Journal of Signal and Information Processing, **2013, 4, 43-51**
doi:10.4236/jsip.2013.43B008 Published Online August 2013 (http://www.scirp.org/journal/jsip)

Color Texture Image Inpainting Using the Non Local CTV Model

Jinming Duan[1], Zhenkuan Pan[1], Wangquan Liu[2], Xue-Cheng Tai[3]

[1]College of Information Engineering, Qingdao University, China; [2]Department of Computing, Curtin University, Australia; [3]Department of Mathematics, University of Bergen, Norway.
Email: duanmujinming@126.com, zkpan@qdu.edu.cn, w.liu@curtin.edu.au, tai@mi.uib.no

Received April, 2013.

ABSTRACT

The classical TV (Total Variation) model has been applied to gray texture image denoising and inpainting previously based on the non local operators, but such model can not be directly used to color texture image inpainting due to coupling of different image layers in color images. In order to solve the inpainting problem for color texture images effectively, we propose a non local CTV (Color Total Variation) model. Technically, the proposed model is an extension of local TV model for gray images but we take account of the coupling of different layers in color images and make use of concepts of the non-local operators. As the coupling of different layers for color images in the proposed model will increase computational complexity, we also design a fast Split Bregman algorithm. Finally, some numerical experiments are conducted to validate the performance of the proposed model and its algorithm.

Keywords: Color Texture Images; Image Inpainting; NL-CTV Model; TV Model; The Split Bregman Algorithm

1. Introduction

Image inpainting, sometimes referred as image completion or disocclusion, has become one of the fundamental problems in image processing and computer vision due to its broad applications in image editing, compression, object removal from a scene, text or scratch removal and old photo restoration etc.. Its aim is to restore the tarnished/missing parts of a broken image using the existing data, which is a typical ill-posed problem in applied mathematics. There are several possible ways to solve this problem and here we focus on solving this problem with Partial Differential Equations (PDEs) or variational methods due to its outstanding performance [1-4,6,8-11]. In this branch of methodology, the exiting approaches can be classified into two categories: the geometry-oriented methods [2-11] and the texture-oriented methods [12-22,27].

For inpainting those gray non-texture images with small scale broken areas, researchers in [2] proposed a third order PDE model to propagate the surrounding information to the inpainted regions based on a heat diffusion equation. Authors in [3] coped with the similar problems by solving Navier-Stokes equations and investigators in [4] proposed total variation (TV) inpainting model inspired by [5]. In [6] the researchers studied the same problem for binary image via using the Cahn-Hilliard model. For inpainting those gray non-texture images with large scale broken areas, a third order PDE method is proposed in [8] based on curvature-driven diffusions (CDD) mechanism, and in fact this work is inspired by the concept of elastic and principle of continuation in computer vision [7]. Also researches in [9-11] solved the same problem using the variational framework including elastic terms.

The above-mentioned approaches are all the geometry-oriented methods and they only used local information, but failed to solve the problems of broken texture inpainting and object removal. The inpainting problem for texture images must resort to the texture-oriented methods. Up to now, there are three categories of methods to solve the inpainting problem for texture images. The first type of approaches is the texture synthesis proposed in [12] based on patches, which has been acted as the fundamental tool for texture image inpainting, and also numerous examplar-based variational models [13-16] have been proposed using the patch similarities. The second type of approach is based on image decomposition in the original domain or transformed frequency domain. For example, the investigators in [17-19] investigated texture image inpainting with the geometry and texture information separately. Typically, in [18], the images to be inpainted are first transformed into the framelet domain so that it is represented by a set of framelet coefficients; then one performs thresholding on framelet coefficients to propagate the information from outside of broken region into it. Finally, one transforms it back to image domain. This framelet-based image inpainting can remove the random noise in an image and enhance the edges and high frequency features of the image. This is one of the state of the art algorithms currently and we will use it for comparison in our experiments. The third type of approaches for texture image inpainting comes from the idea of non local means (NLM) for image denoising with texture preserving [20]. In [21] the authors not only defined the non local gradient and divergence systematically, but also proposed the NL-TV (Non Local Total Variation) models as extensions to their local counterparts for image denoising and inpainting. The elegance for the non local operators in [21] leads to very similar manipulations between non local models and local models in image processing. For example, the authors in [22] used TV with non local graphs for gray image inpainting and super-resolution, and researchers in [23] applied the NL-TV for gray image deblurring. Moreover, authors in [24] used NL-TV for compressive sensing and [25], [26] proposed some image enhancement models involving an NL-TV regularization term. Though the above mentioned algorithms are all based on TV model for image restoration, there is another way for image restoration based on Mumford-Shah model [28], which is usually used for image segmentation. The authors in [27] combined the MTV regularizer term and non local operators in [21] to implement this Gamma-convergence approximated model for color texture image restoration. This motivates us to consider problem in this paper. As we know, such segmentation-based image restoration model suffers from two problems. First, it contains two variables: one is a piecewise smooth function used for the approximation of the original image, and the other is a piecewise function that represents the image edge and equals 0 on the edge sets and 1 on the smooth region, which makes the numerical implementation complicated, which logically results a low computation efficiency. Second, there are three penalty parameters setting up in this model, so the choice of those parameters will become more difficult. In conclusion, there are too many parameters to solve and tune in such model.

The NL-TV model only includes one variable and one penalty parameter in its energy functional and this model always brings much easier computational process and also provides excellent results in preserving texture of gray images. However, to our best of knowledge, this popular model has not been used for color image inpainting. In this paper, we will focus on the NL-TV model and revise this model for inpainting color texture image by combining the non local operators and CTV (Color Total Variation) model proposed in [29]. The proposed model can make use of the good performance of the non local operators in texture image processing as demonstrated for gray images and CTV model in edge preserving for color images. We observed from [31] that the TV model can preserve edges for denoised images, but would fail to preserve color image edges when used to defuse different layers separately. However, the Multichannel Total Variation (MTV) [30] and CTV [29] can have excellent performance in color image edge preserving. Numerous experiments reported in [31] demonstrated that the CTV model outperforms the MTV model, so we adopt the CTV regularizer as the foundation for the proposed model in this paper. We also observed from [31] that CTV has higher complexity than MTV in implementation. In order to overcome the disadvantage of low efficiency associated with CTV in the proposed model, we will also design a fast Split Bregman algorithm for the proposed model.

