Journal of Computational and Applied Mathematics 311 (2017) 551–564

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Image inpainting using reproducing kernel Hilbert space and Heaviside functions

Si Wang[a], Weihong Guo[b,*], Ting-Zhu Huang[a], Garvesh Raskutti[c]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, PR China

[b] Department of Mathematics, Case Western Reserve University, Cleveland, OH, 44106, USA

[c] Department of Statistics, University of Wisconsin-Madison, Madison, WI, 53706, USA

ARTICLE INFO

Article history:
Received 17 July 2015
Received in revised form 25 March 2016

Keywords:
Image inpainting
Reproducing kernel Hilbert space
Heaviside function
Edge

ABSTRACT

Image inpainting, a technique of repairing damaged images, is an important topic in image processing. In this paper, we solve the problem from an intensity function estimation perspective. We assume the underlying image is defined on a continuous domain and belongs to a space spanned by a basis of a reproducing kernel Hilbert space and some variations of the Heaviside function. The reproducing kernel Hilbert space is used to model the smooth component of the image while Heaviside function variations are used to model the edges. The coefficients of the redundant basis are computed by the discrete intensity at undamaged domain. We test the proposed model through various images. Numerical experiments show the effectiveness of the proposed method, especially in recovering edges.

1. Introduction

Image inpainting is a technique of repairing corrupted/missing image intensity in a visually plausible way [1–6]. The concept of digital image inpainting was first introduced in [7]. Image inpainting has gained a wide range of applications. Examples include restoration of ancient frescoes [8], removing texts and scratches, correcting red eye effects in photos and reducing artifacts [9]. Existing image inpainting methods can be classified into several categories: variational/partial differential equation (PDE) based [7,10–12]; sparsity based [13,14]; texture synthesis based [15]; exemplar based [16,17] and hybrid approaches [18,19]. Note that, these methods are not completely independent. For example, some sparsity based methods may belong to the category of exemplar based methods.

Variational/PDE based approaches inpaint images by diffusing local image structure from known regions into unknown inpainting regions. In [7], the authors smoothly propagated information from the surrounding areas in the direction of isophote lines into unknown regions. Subsequently, many variational models have been applied to image inpainting. For example, in [2], the authors proposed the curve driven diffusion (CDD) method and in [10], the authors introduced a variational framework to restore images by minimizing total variation (TV). An Euler's elastica energy based model was then discussed in [20] and this model can connect level lines over larger distances than TV based methods. In [11], the authors proposed two models which combine TV and wavelet. In [21], a joint interpolation of the image gray-levels and isophote directions was proposed for automatic inpainting in a variational framework. In [22], the authors used Cahn–Hilliard

[*] Corresponding author.
E-mail addresses: uestcsiwang@163.com (S. Wang), wxg49@case.edu (W. Guo), tingzhuhuang@126.com (T.-Z. Huang), raskutti@stat.wisc.edu (G. Raskutti).

http://dx.doi.org/10.1016/j.cam.2016.08.032
0377-0427/© 2016 Elsevier B.V. All rights reserved.

equation to inpaint binary images. The authors then generalized it for grayvalue images by using subgradients of TV functional within the flow and discussed the stationary state of the proposed model in [23]. The above methods need to solve PDEs and are sometimes time consuming. An operator splitting method [24] was then proposed to speed up TV inpainting algorithm. All these PDE based methods take advantage of the property that neighbored pixels are correlative and use local information around the inpainting domain. These algorithms perform well for non-texture images with relatively smaller missing regions.

Texture synthesis based methods are popular for images with textures. The idea is to synthesize new image pixels by learning from similar regions in a texture sample [15,25]. In [15], the authors presented an algorithm to synthesize new texture by taking patches of existing texture and stitching them together. However, they perform not that well when handling natural images that have complex interactions between structure and texture boundaries.

For images containing both structures and textures, hybrid methods are sometimes used. In [18] for instance, texture synthesis method and variational/PDE are combined. It decomposes an image into structure and texture parts and reconstructs the structure part by using a PDE based algorithm and the texture part through a texture synthesis algorithm. It is still hard to reconstruct large structure regions.

In order to reconstruct large missing/corrupted regions, some exemplar based methods are proposed. Exemplar based inpainting methods combine texture synthesis method and isophote driven inpainting method. An exemplar-based inpainting method by using a priority value which determines the fill order for each patch was introduced in [16]. This technique can propagate both linear structure and texture into inpainting domain but has some difficulties in handling curved structures. In [26], the authors proposed a novel method by using the sparsity at the patch level for modeling patch priority and patch representation. Many other exemplar-based methods are introduced in [27–30].

Sparsity based inpainting methods assume the underlying image (or the patch) is sparse under a given transform, such as discrete cosine transform, or wavelet. For instance, the authors used sparse representation over a redundant dictionary technique to fill the hole patch-wisely in [14]. This kind of methods may introduce smooth effect when filling large missing regions.

Some of above methods focus on maintaining structure of the inpainting areas and some concern about texture synthesis. In this paper, we focus on maintaining structure of images. Different from the above mentioned methods, the proposed inpainting approach is based on a reproducing kernel Hilbert space (RKHS) and its extensions. RKHS has been used as a powerful tool in machine learning, but not much in image processing [31–33]. So far, RKHS has only been explored in denoising [34], colorization [35] and segmentation [36]. In [34], the authors proposed an adaptive kernel method to deal with image denoising problems. They treat additive noise in a unified framework using RKHS. It can preserve sharp edges. However, this method needs expensive computation. Inspired by the extensions of RKHS in machine learning, the authors in [35] used a well-known least square regression in RKHS for colorization problems. So far, we have not found RKHS based method for image inpainting. In this paper, we assume that the underlying image is defined on a continuous domain and belongs to a space spanned by a basis of a RKHS and variations of Heaviside function. The coefficients of the redundant basis are computed by the discrete intensity at undamaged domain. Our experiments show the effectiveness of the proposed method.

This paper is organized as follows. In Section 2, we briefly review RKHS and its applications in one-dimension (1-D) signal and two-dimension (2-D) image smoothing. In Section 3, we present a new model for image inpainting problem and discuss the numerical algorithm. In Section 4, some experimental results are presented. Finally, we conclude this work in Section 5.

2. Review on RKHS

We will use splines based RKHS to model the continuous component of images, so we review RKHS, 1D and 2D splines as well as their applications in image smoothing.

2.1. Reproducing Kernel Hilbert space

Definition 1 (*Gram Matrix*)**.** Let χ be a set. Given a function $\kappa : \chi \times \chi \to \mathbb{R}$ and $x_1, \ldots, x_N \in \chi$, the matrix $K = (K_{i,j})^N$ with elements $K_{i,j} = \kappa(x_i, x_j)$, for $i, j = 1, \ldots, N$, is called the Gram matrix (or kernel matrix) of κ with respect to $x_1, \ldots, x_N$.

Definition 2 (*Positive Semidefinite Kernel*)**.** Let χ be a nonempty set. A function $\kappa : \chi \times \chi \to \mathbb{R}$ is called a positive semidefinite kernel, if for all $N \in \mathbb{N}$ and all $x_1, \ldots, x_N \in \chi$, Gram matrix K is semidefinite.

Given a subset $\chi \subset \mathbb{R}$ and a probability measure P on χ, we consider a Hilbert space with a family of real value functions $f : \chi \to \mathbb{R}$, with $\|f\|_{L^2(P)} < \infty$, and a corresponding inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ under which $\mathcal{H}$ is complete. The space $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), if there exists a kernel $\kappa : \chi \times \chi \to \mathbb{R}$ with the following two properties: (a) for each $x \in \chi$, $\kappa(x, \cdot)$ belongs to the Hilbert space $\mathcal{H}$, and (b) κ has the so called reproducing property, i.e. $f(x) = \langle f, \kappa(x, \cdot)\rangle$, for all $f \in \mathcal{H}$, in particular $\kappa(x, y) = \langle\kappa(x, \cdot), \kappa(y, \cdot)\rangle_{\mathcal{H}}$. Any such kernel function must be positive semidefinite.

Wahba proposed two splines based RKHS for smoothing problems in [37]. Our proposed model is partially motivated from it. In what follows, we review 1-D and 2-D spline based RKHS [34,37,38].

2.2. 1-D spline

For a real-valued function f on $[0, 1]$ with $m - 1$ continuous derivatives and $f^{(m)} \in \mathcal{L}_2[0, 1]$, it can be expanded at $t = 0$ by Taylor series

$$f(t) = \sum_{v=0}^{m-1} \frac{t^v}{v!} f^{(v)}(0) + \int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!} f^{(m)}(u)du = f_0(t) + f_1(t), \tag{2.1}$$

with $f_0(t) = \sum_{v=0}^{m-1} \frac{t^v}{v!} f^{(v)}(0)$ and $f_1(t) = \int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!} f^{(m)}(u)du$, where $(x)_+ = x$ for $x \geq 0$ and $(x)_+ = 0$ otherwise.

