An Image Inpainting Model for Grayscale Images Based on TV-$H^{-1}$ Coupled with Perona-Malik Equation

Rajrup Banerjee$^{1,*}$, Rohit Kamal Chatterjee$^2$, Avijit Kar$^1$

$^1$Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
$^2$Department of Computer Science and Engineering, BIT Mesra, Ranchi, India.

How to cite this paper: Rajrup Banerjee, Rohit Kamal Chatterjee, Avijit Kar. (2023) An Image Inpainting Model for Grayscale Images Based on TV-$H^{-1}$ Coupled with Perona-Malik Equation. Journal of Applied Mathematics and Computation, 7(1), 152-161. DOI: 10.26855/jamc.2023.03.017

Received: March 15, 2023
Accepted: April 12, 2023
Published: May 6, 2023

*Corresponding author: Rajrup Banerjee, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.

Abstract

Image inpainting is the interpolation of missing or damaged portions of images employing information from the boundary and adjacent areas. Several fourth order Partial Differential Equation (PDE) based models are available in the literature to solve the inpainting problem, e.g., various Curvature Driven Diffusion methods, Cahn Hilliard Equation, TV-$H^{-1}$ etc. This paper presents a new fourth order PDE for image inpainting based on TV-$H^{-1}$ coupled with the Perona Malik equation. Perona and Malik proposed a nonlinear PDE to regulate diffusion by replacing the conductivity term with a function that enhances diffusion in homogenous regions but prohibits diffusion across edges in the image. An explicit-implicit numerical scheme is proposed in this paper based on splitting two convex energy terms, followed by Fourier spectral method on those, to solve the proposed PDE model. The results indicate that the proposed method generates better results in far less computational time than state-of-the-art methods.

Keywords

Image inpainting, Total Variation, Sobolev dual space, Cahn Hilliard equation, Perona Malik equation, Fourier spectral method

1. Introduction

Image inpainting is filling in missing parts of damaged images based on information extracted from surrounding areas. This problem can be considered an interpolation problem. The image inpainting problem has a wide range of applications, from restoring antique paintings to reducing specular reflections in biomedical images and many others.

Mathematically, image inpainting is the problem of reconstructing the image $u$ from a given damaged image $f$. The image domain is denoted by $\Omega$, and the damaged domain is denoted by $D \subset \Omega$, i.e., $D$ is a subset of the image domain $\Omega$.

Many mathematical inpainting models have been proposed in the last few decades, e.g., exemplar-based inpainting, stochastic, wavelet, and interpolation methods. But it was the PDE-based models which gained more popularity.

Bertalmio et al. [1] was the pioneer in this domain. They devised a nonlinear PDE model which propagated image information (the Laplacian of the image) in the direction of the sharp isophotes (lines of the same grey values, typically edges) continuously into the interior of the regions to be inpainted. This PDE model was named as transport inpainting model. In later work with Bertozzi [2], they found that the earlier method is closely related to 2-D fluid dynamics through the Navier–Stokes equation with the introduction of a diffusion term. This generated the idea that diffusion is required in the original inpainting problem. In practice, nonlinear diffusion [3], [4] gives excellent results in avoiding blurring edges...
in the inpainting.

After the work of Bertalmio et al. [1], more PDE-based models for image inpainting were devised, such as the Total Variation (TV) inpainting models [5] proposed by Chan and Shen, which were second-order PDEs. It was found that these image inpainting models could not connect the edges over longer distances or smoothly propagate isophotes into the damaged areas. Another third-order variational approach was devised, which was named Curvature Driven Diffusion (CDD) method [6], [7]. Still, it was found that this method may introduce artefacts in the isophotes along the boundary of the inpainting areas.

All these drawbacks in these models point to the fact that higher-order PDE models are needed for better inpainting performance. Consequently, several fourth-order PDE models like the Cahn-Hilliard model [8] and TV-H^{-1} model [9] are proposed for image inpainting.