Technically, the standard Split Bregman algorithm for TV model in [32] is redeveloped through introducing an auxiliary variable and a Bregman iterative parameter with an aim to transform the original model into two simple sub problems. These two sub problems can be solved via alternating optimization technique; further the previous Euler-Lagrange equation with curvature associated with the CTV regularizer term is replaced with a simple one only associated with the Laplacian [31]. Though a generalized soft thresholding formula is derived in an analytical form for TV model in [32], and this simplifies the computation complexity significantly there. However, we noted that for the Split Bregman algorithm of the proposed CTV model, the exact generalized soft thresholding formula can not be derived as elegantly as in [32], and here we design an approximate one to simplify the calculations.

In summary, the contributions of this paper can be summarized as follows. First, we propose a model which can solve the inpainting problem for color texture images by using the CTV model in [29] and the non local operators in [21]. This model combines the advantages of the CTV and the non-local operators nicely and can produce a high performance. Second, In order to improve the implementation efficiency of the proposed model, we develop a fast Split Bregman algorithm based on a simple discrete finite difference scheme. Also we found that the direct application of NL-TV model to color images for inpainting does not work properly. Finally, we validate the proposed model and algorithm via extensive experiments.

2. NL-TV Model for Image Inpainting and Its Split Bregman Algorithm

In this section, we first introduce some important concepts, definitions and technical algorithms used in the remaining parts of this paper. First we present the concept of non-local operators.

2.1. The Non Local Operators and the Split Bregman Algorithm

Due to the significance of the non local operators played in this paper, we first present the relevant definitions provided in [21].

Let $\Omega \subset R^n$ be the domain of a gray image and $x \in \Omega$, $u(x):\Omega \to R$ is a real function defined on Ω to represent the pixel values of an image. The non local gradient for two points x and y in the image is defined as

$$\nabla_{NL}u(x,y) \triangleq \left(u(y)-u(x)\right)\sqrt{w(x,y)} \tag{1.1}$$

where, $w(x,y):\Omega\times\Omega \to R$ is a non-negative, symmetric weight between points x, y for any pair $(x,y)\in\Omega\times\Omega$ and it measures the similarities of these two points. It should be noted that eq (1.1) is not a vector field in the standard sense, it is only a mapping: $\Omega\times\Omega\to R$. Now we denote any NL mapping as $\vec{p} = p(x,y):\Omega\times\Omega\to R$. For a pair of NL mappings, their dot product is defined as follows.

$$\left(\vec{p}_1\cdot\vec{p}_2\right)(x) \triangleq \int_\Omega p_1(x,y)\,p_2(x,y)\,dy \tag{1.2}$$

And their inner product is defined as

$$\left\langle \vec{p}_1,\vec{p}_2\right\rangle \triangleq \left\langle \vec{p}_1\cdot\vec{p}_2,1\right\rangle = \int_{\Omega\times\Omega} p_1(x,y)\,p_2(x,y)\,dxdy \tag{1.3}$$

The magnitude of a NL mapping will be given by

$$\left|\vec{p}\right|(x) \triangleq \sqrt{\left(\vec{p}\cdot\vec{p}\right)} = \sqrt{\int_\Omega \left(p(x,y)\right)^2 dy} \tag{1.4}$$

With the above inner product, the non local divergence $\nabla_{NL}\cdot\vec{p}(x):\Omega\times\Omega\to\Omega$ will be defined as the adjoint of the non local gradient, which is given by

$$\left(\nabla_{NL}\cdot\vec{p}\right)(x) \triangleq \int_\Omega \left(p(x,y)-p(y,x)\right)\sqrt{w(x,y)}dy \tag{1.5}$$

Finally, the Laplacian of a point x in an image can be defined now by

$$\begin{aligned}\Delta_{NL}u(x) &\triangleq \frac{1}{2}\nabla_{NL}\cdot\left(\nabla_{NL}u(x)\right)\\ &= \int_\Omega \left(u(y)-u(x)\right)w(x,y)\,dy\end{aligned} \tag{1.6}$$

Based on the above mentioned definitions, we can give the NL norm of the NL gradient for a function *u* as follows.

$$\left|\nabla_{NL}u\right|(x) = \sqrt{\int_\Omega \left(u(y)-u(x)\right)^2 w(x,y)\,dy} \tag{1.7}$$

All above preliminaries are for a function of an image. Next we explain the NL-TV models for gray images and their Split Bregman algorithm. For a broken scalar texture image $f(x):\Omega\to R$, let $D\subset\Omega$ denote the domain to be inpainted, the proposed NL-TV model for a gray image inpainting reported in [21] is given as follow.

$$\underset{u}{Min}\left\{E(u) = \int_\Omega \left|\nabla_{NL}u\right|(x)\,dx + \frac{1}{2}\int_\Omega \lambda_D\left(u-f\right)^2 dx\right\} \tag{2}$$

where, $\lambda_D(x) = \begin{cases}0 & x\in D\\ 1 & x\in\Omega/D\end{cases}$ is the mask function to represent the broken region. This problem aims to find u in the already known broken region D such that (2) is minimized.

The computation of local TV model utilizing the non local operators becomes very expensive, which will be demonstrated in experiment section. In order to improve its computational efficiency, authors in [21] designed a dual method for NL-TV models. Such dual metod is not suitable for our proposed NL-CTV model in this paper due to a fact that it involves complicated coupling feature in CTV regularizer term. Here, we alternatively extend the Split Bregman algorithm reported in [32] to our proposed NL-CTV model which is much easier than the dual method.

