown pixels, computational costs for higher orders can increase significantly.

We observe from these experiments that for a large number of missing data, a low spline order may no longer increase the PSNR value. Thus, it is even more important to choose a spline order 3 or higher in these cases. Furthermore, more advanced functionals can improve the results.

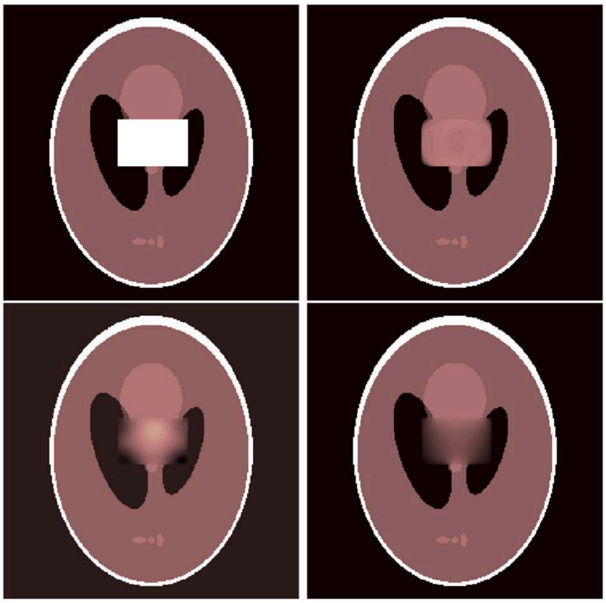

FIGURE 11 | Hole filling (including edges): damaged image (top, left), reconstruction using standard TV inpainting (top, right) and reconstruction using spline inpainting of order 4 with TV norm (bottom, left), and squared TV norm (bottom, right).

4.3. Numerical Examples: Noisy Data

As a last example, we consider an image that is corrupted by salt-and-pepper noise, i.e., some pixels are randomly set to the minimal or maximal value. Since this noise deletes

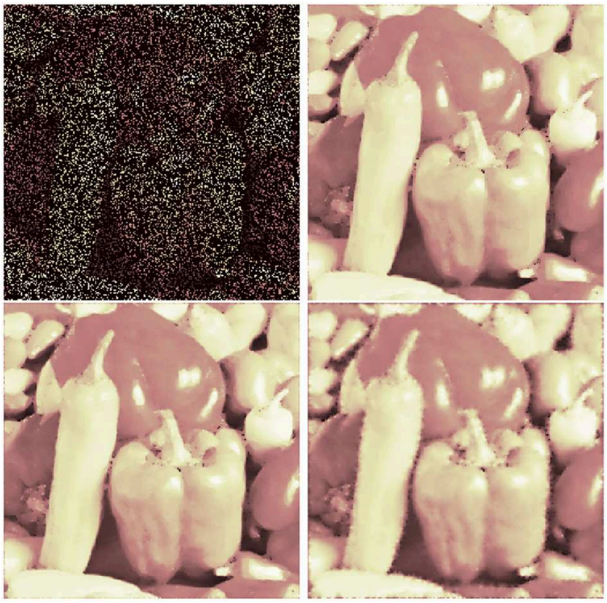

FIGURE 12 | Inpainting for 80% missing pixels: damaged image (top, left), standard TV inpainting (top, right) and spline inpainting of order 3 with TV norm (bottom, left), and squared TV norm (bottom, right).

