©Journal of Computational Mathematics (2008)

http://www.global-sci.org/jcm/

Multigrid Method For a Modified Curvature Driven Diffusion Model for Image Inpainting

Carlos Brito-Loeza and Ke Chen

Abstract

Digital inpainting is a fundamental problem in image processing and many variational models for this technique have appeared recently in the literature. Among them are the very successfully Total Variation (TV) model [1] designed for local inpainting and its improved version for large scale inpainting: the Curvature-Driven Diffusion (CDD) model [2]. For the above two models, their associated Euler Lagrange equation is a highly non linear partial differential equation. For the TV model there exists a relatively fast and easy to implement fixed point method, so adapting the multigrid method of [3] to here is immediate. For the CDD model however, so far only the well known but usually very slow explicit time marching method has been reported and we explain why the implementation of a fixed point method for the CDD model is not straight forward. Consequently the multigrid method as in [3] will not work here. This fact represents a strong limitation to the range of applications of this model since usually fast solutions are expected. In this paper, we introduce a modification designed to enable a fixed point method to work and to preserve the features of the original CDD model. As a result, a fast and efficient multigrid method is developed for the modified model. Numerical experiments are presented to show the very good performance of the fast algorithm.

AMS subject class: 68U10, 65F10, 65K10.

Keywords: Image inpainting, variational models, regularization, multilevel methods.

1 Introduction

Image inpainting has been defined as the process of reconstituting the missing or damaged portions of an image, in order to make it more legible and to restore its unity. The aim of inpainting is then to modify an image in a way that is non-detectable for an observer who does not know the original image [4].

There are a variety of reasons why images can have damaged parts, for instance because of some physical degradation like aging, weather or intentional scratching. Not only that, we also would like to recover parts of objects of an image occluded by other objects or to reconstruct parts that have been missing due to digital communication processes. We can imagine a number of applications of this technique: among the most known are the restoration of old pictures with scratches or missing patches [4], text removal, digital zooming and superresolution [1], error concealment [5], disocclusion in computer vision, X-Ray CT artifacts reduction [6], and the long list continues. Inpainting techniques deal with these kinds of problems trying to reconstruct in the best possible way the missing or damaged parts of the image from the available information.

The mathematical interest on this field became increasingly active by the end of the last decade with the very first works on image interpolation of Mumford, Nitzberg and Shiota [7], Masnou and Morel [8] and Caselles, Morel and Sbert [9]. However it was the pioneering work of Bertalmio *et al* [4] who proposed an algorithm to imitate the work of inpainting artists who manually restore old damaged pictures which mainly motivated all the subsequent research in this field [1]. This algorithm cleverly transports a smoothness image measure (namely the Laplacian of the image) along the level lines (contours of the same image intensity) directed into the inpainting domain; in their paper, they also showed that some intermediate steps of anisotropic diffusion are necessary to avoid blurring of edges. This algorithm was created mostly intuitively but later on turned out to be closely related to the Navier-Stokes equation, as showed by Bertozzi *et al*, see [10]. Since then, many other authors have proposed different models for digital inpainting.

Chan and Shen [1] introduced the Total Variation (TV) model for local inpainting based on the celebrated total variation based image denoising model of Rudin, Osher and Fatemi [11]. Later on the same authors modified this model to improve its performance for large scale inpainting, and created the so-called Curvature-Driven Diffusion (CDD) model [2]. Furthermore they, together with Kang, introduced a higher order variational model [12] based on the Euler's elastica which connects the level lines by using Euler elastica curves [13] instead of using straight lines as the TV model does. Unfortunately for the latter two

models there appear to exist no fast methods to find the numerical solution. The aim of this paper is to develop a fast multigrid algorithm for the CDD model.

A related inpainting model was proposed by Esedoglu and Shen [14] and is based on the very successfully Mumford-Shah image segmentation model. This model is also good for local inpainting but shares the same problem of the TV model that it cannot reconnect far apart separated parts of broken objects. To fix this problem, the same authors [14] proposed the Mumford-Shah-Euler inpainting model which in the same fashion of the Euler's elastica model uses the information encoded in the curvature to reconnect smoothly the level lines. More recently, in a separate idea, Bertozzi, Esedoglu and Gillete [15] proposed a model to inpaint binary images based on the Cahn-Hilliard equation and Grossauer and Scherzer [16] proposed a model based on the complex Ginzburg-Landau equation. It remains to develop a fast multigrid method for this model.

Each one of the above models has its own particular features which may not suit all applications. However as rightly remarked in [17] one of the most interesting open problems in digital inpainting (whatever the model) is the fast and efficient digital realization. The new multigrid method for the CDD model is our first step in developing fast algorithms for this technique.

The rest of this paper is organized as follows. Section 2 introduces the image inpainting problem and 2 variational models, followed by the review of a commonly-used numerical method in Section 3. Section 4 describes first the modified CDD model and then the framework of a nonlinear multigrid method with emphasis on the local smoother. A local Fourier analysis is shown to give an indication of the effectiveness of the smoother. Finally Section 5 presents some testing results illustrating the effectiveness of the modified model and the associated multigrid method.

2 Problem Formulation and Variational Models

Given an image $z = z(x, y)$ defined on a domain $\Omega \in \Re^2$ and one subset $D \subset \Omega$ where the pixel values of z are missing or damaged due to some reason as illustrated in Figure 1. The typical inpainting problem is to try to reconstruct the values of z in D from the available information on $\Omega \backslash D$ which may contain noise. The subset D which is known as the inpainting domain may have complicated topology and be not necessarily connected.

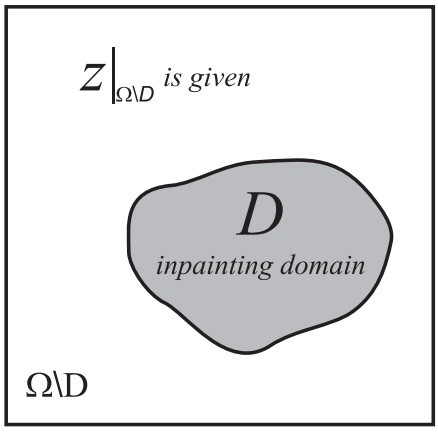

Fig. 1: Illustration of a typical inpainting problem.