The author in [33] proposed the Split Bregman algorithm for NL-TV denosing model. Here, we first present their algorithm for the NL-TV inpainting model as reported in [32]. To present the Split Bregman algorithm for (2), an NL auxiliary variable $\vec{v} = v(x,y):\Omega\times\Omega\to R$ is introduced and the objective function is transformed into the following:

$$\underset{u,\vec{v}}{Min}\left\{E(u,\vec{v}) = \int_\Omega \left|\vec{v}\right|(x)\,dx + \frac{1}{2}\int_\Omega \lambda_D\left(u-f\right)^2 dx\right\} \tag{3}$$

$$\text{s.t. } \vec{v} = \nabla_{NL}u \tag{4}$$

The constraint $\vec{v} = \nabla_{NL}u$ is enforced using the efficient Bregman iteration by introducing a Bregman parameter $\vec{b} = b(x,y):\Omega\times\Omega\to R$ as reported in [32]. Then we can transform (3) into the following iterative optimization formulation.

$$\begin{aligned}E(u,\vec{v}) &= \int_\Omega \left|\vec{v}\right|(x)\,dx + \frac{1}{2}\int_\Omega \lambda_D\left(u-f\right)^2 dx\\ &+\frac{\theta}{2}\int_\Omega \left|\vec{v}-\nabla_{NL}u-\vec{b}^{k+1}\right|^2(x)\,dx\end{aligned} \tag{5}$$

with constraints $\vec{b}^{k+1} = \vec{b}^k + \nabla_{NL}u^k - \vec{v}^k, \vec{b}^0 = \vec{v}^0 = 0$. By using the same technique as reported in [32], first fixing $\vec{v}$ for u, and then fixing u for $\vec{v}$, we can obtain the solution of Euler-Lagrange equation of *u* and the generalized soft thresholding formula of $\vec{v}$ via such alternating minimization process as stated in (6) and (7). A fast approximate solution of (6) is provided by a Gauss-Seidel iterative scheme, and it is very convenient to find an analytic solution of (7) without any iteration. However, when it comes to NL-CTV model, such exact soft thresholding formula as (7) can not be directly derived that we could not extend the efficient Split Bregman algorithm to our proposed model directly. Therefore, in the next section, we will focus on this tough problem.

$$\lambda_D(u-f)+\theta\nabla_{NL}\cdot\left(\vec{v}^k-\nabla_{NL}u-\vec{b}^{k+1}\right)=0 \tag{6}$$

$$\vec{v}^{k+1}=Max\left(\left|\nabla_{NL}u^{k+1}+\vec{b}^{k+1}\right|-\frac{1}{\theta},0\right)\frac{\nabla_{NL}u^{k+1}+\vec{b}^{k+1}}{\left|\nabla_{NL}u^{k+1}+\vec{b}^{k+1}\right|} \tag{7}$$

2.2. The NL-CTV Model and Its Split Bregman Algorithm

The previous model (2) is for gray images and if we use (2) directly to different layers separately for color image denoising or inpainting, and we found that this will lead to smear edges as demonstrated in **Figure 6**, though it can be solved via coupled regularizers such as MTV regularizer [30] or CTV regularizer [29]. The MTV and CTV have excellent performance in color image edge preserving as reported in [31]. However, numerous experiments reported in [31] demonstrated that the CTV model outperforms the MTV model, so we adopt the CTV regularizer as the foundation for the proposed model. Therefore, in this paper, we extend the CTV to NL-CTV model for color texture image inpainting. For color image denoising, the authors in [29] have proposed the following CTV model

$$\underset{u}{Min}\left\{E(u)=\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\nabla u_i|(x)dx\right)^2}+\frac{\lambda}{2}\sum_{i=1}^{n}\int_\Omega(u_i-f_i)^2dx\right\} \tag{8}$$

where $f=(f_1,f_2,\ldots,f_n)$ is the original image, $u=(u_1,u_2,\ldots,u_n)$ is the restored image. Based on these results, we propose the following NL-CTV model by combining (2) and (8) for color texture image inpainting.

$$\underset{u}{Min}\left\{E(u)=\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\nabla_{NL}u_i|(x)dx\right)^2}+\frac{1}{2}\sum_{i=1}^{n}\int_\Omega\lambda_D(u_i-f_i)^2dx\right\} \tag{9}$$

It should be noted that (2) is an NL-TV model and can only use to deal with the inpainting problem for gray images. By taking account of the coupling of different layers of color images and introducing the coupled NL-CTV regularizer term

$$\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\nabla_{NL}u_i|(x)dx\right)^2} \tag{10}$$

which is inspired by (8), we proposed the model (9). One can see that the differences between models (8) and (9) are as follow: First, model (8) is a denosing model and (9) is a inpainting model, and λ in (8) is a penalty parameter that ensures that the denoised image is as close as possible to the original image; but λ_D in (9) is a mask function that labels the broken region of image. Second, model (8) replaces the local CTV regularizer term

$$\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\nabla u_i|(x)dx\right)^2} \tag{11}$$

with NL-CTV regularizer term (10) by using the non local gradient operator $\nabla_{NL}u$. Such choice is done since the NL-CTV regularizer term will lead to an extremely complicated Euler-equation similar to (9), which is very difficult for discrete numerical calculation.