Let $\phi_v(t) = t^{v-1}/(v-1)!$ for $v = 1, \ldots, m$, $\mathcal{H}_0 = span\{\phi_1, \ldots, \phi_m\}$, then $f_0 \in \mathcal{H}_0$ can be expanded by $\{\phi_j\}_{j=1}^m$. $\mathcal{H}_0$ is a Hilbert space with the square norm $\|\phi\|^2 = \sum_{v=0}^{m-1} [(D^v \phi)(0)]^2$. From Chapter 1 in [37], we know that $\mathcal{H}_0$ is a reproducing kernel Hilbert space (RKHS) with reproducing kernel (r.k.) $R^0(s, t) = \sum_{v=1}^m \phi_v(s)\phi_v(t)$.

Let $\mathcal{B}_m$ denote the class of functions satisfying the boundary conditions $f^{(v)}(0) = 0$, for $v = 0, 1, \ldots, m-1$, $\mathcal{H}_1 = \{f : f \in \mathcal{B}_m, f, f', \ldots, f^{(m-1)}$absolutely continuous$, f^{(m)} \in \mathcal{L}_2\}$, then $f_1 \in \mathcal{H}_1$. $\mathcal{H}_1$ is a Hilbert space with the square norm $\|f\|^2 = \int_0^1 (f^m(t))^2 dt$. It has shown in [37], $\mathcal{H}_1$ is a RKHS with r.k. $R^1(s, t) = \int_0^1 G_m(s, u)G_m(t, u)du$, where $G_m(t, u) = (t-u)_+^{m-1}/(m-1)!$. $\{\xi_i = R^1(s_i, \cdot)\}_{i=1}^n$ forms a basis of $\mathcal{H}_1$ and $f_1 \in \mathcal{H}_1$ can be expanded by it.

The author in [37] then constructs a Sobolev–Hilbert space $\mathcal{H}_R : \mathcal{H}_R[0, 1] = \{f : f, f', \ldots, f^{(m-1)}$ absolutely continuous, $f^{(m)} \in \mathcal{L}_2\}$. For each function $f \in \mathcal{H}_R$, it has a unique decomposition

$$f = f_0 + f_1,$$

with $f_0 \in \mathcal{H}_0$ and $f_1 \in \mathcal{H}_1$. If we endow $\mathcal{H}_R$ with the square norm $\|f\|^2 = \sum_{v=0}^{m-1} [(D^v f)(0)]^2 + \int_0^1 (D^m f)^2(u)du$, then $\mathcal{H}_0$ and $\mathcal{H}_1$ will be perpendicular and r.k. for $\mathcal{H}_R$ is $R(s, t) = R^0(s, t) + R^1(s, t) = \sum_{v=1}^m \phi_v(s)\phi_v(t) + \int_0^1 G_m(s, u)G_m(t, u)du$. From the above discussion, we can get that $f(t)$ can be expressed as $f(t) = \sum_{v=1}^m d_v \phi_v(t) + \sum_{i=1}^n c_i \xi_i(t)$.

The basic data model for smoothing problems can be expressed as:

$$\overrightarrow{g} = \overrightarrow{f} + \epsilon, \tag{2.2}$$

where $\overrightarrow{f} = (f(t_1), \ldots, f(t_n))', f(t_i)$ is an intensity value of f at $t_i \in [0, 1], i = 1, 2, \ldots, n$, $\overrightarrow{g}$ is a noisy observation signal and ϵ represents an additive noise. From [37], we get that the solution of (2.2) is $\overrightarrow{f} = Td + \Sigma c$, where T is a $n \times m$ matrix with $T(i, j) = \phi_j(t_i)$, Σ is a $n \times n$ matrix with $\Sigma(i, j) = \{\langle \xi_i, \xi_j \rangle\}$, $\xi_i = R^1(s_i, \cdot)$, $d = (d_1, d_2, \ldots, d_m)'$, $c = (c_1, c_2, \ldots, c_n)'$. c, d are obtained by minimizing:

$$\frac{1}{n} \|\overrightarrow{g} - Td - \Sigma c\|^2 + \lambda c' \Sigma c, \tag{2.3}$$

then $c = M_1^{-1}(I - T(T'M_1^{-1}T)^{-1}T'M_1^{-1})\overrightarrow{g}$, $d = (T'M_1^{-1}T)^{-1}T'M_1^{-1}\overrightarrow{g}$, where $M_1 = \Sigma + n\lambda I$ and Σ is the same with the above definition.

Once we get c, d from (2.3), we can estimate the signal function $f(t)$ through

$$f(t) = \sum_{v=1}^m d_v \phi_v(t) + \sum_{i=1}^n c_i \xi_i(t)$$

for any $t \in [0, 1]$. Next, we will review 2-D thin-plate spline.

2.3. 2-D thin-plate spline

In this section, we present some basic theoretical foundations for the thin-plate spline and its application for smoothing problems. Further details can be found in [39].

Let f be an intensity function defined on a continuous domain E^2. Without loss of generality, we assume $E^2 = [0, 1] \times [0, 1]$. Assume $f \in \mathcal{K}$, a space of functions whose partial derivation of total order m being in $\mathcal{L}_2(E^2)$. Let $\overrightarrow{f} = (f(t_1), \ldots, f(t_n))$ be discrete form of f on grids $t_i = (x_i, y_i) \in E^2$, then the noisy image of vector form can be described as:

$$\overrightarrow{g} = \overrightarrow{f} + \epsilon,$$

where $\overrightarrow{f}$ is the original image and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)'$ represents an additive noise. Under two-dimensional condition, the thin-plate penalty functional is defined as following

$$J_m(f) = \sum_{v=0}^m \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C_m^v \left(\frac{\partial^m f}{\partial x^v \partial y^{m-v}} \right)^2 dxdy,$$

where notation C means the combination formula, m is a parameter related to the degree of the polynomials. We can get an optimal estimation of f for spline smoothing problem by minimizing the following variational problem

$$\frac{1}{n}\|\overrightarrow{g}-\overrightarrow{f}\|^2+\lambda J_m^d(f), \tag{2.4}$$

where d is the dimension of f. As we focus on 2-D images, then $d = 2$. From Chapter 2 in [37], we know that the null space of the penalty J_m^d is consisted of $M = C_{d+m-1}^d$ polynomials of degree no more than $m - 1$. In the experiments, we let $m = 2$ (higher-order of m leads to similar results), then the null space of J_m^d can be spanned by $\phi_1(x, y) = 1$, $\phi_2(x, y) = x$, $\phi_3(x, y) = y$. Duchon [39] showed that, if $t_1, \dots, t_n$ are such that least square regression on $\phi_1, \dots, \phi_M$ is unique, then (2.4) has a unique minimizer f, with representation

$$f(t)=\sum_{v=1}^{M}d_v\phi_v(t)+\sum_{i=1}^{n}c_iE_m(t,t_i), \tag{2.5}$$

where $t \in [0, 1] \times [0, 1]$, E_m is a Green's function for the m-iterated Laplacian defined as: $E(\tau) = \theta_{m,d}|\tau|^{2m-d}\ln|\tau|$, and $\theta_{m,d} = \frac{(-1)^{d/2+1+m}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!}$ (see [37,40] for more details). Let $|t - t_i| = (\sum_{j=1}^d (x_j - x_j(i))^2)^{1/2}$, $E_m(s, t)$ is given by $E_m(s, t) = E(|s - t|)$. When $s = t$, we let $E_m(s, s) = 0$. Model (2.4) can be rewritten as:

$$\frac{1}{n}\|\overrightarrow{g}-Td-Kc\|^2+\lambda c'Kc, \tag{2.6}$$

where K is the $n \times n$ matrix with the ijth entry $K(i, j) = E_m(t_i, t_j)$, T is the $n \times M$ matrix with $T(i, j) = \phi_j(t_i)$. c, d can be obtained by minimizing (2.6), then we get the solution of (2.4) through (2.5).

Once we get c and d from (2.6), we can estimate function $f(s)$ on E^2 through

$$f(s)=\sum_{v=1}^{M}d_v\phi_v(s)+\sum_{i=1}^{n}c_iE_m(s,t_i), \tag{2.7}$$

for any $s \in E^2$.

3. The proposed method

Model (2.6) is for image smoothing. It is pretty straightforward to apply it to image inpainting. In what follows, we will show that splines based RKHS itself is not enough to inpaint images with high quality. It leads to over smoothness and sometimes blur results. We propose to combine thin-plate splines and variations of the Heaviside function to inpaint images.