2.1 The Total Variation Model

In this section we review very briefly the TV model [1]. This model was constructed based on three principles that according to [1] a good inpainting model must satisfy. They are: *Locality* meaning that to carry out the inpainting process only the information surrounding the inpainting domain must be used. *Restoration of narrow edges* meaning that the model must be able to reconstruct the missing parts of the edges which give the most visually information in an image. And finally due to images usually have some amount of noise included, the model must be *Robust to noise* meaning that it must get rid of that noise (up to some reasonable level) and restore the missing part.

Based on the image denoising model of Rudin, Osher and Fatemi [11] and assuming that $u = u(x, y)$ and $n = n(x, y)$ are the true image and the unknown additive Gaussian noise respectively and they satisfy $z = u + n$ on $\Omega\backslash D$. The TV inpainting model is as follows

$$\min_u \int_\Omega |\nabla u| \; dxdy \;\; + \;\; \frac{\lambda}{2} \int_{\Omega\backslash D} (u - z)^2 \; dxdy. \tag{1}$$

Even thought direct minimization ideas from [18] could be applied, so far the above minimization is numerically solved by means of its associated Euler-Lagrange equation given by

$$\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda_E (z - u) = 0 \quad \text{with} \quad \frac{\partial u}{\partial \vec{n}} = 0 \;\text{ in }\; \partial\Omega, \tag{2}$$

where $\vec{n}$ is the unit outward normal on the boundary $\partial\Omega$ and λ_E

$$\lambda_E = \begin{cases} \lambda > 0 & u \in \Omega\backslash D \\ 0 & u \in D\,. \end{cases} \tag{3}$$

In D, where $\lambda_E = 0$, equation (2) reduces to an ill-conditioned boundary value problem with non unique solution as it was shown in [9].

To illustrate the virtues and limitations of the TV model we shall present two inpainting examples. In Figure 2, the inpainting domain D represented by the horizontal noisy bars has relatively small size compared with the characteristic feature of the image, therefore the TV model performs very well and carry out a good inpainting. In Figure 3 however, D which is represented by the noisy triangle is relatively large and therefore the TV model gives an unpleasant result.

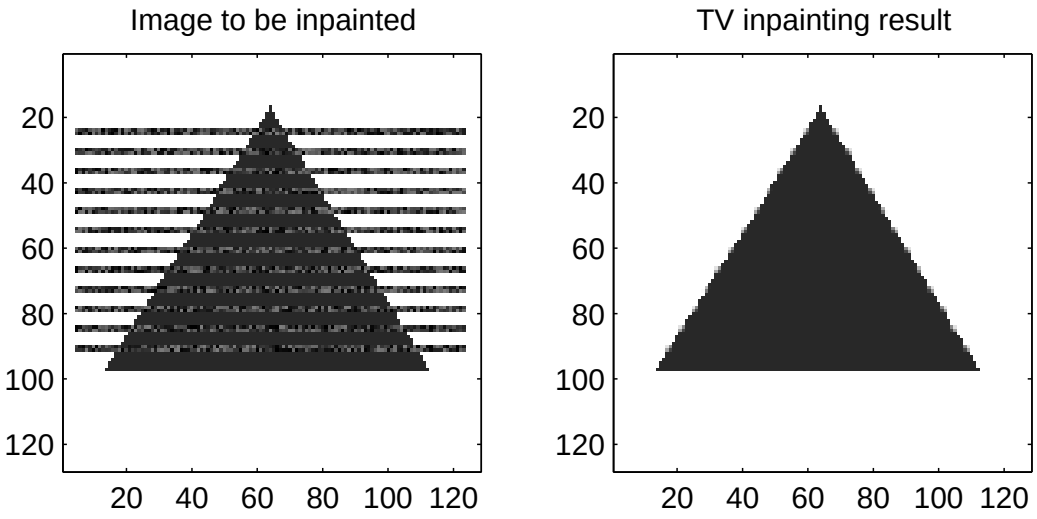

Fig. 2: Example of a good TV inpainting.

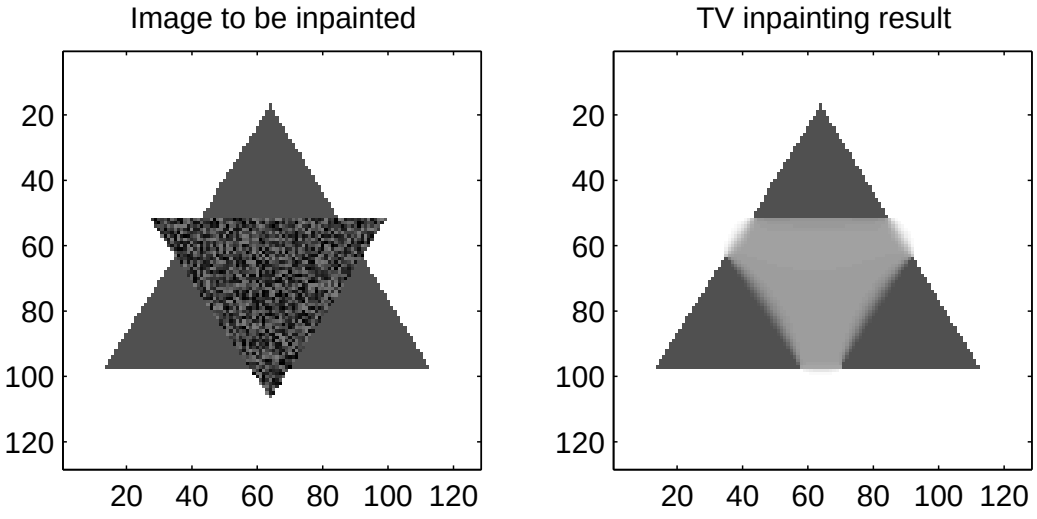

Fig. 3: Example of one of the limitations of the TV model.

2.2 The Curvature-Driven Diffusion Model

The CDD model which we review in this section was designed to correct the failure of the TV model (due to its locality feature) to reconnect far apart separated parts of broken objects.