In order to solve (9) efficiently, we need to design a new Split Bregman algorithm similar to that reported in [32]. For such purpose and by using the same manner as reported in last section, we introduce an auxiliary variable $\vec{v}=(\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_n)$ and a Bregman iterative parameter $\vec{b}=(\vec{b}_1,\vec{b}_2,\ldots,\vec{b}_n)$, and then transform (9) into the following iterative optimization formulation.

$$\begin{aligned}E(u,\vec{v})=&\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\vec{v}_i|(x)dx\right)^2}+\frac{1}{2}\sum_{i=1}^{n}\int_\Omega\lambda_D(u_i-f_i)^2dx\\&+\frac{\theta}{2}\sum_{i=1}^{n}\int_\Omega\left|\vec{v}_i-\nabla_{NL}u_i-\vec{b}_i^{k+1}\right|^2(x)dx\end{aligned} \tag{12}$$

With constraints $\vec{b}_i^{k+1}=\vec{b}_i^k+\nabla_{NL}u_i^k-\vec{v}_i^k,\ \vec{b}_i^0=\vec{v}_i^0=0$. By using the alternating minimization strategy, we can obtain the Euler-Lagrange equations for u and $\vec{v}$ separately as follows.

$$\lambda_D(u_i-f_i)+\theta\nabla_{NL}\cdot\left(\vec{v}_i^k-\nabla_{NL}u_i-\vec{b}_i^{k+1}\right)=0 \tag{13}$$

$$\theta\left(\vec{v}_i-\nabla_{NL}u_i^{k+1}-\vec{b}_i^{k+1}\right)+\frac{\int_\Omega|\vec{v}_i|(x)dx}{\sqrt{\sum_{i=1}^{n}\left(\int_\Omega|\vec{v}_i|(x)dx\right)^2}}\frac{\vec{v}_i}{|\vec{v}_i|(x)}=0 \tag{14}$$

In order to show how to implement (13) in detail, we give the discrete version of (13) as follows.

$$\lambda_{Dl}(u_l-f_l)+\theta\left(\nabla_{NL}\cdot\vec{v}^k-\nabla_{NL}\cdot(\nabla_{NL}u)-\nabla_{NL}\cdot\vec{b}^{k+1}\right)_l=0 \tag{15}$$

According to (1.5) and (1.6), we have

$$\left(\nabla_{NL}\cdot\vec{v}^k\right)_l=\sum_h\left(v_{l,h}^k-v_{h,l}^k\right)\sqrt{w_{l,h}} \tag{161}$$

$$\left(\nabla_{NL}\cdot\vec{b}^{k+1}\right)_l=\sum_h\left(b_{l,h}^{k+1}-b_{h,l}^{k+1}\right)\sqrt{w_{l,h}} \tag{16.2}$$

$$\left(\nabla_{NL}\cdot(\nabla_{NL}u)\right)_l=2\sum_h(u_h-u_l)w_{h,l} \tag{16.3}$$

where the non-negative weight $w(x,y)$ is chosen as

$$w(x,y)=\exp\left\{-\frac{G_\sigma*\left(\left\|u(x+\cdot)-u(y+\cdot)\right\|\right)^2}{r^2}\right\} \tag{17}$$

In which G_σ is the Gaussian kennel function, r is the thresholding parameter for similarities between two patch windows, and its discrete version is given by

$$w_{l,h}=\exp\left\{-\frac{G_\sigma*\left(\left\|u(l+\cdot)-u(h+\cdot)\right\|\right)^2}{r^2}\right\} \tag{18}$$

In order to present the algorithm clearer, we use the following diagram (**Figure 1**) to demonstrate the idea. When given a point *l* in the image, we can have a square search window and a patch window, in which *l* is center point of them. *h* represents any pixel point in the search window. When *h* is fixed in the search window, a square patch window is created subsequently and the weight $w_{l,h}$ between *l* point and *h* point can be computed as (18), in which their respective patch windows are needed.

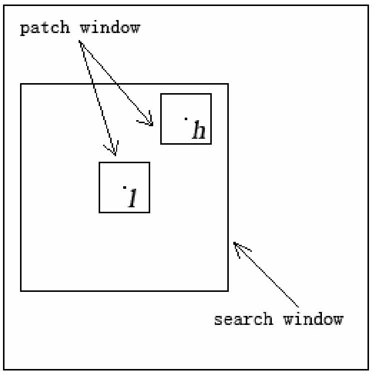

Figure 1. Illustration of patch window and search window. *l* and *h* are the position of two pixel points in the image, but *h* is only fixed at the search window in which *l* is the center point. In addition, the two patch windows in which *l* and *h* are the center points respectively are used to compute the weight *w*$w_{l,h}$.

Then the discrete iterative scheme of u_l^{k+1} can be obtained by

$$u_l^{k+1} = \frac{1}{\lambda_{Dl} + 2\theta\sum_h w_{l,h}}\Bigg(2\theta\sum_h u_h^k w_{l,h} + \lambda_{Dl} f_l - \theta\Big(\sum_h\left(v_{l,h}^k - v_{h,l}^k\right)\sqrt{w_{l,h}} - \sum_h\left(b_{l,h}^{k+1} - b_{h,l}^{k+1}\right)\sqrt{w_{l,h}}\Big)\Bigg) \quad (19)$$

Now considering the equation (14), we can obtain v^{k+1} as follows.

$$\vec{v}_i^{k+1} \approx Max\left(\left|\nabla_{NL}u_i^{k+1} + \vec{b}_i^{k+1}\right| - \frac{\int_\Omega\left|\vec{v}_i^k\right|(x)\,dx}{\theta\sqrt{\sum_{i=1}^n\left(\int_\Omega\left|\vec{v}_i^k\right|(x)\,dx\right)^2}}, 0\right) \cdot \frac{\nabla_{NL}u_i^{k+1} + \vec{b}_i^{k+1}}{\left|\nabla_{NL}u_i^{k+1} + \vec{b}_i^{k+1}\right|} \quad (20)$$

Obviously, (20) is not the exact generalized soft thresholding formulas as one should expect. In order to calculate it effectively, we propose the above approximate formulations to simplify implementation of (14) and speed up computation. Although (20) is not the analytic solution for $\vec{v}$ and may cause a little error, the Bregman iterations may correct it automatically. This issue is confirmed in our experiments in section 4 **Figure 6**. By using the same manner as (19), (20) can be also rewritten in a discrete version as below.

$$\vec{v}_{i,l}^{k+1} \approx Max\left(\left|A_{i,l}^{k+1}\right| - \frac{B_i^k}{\theta\sqrt{\sum_{i=1}^n\left(B_i^k\right)^2}}, 0\right)\frac{A_{i,l}^{k+1}}{\left|A_{i,l}^{k+1}\right|} \quad (21)$$

where,

$$A_{i,l}^{k+1} = \sum_h\left(\left(u_{i,h}^{k+1} - u_{i,l}^{k+1}\right)\sqrt{w_{l,h}} + b_{i,l,h}^{k+1}\right) \quad (22.1)$$

$$B_i^k = \sum_l\sqrt{\sum_h\left(v_{i,l,h}^k\right)^2} \quad (22.2)$$

Now it is time for us to give the NL-CTV algorithm in detail.