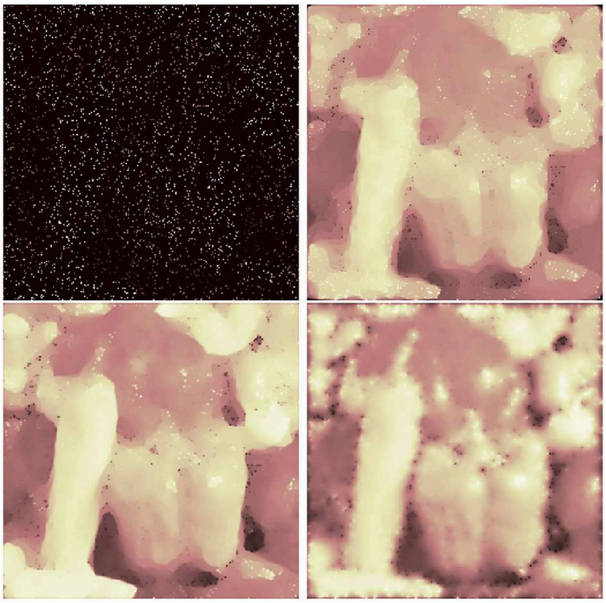

FIGURE 13 | Inpainting for 95% missing pixels: damaged image (top, left), standard TV inpainting (top, right) and spline inpainting of order 3 with TV norm (bottom, left), and squared TV norm (bottom, right).

all information about the original pixel value, we can model the reconstruction as inpainting problem where the inpainting region is given by all pixels with minimal or maximal value. This coincides with our random inpainting area model.

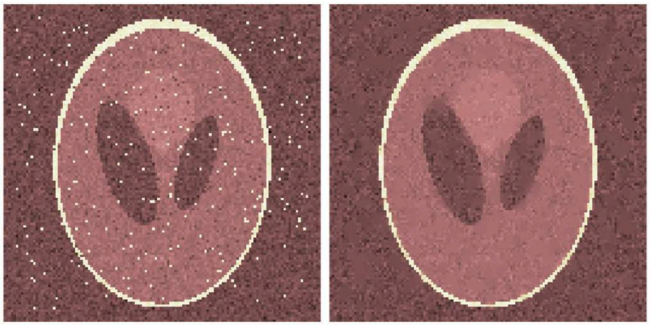

FIGURE 14 | Cartoon-like image corrupted by Gaussian and salt-and-pepper noise **(Left)** and its reconstruction using spline inpainting with order 3 **(Right)**.

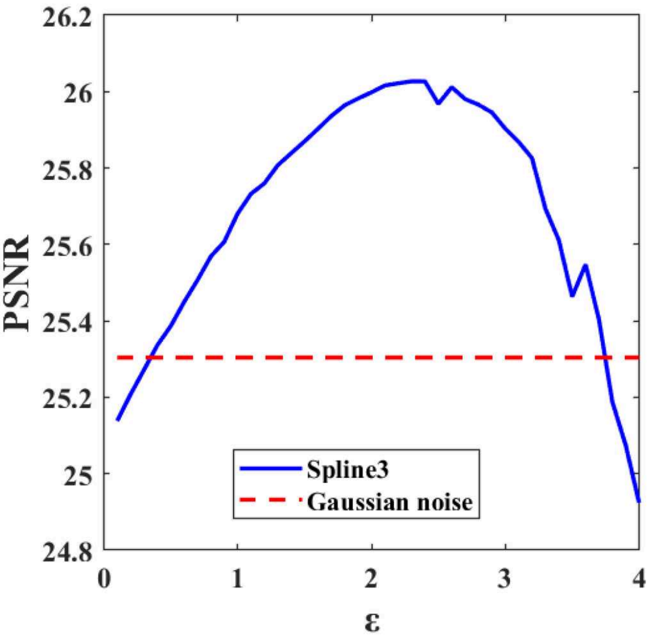

FIGURE 15 | PSNR value obtained by spline inpainting with order 3 and different parameters ε. The PSNR of the Gaussian noisy image is plotted as comparison.