Looking for a solution to this problem, Chan and Shen [2] realized that in the TV model the diffusion coefficient given by $\hat{D} = \frac{1}{|\nabla u|}$ only depends on the contrast or strength of the level lines and it does not depend on the geometric information of the level lines themselves. They found that the curvature $\kappa = \kappa(x, y)$ defined as $\kappa = \bigtriangledown \cdot \frac{\nabla u}{|\nabla u|}$ could be used to modify the diffusion coefficient $\hat{D}$ by introducing a function $g = g(\kappa)$ within it. This way the geometric information encoded in κ is used to get the diffusion coefficient stronger where it is necessary. The new $\hat{D}$ is then given by

$$\hat{D} \;=\; \frac{g(|\kappa|)}{|\nabla u|}, \quad \text{with} \quad g(s) = \begin{cases} 0 & s = 0 \\ \infty & s = \infty \\ > 0, & 0 < s < \infty. \end{cases} \tag{4}$$

On one hand, the choice of $g(\infty) = \infty$ was selected to take advantage of those points with very high or infinity curvature and use them to encourage reconnection increasing $\hat{D}$ as much as possible.

On the other hand, the choice of $g(0) = 0$ is to avoid the CDD model degenerating to the TV model. According to Chan and Shen the choice of $g(0) = a \neq 0$ could endanger the connectivity principle, see [2]. We will discuss more on this subject later on. They suggested [2] the use of

$$g(s) \;=\; s^p, \;\; \text{with} \;\; s > 0, \;\; p \geq 1. \tag{5}$$

To get rid of possible noise present on the initial image which could be propagated at the interior of the inpainting domain, a fidelity term is used as in the TV model. Thus, by defining $G = G[(x, y), |\kappa|]$ as

$$G = \begin{cases} 1 & (x, y) \in \Omega \backslash D \\ g(s) & (x, y) \in D, \end{cases} \tag{6}$$

the CDD scheme is to solve the following third order nonlinear equation:

$$\nabla \cdot V + \lambda_E (z - u) = 0 \quad \text{with} \quad \frac{\partial u}{\partial \vec{n}} = 0 \;\; \text{in} \;\; \partial\Omega, \tag{7}$$

where $\vec{n}$ and λ_E are defined as before, and the vector field $V = \langle V^1, V^2 \rangle$ defined as $V = G\frac{\nabla u}{|\nabla u|}$. Since V contains the term $|\nabla u|^{-1}$, to avoid the singularity at flat regions $|\nabla u|_\beta \stackrel{def}{=} \sqrt{|\nabla u| + \beta}$ is used instead of $|\nabla u|$, where β is a small parameter. We denote the left hand side of (7) as a function $g(u)$ so the equation will become $g(u) = 0$, and denote by

$$\alpha = \begin{cases} \frac{1}{\lambda} & \text{in } \Omega \backslash D \\ 1 & \text{in } D \end{cases} \quad \text{and} \;\; \chi = \begin{cases} 1 & \text{in } \Omega \backslash D \\ 0 & \text{in } D \end{cases} \tag{8}$$

so that (7) will be the same as $\alpha\nabla \cdot V + \chi(z-u) = 0$.

3 Review of Numerical Methods

In this section, we intend to review the state-of-the-art methods for numerically solving the CDD model. Surprisingly the list is very short and it only has been solved using an explicit time marching scheme.

3.1 Discretization

We start by discretizing (7) as follows

$$\frac{V^1_{i+\frac{1}{2},j} - V^1_{i-\frac{1}{2},j}}{h} + \frac{V^2_{i,j+\frac{1}{2}} - V^2_{i,j-\frac{1}{2}}}{h} + \lambda_E(z_{i,j} - u_{i,j}) = 0. \tag{9}$$

The scheme is illustrated in Figure 4. Now we have to approximate V^1 and V^2 at the half-points, for instance at $(i+\frac{1}{2}, j)$, u_x is approximated by central differences $(u_x)_{i+\frac{1}{2},j} = (u_{i+1,j} - u_{i,j})/h$, u_y by average approximation $(u_y)_{i+\frac{1}{2},j} = (u_{i+1,j+1} - u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1})/4h$ and $|\nabla u|_\beta$ in the natural way :

$$\frac{1}{h}\sqrt{(u_{i+1,j} - u_{i,j})^2 + \left(\frac{1}{4}(u_{i+1,j+1} - u_{i+1,j-1} + u_{i,j+1} - u_{i,j-1})\right)^2 + h^2\beta} \tag{10}$$

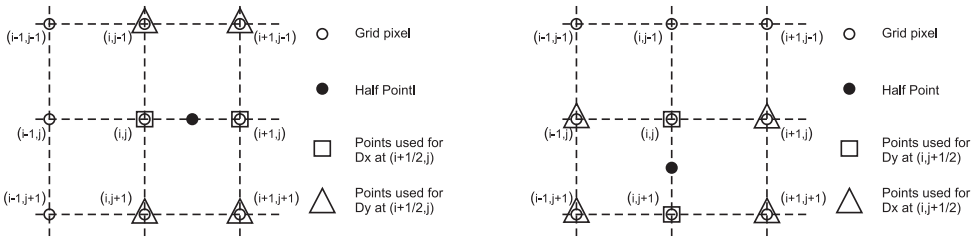

Fig. 4: On the left side an x-half-point and on the right side a y-half-point.

Hence we have

$$\begin{aligned} -G_{i+\frac{1}{2},j}\left(\frac{\alpha(u_x)_{i+\frac{1}{2},j}}{h|\nabla u|_{i+\frac{1}{2},j}}\right) + G_{i-\frac{1}{2},j}\left(\frac{\alpha(u_x)_{i-\frac{1}{2},j}}{h|\nabla u|_{i-\frac{1}{2},j}}\right) - G_{i,j+\frac{1}{2}}\left(\frac{\alpha(u_y)_{i,j+\frac{1}{2}}}{h|\nabla u|_{i,j+\frac{1}{2}}}\right) \\ +G_{i,j-\frac{1}{2}}\left(\frac{\alpha(u_y)_{i,j-\frac{1}{2}}}{h|\nabla u|_{i,j-\frac{1}{2}}}\right) + \chi u_{i,j} = \chi z_{i,j}. \end{aligned} \tag{11}$$

To approximate the curvature term κ in G and in (4), we use the same idea of the ghost half points to approximate the divergence operator

$$\kappa = \nabla \cdot \frac{\nabla u}{|\nabla u|} = \frac{\partial}{\partial x}\left[\frac{u_x}{|\nabla u|}\right] + \frac{\partial}{\partial y}\left[\frac{u_y}{|\nabla u|}\right]. \tag{12}$$