Let f be an intensity function defined on a continuous domain $E^2 = [0, 1] \times [0, 1]$ and $\overrightarrow{f}$ represent discrete form of f on grids $t_i = (x_i, y_i) \in E^2$. We assume the size of $\overrightarrow{f}$ is $n \times q$. For simplicity, we use $\overrightarrow{f}$ to interchangeably represent a matrix and its vectorized version. The discrete formula of image inpainting problem can be formulated as

$$\overrightarrow{g}_i=\begin{cases}\overrightarrow{f}_i+\epsilon_i, & i\in\Lambda,\\ \overrightarrow{v}_i, & i\in\Lambda^c,\end{cases} \tag{3.1}$$

where $\overrightarrow{g}$ is the observed corrupted image, $\overrightarrow{f}$ is the original image, ϵ is i.i.d. additive Gaussian white noise, and Λ is the index set of certain pixels of the image [41]. The index set Λ^c is called inpainting domain. Let the number of elements in Λ be n_1, $\Lambda = \{\Lambda_1, \Lambda_2, \dots, \Lambda_{n_1}\}$. Image inpainting is to restore $\overrightarrow{f}$ from the known data $\overrightarrow{g}$. The ith pixel value of $\overrightarrow{f}$ is given by $\overrightarrow{f}_i = f(t_i)$, where $t_i = (x_i, y_i)$, $x_i \in \{1/n, 2/n, \dots, 1\}$, $y_i \in \{1/q, 2/q, \dots, 1\}$.

A natural extension of (2.6) for inpainting is to minimize the following energy:

$$\frac{1}{n_1}\|\overrightarrow{g}|_\Lambda-Td-Kc\|^2+\lambda c'Kc, \tag{3.2}$$

where $K \in \mathrm{R}^{n_1\times n_1}$ with $K(i, j) = E_m(t_{\Lambda_i}, t_{\Lambda_j})$, $T \in \mathrm{R}^{n_1\times M}$ with $T(i, j) = \phi_j(t_{\Lambda_i})$. Let $N = nq$, $\widetilde{K} \in R^{N\times n_1}$ with $\widetilde{K}(i, j) = E_m(t_i, t_j)$ and $\widetilde{T} \in R^{N\times M}$ with $\widetilde{T}(i, j) = \phi_j(t_i)$, then the restored image $\overrightarrow{f}$ is given by

$$\overrightarrow{f}=\widetilde{K}c+\widetilde{T}d=\widetilde{K}M_1^{-1}(I-T(T'M_1^{-1}T)^{-1}T'M_1^{-1})\overrightarrow{g}|_\Lambda+\widetilde{T}(T'M_1^{-1}T)^{-1}T'M_1^{-1}\overrightarrow{g}|_\Lambda, \tag{3.3}$$

where $M_1 = K + n_1\lambda I$. We test model (3.2) on a simple synthetic image. In Fig. 3.3(a)–(c), we show the ground truth image, the damaged image (the intensity at the middle rectangular region is missing) and the restored image through model (3.2) respectively. From the inpainting result Fig. 3.3(c), we can see that edges are not recovered so well. The main reason is that splines based RKHS is good at modeling smooth components of images but not edges.

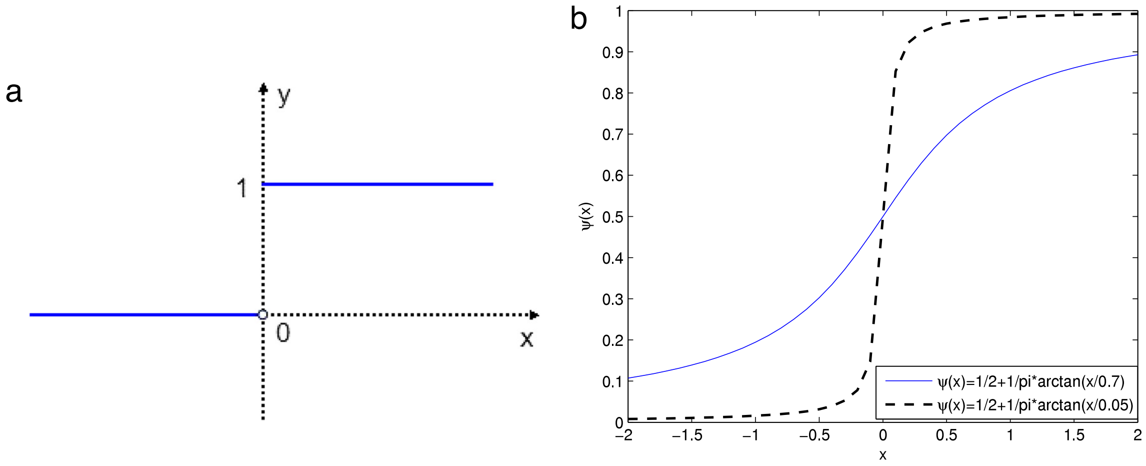


Fig. 3.1. (a) 1D Heaviside function; (b) Two approximated Heaviside functions with $\xi = 0.7$ (blue solid line) and $\xi = 0.05$ (black dash line), respectively. Figures are better visualized in the color file that is available on the web version of this article.

In order to recover edges, we use variations of the Heaviside function to model edges with various directions at various locations. We then combine splines based RKHS and these variations of Heaviside function to inpaint images. The Heaviside (step) function is defined as

$$\phi(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases}$$

An approximation to the Heaviside function ϕ is $\psi(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(\frac{x}{\xi})$ which approaches $\phi(x)$ when $\xi \to 0$. In Fig. 3.1, we show examples of ψ for two different ξ values. One can see that ξ controls the smoothness. To model one dimensional jumps, small ξ value should be adopted. We choose $\xi = 10^{-3}$ for numerical experiments. To model two dimensional step edges with various angles and locations, we define $\psi_{k_i}(x, y) = \psi(\cos\theta_k x + \sin\theta_k y + c_i)$, $i = 1, \dots, n_1$ [34]. Theoretically, the more ψ_{k_i} the better, but it would increase computation time significantly. In our experiments, we let $\theta_k \in \{\pi/8, 2\pi/8, 3\pi/8, \dots, 15\pi/8\}$ while $c_i \in \{s/n_1, 2s/n_1, 3s/n_1, \dots, 1\}$ for an image with n_1 known pixels, where s controls the resolution. For instance, for an image with n_1 known pixels and $s = 4$, $c_i \in \{4/n_1, 2 * 4/n_1, 3 * 4/n_1, \dots, 1\}$. Some examples of ψ_{k_i} are shown in Fig. 3.2.

Under the assumption that the underlying image intensity function is the sum of smooth components and edges, the underlying discrete image is given by

$$\overrightarrow{f} = \widetilde{K}c + \widetilde{T}d + \widetilde{\Psi}\beta \tag{3.4}$$

where $\widetilde{K}$ and $\widetilde{T}$ are defined similarly as in (3.3), $\widetilde{\Psi} \in \mathrm{R}^{N\times(16n_1/s)}$ with $\widetilde{\Psi}(i,j) = \psi_j(t_i)$. For simplicity, let $p = \frac{16n_1}{s}$. Note that the set $\{\psi_k\}$ is pretty exhaustive and edges contained in an image are much more sparse, it is thus reasonable to expect coefficients of $\{\psi_{k_i}\}_{k_i=1}^{p}$ to be sparse. We achieve this by a ℓ_1 regularity on β. Technically, ℓ_0 norm should be used but it is not convex. We use ℓ_1 norm instead. The proposed nonlinear model is

$$\min_{c,d,\beta}\left\{\frac{1}{n_1}\|\overrightarrow{g}|_{\Lambda} - Td - Kc - \Psi\beta\|^2 + \lambda_1 c'Kc + \mu\|\beta\|_1\right\}, \tag{3.5}$$

where $K(i,j) = E_m(t_{\Lambda_i}, t_{\Lambda_j})$, $T(i,j) = \phi_j(t_{\Lambda_i})$, $\Psi(i,j) = \psi_j(t_{\Lambda_i})$. Similar to [42–44], we first transform (3.5) to a linear constrained problem of the form

$$\min_{c,d,\beta,\omega}\left\{\frac{1}{n_1}\|\overrightarrow{g}|_{\Lambda} - Td - Kc - \Psi\beta\|^2 + \lambda_1 c'Kc + \mu\|\omega\|_1 : \omega = \beta\right\}. \tag{3.6}$$

We apply ADMM [45] to solve the above problem. Let augmented Lagrangian function of (3.6) be denoted by

$$\mathcal{L}_A(c, d, \beta, \omega, \lambda) = \frac{1}{n_1}\|\overrightarrow{g}|_{\Lambda} - Td - Kc - \Psi\beta\|^2 + \lambda_1 c'Kc + \mu\|\omega\|_1 + \alpha\|\omega - \beta\|^2 - \lambda^T(\omega - \beta), \tag{3.7}$$

where $\lambda \in \mathbb{R}^p$, and α is a penalty parameter. The iterative scheme of (3.6) is given by

$$\begin{cases} (c, d, \beta)^{k+1} = \arg\min\limits_{c,d,\beta} \mathcal{L}_A(c, d, \beta, \omega^k, \lambda^k), \\ \omega^{k+1} = \arg\min\limits_{w} \mathcal{L}_A(c^{k+1}, d^{k+1}, \beta^{k+1}, \omega, \lambda^k), \\ \lambda^{k+1} = \lambda^k - 2\gamma\alpha(\omega^{k+1} - \beta^{k+1}), \end{cases} \tag{3.8}$$