NL-CTV Algorithm

```
1. Initialization: $k = 0$, $\vec{b}_i^0 = \vec{v}_i^0 = 0$, $u_i^0 = f_i^0$; for $i$ = 1, …, $n$;
2. Repeat
3.     Update each weight $w_i$ by (18);
4.     Compute each $\vec{b}_i^{k+1} = \vec{b}_i^k + \nabla_{NL}u_i^k - \vec{v}_i^k$;
5.     Compute each $u_i^{k+1}$ from (19);
6.     Compute each $\vec{v}_i^{k+1}$ from (21);
7.     $k=k+1$;
8. Until a stopping criterion is satisfied.
```

In above ***NL-CTV Algorithm***, the stopping criterion is usually chosen as $\left|E^{k+1} - E^k\right|/E^k \le \varepsilon$, where E is the energy in the proposed model and ε is a very small tolerance parameter.

3. Numerical Experiments and Analysis

In this section, we will present several numerical experiments to show the effectiveness and performance of the NL-CTV model proposed in this paper for color texture image inpainting in terms of vision and peak signal to noise ratio (PSNR). PSNR is defined as in (23), and all experiments are performed using the Matlab 2010 b on a Windows 7 platform with an Intel Core 2 Duo CPU at 2.33 GHz and 2GB memory.

$$PSNR = 10\log_{10}\frac{nMN \times MAX^2}{\sum_{i=1}^n \| f_i - u_i \|_2^2} \quad (23)$$

where n stands for the layers of the color image. M and N are respective the height and width of the original image. MAX is 255. u is the restored image, and f is the original image.

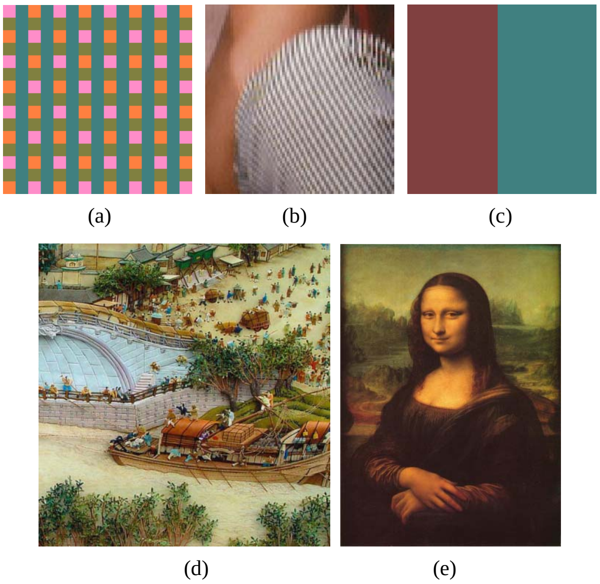

Figure 2. Original image. (a) Color chess board Image. (b) Color cloth Image. (c) Color bar image.(d) River of Wisdom Image. (e) Monalisa Image.

We first present the original images in **Figure 2** and we can make visual comparisons with the inpainted images subsequently and then compute their exact PSNRs. **Figure 2 (a)** is a synthetic color texture image and **Figure 2(b)** is real color texture image. These two images are set up for testing the capability of our proposed model for color texture image inpainting. **Figure 2(c)** is a color bar image presented here to observe the edge preserving phenomenon. The last two images are two real famous images blended with texture and non-texture, and we will use them to further demonstrate their inpainting results in the subsequent experiments.

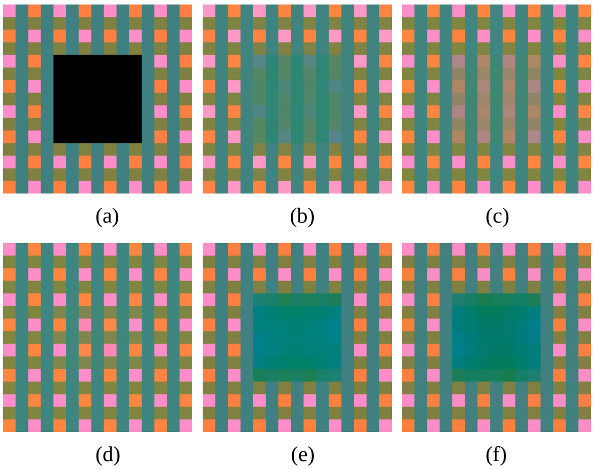

Figure 3. Color chess board inpainting. (a) Image with mask marked by black square; (b)-(c) Intermediate results by proposed NL-CTV model; (d) Final result by proposed NL-CTV model; (e) Final result by TV inpainting method [4]; (f) Final result by elastica inpainting method [11].

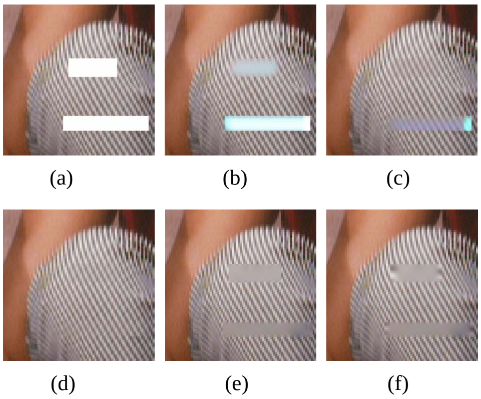

Figure 4. Color cloth inpainting. (a) Image with mask marked by white rectangle; (b)-(c) Intermediate results by proposed NL-CTV model; (d) Final result by proposed NL-CTV model; (e) Final result by TV inpainting method [4]; (f) Final result by elastica inpainting method [11].