This paper proposes another fourth-order PDE model for image inpainting based on TV-H^{-1} equation coupled with Perona–Malik equation. The numerical scheme for solving the proposed PDE model uses the idea of convex energy splitting, like the Cahn–Hilliard and TV-H^{-1} model scheme discussed in [8], [9], [10] and [11]. Lastly, the proposed model is tested on several grayscale images. The numerical experiments show that the proposed model produces better inpainting results with less computational time than the Cahn Hilliard Perona Malik model [10] for grayscale images. The results also suggest that the proposed model is better than many state-of-the-art PDE-based image inpainting models.

2. State of the Art

2.1 Cahn-Hilliard Inpainting Model

We start with the Cahn Hilliard Inpainting model discussed in [8], which is applicable only for black and white images, i.e., images with pixel values of either 0 or 1.

Let \( f(\bar{x}) \) where \( \bar{x} = (x, y) \), denote the image intensity function of the given image in the domain \( \Omega \), and let \( D \subset \Omega \) be the domain of inpainting. Let \( u(\bar{x}, t) \) evolve in time to become a fully inpainted version of \( f(\bar{x}) \in L^2(\Omega) \) under the following equation:

\[
\frac{\partial u}{\partial t} = \Delta \left( -\varepsilon \Delta u + \frac{1}{\varepsilon} W'(u) \right) + \lambda_0 \mathbb{1}_{\Omega \setminus D}(f - u) \tag{1}
\]

Where \( \mathbb{1}_{\Omega \setminus D}(\bar{x}) \) is the characteristic function of the complement of the inpainting domain

\[
\mathbb{1}_{\Omega \setminus D}(\bar{x}) = \begin{cases} 0 & \text{if } \bar{x} \in D \\ 1 & \text{if } \bar{x} \in \Omega \setminus D \end{cases} \tag{2}
\]

The constant \( \lambda_0 \gg 1 \) maintains the inpainted image close to the original image in \( \Omega \setminus D \). The function \( W(u) \) in Eq. (1) is a double well potential function with wells at \( u = 0 \) and \( u = 1 \), as binary images are only considered. In the current discussion, the double well potential function \( W(u) = u^2(u - 1)^2 \) is used, though, the use of other functions are also possible.

The Cahn-Hilliard equation can be perceived to be a \( H^{-1} \) the gradient flow of Helmholtz free energy functional \( E_1 \), and the \( L^2 \) gradient flow of the fidelity norm \( E_2 \)

\[
E[u] = E_1[u] + E_2[u] = \int_{\mathbb{R}^2} \frac{1}{2} |\nabla u|^2 + \frac{1}{\varepsilon} W(u) \, d\bar{x} + \frac{1}{2} \lambda_0 \mathbb{1}_{\Omega \setminus D} \int_D (f - u)^2 \, d\bar{x} \tag{3}
\]

From the Calculus of Variations, it is known that minimizing a functional \( E[u] \) of the function \( u(\bar{x}) \) is the exercise of finding the steady-state solution of the gradient flow equation

\[
\frac{\partial u}{\partial t} = -\delta E \frac{\delta E}{\delta u} \tag{4}
\]

Where \( \frac{\delta E}{\delta u} \) is the Functional Derivative or Variational Derivative of the Functional \( E[u] \) with respect to the function \( u(\bar{x}) \).

This is a PDE which we need to solve. But since, in Eq. (1), the two terms are gradient flows of different norms, one being \( H^{-1} \) and the other being \( L^2 \), we need to use the convexity splitting discussed in [8] for gradient flow.

The Cahn-Hilliard equation, though a fast equation, has some shortcomings. From the experimental results discussed in [8], it can be seen that the inpainted image does not contain smooth edges. This happens because of the term \( \Delta^2 u = \Delta^2 u = \nabla^4 u \) in Eq. (1). This fourth order derivative term makes the edges non-smooth. Also, Cahn-Hilliard is unable to deal with grayscale images, since the potential function is a double well potential.
2.2 TV-H^{-1} inpainting model