By using again central differences and averages we have for example at $(i+\frac{1}{2},j)$ that

$$h\cdot\frac{\partial}{\partial x}\left[\frac{u_x}{|\nabla u|}\right]_{i+\frac{1}{2},j}=\left[\frac{u_x}{|\nabla u|}\right]_{i+1,j}+\left[\frac{u_x}{|\nabla u|}\right]_{i,j}. \tag{13}$$

$$\begin{aligned}4h\cdot\frac{\partial}{\partial y}\left[\frac{u_y}{|\nabla u|}\right]_{i+\frac{1}{2},j} &= \left[\frac{u_x}{|\nabla u|}\right]_{i+1,j+1}-\left[\frac{u_x}{|\nabla u|}\right]_{i+1,j-1}\\ &+ \left[\frac{u_x}{|\nabla u|}\right]_{i,j+1}-\left[\frac{u_x}{|\nabla u|}\right]_{i,j-1}.\end{aligned} \tag{14}$$

Finally (7) becomes a system of nonlinear algebraic equation denoted by $g(u_{i,j})=0$ or

$$\begin{aligned}u_{i,j}\mathbb{S}_{i,j}-u_{i+1,j}\left(C_{i+\frac{1}{2},j}\right)-u_{i-1,j}\left(C_{i-\frac{1}{2},j}\right)-u_{i,j+1}\left(C_{i,j+\frac{1}{2}}\right)\\ -u_{i,j-1}\left(C_{i,j-\frac{1}{2}}\right)-\chi z_{i,j}=0\end{aligned} \tag{15}$$

where the new C notation represents the nonlinear terms, for instance,

$$C_{(i+\frac{1}{2},j)}=\frac{\alpha\, G_{(i+\frac{1}{2},j)}}{h|\nabla u|_{(i+\frac{1}{2},j)}} \tag{16}$$

and similarly we can construct C for the other three half-points and $\mathbb{S}_{i,j}$ is defined as

$$\mathbb{S}_{i,j}=C_{i+\frac{1}{2},j}+C_{i-\frac{1}{2},j}+C_{i,j+\frac{1}{2}}+C_{i,j-\frac{1}{2}}+\chi. \tag{17}$$

3.2 Explicit Time Marching Method

By using this method one assumes that the solution is expected to be the steady-state solution of a parabolic equation of the form:

$$\frac{\partial u}{\partial t}=\nabla\cdot V+\lambda_E(z-u), \tag{18}$$

or, $\frac{\partial u}{\partial t}=\alpha\nabla\cdot V+\chi(z-u)$, with the initial condition $u(x,y,0)=z(x,y)$ and appropriate boundary conditions. and using an explicit Euler method for the left hand side, we get

$$u_{i,j}^{k+1}=u_{i,j}^{k}-\tau g(u_{i,j}^{k}),\quad k=0,1,\ldots \tag{19}$$

Here a size restriction on the time step $\tau=\Delta t$ has to be imposed to guarantee the stability of the numerical solution. This is the main drawback of the time marching method, the problem being that due to its high nonlinearity, τ must be chosen very small which implies a large number of iterations to reach a meaningful solution. One option is to accelerate this method using the ideas developed in [19]. However, even in that case the cpu-time consumed by the resulting algorithm is still not appropriate for large images.

3.3 A possible fixed point method

For numerically solving the nonlinear algebraic equation (11) at each (i,j) point we fix the nonlinear terms G and $|\nabla u|$ at some k-step and solve for the $k+1$ step as in [20, 21] for other problems. We have from (15) that

$$\begin{aligned} u_{i,j}^{k+1}\mathbb{S}_{i,j} - u_{i+1,j}^{k+1}\left(C_{i+\frac{1}{2},j}^{k}\right) - u_{i-1,j}^{k+1}\left(C_{i-\frac{1}{2},j}^{k}\right) - u_{i,j+1}^{k+1}\left(C_{i,j+\frac{1}{2}}^{k}\right) \\ -u_{i,j-1}^{k+1}\left(C_{i,j-\frac{1}{2}}^{k}\right) = \chi z_{i,j} \end{aligned} \tag{20}$$

where similar to before

$$C_{(i+\frac{1}{2},j)}^{k} = \frac{\alpha\, G_{(i+\frac{1}{2},j)}^{k}}{h|\nabla u|_{(i+\frac{1}{2},j)}^{k}} \tag{21}$$

and so on, and

$$\mathbb{S}_{i,j} = C_{i+\frac{1}{2},j}^{k} + C_{i-\frac{1}{2},j}^{k} + C_{i,j+\frac{1}{2}}^{k} + C_{i,j-\frac{1}{2}}^{k} + \chi. \tag{22}$$

Then such a fixed point method amounts to solving the linear system of (20)

$$A(\mathbf{u}^k)\mathbf{u}^{k+1} = \mathbf{z}, \tag{23}$$

where $\mathbf{u}^k$ and $\mathbf{z}$ vectors are defined as $\mathbf{u}^k = [u_{1,1}^k, u_{2,1}^k \ldots, u_{n,1}^k,\ u_{1,2}^k, \ldots, u_{n,m}^k]$ and $\mathbf{z} = [\chi z_{1,1}, \chi z_{2,1} \ldots,\ \chi z_{n,1}, \chi z_{1,2}, \ldots, \chi z_{n,m}]$.

The selection of G on D plays a crucial role on the feasibility of the implementation of the numerical scheme. According to Chan and Shen [2], on the inpainting domain, G must obey equations (5) and (6). Therefore it must be

$$G = \begin{cases} 0 & \kappa = 0 \\ \infty & \kappa = \infty \\ |\kappa|^p, \ \text{ with } \ p \geq 1 & 0 < \kappa < \infty \end{cases}. \tag{24}$$

However if we allow G to be equal to zero when $\kappa = 0$, matrix $A(\mathbf{u}^k)$ is singular i.e. every time $\kappa = 0$ at one (i,j) pixel of the image then $A(\mathbf{u}^k)$ losses one degree of its rank. Therefore the fixed point (FP) method (32) does not work for the CDD model with (24).

4 Nonlinear Multigrid

Multigrid methods (MG) have proved to be very useful when solving many linear (and some nonlinear) partial differential equations (PDEs) such as those arising from image restoration problems and others, see [3, 22, 23, 24, 25, 26] for successful examples. Usually for a multigrid method to converge, finding a suitable smoother is the key and the task is nontrivial for a nonlinear problem.