In the first experiment as shown in **Figure 3**, we aim to restore a toy image with regular grids and a large black broken region. Here we also show the inpainting results of TV model proposed in [4] and elastica inpainting model reported in [11]. The advantages of TV inpainting model are its simple manipulation and fast computation, but the major drawback of this model is that it does not restore satisfactorily a single object when the disconnected remaining parts are separated far apart by the inpainting domain. One can observe in **Figure 3(e)** that one cannot get a desirable result when dealing with the non-texture images of large broken domain. In order to overcome this problem, the elastica inpainting model [11] is subsequently tested. Remind that this is a model only suitable for non-texture images with large broken domain, and the result is shown in **Figure 3(f)**. In fact, the model in [11] has meaningful statistical fluctuations in textures and the textures are often smoothed out by its PDEs. In conclusion, we notice that the proposed NL-CTV using the non local information can obtain a perfect result as shown in **Figure 3(d)** in comparison with the original image in **Figure 2(a)**. In fact, the TV inpainting model [4] and elastica inpainting model [11] cannot find the changes of the texture and thus could not recover this large broken texture region as shown **Figures 3(e) (f)**.

In order to validate our proposed model with the proposed fast algorithm on real texture images and demonstrate a further illustration, we now set up the second experiment properly. In this experiment, the regions of missing data are quite large with respect to textures as shown in **Figure 4** (size is 81×81). The size of missing region in **Figure 4(a)** is 10×23 above and 8×46 below, and that means the proportion of total texture missing part is about 10% of the original image. Here, we use 41×41 of search window to inpaint the missing region. In this case, the proposed NL-CTV model can obtain perfect results while the other two approaches failed as shown in **Figure 4**. Next we will show that choosing a suitable search window for our model is a very crucial issue. In **Figure 5**, several approximations of **Figure 4(a)** have been calculated with different sizes of search windows and they are listed for comparison with each other. When the sizes of search windows are 11×11 and 21×21, the inpainting results are obviously not satisfactory as shown in **Figures 5(a)** and **(b)**. So two relative larger ones such as sizes of 31×31 and 41×41 are used further and we can see better visual results in **Figures 5(c)** and **(d).** However, the final PSNR values from Table1 tell us that the best result comes from implementation with the largest search window. However, the computation time becomes much more expensive with increase of the search window size consequently as demonstrated in Table 1. In conclusion, for these diverse search windows, we find that small search window is not sufficient to the success of inpainting the large missing part of the texture image, and usually a large search window is needed, which is usually time consuming. In order to show time complexity, the time of constructing the weight function $w(x,y)$ and the total computation time in the experiments are shown in **Table 1**. In fact, in our experiment to obtain **Figure 5**, we update $w(x,y)$ every 30 iterative steps in order to improve their computational efficiency, and the total iterative numbers are set to be 300. Technically, decreasing the updating frequency of the weight function $w(x,y)$ will lead to increasing of iteration steps, and how to balance them is our future research topic.

Except for the visual results, we can calculate the quantity evaluations for our inpainting results in **Figure 3**, **Figure 4** with different techniques as shown in **Table 2**. The results in this Table show a great success of our proposed model in inpainting both the synthetic and real images with color texture.

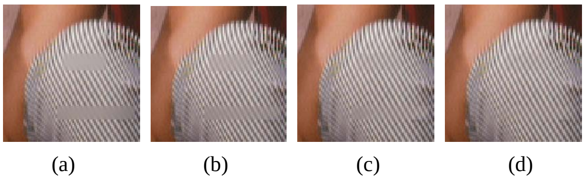

Figure 5. Test the effect of different search windows on inpainting results. (a)-(d) are results with search windows of 11 × 11, 21 × 21, 31 × 31 and 41 × 41, respectively.

The NL-TV model is for gray images and if we use it directly to different layers separately for color image inpainting, we find that this will lead to smear edges as demonstrated in **Figure 6(b)**. However, considering the coupling of different layers of color images and using the NL-CTV regularizer term in our proposed model, one can observe that the edge is perfectly preserved as shown in **Figure 6(c)**. Moreover, **Figure 6(d)** presents the energy function with each iteration of the proposed algorithm and it shows the convergence of the proposed algorithm in this simulation. This experiment not only proves that our model does an excellent job in color image edge preserving but also shows that the Split Bregman designed for the proposed model is convergent.

Table 1. Comparisons of PSNRs and computation time using different search windows.

Images	**Figure 2(a)**	**Figure 2(b)**	**Figure 2(c)**	**Figure 2(d)**
Size of the search window	11×11	21×21	31×31	41×41
PSNR of the restored image using different search window	30.076	32.601	36.581	40.054
Time(s) of constructing the weight function	15.054	47.464	96.659	149.508
Total computation time(s)	623.867	1510.226	2786.269	4666.216

Table 2. Comparisons of PSNRs using different methods.

Experiments	**Figure 3**	**Figure 4**
PSNR of the damaged image	11.795	17.873
PSNR of the restored image using NL-CTV model	36.059	40.054
PSNR of the restored image using TV inpainting method	15.147	26.665
PSNR of the restored image using elastica inpainting method	16.842	28.862

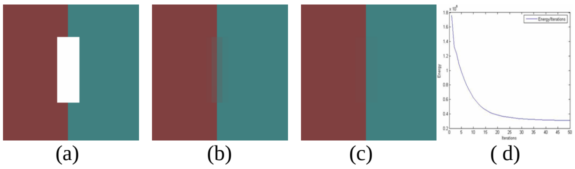

Figure 6. Edge preserving test using different non local model and show convergence of NL-CTV. (a) Damaged image; (b) Final result by NL-TV directly using to different layers of color image; (c) Final result by proposed NL-CTV. (d) Energy decreasing plot of NL-CTV.