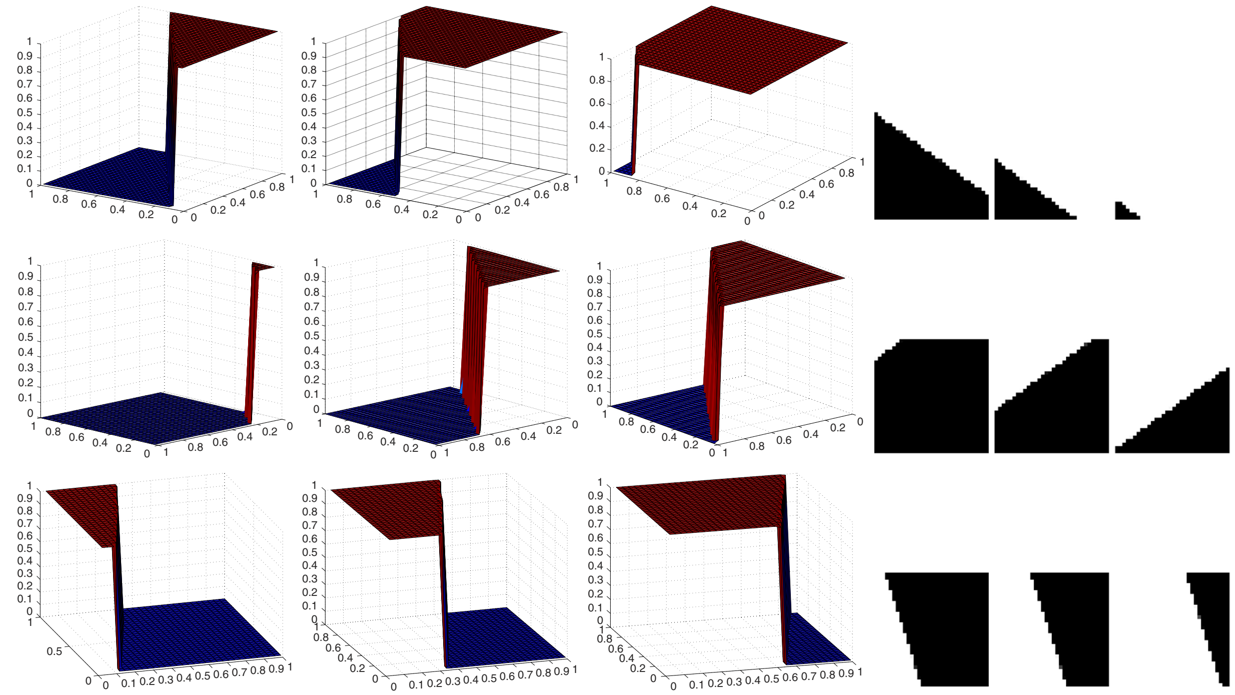


Fig. 3.2. Left panel: surface images of ψ under $\xi = 10^{-3}$ and nine parameter pairs (θ, c); right panel: the corresponding 2D intensity images. Some surface images are slightly rotated for better visualization.

where γ is a relaxation parameter. The (c, d, β)-subproblem of (3.8) is equivalent to

$$\min_{c,d,\beta} Q(c,d,\beta) = \left\{ \frac{1}{n_1} \|\overrightarrow{g}|_\Lambda - Td - Kc - \Psi\beta\|^2 + \lambda_1 c'Kc + \alpha \left\| \omega^k - \beta - \frac{\lambda^k}{2\alpha} \right\|^2 \right\}. \tag{3.9}$$

Let $\triangledown Q(c, d, \beta) = 0$, we get that

$$\begin{pmatrix} K'K & K'T & K'\Psi \\ T'K & T'T & T'\Psi \\ \Psi'K & \Psi'T & \Psi'\Psi \end{pmatrix} \begin{pmatrix} c \\ d \\ \beta \end{pmatrix} + \begin{pmatrix} n_1\lambda_1 Kc \\ 0 \\ \alpha n_1 \beta \end{pmatrix} = \begin{pmatrix} K'\overrightarrow{g}|_\Lambda \\ T'\overrightarrow{g}|_\Lambda \\ \Psi'\overrightarrow{g}|_\Lambda + \alpha n_1 \left(\omega^k - \frac{\lambda^k}{2\alpha} \right) \end{pmatrix}. \tag{3.10}$$

This is equivalent to solving the following three equations together:

$$(K'K + n_1\lambda_1 K)c + K'Td + K'\Psi\beta = K'\overrightarrow{g}|_\Lambda, \tag{3.11}$$

$$T'Kc + T'Td + T'\Psi\beta = T'\overrightarrow{g}|_\Lambda, \tag{3.12}$$

$$\Psi'Kc + \Psi'Td + (\Psi'\Psi + \alpha n_1 I)\beta = \Psi'\overrightarrow{g}|_\Lambda + \alpha n_1 \left(\omega^k - \frac{\lambda^k}{2\alpha} \right). \tag{3.13}$$

From Eq. (3.13), we get that

$$\beta = (\Psi'\Psi + \alpha n_1 I)^{-1} \left(\Psi'\overrightarrow{g}|_\Lambda + \alpha n_1 \left(\omega^k - \frac{\lambda^k}{2\alpha} \right) - \Psi'Kc - \Psi'Td \right), \tag{3.14}$$

then we substitute β in Eqs. (3.11) and (3.12) by (3.14), the solution of Eq. (3.10) is

$$\begin{aligned} c &= (A_1 - B_1 B_2^{-1} A_2)^{-1} (c_1 - B_1 B_2^{-1} c_2), \\ d &= B_2^{-1}(c_2 - A_2 c), \\ \beta &= (\Psi'\Psi + \alpha n_1 I)^{-1} \left(\Psi'\overrightarrow{g}|_\Lambda + \alpha n_1 \left(\omega^k - \frac{\lambda^k}{2\alpha} \right) - \Psi'Kc - \Psi'Td \right), \end{aligned}$$

where

$$
\begin{aligned}
A_1 &= (K'K + n_1\lambda_1 K) - K'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\Psi' K,\\
A_2 &= T'K - T'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\Psi' K,\\
B_1 &= K'T - K'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\Psi' T,\\
B_2 &= K'\overrightarrow{g}|_\Lambda - K'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\left(\Psi'\overrightarrow{g}|_\Lambda + \alpha n_1\left(\omega^k - \frac{\lambda^k}{2\alpha}\right)\right),\\
c_1 &= T'T - T'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\Psi' T,\\
c_2 &= T'\overrightarrow{g}|_\Lambda - T'\Psi(\Psi'\Psi + \alpha n_1 I)^{-1}\left(\Psi'\overrightarrow{g}|_\Lambda + \alpha n_1\left(\omega^k - \frac{\lambda^k}{2\alpha}\right)\right).
\end{aligned}
$$

It is easy to see that the minimizer ω^{k+1} is given by:

$$
\begin{aligned}
\omega^{k+1} &= \arg\min_{\omega} \mu\|\omega\|_1 + \alpha\left\|\omega - \beta^{k+1} - \frac{\lambda^k}{2\alpha}\right\|^2 &(3.15)\\
&= \max\left\{\left|\beta^{k+1} + \frac{\lambda^k}{2\alpha}\right| - \frac{\mu}{2\alpha}, 0\right\} sign\left(\beta^{k+1} + \frac{\lambda^k}{2\alpha}\right) &(3.16)
\end{aligned}
$$

where all operations are done component-wisely, and $sign(\cdot)$ represents the signum function.

The proposed algorithm is laid out below:

```
Algorithm 1.
Inputs: image $\overrightarrow{g}$ to be inpainted, model parameters $\mu, \lambda_1, \alpha > 0$ and $\gamma \in (0, (\sqrt{5}+1)/2)$.
Initialization: $\omega = \omega^0, \lambda = \lambda^0$ and set $k = 0$.
While "not converged", do
(1) compute $c^{k+1}$, $d^{k+1}$, and $\beta^{k+1}$ through (3.10),
(2) compute $\omega^{k+1}$ through (3.15),
(3) update $\lambda^k$ by $\lambda^{k+1} = \lambda^k - 2\gamma\alpha(\omega^{k+1} - \beta^{k+1})$,
(4) $k \leftarrow k + 1$.
End Do
(5) compute $\overrightarrow{f}$ through (3.4).
```

Fig. 3.3(d) shows the restored image obtained by Algorithm 1. From the result, we can see that the proposed model can keep edges better than model (3.2) as shown in Fig. 3.3(c).

The convergence of Algorithm 1 is giving by the following theorem, the proof of which can be found in [46].

Theorem 1. *For any $\alpha > 0$ and $\gamma \in (0, (\sqrt{5}+1)/2)$, the sequence $\{(c^k, d^k, \beta^k)\}$ generated by Algorithm 1 from any starting point (ω^0, λ^0) converges to a solution of (3.5).*

It is very straightforward to extend the above algorithm to color images. We process three channels separately. The details are omitted here.