We now proceed to develop a multigrid algorithm for the CDD formulation (7):

$$\nabla \cdot \left(G_{i,j} \frac{(\nabla u)_{i,j}}{|\nabla u|_{i,j}} \right) + \lambda_E (z_{i,j} - u_{i,j}) = 0. \tag{25}$$

First of all, we start by introducing new notation and rewriting the equation for the purpose of making it more tractable for computing implementation. To present a nonlinear multigrid method, write (25) as

$$(Nu)_{i,j} = -\alpha \nabla \cdot \left(G_{i,j} \frac{(\nabla u)_{i,j}}{|\nabla u|_{i,j}} \right) + \chi u_{i,j} = \chi z_{i,j}. \tag{26}$$

where we have denoted by $Nu = \chi z$ the main nonlinear operator equation, always keeping in mind the action of the indictor function χ. Since we have to approximate this equation on grids of different sizes we will denote by $N_h u_h = \chi z_h$ the discrete approximation on the finest grid Ω_h of size h and denote by $N_{2h} u_{2h} = \chi z_{2h}$ the same on Ω_{2h} which is obtained by standard coarsening i.e the nonlinear operator N_{2h} which results from discretizing equation (26) using a cell-centered grid Ω_{2h} with grid spacing $2h$. The same way we can generate a sequence of L coarse levels $4h, 8h, \ldots, L$.

Next we briefly mention the standard intergrid transfer operators. Denote by R_h^{2h} (restriction) and I_{2h}^h (interpolation) respectively two transfer operators between Ω_h and Ω_{2h} which on cell-centered grids are defined by the following equations.

The Restriction operator is defined by $R_h^{2h} v_h = v_{2h}$ where

$$\begin{gathered} (v_{2h})_{i,j} = \tfrac{1}{4} \left[(v_h)_{2i-1,2j-1} + (v_h)_{2i-1,2j} + (v_h)_{2i,2j-1} + (v_h)_{2i,2j} \right] \\ 1 \le i \le n/2, \quad 1 \le j \le m/2. \end{gathered} \tag{27}$$

The Interpolation operator is defined by $I_{2h}^h v_{2h} = v_h$ where

$$(v_h)_{2i,2j} = \begin{cases} \frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i+1,j+1}] \\ \frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i-1,j+1}] \\ \frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i+1,j-1}] \\ \frac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i-1,j-1}] \\ 1 \le i \le n/2, \quad 1 \le j \le m/2. \end{cases} \tag{28}$$

Finally we discuss how to treat the two operators across the interface of the inpainting domain D.

(*) ADD details.....

For the smoother which in a general way we denote by $FPGS$ we shall give its specific definition later on. Now we proceed to state our nonlinear MG which was selected to be a

V-cycle method meaning that just one recursive call to the algorithm is made on each level to approximately solve the coarse grid problem.

Algorithm *(Nonlinear Multigrid Method)*

Select an initial guess v_h

set k=0

While $\|v_h^{k+1} - v_h^k\|_2 < tol$

$v_h^{k+1} \leftarrow FAS(v_h^k, N_h^k, z_h, M_h, v_0, v_1, v_2, gsiter)$

k=k+1

end

The algorithm *FAS* [26] is defined recursively as follows:

$$v_h \leftarrow FAS(v_h, N_h, z_h, M_h, v_0, v_1, v_2, gsiter)$$

Algorithm *(FAS)*

1. *If* Ω^h = *coarsest grid, solve* $N_h v_h = z_h$ *accurately (i.e.* ν_0 *iterations) and return Else*

2. *Pre-smoothing: For* $l = 1$ *to* ν_1, $\quad v_h \leftarrow FPGS(v_h, z_h, gsiter, M_h)$.

3. *Restrict to the coarse grid,* $M_{2h} \leftarrow R_h^{2h} M_h$ *and* $v_{2h} \leftarrow R_h^{2h} v_h$.

4. *Set the initial solution for the next level,* $\bar{v}_{2h} \leftarrow v_{2h}$.

5. *Compute* $z_{2h} \leftarrow R_h^{2h}(z_h - N_h v_h) + N_{2h} v_{2h}$.

6. *Implement* $v_{2h} \leftarrow FAS_{2h}(v_{2h}, N_{2h}, z_{2h}, M_{2h}, v_0, v_1, v_2, gsiter)$.

7. *Add the residual correction,* $v_h \leftarrow v_h + I_{2h}^h(v_{2h} - \bar{v}_{2h})$.

8. *Post-smoothing: For* $l = 1$ *to* ν_2, $\quad v_h \leftarrow FPGS(v_h, z_h, gsiter, M_h)$.

Here M_h is the mask for the inpainting domain and M_{2h} is the mask for the coarser grid which results from applying the restriction operator I_h^{2h} to M_h. Since M_h is composed only by zeros representing the pixels in $\Omega \backslash D$ and ones representing the pixels in D, the mask on the next coarse level $M_{2h} = I_h^{2h} M_h$ will have some non zero neither one entries due to the action of the operator I_h^{2h} on M_h, so we have to reset those entries to zero.

The number of pre and post-correction smoothing steps is given by ν_1 and ν_2 respectively, and ν_0 represents the number of iterations carried out at the coarsest grid which in our experiments is always 2×2. On the other hand, *gsiter* represents the number of Gauss-Seidel iterations at each pre or post-smoothing step.

4.1 The modified CDD and the FP smoother

Due to the problem described above in §3.3, we decide to introduce two positive parameters a and b in the function g emulating the Euler's Elastica model [12]. Then our proposal for G is

$$G = \begin{cases} a & \kappa = 0 \\ \infty & \kappa = \infty \\ a + b|\kappa|^p, & \text{with } \ p \geq 1 \ \text{ and } \ 0 < \kappa < \infty. \end{cases} \tag{29}$$

For a general p, our modified CDD takes the form

$$\nabla \cdot (a + b\kappa^p)\frac{\nabla u}{|\nabla u|} + \lambda_E(z - u) = 0. \tag{30}$$

Now we can similarly consider a fixed point method for numerically solving the discretized equation of (30) at each (i, j) point we fix the nonlinear terms G and $|\nabla u|$ at the current k-step and solve for the new k+1 step. Thus we again obtain that

$$\begin{aligned} u_{i,j}^{k+1}\mathbb{S}_{i,j} - u_{i+1,j}^{k+1}\left(C_{i+\frac{1}{2},j}^{k}\right) - u_{i-1,j}^{k+1}\left(C_{i-\frac{1}{2},j}^{k}\right) - u_{i,j+1}^{k+1}\left(C_{i,j+\frac{1}{2}}^{k}\right) \\ -u_{i,j-1}^{k+1}\left(C_{i,j-\frac{1}{2}}^{k}\right) = \chi z_{i,j} \end{aligned} \tag{31}$$

where

$$C_{(i+\frac{1}{2},j)}^{k} = \frac{\alpha\, G_{(i+\frac{1}{2},j)}^{k}}{h|\nabla u|_{(i+\frac{1}{2},j)}^{k}}, \qquad \mathbb{S}_{i,j} = C_{i+\frac{1}{2},j}^{k} + C_{i-\frac{1}{2},j}^{k} + C_{i,j+\frac{1}{2}}^{k} + C_{i,j-\frac{1}{2}}^{k} + \chi, \tag{32}$$

whose corresponding linear system $A(\mathbf{u}^k)\mathbf{u}^{k+1} = \mathbf{z}$ is now never singular and hence solvable.