So far, we discussed noiseless data outside the inpainting domain. Therefore, the operator G was chosen such that only interpolating solutions were allowed. However, when dealing with noisy images it is not unlikely that the image is corrupted by several types of noise. Hence, we now use the relaxed model (12). We can still use the primal-dual algorithm, but now with the operator $G(\mathbf{g}) = \frac{\varepsilon}{2}\|\mathbf{B}_{\Xi^*}\mathbf{g} - \mathbf{f}_{\Xi^*}\|_2^2$ and its proximity operator

$$\underset{\lambda G}{prox}(\mathbf{g}) = (\mathbf{I} + \lambda\varepsilon\mathbf{B}_{\Xi^*}^T\mathbf{B}_{\Xi^*})^{+}(\mathbf{g} + \lambda\varepsilon\mathbf{B}_{\Xi^*}^T\mathbf{f}_{\Xi^*}), \tag{41}$$

where $\mathbf{I}$ is the identity matrix. Note that we cannot use the relaxed model (12) to denoise the image far away from the inpainting domain since the TV minimization is only performed in a surrounding of the unknown pixels. Hence, for a good reconstruction we need to choose a higher spline order such that the spline support enlarges. Moreover, a large constant ε should be chosen, otherwise the solution of the relaxed model (12) is nearly constant. A simultaneous denoising may be obtained by expanding the TV term in model (12) to the complete image. However, this will drastically increase the number of Gauss-Legendre points used for the quadrature formula and, thus, increase the computational complexity.

Figure 14 shows the result of spline inpainting with order 3 using the relaxed model on an image that was corrupted by Gaussian and salt-and-pepper noise. The parameter ε is chosen such that the best PSNR is obtained. This is illustrated in **Figure 15** where the PSNR for different choices of ε is shown. We notice that for too small as well as too large ε the reconstruction quality decreases. **Figure 14** shows the reconstruction for $\varepsilon = 2.3$ which leads to an PSNR of 26.03. As a comparison, the PSNR of the noisy image is 18.33 in case of Gaussian and salt-and-pepper noise, and 25.30 if only Gaussian noise is considered, i.e., this is the PSNR outside the inpainting region.

5. CONCLUSION

We presented a new approach to model the discrete inpainting problem using TV regularization and splines. The spline order can be chosen adapted to the underlying image and inpainting area. The advantages of this method were demonstrated in numerical experiments. Especially, when the images are complex and/or the inpainting domains are large, the reconstruction quality can highly profit from the new concepts by choosing a higher spline order. But also for cartoon-like images and low spline orders the method returns reasonable results. As a slight disadvantage, the new methods requires precalculation of the spline basis, Greville abscissae and Gauss-Legendre points what increases the runtime of the algorithm.

Although the method increases the reconstruction quality, it is still based on TV minimization and, thus, cannot overcome the known issues of TV inpainting completely. Depending on the given situation other variations of the Spline Inpainting Models 1 should be used. As already mentioned in section 3.2, this can be useful if the considered functionals depend on the function and its derivatives. Moreover, we want to extend the method on more general domains. Depending on the image, it might not be necessary to interpolate every pixel which could be exploited by the use of non-uniform knots (and interpolation points) and/or the combination with hierarchical splines.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

Most of this research was done while the third author was with the University of Passau. At the Technical University of Munich the third author was partially supported by the Emmy Noether Research Group KR 4512/1-1 and by the DFG Collaborative Research Center TRR 109, Discretization in Geometry and Dynamics. The authors thank the reviewers for their comments which helped to improve our paper.