4. Numerical results

In this section, we apply the proposed algorithm on various images and compare with some competitive approaches. For quantitative comparison, we use mean squared error (MSE) and peak signal to noise ratio (PSNR):

$$\mathrm{MSE} = \frac{1}{N}\|I - J\|_F^2,$$

and

$$\mathrm{PSNR} = 20\log_{10}\frac{\mathrm{MAX}_I}{\sqrt{\mathrm{MSE}}},$$

where I,J, N and MAX_f are the ground truth image, the recovered image, the number of pixels of the image, and the maximum possible pixel value of the image, respectively. For color images with three channels, the definition of PSNR is the same except that

$$\mathrm{MSE} = \frac{1}{3N}\|I - J\|_F^2.$$

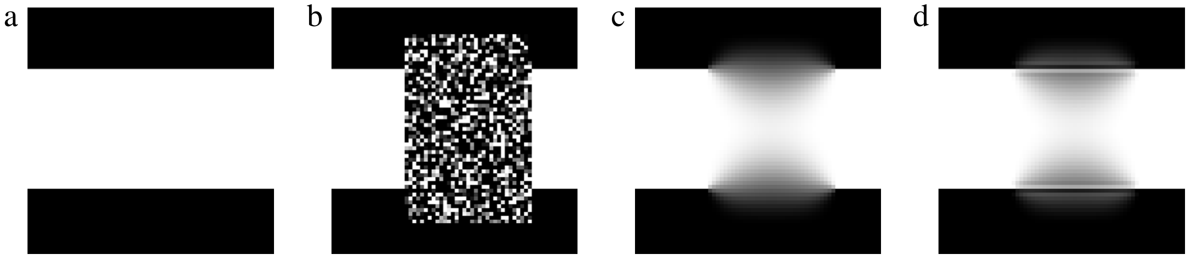


Fig. 3.3. (a) The ground truth image; (b) the observed image (random values are assigned for better visualization in the missing region); (c) the restored image by model (3.2); (d) the restored image by the proposed Algorithm 1.

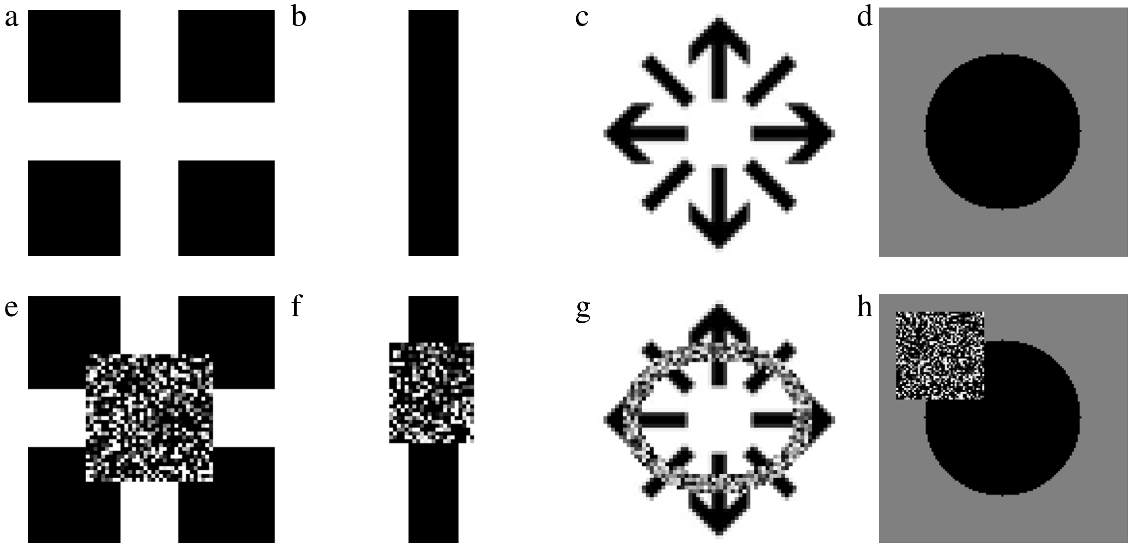


Fig. 4.1. The first row: ground truth images with size 64 × 64, 64 × 64, 64 × 64, 128 × 128. The second row: the observed images.

In all experiments, we let parameters $\alpha = 0.1$, $\mu = 10^{-5}$, and $\gamma = 1$ for all testing images. λ_1 is set to be 10^{-10} for images with size 64 × 64 and 10^{-12} for images with size 128 × 128. We terminate Algorithm 1 when the relative error between two consecutive iterates is smaller than a tolerance, i.e.

$$\frac{\|\omega^{k+1} - \omega^k\|_2}{\|\omega^k\|_2} < tol.$$

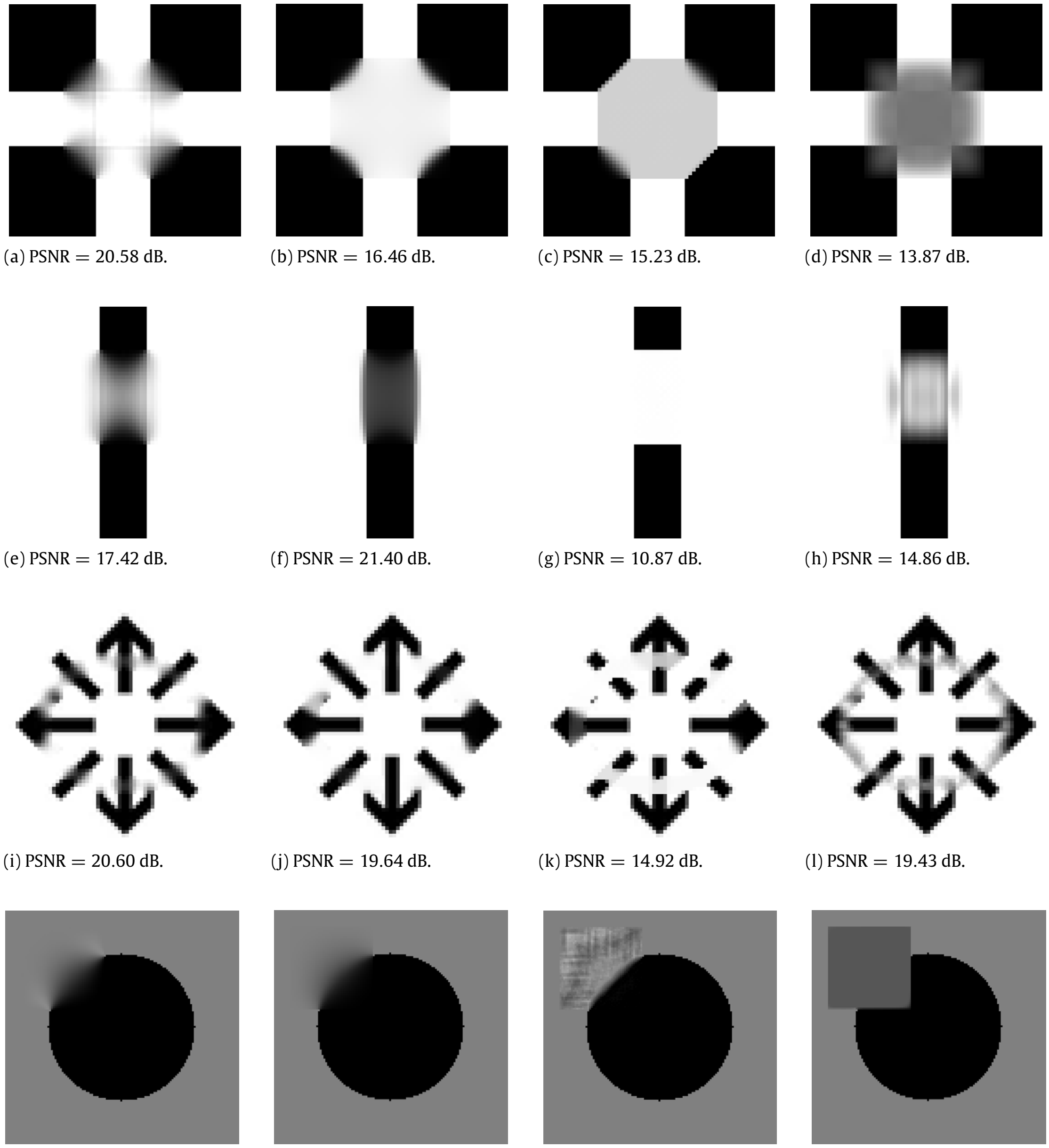
We select $tol = 10^{-3}$. The experiments are performed under MATLAB R2015b with Intel Xeon CPU E5504 at 2.00 GHz and 48.0 GB memory.