The Total Variation of a $C^1(\Omega)$ function $u$ defined on a bounded open set $\Omega \subseteq \mathbb{R}^n$ with boundary $\partial \Omega$ of class $C^1$ can be expressed as

$$V(u, \Omega) = \int_{\Omega} |\nabla u(\vec{x})| \, d\vec{x}$$

The Total Variation based noise removal algorithm proposed by Rudin et al. [4] for a given image $f$ is a $L^2$ gradient flow of the Total Variation functional, which, along with the regularizing term, generates the evolution equation

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{\nabla u}{|\nabla u|} \right) + \lambda_0 (f - u)$$

(6)

Since the Cahn-Hilliard inpainting model is only for binary images, a generalization of gray value images was made by Burger et al. [9], termed the TV-H^{-1} inpainting model. On similar lines with the Cahn -Hilliard equation, which is a $H^{-1}$ gradient flow of the HelmHoltz free energy functional, the TV -H-1 inpainting equation is derived as a $H^{-1}$ the gradient flow of the Total Variation functional, which, along with the regularizing term, generates the evolution equation

$$\frac{\partial u}{\partial t} = -\Delta \left( \nabla \left( \frac{u}{|\nabla u|} \right) \right) + \lambda_0 \mathbb{1}_{\Omega \setminus D} (f - u)$$

(7)

In practice, to avoid dividing by 0 in Eq. (6) and (7), $\sqrt{|\nabla u|^2 + \delta^2}$ is used instead of $|\nabla u|$. Thus the double well potential function has been dropped which makes the equation suitable for grayscale images. But, from the experimental results in [9], we can see that a better model is needed to make the edges even more smooth.

2.3 Perona-Malik Equation

The Perona-Malik Equation discussed in [3] replaces the thermal constant in the classical heat equation with a conductivity function. The equation thus becomes

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( g(|\nabla u|) \nabla u \right)$$

(8)

Typically, the conductivity functions used are

$$g(|\nabla u|) = \frac{1}{1 + \frac{|\nabla u|^2}{\varepsilon^2}}$$

(9)

and

$$g(|\nabla u|) = e^{-\frac{|\nabla u|^2}{\varepsilon^2}}$$

(10)

Thus, diffusion will be lower across high gradient regions, and higher across low gradient regions. Since a high gradient across a line means the line is between two different objects or an object and a background, this essentially means diffusion will be higher in regions inside a logical object or background and lower across an object.

2.4 Cahn-Hilliard Perona-Malik Equation

The Cahn-Hilliard Perona-Malik equation proposed in [10] incorporates the Perona-Malik conductivity function in Eq. (1) thus making it

$$\frac{\partial u}{\partial t} = -\Delta \left( \varepsilon \nabla \cdot \left( g(|\nabla u|) \nabla u \right) \right) + \frac{1}{\varepsilon} \Delta \left( W'(u) \right) + \lambda_0 \mathbb{1}_{\Omega \setminus D} (f - u)$$

(11)

Equation (11) can be perceived to be the $H^{-1}$ gradient flow of a free energy functional $E_1$, and the $L^2$ gradient flow of the fidelity norm $E_2$

$$E[u] = E_1[u] + E_2[u] = \int_{\Omega} \frac{\varepsilon}{2} \mathcal{L}(|\nabla u|) + \frac{1}{\varepsilon} W(u) \, d\vec{x} + \frac{1}{2} \lambda_0 \mathbb{1}_{\Omega \setminus D} \int_{\Omega} (f - u)^2 \, d\vec{x}$$

(12)

Where $\mathcal{L}(|\nabla u|)$ is a function such that

$$\nabla_{H^{-1}} \left( \int_{\Omega} \mathcal{L}(|\nabla u|) \, d\vec{x} \right) = \Delta \left( \nabla \cdot \left( g(|\nabla u|) \nabla u \right) \right)$$

(13)

But due to the presence of the double well potential function $W(u)$, Eq. (11) still remains valid for binary or only black and white images. To use it for grayscale images, the image needs to be sliced at every bit of the grayvalue and the inpainting algorithm needs to be run at every bit plane and then the results need to be assembled to get the grayscale output. This makes the algorithm considerably slow.
3. Proposed model and numerical scheme - TV-H\(^{-1}\) Coupled with Perona-Malik Equation