Note that, setting $b = 1$ and $0 < a < 1$ reduces the modified model to the original CDD model [2]. On the other hand, setting $a = 1$ and $b = 0$ transforms the modified CDD to the TV model [1].

In order not to violate the connectivity principle [1] we must select $a < b$. However if we select a too small compared with b we introduce instability to the linear system (15). Experimentally we found that $5 < \frac{b}{a} < 50$ works well.

With this new proposal for G, the main virtue of the CDD model of reconstructing large scale missing parts is kept with the additional advantage that now we can implement a fast and efficient numerical method.

(*) What about rows for D part. DD is Not true, only weakly DD. We can check that in this case $A(\mathbf{u}^k)$ is a symmetric and sparse block-tridiagonal matrix with the important feature that it is strictly diagonally dominant. To show this, we can choose and arbitrary row of it and see that all the $a_{i,j}$ with $i \neq j$ entries but at most four are equal to zero. The nonzero entries in this row are given by $C^k_{i+\frac{1}{2},j}$, $C^k_{i-\frac{1}{2},j}$, $C^k_{i,j+\frac{1}{2}}$ and $C^k_{i,j-\frac{1}{2}}$ respectively and all are positive. The diagonal entry $a_{i,i}$ on the other hand is computed using (32). Thus, we have that $a_{i,i} > \sum_{i\neq j}^{n} |a_{i,j}|$ in Ω and therefore $A(\mathbf{u}^k)$ is strictly diagonally dominant. Furthermore, because $G \equiv 1$ in $\Omega \backslash D$ and $G = a + b|\kappa|^p$ with $p \geq 1$ in D we can deduce by using the Gerschgorin theorem that $A(\mathbf{u}^k)$ is Positive Definite.

Therefore, the FPGS smoother is simply to apply some Gauss-Seidel relaxation steps to the linear system (15). The FPGS algorithm is defined as $v_h \leftarrow FPGS(v_h, z_h, gsiter, M_h)$ as follows :

Algorithm *(FPGS Smoother)*

Choose initial guess v_h.

For $k = 1$ *to gsiter,*

apply Gauss Seidel iterations to the linear system

$A_h(\boldsymbol{u}_h^k)\boldsymbol{u}_h^{k+1} = \boldsymbol{z}_h$

end

(*) Need to add LFA results to show why 10-50 iterations are needed. We observed that by solving as accurately as possible each fixed point iteration the multigrid algorithm converges faster. However could be unnecessary and a waste of time to do that. Experimentally we found that between 10 and 50 Gauss Seidel iterations (depending on the size of the inpainting domain) give a good convergence rate with reasonable cpu-time consuming.

Remark 1: *We also considered another two local smoother's.* **(*) Not clear here! Need to give details of formulation.** *A local Newton-Gauss-Seidel algorithm, as shown in [22] and [23] which is known to converge only with heavy under-relaxation and with very bad smoothing capabilities. Also we constructed a fixed-point smoother similar to the one developed in [22] and [23] and later on used successfully in [21]. The later was tested in our MG algorithm resulting in very slow convergence and sometimes without reaching the desired quality of reconstruction.*

5 Numerical results

In this section, we shall give results of three different inpainting problems designed to test the performance of the multigrid algorithm, as illustrated in Figures 6, 7 and 8. We also include the problem of Figure 5 to show the performance of the algorithm on a real old photograph. Actually, this experiment is also showed in the original paper of the CDD model [2] where only a reduced size (due to slow convergence of the numerical method) is tested.

We choose several resolutions of the three new problems and display the results in Table 1. To measure the restoration quality, we found it useful to use the Peak-Signal-To-Noise-Ratio (PSNR) to check how similar two images u and u^0 of size $m \times n$ are each other. The PSNR is defined as

$$PSNR = 20\ log_{10}\left(\frac{255}{RMSE(u,u^0)}\right), \quad RMSE(u,u^0) = \sqrt{\frac{\sum_{i,j}(u_{i,j}-u^0_{i,j})^2}{mn}}. \tag{33}$$

In real life situation, such a measure is no possible because u^0 is not known. **(*) Can we show some PSNR/RMSE values for the original image and the newly restored image?**

Comparison. Even for small images like those of size 128×128 the time marching scheme takes thousands of iterations to reach a meaningful result not say convergence and because of this its outcome can take hours or days for larger images to be available. Therefore, practically there is no point of comparison with the multigrid algorithm being the last many times faster. **(*) Need to give some cpu timings even in text, if not in Table 1.**

Full Multigrid. We observed that for very large scale inpainting domains as those of the ring problem illustrated in Figure 7, our MG algorithm is slightly dependent on the initial guess. Even though our MG always converge no matter the initial guess, its rate of convergence was found to be dependent on the initial guess in the cases described above.

In order to reduce this dependence and to improve even more the speed of convergence we constructed a Full Multigrid method (FMG) as described in [26] obtaining even better results, see Table 1.

6 Conclusions

In this paper we developed a fast and efficient nonlinear MG algorithm for solving the CDD model.

Problem	Image Size	MG		FMG	
		MG cycles	CPU	MG cycles	CPU
Lena	128×128	2	8	1	9
	256×256	3	43	1	24
	512×512	3	473	1	218
Bars	128×128	6	26	6	28
	256×256	9	130	9	140
	512×512	8	657	8	689
Ring	128×128	4	33	2	25
	256×256	4	134	2	85
	512×512	3	496	1	217

Tab. 1: Overall results

(*) If Page 10 does not have more details specific to inpainting, and we don't add LFA, then it is hard to claim we "developed" anything. So far only a very slow time marching scheme had been reported and therefore the model only was able to reconstruct very small-size images in an acceptable amount of time. Moreover, the nonilnear MG does not work for the original CDD model of [1].