In the next two experiments, we will compare the proposed model with the state-of-the-art method proposed in [18] both quantitatively and visually. **Figure 7** presents the picture of River of Wisdom inpainting in different cases, while **Figure 8** shows the picture of Monalisa inpainting. In **Figure 7**, We made the damage types marked by varied color paintings to illustrate that our model can adaptively inpaint such complicated contamination effectively. Though we cannot see visual differences between **Figures 7(b)** and **(c)** but the PSNR values acquired by the proposed method is about 4.3 higher than that obtained by the state-of-the-art method [18]. In order to see the detailed difference, we crop a small block from the damaged images in two cases as shown in **Figures 7(e)-(f)** and **Figures 8(e)-(f)**, we can see that the proposed NL-CTV performs better in micro structures for the poles of the ship in **Figure 7(f)**. Similarly, the visual effect and PSNR value show the benefit of the proposed model and the micro structures in hand show the edge persevering differences in **Figures 8(e)-(f)**. These experiments demonstrate the advantages of the proposed model in color texture image inpainting.

In the above experiments, the damaged areas D are different one can see that in all cases, the proposed model works well with extraordinary performance. Also the types of damages are different.

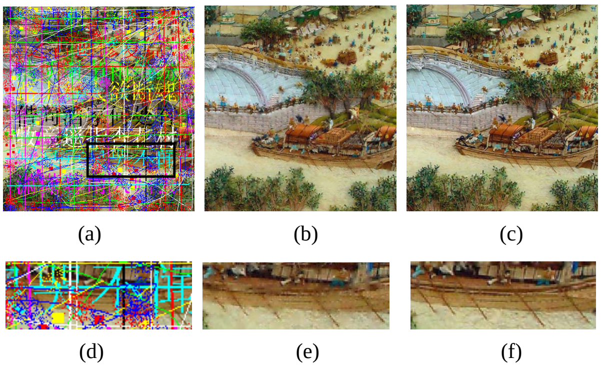

Figure 7. The River of Wisdom inpainting. (a) Damaged image; (b) Final result by [18], PSNR = 31.58; (c) Final result by proposed NL-CTV model, PSNR = 35.89; (d)-(f) Zoomed small subregions (indicated by black rectangle in (a)) of the images in (a)-(c) for detail comparison.

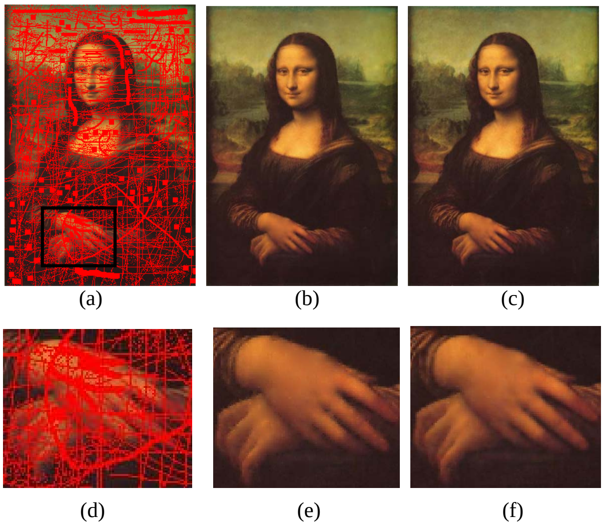

Figure 8. Monalisa inpainting. (a) Damaged image; (b) final result by method in [18], PSNR = 36.15; (c) final result by proposed NL-CTV model, PSNR = 39.85; (d)-(f) zoomed small subregions (indicated by black rectangle in (a)) of the images in (a)-(c) for detail comparison.

4. Conclusions

In this paper, by using the relevant concepts of non local operators and the CTV model, we proposed the NL-CTV model for color texture image inpainting as an extension of CTV model for color image denoising. In this model, the mask is automatically assigned. Then we design a new Split Bregman algorithm and provide their implementations in detail. Numerical experiments validate the performance of the proposed model for color texture image inpainting in different cases.

REFERENCES

[1] G. Aubert. and P. Kornprobst, “Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations 2nd ed. Berlin, Germany: Springer-Verlag, 2006.

[2] M. Bertalmío, G. Sapiro, V. Caselles and C. Ballester, “Image Inpainting,” *In Proceedings of SIGGRAPH*, 2000, pp. 417-424. doi:10.1145/344779.344972

[3] M. Bertalmio, A. Bertozzi and G. Sapiro, “Navier-Stokes, Fluid-Dynamics, and Image and Video Inpainting,” *In Proc. of IEEE-CVPR*, 2001, pp. 355-362.

[4] T. Chan and J. Shen, “Mathematical Models of Local Non-texture Inpaintings,” *SIAM Journal of Applied Mathematics*, Vol. 62, No. 3, 2002, pp. 1019-1043. doi:10.1137/S0036139900368844

[5] L. Rudin, S. Osher and E. Fatemi, “Nonlinear Total Variation Based Noise Removal Algorithms,” *Physica D*:Nonlinear Phenomena, Vol. 60, No. 1/4, 1992, pp. 259-268. doi:10.1016/0167-2789(92)90242-F

[6] A. Bertozzi, S. Esedoglu and A. Gillette, “Inpainting of Binary Images Using the Cahn-Hilliard Equation,” *IEEE Trans. Image Processing*, Vol. 16, No. 1, 2007, pp. 285-291. doi:10.1109/TIP.2006.887728

[7] M. Nitzberg, D. Mumford and T. Shiota, *Filering, segmentation, and depth, LNCS*, Vol. 662, Berlin, Germany: Springer-Verlag, 1993. http://dx.doi.org/10.1007/3-540-56484-5

[8] T. Chan and J. Shen, “Non-texture Inpainting by Curvature-driven Diffusions (CDD),” *Journal of Visual Communication and Image Representation.*, Vol. 12, No. 4, 2001, pp. 436-449. doi:10.1006/jvci.2001.0487

[9] T. Chan, S. Kang and J. Shen, “Euler's Elastica and Curvature-based Image Inpainting,” *SIAM J. Appl. Math.*, Vol. 63, No. 2, 2002, pp. 564-592.