The proposed model is TV-H\(^{-1}\) inpainting equation coupled with Perona-Malik conductivity function thus generating the equation

\[
\frac{\partial u}{\partial t} = -\nabla \left( \nabla \left( g( |\nabla u|) \frac{\nabla u}{|\nabla u|} \right) \right) + \lambda_0 \mathbb{1}_{\partial \Omega}(f - u)
\]  

(14)

The boundary conditions in Eq. (14) are set to a natural zero Neumann boundary condition

\[
\nabla u \cdot \vec{n} = 0 \quad \text{on } \partial \Omega
\]  

(15)

To solve Eq. (14) numerically, the idea of convex energy splitting is used. The function \(\mathcal{L}(x)\) is defined such that

\[
\mathcal{L}( |\nabla u|) \nabla u \cdot \vec{n} = \nabla \left( g( |\nabla u|) \frac{\nabla u}{|\nabla u|} \right)
\]  

(16)

For different \(g(|\nabla u|)\), there will be different \(\mathcal{L}( |\nabla u|)\).

Next, the energy functionals are defined as

\[
E_1 = \int_{\Omega} \mathcal{L}( |\nabla u|) \, d\vec{x}
\]

\[
E_2 = \frac{\lambda_0}{2} \int_{\partial \Omega} (f - u)^2 \, d\vec{x}
\]  

(17)

Thus, Eq. (14) can be treated as

\[
\frac{\partial u}{\partial t} = -\nabla \mathcal{L}_{-1} (E_1) - \nabla L^2 (E_2)
\]  

(18)

The idea of convexity splitting is used to split \(E_1\) into two convex energies \(E_{11}\) and \(E_{12}\) such that

\[
E_1 = E_{11} - E_{12}
\]

\[
E_{11} = \int_{\Omega} \frac{C_1}{2} |\nabla u|^2 \, d\vec{x}
\]

\[
E_{12} = \int_{\Omega} \frac{C_1}{2} |\nabla u|^2 - \mathcal{L}( |\nabla u|) \, d\vec{x}
\]  

(19)

Also \(E_2\) is split into two convex energies \(E_{21}\) and \(E_{22}\) such that

\[
E_2 = E_{21} - E_{22}
\]

\[
E_{21} = \int_{\partial \Omega} \frac{C_2}{2} u^2 \, d\vec{x}
\]

\[
E_{22} = \int_{\partial \Omega} \frac{\lambda_0}{2} (f - u)^2 + \frac{C_2}{2} u^2 \, d\vec{x}
\]  

(20)

All the energies \(E_{11}, E_{12}, E_{21}\) and \(E_{22}\) must be convex. But since the convexity of \(E_{12}\) and \(E_{22}\) cannot be guaranteed, two constants \(C_1 > 0\) and \(C_2 > 0\) must be introduced so that it becomes possible to make \(E_{12}\) and \(E_{22}\) convex. From Eq. (17), (18), (19) and (20), it is to be noted that \(C_1 > \frac{1}{\delta}\) and \(C_2 \geq \lambda_0\) are conditions for convexity.

Using the above convexity splitting, the time-stepping scheme with time step-size \(\delta t\) can be formulated as

\[
\frac{u^{n+1} - u^n}{\delta t} = -\nabla \mathcal{L}_{-1} (E_{11}^{n+1} - E_{12}^{n+1}) - \nabla L^2 (E_{21}^{n+1} - E_{22}^{n+1})
\]  

(21)

which leads to

\[
\frac{u^{n+1} - u^n}{\delta t} + \nabla \mathcal{L}_{-1} (E_{11}^{n+1}) + \nabla L^2 (E_{21}^{n+1}) = \nabla \mathcal{L}_{-1} (E_{12}^{n}) + \nabla L^2 (E_{22}^{n})
\]  

(22)