REFERENCES

1. Rudin LI, Osher S. Total variation based image restoration with free local constraints. In: *Proceedings of the International Conference in Image Processing* (Austin, TX). Vol. 1 (1994). p. 31–5.
2. Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D.* (1992) **60**:259–68.
3. de Boor C. *A Practical Guide to Splines*. Applied Mathematical Sciences. New York, NY: Springer New York (2001). Available online at: https://books.google.de/books?id=m0QDJvBI_ecC
4. Schumaker L. *Spline Functions: Basic Theory*. New York, NY: Wiley-Interscience (1980).
5. Bertalmio M, Vese L, Sapiro G, Osher S. Simultaneous structure and texture image inpainting. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* Vol. 2 (2003). **12**:882–89. doi: 10.1109/TIP.2003.815261
6. Efros A, Leung T. Texture Synthesis by Non-parametric Sampling. In: *In International Conference on Computer Vision* (Corfu) (1999). p. 1033–8.
7. Chan TF, Shen J. Nontexture inpainting by curvature-driven diffusions. *J Visual Commun Image Represent.* (2001) **12**:436–49. doi: 10.1006/jvci.2001.0487
8. Bertalmío M, Sapiro G, Caselles V, Ballester C. Image inpainting. In: *Proceedings of SIGGRAPH 2000* (New Orleans, LA) (2000).
9. Bornemann F, März T. Fast image inpainting based on coherence transport. *J Math Imaging Vision.* (2007) **28**:259–78. doi: 10.1007/s10851-007-0017-6
10. Schönlieb CB. *Modern PDE Techniques for Image Inpainting*. University of Cambridge (2009).
11. Chan TF, Shen J. Variational image inpainting. *Commun Pure Appl Math.* (2005) **58**:579–619. doi: 10.1002/cpa.20075
12. Chan TF, Shen J, Zhou HM. Total variation wavelet inpainting. *J Math Imaging Vision.* (2006) **25**:107–25. doi: 10.1007/s10851-006-5257-3
13. Hong QY. *Bivariate Splines for Image Enhancements based on Variational Models* (Athens, Georgia). University of Georgia (2011).
14. Getreuer P. Total variation inpainting using split Bregman. *Image Process.* (2012) **2**:147–57. doi: 10.5201/ipol.2012.g-tvi
15. Hong QY, Lai MJ, Messi LM. *The Rayleigh-Ritz Method for Total Variation Minimization Using Bivariate Splines Functions on Triangulation* (2014). available online at: http://alpha.math.uga.edu/~mjlai/papers/RayleighRitzTV.pdf
16. Cai JF, Chan RH, Shen Z. A framelet-based image inpainting algorithm. *Appl Comput Harm Anal.* (2008) **24**:131–49. doi: 10.1016/j.acha.2007.10.002
17. Hoffmann S, Plonka G, Weickert J. Discrete Green's functions for harmonic and biharmonic inpainting with sparse atoms. In: Tai XC, Bae E, Chan TF, Lysaker M, editors. *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Cham: Springer International Publishing (2015). p. 169–82.
18. Voronin V, Marchuk V, Sherstobitov A, Egiazarian K. Image inpainting using cubic spline based edge reconstruction. In: *Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II. Vol. 8295 of Proceedings of SPIE* (Burlingame, CA) (2012). p. 1–10.
19. Unser M. Splines: a perfect fit for signal and image processing. *IEEE Signal Process Mag.* (1999) **16**:22–38.

20. Achanta R, Arvanitopoulos N, Süsstrunk S. Extreme image completion. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing* (New Orleans, LA: ICASSP) (2017). p. 1333–7.

21. Chan TF, Shen J. Image processing and analysis: variational, PDE, wavelet, and stochastic methods. *Soc Ind Appl Math.* (2005). doi: 10.1137/1.9780898717877

22. Aubert G, Kornprobst P. *Mathematical Problems in Image Processing*. 2nd ed. New York, NY: Springer-Verlag New York (2006).

23. Chan TF, Shen J. Mathematical models for local nontexture inpaintings. *SIAM J Appl Math.* (2002) **62**:1019–43. doi: 10.1137/S0036139900368844

24. Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vision.* (2011) **40**:120–45. doi: 10.1007/s10851-010-0251-1

25. de Boor C. On uniform approximation by splines. *J Approx Theory.* (1968) **1**:219–35. Available online at: //www.sciencedirect.com/science/article/pii/0021904568900269

26. Mößner B, Reif U. Stability of tensor product B-splines on domains. *J Approx Theory.* (2008) **154**:1–19. doi: 10.1016/j.jat.2008.02.001

27. Höllig K, Hörner J. *Approximation and Modeling with B-splines*. Philadelphia, PA: SIAM (2013).

28. Peyre G. *Homepage* (2017). Available online at: http://www.gpeyre.com/ (accessed March 10, 2017).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Boßmann, Sauer and Sissouno. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.