4.1. Test on gray images

Experiment 1: We test our algorithm on gray synthetic images with big gaps. The ground truth and observed images are given in Fig. 4.1. We compare our method with three other approaches: TV-H^{-1} inpainting proposed in [23], a fast TV algorithm proposed in [24] and deterministic annealing based image recovery method with hybrid sparse representation proposed in [47]. For simplicity, we abbreviate the fast TV algorithm by FTV and deterministic annealing based method [47] by DASR. The initial values are set to be the observed images in all experiments. We stop the iteration of TV-H^{-1} inpainting and FTV when the iteration number reaches 2000 (as increasing the number of iteration gives similar performance while consumes much more time). For a fair comparison among the competing methods, we have carefully tuned the parameters in TV-H^{-1} inpainting, FTV and DASR for each image to give the best possible performance. In Fig. 4.2, we show the restored results by the proposed method, TV-H^{-1} inpainting, FTV, and DASR. From Fig. 4.2(c), (d) and (o), (p), we can see that artificial boundaries are introduced by FTV and DASR, while the restored results by our method look much natural. In Fig. 4.2(e), (f), (h), the "bar" was connected by our method, TV-H^{-1} inpainting method and DASR method while the bar was broken in Fig. 4.2(g) by FTV method. The proposed method and TV-H^{-1} inpainting give better performance than FTV and DASR in this experiment. FTV is the fastest method among the compared methods. Though the computation time of our method is longer than FTV method, we give better results. The computation time of the proposed method is about one third of DASR method for images with size 64 × 64, and competitive with TV-H^{-1} inpainting method.

Table 4.1
The ratio between the number of nonzero elements and the length of β for the four images in Fig. 4.1, where n_1, n_0 represent the length of β and the number of significant nonzero coefficients in β after thresholding.

Image	4.1(a)	4.1(b)	4.1(c)	4.1(d)
n_1	12 032	14 096	13 936	14 368
n_0	188	134	475	571
Ratio	1.56%	0.95%	9.54%	3.97%



(a) PSNR = 20.58 dB. (b) PSNR = 16.46 dB. (c) PSNR = 15.23 dB. (d) PSNR = 13.87 dB.

(e) PSNR = 17.42 dB. (f) PSNR = 21.40 dB. (g) PSNR = 10.87 dB. (h) PSNR = 14.86 dB.

(i) PSNR = 20.60 dB. (j) PSNR = 19.64 dB. (k) PSNR = 14.92 dB. (l) PSNR = 19.43 dB.

(m) PSNR = 30.24 dB. (n) PSNR = 30.41 dB. (o) PSNR = 27.01 dB. (p) PSNR = 22.05 dB.

Fig. 4.2. In each row, from left to right: the inpainting results by the proposed method, TV-H^{-1}, FTV, and DASR.

To verify the assumption that the coefficients of $\{\psi_{k_i}\}_{k_i=1}^{p}$ are sparse, in other words, many coefficients in β are close to zero, Table 4.1 lists the ratio between the number of nonzero coefficients in β after thresholding and the length of β for the four images in Fig. 4.1. In this table, we set the coefficients smaller than 10^{-6} to zero. From Table 4.1, it can be seen that the coefficient β is sparse.

Table 4.2
PSNRs around $\lambda_1 = 10^{-10}$, $\mu = 10^{-5}$, and $\alpha = 0.1$ for image 4.1(b).

(λ_1, μ)	α		
	0.05	0.1	0.2
$(5 \times 10^{-11}, 10^{-5})$	18.30	17.65	17.31
$(1 \times 10^{-10}, 10^{-5})$	17.88	**17.42**	17.33
$(5 \times 10^{-10}, 10^{-5})$	13.58	17.15	18.56
$(10^{-10}, 5 \times 10^{-6})$	18.17	17.94	17.54
$(10^{-10}, 5 \times 10^{-5})$	23.58	20.47	18.63

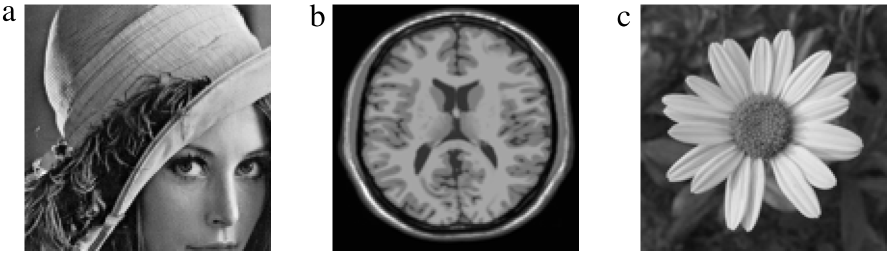


Fig. 4.3. Original test images.

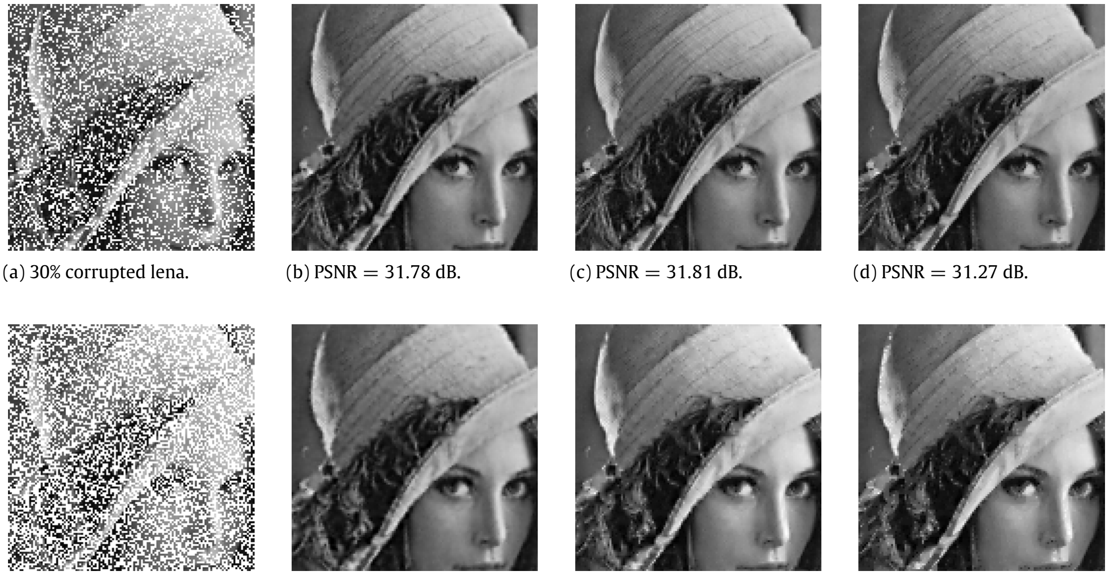
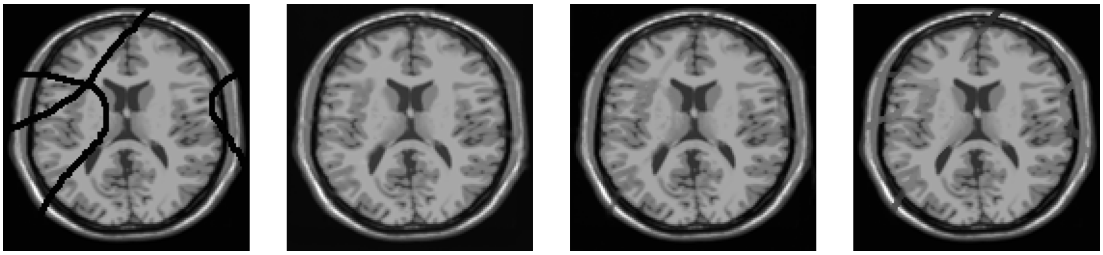

(a) 30% corrupted lena.
(b) PSNR = 31.78 dB.
(c) PSNR = 31.81 dB.
(d) PSNR = 31.27 dB.
(e) 50% corrupted lena.
(f) PSNR = 28.04 dB.
(g) PSNR = 27.72 dB.
(h) PSNR = 27.38 dB.

Fig. 4.4. (a) The lena image corrupted by 30% random noise; the restored images from (a) using the proposed method (b), FRAMC (c) and FTV (d); (e) The lena image corrupted by 50% random noise; the restored images from (e) using the proposed method (f), FRAMC (g) and FTV (h).

Note that there are four parameter λ_1, μ, α and γ in Algorithm 1, we just set $\mu = 10^{-5}, \alpha = 0.1$ and $\gamma = 1$ for all testing images. For any $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, the proposed algorithm is convergence. We can simply let $\gamma = 1$ for all experiments. We list the PSNRs around $\lambda_1 = 10^{-10}, \mu = 10^{-5}$ and $\alpha = 0.1$ for Fig. 4.1(b) in Table 4.2 as an example. It can be seen that the selected parameters cannot promise the best result for test image 4.1(b), but the result is already good enough.

Experiment 2: In Fig. 4.4, an 128×128 image corrupted by 30% and 50% random noise is tested by the proposed algorithm. The original image is shown in Fig. 4.3(a). We compared our method with the frame based method [48] and FTV. To make a fair comparison, we have tried our best to tune the parameters of the compared methods for the purpose of achieving the best possible performance [49]. In this paper, we abbreviate the method proposed in [48] by FRAMC. The restored results are given in Fig. 4.4. We get higher PSNR than FTV and the results are comparable to FRAMC.

Experiment 3: To test the robustness of the proposed algorithm, we apply it on more noisy images. Some 128×128 images corrupted by some scratches or texts are tested in this experiment. The original images are shown in Fig. 4.3(b)–(c).



(a) The observed image.

(b) PSNR = 37.52 dB.

(c) PSNR = 35.65 dB.

(d) PSNR = 31.57 dB.