Substituting \(E_{11}, E_{12}, E_{21}\) and \(E_{22}\) in Eq. (22), it becomes

\[
\frac{u^{n+1} - u^n}{\delta t} + C_1 \Delta^2 u^{n+1}(\vec{x}) + C_2 \mathbb{1}_{\partial \Omega}(u^{n+1}(\vec{x})) = C_1 \Delta^2 u^n(\vec{x}) - \Delta \left( \nabla \left( g( |\nabla u|) \frac{\nabla u}{|\nabla u|} \right) \right) + C_2 \mathbb{1}_{\partial \Omega}(u^n(\vec{x})) + \lambda_0 \mathbb{1}_{\partial \Omega}(f(\vec{x}) - u^n(\vec{x}))
\]  

(23)
Multiplying both sides of Eq. (23) by $\delta t$ and rearranging all $u^{n+1}$ terms to the left side, it becomes

$$u^{n+1}(\bar{x}) + C_1 \delta t D^2 u^{n+1}(\bar{x}) + C_2 \delta t \mathbf{1}_{\Omega \setminus D} u^{n+1}(\bar{x}) = \mathbf{\nabla} \cdot \left( g(\mathbf{\nabla} u^n) \right) + C_2 \delta t \mathbf{1}_{\Omega \setminus D} u^n(\bar{x}) + \lambda_0 \delta t \mathbf{1}_{\Omega \setminus D} (f(\bar{x}) - u^n(\bar{x})) + u^n(\bar{x})$$

(24)

To solve Eq. (24), Fourier Spectral Method will be used. Taking two-dimensional Discrete Fourier Transform of Eq. (24), it becomes

$$\tilde{u}_{ij}^{n+1} + C_1 \delta t L_{ij} \tilde{u}_{ij}^{n+1} + C_2 \delta t \mathbf{1}_{\Omega \setminus D} \tilde{u}_{ij}^{n+1} = C_1 \delta t L_{ij} \tilde{u}_{ij}^n - \delta t L_{ij} \left( \mathbf{\nabla} \cdot \left( g(\mathbf{\nabla} u^n) \right) \right)_{ij}$$

(25)

$$\text{Where the fact that}$$

$$\tilde{u} = \hat{L} u$$

(26)

has been used, where $L$ is the matrix of the two-dimensional DFT of the Discrete Laplacian as discussed in [8], [12], and [13].

If $u$ be the image of size $M \times N$, then $L$ is the matrix of size $M \times N$ with the elements being

$$l_{i,j} = M^2 \left( e^{\frac{2\pi i j}{M}} + e^{-\frac{2\pi i j}{M}} \right) + N^2 \left( e^{\frac{2\pi i i}{N}} + e^{-\frac{2\pi i i}{N}} \right) = M^2 \left( 2 \cos \frac{2\pi i}{M} - 2 \right) + N^2 \left( 2 \cos \frac{2\pi i}{N} - 2 \right)$$

(27)

Where $I = \sqrt{-1}$ is the imaginary unit.

Rearranging terms in Eq. (25) and expressing $\tilde{u}_{ij}^{n+1}$ in terms of $\tilde{u}_{ij}$, it becomes

$$\tilde{u}_{ij}^{n+1} = \frac{C_1 \delta t L_{ij} \tilde{u}_{ij} - \delta t L_{ij} \left( \mathbf{\nabla} \cdot \left( g(\mathbf{\nabla} u^n) \right) \right)_{ij}}{1 + C_1 \delta t L_{ij} + C_2 \delta t \mathbf{1}_{\Omega \setminus D}}$$

(28)

Equation (28) will be used as the iteration scheme. The iterations will be done in the Fourier domain and then two-dimensional inverse DFT is to be taken to extract $u_{ij}^{n+1}$ in the spatial domain.

4. Experimental Results and Discussion

Some experimental results will be discussed using the proposed image inpainting model for grayscale images. The results will be compared against other popular PDE-based image inpainting methods like the Transport Inpainting model proposed by Bertalmio et al. [1] and the TV-H' model proposed by Burger et al. [9].

The Transport Inpainting model solves the following PDE for image inpainting

$$\frac{\partial u}{\partial t} = \nabla \perp u + \epsilon \mathbf{\nabla} \cdot (g(\mathbf{\nabla} u) \mathbf{\nabla} u)$$

(29)

where $\epsilon$ is a small parameter, i.e. $0 < \epsilon \ll 1$. The operator $\nabla \perp$ denotes the perpendicular gradient, which in the two-dimensional case becomes $\left( \frac{\partial u}{\partial y}, \frac{\partial u}{\partial x} \right)$.