By finding out why a fixed-point method is not feasible for the original CDD model because the model degenerates when the curvature κ vanishes in flat regions of the image, we then proposed a modified CDD model for which a fixed-point method is feasible. We demonstrated that a nonlinear MG is applicable to the modified model. Numerical results confirmed that the multigrid method with the fixed point type smoother is very fast and the modified model retains the desirable property of the original model of reconnecting the level lines across large distances.

References

[1] Tony F. Chan and Jianhong Shen. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[2] Tony F. Chan and Jianhong Shen. Non-texture inpaintings by curvature-driven diffusions. *J. Visual Communication and Image Representation*, 12(4):436–449, 2001.

[3] Joseph Savage and Ke Chen. An improved and accelerated non-linear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82(8):1001–1015, 2005.

[4] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image

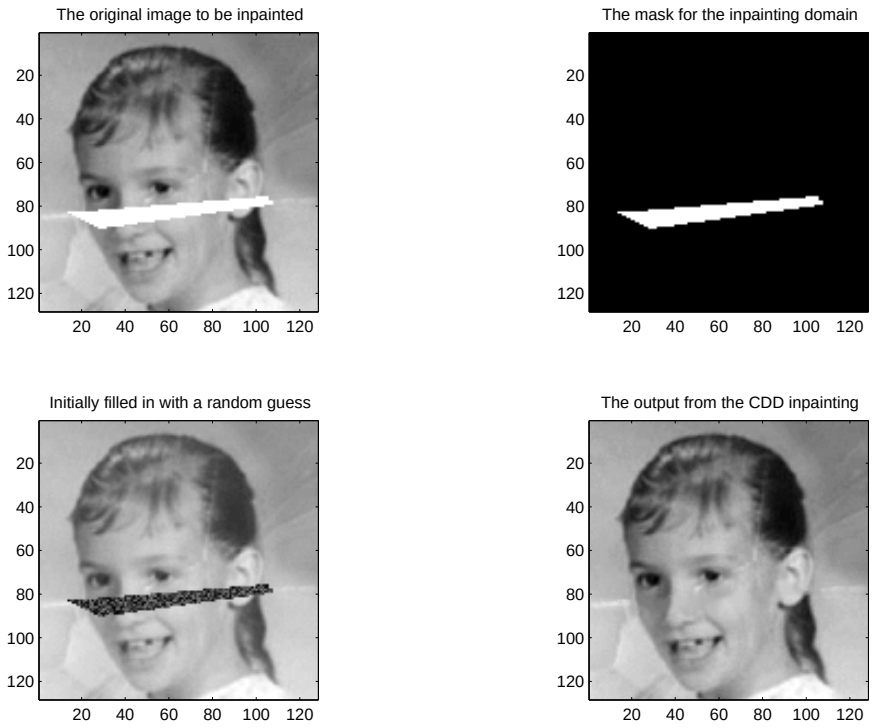

Fig. 5: An example of CDD inpainting for scratch removal in a real old-photograph. Notice the big size of the scratch and the very good reconstruction. We inpainted this image by using our MG algorithm with the following parameters: $a = \frac{1}{5}$, $b = 1$, $gsiter = 10$, $v_0 = 500$, $v_1 = 3$, $v_2 = 3$, $\lambda = 1000$ and $\beta = 10^{-6}$. The algorithm performed 3 V-cycles in only 9.60 seconds.

inpainting. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 417–424. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[5] Tony F. Chan, Jianhong Shen, and Haomin Zhou. Total variation wavelet inpainting. *Journal of Mathematical Imaging and Vision*, 25:107–125, 2006.

[6] Jianwei Gu, Li Zhang, Guoqiang Yu, Yuxiang Xing, and Zhiqiang Chen. X-ray ct metal artifacts reduction through curvature based sinogram inpainting. *Journal of X-Ray Science and Technology*, 14(2):73–82, 2006.

[7] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation, and Depth*, volume 662. Springer-Verlag, Berlin, 1993. Lecture Notes in Computer Science.

[8] S. Masnou and J.M. Morel. Level lines based disocclusion. volume 3, page 259:263, 1998.

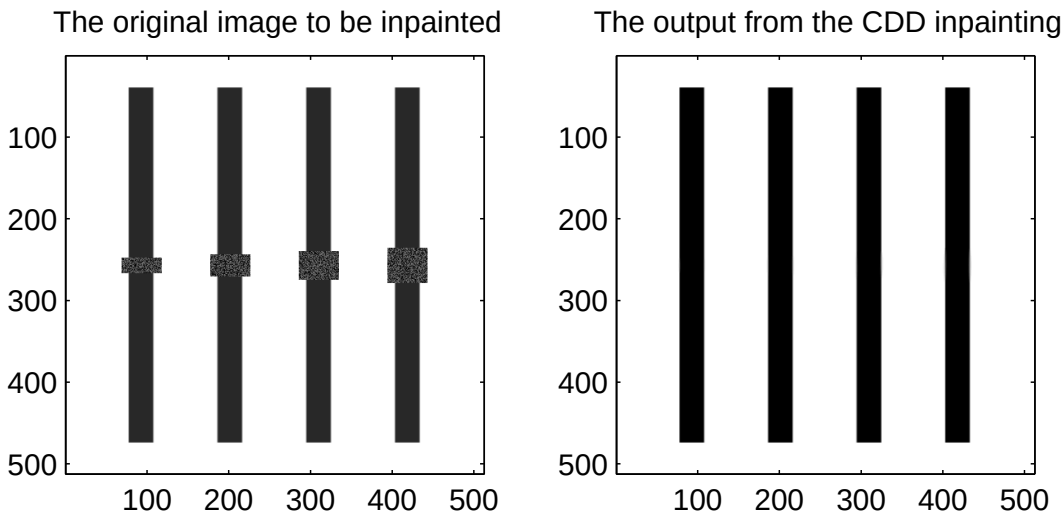

Fig. 6: This experiment shows the power of the CDD model to reconnect contours across very large distances. The size of the largest gap is one and a half times the with of the bar. Again our MG algorithm performed very well reconstructing this image in a reasonable amount of time, see Table 1.

[9] V. Caselles, J.M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7:376–386, 1998.

[10] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:I–355 – I–362, 2001.