Fig. 4.5. (a) The observed MRI image; the restored images by the proposed method (b), FRAMC (c) and FTV (d).

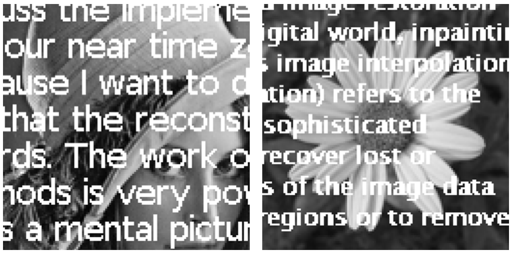
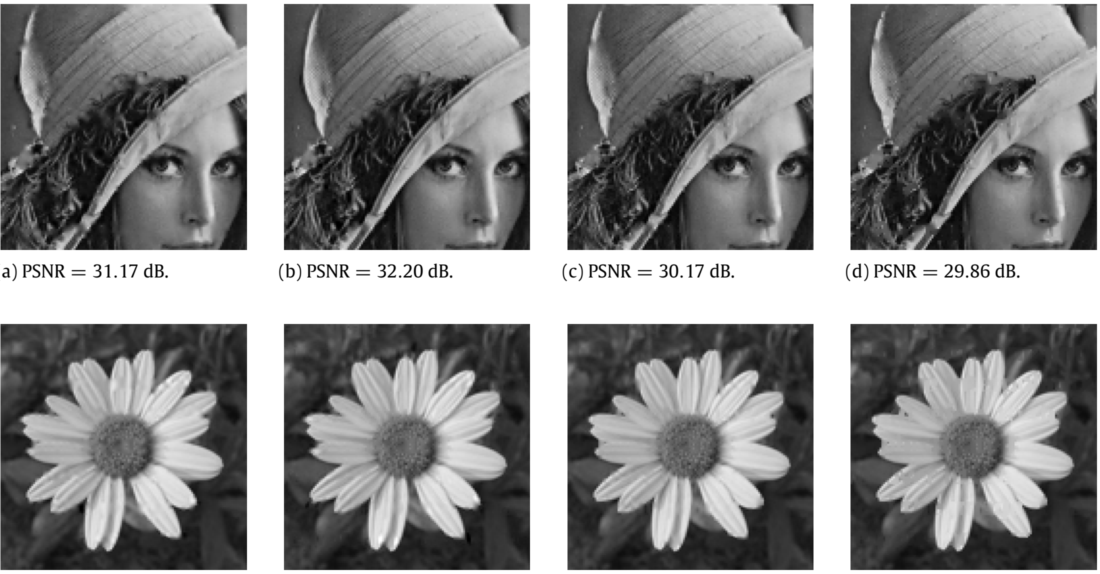


Fig. 4.6. Observed images.



(a) PSNR = 31.17 dB.

(b) PSNR = 32.20 dB.

(c) PSNR = 30.17 dB.

(d) PSNR = 29.86 dB.

(e) PSNR = 33.88 dB.

(f) PSNR = 33.79 dB.

(g) PSNR = 33.42 dB.

(h) PSNR = 32.25 dB.

Fig. 4.7. In each row, from left to right: the inpainting results by the proposed method, DASR, FRAMC and FTV.

In Fig. 4.5, the proposed method leads to the highest PSNR. Fig. 4.6 shows testing images corrupted by texts. The restored results are shown in Fig. 4.7. DASR and the proposed method lead to comparable results which are better than those of FTV and FRAMC. DASR is however slower than the proposed method. FTV and FRAMC lead to lower PSNR but are faster. Note that, we have a lot of room to speed up our non-optimized Matlab code. In Fig. 4.8, the image is corrupted by 2% Gaussian white noise and some scratches. The restored result by the proposed method is given in Fig. 4.8(b). From the above results, we can see that the performance of the proposed method is quite good.

4.2. Test on color images

Experiment 4: We also test our algorithm on color images inpainting in Figs. 4.9–4.11. In Fig. 4.9, we compare our algorithm with FTV. We still see scratches in Fig. 4.9(d) while the visual effect of 4.9(c) is more acceptable. We compare our method with FRAMC and FTV in Figs. 4.10 and 4.11. The proposed approach leads to the highest PSNR.

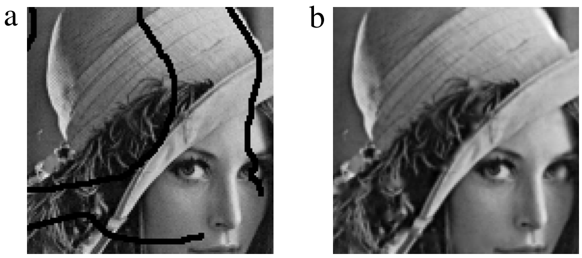
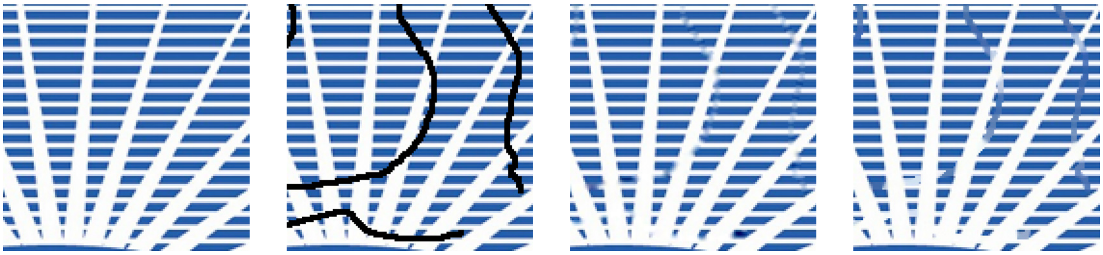
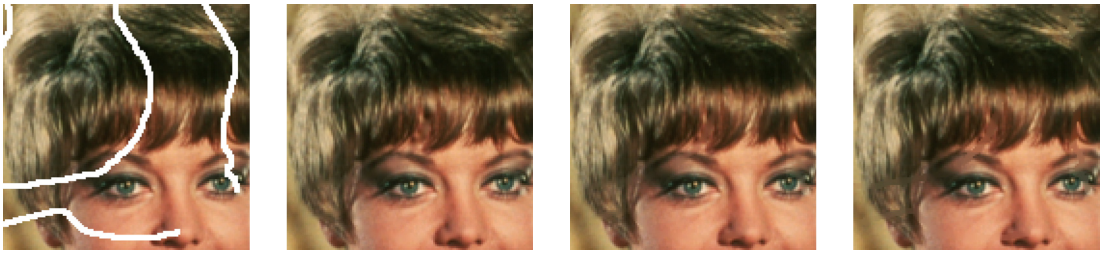


Fig. 4.8. The observed image (left) and the restored image (PSNR = 32.54 dB) by the proposed method.



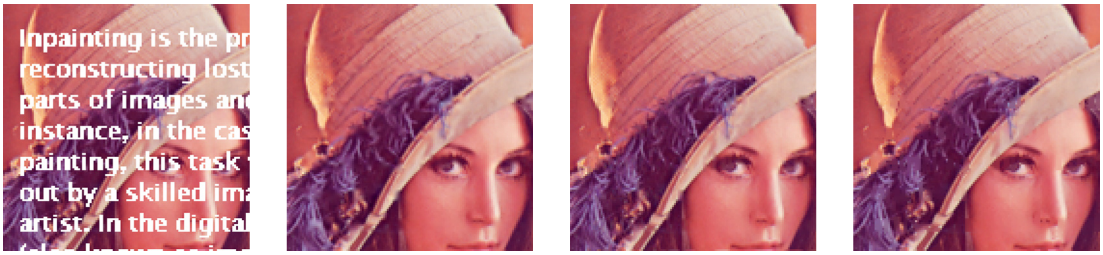
(a) The ground truth.
(b) The observed image.
(c) PSNR = 28.77 dB.
(d) PSNR = 23.23 dB.

Fig. 4.9. (a) The ground truth image; (b) the observed image; the restored images by the proposed method (c) and FTV (d). Figures are better visualized in the color file that is available on the web version of this article.



(a) The observed image.
(b) PSNR = 35.78 dB.
(c) PSNR = 34.91 dB.
(d) PSNR = 33.59 dB.

Fig. 4.10. The observed image (a) and restored images by the proposed method (b), FRAMC (c) and FTV (d). Figures are better visualized in the color file that is available on the web version of this article.



(a) The observed image.
(b) PSNR = 32.24 dB.
(c) PSNR = 32.06 dB.
(d) PSNR = 31.58 dB.

Fig. 4.11. The observed image (a) and restored images by the proposed method (b), FRAMC (c) and FTV (d). Figures are better visualized in the color file that is available on the web version of this article.