The numerical scheme is to iterate the smoothness transport for $M$ times, then iterate the Perona-Malik equation for $N$ times and repeat the above process until a stable solution is reached.

The TV-H' inpainting model was explained in Eq. (7), but still it is mentioned here for continuity.

In the TV-H' model, the following equation is being solved

$$\frac{\partial u}{\partial t} = -\Delta \left( g \left( \frac{u}{\sqrt{[u^n]'^2 + \delta^2}} \right) \right) + \lambda_0 \mathbf{1}_{\Omega \setminus D} (f - u)$$

(30)

where $\delta$ is a small parameter, i.e. $0 < \delta \ll 1$.

The results have been compared quantitatively based on four metrics: Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Relative $L^1$ Error as discussed in [14]. The results shown are experimental. Better results might be obtained by fine-tuning the parameters.

The metrics are defined first.

Let $I_1$ and $I_2$ be two images having the same dimensions, $P$ and $Q$, where $I_1$ is the original image and $I_2$ is the inpainted version of $I_1$.

The Mean Square Error, or MSE is defined as
\[ \text{MSE} = \frac{1}{PQ} \sum_{i,j} (I_1(i,j) - I_2(i,j))^2 \]

PSNR is defined as
\[ \text{PSNR} = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}} \]

where MAX is the maximum possible pixel value of the images.

SSIM is defined as
\[ \text{SSIM} = \frac{(2\mu_{I_1}\mu_{I_2} + C_1)(2\sigma_{I_1,I_2} + C_2)}{\mu_{I_1}^2 + \mu_{I_2}^2 + C_1(\sigma_{I_1}^2 + \sigma_{I_2}^2 + C_2)} \]

where \( \mu \) and \( \sigma \) are the mean and variance, respectively. The two constants \( C_1 \) and \( C_2 \) are defined as \( C_1 = (0.01L)^2 \) and \( C_2 = (0.03L)^2 \), \( L \) being the maximum pixel value of the images.

Relative \( L^2 \) error is defined as
\[ \text{Error} = \frac{\|I_2 - I_1\|_2}{\|I_1\|_2} \]

Fig 1. Three grayscale testing images.

Table 1 contains the parameter settings used to run the inpainting algorithms. We can have better results by tweaking the parameters for individual images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>Parameters settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman #1</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 800, C_2 = \lambda = 350 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
<tr>
<td>Cameraman #2</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 500, C_2 = \lambda = 280 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
<tr>
<td>Lena #1</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 650, C_2 = \lambda = 250 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
<tr>
<td>Lena #2</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 500, C_2 = \lambda = 250 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
<tr>
<td>Peppers #1</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 800, C_2 = \lambda = 300 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
<tr>
<td>Peppers #2</td>
<td>Transport</td>
<td>( \varepsilon = 10^{-10}, M = 40, N = 2 )</td>
</tr>
<tr>
<td></td>
<td>TV-H(^1)</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 5 )</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona-Malik</td>
<td>( \varepsilon = [200, 0.8], C_1 = 500, C_2 = \lambda = 70 )</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H(^1) Perona Malik</td>
<td>( \delta = 0.01, C_1 = 1000, C_2 = \lambda = 10, k = 10 )</td>
</tr>
</tbody>
</table>
The proposed model was tested using commonly used three standard testing images: Cameraman, Lena, and Peppers, shown in Fig 1. Two ways were used to damage the original images. Then the inpainting algorithms were executed to inpaint the damaged images. The results of the experiments are shown in Fig 2.

Table 2 contains the quantitative report of inpainting. It can be seen that our proposed model generates better results than the accepted PDE models, and is generally faster too.