[10] S. Esedoglu and J. Shen, “Digital Inpainting Based on the Mumford-Shah-Euler Image Model,” *European Journal of Applied Mathematics*, Vol. 13, No. 4, 2002, pp. 353-370. doi:10.1017/S0956792502004904

[11] X. Tai, J. Hahn and G. Chung, “A Fast Algorithm for Euler’s Elastica Model Using Augmented Lagrangian Method,” *SIAM J. Appl. Math.*, Vol. 4, No. 1, 2011, pp. 313-344.

[12] A. Efros and T. Leung, “Texture Synthesis by Nonpara-Metric Sampling,” *In Proc. of the IEEE ICCV*, 1999, pp. 1033-1038.

[13] A. Criminisi, P. Pérez, and K. Toyama, "Region Filling and Object Removal by Exemplar-based Inpainting," *IEEE Transactions on Image Processing*, Vol. 13, No. 9, 2004, pp. 1200-1212. doi:10.1109/TIP.2004.833105

[14] Z. Xu and S. Jian, "Image Inpainting by Patch Propagation Using Patch Sparsity," *IEEE Transactions on Image Processing*, Vol. 19, No. 3, 2010, pp. 1153-1165.

[15] P. Arias, V. Caselles, G. Facciolo and G. Sapiro, "A Variational Framework for Exemplar-based Image Inpainting," *International Journal of Computer Vision*, Vol. 93, No. 3, 2011, pp. 1-29. doi:10.1007/s11263-010-0418-7

[16] P. Arias, V. Caselles and G. Facciolo, "Analysis of a Variational Framework for Exemplar-Based Image Inpainting," *SIAM Multiscale Modeing Simulation*, Vol. 10, No. 2, 2012, pp. 473-514. doi:10.1137/110848281

[17] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, "Simultaneous Structure and Texture Image Inpainting," *IEEE Transactions Image Processing*, Vol. 12, No. 8, 2003, pp. 882-889. doi:10.1109/TIP.2003.815261

[18] J. Cai, R. Chan and Z. Shen, "A Framelet-based Image Inpainting Algorithm," *Applied and Computational Harmonic Analysis*, Vol. 24, No. 2, 2008, pp. 131-149. doi:10.1016/j.acha.2007.10.002

[19] J. Cai, R. Chan and Z. Shen, "Simultaneous Cartoon and Texture Inpainting," *Inverse Problems and Imaging*, Vol. 4, No. 3, 2010, pp. 379-395. doi:10.3934/ipi.2010.4.379

[20] A. Buades, B. Coll and J. Morel, "A Review of Image Denoising Algorithms, with a New One," *SIAM Multiscale Modeling Simulation*, Vol. 4, No. 2, 2005, pp. 490-530. doi:10.1137/040616024

[21] G. Gilboa and S. Osher, "Nonlocal Operators with Applications to Image Processing," *SIAM Multiscale Modeling Simulation*, Vol. 7, No. 3, 2008, pp. 1005-1028. doi:10.1137/070698592

[22] G. Peyré, S. Bougleux and L. Cohen, "Non-local Regularization of Inverse Problems," *LNCS*, Vol. 5304, 2008, pp. 57-68.

[23] Y. Lou, X. Zhang, S. Osher and A. Bertozzi, "Image recovery via non local operators," *Journal of Science Computer*, Vol. 42, No. 2, 2010, pp. 185-197.

[24] X. Zhang, M. Burger, X. Bresson and S. Osher, "Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction," *SIAM Journal Imag. Science*, Vol. 3, No. 3, 2010, pp. 253-276.

[25] W. Ma and S. Osher, "A TV Bregman iterative model of Retinex theory," Univ. California, Los Angeles, UCLA CAM Rep. 10-13, 2010.

[26] D. Zosso, G. Tran and S. Osher, "A Unifying Retinex Model Based on Non-local Differential Operators," Univ. California, Los Angeles, UCLA CAM Rep. 13-03, 2013.

[27] M. Jung, X. Bresson, T. F. Chan and L. A. Vese. "Nonlocal Mumford-Shah Regularizers for Color Image Restoration," *IEEE Transactions on Image Processing*, Vol. 20, No. 6, 2011, pp. 1583-1598. doi:10.1109/TIP.2010.2092433

[28] D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems," CPAM, Vol. XLII, 1989, pp. 577-685.

[29] P. Blomgren and T. Chan, "Color TV: Total Variation Methods for Restoration of Vector-Valued Images," *IEEE Transactions on Image Processing*, Vol. 7, No. 3, 1998, pp. 304-309. doi:10.1109/83.661180

[30] J. Yang, W. Yin, Y. Zhang and Y. Wang, "A Fast Algorithm for Edge-Preserving Variational Multichannel Image Restoration," *SIAM Journal on Imaging Sciences*, Vol. 2, No. 2, 2009, pp. 569-592. doi:10.1137/080730421

[31] Y. Yu, Z. Pan, W. Wei and J. Jiang, "Edge Preserving of Some Variational Models for Vectorial Image Denoising," *Joural of Graphics and Images*, Vol. 16, No. 12, 2011, pp. 2223-2230.

[32] T. Goldstein and S. Osher, "The Split Bregman Algorithm for L1 Regularized Problems," *SIAM Journal on Imaging Sciences*, Vol. 2, No. 2, 2009, pp. 323-343. doi:10.1137/080725891

[33] Q. Wang, Z. Pan, W. Wei, Z. Zhang, and C. Wang , "The Generalized MTV-L1 and its Split Bregman Algorithm for Noise Removal of Color Images with Textures," *in The Chn. Conf. Pattern Recognition (CCPR)*. 2010, pp. 1-6.