5. Conclusion

We have proposed a novel inpainting method based on intensity function estimation. We assume the underlying intensity function is defined on a continuous domain and belongs to a space spanned by a basis of a reproducing Kernel Hilbert space and some variations of Heaviside function. The proposed minimization problem is solved using ADMM. The parameters are easy to select. We use the same parameters for testing images with the same size. Many numerical experiments show that the proposed method is better or at least comparable with some competitive methods. It has potential in inpainting big gaps. As a future work, we will inpaint bigger gaps using divide and conquer idea. We can start the inpainting from the boundary of the inpainting domain and gradually move inward. We expect even better performance this way. Compared to variational/PDE based methods which always need to assume boundary conditions, the new proposed method does not have these problems. The proposed model can also be applied to image denoising and super-resolution.

Acknowledgments

The first and third authors thank the support by 973 Program (2013CB329404), NSFC (61170311). The second author is supported by USA NSF grant DMS-1521582. The first author is supported by fundamental Research Funds for the Central Universities (A03009023401040), National Science Foundation of China (11401081). The fourth author was supported by USA NSF grant DMS 1407028.

References

[1] V. Caselles, Exemplar-based image inpainting and applications, SIAM News 44 (2011).
[2] T.F. Chan, J.H. Shen, Nontexture inpainting by curvature-driven difussions, J. Vis. Commun. Image Represent. 12 (2001) 436–449.
[3] C. Guillemot, O.L. Meur, Image inpainting: overview and recent advances, IEEE Signal Process. Mag. 127 (2014) 127–144.
[4] M.S. Ishi, L. Singh, M. Agrawal, A review on image inpainting to restore image, ISOR J. Comput. Eng. 13 (2013) 08–13.
[5] C.B. Schönlieb, A. Bertozzi, Unconditionally stable schemes for higher order inpainting, Commun. Math. Sci. 9 (2011) 413–457.
[6] X.Q. Zhang, T.F. Chan, Wavelet inpainting by nonlocal total variation, Inverse Probl. Imaging 4 (2010) 191–210.
[7] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Computer Graphics, 2000, pp. 417–424.
[8] W. Baatz, M. Fornasier, P. Markowich, C.B. Schönlieb, Inpainting of ancient Austrain frescoes, in: Proceeding of Bridges 2008, 2008, pp. 150–156.
[9] J.W. Gu, L. Zhang, X.Y. Xing, Z.Q. Chen, X-ray CT metal artifacts reduction through curvature based sinogram inpainting, J. X-Ray Sci. Technol. 14 (2006) 73–82.
[10] T.F. Chan, J.H. Shen, Mathematical models for local nontexture inpaintings, SIAM J. Appl. Math. 62 (2002) 1019–1043.
[11] T.F. Chan, J.H. Shen, H.M. Zhou, Total variation wavelet inpainting, J. Math. Imaging Vision 25 (2006) 107–125.
[12] Y.W. Wen, R.H. Chan, A.M. Yip, A primal–dual method for total-variation-based wavelet domain inpainting, IEEE Trans. Image Process. 21 (2012) 106–114.
[13] T. Ogawa, M. Haseyama, Image inpainting based on sparse representation with a perceptual metric, EURASIP J. Adv. Signal Process. 179 (2013).
[14] B. Shen, W. Hu, Y.M. Zhang, Y.J. Zhang, Image inpainting via sparse representation, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 697–700.
[15] A. Efros, W.T. Freeman, Image quilting for texture synthesis and transfer, in: ACM Conference on Computer Graphics and Interactive, 2001, pp. 341–346.
[16] A. Criminisi, P. Perez, K. Toyama, Object removal by exemplar-based inpainting, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2 2003, pp. 721–728.
[17] A.S. Hareesh, V. Chandrasekaran, Exemplar-based color image inpainting: a fractional gradient function approach, Pattern Anal. Appl. 17 (2014) 389–399.
[18] M. Bertalmio, L. Vese, G. Sapior, S. Osher, Simultaneous structure and texture image inpainting, IEEE Trans. Image Process. 12 (2003) 882–889.
[19] M. Elad, J.L. Starck, P. Querre, D.L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), Appl. Comput. Harmon. Anal. 19 (2005) 340–358.
[20] T.F. Chan, S.H. Kang, J.H. Shen, Euler's elastica and curvature-based inpainting, SIAM J. Appl. Math. 63 (2002) 564–592.
[21] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, Filling-in by joint interpolation of vector fields and gray levels, IEEE Trans. Image Process. 10 (2001) 1200–1211.
[22] A. Bertozzi, S. Esedo$\bar{G}$lu, A. Gillette, Inpainting of binary images using Cahn-Hilliard equation, IEEE Trans. Image Process. 16 (2007) 285–291.
[23] M. Burger, L. He, C.B. Schönlieb, Cahn-Hilliard inpainting and a generalization for grayvalue images, SIAM J. Imag. Sci. 2 (2009) 1129–1167.
[24] F. Li, C.M. Shen, R.H. Liu, J.S. Fan, A fast implementation algorithm of TV inpainting model based on operator splitting method, Compt. Electr. Eng. 37 (2011) 782–788.
[25] H. Yamauchi, J. Haber, H.-P. Seidel, Image restoration using multiresolution texture synthesis and image inpainting, in: Computer Graphics International (CGI), 2003, pp. 120–125.
[26] Z.B. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity, IEEE Trans. Image Process. 19 (2010) 1153–1165.
[27] J. Drori, D. Cohen-Or, H. Yeshurun, Fragment-based image completion, ACM Trans. Graph. 22 (2003) 303–312.
[28] C.W. Fang, J.J. Lien, Rapid image completion system using multiresolution patch-based directional and nondirectional approaches, IEEE Trans. Image Process. 18 (2009) 2769–2779.
[29] A. Wong, J. Orchard, A nonlocal-means approach to exemplar-based inpainting, in: IEEE International Conference on Image Processing (ICIP), 2008, pp. 2600–2603.
[30] J.Y. Wu, Q.Q. Ruan, Object removal by cross isophotes exemplar-based image inpainting, in: IEEE International Conference on Pattern Recognition (ICPR), 2006, pp. 810–813.
[31] P. Bouboulis, M. Mavroforakis, Reproducing Kernel Hilbert Spaces and fractal interpolation, J. Comput. Appl. Math. 235 (2011) 3425–3434.
[32] J. Shawe-Talor, N. Cristianini, Kernel Methods for Pattern Analysis, 2004.
[33] K. Slavakis, P. Bouboulis, S. Theodoridis, Adaptive multiregression in reproducing kernel Hilbert spaces: the multiaccess MIMO channel case, IEEE Trans. Image Process. 23 (2012) 260–276.
[34] P. Bouboulis, K. Slavakis, S. Theodoridis, Adaptive kernel-based image denoising employing semi-parametric regularization, IEEE Trans. Image Process. 6 (2010) 1465–1479.
[35] M.H. Quang, S.H. Kang, T.M. Le, Image and video colorization using vector-valued Reproducing Kernel Hilbert Spaces, J. Math. Imaging Vision 37 (2010) 49–65.
[36] S.H. Kang, B. Shafei, G. Steidl, Supervised and transductive multi-class segmentation using p-laplacians and RKHS methods, J. Vis. Commun. Image Represent. 25 (2014) 1136–1148.
[37] G. Wahba, Spline Models for Observational Data, 1990.
[38] G. Wahba, J. Wendelberger, Some new mathmatical methods for variational objective analysis using splines and cross-validation, Mon. Weather Rev. 108 (1980) 1122–1145.
[39] J. Duchon, Spline minimizing rotation-invariant semi-norms in Sobolev spaces, in: Handbook of Constructive Theory of Functions of Several Variables, 1997, pp. 85–100.
[40] G. Kimeldorf, G. Wahba, Some results on Tchebycheffian spline functions, J. Math. Anal. Appl. 33 (1971) 82–95.
[41] B. Dong, H. Ji, J. Li, Z.W. Shen, Y.H. Xu, Wavelet frame based blind inpainting, Appl. Comput. Harmon. Anal. 32 (2012) 268–279.
[42] X.L. Zhao, F. Wang, M.K. Ng, A new convex optimization model for multiplicative noise and blur removal, SIAM J. Imag. Sci. 7 (2014) 456–475.

[43] Y.H. Xiao, J.F. Yang, X.M. Yuan, Alternating algorithm for total variation image reconstruction form random projections, Inverse Probl. Imaging 6 (2012) 547–563.
[44] R.H. Chan, M. Tao, X.M. Yuan, Constrained total variation deblurring model and fast algorithms based on alternating direction method for multipliers, SIAM J. Imag. Sci. 60 (2013) 680–697.
[45] J.F. Yang, Y. Zhang, Alternating direction algorithms for l_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (2011) 250–278.
[46] B.S. He, H. Yang, Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities, Oper. Res. Lett. 23 (1998) 151–161.
[47] X. Li, Image recovery via hybrid sparse representations: a deterministic annealing approach, IEEE J. Sel. Top. Sign. Proces. 5 (2011) 953–962.
[48] J.F. Cai, S. Osher, Z.W. Shen, Split bregman methods and frame based image restoration, SIAM Multiscale Model. Simul. 8 (2009) 337–369.
[49] W. Dong, G. Shi, X. Li, Y. Ma, F. Huang, Compressive sensing via nonlocal low-rank regularization, IEEE Trans. Image Process. 23 (2014) 3618–3632.