### Table 2. Report for Image Inpainting

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>Mean Square Error</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Relative L² Error</th>
<th>CPU Time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman #1</td>
<td>Transport</td>
<td>0.011</td>
<td>19.5683</td>
<td>0.6966</td>
<td>0.0839</td>
<td>222.063</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0029</td>
<td>24.1291</td>
<td>0.9191</td>
<td>0.0462</td>
<td>109.873</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0038</td>
<td>24.2502</td>
<td>0.8652</td>
<td>0.0503</td>
<td>337.998</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.0022</td>
<td>26.6743</td>
<td>0.9248</td>
<td>0.0452</td>
<td>110.377</td>
</tr>
<tr>
<td>Cameraman #2</td>
<td>Transport</td>
<td>0.0104</td>
<td>19.8391</td>
<td>0.6778</td>
<td>0.0799</td>
<td>226.034</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0019</td>
<td>25.4859</td>
<td>0.9248</td>
<td>0.0279</td>
<td>117.229</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0033</td>
<td>24.8516</td>
<td>0.8826</td>
<td>0.041</td>
<td>342.74</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.0016</td>
<td>27.9879</td>
<td>0.944</td>
<td>0.0261</td>
<td>113.8</td>
</tr>
<tr>
<td>Lena #1</td>
<td>Transport</td>
<td>0.0128</td>
<td>18.9296</td>
<td>0.6364</td>
<td>0.1161</td>
<td>226.198</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0018</td>
<td>26.1292</td>
<td>0.9061</td>
<td>0.0395</td>
<td>101.974</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0034</td>
<td>24.7148</td>
<td>0.8364</td>
<td>0.0538</td>
<td>348.554</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.0017</td>
<td>27.7565</td>
<td>0.9141</td>
<td>0.0383</td>
<td>122.231</td>
</tr>
<tr>
<td>Lena #2</td>
<td>Transport</td>
<td>0.0115</td>
<td>19.4052</td>
<td>0.6302</td>
<td>0.1021</td>
<td>226.318</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0036</td>
<td>24.3243</td>
<td>0.9081</td>
<td>0.0486</td>
<td>102.881</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0052</td>
<td>22.8038</td>
<td>0.8338</td>
<td>0.0538</td>
<td>340.31</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.0034</td>
<td>24.6963</td>
<td>0.9141</td>
<td>0.0473</td>
<td>126.867</td>
</tr>
<tr>
<td>Peppers #1</td>
<td>Transport</td>
<td>0.0115</td>
<td>19.3851</td>
<td>0.6519</td>
<td>0.0846</td>
<td>226.669</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0018</td>
<td>27.4839</td>
<td>0.9031</td>
<td>0.0549</td>
<td>108.629</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0025</td>
<td>26.0352</td>
<td>0.8737</td>
<td>0.0362</td>
<td>345.192</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.0014</td>
<td>28.5131</td>
<td>0.9206</td>
<td>0.0359</td>
<td>107.088</td>
</tr>
<tr>
<td>Peppers #2</td>
<td>Transport</td>
<td>0.0153</td>
<td>18.1539</td>
<td>0.6002</td>
<td>0.0986</td>
<td>226.457</td>
</tr>
<tr>
<td></td>
<td>TV-H⁻¹</td>
<td>0.0011</td>
<td>29.6866</td>
<td>0.9365</td>
<td>0.0215</td>
<td>102.945</td>
</tr>
<tr>
<td></td>
<td>Cahn-Hilliard Perona Malik</td>
<td>0.0025</td>
<td>26.0724</td>
<td>0.8857</td>
<td>0.0314</td>
<td>344.587</td>
</tr>
<tr>
<td></td>
<td>Proposed TV-H⁻¹ Perona Malik</td>
<td>0.001</td>
<td>30.0634</td>
<td>0.9506</td>
<td>0.0205</td>
<td>113.724</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we introduced a new image inpainting model based having TV-H⁻¹ coupled with Perona–Malik equation. We used a convex energy splitting scheme to solve the proposed PDE model numerically. We then used Fourier spectral method to implement the numerical scheme. We executed the proposed model on commonly used grayscale images. The numerical experiments show that the proposed inpainting model generally works better than most of the accepted PDE models for image inpainting.
Fig 2. Inpainting Results with different algorithms.
References


Appendix A: Function Spaces

\( BV(\Omega) \quad u \in L^1(\Omega) : V(u, \Omega) < +\infty \)