[11] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[12] Tony F. Chan, Sung Ha Kang, and Jianhong Shen. Euler's elastica and curvature based inpaintings. *Journal of Applied Mathematics*, 63(2):564–592, 2002.

[13] A.E.H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Dover, New York 4^{th} ed., 1927.

[14] Selim Esedoglu and Jianhong Shen. Digital inpainting based on the mumford-shah-euler image model. *European Journal of Applied Mathematics*, 13:353–370, 2002.

[15] A.L. Bertozzi, S. Esedoglu, and A. Gillet. Inpainting of binary images using the cahn-hilliard equation. *IEEE Transactions on image processing*, 16(1):285–291, 2007.

[16] Harald Grossauer and Otmar Scherzer. Using the complex ginzburg-landau equation for

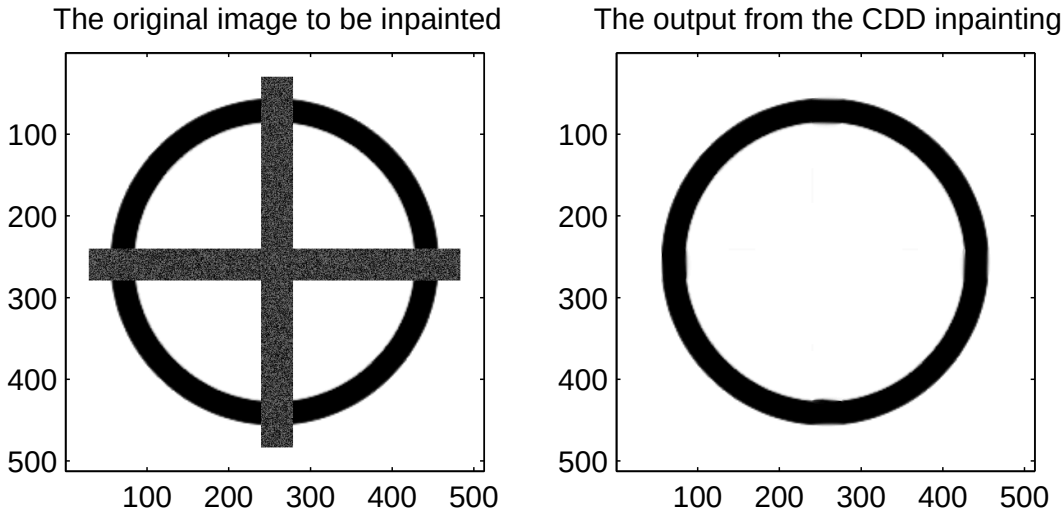

Fig. 7: This experiment is presented to emphasis that the CDD model uses straight lines to reconnect the level line contours. The experiment is also particularly important to show the performance of our MG algorithm when inpainting very large scale domains.

digital inpainting in 2d and 3d. *Scale Space Methods in Computer Vision*, 2695:225–236, 2003. Lecture Notes in Computer Sciences.

[17] Tony F. Chan and Jianhong Shen. Variational image inpainting. *Communications on Pure and Applied Mathematics*, 58:579–619, 2005.

[18] Tony F. Chan and Ke Chen. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation. *Journal of Numerical Algorithms*, 41(4):387–411, 2006.

[19] Antonio Marquina and Stanley Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM Journal on Scientific Computing*, 22(2):387–405, 2000.

[20] Curt Vogel. *Computational Methods For Inverse Problems*. SIAM, 2002.

[21] Noor Badshah and Ke Chen. Multigrid method for the chan-vese model in variational segmentation (to appear). *Communications in Computational Physics*, 2007.

[22] Joseph Savage and Ke Chen. *On Multigrids for Solving a Class of Improved Total Variation Based Staircasing Reduction Models*. Springer, 2007. Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems, CMA, Oslo, 2005.

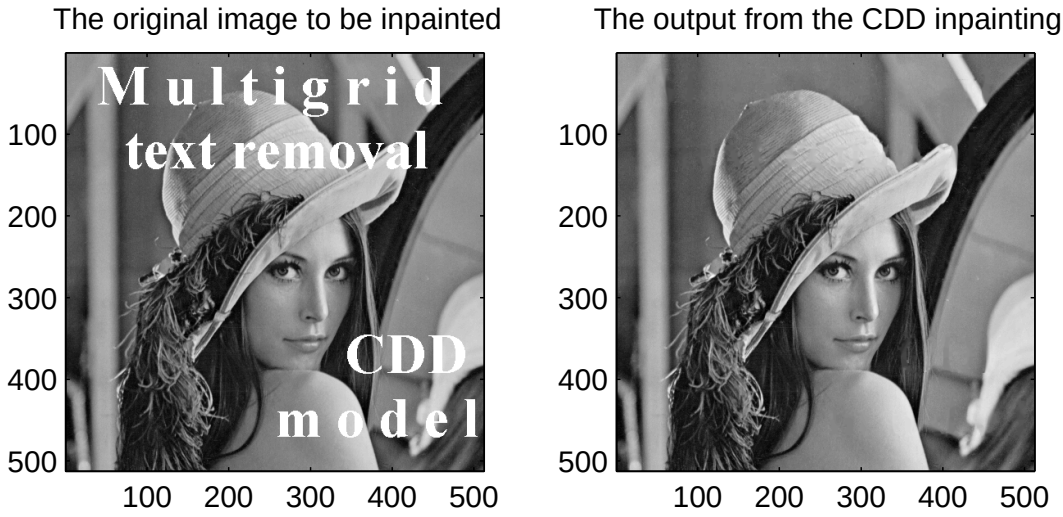

Fig. 8: A practical text removal example in an image of considerable high resolution of 512×512 pixels. The performance of the MG algorithm in both quality of reconstruction and time consuming is remarkable.

[23] Ke Chen and Joseph Savage. An accelerated algebraic multigrid algorithm for total variation denoising. *BIT Numerical Mathematics*, 47:277–296, 2007.

[24] Tony F. Chan and Ke Chen. An optimisation-based multilevel algorithm algorithm for total variation image denoising. *SIAM J. Multi. Mod. Simu.*, 5(2):615–645, 2006.

[25] Tony F. Chan, Ke Chen, and Jamylle L. Carter. Iterative methods for solving the dual formulation arisin from image restoration. *Electronic Transactions on Numerical Analysis*, 26:299–311, 2007.

[26] Ulrich Trottenberg, Cornelis Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, 2001.