With \( 1 \leq p < \infty \) : Sobolev space of functions \( f \in L^p(\Omega) \) such that all derivatives up to \( p \)th order belong to \( L^p(\Omega) \). The space \( W^{p,q}(\Omega) \) is a Banach space with a norm

\[ \| f \|_{W^{p,q}(\Omega)} = \left( \sum_{k=1}^{p} \int_{\Omega} |D^k f|^q \, dx \right)^{\frac{1}{q}} , \]

where \( D^k \) denotes the \( k \)-th distributional derivative of \( f \)

\( W^p_0(\Omega) \quad \{ f \in W^p(\Omega) : f|_{\partial \Omega} = 0 \} \)

\( W^{p,2}(\Omega) \). This is a Hilbertian Sobolev space with a corresponding inner product.

\( H^p(\Omega) \quad \langle f, g \rangle_{H^p(\Omega)} = \sum_{k=0}^{p} \int_{\Omega} D^k f \cdot D^k g \, dx \) . For this special Sobolev space, we write \( \| \cdot \|_{H^p(\Omega)} \triangleq \| \cdot \|_{W^{p,2}(\Omega)} \) for its corresponding norm.

\( H^1(\Omega) \quad W^{1,2}(\Omega) \)

\( H^1_0(\Omega) \quad (H^1_0(\Omega))^* \) i.e., the dual space of \( H^1_0(\Omega) \) with corresponding norm \( \| \cdot \|_{H^{-1}(\Omega)} = \| \cdot \|_{(H^1_0(\Omega))^*} \) and the inner product \( \langle \cdot, \cdot \rangle_{H^{-1}(\Omega)} = \langle \cdot, \cdot \rangle_{(H^1_0(\Omega))^*} \). Thus \( H^{-1}(\Omega) \) is the inverse of the negative Laplacian \( -\Delta \) with zero Dirichlet boundary conditions.

\( \langle \cdot, \cdot \rangle_{-1} \quad \| \cdot \|_{H^{-1}(\Omega)} \)

Appendix B: The Space \( H^{-1} \) and the Inverse Laplacian \( \Delta^{-1} \)

For functions \( f, g \in H^{-1}(\Omega) \) the norm and inner product are defined as

\[ \| f \|_{-1}^2 = \| \nabla^{\Delta^{-1}} f \|_2^2 = \int_{\Omega} (\nabla^{\Delta^{-1}} f)^2 \, dx \]

\[ \langle f, g \rangle_{-1} = \langle \nabla^{\Delta^{-1}} f, \nabla^{\Delta^{-1}} g \rangle_2 = \int_{\Omega} (\nabla^{\Delta^{-1}} f) \cdot (\nabla^{\Delta^{-1}} g) \, dx \]

The operator \( \Delta^{-1} \) denotes the inverse of the negative Dirichlet Laplacian, i.e., \( u = \Delta^{-1} f \) is the unique solution to

\[ \begin{align*}
-\Delta u &= f & \text{in } \Omega \\
u &= 0 & \text{on } \partial \Omega
\end{align*} \]

Appendix C: \( H^{-1} \) Gradient of a Functional

Let \( u(\vec{x}) \) where \( \vec{x} = (x, y) \) be a function in a domain \( \Omega \)

We know that in 2 dimensions, the \( L^2 \) gradient of the functional \( J[u] = \int_{\Omega} L(\vec{x}, u(\vec{x}), \nabla u(\vec{x})) \, d\vec{x} \) is

\[ \frac{\delta J}{\delta u} = \nabla_J [u] = \nabla [u] = \frac{\partial L}{\partial u} - \nabla \cdot \frac{\partial L}{\partial \nabla u} = \frac{\partial L}{\partial u} - \frac{\partial \partial L}{\partial x \, \partial u_x} - \frac{\partial \partial L}{\partial y \, \partial u_y} \]

Based on the discussion above, we come up with the \( H^{-1} \) gradient of the functional \( J[u] \)

\[ \nabla_{H^{-1}} J[u] = -\Delta J[u] \]